

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

MATHEUS PLAUTZ PRESTES

**ESTUDO EXPLORATÓRIO SOBRE DESIGN THINKING NO
DESENVOLVIMENTO DE SOFTWARE**

Porto Alegre
2020

PÓS-GRADUAÇÃO - *STRICTO SENSU*



Pontifícia Universidade Católica
do Rio Grande do Sul

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**ESTUDO EXPLORATÓRIO
SOBRE DESIGN THINKING NO
DESENVOLVIMENTO DE
SOFTWARE**

MATHEUS PLAUTZ PRESTES

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientadora: Profa. Sabrina Marczak, PhD

**Porto Alegre
2020**

Ficha Catalográfica

P936e Prestes, Matheus Plautz

Estudo Exploratório sobre Design Thinking no Desenvolvimento de Software / Matheus Plautz Prestes . – 2020.

189 p.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Sabrina dos Santos Marczak.

1. Design Thinking. 2. Engenharia de Software. 3. Desenvolvimento de Software. 4. Mapeamento Sistemático da Literatura. 5. Survey. I. Marczak, Sabrina dos Santos. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).

Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

MATHEUS PLAUTZ PRESTES

**ESTUDO EXPLORATÓRIO SOBRE DESIGN THINKING NO
DESENVOLVIMENTO DE SOFTWARE**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 06 de Março de 2020.

BANCA EXAMINADORA:

Profa. Dra. Edna Dias Canedo (UNB)

Prof. Dr. Afonso Henrique Corrêa de Sales (PUCRS)

Profa. Sabrina Marczak, PhD (PUCRS - Orientadora)

ESTUDO EXPLORATÓRIO SOBRE DESIGN THINKING NO DESENVOLVIMENTO DE SOFTWARE

RESUMO

Na visão do usuário e requisito, a Engenharia de Requisitos e abordagens ágeis vieram para auxiliar e melhorar à análise dos requisitos, validação e aproximação com o consumidor final. Sobre diversas formas, existe o *Design Thinking*, que busca através de princípios do *design*, entender as necessidades dos usuários e ajuda na validação de soluções de forma iterativa. Esta abordagem despertou o interesse através de sua forma de co-criação sobre equipes multidisciplinares e é um receptor para inovação, mas ainda conhece-se pouco sobre a sua aplicação no desenvolvimento de software. Esta dissertação apresenta uma pesquisa que foi realizada em dois estudos: um Mapeamento Sistemático na Literatura (Estudo 1) e uma *Survey* (Estudo 2), na qual formam um corpo de conhecimento e caracterizam o uso do *Design Thinking* na Engenharia de Software, tanto na visão da literatura como na indústria, entendendo seus aspectos relacionados ao uso de sua abordagem, cenários utilizados, seus benefícios e dificuldades de aplicação. O Mapeamento da Literatura identificou 115 artigos os quais apresentaram 19 modelos de processo de *Design Thinking*, 46 técnicas e outras informações de contexto de uso no desenvolvimento de software, posteriormente usados como fonte para o *design* do instrumento de coleta da *Survey*. A *Survey*, por sua vez, obteve 158 respostas, coletadas durante 2 meses e respondidas por profissionais que trabalham com *Design Thinking* na indústria. Um corpo de conhecimento foi construído para entender o *Design Thinking* na Engenharia de Software para caracterizar o uso de *Design Thinking* no desenvolvimento de software.

Palavras-Chave: Design Thinking, Engenharia de Software, Desenvolvimento de Software, Mapeamento Sistemático da Literatura, Survey.

EXPLORATORY STUDY ON DESIGNING THINKING IN SOFTWARE DEVELOPMENT

ABSTRACT

In the view of the user and requirement, Requirements Engineering and approaches are provided to assist and improve requirements analysis, validation and approximation with the final consumer. In several ways, there is Design Thinking, which seeks through design principles, understand how users' requests and help in validating solutions in an iterative way. This approach sparked interest through its form of co-creation on multidisciplinary teams and is a receiver for innovation, but little is known about its application in software development. This dissertation presents a research that was carried out in two studies: a Systematic Mapping in Literature (Study 1) and a Survey (Study 2), in which they form a body of knowledge and characterize the use of Design Thinking in Software Engineering, both in view of literature as in the industry, understanding its aspects related to the use of its approach, scenarios used, its benefits and application difficulties. The Literature Mapping identified 115 articles which presented 19 models of Design Thinking process, 46 techniques and other context information for use in software development, later used as a source for the design of the Survey collection instrument. The Survey, in turn, obtained 158 responses, collected over 2 months and answered by professionals working with Design Thinking in the industry. A body of knowledge was built to understand Design Thinking in Software Engineering to characterize the use of Design Thinking in software development.

Keywords: Design Thinking, Software Engineering, Software Development, Systematic Mapping Review, Survey.

LISTA DE FIGURAS

Figura 1.1 – Metodologia de Pesquisa	13
Figura 3.1 – Processo de Mapeamento Sistemático da Literatura	19
Figura 3.2 – Processo KAPPA	26
Figura 3.3 – Resultado do Processo de Extração	27
Figura 3.4 – Quantidade de Artigos por Ano	28
Figura 3.5 – Propósitos de Uso do DT	30
Figura 3.6 – Uso do DT em Diversos Contexto	33
Figura 3.7 – Modelo Sandino et al.	35
Figura 3.8 – Modelo Nordstrom	37
Figura 3.9 – Modelo HPI	38
Figura 3.10 – Modelo Diving Drive	40
Figura 3.11 – Modelo <i>Stanford University's + HPI</i>	42
Figura 3.12 – Modelo <i>Stanford University's</i>	44
Figura 3.13 – Modelo Meinel e Leifer - (HPI)	45
Figura 3.14 – Modelo IBM-DT	47
Figura 4.1 – Processo da <i>Survey</i>	62
Figura 4.2 – Lacunas do MSL x <i>Survey</i>	65
Figura 4.3 – Estrutura do Questionário	66
Figura 4.4 – Processo de Validação do Questionário	68
Figura 4.5 – Processo de Distribuição	71
Figura 4.6 – Contagem x Dias de Dezembro	72
Figura 4.7 – Contagem x Dias até Janeiro	73
Figura 4.8 – Contagem x Horário de Respostas	74
Figura 4.9 – Respondentes por Região	75
Figura 4.10 – Experiência em Anos Sobre os Profissionais	76
Figura 4.11 – Experiência em Anos Sobre os Profissionais com Maior Seleção Entre as Duas Opções	77
Figura 4.12 – Resultado - Modelos Mais Referenciados	80
Figura 4.13 – Modelo DT da ECHOS	81
Figura 4.14 – Técnicas mais Utilizadas	82
Figura 4.15 – Resultado do Grau de Dificuldade	85
Figura 4.16 – Resultado dos Softwares que Apoiam os Profissionais	86

Figura 5.1 – Valores Ágeis x Características DT	100
Figura 5.2 – Cenário do DT no Desenvolvimento de Software	105
Figura 5.3 – Caracterização: Modelo Sandino et al. [101]	107
Figura 5.4 – Caracterização: Modelo HPI	108
Figura 5.5 – Caracterização: Modelo <i>Stanford University's D.School</i>	109
Figura 5.6 – Caracterização: Modelo Nordstrom	110
Figura 5.7 – Caracterização: Modelo <i>Diving Board</i>	110
Figura 5.8 – Caracterização: Modelo HCAW	111
Figura 5.9 – Caracterização: Modelo IBM	112
Figura 5.10 – Caracterização: Modelo CoDICE	113
Figura 5.11 – Simplificação de Técnicas e Ferramentas do DT	115
Figura 5.12 – Benefícios e Desafios do DT	117
Figura 5.13 – Trabalho Correlato: Processo do Mapeamento Sistemático	119
Figura 5.14 – Trabalho Correlato: Modelos de <i>Design Thinking</i> e suas Fases	120
Figura 5.15 – Trabalho Correlato: Técnicas de <i>Design Thinking</i> mais Citadas	121
Figura 6.1 – Síntese das Respostas do Estudo 1 e Estudo 2	125

LISTA DE TABELAS

Tabela 3.1 – <i>String</i> de Busca	22
Tabela 3.2 – Critérios de Inclusão e Exclusão	23
Tabela 3.3 – Resultados da Pesquisas nas Fontes de Dados	24
Tabela 3.4 – Classificação KAPPA	25
Tabela 3.5 – Execução do Processo de Mapeamento	26
Tabela 3.6 – Pesquisadores com Maior Número de Estudos Publicados	28
Tabela 3.7 – Conferências x Publicações	29
Tabela 3.8 – <i>Agile</i> x DT	33
Tabela 3.9 – Estudos que Trabalharam o DT Integrado com Outro Contexto	34
Tabela 3.10 – DT em Outros Contextos	34
Tabela 3.11 – Técnicas de DT mais Citadas	49
Tabela 3.12 – Modelos de DT	58
Tabela 3.13 – Benefícios e Desafios/Limitações do DT	61
Tabela 4.1 – Resultado Escolha das Técnicas	84
Tabela 4.2 – Modelos com Mais Relações com as Técnicas	88
Tabela 4.3 – Propósito de Uso do DT	91
Tabela 4.4 – Cenários de Uso mais Selecionados	93
Tabela 5.1 – Estudo 1 x Trabalho Correlato	122

LISTA DE SIGLAS

- BMC – *Business Model Canvas*
- DT – *Design Thinking*
- ER – Engenharia de Requisitos
- ES – Engenharia de Software
- FDD – *Feature-Driven Development*
- HCAW – *Human-Centered Agile Workflow*
- HCD – *Human-Centred Design*
- ICSE – *International Conference on Software Engineering*
- IHC – Interação Humano-Computador
- MSL – Mapeamento Sistemático da Literatura
- QP – Questão de Pesquisa
- RAD – *Rapid Application Development*
- TDC – *The Developer's Conference*
- UCD – *User-Centered Design*
- UX – *User Experience*
- XP – *Extreme Programming*

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivo Geral e Objetivos Específicos	12
1.2	Visão Geral da Metodologia de Pesquisa	12
1.3	Contribuição	13
1.4	Estrutura do Trabalho	14
2	REFERENCIAL TEÓRICO	15
2.1	Design Thinking	15
2.2	Design Thinking no Desenvolvimento de Software	17
3	MAPEAMENTO SISTEMÁTICO DA LITERATURA	19
3.1	Mapeamento Sistemático da Literatura Segundo Petersen e Colegas	19
3.2	Planejamento e Realização do Mapeamento	20
3.3	Resultado do Mapeamento Sistemático	27
3.3.1	QP1: Propósito de uso do DT na Engenharia de Software	29
3.3.2	QP2: Contexto de uso do DT no desenvolvimento de software	32
3.3.3	QP3: Modelos, métodos, técnicas e/ou abordagens utilizadas do DT na Engenharia de software	34
3.3.4	QP4: Ferramentas de apoio no DT para o desenvolvimento de software?	58
3.3.5	QP5: Seleção de uso dos Modelos, métodos, técnicas, abordagens e ferramentas no DT	59
3.3.6	QP6: Recomendações e dificuldades do DT no desenvolvimento de software	60
4	SURVEY	62
4.1	<i>Survey</i> segundo Kitchenham e Pfleeger	62
4.2	Planejamento e Realização da <i>Survey</i>	63
4.3	Coleta e Análise dos Resultados	70
4.4	Resultados da <i>Survey</i>	78
4.4.1	Bloco 1: Modelos, Técnicas, Ferramentas: Grau de Dificuldades de escolha (QP3, QP4 e QP5)	79
4.4.2	Bloco 2: Propósito e Cenários de uso: Benefícios e Dificuldades (QP1, QP2 e QP6)	90

5	DISCUSSÃO	99
5.1	Caracterização do DT na Engenharia de Software	99
5.2	Entendendo o Propósito de Uso e Diferentes Cenários (QP1 e QP2)	103
5.3	Modelos, Técnicas e Ferramentas do DT (QP3, QP4 e QP5)	105
5.4	Benefícios e Desafios no uso do Design Thinking (QP6)	115
5.5	Estudo Correlato	117
6	CONSIDERAÇÕES FINAIS	123
6.1	Conclusão	123
6.2	Limitações	126
6.3	Trabalhos Futuros	126
	REFERÊNCIAS	127
	APÊNDICE A – Lista de Artigos Seleccionados	138
	APÊNDICE B – SURVEY - Instrumento de Coleta	145
	APÊNDICE C – SURVEY - Respostas	157

1. INTRODUÇÃO

Segundo Pressman e Maxim [94], o desenvolvimento de software, em todas as suas formas e em todos os seus campos de aplicação, deve passar pelos processos de engenharia. Baseado na definição da IEEE [96], estes autores [94] argumentam que a Engenharia de Software (ES) é uma disciplina organizada em camadas—ferramentas, métodos e processos, direcionadas pelo foco na qualidade do que é desenvolvido. A camada de processos, realizada através de metodologias, atividades, e práticas diversas, torna-se uma camada essencial, garantindo o gerenciamento do ciclo de desenvolvimento e estabelecendo um contexto para a garantia da qualidade.

A implementação de um processo para o desenvolvimento de software relaciona-se à alguns critérios gerenciáveis, que visam a preocupação associada à qualidade, a possíveis mudanças nos requisitos do cliente durante o desenvolvimento, ao tempo e ao custo. Com isso, modelos de ciclo de vida são criados a procura de tornarem-se facilitadores, afetando de forma significativa estas diversas preocupações na criação de um produto de software, tornando-se assim um processo diretamente pertinente ao resultado final, onde muitas vezes quando o processo é considerado débil em sua natureza, o produto final se torna inadimplente sobre o escopo estabelecido. A realidade traz fatores desafiantes, projetos ultrapassam o orçamento, são finalizados com errôneas recorrências em entregas, fugindo das expectativas do usuário e aos requisitos do sistema [57]. Porém, a aproximação dos usuários ao longo do desenvolvimento do software tornou-se um fator factível para a projeção de software que atenda não apenas os requisitos do produto, mas também às expectativas do usuário [18].

Por exemplo, a Engenharia de Requisitos (ER), uma das disciplinas da ES, busca auxiliar no levantamento, análise, especificação e validação dos requisitos bem como no gerenciamento de mudanças destes durante o desenvolvimento. Entretanto, a indústria de software tem aderido a abordagens *lights* (e.g., ágeis), para melhor atender às demandas do mercado consumidor, ou seja, atualizações e inserção de novas funcionalidades rapidamente para sobreviver a acirrada competitividade. Assim, os métodos ágeis permitem um rápido levantamento de requisitos associado a validações concomitantes pelo engajamento mais próximo aos usuários e pela filosofia de desenvolvimento em curtos ciclos iterativos. Mudanças são então mais rápidas [50].

Neste contexto, a ES como um todo, incluindo a ER, tem buscado novas formas de melhorar a definição do escopo do software e enriquecer a qualidade do produto entregável. O *Design Thinking* (DT), por exemplo, é uma abordagem que vem ganhando espaço por buscar a resolução de problemas através de princípios de *design*, entendendo as necessidades do usuário e validando propostas de solução através de protótipos, oferecendo uma aprendizagem iterativa [50], [68].

O DT, através da co-criação entre equipes multidisciplinares, e da inovação de ideais que sejam rentáveis e sustentáveis para o negócio [13], despertou o interesse da indústria de desenvolvimento de software e, conseqüentemente, da comunidade acadêmica de ES. Entretanto, apesar do interesse, ainda se sabe pouco sobre o uso dessa abordagem. Por exemplo, por quais razões equipes de software optam por usar o DT, em quais contextos é usado, quais modelos e técnicas são adotados, quais são os benefícios ou desafios no uso, entre outros.

1.1 Objetivo Geral e Objetivos Específicos

Dado o contexto apresentado, esta Dissertação de Mestrado teve como objetivo geral *caracterizar o uso de Design Thinking no desenvolvimento de software*. Essa caracterização envolve o entendimento de aspectos relacionados ao uso da abordagem, seus benefícios e dificuldades enfrentadas.

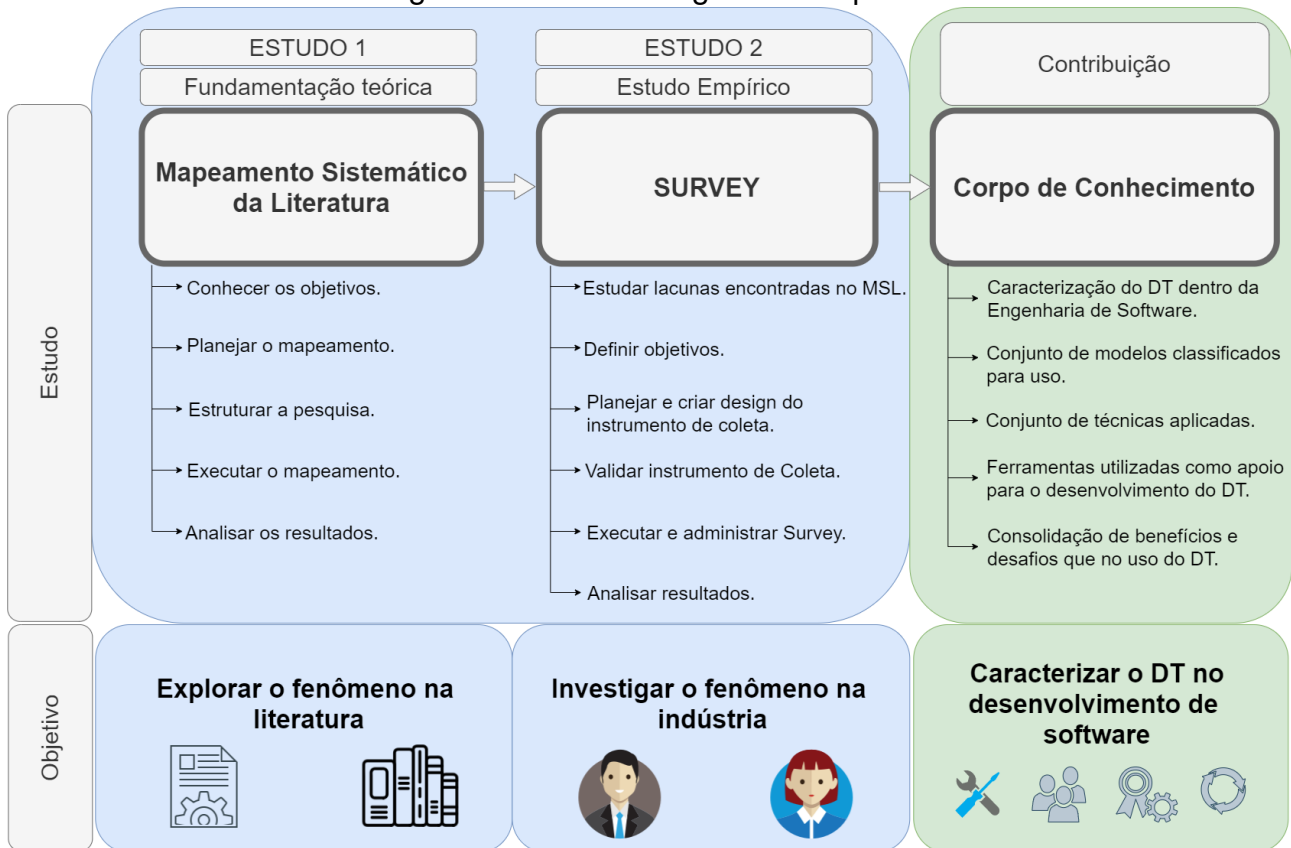
Para atender ao objetivo geral deste trabalho, definiu-se os seguintes objetivos específicos:

1. Investigar na literatura sobre o uso do *Design Thinking* no desenvolvimento de software, seus benefícios e dificuldades no uso;
2. Identificar na indústria de software como o *Design Thinking* é usado, seus benefícios e dificuldades percebidas;
3. Gerar um corpo de conhecimento organizado por aspectos que se mostraram relevantes a partir do estudo da literatura e da prática caracterizando o uso de *Design Thinking* no desenvolvimento de software.

1.2 Visão Geral da Metodologia de Pesquisa

Para atender ao objetivo geral e objetivos específicos deste trabalho, definiu-se dois estudos, conforme ilustrado na Figura 1.1, quais sejam: um Mapeamento Sistemático da Literatura (MSL) [90] e uma *Survey* [91]. O primeiro trouxe o entendimento a partir da literatura, atendendo ao Objetivo Específico 1; e o segundo, da prática, referente ao Objetivo Específico 2. A *Survey* foi projetada considerando os resultados do Estudo 1 como insumo. A caracterização do uso de *Design Thinking* no desenvolvimento de software, através da organização do corpo de conhecimento no tópico (Objetivo Específico 3), é então resultante da análise crítica dos dois estudos anteriores e representa a principal contribuição deste trabalho.

Figura 1.1 – Metodologia de Pesquisa



FONTE: Autor, 2019

1.3 Contribuição

A contribuição principal dessa Dissertação de Mestrado é a caracterização do *Design Thinking* no âmbito da Engenharia de Software. Essa caracterização está organizada da seguinte forma: conhecimento do contexto que o DT é utilizado no processo de desenvolvimento de software, os diversos modelos e técnicas que representam e constituem o uso do DT. Além disso, conhece-se ferramentas que são utilizadas como apoio para o desenvolvimento do DT e benefícios e desafios que o DT trás para a comunidade da Engenharia de Software. Dentre estes fatores que juntos caracterizam este fenômeno, esta pesquisa buscou caracterizar o DT por meio de uma pesquisa criteriosa na literatura e encontrar posicionamento de profissionais experientes para reforçar e defender o conhecimento obtido sobre o *Design Thinking* na Engenharia de Software.

Os resultados desta pesquisa são fundamentais e importantes para entender como usar o DT no desenvolvimento de software e como entendê-lo de tal forma que ajude a comunidade da Engenharia de Software na execução do DT e auxilie no atendimento do real do problema trabalhado. A caracterização do DT sobre o desenvolvimento de software será benéfica para a academia e indústria. A união destas esferas é benéfica para o aprimora-

mento do uso de técnicas e processos da qual possuem conteúdos ricos e diversificados que são explorados neste trabalho a prol da qualidade da construção de softwares que atendam e correspondam diretamente as peculiaridade e problemas trazidos por seus usuários.

Além disso também, os dados obtidos nesta dissertação contribuíram para literatura. Serviram como base para proposta de uma ferramenta de recomendação do DT para requisitos de software. Essa contribuição é apresentada através das seguintes publicações:

- *Parizi, R.; Moreira da Silva, M.; de Souza Couto, I.; Pavin Trindade, K.; Plautz, M.; Marczak, S.; Conte, T.; Candello, H. "Design thinking in software requirements: What techniques to use? a proposal for a recommendation tool". In: Proceedings of the Ibero-American Conference-American on Software-American Engineering, 2020, pp. 14 [89].*
- *Prestes, M.; Parizi, R.; Marczak, S.; Conte, T. "On the use of design thinking: A survey of the brazilian agile software development community". In: Proceedings of the International Conference on Agile Software Development, 2020, pp. 13 [95].*

1.4 Estrutura do Trabalho

Essa dissertação está organizada em 6 capítulos, como segue: o **Capítulo 1** apresenta uma introdução sobre o trabalho, a motivação, os objetivos, uma visão geral sobre a metodologia aplicada nesse trabalho e a contribuição; o **Capítulo 2** apresenta o embasamento teórico dos conceitos sobre DT e DT no desenvolvimento de software; o **Capítulo 3** mostra todo o trabalho, os resultados e discussão sobre os dados obtidos do Estudo 1, na qual realizou-se um Mapeamento Sistemático na Literatura; o **Capítulo 4** apresenta o trabalho realizado no Estudo 2, a Survey, junto com os resultados e discussão sobre os dados coletados; o **Capítulo 5** apresenta a discussão sobre os resultados expostos no Capítulo 3 e Capítulo 4, considerando todo o embasamento obtido no decorrer desse trabalho; e, por final, o **Capítulo 6** que apresenta as considerações finais, as limitações encontradas ao longo desta dissertação, conclusões e trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo tem como objetivo buscar um embasamento teórico sobre o DT em sua origem e o DT no desenvolvimento de sistema, para melhor entendê-lo servir como orientação para a realização dos estudos deste trabalho.

2.1 Design Thinking

A forma que um *designer* enxerga a solução de um problema tornou-se atrativa pela associação à qualidade ou na aparência estética de produtos, visando a promoção do bem-estar da vida das pessoas, abrindo desta forma caminhos para a inovação empresarial. Ao enxergar um problema e como esse pode prejudicar ou impedir a experiência das pessoas, faz com que a tarefa principal do *designer* seja a identificação de problemas e a geração de soluções. Desta forma, através do mapeamento da cultura inserida no problema, o contexto apresentado e diversos outros fatores envolvidos, faz com que o profissional identifique as causas e as consequências das dificuldades inseridas neste meio. Assim, o DT traz na forma que o *designer* trabalha para a compreensão do fenômeno, através de um pensamento abduutivo, um raciocínio que busca a criatividade e inovação [116]. Desta forma, pode-se dizer que o DT é uma abordagem para resolução de problemas inovadores, tendo duas principais perspectivas, sendo elas: a solução de problemas e a inovação.

De acordo com Tschimmel [113], o DT depende da capacidade do *designer* de considerar as necessidades humanas e novas visões de viver bem, o material disponível e os recursos técnicos, e as restrições e oportunidade de um projeto ou negócio.

A integração desses três fatores, conforme Tschimmel [113], é a exigência do *designer* de ter a capacidade de ser ao mesmo tempo analítica e enfática, racional e emocional, metódica e intuitiva, e orientada por planos e restrições. Desta forma, novamente, destaca-se esse raciocínio dualista que os *designers* pensam utilizando o termo do "pensamento abduutivo". Pensamento desenvolvimento por Charles Sander Pierce, onde defendia que nenhuma ideia nova poderia ser produzida por dedução ou indução usando dados do passado, definindo-se então este pensamento como uma forma de pensar em perspectivas novas e diferentes e em possibilidade futuras, que não se encaixam nos modelos existentes, finalizando como uma forma de pensar em que os sentimentos e emoções são tão importantes quanto a racionalidade [113].

Dada a natureza de busca de inovação do DT, características típicas dessa abordagem são o envolvimento de equipes multidisciplinares e o uso extensivo de ferramentas, como notas adesivas, quadros brancos e todos os tipos de materiais de prototipagem [46].

Assim, Steinbeck [111] ressalta que o DT se concentra no processo de *design* e não em um produto, ou seja, algo que pode emergir dele.

Além disto, Weigel [118] menciona DT como uma das formas mais eficazes no descobrimento de informações detalhadas, visando a empatia dos usuários e entendimento do contexto de uso para desenvolver uma maior compreensão da necessidade dos usuários. Brown [12] destaca três pontos importantes da abordagem, em resumo às definições anteriores: (i) é uma abordagem de *human-centred design* (HCD), (ii) visa a criação de soluções de projetos viáveis que atendam às necessidades dos clientes com valor agregado e (iii) busca projetar artefatos através de princípios de *design*.

Independente das diversas definições de DT, todas elas gravitam na mesma filosofia que abrange o pensamento criativo, o conhecimento contextual e de cenários na aplicação em que uma solução seria aplicada [75]. Tschimmel [113] indica que quando DT é aplicada nas organizações, funciona como um catalisador para o processo de inovação, oferecendo a possibilidade de inovar mais e melhor. Diante disto, o uso de DT em organizações pode beneficiar a forma com que estas atuam no mercado, pois assume como a primeira preocupação a satisfação dos clientes, compreendendo as necessidades, desejos e motivações, para posteriormente avançar com as propostas e produtos/serviços [7].

Conforme Brenner, Uebernickel e Abrell [11], o processo de *design*, seguindo a metodologia clássica de *design*, foi dividido em várias etapas para facilitar o planejamento das tarefas do projeto, atividades coletivas e de produção e cronogramas. As primeiras referências a uma estrutura multifásica do processo criativo em geral remontam a Poincaré (1924), que através de suas reflexões sobre seu próprio processo de pensamento criativo na solução de problemas matemáticos, deu impulso a Wallas (1926) para dividir o processo criativo em quatro fases: a fase de preparação, a incubação, a iluminação e a fase de verificação. Essa classificação foi o ponto de partida dos movimentos de pesquisa do *design*, que buscavam novos modelos para melhor descrever as fases de um processo criativo de resolução de problemas.

A literatura oferece uma infinidade de modelos de processos de DT para abstrair e representar os espaços de trabalho existentes do mesmo, como exemplo o modelo 3I (inspiração, idealização e implementação), criado pela IDEO em 2001 no contexto da inovação social, conhecido como modelo do Brown, baseando-se de conceitos de HCD¹, onde o usuário se torna um participante do processo. Este modelo utiliza o próprio conceito de HCD e também possui *kits*² de apoio que auxiliam na aplicação desta filosofia, trazendo o intuito de ajudar a entender as necessidades dos usuários através de novas formas. Seguindo o exemplo, o modelo *Hasso-Plattner Institute* (HPI), similar ao modelo de Brown, que foi desenvolvido em um contexto educacional, criado pela escola HPI, ligada diretamente à Universidade de *Stanford* e IDEO, onde o modelo é visualizado por partes conectadas por

¹<https://designthinking.ideo.com/>. Acessado em: Fev/2020

²<https://www.ideo.com/post/design-kit>. Acessado em: Fev/2020

linhas e executadas em *loops* iterativos [113]. Tais modelos de processos são seguidos por um conjunto de ferramentas e métodos distintos que formam um *kit*. O DT definido por Brenner, Ueberschick e Abrell [11] é apresentado por três frentes. São elas: mentalidade, onde o DT é caracterizado por vários princípios fundamentais, como a combinação de um pensamento divergente e convergente, orientados para as necessidades dos clientes e usuários. O processo, onde é definido por uma combinação de micro e macro processo. Sendo o micro como o processo de inovação em si, representados pela definição do problema, encontrar as necessidades e sínteses, idealizar, prototipar e testar, e o macro processo é dividido dentro de espaços da convergência e divergência. Por fim a caixa de ferramentas, onde o DT refere-se à aplicação de diversos métodos e técnicas de diferentes disciplinas (design, engenharia, informática e psicologia).

A literatura também mostra que o DT funciona apenas quando as técnicas utilizadas estão alinhadas com essa nova maneira de pensar [11]. Portanto, escolher quais técnicas usar pode ser um desafio para os profissionais iniciantes em TI no uso da DT, visto que como qualquer técnica ou ferramenta, quem as utiliza deve possuir o conhecimento, a experiência e a competência certa para aplicação. Embora alguns especialistas, como os da IDEO, sugiram *kits* de ferramentas organizados por espaços de trabalho ou por perguntas para conduzir o trabalho de *design* (por exemplo, que ferramentas eu posso usar para entender as pessoas?), ainda há uma infinidade de técnicas para escolher. Por exemplo, Brenner et al. [11] relatam que mais de 45 técnicas foram usadas no *Institute of Information Management at the University of St. Gallen*.

2.2 Design Thinking no Desenvolvimento de Software

Sobre a perspectiva do uso do DT no desenvolvimento de software, Soledade et al. [106] destacam que DT é um conjunto de técnicas e ferramentas centrado no usuário que suporta um processo iterativo para produzir e analisar de forma criativa soluções para os desafios reais. A aplicação destas técnicas apoiam as tarefas de refinamento de requisitos dentro do contexto considerado apropriado. Por outro lado, para Vetterli et al. [115] o DT fornece uma metodologia para conhecer as necessidades do cliente, e não os requisitos, a fim de produzir uma série de protótipos rápidos e simples que convergem em soluções inovadoras. Também o DT incorpora características do desenvolvimento ágil sobre aplicações móveis como o envolvimento dos usuários e que ajuda na identificação de problemas mal definidos.

O DT traduz observações em *insights* e, sobre estes, em serviços ou produtos buscando melhorar a vida das pessoas. Dessa forma, o DT foca em atender as necessidades dos usuários, além de também ser apresentado através de fases que assemelham-se à etapas do ciclo de desenvolvimento de software [13]. Fortalecendo a ideia do foco na

necessidade dos usuários, Seyff et al. [104] abordam que o envolvimento do usuário final sobre a área da Engenharia de Software é focado e importante para o desenvolvimento de sistemas utilizáveis e úteis para o usuário. Com isso o DT mostra ser capaz de prover a solução sobre a real necessidade do usuário, através de uma característica multidisciplinar, colaborativa e convergida sobre diversos pensamentos e processos que resultam em soluções para o real problema [116].

Para que se pudesse entender mais sobre o uso de *Design Thinking* no desenvolvimento de software, incluindo os seus benefícios e desafios e suas diferentes formas de uso, iniciou-se estudando a literatura (Objetivo Específico 1, Estudo 1).

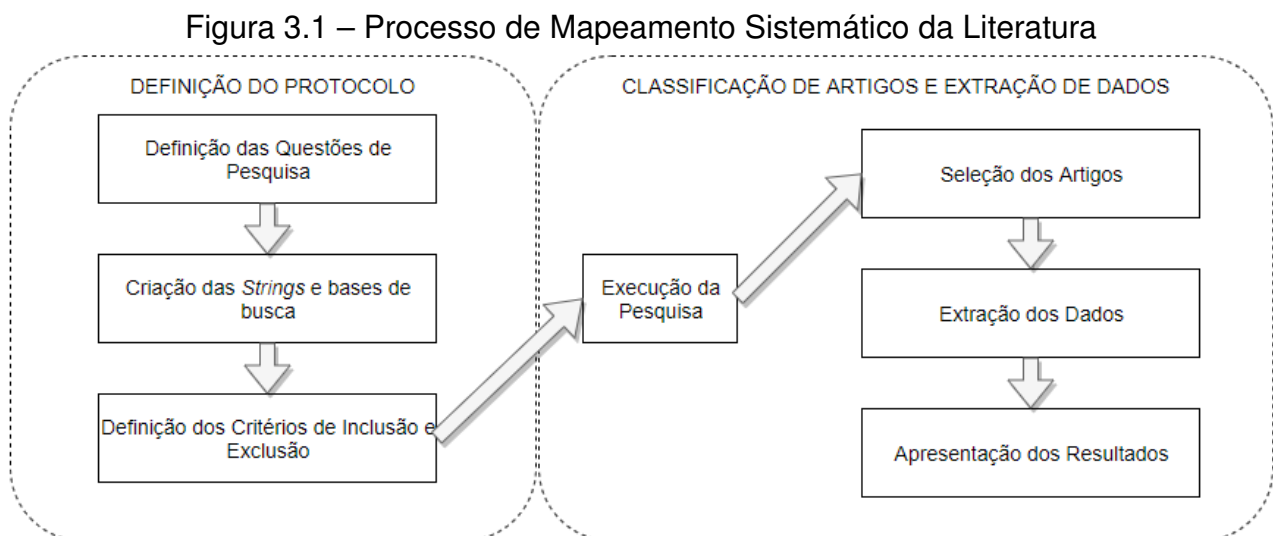
3. MAPEAMENTO SISTEMÁTICO DA LITERATURA

Com intuito de investigar na literatura sobre o uso do *Design Thinking* (DT) no desenvolvimento de software, seus benefícios e dificuldades no uso, realizou-se um Mapeamento Sistemático da Literatura conforme descrito nesse capítulo.

3.1 Mapeamento Sistemático da Literatura Segundo Petersen e Colegas

Para atender o Objetivo Específico 1 dessa Dissertação de Mestrado, decidiu-se pelo estudo da literatura de maneira sistemática, seguindo o protocolo definido por Petersen e colegas [90]. Um Mapeamento Sistemático fornece um mapa (ou resumo) dos resultados obtidos através de um processo definido por etapas essenciais e aplicados de forma sistemática. Cada etapa do processo possui um resultado, sendo considerado o resultado final um mapeamento sistemático [90]. Este processo tem como objetivo a exploração na literatura sobre o uso do DT no contexto da ES visando a investigação e o conhecimento deste fenômeno para aplicá-lo da melhor forma na área.

Segundo os autores [90], um conjunto de passos devem ser seguidos, conforme ilustrado na Figura 3.1.



FONTE: Petersen e colegas [90]

Conforme apresenta Figura 3.1, a seguir define-se cada fase que conduz este processo de mapeamento, de acordo com Petersen e colegas [90].

- 1. Definição das Questões de Pesquisa:** Definição dos objetivos desta pesquisa para verificação da necessidade de realizar este mapeamento. Contempla-se desta forma

o entendimento dos objetivos e formam questões de pesquisa que representem a busca por evidências na literatura para serem respondidas;

2. **Criação de *Strings* e bases de busca:** Conhecendo o objetivo da pesquisa e denominada as questões de pesquisa, cria-se um plano para conduzir a execução da pesquisa, identificando a *string* de busca e o local, por exemplo, conferências ou bibliotecas para seleção dos estudos primários;
3. **Definição dos Critérios de Inclusão e Exclusão:** Após execução prévia e obtido os estudos primários nas bases de dados, aplica-se uma triagem dos estudos através de critérios de inclusão e exclusão definidos no protocolo por meio de uma leitura sobre o título do estudo, suas palavras-chave e resumo;
4. **Agrupamento por categorias:** Durante o processo de leitura na etapa anterior, previamente definam-se classificações sobre os estudos aceitos, para facilitar e identificar o tópico que cada estudo representa;
5. **Extração dos dados:** Coleta-se ao final um conjunto de dados a partir de uma leitura completa sobre os artigos aceitos na etapa de seleção, extraindo as evidências e formando argumentações que respondem as questões de pesquisa criadas para representar o objetivo do mapeamento. Geralmente criam-se planilhas sumarizadas com os dados obtidos e gráficos que facilitam e organizam as ideias encontradas nos trabalhos.

Como mencionado, o Mapeamento da Literatura realizado nessa dissertação (Estudo 1) seguiu os passos definidos por Petersen e colegas [90]. Assim, realizou-se os mesmos passos apresentados na Figura 3.1, detalhados nas seções a seguir.

3.2 Planejamento e Realização do Mapeamento

Questões de Pesquisa. Para atingir-se o objetivo deste estudo, as seguintes questões de pesquisa (QP) foram definidas para guiar esse Mapeamento Sistemático:

- QP1: Qual o propósito de se usar DT na engenharia de software?
- QP2: Em qual contexto de desenvolvimento de software o DT está sendo usado? Como por exemplo, em equipes distribuídas, com cliente presente, associado a algum método ágil.
- QP3: Quais modelos, métodos, técnicas e/ou abordagens utilizadas do DT na Engenharia de Software?

- QP4: Quais ferramentas são utilizadas para apoiar a realização de DT no desenvolvimento de software?
- QP5: Como estes modelos, métodos, técnicas, abordagens e ferramentas são selecionados para o uso?
- QP6: Quais são as lições aprendidas, boas práticas, recomendações e dificuldades experimentadas no DT no desenvolvimento de software?

Condução das Buscas. Conforme apresentado na estrutura do mapeamento na Seção 3.1, a condução da busca na literatura é representada pela criação das *strings* de busca e as bases selecionadas para seleção dos artigos. A busca foi realizada com base nos principais termos extraídos das questões de pesquisas estruturadas usando os critérios de População, Intervenção, Comparação e Resultados (PICO) sugeridos por Kitchenham e Charters [60], praticados em revisões de literatura [38], [90]. Esses critérios concentram-se na identificação de palavras-chave em busca de formular *strings* de pesquisa com base em questões de pesquisa. Neste estudo estabeleceu-se os seguintes critérios do PICO:

- População: projetos de desenvolvimento de software que usam DT durante o processo de desenvolvimento;
- Intervenção: Petersen [90] define este tópico, no contexto de engenharia de software, como metodologia de software, tecnologia, ferramenta ou procedimento. Definiu-se neste estudo intervenção de como DT está em desenvolvimento de software;
- Comparação: Neste estudo, não identificou-se um sistema de comparação;
- Resultados: Entender como o DT está sendo usado no desenvolvimento de software e qual conhecimento está sendo gerado a partir deste uso no desenvolvimento de software.

Com base nas definições do PICO, criou-se as palavras-chave e sinônimo mencionados na Tabela 3.1 para ajudar a construir as sequências de pesquisa e aplicar em fontes de dados de bibliotecas digitais na web, com foco na recuperação de um conjunto adequado de documentos relevantes para esta investigação. Foram utilizadas as seguintes bibliotecas digitais como fonte: Scopus¹, IEEEExplore², Springer Database³, ACM Digital Library⁴, Science Direct⁵ e Wiley Online Library⁶. Estas fontes foram escolhidas por agregarem as informações da área de engenharia de software e seguindo as recomendações de

¹<https://www.scopus.com>

²<https://ieeexplore.ieee.org>

³<https://link.springer.com/>

⁴<https://dl.acm.org/>

⁵<https://www.sciencedirect.com/>

⁶<https://onlinelibrary.wiley.com/>

Tabela 3.1 – *String* de Busca

Critério PICO	Palavras-Chave e Sinônimos
População	<i>software engineering</i> OR <i>software development</i> OR <i>software industry</i> OR <i>software construction</i> OR <i>software project</i> OR <i>software process</i> OR <i>Software project management</i> OR Engenharia de software OR Desenvolvimento de Software OR Gerenciamento de projeto
Intervenção	<i>design think*</i> OR <i>design session</i>

FONTE: Autor, 2019

Kitchenham e colegas [60]. Como ferramenta de apoio para extração dos dados e triagem, utilizou-se o Start [37], onde auxilia no processo de mapeamento sistemático. Portanto, neste trabalho, as fontes de dados possuem os seguintes critérios:

- Fontes de Dados que incluem artigos de periódicos, conferências e trabalhos de *workshops* focados em Engenharia de Software e tomada de decisão;
- Fontes de dados com um mecanismo de pesquisa avançados que permite a filtragem dos resultados por palavras-chave;
- Fontes de dados que fornecem acesso a artigos completos escritos em inglês e português.

Em sequência, apresenta-se um exemplo de *string* de pesquisa:

*("engenharia de software") OR ("software engineering") OR ("desenvolvimento de software") OR ("software development") OR ("indústria de software") OR ("software industry") OR ("construção de software") OR ("software construction") OR ("projeto de software") OR ("software project") OR ("processo de software") OR ("software process") OR ("gestão de projeto de software") OR ("software project management") OR ("Gerenciamento de projeto ") ("design think *") OR ("sessão de design"))*

Triagem dos estudos. Após estruturação das *strings* de busca e execução nas diversas bases escolhidas, realizou-se um procedimento de inclusão e exclusão. A Tabela 3.2 mostra os critérios criados para inclusão e exclusão dos artigos retornados nas fontes de dados:

- Realização da leitura do título, resumo e palavras-chave de todas publicações retornadas;
- Publicações que não atendiam os critérios de inclusão foram excluídas. Caso houvesse dúvida sobre sua publicação, a mesma era colocado para o próximo passo;
- Leitura completa das publicações selecionadas, considerando as mesmas escolhidas como relevantes para o estudo;

Tabela 3.2 – Critérios de Inclusão e Exclusão

Critérios de Inclusão	
CI1	Artigos que relatam o <i>design thinking</i> no desenvolvimento de software.
CI2	Estudos Primários.
Critérios de Exclusão	
CE1	Artigos que não atendam o CI1 e CI2 e não abordam o uso <i>design thinking</i> no desenvolvimento de software.
CE2	Artigos duplicados.
CE3	Artigos não disponibilizados para <i>download</i> .
CE4	Artigos que não estejam escritos em inglês ou português.
CE5	Artigos que não foram revisados por pares (<i>peer review</i>).

FONTE: Autor, 2018

- Ao final da extração e leitura dos artigos, obtém-se um conjunto de publicações potencialmente relevantes.

Após a realização do procedimento de inclusão e exclusão e com intuito de aumentar a confiabilidade da classificação das publicações, os artigos selecionados foram classificados por dois pesquisadores individualmente (pesquisador 1 e pesquisador 2), verificando desta forma o nível de concordância entre eles. Para isso utilizou-se o Kappa [40], considerado uma medida de concordância interobservador que mede o grau de concordância. Desta forma, os artigos que obtiveram a melhor avaliação, ou seja, aceito por ambos pesquisadores, foram inseridos como válidos para este mapeamento. Caso contrário, os artigos que resultaram em uma discrepância entre as avaliações, foram discutidos com um terceiro pesquisador para decisão final [69]. O pesquisador 1 é representado pelo autor desta Dissertação, na qual trabalhou no processo do Kappa [40] junto com aluno de doutorado que participa do mesmo grupo de pesquisa, aqui citado como pesquisador 2. A terceira pesquisadora aqui apresentada refere-se a orientadora desta dissertação.

Execução. Desta-se no processo de execução cada etapa e procedimento realizado, conforme mostra a Figura 3.1. Com base em sequências de pesquisa definidas e bancos de dados *online* selecionados, realizou-se buscas para capturar os estudos relacionados para esta pesquisa.

A Tabela 3.3 descreve o número de artigos recuperados de pesquisas executadas nas fontes de dados. A primeira coluna mostra a fonte dos dados e a segunda apresenta o número de estudos trazidos pela pesquisa, resultando um conjunto inicial de 2588 estudos.

Com base nos resultados da execução das buscas, prosseguiu-se para a etapa de seleção dos estudos, tendo em conta os critérios estabelecidos.

Tabela 3.3 – Resultados da Pesquisas nas Fontes de Dados

Base de Dados	Resultado
ACM	68
IEEE Xplore	513
Science Direct	232
Scopus	1131
Springer	484
Wiley	160
Total	2588

FONTE: Autor, resultados obtidos em dezembro/2018

Inicialmente o processo foi executado pelo primeiro pesquisador, e pela execução das sequências de busca, coletando os artigos recuperados das fontes de dados. Em seguida, outras três etapas foram realizadas com a participação de outros três pesquisadores para qualificar o processo de seleção. Os três passos executados para o processo de seleção de estudos foram:

[ETAPA 1] Aplicação dos critérios de exclusão CE1, CE2, CE3, CE4 e CE5.

A primeira etapa do processo de seleção levou em consideração o conjunto de artigos obtidos com as strings de pesquisa. Então o pesquisador 1 trabalhou na remoção de documentos duplicados usando a ferramenta *StArt* [37] - uma ferramenta desenvolvida para fornecer apoio ao processo de Pesquisa Sistemática. Além disso, o mesmo trabalhou para a remoção de artigos não revisados por pares, eliminando os artigos não disponíveis para *download*, e remoção de estudos que não foram escritos em inglês ou português, de acordo com os critérios apresentados na Tabela 3.2. Esta primeira etapa resultou na exclusão de 6% do total de artigos (6% de 2488 = 155 artigos), permanecendo 2433 estudos.

[ETAPA 2] Análise de título, resumo e palavras-chave (primeira rodada)

Nesta etapa, o primeiro e pesquisador 2 realizaram a primeira rodada de seleção de estudos lendo o título, palavras-chave, e resumo. No total, foram lidos todos os 2433 artigos, baseados nas questões de pesquisa definidas e objetivos deste trabalho, aplicando critérios CI1, CI2 ou CE1. Nesta fase, 1650 artigos não satisfizeram os critérios de inclusão e exclusão estabelecidos, sendo desta forma excluídos desta pesquisa. Assim, restaram 783 artigos, equivalentes a 32,18% do total inicial de 2588 artigos.

Tabela 3.4 – Classificação KAPPA

KAPPA	Concordância
< 0	Sem concordância
0.01 - 0.20	Baixa
0.21 - 0.40	Razoável
0.41 - 0.60	Moderado
0.61 - 0.80	Substancial
0.81 - 0.99	Perfeito

FONTE: Garrett e Viera [117]

[ETAPA 3] Análise de título, resumo e palavras-chave (segunda rodada)

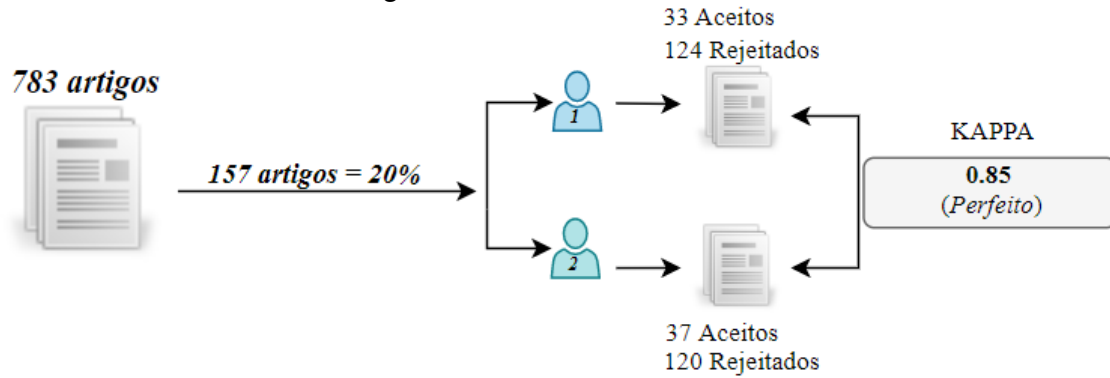
Na terceira etapa do processo de seleção dos estudos, foi realizada uma avaliação de concordância usando o coeficiente Kappa de Cohen [22]. Segundo Unnikrishnan [89] [114], esse coeficiente é uma medida tradicional para estimar o grau de similaridade (ou concordância) entre dois avaliadores. Nesta etapa, já descrita anteriormente, garantiu a acurácia dos trabalhos selecionados. Assim, uma etapa de refinamento e garantia de qualidade foi realizada nos trabalhos selecionados anteriormente. Os 3 pesquisadores examinaram, cada um, 157 artigos para calcular o coeficiente de KAPPA, resultando em uma pontuação de 85,31% de concordância, considerada substancial de acordo com os índices de classificação de KAPPA [117], tal como apresentado na Tabela 3.4. Estes artigos foram selecionados de forma aleatória para execução do KAPPA e foram selecionados entre as bases de dados, ou seja, não foi realizado uma aleatoriedade em apenas uma base, mas sim sobre todas para que nenhuma ficasse de fora da participação do processo do KAPPA.

A Figura 3.2 mostra o processo realizado para avaliar a precisão do processo de seleção usando o KAPPA. Como é possível ver, os 3 pesquisadores analisaram o título, resumo e palavras-chave do mesmo subconjunto de artigos, compreendendo 157 artigos. O terceiro passo levou à inclusão de 115 artigos (C11 e C12), representando 14,68% dos 783 estudos que resultaram da segunda etapa. Consequentemente, os outros 668 artigos foram excluídos desta etapa usando os critérios CE1.

Estas três etapas representam o fluxo de processo trabalhado sobre a etapa de execução e seleção para atingir a meta, com a seleção destes 115 artigos como aceitos, iniciou-se um processo de extração e leitura dos dados a fim de encontrar nas suas essências, as evidências necessárias para responder as questões de pesquisa e objetivo deste mapeamento. A Tabela 3.5 mostra o processo resumido de cada etapa descrito acima e seus resultados.

Ao final do processo iterativo de seleção de estudos, teve-se com 115 artigos selecionados para leitura completa, visando a extração dos dados para responder as questões

Figura 3.2 – Processo KAPPA



FONTE: Autor, 2019

Tabela 3.5 – Execução do Processo de Mapeamento

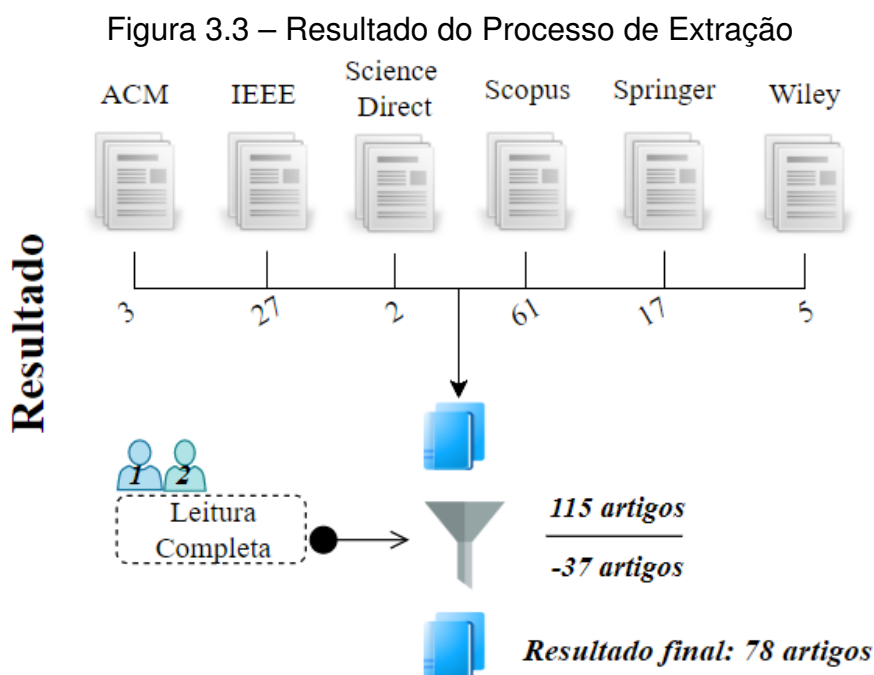
Etapa	Bases	Qtd. de artigos avaliados	Qtd. de artigos removidos	Triagem	Resultado	Kappa			%
						Pesquisador	Aceitos	Rejeitados	
1	ACM	68	35	CE2 e CE5	115 excluídos Resultado 2433 Artigos		N/A		
	IEEE	513	10						
	SCIENCE DIRECT	232	41						
	SCOPUS	1131	53						
	SPRINGER	484	16						
	WILEY	160	0						
2	ACM	33	17	CE1 e CE3	1650 excluídos Resultado 783 Artigos		N/A		
	IEEE	503	285						
	SCIENCE DIRECT	191	146						
	SCOPUS	1078	701						
	SPRINGER	468	390						
	WILEY	160	111						
3	ACM	16	13	CE1, CE3 e CE4	668 excluídos Resultado 115 Artigos	Pesquisador 1	33	124	0.85 (Perfeito)
	IEEE	218	191						
	SCIENCE DIRECT	45	43			Pesquisador 2	37	120	
	SCOPUS	377	316						
	SPRINGER	78	61						
	WILEY	49	44						

FONTE: Autor, 2019

de pesquisa. Esse número de artigos representa 4,44% da quantidade inicial de estudos da pesquisa realizada na primeira etapa.

Foi realizada a extração de dados através da leitura completa dos 115 artigos. A extração dos dados, para responder às seis questões de pesquisa estabelecidas neste estudo, foi organizada em uma planilha eletrônica, composta pelas seguintes colunas: identificador de estudo; base de busca; título; um resumo para responder aos questões; autores; ano de publicação e; revista ou conferência. A Figura 3.3 apresenta que, após o processo de leitura completa, foram removidos 37 artigos, visto que estes não foram compreendidos como DT na Engenharia de Software.

Limitações. A limitação encontrada no Estudo 1 foi o fato de não saber se as fontes que foram utilizadas cobrem todos os artigos existentes. A ação realizada de mitigação foi pesquisar sistematicamente em todas as fontes recomendadas na computação. Porém, artigos que focam em DT tendo a ES de exemplo podem aparecer em eventos específicos do de-



FONTE: Autor, 2019

sign de produtos, que possivelmente não fazem parte das fontes de dados da computação. Também, por ter sido considerado um trabalho que ocupou todo o primeiro semestre do ano de 2019 e os artigos coletados foram selecionados entre janeiro e fevereiro de 2019, acredita-se que novos artigos possam ter surgido durante o processo de seleção e extração destes, podendo dessa forma não ter absorvido resultados sobre pesquisas mais recentes na área. Uma atualização destes dados sobre os anos de 2019 e 2020 podem vir à ajudar na descoberta de novas evidências.

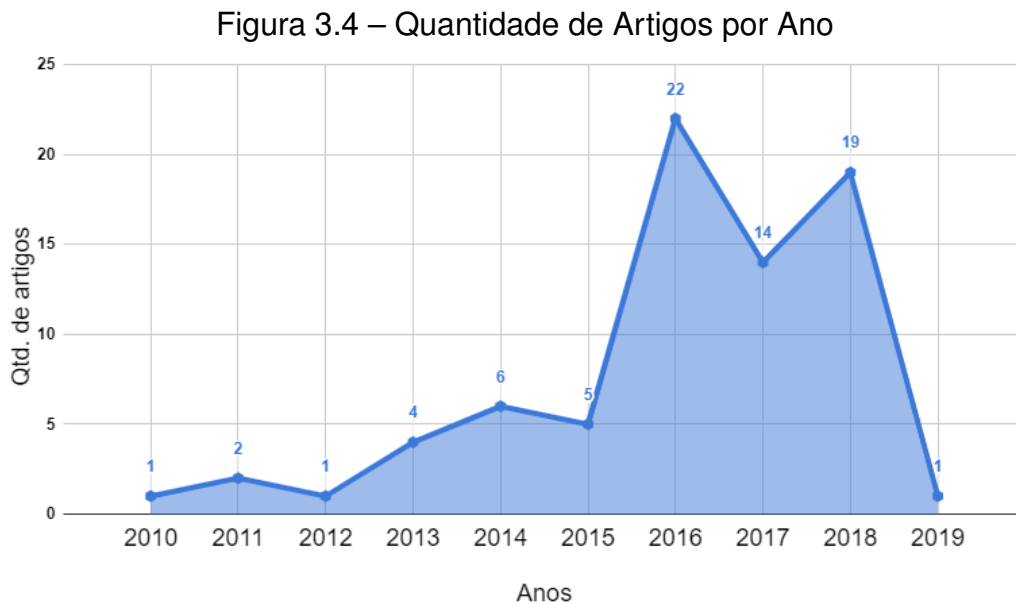
A seguir apresentam-se os resultados sobre um conjunto de 78 artigos resultantes deste mapeamento para responder as questões de pesquisa.

3.3 Resultado do Mapeamento Sistemático

Antes de iniciar a discussão sobre os resultados obtidos para as questões de pesquisa, realizou-se 3 levantamentos com viés quantitativo para ilustrar e caracterizar a grandeza dos dados obtidos neste mapeamento sistemático.

Meta-informação dos estudos selecionados. Os estudos selecionados neste trabalho foram publicados de 2010 a 2018, obtendo neste contexto a introdução deste tema na literatura a partir do ano de 2010. Porém, pelo fato de ter realizado a triagem em dezembro/2018, foi retornado apenas 1 artigo em 2019. Além disso pode-se apontar que o maior número de publicações ocorreram em 2016, com um total de 22 publicações, conforme mostra a

Figura 3.4. Com isso entende-se que houve um crescimento do interesse sobre a abordagem de trabalhar com o DT a partir de 2015. Entretanto este dado mostra apenas em um olhar quantitativo do quanto o DT vem sendo praticado em empresas de desenvolvimento de software nos últimos anos.



FONTE: Autor, 2019

Sobre este conjunto de artigos publicados neste período, destacam-se na Tabela 3.6 os pesquisados com maior número de artigos publicados. Além destes, surgiram 181 autores neste mapeamento sistemático.

Tabela 3.6 – Pesquisadores com Maior Número de Estudos Publicados

Pesquisadores	Citações	Observação
Sisira Adikari	[1][2][48][58][103]	Total de 5 citações
Tayana Conte	[24][39][108]	Total de 3 citações, sendo 2 em conjunto com Bruna Ferreira
Bruna Ferreira	[76][39][108]	Total de 3 citações, sendo 2 em conjunto com Tayana Conte
Heath Keighran	[1][58][103]	Total de 3 citações, sendo 2 em conjunto com Sisira Adikari
Meira Levy	[71][72][73]	Total de 3 citações

FONTE: Autor, 2019

No total, sobre os 78 artigos finais, foram encontradas 53 conferências. A Tabela 3.7 mostra as três conferências que mais se ofereceram conteúdo sobre o uso do DT na ES. Considera-se então este tema de interesse para áreas como Design, *User Experience* e Usabilidade (HCI, DS) e relacionados à Engenharia de Software, como a *International Conference on Software Engineering (ICSE)*.

Tabela 3.7 – Conferências x Publicações

Conferências	Qtd.	Referências
HCI - International conference of Design, User experience, and Usability	9	[1][2][19][29] [48][58][82] [101][103]
Proceedings of NordDesign	3	[32][47][80]
ICSE - International Conference on Software Engineering	2	[24][87]

FONTE: Autor, 2019

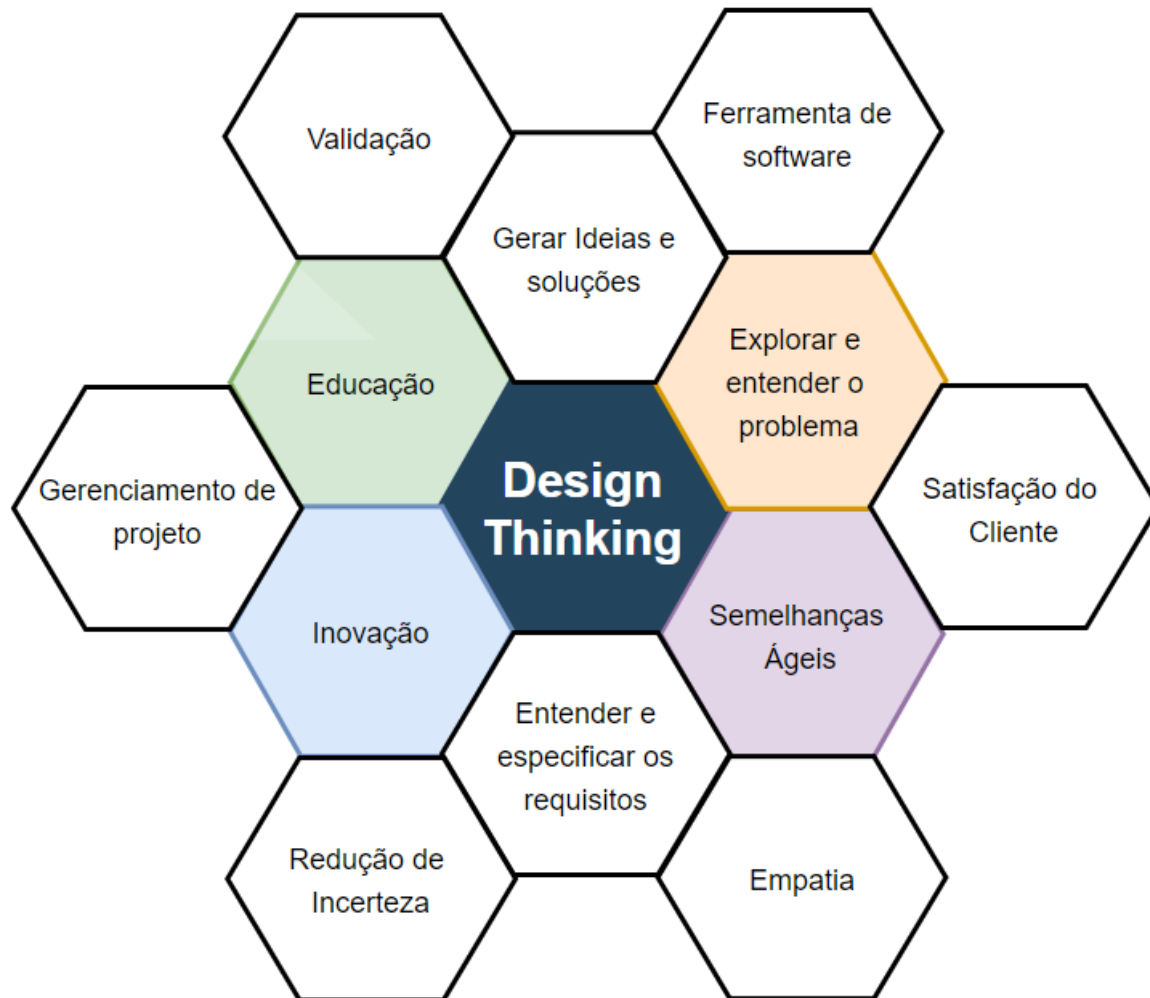
Análise dos resultados. Após abordar e caracterizar o conjunto de dados aceitos para responder as questões de pesquisa deste mapeamento sistemático, elaborou-se uma análise qualitativa sobre o conjunto de informações obtidas com objetivo de entender como é caracterizado o DT no desenvolvimento de software. Para resposta da QP principal, destaca-se a seguir o estudo consolidado em respostas sobre as questões específicas.

3.3.1 QP1: Propósito de uso do DT na Engenharia de Software

Grande parte dos artigos apresentaram o DT como uma abordagem para elucidar os problemas e facilitar no levantamento de requisitos no início do processo de desenvolvimento de software. Através das 12 classificações criadas nesse mapeamento, entendeu-se como a literatura está tratando deste assunto e entendendo o propósito de uso do DT para Engenharia de Software em 4 frentes: DT na educação de cursos de Engenharia de Software, DT inserido no desenvolvimento devido a suas semelhanças com os métodos ágeis, DT com propósito para o entendimento real do problema e DT utilizado para inovação, conforme mostra a Figura 3.5. Sobre estas frentes, existem outras classificações que são ressaltadas durante a explicação de cada uma das 4 frentes a seguir. A classificação completa dos artigos encontra-se destacada no Apêndice A.

A primeira frente é a educação na área da Engenharia de Software. O DT está sendo utilizado no ensino de programação orientada a objetos por fomentar a criatividade das ideias e geração de soluções sobre os problemas [97]. Também, inserido durante o curso da Engenharia de Software no seu ciclo de desenvolvimento para a realização da prática de levantamento de requisitos e por ajudar na empatia entre os estudantes [72]. Sabe-se que esse estudo procura principalmente o uso do DT na prática da indústria em si, mas cabe ressaltar que de início de todo processo a educação nos cursos da área podem se tornar grandes influenciadores no processo inserido na indústria por meio da ligação com as pessoas envolvidas. Entretanto, este tópico com abordagem a educação não atende sozinha como resposta para esta questão, com isso destacam-se como principais os demais propósitos de uso encontrados. O segundo tópico é a forte semelhança do DT com os métodos ágeis, fazendo com que o DT aproxima-se durante o processo de desenvolvimento apoiando os métodos ágeis.

Figura 3.5 – Propósitos de Uso do DT



FONTE: Autor, 2020

Inicialmente pode-se encontrar uma possível integração do DT com o XP. O XP traz uma forma rápida de desenvolvimento e o DT enfatiza o *feedback* dos clientes em relação aos protótipos, garantindo critérios de usabilidade, onde juntos conseguem garantir entregas rápidas e de acordo com uma pré-análise dos usuários [105]. Ressalta-se como semelhança e servindo como propósito de uso do DT no desenvolvimento de software, o relacionamento com *agile* por ambos fornecerem envolvimento do usuário, prototipação rápida e testes. Entretanto, por mais semelhantes possam ser, suas abordagens atendem momentos diferentes do desenvolvimento. O DT vem com foco na descoberta da solução para o problema inserido, definindo desta forma requisitos de acordo com a real necessidade do usuário. Com este embasamento que o DT fornece, os métodos ágeis atuam na codificação para estes requisitos, ou seja, o DT sozinho não consegue gerar um serviço ou software sem o apoio dos métodos ágeis, mas também os métodos ágeis possuem uma carência em codificar sem o levantamento inicial da solução sobre a real necessidade e na aproximação com usuário [55]. Os métodos ágeis dependem muito da entrada do cliente

no local, substituindo a documentação de requisitos pré-definida. Isso coloca desafios relacionados ao cliente como o conhecimento do mesmo sobre a exigência de um projeto complexo. A abordagem *agile* reduz a pesquisa inicial e a análise necessária comparado pelas metodologias tradicionais, gerando desta forma entregas mais rápidas. Porém, em alguns casos, as entregas rápidas não satisfazem as necessidades do usuário final. Por outro lado, a abordagem centrada no usuário, presente no DT, gasta mais tempo para pesquisa e análise antes do desenvolvimento e, por consequência, leva mais tempo para entregar o produto [48]. Neste caso concluí-se a possibilidade de colocar na balança e verificar o quão necessário é a dedicação inicial sobre o entendimento do problema através do uso do DT para facilitar e amadurecer os requisitos que servirão de entrada em um modelo ágil.

Outro fator importante sobre a semelhança dos métodos ágeis com o DT, é que ambos enfatizam a capacidade adaptativa do planejamento, desenvolvimento evolutivo, entrega antecipada e melhoria contínua com propósito de encorajar a resposta rápida e flexível à mudança [5]. Estas semelhanças destacadas nos artigos encontrados mostra que o DT vem com propósitos parecidos dos métodos ágeis. Porém, cada abordagem enfatiza objetivos que se complementam, ou seja, o DT fortalece a inserção do cliente ou usuário final na fomentação dos requisitos que atendam de fato a necessidade, gerando informações com valor de entrada para os métodos ágeis, gerando a execução de um processo rápido e eficiente, aumentando a garantia do desenvolvimento da solução, finalizando ambas abordagens juntas e atendendo, não apenas os requisitos definidos, mas sim construindo uma solução que atenda as necessidades dos usuários.

De acordo Nedeltcheva e Shoikova [86], o DT ajuda a entender o trabalho que deverá ser feito e, uma metodologia ágil como Scrum, oferece autonomia para decidir como fazer. Ambas formas assemelham-se por serem iterativos e com a mentalidade e abordagens necessárias para alcançar com sucesso o objetivo. As organizações estão acoplando os processos do DT e *Agile* com objetivo de identificar a solução correta e, em seguida, focar na construção de um produto melhor. O entendimento do problema abre o caminho para a execução de soluções mais criativas e úteis por meio de processos ágeis, ou seja, oferece a verdadeira satisfação do cliente.

Para o final, pode-se agrupar a terceira e quarta frentes encontradas nos artigos. Entende-se melhor o problema e busca a inovação, respectivamente. O DT é definido como uma abordagem centrada no ser humano para inovação na qual utiliza um *kit* de ferramentas de *designer* para integrar as necessidades das pessoas e possibilidades de tecnologia [50]. Focado como um gerador de ideias e inovação, reconhecido como um sistema iterativo com processo que consiste em diversas etapas para o entendimento do problema com propósito de oferecer uma solução inovadora [1]. Luchs [79] descreve o DT como melhor aplicado em situações em que o problema, ou oportunidade, não está bem definido e a ideia ou conceito inovador é necessário, isto é, uma ideia que tenha um impacto significativo e positivo. Quando aplicado sobre a criação de protótipos, por exemplo, não

referem-se aos protótipos prontos e totalmente funcionais mas sim para o fornecimento de uma experiência básica de um produto ou característica do mesmo, traduzindo para o desenvolvimento de software, para a criação de um sistema inovador ou funcionalidade.

Também, de acordo com Batista, Borba e Souza [10], o DT não está dentro da Engenharia de Software, mas participa do processo. Visando o olhar em *startups* de software, o DT pode ser utilizado para transformar ideias em propostas criativas de soluções. Desta forma utiliza-se na parte de levantamento, avaliando diferentes alternativas para explorar as oportunidades de inovação, concebendo e prototipando ideias, avaliando alternativas de solução, especificando curvas de valor da solução proposta, ilustrando potenciais cenários de uso e definindo uma visão da solução através de diversas técnicas trabalhadas nos espaços de trabalho do DT.

3.3.2 QP2: Contexto de uso do DT no desenvolvimento de software

Para responder esta questão realizou-se uma análise para estabelecer semelhanças em relação ao contexto em que o DT é aplicado na Engenharia de Software, seja no levantamento dos requisitos ou no processo de desenvolvimento. A Figura 3.6 mostra a porcentagem que o DT foi encontrado em cada contexto. Na primeira categorização, pode-se descrever que existe uma forte relação entre o DT e o *agile*, apresentando 46,30% nos artigos extraídos. Isso leva a considerar uma possível importância do DT no âmbito ágil sobre o processo de software, ainda de forma a melhorar e potencializar o seu desenvolvimento.

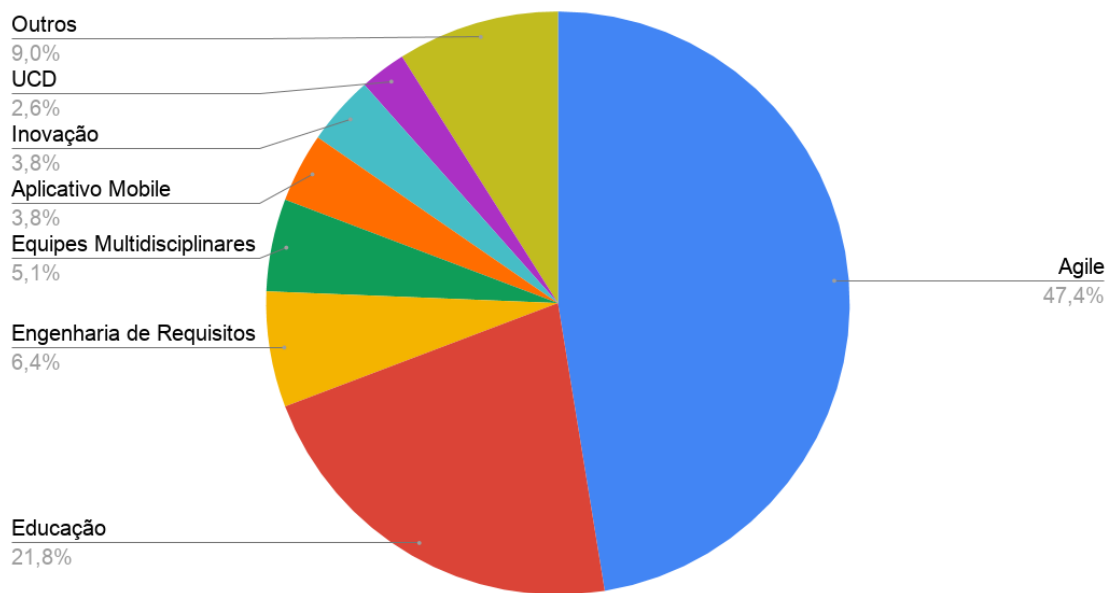
Outro ponto que percebe-se com o gráfico da Figura 3.6, é a representatividade que o DT apresenta-se no âmbito da educação sobre a área de desenvolvimento de software. Percebe-se desta forma que o DT não vem apenas ainda como uma forma de apoiar os métodos ágeis, conforme já respondido na QP1, mas também vem sendo uma abordagem na educação para fortificar o ensino, em sua prática, de todo o processo.

Para o próximo contexto percebe-se a ER, que pode ser entendida e inserida com auxílio do contexto ágil e também por se tornar uma das grandes formas que auxilia no entendimento do contexto para verificação e validação por seu processo de produção e conter em sua gerência a possível mudança de requisitos e qualidade, sendo estas características encontradas tanto nas abordagens ágeis como também no DT.

Analisando na Figura 3.6 a forte representatividade do Agile, sendo ele praticamente o contexto mais utilizado junto com o DT, discrimina-se os 46,30% sobre as referências e apresenta-se na Tabela 3.8 suas relações percentuais sobre o total.

De modo geral, o que mais apresenta-se é o termo de métodos ágeis, indiferente do método específico. Isso pode-se representar que o DT está mais próximo de fato dos

Figura 3.6 – Uso do DT em Diversos Contexto



FONTE: Autor, 2019

Tabela 3.8 – Agile x DT

% Total	% Parcial	Agile	Referência	Qtd.
46,3%	23,76%	Métodos Ágeis	[87], [71], [27], [70], [88], [93], [86], [80], [83], [78], [10], [74], [29], [2], [48], [20], [21], [5], [44], [14]	19
	10,01%	Lean/Lean UX/Lean Startup	[36], [10], [41], [74], [29], [31], [5]	8
	8,75%	Scrum	[23], [107], [6], [36], [32] [26], [31]	7
	1,26%	XP	[105]	1
	1,26%	Kanban	[27]	1
	1,26%	Agile UX Design	[2]	1

FONTE: Autor, 2019

valores dos métodos ágeis e não exclusivamente de algum determinado método em específico.

Descobriu-se também o uso do DT em conjunto com mais de um contexto sobre o mesmo estudo, não apenas auxiliando ou participando com apenas uma abordagem específica, mas sim sobre um conjunto de mais de um contexto, ou seja, o DT mostra-se ser uma forma de beneficiar um processo composto por mais de uma abordagem de trabalho. A Tabela 3.9 mostra os conjuntos que o DT se relacionou.

Para conhecimento também discrimina-se na Tabela 3.10 o conjunto dos outros contextos, representado por 8,8% na Figura 3.6.

Tabela 3.9 – Estudos que Trabalharam o DT Integrado com Outro Contexto

Composto	Referência
Scrum + Lean + Oceano Azul	[36], [10]
Agile + Lean UX/Startup	[74]
Agile + Lean Startup	[29], [31]
Agile + Lean	[5]

FONTE: Autor, 2019

Tabela 3.10 – DT em Outros Contextos

Outros	Referência
<i>Crowdsourcing</i>	[84]
BPM	[102]
CPM/PDD	[80]
Mudar processo de Desenvolvimento	[52]
<i>Rapid Application Development (RAD)</i>	[3]
HCD	[58]
<i>Participatory Design</i>	[87]

FONTE: Autor, 2019

3.3.3 QP3: Modelos, métodos, técnicas e/ou abordagens utilizadas do DT na Engenharia de software

Esta questão foi a mais elaborada visto a diversidade de modelos que representam o uso de DT. Sobre estes o modelo mais citado foi da *Stanford University's D-School*, composto por um método seguido por fases de Empatia, Definição, Ideação, Prototipação e Teste. Os modelos possuem fases diferentes mas acabam seguindo a mesma linha de raciocínio para a execução do DT no contexto inserido. Muitos modelos são adaptados para atender a realidade do contexto e outros seguem alguns modelos já conhecidos, como modelo do Brow (conhecido também como modelos 3I), *Stanford University's D-School* ou HPI. Para entender as fases contidas em cada modelo de processo e entender do uso e contexto de cada um, seguem as descrições, contexto utilizado e definição das fases de cada modelo de processo.

Os modelos encontrados nos artigos trazem não apenas exemplo de uso na área mas também alguns foram citados como embasamento e referencial teórico. Dentre estes, 10 foram aplicados na indústria. São eles:

1. Sandino et al. [101]:

- **Contexto:** Esta metodologia foi aplicada em um projeto simulador em tempo real para treinamento de operadores em pulverização de concreto, utilizando esta me-

metodologia DT para obter informações necessárias para o desenvolvimento desta aplicação. Esta aplicação foi desenvolvida de acordo com os requisitos e os resultados trazidos de todo este modelo DT.

- **Definição:** Definido em cinco etapas, inicia-se com a estruturação para o trabalho futuro da equipe, em seguida, explora tudo o que se relaciona ao projeto, proporcionando assim um ponto de partida para divergir nas inúmeras soluções possíveis que serão selecionadas posteriormente. Após o processo, a fase de prototipagem é estimulada, seguida pela escolha dos papéis de cada um dos integrantes da equipe, colocando variáveis de tempo e recursos. No final, a equipe apenas pode considerar a conclusão do projeto quando ele é revisado e verificado se está atendendo às necessidades do usuário, mesmo que o projeto já esteja em uso. As etapas são:
 - **Definir:** Os *designers* ou o projeto definem uma estrutura apropriada para o trabalho a ser feito. Essa estrutura pode ser vista como uma série de restrições que orientam o trabalho subsequente.
 - **Explorar:** Busca coletar informações sobre todos os aspectos do projeto a ser desenvolvido, tais como usuários em potencial e suas necessidades, soluções anteriores que foram projetadas para o mesmo problema, entre outras.
 - **Idealizar:** É usado para identificar as questões relevantes para as pessoas envolvidas na atividade e para gerar o maior número possível de ideias para obter respostas às necessidades.
 - **Prototipar:** Destina-se a construir protótipos detalhados ou funcionais trazendo informações úteis para serem discutidas com as partes interessadas, especialmente aqueles que serão usuários finais da solução futura.
 - **Selecionar:** Usado para selecionar os protótipos que resolvem o problema.
 - **Implementar:** N/A.
 - **Revisar:** Fornece a apresentação dos resultados aos usuários e as análises são adequadas para os propósitos para os quais foram desenvolvidos. Desta forma, o *feedback* é capturado para avaliar o trabalho realizado.
- **Diagrama:** Segue Diagrama do modelo na Figura 3.7.

Figura 3.7 – Modelo Sandino et al.

Define ► Explore ► Ideate ► Prototype ► Select ► Implement ► Review

FONTE: Sandino et al. [101]

2. Nordstrom [29]:

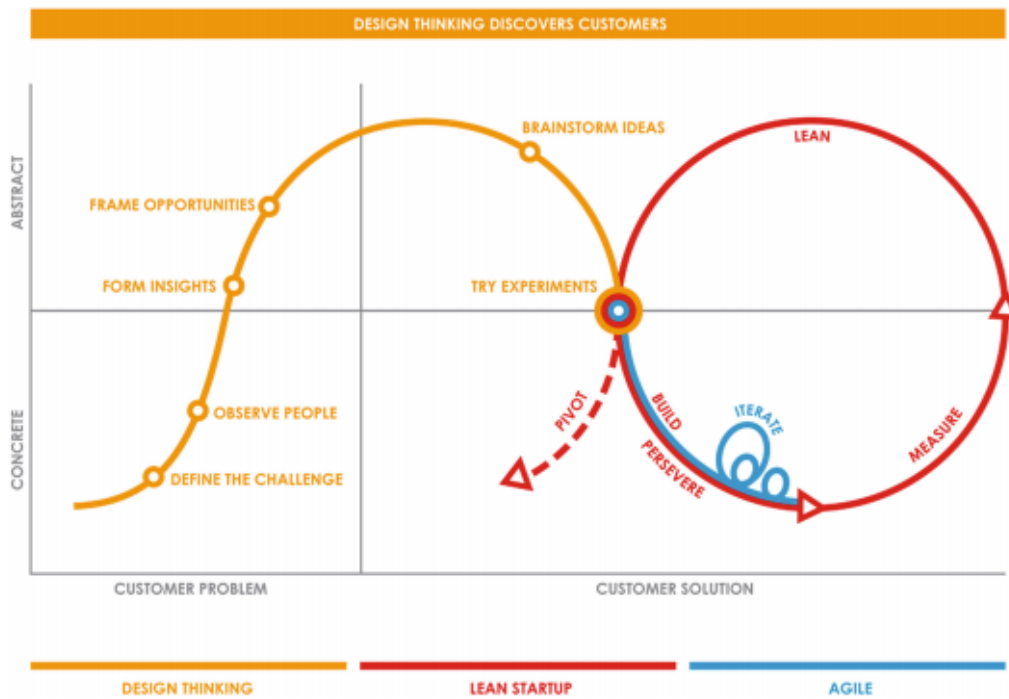
- **Contexto:** Combinando Design Thinking, Metodologia Ágil e Lean Startup - O Modelo Nordstrom foi introduzido em uma equipe de TI que foi incubada no BlackBerry Tech Center Recife (Brasil). A equipe foi composta por dois desenvolvedores e um designer. O Tech Center foi escolhido por causa da possibilidade de monitorar as equipes nas atividades diárias por um ano.
- **Definição:** Nordstrom é um modelo de desenvolvimento composto pelo uso de DT, Lean Startup e *Agile*. Argumenta-se que em todo desenvolvimento de um novo produto deve começar com DT porque a equipe precisa descobrir as necessidades dos usuários, identificar os problemas e propor soluções. Todas as práticas da DT seguem o IDEO UCD Toolkit⁷. Somente após a implementação passo a passo da IDEO, a equipe é capaz de iniciar os próximos modelos, seguidos pelas práticas ágeis e a filosofia Lean. O uso de DT neste modelo oferece uma maneira melhor de resolver problemas, pois fornece um conjunto de técnicas sobre como abordar os usuários em caso de dúvidas. As etapas são:
 - **Defina o desafio:** Obtenha informações sobre o desafio de design (problema a ser resolvido) e defina a sequência de atividades para sua solução.
 - **Observe as pessoas:** Atividade de imersão, que é um momento de observação para o designer observar o máximo que puder, para registrar exatamente o que ele vê e ouve obtendo detalhes e dados detalhados daqueles com as partes interessadas.
 - **Forme insights:** Atividade que visa criar frases do Insight, que são essenciais para a atividade de brainstorming, pois permitem direcionar possíveis discussões sobre as soluções.
 - **Quadro de Oportunidades:** Crie uma representação visual como um mecanismo para entender os dados, sendo capaz de destacar os principais relacionamentos e desenvolver a estratégia da solução. Os padrões começarão a surgir e deverão ser desenhados, inicialmente com estruturas mais simples, como os diagramas de Venn, por exemplo, que serão úteis para construir conhecimento sobre o desafio.
 - **Ideias de brainstorming:** O objetivo desta atividade é a geração de inúmeras ideias para que, a partir delas, seja possível ter soluções que atendam às necessidades.
 - **Experimente:** O objetivo é aplicar uma prototipagem ao vivo que visa testar a solução em condições reais. Esse procedimento pode durar de alguns dias a algumas semanas e é uma oportunidade de aprender como a solução

⁷<http://www.designkit.org/methods#filter>

funciona na prática. Protótipos vivos são sobre entender a viabilidade da ideia.

- **Diagrama:** Segue diagrama do modelo na Figura 3.8.

Figura 3.8 – Modelo Nordstrom



FONTE: Paula e Araujo [29]

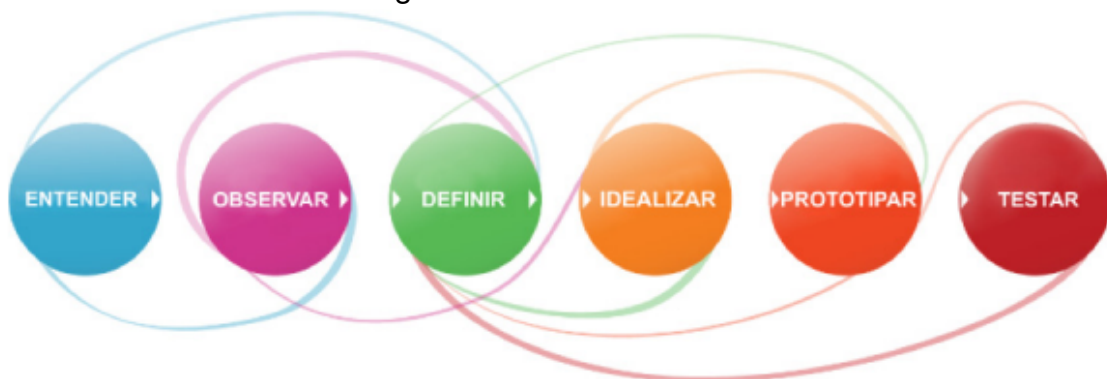
3. *Hasso-Plattner Institute D-School* [80][9][31]:

- **Contexto:** Os autores [80] examinam o uso da abordagem Modelagem de Propriedades de Características / Desenvolvimento Orientado a Propriedades (PM/PDD) para mapear o processo de desenvolvimento ágil, incluindo o DT. Eles aplicam o método proposto em trabalho interdisciplinar entre duas instituições de pesquisa e um provedor de serviços com o objetivo de investigar a aplicação de métodos inovadores de desenvolvimento de produtos, especialmente a fabricação de máquinas. A partir de estudos de caso, como o desenvolvimento de uma ferramenta de programação na indústria automotiva, eles propuseram o modelo.
- **Definição:** Inicialmente, busca compreender o problema e o contexto, captando informações relevantes, recolhendo insights sobre as necessidades do usuário, que podem ser obtidos por meio de entrevistas com os clientes ou por meio de pesquisas. Por sua vez, os participantes compartilham insights obtidos anteriormente, convertendo-os em uma visão de compreensão. Durante a etapa de Ideação, a equipe multidisciplinar trabalha com possíveis soluções que são prototipadas (o núcleo desse modelo) colocando as ideias em prática. Finalmente, testa para avaliar se atende às necessidades e exigências do usuário.

Este modelo é utilizado para uma variedade de aplicações, como para a educação, com a construção de objetos virtuais de aprendizagem [26], que são ferramentas que permitem a aprendizagem através de tecnologias. Em [9], foi aplicado a inovação no desenvolvimento de interfaces de usuário. Os autores sugerem que as etapas do DT sejam aplicadas e em conjunto com as fases do desenvolvimento para obter vantagens mais significativas e, para atestar, propõe a aplicação do DT no design de interface de usuário de busca via protótipos participativos, que mescla usuários e desenvolvedores. As etapas são:

- **Entender:** Entenda o problema e o contexto através da coleta de informações sobre o tema ou problema a ser resolvido.
 - **Observar:** Coleta insights sobre as necessidades do usuário, realizando entrevistas com os clientes e pesquisas. Além disso, externalize os problemas futuros dos usuários.
 - **Definir:** Coleta as informações das duas primeiras fases e as compartilha entre o grupo e as converte em uma estrutura visual.
 - **Idealizar:** Nesta fase, uma equipe interdisciplinar passa pelos processos que observou até o momento e cria uma solução com oportunidades de mudança durante a implementação.
 - **Prototipar:** A fase central do DT é o Protótipo porque transforma ideias em ação. Permite visualizar e comunicar ideias com a ajuda de protótipos rápidos e baratos com papel.
 - **Testar:** Avalia os requisitos e as necessidades do usuário.
- **Diagrama:** Segue diagrama do modelo na Figura 3.9.

Figura 3.9 – Modelo HPI



FONTE: Júnior e Nascimento [26]

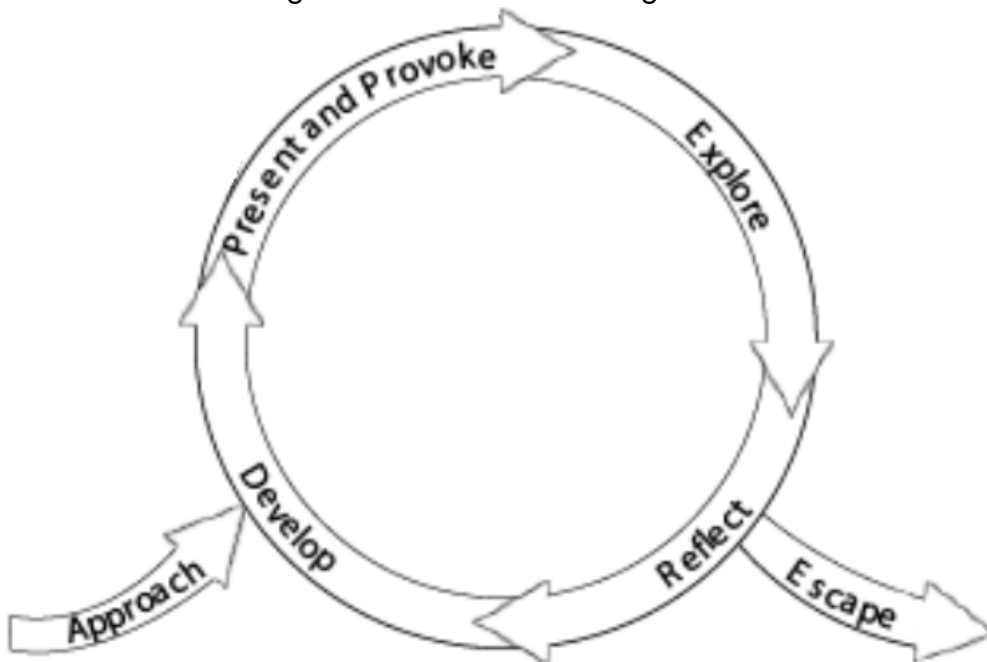
4. *Diving Board* [87]:

- **Contexto:** Este modelo foi aplicado como um meio criativo para explorar com os membros da comunidade Tiree possíveis mecanismos digitais que podem ajudar as comunidades a se adaptarem a um cenário energético onde os recursos não estão continuamente disponíveis. Foi executado uma série de oficinas de design em intervalos de aproximadamente 3 semanas.
- **Definição:** O modelo consiste nas etapas de Aproximação, Desenvolver, Apresentar e Provocar, Explorar, Refletir e Escapar. Na abordagem, uma primeira reunião é realizada para que os participantes se conheçam e compartilhem suas histórias. Esses participantes são pesquisadores e partes interessadas, que podem fornecer informações úteis sobre o contexto, e os pesquisadores podem fornecer *insights* sobre metodologia e sua experiência. Em desenvolvimento, protótipos de baixo nível são construídos, com base nas informações obtidas no estágio anterior e no feedback dos participantes, que tomam conhecimento da tecnologia que será desenvolvida. Na fase de apresentar e provocar, a apresentação dos protótipos é feita para que, através de um ambiente colaborativo, se possa realizar a ideação de tais protótipos e discutir o seu atendimento às necessidades. O Explore, por sua vez, estabelece uma oportunidade para o espaço problemático ser explorado com os participantes e também oferece a oportunidade de compartilhar informações com a comunidade participante. Em seguida, as partes interessadas refletem sobre o *feedback* e as informações obtidas a partir dos passos anteriores, a fim de obter uma lista de requisitos, assim como o último passo para realizar a prototipagem e, portanto, encerrar o processo por meio da atividade de escape. Esse modelo foi aplicado em [87], ligado à *Social Software Engineering* (Social SE) como desenvolvimento e manutenção de software, com o objetivo de promover uma mudança positiva na sociedade, tendo um estudo de caso OnSupply, que envolveu trabalhar em parceria com uma comunidade a explorar o potencial de sincronizar o uso doméstico diário de energia com a disponibilidade de um suprimento de energia que inclui um componente de energia renovável que varia muito do tempo. Como resultado da aplicação desse modelo, em conjunto com o *framework* ágil e participativo do *Speedplay*, eles perceberam que essa abordagem fornece um poderoso mecanismo para gerar e desenvolver soluções de software para problemas não especificados e abertos. As etapas são:

- **Abordagem:** É uma primeira reunião, importante para que tanto pesquisadores como partes interessadas compartilhem sua própria história. As partes interessadas podem fornecer informações contextuais sobre a comunidade e os pesquisadores podem fornecer informações sobre sua metodologia e experiência.

- **Desenvolver:** Um passo que prevê a construção de protótipos de baixo nível baseados em artefatos e ideias apresentados em workshops anteriores e feedback da comunidade. Útil para dar aos participantes uma visão inicial de quais tecnologias estão disponíveis ou poderiam ser desenvolvidas.
 - **Apresentar e Provocar:** Pretende-se que os participantes apresentem a mais recente iteração dos protótipos físicos, que são projetados para incentivar a ideação mais significativa dos outros participantes através de discussão e feedback sobre os artefatos.
 - **Explorar:** Facilita a exploração do espaço do problema com os participantes da comunidade e também permite que eles compartilhem mais informações contextuais sobre a comunidade.
 - **Refletir:** Pesquisadores e participantes refletem sobre o feedback mútuo e sobre assuntos cobertos pelo workshop anterior.
 - **Escapar:** Uma vez que o problema tenha sido explorado e uma lista de requisitos tenha sido eliciada, o desenvolvimento de protótipos que tentam abordar o espaço do problema pode começar.
- **Diagrama:** Segue diagrama do modelo na Figura 3.10.

Figura 3.10 – Modelo Diving Drive



FONTE: Newman et al. [87]

5. *Stanford University's + Hasso-Plattner Institute of Design d.school* [15]:

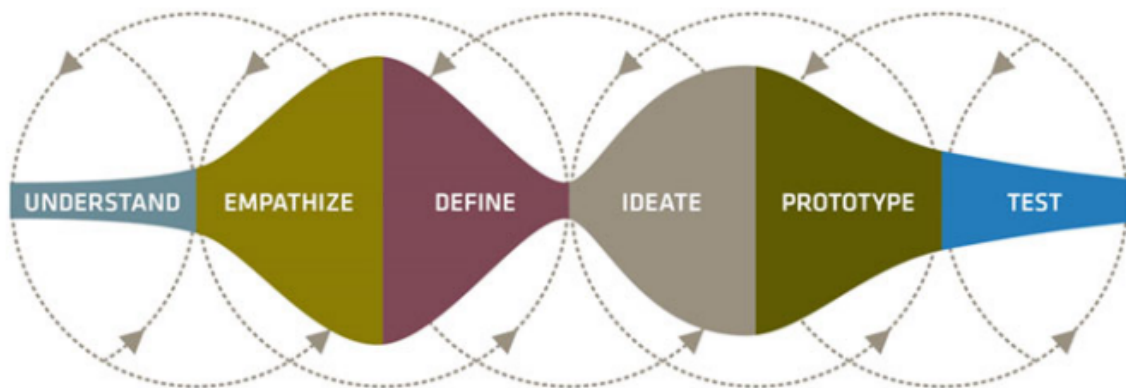
- **Contexto:** Projeto 1 - Compras *Online* de Jóias O projeto de DT foi adaptado para caber em 5 dias de oficina. O design A equipe de pensamento consistia de 12 pessoas de diferentes profissões (três representantes de clientes da área de TI do cliente, dois designers da Web, dois aplicativos desenvolvedores, dois desenvolvedores de conceito, um secretário e dois moderadores). Projeto 2 - Desenvolvendo Software Inovador para Dentistas Para este projeto, a equipe de pensamento de design consistia de especialistas em TI, educadores, dentistas e cientistas sociais. Os clientes a serem pesquisados eram especificamente selecionado pelo cliente. O pré-requisito era que os dois dentistas também como especialistas não-médicos (usuários da futura solução) deveriam ser pesquisados.
- **Definição:** Esse modelo baseia-se nos seguintes princípios: as necessidades do usuário como o centro das atenções, o entendimento profundo, e não o grande número de casos, equipes interdisciplinares e processos claros a serem seguidos. Estes princípios específicos diferenciá-lo de outros processos criativos, tornando-o útil para o contexto de TI. Além disso, pode ser aplicado no processo de desenvolvimento de software, começando com a definição de como a estrutura DT será. Para tanto, uma equipe de especialistas em TI e profissionais de outras áreas de abrangência da solução a ser proposta deve ser criada.

Após o estágio do processo de DT nesse contexto, as pessoas afetadas (partes interessadas) e os usuários de um sistema começam a trabalhar na fase "Compreender". Neste ponto, as personas devem ser desenvolvidas, através da etapa Enfatizar, de modo que seja possível discutir suas características e servir como fonte para os requisitos de software. Essas atividades devem ser realizadas em contextos colaborativos, pois permitem a geração de muitos *insights* para a equipe de DT, estabelecendo o estágio de empatia. Por fim, uma série de ideias foram geradas para cada pessoa durante a ideação, levando os protótipos a serem implementados (prototipagem). Durante a fase final (Teste), perguntas devem ser feitas sobre as soluções para entender por que isso deveria ser apropriado para uma pessoa. As etapas são:

- **Entender:** O foco principal é entender o problema que é colocado e seus elementos essenciais. A ideia é principalmente pesquisar dados e fatos, identificar atores e situações relevantes e explorar o possível escopo do design.
- **Enfatizar:** Esta fase é essencialmente sobre a criação de empatia com os usuários, a fim de compreender suas visões do mundo ou do problema. Essa fase expande o foco (divergência) definido pelo desafio e fornece *insights* profundos sobre o que os usuários pensam e sentem.

- **Definir:** Neste ponto do processo de DT, todas as informações e percepções coletadas até agora são examinadas em detalhes, padrões são identificados. Todos os membros da equipe compartilham todas as suas impressões e informações uns com os outros. As impressões e experiências coletadas são usadas para reformular e especificar o problema inicial (convergência).
 - **Idealizar:** Esta fase começa com o desafio reformulado (ponto de vista). Inicialmente, as personas são desenvolvidas para poder concentrar-se no grupo-alvo para a solução futura da forma mais concreta possível. O *brainstorming* pode ser usado para gerar um grande número de ideias. Após um grande número de ideias ter sido coletado dessa maneira, as soluções são avaliadas e as promissoras são selecionadas.
 - **Prototipar:** Destina-se a construir protótipos que servem para visualizar uma ideia de forma rápida e concisa.
 - **Testar:** Destina-se a testar os protótipos em um contexto real que será usado. O teste na situação real cria empatia para o grupo-alvo e suas necessidades e fornece uma visão dos fatores relacionados ao contexto a serem considerados para a solução final.
- **Diagrama:** Segue diagrama do modelo na Figura 3.11.

Figura 3.11 – Modelo *Stanford University's + HPI*



FONTE: Carell et al. [15]

6. *Stanford University's D-School* [76] [103] [115]:

- **Contexto:** Projeto 1 - O processo de DT foi utilizado para desenvolver um aplicativo chamado LIKE, no qual visa aumentar a visibilidade das contribuições dos profissionais, facilitando o reconhecimento entre os colegas, explicando o conhecimento através da gravação de depoimentos e permitindo o recebimento de recompensas [76].

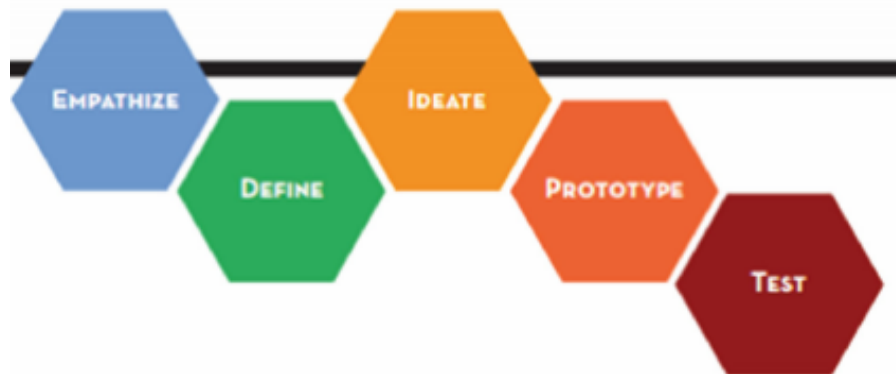
Projeto 2 - A estrutura de pensamento de design do PPM foi introduzida para compreender como os dados podem ser coletados e analisados [103].

Projeto 3 - A empresa de cartão de crédito da Suíça resolveu um problema de gerenciamento de relacionamento com o cliente usando o DT para produzir um novo aplicativo de *tablet* para seus clientes. [115].

- **Definição:** O modelo de *Stanford* permite a produção de soluções inovadoras baseadas em um processo de cinco etapas: (i) Empatia para reunir os interessados (partes interessadas, usuários, especialistas) para entender as necessidades dos usuários da solução. Sessões de *brainstorming* são realizadas para promover a criatividade. (ii) Definição que fornece a definição de declarações de problemas com base na síntese das informações obtidas durante o passo de empatia e, em seguida, compila-as em *insights* que podem ser resolvidos. Em seguida, o método de *Stanford* estrutura o estágio de ideação (iii), que incentiva a produção de inúmeras propostas de solução (muitas ideias) na forma de enunciados para o problema, de forma colaborativa, com a prática da criatividade de equipes multidisciplinares. A quarta e quinta etapas são (iv) prototipagem e (v) testes, nos quais a prototipagem se concentra no desenvolvimento de diferentes níveis de protótipos para torná-los compreensíveis para os stakeholders e, teste, visa apresentar soluções projetadas para capturar as reações dos clientes. Este modelo é amplamente utilizado em diversas aplicações. Isso inclui o desenvolvimento de soluções de dados governamentais abertas, sistemas incorporados e engenharia de requisitos de cidades inteligentes. Desta forma, este modelo visa promover o desenvolvimento de empatia com o usuário, endossando uma imersão na experiência do usuário, projetando perspectivas do usuário em sessões de prototipagem de inúmeras opções e pode ser integrado com metodologias de desenvolvimento de software como, por exemplo, métodos ágeis, que se concentra no desenvolvimento incremental e no funcionamento das condições do produto, o mais rapidamente possível, através de versões menores, porém mais frequentes, do sistema. As etapas são:
 - **Empatia:** Entender o que o usuário ou cliente precisa, em termos do que o cliente sente.
 - **Definição:** Definir as necessidades do cliente em termos de um problema que pode ser resolvido.
 - **Idealizar:** Proposta de uma solução (produto, serviço, experiência) que atenda às necessidades do cliente.
 - **Prototipar:** Criar uma solução que cubra uma seleção dos recursos exigidos pelo usuário, para que o cliente final possa experimentá-lo.
 - **Testar:** Utilização da solução, de forma que o cliente possa fornecer *feedback* oportuno e o processo possa iterar.

- **Diagrama:** Segue diagrama do modelo na Figura 3.12.

Figura 3.12 – Modelo *Stanford University's*



FONTE: Adikari et al. [103]

7. Meinel e Leifer - (Hasso Plattner I HPI) [58]:

- **Contexto:** Quatro equipes dinâmicas de um total de sete membros da equipe participaram ativamente de dois designers de negócios em co-design. Após a criação, todos os membros da equipe trabalharam com a nova estrutura da equipe por sete semanas, e o estudo teve como objetivo criar uma estrutura de equipe de alto desempenho que promova a criatividade e a inovação por meio do uso de um sistema de gerenciamento de ideias. Esta pesquisa foi conduzida para avaliar se a aplicação da DT pode levar ao desenvolvimento de equipes altamente funcionais que proporcionam desempenho e resultados de equipe eficazes.
- **Definição:** O DT definido neste modelo é uma metodologia centrada no ser humano que integra a expertise de diferentes disciplinas, como design, ciências sociais, engenharia e negócios, para criar problemas, resoluções e design. Ele se concentra no usuário final, na colaboração multidisciplinar e na melhoria iterativa para produzir produtos, sistemas e serviços inovadores. Resume-se que o DT trazido por este modelo tem uma abordagem primária de HCD para obter uma compreensão mais profunda das equipes de projeto e melhorar a situação atual para um estado futuro desejável sendo projetado com todos os usuários (membros da equipe). As etapas são:
 - **(Re)definir o problema:** Definição do problema a ser resolvido, envolvendo a identificação dos *stakeholders*. Em uma segunda iteração, ele servirá como uma análise dos resultados do teste.
 - **Entender:** Entenda o problema e o contexto através da coleta de informações sobre o tema ou problema a ser resolvido.

- **Idealizar:** Fase para gerar e desenvolver ideias que podem levar a soluções. Pode ser usado a técnica de bodystorming para obter informações dos usuários.
 - **Prototipar:** Construindo uma visão gráfica das ideias geradas para apresentar a solução.
 - **Testar:** Pretende-se testar os protótipos em um contexto real que será utilizado para avaliá-los.
- **Diagrama:** Segue diagrama do modelo na Figura 3.13.

Figura 3.13 – Modelo Meinel e Leifer - (HPI)



FONTE: Adikari e Keighran [58]

8. Human-Centered Agile Workflow (HCAW) [41]:

- **Contexto:** O modelo HCAW está atualmente em uso em três projetos de design e implementação de serviços no LINC. Algumas partes das equipes dos projetos internacionais são co-localizadas, mas a maioria dos membros da equipe trabalha em locais remotos. A *LINC Interaction architects GmbH* é uma empresa de design de serviços com sede em Munique, na Alemanha.
- **Definição:** O HCAW combina o desenvolvimento de software ágil e colaborativo com o design centrado no cliente. Sua metodologia de desenvolvimento de software segue a abordagem estabelecida do Scrum, que por sua vez suporta os valores centrais do Manifesto Ágil. A abordagem de design segue os princípios do HCD. Sobre sua aplicação prática, não há restrições sobre a disciplina profissional dos membros da equipe, sua localização, tamanho da equipe ou escopo do projeto. Além disso, em relação ao uso do DT, ele é usado em cada processo de design do sprint. No primeiro sprint, todas as etapas são realizadas, desde a pesquisa até a avaliação. Nos outros sprints, apenas a ideação é

completada, focando na cocriação com workshops, apresentação de protótipos, validados com equipes de desenvolvimento, e avaliação que pode ser entregue como um relatório, servindo como artefatos de entrada para a especificação. Assim, o DT é trabalhado no estágio de concepção da ideia e serve como entrada para o processo de implementação.

- **Pesquise:** Os pesquisadores começam a planejar suas atividades escolhendo métodos adequados, realizando pesquisas e concluindo sua contribuição publicando um relatório do *Insight*.
 - **Idealize:** Execução de um Workshop de co-criação, que é hospedado pela equipe de ideação, convidando colegas de equipes de pesquisa e prototipagem e potencialmente outros para discutir os *insights*.
 - **Prototipe:** A equipe de prototipagem é baseada no resultado do estágio de ideação, constrói os protótipos e os apresenta para a equipe de avaliação.
 - **Avalie:** A equipe de avaliação decide sobre um método apropriado, conduz a avaliação e descreve suas descobertas no Workshop de Revisão, organizado pela equipe de ideação. Dependendo do relatório de avaliação e do resultado do workshop, as equipes de prototipagem e avaliação podem estar envolvidas no retrabalho e na avaliação do protótipo.
- **Diagrama:** N/A.

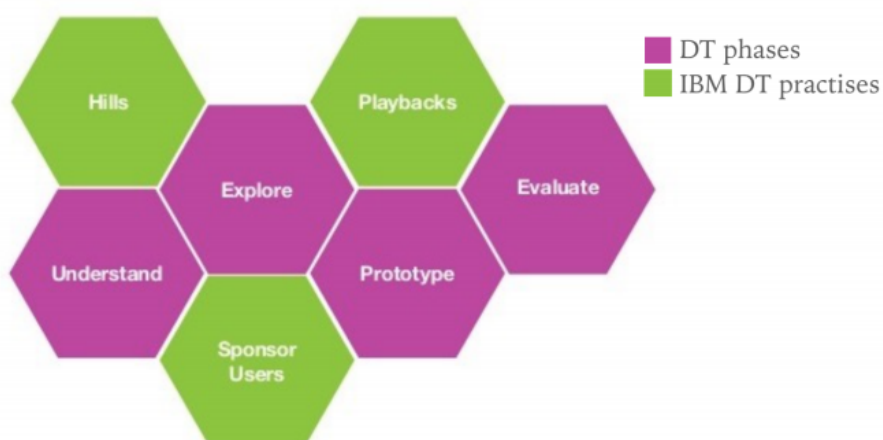
9. IBM-DT [78] [6]:

- **Contexto:** Projeto 1 - O IBM DT estende os princípios do DT a serem aplicados no desenvolvimento de software que captura as necessidades dos usuários na velocidade e na escala necessárias para o desenvolvimento incremental de software [78]. Projeto 2 - *Hackathons* [6].
- **Definição:** O objetivo da Estrutura de Desenvolvimento de Software do IBM DT é estender os princípios de DT a serem aplicados no desenvolvimento de software, capturando as necessidades do usuário na velocidade e na escala necessárias para o desenvolvimento incremental de software. Embora compartilhe algumas semelhanças com outros métodos de DT, ele possui algumas modificações, incluindo três práticas que são exclusivas da estrutura: Patrocinando Usuários, Reproduções e *Hills*. *Hill* introduz uma nova maneira de expressar as necessidades dos usuários nos requisitos do projeto. Cada *Hill* articula um objetivo claro e contém um escopo definido para ser atingido em um lançamento ou em um conjunto finito de lançamentos. Um *Hill* deve ser escrito para atender a um problema de usuário específico e claramente definido que é relatado pela pesquisa do usuário. Embora *Hills* seja escrito de uma perspectiva do cliente, eles também

ênfatisam interseções importantes entre as expectativas do usuário e os requisitos de negócios. As colinas são compostas de três elementos: "*Who*": aquele que descreve um usuário específico ou um grupo de usuários; "*What*": o que descreve um problema que precisa ser resolvido; e "*How*": um alvo mensurável para a conclusão de *Hill*. As reproduções são pontos de verificação quando a equipe do projeto e os Usuários do Patrocinador estão revisando o status do projeto e planejando as próximas etapas. As reproduções são um espaço seguro para fornecer e receber *feedback* e ocorre no final de cada fase de desenvolvimento do projeto e tem objetivos diferentes. As etapas são:

- **Entender:** A primeira fase do processo de DT consiste no modo Empatia, um conjunto de atividades para ajudar a entender os usuários, dentro do contexto de seus problemas. A primeira atividade a ser executada consiste em observar como os usuários interagem com seu ambiente.
 - **Explorar:** Concentre-se na geração de novas ideias, a fim de evitar soluções óbvias e, assim, aumentar o potencial de inovação. *Brainstorming* é uma técnica comum usada nesta fase que oferece pensamentos divergentes de forma a explorar novas ideias e soluções.
 - **Prototipar:** A geração iterativa de artefatos destina-se a responder a perguntas para resolver o problema de design. O objetivo da fase de protótipo é validar as ideias propostas durante a fase de exploração.
 - **Avaliar:** Solicita aos usuários *feedback* sobre o protótipo criado. Técnicas de avaliação da UX podem ser usadas para testar o protótipo.
- **Diagrama:** Segue diagrama do modelo na Figura 3.14.

Figura 3.14 – Modelo IBM-DT



FONTE: Lucena et al.[78]

10. *Divergent-Convergent based Design Thinking Model* [10]:

- **Contexto:** O projeto assume uma abordagem de co-design e, portanto, o consórcio multidisciplinar onde três museus participam (*Stichting Museon, Museu Al-lard Pierson e Museo Storico Italiano Della Guerra*) realiza oficinas de co-design que funcionam durante o primeiro ano do projeto. uma fonte valiosa de informações para visualizar os espaços e entidades de design necessários no CoDICE e também forneceram o contexto perfeito para iterativamente projetá-lo, obtendo *feedback* de uma equipe multidisciplinar, incluindo gerentes, pesquisadores, artistas, designers, curadores e desenvolvedores de software. Equipes heterogêneas estavam envolvidas em atividades como personas apreciativas e cenários que visavam criar um terreno comum para compartilhar conhecimento e ideias.
- **Definição:** O modelo visa a construção da solução em termos de divergir inicialmente através da proposta de muitas ideias para a solução para então, na sequência, convergir, momento de alinhar com as propostas feitas e tender à solução. A abordagem divergente e convergente da DT não cria o software, mas identifica a oportunidade e a traduz mais de perto da realidade. Em Borba, Glauber e Ricardo [10] apresentam o uso do DT juntamente com uma abordagem de Oceano Azul e Lean Startup. A Estratégia do Oceano Azul está preocupada em analisar os fatores competitivos do mercado. DT está preocupado em transformar ideias em soluções criativas. A abordagem Lean Startup preocupa-se em criar uma versão enxuta do produto que demonstre a entrega de valor ao público-alvo. As etapas são:
 - **Divergir:** O design divergente é um passo que visa gerar um conjunto múltiplo de ideias. Na fase de pensamento divergente, buscar informações relevantes e criar novas informações sobre a tarefa proporcionará uma melhor percepção e também equilibrará a falta de experiência empreendedora que uma equipe possa ter.
 - **Convergir:** O design convergente é focado na produção de soluções para ideias selecionadas. A última fase do processo implica colocar as ideias em ação, adotando um pensamento mais convergente e empregar uma solução inovadora.
- **Diagrama:** N/A.

Estes modelos apresentam divergências sobre os seus usos em diversas situações e contextos, porém todos são direcionados para a indústria. Demais modelos não descritos são utilizados tanto como referências para uso do DT como também usados para educação. Todo método de DT é caracterizado por um conjunto de técnicas que se fazem essenciais para sua execução. Este mapeamento identificou 46 técnicas utilizadas em diversos modelos encontrados. Para destacar as mais citadas, criou-se uma Tabela 3.11.

Tabela 3.11 – Técnicas de DT mais Citadas

Técnicas	Citações
Personas	[23] [24] [88] [39] [102] [108] [34] [80] [9] [19] [99] [78] [105] [112] [15]
Brainstorming	[23] [85] [87] [97] [6] [24] [102] [17] [108] [67] [26] [78] [10] [43]
Storyboard	[73] [88] [108] [86] [67] [78] [10] [112] [16]
Interview	[87] [102] [17] [108] [9] [26] [3] [78] [43]
Empathy Map	[73] [24] [39] [102] [108] [86] [67] [78] [81]
Observation	[17] [10] [112] [16] [43]
Scenario Map	[86] [10] [105] [112] [81]
Storytelling	[36] [17] [108] [81]
Questionnaires	[108] [101] [26] [78]
Customer Journey Map	[88] [19] [81]
Mental map	[17] [26] [76]
User Stories	[23] [99] [105]
Mockups	[6] [78]
Insight cards	[102] [108]
User Journey	[23] [102]
Workshop	[85] [24]
Sketching	[97] [17]
Fly the wall	[108] [101]
Quick Ethnography	[108] [101]
Exploratory research	[108] [26]
Roadmap Experience	[86] [67]
Search Desk	[67] [26]
Canvas	[67] [76]

FONTE: Autor, 2020

As outras 23 técnicas foram citadas apenas uma única vez. Após identificar as técnicas, foi realizado um *merge* sobre quais modelos utilizaram quais técnicas. Também, cabe descrevê-las para melhor entendimento de como são caracterizadas. A seguir uma descrição do uso destas técnicas relacionadas ao seus métodos de uso. As referencias inseridas a seguir apontam os estudos apresentaram o modelo a técnica. Entretanto não são todas técnicas que estão relacionadas ao modelo em si, visto que alguns estudos apenas trouxeram a técnica aplicada. A descrição de cada técnica foi escrita de acordo com o entendimento do autor sobre a leitura dos estudos deste mapeamento.

1. 5w2h:

- **Modelo:** *Diving Board* [87].
- **Descrição:** Uma técnica frequentemente usada para descrever atividades sobre o que será feito em termos de aprimoramento do desenvolvimento usando as palavras por que, onde, quando, quem, como e quanto.

2. *A day in the life*

- **Modelo:** Hiremath e Sathiyam [52] e Coutinho et al [25].
- **Descrição:** Uma espécie de etnográfico onde o pesquisador segue e observa um usuário durante um dia típico. O objetivo dessa atividade é que o pesquisador entenda a rotina e as operações diárias de um usuário e o que o usuário realiza por prática.

3. *Behaviour Archaeology*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** A equipe de projeto aprende como as pessoas organizam o espaço onde a atividade acontece, o que elas devem fazer, como usam e organizam seus objetos e assim por diante.

4. *Blueprint*

- **Modelo:** N/A.
- **Descrição:** Uma ferramenta operacional que explica as características da interação de serviço com detalhes suficientes para verificar, implementar e manter esse serviço. É baseado em um método gráfico que ilustra as funções do processo em termos de visibilidade para o usuário, ou seja, como o processo se relaciona com o usuário. O termo Blueprint está associado ao fato de que é comumente impresso em um fundo azul com letras brancas.

5. *BodyStorming*

- **Modelo:** Coutinho et al [25].
- **Descrição:** Uma técnica que visa obter novas ideias a partir da execução de experimentos em que são considerados aspectos físicos dos participantes frente a uma experiência. Por isso, visa fisicamente experimentar uma situação para obter novas ideias.

6. *Brainstorming*

- **Modelo:** Coutinho et al [25], *Diving Board* [87] e *Stanford University's + HPI* [15].
- **Descrição:** Uma técnica de criatividade colaborativa que visa gerar um vasto número de ideias para possíveis abordagens de solução para um problema específico.

7. *Brainwriting*

- **Modelo:** *Diving Board* [87]
- **Descrição:** Uma técnica que estimula a geração de ideias a partir da escrita no papel e depois a discussão com os outros participantes.

8. *Business Model Canvas (BMC)*

- **Modelo:** *Diving Board* [87]
- **Descrição:** Um modelo de gerenciamento para documentar ou criar modelos de negócios. É um mapa visual com componentes que descrevem a proposta de valor de um produto, infraestrutura, clientes e finanças.

9. *Card Sort*

- **Modelo:** Hiremath e Sathiyam [52] e Coutinho et al [25].
- **Descrição:** É um procedimento participativo, centrado no usuário, aplicado para elucidar as perspectivas, valores, desejos e comportamentos das partes interessadas, nas quais são dados vários itens em cartões individuais e é solicitado a categorizar repetidamente os itens de acordo com os critérios definidos pelo pesquisador. ou de acordo com critérios que façam sentido para os participantes.

10. *Character Profiles*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** É um método de síntese pelo qual uma equipe constrói um caractere (semi) ficcional que incorpora as observações humanas que a equipe possui, que podem incluir características, comportamentos e outros aspectos que a equipe identificou em seu grupo de usuários.

11. *Co-creation Workshops*

- **Modelo:** N/A.
- **Descrição:** Oficinas de cocriação envolvem múltiplos participantes de diferentes contextos compartilhando ideias em uma sessão colaborativa em um tempo e espaço específicos.

12. *Cognitive and Behaviour Map*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** Os designers pedem às pessoas envolvidas na atividade que mostrem como pensam e como fazem suas ações.

13. *Conceptual mind maps*

- **Modelo:** N/A.
- **Descrição:** São usados para organizar e visualizar conhecimentos incompreendidos, analisando problemas sérios, identificando respostas e atuando. Eles usam verbos de ação, como contribuições, causas, requer e levam a representar o espaço da solução.

14. *Customer Journey Maps*

- **Modelo:** *Diving Board* [87] e Dunne e Martin [2] [33]
- **Descrição:** É uma visão geral do processo pelo qual um cliente passa para atingir uma meta. Ele é usado para entender e atender às necessidades dos clientes e suas "dores". Ele começa compilando uma série de metas e ações do usuário em um esqueleto da linha do tempo. Então o esqueleto é desenvolvido com os pensamentos e emoções dos usuários para criar uma narrativa.

15. *Desk Research*

- **Modelo:** Coutinho et al [25].
- **Descrição:** Uma técnica que visa realizar um extenso levantamento de material já publicado em relação ao público-alvo ou produto (ou pessoa) pesquisado. A fonte de dados geralmente são documentos oficiais, pesquisas já realizadas ou material de mídia, como jornais, revistas e sites.

16. *Empathy Maps*

- **Modelo:** Coutinho et al [25] e *Diving Board* [87].
- **Descrição:** É uma visão usada para organizar o que é conhecido sobre um determinado tipo de usuário. Ele possibilita uma compreensão central das necessidades do usuário e ajuda na tomada de decisões. A representação de um mapa de empatia é geralmente dividida em quatro quadrantes que mostram o que o usuário diz, pensa, sente e sente.

17. *Error Analysis*

- **Modelo:** Hiremath e Sathiyam [52]
- **Descrição:** Essa técnica ajuda a equipe encarregada do projeto a identificar todos os erros possíveis na atividade e no protótipo.

18. *Extreme User interviews*

- **Modelo:** Coutinho et al [25].
- **Descrição:** Uma técnica que fornece entrevistas com substitutos, ou seja, pessoas de dentro da organização que falam diretamente com os usuários e estão bem conscientes de suas expectativas e necessidades não atendidas, que não foram consideradas como usuários finais à primeira vista. Usuários extremos são pessoas que apontam pontos muito diferentes dentro de um grupo de clientes.

19. *fishbowl*

- **Modelo:** N/A.
- **Descrição:** Permite a facilitação do diálogo em grandes grupos, concentrando-se em uma discussão em pequenos grupos em um círculo interno, enquanto o restante do grupo ouve e observa do círculo externo.

20. *Five Whys?*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** Uma técnica útil de resolução de problemas que fornece uma abordagem consistente para entender as causas. Cinco porquês devem ser aplicados em um modo livre, onde o foco é encontrar o "porquê" em vez de culpar "quem".

21. *Fly on the wall*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** Ajuda a entender como as pessoas se comportam enquanto realizam a atividade que está sendo analisada e os projetistas coletam as informações através da observação. É essencial que as pessoas não saibam que estão sendo observadas.

22. *How-Might-We-Question*

- **Modelo:** Coutinho et al [25] e *Diving Board* [87].
- **Descrição:** As perguntas "Como podemos"(HMW) são perguntas curtas que lançam brainstorm. É um formato para reformular problemas em declarações de oportunidade para estimular a criatividade, o brainstorming e a inovação. Exemplo: Como podemos tornar nossa entrada mais convidativa para que nossos visitantes se sintam mais em casa? As Perguntas da HMW são uma maneira simples de melhorar seus produtos ou serviços.

23. *I like I wish*

- **Modelo:** N/A.
- **Descrição:** Uma atividade de feedback de equipe realizada em sessões em que os membros de uma equipe envolvida em um projeto criativo recebem feedback positivo e construtivo em formato escrito e falado.

24. *Insight cards*

- **Modelo:** *Diving Board* [87].
- **Descrição:** Consistindo de reflexões baseadas em percepções reais obtidas ao longo dos estudos exploratórios, de mesa e em profundidade. São descobertas que mostram oportunidades de solução para o projeto.

25. *Interview*

- **Modelo:** Coutinho et al [25] e *Diving Board* [87].
- **Descrição:** Uma técnica para aprender mais sobre o usuário / cliente. Ele mostra como diferentes clientes se sentem sobre o problema que estamos tentando resolver e como eles podem consertá-lo.

26. *Living Labs*

- **Modelo:** Meinel e Leifer (HPI) [58].
- **Descrição:** É um local de múltiplos participantes para realizar projetos de inovação que seguem os princípios da inovação aberta e do usuário e se concentram na experimentação da vida real.

27. *Mind Mapping*

- **Modelo:** *Diving Board* [87] e Dunne e Martin [2] [33].
- **Descrição:** Um método eficiente que organiza todas as informações de forma visualmente legível. Eles usam um conjunto de formas e setas para organizar informações logicamente.

28. *Mockups*

- **Modelo:** Meinel e Leifer (HPI) [58].
- **Descrição:** Uma representação de como a solução de software será desenvolvida e funcionará como uma forma visual dos protótipos. Mockups de entrada uma parte do processo criativo e de desenvolvimento permite rápido e fácil idealização e iteração sobre a solução antes de investir no processo de codificação real.

29. *Narration*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** Informações sobre o projeto podem ser obtidas através da narrativa dos usuários, interagindo com o software. Pode ser usado para validar uma solução.

30. *Observation*

- **Modelo:** Coutinho et al [25], *Diving Board* [87] e Meinel e Leifer (HPI) [58].
- **Descrição:** Uma técnica que permite ao designer ter empatia com a experiência do cliente, compreender seu contexto, descobrir necessidades ocultas e ouvir seus comentários. Envolve observar e ouvir atentamente os usuários enquanto eles trabalham com um produto. Embora seja possível coletar dados muito mais elaborados, observar os usuários é uma maneira rápida de obter uma visão objetiva de um produto.

31. *Personal Inventory*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** Um estudo das relações que os indivíduos desenvolvem com seus pertences em casa ou no trabalho. Esse método usa informações sobre os relacionamentos das pessoas com produtos físicos e digitais para projetar protótipos ou modelos.

32. *Personas*

- **Modelo:** HPI [80], *Diving Board* [87] e Dunne e Martin [2].
- **Descrição:** Uma das técnicas mais conhecidas e usadas no DT, o Personas é uma representação abstrata de um usuário com características semelhantes. Uma persona é um personagem fictício que representa um tipo de cliente ou usuário do produto. Esse personagem fictício é criado com base em uma síntese do que foi aprendido sobre os clientes e usuários reais.

33. *Positioning Matrix*

- **Modelo:** *Diving Board* [87].
- **Descrição:** Uma ferramenta que permite uma análise estratégica das ideias geradas, utilizando a validação dessas ideias, bem como as necessidades das Personas criadas no projeto. O objetivo desta técnica é apoiar o processo de tomada de decisão, através da comunicação eficiente dos benefícios e desafios de cada solução, para que as ideias mais estratégicas sejam selecionadas e prototipadas.

34. *Rapid Ethnography*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** Uma coleção de métodos de campo para fornecer aos projetistas uma compreensão razoável dos usuários e suas atividades, dada a quantidade limitada de tempo gasto no campo de coleta de dados.

35. *Role-Playing*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** Fornece uma compreensão mais profunda de como o protótipo pode funcionar, entrando no suporte do usuário como designer.

36. *Scenarios Mapping*

- **Modelo:** Hiremath e Sathiyam [52] e *Diving Board* [87].
- **Descrição:** Os cenários são ferramentas úteis para comunicar ideias sobre ações do usuário. Os cenários de design de mapeamento também têm o benefício adicional de ajudar a formalizar ideias e a adotar abordagens criativas para essas ideias.

37. *Shadowing*

- **Modelo:** Hiremath e Sathiyam [52] e Coutinho et al [25].
- **Descrição:** Uma técnica na qual o pesquisador acompanha o usuário e observa como ele usa o produto ou serviço em seu ambiente natural.

38. *Social Network Mapping*

- **Modelo:** Hiremath e Sathiyam [52].
- **Descrição:** Permite que os designers identifiquem os diferentes papéis no processo; isto é, os diferentes empregos e responsabilidades de cada pessoa envolvida.

39. *Stakeholder Mapping*

- **Modelo:** Dunne e Martin [2] [33]
- **Descrição:** É um processo colaborativo de pesquisa, debate e discussão que se baseia em várias perspectivas para determinar uma lista importante de partes interessadas em todo o espectro das partes interessadas.

40. **Storyboard**

- **Modelo:** HPI [80]
- **Descrição:** É a representação de casos de uso por meio de uma série de desenhos ou imagens, reunidos em uma sequência narrativa. O storyboard mostra a manifestação de cada relacionamento entre os casos de uso e o usuário na criação da experiência.

41. **Storytelling**

- **Modelo:** Coutinho et al [25] e *Diving Board* [87].
- **Descrição:** Uma ferramenta que suporta a exploração de ideias, que através do uso de palavras simples, um contador deve ilustrar a solução, como se fosse uma história. Isso permite a comunicação da ideia ao grupo, permitindo que lacunas sejam preenchidas pelas sugestões de outros stakeholders e usuários.

42. **Surveys Questionnaires**

- **Modelo:** Hiremath e Sathiyam [52]
- **Descrição:** Técnica usada para obter informações dos usuários em relação às suas necessidades.

43. **Try it yourself**

- **Modelo:** N/A.
- **Descrição:** Obtém informações úteis fazendo com que a equipe de design use e interaja com o aplicativo projetado.

44. **Feedback grid**

- **Modelo:** *Diving Board* [87].
- **Descrição:** É uma matriz de quatro partes que facilita a estruturação de cotações de feedback e observações para cada protótipo que permite processar várias entrevistas de feedback, ajudando a decidir o que manter, o que iterar e o que descartar na próxima iteração.

45. **User story**

- **Modelo:** Nordstrom [29] e *Stanford D-School* [103]
- **Descrição:** São registros simples e curtos de um recurso narrado do ponto de vista da pessoa que deseja o novo recurso, geralmente um usuário ou cliente. Há um modelo simples: como um <tipo de usuário>, quero <algum objetivo> para que <algum motivo>. As histórias de usuários geralmente são escritas em cartões ou notas adesivas.

46. *World Café*

- **Modelo:** *Diving Board* [87].
- **Descrição:** O World-Café é um método de workshop, adequado para grupos de 12 a 2 mil participantes. É um processo de conversação estruturado destinado a facilitar a discussão aberta e íntima. Ele liga idéias dentro de um grupo maior para acessar "inteligência coletiva" dos participantes e para entender / aprender de múltiplos pontos de vista.

Todos os modelos estão consolidados e referenciados na Tabela 3.12.

Tabela 3.12 – Modelos de DT

Etapas	Modelos	Referências
2	DT-Macro Process University of St. Gallen DCIDT e CoDICE	[73] [98], [34], [19], [10], [2]
3	Carina González, Evelio González, Vanesa Cruz e José Saavedra HCD-Toolkit da IDEO Brown Souza e Silva	[42] [29], [14] [24], [88], [36], [102], [108], [105], [2] [39], [28]
4	IBM DT HCAW - Human-Centered Agile Workflow Dunne e Martin	[6], [78] [41] [2] [33]
5	Meinel e Leifer - (Hasso Plattner I HPI) DT-Micro Process University of St. Gallen (Adaptação da Stanford University) Stanford University's D-School	[58] [73] [23], [85], [97], [4], [107], [27], [70], [115], [93], [17], [86], [73], [83], [55], [50], [76], [81], [103], [16], [5], [43]
6	Stanford University's Hasso-Plattner Institute of Design d.school Driving Board Hasso-Plattner I D-School Model De paula e Araújo	[15] [87] [92], [80], [9], [51], [26], [31], [48], [100] [29]
7	Coutinho et al Sandino et al Hiremath e Sathiyam	[25] [101] [52]

FONTE: Autor, 2019

3.3.4 QP4: Ferramentas de apoio no DT para o desenvolvimento de software?

Pouco foi citado sobre o uso de ferramentas que auxiliam no desenvolvimento do DT em seus diversos modelos, métodos ou técnicas. Porém encontrou-se um pequeno conjunto de softwares descritos como ferramentas computacionadas usadas para o DT. Chasanidou et.al. [19] lista algumas ferramentas sem apresentar detalhes, e relaciona-se a técnicas que podem se beneficiar das mesmas, como *Personas* (Smapply⁸), *Stakeholder Map*

⁸<http://smapply.com>

(Circle8⁹), *Customer Journey* (Touchpoint dashboard⁹¹⁰), *Service blueprint* (creately¹⁰¹¹), *Business Model Innovation* (Strategyzer¹¹¹²) e *Rapid Prototype* (Axure RP¹²¹³).

O estudo de Avalos et.al [6] apresenta o uso do IBM *Bluemixtool* para a implementação dos protótipos. O baixo número de ferramentas encontradas mostra que existe uma área a ser explorada ainda, com foco em ferramentas que podem auxiliar no desenvolvimento e na prática do DT, sendo esta a oportunidade até da criação de uma ferramenta em específico. A literatura não aborda em profundidade as ferrametnas que suportam o DT em ES, nem mesmo como eles podem ser escolhidos para uso.

3.3.5 QP5: Seleção de uso dos Modelos, métodos, técnicas, abordagens e ferramentas no DT

Poucos artigos relataram o motivo da escolha dos métodos, modelos ou técnicas em determinados momentos e sendo respondido pela interpretação da escolha dos mesmos conforme leitura dos artigos. Acredita-se que pelo fato do DT mostrar-se ser bem definido sobre seus diferentes tipos de modelos e métodos, as técnicas, abordagens e ferramentas utilizadas já estão selecionadas visando atingir os objetivos de cada etapa de um modelo específico do DT [52] [3] [112] [28].

Souza e Silva [28] confirmam esta análise. Os autores [28] descrevem o DT para o processo de ER, propondo a execução de um processo do DT composto por um modelo iniciado pela etapa de imersão, com objetivo de tratar a informação disponibilizada pelos usuários para classificação e categorização. Etapa de ideação, para trabalhar as ideias mais referenciadas nas informações disponibilizadas pelos usuários e por final a etapa de prototipação. Para a primeira etapa, de imersão, procuraram encontrar técnicas que favorecessem o objetivo necessário ser alcançado. Neste caso o levantamento de ideias, análise e organização se utilizou-se técnicas como *brainstorming* e cartões de *insights* para obter a observação dos usuários sobre o contexto. Desta forma estas técnicas foram escolhidas visto que beneficiam a finalidade de imergir no universo do usuário, em busca de sua real necessidade a partir de um problema macro.

Levy e Hadar [73] apresentam no seu artigo a escolha da etapa da empatia, presente em muitos modelos do DT, da seguinte forma: a empatia é escolhida por ser de fato o primeiro passo de qualquer processo de DT e, portanto, uma boa escolha para iniciar e na segunda justificativa, tendo como base um contexto de requisitos de privacidade, acredita-

⁹<http://stakeholder-management.com>

¹⁰<http://touchpointdashboard.com>

¹¹<http://creately.com>

¹²<http://strategyzer.com>

¹³<http://axure.com>

se que neste domínio a empatia é muitas vezes ausente e poderia potencialmente fornecer benefícios significativos.

Outro exemplo também importante para fortalecer esta resposta encontra-se no trabalho de Braz et al. [78], onde o modelo do DT foi dividido em 4 fases: entender, explorar, prototipar e avaliar. Em resumo cita-se apenas a fase de avaliação do modelo do DT, onde abordaram a técnica de *feedbacks* sobre o protótipo criado, tendo como uma das técnicas também a abordagem da experiência do usuário (UX) com propósito de testes do protótipo, ou seja, as escolhas das técnicas nesta fase, definida como avaliação, frisaram abordagens que buscam alcançar o objetivo que deseja-se alcançar ao final desta etapa.

Os critérios para escolher os métodos estão em suas técnicas de visualização e capacidade de melhorar a comunicação dentro de equipes multiculturais, mas também na simplicidade em uso por não especialistas [19]. Logo, este conjunto de critérios são selecionados para uso de acordo com a necessidade que o contexto está oferecendo, pelo conhecimento da equipe que irá trabalhar com estas técnicas e pela forma que se deseja conceber as informações.

3.3.6 QP6: Recomendações e dificuldades do DT no desenvolvimento de software

O estudo revelou diversos pontos positivos e negativos para se aderir o DT no desenvolvimento de software. Ressalta-se o DT como um método que propõe ajudar na elucidação das necessidades do usuário em construção de requisitos focados sobre soluções viáveis e aderentes a realidade sobre os envolvidos, oferecendo um caminho benéfico tanto para equipe de desenvolvimento mas também, e principalmente, para a soluções que contemplarão os problemas dos usuários.

Em seus benefícios e boas práticas o DT vem com propósito de melhorar o entendimento das necessidades dos usuários, orientando a todos a reflexão e comunicação de todo o processo de desenvolvimento, de forma iterativa e dinâmica. Diversas técnicas como *Brainstorming* ou mapa de empatia, por exemplo, aderem o objetivo de compartilhamento de ideias de forma técnica e divertida, envolvendo desta forma a colaboração dos participantes [24].

De acordo com Rhinow e Meinel [99], no olhar de gerenciamento de projeto, o DT torna-se uma estrutura poderosa, porém limitada para capacitar o trabalho em equipe para desenvolver propostas promissoras de valor que melhorem continuamente os processos e estruturas.

O DT pode gerar muitas ideias inovadoras e de soluções viáveis para os problemas sem precisar escrever uma linha de código, nem modelos de *wireframes* ou *mockups*, sendo uma estratégia para aprender sobre os potenciais clientes e soluções futuras de TI

[15]. Sendo uma vantagem principal é que as necessidades estão no centro do processo, tornando-se na prática útil para medição de fatores gerenciais, como orçamento, qualidade e tempo, onde estes fatores deverão garantir o foco necessário em conjunto com ponto de vista do usuário e evitar conflitos de interesse, pois em algumas situações o arquiteto pode interferir na decisão a favor do usuário para favorecer uma solução simples [15].

Também, por se aproximar da abordagem ágil visando centradas ao ser humano, o DT traz benefícios como: organização de escopo de projeto, solução direcionada ao mercado em apoio no entendimento do problema com o cliente presente, não necessariamente controlado por gerentes, testes contínuos, melhoria rápida e flexível dos primeiros protótipos, trabalho com equipes multifuncionais (*designers*: perspectiva humana, desenvolvedores: perspectiva da tecnologia e *marketing/vendas*: perspectiva comercial), aumentando assim a sincronização das pessoas envolvidas no projeto, trabalhando de forma coordenada e eliminando aspectos como a transição de um nível de desenvolvimento entre áreas [14]. A Tabela 3.13 mostra pontos positivos e negativos de acordo com os estudos encontrados no mapeamento sistemático.

Tabela 3.13 – Benefícios e Desafios/Limitações do DT

Benefícios	Referência
Ajuda na criação de MVP	[23], [93], [10]
Projetos/Ideias/Soluções Inovadores e criativas	[23], [24], [86], [9], [72], [83], [29], [55]
Elucidar Requisitos	[87], [39], [51], [55], [28]
+ Eficaz para projetos de pequeno e médio porte	[27]
Integração com métodos ágeis	[27], [86], [10], [105], [29], [55], [100], [81]
Entender melhor a necessidade do cliente e aproximá-lo	[24], [39], [93], [101], [72], [55], [76], [100]
Melhora a comunicação	[24], [55]
Compartilhamento de ideias	[24]
Conhecer o real problema	[39]
Reduz o risco do esforço para o desenvolvimento	[86]
Promove soluções sustentável	[80]
Apto a mudanças frequentes	[72]
Validação do produto ou experiência do usuário	[105], [29]
Desafios/Limitações	Referência
Difícil escalar para programas e sistemas maiores	[27]
Tempo integral com todos participantes	[27], [52]
Disponibilidade do Cliente	[27], [52], [28]
Necessidade de Tempo	[73], [93], [51], [105], [48]
Integrar com métodos ágeis	[29]
Necessidade de Espaço	[73], [52]
Falta de empatia pode causar dificuldades no entendimento do problema	[24]
Interação em equipe no início	[24], [52]
Pensar de forma criativa	[24]
Uso e aplicação inicial das técnicas	[67]
Limitada	[99]
Falta de rastreabilidade	[51]
Falta de requisitos de contexto	[51]
Motivação, mudança de mindset dos participantes	[51], [52], [78]
Falta de suporte para conhecimento implícito da equipe	[51]
Melhor aplicado quando existe um cenário definido de produto de software	[52]
Necessidade de equipes multidisciplinares	[78]
Integrar com métodos ágeis	[29]

4. SURVEY

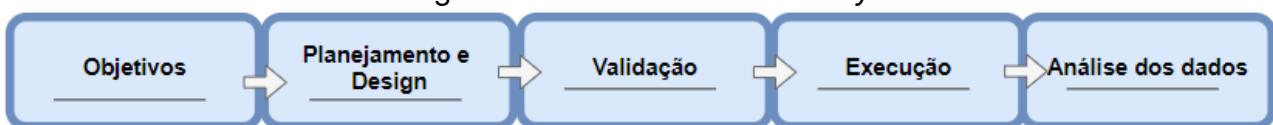
O MSL trouxe diversas formas de se trabalhar com o DT em diversos contextos, ou seja, o DT como relatado na literatura apresenta diversos modelos, diversas técnicas e que estão sendo utilizados em diversas situações mas não trouxe com exatidão o por que usá-las. Nos artigos aceitos, não se falava em: Por que usar este modelo? Por que usar esta técnica? No momento em que existem diversas formas de se aplicar, diversas técnicas para serem usadas, não se consegue definir e formar um único modelo e técnica para uma situação em específico. Talvez o DT possui diversas formas de aplicação e todas são benéficas dentro de cada cenário inserido. O mapeamento sistemático sozinho trouxe um leque de informações muito importantes sobre o DT no desenvolvimento de software, entendendo formas possíveis de se usar com exemplos descritos, mas surgiram algumas lacunas sobre o entendimento do DT. Estas lacunas são apresentadas neste Capítulo através da Figura 4.2.

Dessa forma, com o conhecimento adquirido no trabalho do mapeamento sistemático, destacado no Capítulo 3, procura-se relacionar os resultados anteriores com o entendimento do DT no desenvolvimento por profissionais que atuam na indústria e possuem experiência para se posicionar sobre este assunto. Para que isso ocorra, planejou-se uma *survey* para suprir a necessidade de preencher questões não tão claras trazidas na literatura.

4.1 Survey segundo Kitchenham e Pfleeger

A *Survey* não é apenas um questionário e sim um sistema abrangente de coleta informações para descrever, comparar ou explicar conhecimentos, atitudes e comportamentos, servindo como uma parte de um processo maior, conforme mostra a Figura 4.1 [91].

Figura 4.1 – Processo da *Survey*



FONTE: Kitchenham e Pfleeger [91]

A seguir define-se cada etapa do processo da *survey*:

1. **Estabelecer e definir os Objetivo específicos:** Defini-se os objetivos de acordo com a necessidade percebida após análise dos dados do mapeamento sistemático.

Entende-se que é essencial a dedicação de tempo para analisar os objetivos ao ponto de torná-los claros e mensuráveis, visto que estes são importantes para todas as atividades subsequentes do processo de pesquisa, onde primeiro se determina o que a pesquisa perguntará, qual será sua população e quais informações serão coletadas [91];

2. **Planejamento e Design:** Projetar o estudo e preparar o instrumento de coleta, analisando o processo de criação do questionário. Inicia-se pelo processo pela revisão dos objetivos, examinar o público-alvo e decidir qual a melhor forma de obter as informações necessárias. Além de considerar o fator do tamanho apropriado da amostra e garantir maior taxa de respostas possíveis [64];
3. **Processo de validação do instrumento de coleta:** Realização da validação do instrumento de coleta antes da liberação do mesmo. Conhecer resultados de pesquisas anteriores para reduzir as despesas gerais da construção da pesquisa [64]. Além de também procurar evitar perguntas tendenciosas no questionário e saber motivar as pessoas a completar todo o instrumento [61];
4. **Executar e administrar a coleta:** Processo de executar e administrar a coleta da *survey* durante o período que o instrumento estiver em campo liberado para coletar respostas, para se obter um conjunto de respostas válidas da amostra [62];
5. **Analisar e reportar os dados obtidos:** Verificar e gerenciar os dados fornecidos pela pesquisa, através de técnicas corretas, verificando a integridade e consistência das respostas [63].

A estrutura a seguir está representada na Figura 4.1 na qual ilustra o processo de desenvolvimento da *survey* sobre o entendimento do DT no desenvolvimento de software através de um questionário aplicado na indústria, dada uma amostra classificada por profissionais atuantes na área da Engenharia de Software ou que já trabalharam em projetos similares.

4.2 Planejamento e Realização da *Survey*

Objetivo. O objetivo da *survey* é suprir lacunas encontradas no mapeamento sistemático a fim de entender como profissionais, que utilizam DT para o desenvolvimento de software, estão atuando. Dessa forma identifica-se quais modelos e técnicas estão sendo utilizadas e se existe alguma dificuldade na escolha das técnicas sobre as fases dos modelos. A execução desta *survey* fortalecerá o entendimento do DT na Engenharia de Software e ajudará na caracterização deste fenômeno.

Baseado nos resultados do Mapeamento da literatura, entende-se que o DT está sendo utilizado de diversas formas, diferentes modelos, técnicas e ferramentas, mas em nenhum momento foi trabalhado o motivo das escolhas. As questões que a *survey* traz estão diretamente relacionado com o mapeamento do Capítulo 3 para prover um melhor entendimento e assim maior embasamento.

Planejamento e Design. A *survey* desenvolvida nesse trabalho está caracterizada da seguinte forma [91], [56]:

- Explicativa: Buscar o entendimento do fenômeno através das informações coletadas;
- Semi-supervisionada: Onde o pesquisador apresenta a pesquisa e seu objetivo, ficando disposto para esclarecimento de dúvidas mas permite que os participantes respondam por si próprios.

Como já apresentado na Figura 4.2, o design da *survey* esmiuçar as respostas trazidas no mapeamento, formando assim um novo conjunto de questões que nos forneça um melhor entendimento do fenômeno. Importante ressaltar não apenas a construção de cada questão mas também a forma que será apresentada através de um questionário online. As questões foram planejadas para ser claras e atacar diretamente o ponto necessário, ou seja, questões mais pessoais, que geralmente são colocadas no início (por exemplo, sobre tempo de experiência), foram colocadas por último, visando que mesmo que o respondente acabe desistindo na metade da pesquisa, as questões principais já foram respondidas.

A Figura 4.3 mostra o trajeto desenvolvido do questionário, focando como prioridade as questões mais importantes, deixando para o final as que não serão tão prejudiciais caso não respondidas. O questionário encontra-se no Apêndice B.

Pensou-se em contextualizar antes de começar perguntando a dificuldade de escolha ou como são escolhidas os modelos e técnicas, ou seja, iniciou-se mostrando um conjunto de modelos e técnicas de DT que foram encontradas no MSL, apresentando o que a pesquisa está dizendo que é modelo e o que são técnicas de DT. Dessa forma o respondente já estará se beneficiando caso não conheça algum modelo ou técnica, pois acabará se deparando com diferentes modelos e técnicas e pode despertar mais interesse. Posterior a isso, conforme mostra a Figura 4.3, começa o filtro em tentar entender o quão difícil é escolher uma técnica para uma determinada situação e através de qual forma as mesmas são escolhidas. Com esse conjunto de respostas dos profissionais, consegue-se atender as lacunas encontradas nas questões QP3, QP4 e QP5, de acordo com a Figura 4.1.

Em seguida destacam-se as lacunas encontradas na QP1, QP2 e QP6, procurando entender qual o verdadeiro propósito de se usar todo esse conjunto de modelos e técnicas de DT, convergindo as ideias e propósitos já trazidos na literatura com as respostas elaboradas pelos profissionais. Ao final, não tão importante mas de suma peculiaridade,

Figura 4.2 – Lacunas do MSL x Survey

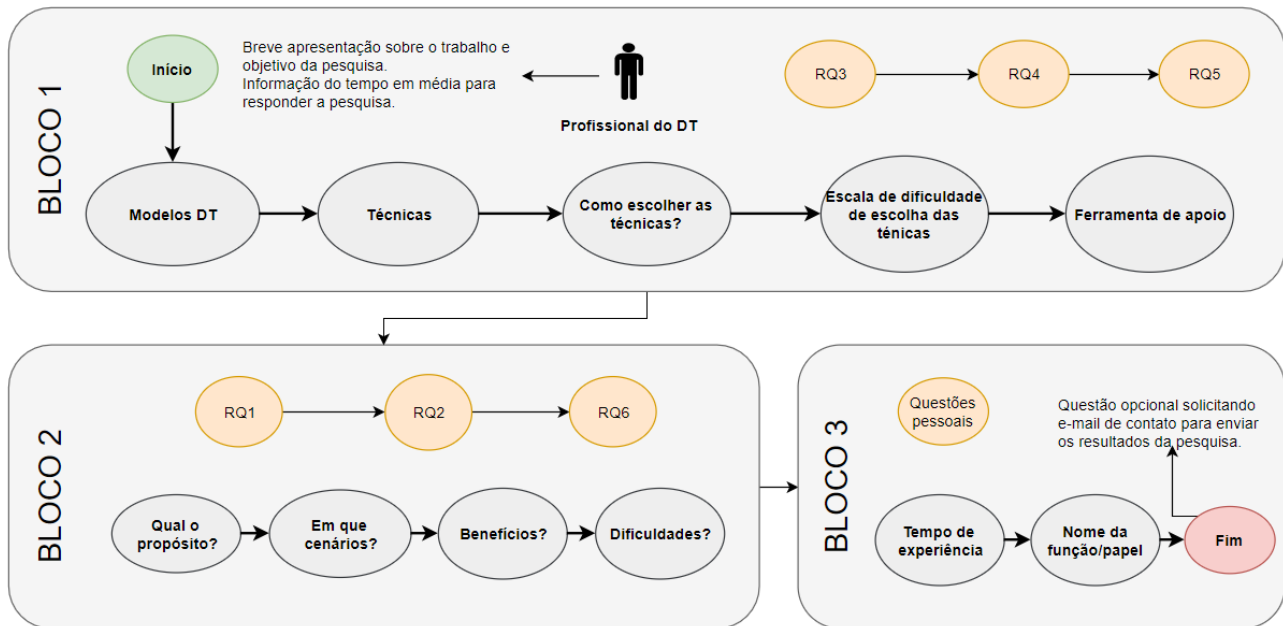


FONTE: Autor, 2019

entender quais são as dificuldades e os benefícios que estes profissionais estão encontrando ao tentar aplicar o DT no desenvolvimento de software, seguido pelo último bloco por questões pessoais, para identificar melhor quem é este profissional e quanto tempo de experiência ele possui, dessa forma identificando o perfil júnior ou sênior.

Com o público-alvo, procurou-se profissionais que trabalham com o DT no processo de desenvolvimento de software, podendo ter sido usado no início do processo ou no final, sendo o importante já ter trabalhado ao ponto de fornecer um conjunto de respostas

Figura 4.3 – Estrutura do Questionário



FONTE: Autor, 2019

com viés sobre o fenômeno desta pesquisa. A busca foi focada na distribuição do questionário através da rede de negócios conhecida como *LinkedIn Corporation*¹. Os respondentes foram encontrados de duas formas:

- Amostra inicial: Todos os palestrantes das trilhas de UX e DT do evento *The Developers Conference*, conhecido como TDC, desde 2005 até 2019;
- Amostra definitiva: Todos os profissionais retornados pelo filtro do *LinkedIn* de acordo com as strings: "*design thinking*" and "*software*", "*design thinker*" and "*software*" e "*design thinking*". Todos foram filtrados pela nacionalidade brasileira.

O planejamento sobre análise dos resultados é focar através de uma análise qualitativa sobre os dados trazidos nas visões dos profissionais de DT, buscando uma análise descritiva a fim de sintetizar através dos resultados tabulados e em gráficos como o DT está sendo usado pelo profissional. Durante a interpretação das respostas, realizou-se uma meta-análise sobre os dados do mapeamento e desta *survey* a fim de entender se estes dois estudos convergiram para mesma ideia, para o mesmo propósito. A meta-análise acaba-se sendo realizada indiretamente, visto que os dados construídos no questionário vieram da pesquisa do mapeamento, ou seja, a consolidação por si só já está inferida no próprio resultado da *survey*. Este estudo não realiza cálculos estatísticos e não possui um viés quantitativo, apenas sintetiza através dos resultados uma análise descritiva sobre os dados

¹<https://br.linkedin.com>

ao ponto de fornecer o entendimento deste fenômeno na indústria, ou seja, a preocupação em obter-se resultados qualificados e suficientes para traduzir o DT no desenvolvimento de software pelos profissionais é mais importante nesta pesquisa do que coletar os dados em grande escala e quantidade. Posteriormente a isso os dados foram tabulados e mostrados em gráficos, agregando os resultados entre as questões, provendo melhor entendimento sobre o assunto e se as respostas estão de acordo com o objetivo desta pesquisa [60].

Com base na estrutura e planejamento da *survey*, citado acima, desenvolveu um questionário que passou por *sprints* de validação dentro do grupo de pesquisa antes de ser liberado para distribuição para os selecionados.

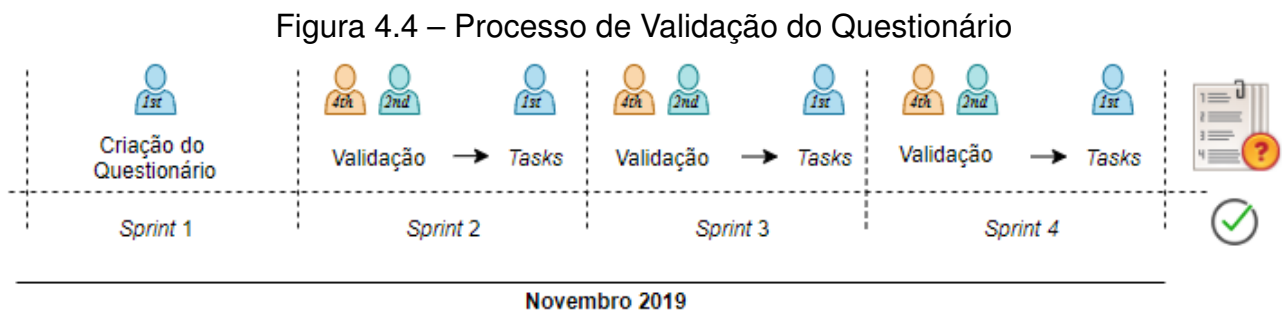
Validação. Nesta seção descreve-se o processo de validação do questionário. O processo de validação foi realizado pelos mesmos pesquisadores do MSL, com exceção do terceiro pesquisador, que não participou. Antes do processo de validação o pesquisador 1 realizou um trabalho de embasamento sobre aplicação de *survey* na área da computação. Acaba-se que, devido ao planejamento realizado dentro do escopo e buscando atender o cronograma, é difícil encontrar um espaço de tempo para validação do instrumento por completo, acabando muitas vezes não sendo melhor desenvolvido. De acordo com Kitchenham e Pfleeger [65], raramente instrumentos são validados e por esse motivo é de suma importância a pesquisa sobre como criar um instrumento de coleta e quais cuidados que devem ser tomados para projetar uma *survey*.

Antes da distribuição do questionário para validação pelos pesquisadores 2 e 4, foi realizada um breve levantamento na literatura sobre aplicação de *survey* no âmbito da computação para melhor criar um questionário e fugir de erros comuns. Com isso resume-se alguns cuidados [77]:

- Identificação do público-alvo: De acordo com Kitchenham et al [66], não saber caracterizar a população da qual as amostras fazem parte interfere sobre os resultados que podem ser obtidos;
- Usar uma linguagem simples e evitar termos técnicos;
- Evitar questões muito exigentes e realizar questões curtas;
- A ordem das perguntas podem afetar o resultado do estudo (afetar a taxa de respostas).

Os critérios acima podem parecer fáceis de inserir em um questionário, mas muitas vezes o vício que o pesquisador já está sobre sua pesquisa acaba sobrepondo estas questões simples, onde para o pesquisador tudo já está claro mas para quem for ler a primeira vez é necessário total clareza do objetivo de suas questões e uma linguagem totalmente simples, não deixando dúvidas para o respondente. O processo de validação mostrou algu-

mas armadilhas que o pesquisador não enxergava mais, mas que foram encontrados pelos outros pesquisadores. O processo de validação foi realizado conforme mostra a Figura 4.4.



FONTE: Autor, 2019

Cada *sprint* é representando por uma semana do mês de novembro. A primeira *sprint* o pesquisador 1 realizou a criação do questionário pela plataforma QuatricsXM, em sua versão licenciada para a PUCRS². As questões foram realizadas seguindo a ordem dos tópicos abaixo:

- Texto formal de apresentação sobre o trabalho e seu objetivo;
- Questões pessoais: Tempo de experiência e nome da profissão;
- Questões sobre modelos, técnicas e ferramentas;
- Questões sobre o contexto, cenário e benefícios/dificuldades.

A primeira criação do questionário é um pouco diferente sobre os blocos separados e mostrados na Figura 4.3, onde as questões pessoais foram colocadas em primeiro lugar, pois não se tinha ideia em colocar as questões importantes em primeiro e sim seguir sempre a lógica: conhecer o respondente e apresentar as questões específicas sobre assunto. Na primeira entrega para validação, retornaram as seguintes *tasks*:

- Arrumar texto inicial, mostrando mais clareza sobre objetivo e apresentação do trabalho;
- Exibir barra de progresso, isso mostrará ao respondente o tempo que falta para terminar de responder o questionários;
- As questões que possuem a opção "OUTROS", inserir caixa de texto para o respondente indicar na própria resposta;

²<https://pucrs.qualtrics.com>

- Alterar forma de apresentação da questão das técnicas, visto que são mais de 40 técnicas e pode ficar cansativo para o respondente.

Após as correções solicitadas na segunda *sprint* foi novamente enviado para um novo processo de validação. A questão das técnicas foi a que mais aumentou o tempo estimado para responder o questionário e mais trabalhada na segunda *sprint*, onde foi novamente pensada para não cansar o respondente, visto que é uma das mais importantes para ser respondida. A terceira *sprint* apresentou as seguintes *tasks*:

- Questão sobre indicar nível de dificuldade deve transformar em uma questão que mostra uma barra de 1 a 10, indicando possível nível de dificuldades. Deixar claro o que representa o nível 1 e o nível 10 e criar a possibilidade de uma resposta rápida, para não perder o tempo do respondente;
- Verificar se o questionário está acessível para dispositivos móveis e para pessoas com necessidades especiais;
- Escrever em todas as questões com clareza para mostrar se o respondente pode selecionar apenas 1 resposta ou mais de uma, não deixar dúvidas para os respondentes;

As correções foram realizadas conforme levantamentos dos pesquisadores 2 e 4. A maioria das questões possuem a estrutura de várias respostas, ou seja, formato *checkbox*. Entretanto, mesmo sabendo a diferença que *checkbox* indica mais de uma resposta e *radiobox* solicita apenas uma escolha, foi colocado ao final de todas as questões a informação: Marque todas opções que se aplicar, não causando dúvidas para quem for responder.

Ao final, na última *sprint* foi realizado mais um trabalho em conjunto entre todos os pesquisadores para finalizar a criação do instrumento e iniciar a distribuição na primeira semana de dezembro de 2019. O texto inicial foi novamente alterado para não ser tão formal e mostrar em poucas palavras e com clareza o objetivo do trabalho, mostrando o tempo médio que o respondente levará para responder. Também a ordem das questões foram alteradas, colocando as questões pessoais ao final da pesquisa, visto que a desistência no meio do caminho pelo respondente irá afetar menos se não respondido as questões de tempo de experiência ou qual o nome da profissão relacionada a este tipo de profissional que utiliza DT no desenvolvimento de software. A ordem final e correlacionada com as questões do mapeamento sistemático já foram apresentadas na Figura 4.3.

O processo de validação ajudou a eliminar algumas armadilhas que, por mais óbvias sejam, acabam não sendo vistas pelo próprio pesquisador que desenvolveu o instrumento, como: padronizar, indicar sempre na própria questão o tipo de marcação que o respondente deve fazer, apresentar sempre o tempo médio que o respondente levará para o mesmo poder se organizar e responder em seu tempo. Também padronizar o tipo das questões e respostas, oferecendo sempre o mesmo estilo de perguntas e respostas, como

por exemplo, as respostas que possuem a opção (outros?), a mesma deve ser escrita igual em todas questões, padronizando a estrutura do questionário. Apresentação da barra de progresso para o respondente ter uma ideia se falta muito para terminar e no início um texto curto, claro, objetivo e que não canse os olhos de quem irá começar a responder. Sempre se tem a impressão que é necessário inserir todas informações, mas isso nem sempre ajudará, pois provavelmente o respondente irá realizar uma leitura rápida sobre o texto inicial, então é importante resumir o máximo possível o que o respondente precisa realmente saber.

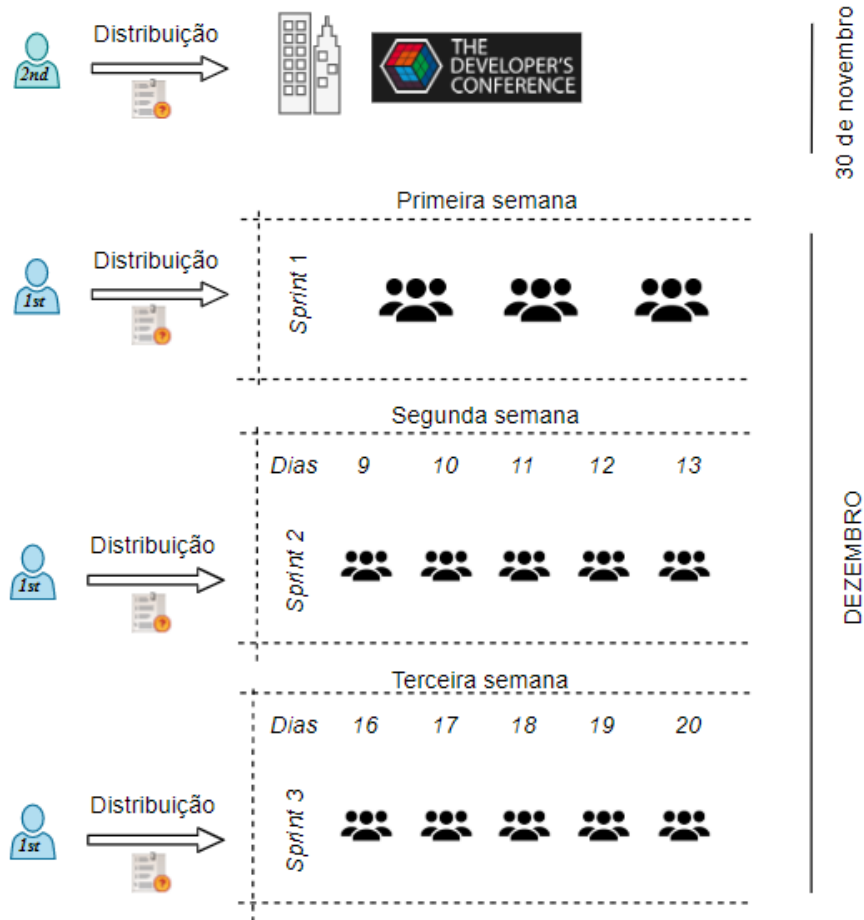
Durante o processo de execução, à amostra inicial trouxe *feedbacks* positivos da pesquisa, mostrando que a mesma está bem estruturada e a pesquisa é interessante. O processo de execução e resultados coletados encontram-se a seguir.

4.3 Coleta e Análise dos Resultados

Execução. A execução iniciou no final de semana que ocorreu o evento *The Developer's Conference* (TDC), em Porto Alegre. O pesquisador 2 participou da trilha de DT no evento e obteve 11 respostas no dia. Esse foi o único momento na qual o questionário foi distribuído de forma presencial. Na semana que precede o evento, iniciou-se a coleta *online*, buscando através da rede social LinkedIn, profissionais que se enquadram no perfil para responder esta *survey*. A Figura 4.5 mostra a estratégia realizada para distribuição do questionário *online* para os respondentes. No início pensou-se em distribuir no início da semana, conforme mostra a *sprint* 1 da Figura 4.5, para uma parte dos respondentes para iniciar o processo de coleta. Entretanto, visando que esta é uma pesquisa semi-supervisionada, enviar para todos e estar a frente para retirar dúvidas iria ser uma demanda trabalhosa e sem necessidade, visto que talvez o pesquisador 1 não conseguisse dar conta de atender a todos caso a pesquisa fosse distribuída por completa para todos. Na primeira *sprint* foi distribuído para 61 pessoas. Infelizmente obteve-se apenas 14 respostas e poucos retornaram com *feedback* sobre a pesquisa, ou seja, 22,95% da amostra distribuída respondeu a pesquisa, uma taxa muito baixa. Com isso estudou uma outra forma de distribuição e aplicou a nova estratégia na segunda semana.

A segunda semana a distribuição passou a ser diária. Foi enviado todo dia de manhã, na primeira hora, para aproximadamente 10 pessoas. Com essa estratégia o pesquisador conseguiu ter tempo para responder quaisquer dúvida que poderia surgir durante a parte da manhã e reforçar, para aqueles que não tinham lido ainda a mensagem, o convite para responder a *survey* durante a tarde. Dessa forma tinha um controle maior da distribuição e uma aproximação com os respondentes.

Figura 4.5 – Processo de Distribuição

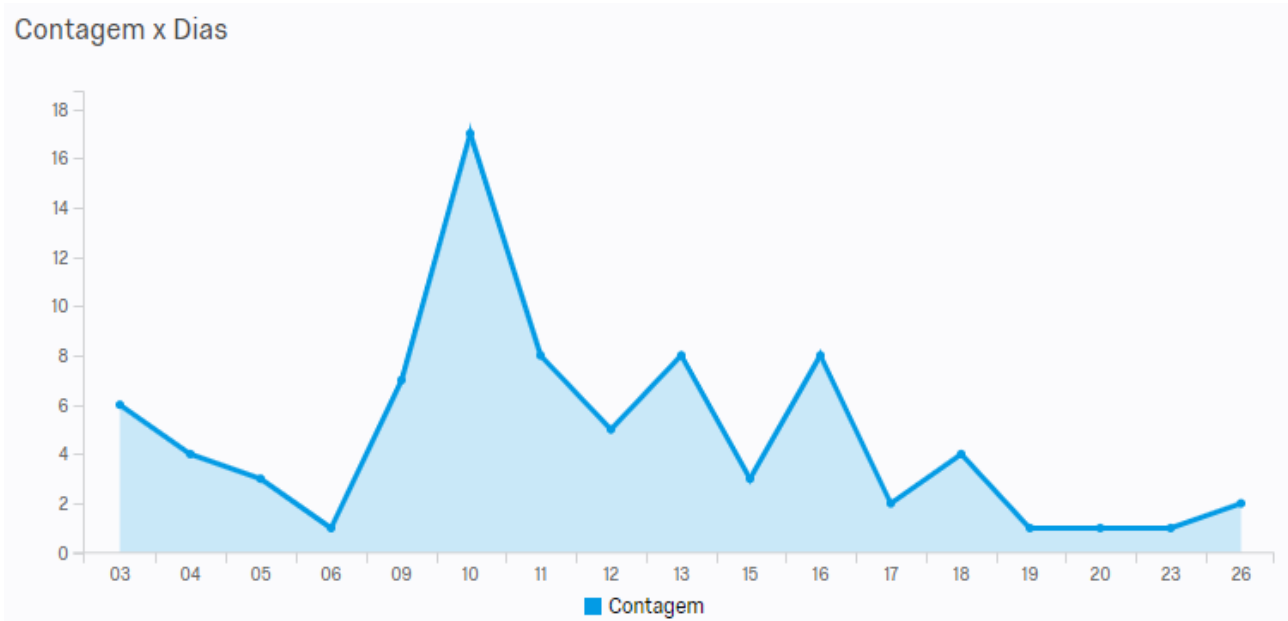


FONTE: Autor, 2019

Essa forma de distribuir teve mais sucesso que distribuir para todos inicialmente. O dia 09/12 teve 7 respostas e com um aumento significativo no segundo dia da *sprint 2*, onde no dia 10 obteve-se 17 respostas. Com o retorno positivo do primeiro dia, surgiu a ideia de aumentar no segundo dia a quantidade de pessoas para convidar a responder a *survey*. Foi um sucesso mas mais trabalhoso, pois além de enviar para mais de 20 pessoas no segundo dia, tinha que estar a disposição para retirar dúvidas e reforçar o convite para aqueles que não tinham respondido ainda. Porém, por mais que seja trabalhoso, foi uma estratégia de distribuição que funcionou e foi adotado pelo pesquisador até o final da *sprint 3*, finalizando no dia 20/12, a semana que antecede a recesso de natal e final de ano.

A Figura 4.6 apresenta o número de respostas por dia dentro das semanas das *sprints* conforme Figura 4.5. Ao final, dentro das 3 *sprints* mostradas na Figura 4.5, foram enviadas mensagens para 176 pessoas no *LinkedIn* e obteve-se 81 respostas 46,02%. Das 81 respostas registradas pela plataforma Qualtrics, 14 apenas entraram no questionário mas não preencheram, fechando aproximadamente 67 respostas válidas. As 67 respostas foram somadas com as 11 respostas obtidas no evento TDC, totalizando 78 respostas completas com conteúdo relevante para serem analisados. Dado o tempo curto de 1 mês

Figura 4.6 – Contagem x Dias de Dezembro



FONTE: Autor, 2019

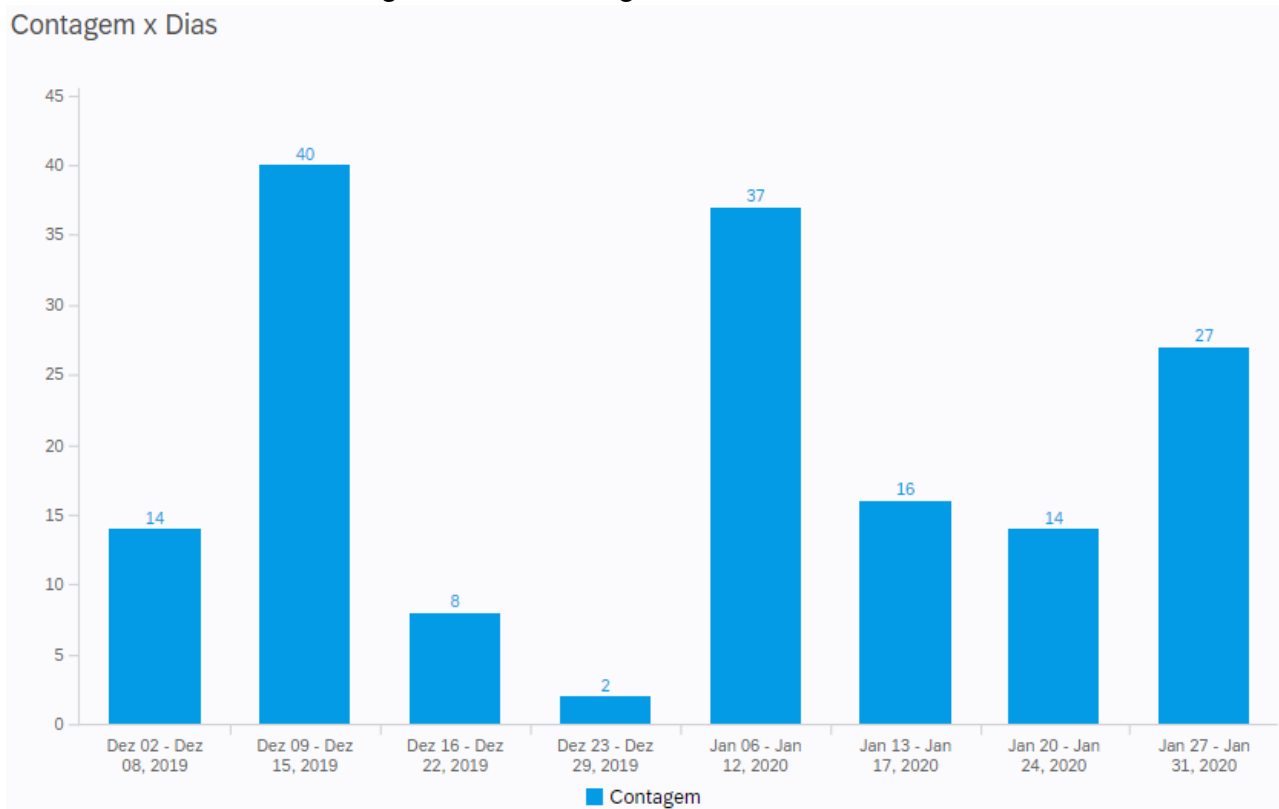
para distribuir e estar a frente para retirar dúvida, esperava-se que no mínimo 30 pessoas respondessem ao questionário apresentado no Apêndice B, com isso pensou-se em enviar o questionário para mais de 150 pessoas, pois se considerar uma margem de erro de 20% sobre 150 pessoas que receberão o questionário, alcançaria à amostra de 30 respostas. Felizmente e por prevenção, foi enviado para 176 pessoas que resultou mais que o dobro do esperado.

Percebe-se na Figura 4.6 que os finais de semana não foram trabalhados e conforme aproxima do período de recesso do natal e ano novo as respostas foram diminuindo. Visto a diminuição devido ao natal e recesso, planejou-se uma nova etapa de coleta em Janeiro de 2020, visando alcançar um aumento significativo e representativo para agregar valor para esta pesquisa. A Figura 4.7 mostra a evolução de dezembro/2019 para Janeiro/2020, após recesso.

Percebe-se que, no período de recesso do natal e ano novo, houve uma diminuição drástica de 40 respostas para 8 e 2. Com isso percebe-se que perdeu-se a oportunidade de alcançar um número muito maior visto que na segunda semana de dezembro foram coletadas 40 respostas. Com isso, realizou-se a segunda coleta dos dados no mês de janeiro e a Figura 4.7 mostra o aumento significativo da quantidade de resposta após as semanas de natal e ano novo. Esses dados foram incluídos e validados após o dia 31/01/2020 e obteve-se no mês de janeiro 94 respostas válidas.

Seguindo a mesma logística realizada no mês de dezembro/2019, foi enviado no mês de janeiro/2020 para 290 pessoas e considerou-se que apenas 20% das pessoas

Figura 4.7 – Contagem x Dias até Janeiro



FONTE: Autor, 2020

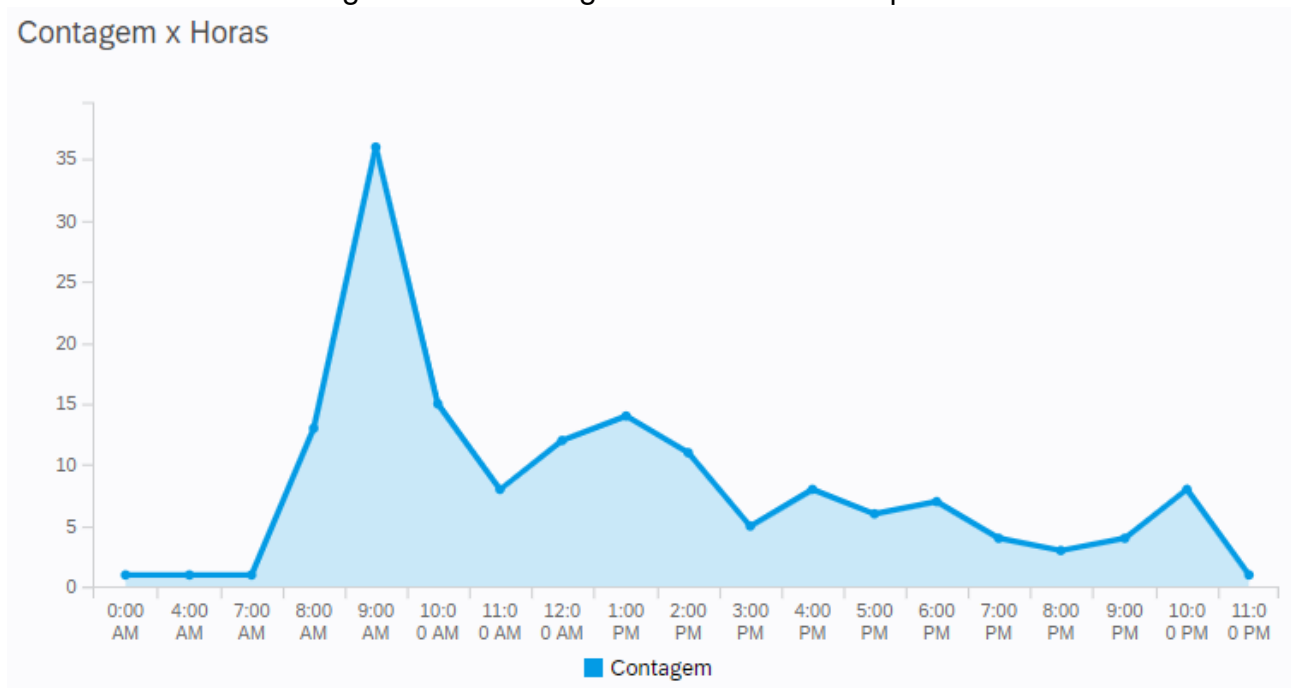
responderiam, ou seja, 58 profissionais. Felizmente novamente quase se chegou ao dobro do que esperava-se. Foi verificado o motivo dos outros que receberam o convite não terem respondido a pesquisa e existem duas justificativas: os respondentes não acessam muito a rede *LinkedIn*, visto que muitos não responderam e não leram a mensagem enviada e o outro motivo foi o retorno do próprio respondente que informou que trabalha com DT mas nunca realizou atividades com DT sobre o contexto da Engenharia de Software, ou seja, não possui o perfil da pesquisa. Em resumo, foi enviado para 466 pessoas no *LinkedIn*, enviando o convite individualmente para cada uma, obtendo-se ao final 158 respostas completas no questionário, ou seja, 33,90% responderam a *survey*.

Foi interessante a experiência de ter visto o retorno daqueles que se preocuparam em ler e verificar se estão dentro da amostra, pois pelo perfil inicial do *LinkedIn*, acreditava-se que os profissionais que negaram ter trabalhado tinham pelo menos participado de um trabalho com desenvolvimento de software. Interessante também foi o retorno de um profissional com perfil de UX, na qual ofereceu ajuda para distribuir para uns colegas que também trabalham na área, conforme descreve o profissional: "*Mandei para UXers pois são as pessoas que valorizam o processo de pesquisa. UX depende de pesquisa para trabalhar então é mais fácil conseguir engajamento.*". Esse retorno foi importante para esta pesquisa, visto

que era um dado desconhecido antes da execução dessa pesquisa e que pode vir ajudar futuras pesquisas sobre esta área.

Outra observação a ser feita foi a forma de envio para os profissionais. Durante a semana, antes das 08:00 horas era enviado o questionário para um conjunto de pessoas encontradas no filtro de pesquisa do *LinkedIn*, convidando-os para responder o questionário, esperando que muitas estão acordando e vendo as notícias na plataforma antes de iniciar o dia de trabalho. Também foi enviado após horário do almoço, imaginando-se que as pessoas estavam retornando do trabalho e poderiam parar para ver as notícias na plataforma e dessa forma aproveitariam o tempo e responderiam o questionário. Percebeu-se, conforme mostra a Figura 4.8 que a taxa de respostas ocorria sempre no horário da manhã e que a tarde a quantidade diminuía. Com a experiência obtida durante execução desta *survey* esses dados são relevantes para futuras pesquisas que desejam executar sua *survey* e ter controle sobre a distribuição de seus instrumentos de coleta.

Figura 4.8 – Contagem x Horário de Respostas



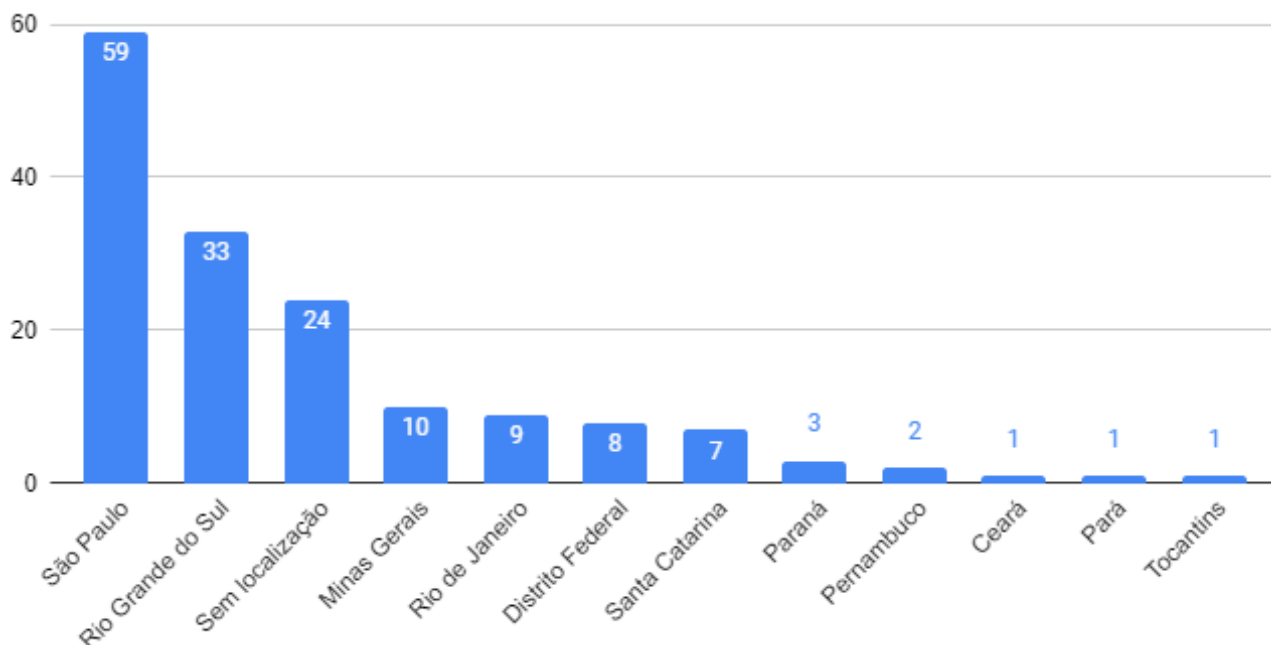
FONTE: Autor, 2020

Após conhecer a forma de execução realizada nesse trabalho, apresentam-se na próxima seção os resultados finais sobre o total de 158 respostas obtidas nessa *survey*, sendo 64 respostas no mês de dezembro/2019 e 94 no mês de janeiro/2020.

Também, através da ferramenta QuattricsXM, que disponibiliza o IP do respondente para aqueles que finalizaram por completo a pesquisa, foi possível verificar a região de cada respondente. A amostra representa diferentes áreas do território brasileiro. O gráfico apresentado na Figura 4.9 mostra a quantidade de respostas por região. A coleta da região por

IP foi realizada individualmente, colocando o IP do respondente na ferramenta WEB³ que tem como base a tecnologia de geolocalização ou geoip, rastreando um IP e identificando sua localização geográfica [54]. Infelizmente a ferramenta não apresentou a localização de 24 respondentes, representados da forma no gráfico com o *label* Sem localização.

Figura 4.9 – Respondentes por Região



FONTE: Autor, 2020

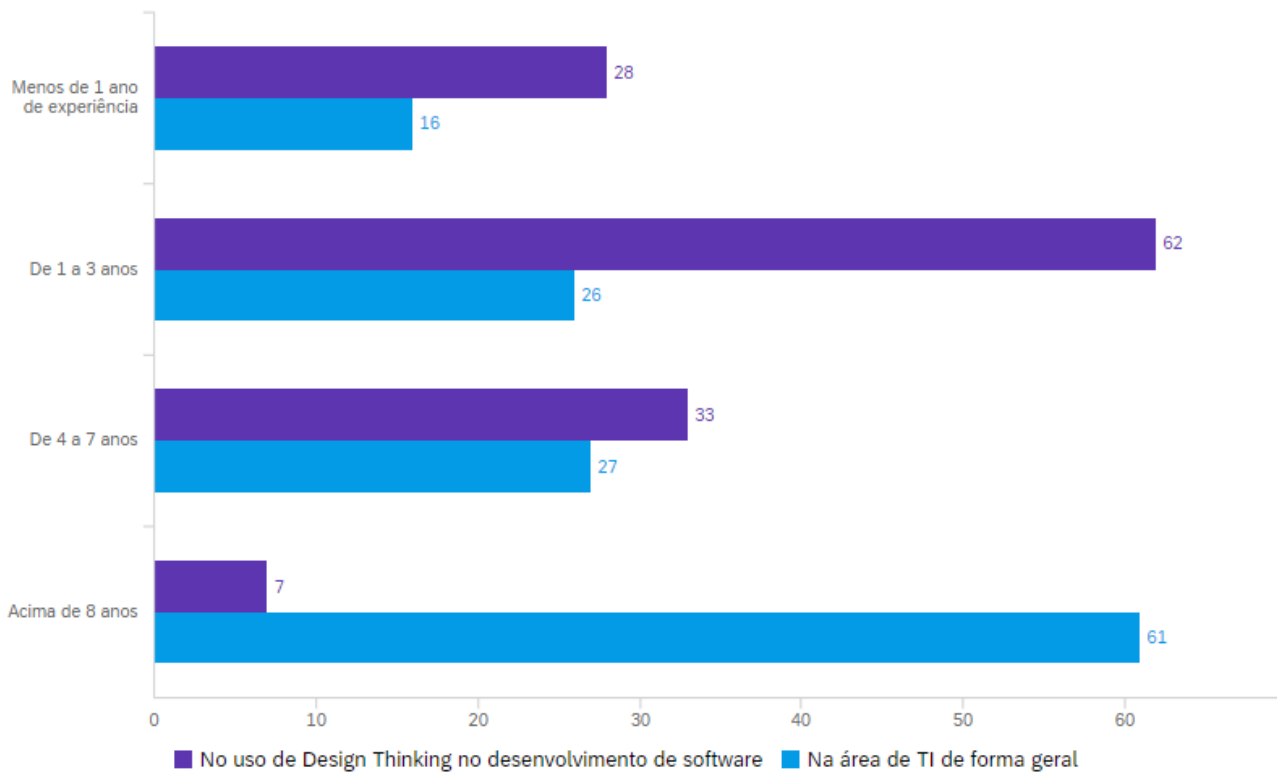
Acredita-se que, se a coleta não fosse perto de um período de recesso, teria um aumento significativo na coleta sem precisar planejar o mês de janeiro. Entretanto, janeiro ainda era um tempo hábil para coleta das informações e poderia ser utilizado dentro do escopo de trabalho para a realização da distribuição e coleta dos dados. Os dados obtidos em janeiro foram relevantes para a representatividade dos dados sobre o fenômeno que está sendo estudado nesse trabalho e é em benefício a comunidade do DT da indústria que pensou-se na qualificação desses resultados. Abaixo mostra a última etapa da *survey* representada pela análise de dados, seguido por outra seção que mostra a discussão dos dados de acordo com cada pergunta realizada no questionário.

Análise dos dados. Após conclusão do processo de execução, com objetivo de encontrar na indústria como o DT está sendo usado no desenvolvimento de software, destaca-se nessa seção o resultado obtido da *survey*. Antes de realizar análise sobre os dados, cabe uma breve descrição do público que respondeu o questionário. Esta descrição foi obtida através das questões pessoais presentes no bloco 3, conforme mostra Figura 4.3. A Figura 4.10 mostra a quantidade de respostas de acordo com a experiência do respondente.

³Site: <https://www.meuenderecoip.com/localizar-ip.php>

Percebe-se que o público possui grande parte da experiência com DT entre 1 a 3 anos e muitos possuem acima de 8 anos de experiência na área de TI de forma geral. Percebe-se uma correlação com a Figura 3.4, onde houve um crescimento na literatura nos anos de 2016 até 2018 sobre o tema DT na área de desenvolvimento de software.

Figura 4.10 – Experiência em Anos Sobre os Profissionais



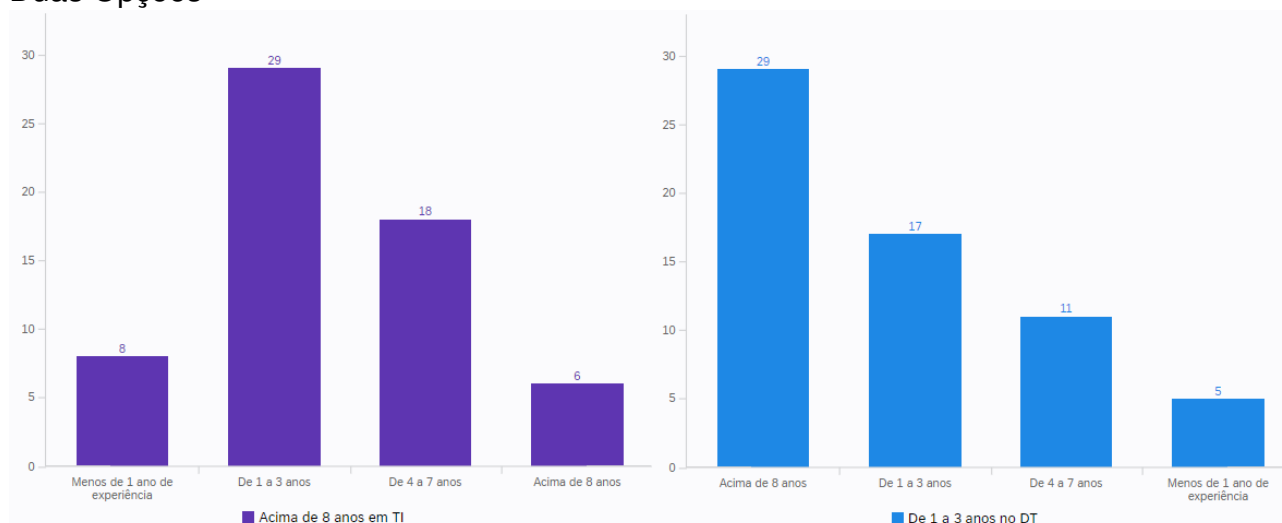
FONTE: Autor, 2020

O público que respondeu esta pesquisa e que possui experiência acima de 8 anos (2012 a 2019) na área de TI de forma geral, também possuem experiência com DT entre 1 a 3 anos ou entre 4 a 7 anos, conforme mostra o lado esquerdo da Figura 4.11. No lado direito da Figura 4.11 apresenta a visão contrária, onde a maior seleção de profissionais que atuam com DT entre 1 a 3 anos, também atuam acima de 8 anos em TI ou em conjunto no mesmo tempo de experiência com DT. Considera-se um público com experiência e viés para tratar sobre este assunto.

A literatura mesmo trouxe um aumento do DT nesta área nos últimos 3 anos e este resultado ficou de acordo com o esperado. Houve uma porcentagem representativa de 25% do público que possui menos de 1 ano de experiência com DT, podendo assim serem considerados mais juniores neste requisito, mas também vai de encontro com o crescimento encontrado no mapeamento sistemático nos últimos anos sobre este tema.

Sobre os dados obtidos dos profissionais que encontraram o seu papel nas opções elencadas, as mais votadas foram: *Agile Coach* com 19, *UX/UI Designer* com 18, *Product*

Figura 4.11 – Experiência em Anos Sobre os Profissionais com Maior Seleção Entre as Duas Opções



FONTE: Autor, 2020

Owner com 12 e especialistas com 10. As demais ficaram abaixo de 10 no número de resposta, sendo o papel design de interação não foi selecionado e apenas 1 profissional marcou a opção Pesquisador.

Entretanto, a maioria dos respondentes não encontrou o seu papel na lista pré-definida e marcaram a opção Outros. A opção 4 foi a única que se denominou um *design thinker*, sendo o mesmo papel descrito do seu LinkedIn e de acordo com a *string* de busca executada para encontrar estes profissionais. As opções 5, 7 e 11 se identificaram com mais de uma opção, convergindo em um papel mais de facilitador e consultor. Segue a lista abaixo descrita pelos profissionais de DT no desenvolvimento de software:

1. *Gestor Ágil;*
2. *Líder de um programa de Inovação Global de uma conta/projeto de mais de 1000 profissionais;*
3. *Coordenador de sistemas (squad leader);*
4. *Design Thinker;*
5. *Analista é a função principal, mas atuo como Facilitadora em outras situações (dentro e fora da empresa onde trabalho);*
6. *Chefe da Área de Inovação;*
7. *Além de UX/UX, acumulo mais alguns dos papéis acima, principalmente o de especialista, analista e consultor;*

8. *Gerente de Projetos;*
9. *Manager;*
10. *Marketing Analyst;*
11. *Na verdade sou uma mistura de várias opções acima, como: Facilitador, Pesquisador, Designer de interação, UX/Designer, Especialista na área de design centrado no usuário. Atuo como consultor;*
12. *Tribe lead;*
13. *UX designer Especialista;*
14. *Chefe de equipe de desenvolvimento;*
15. *Líder de projeto;*
16. *Consultor de Estratégia;*
17. *Coordenador de Inovação, CSPO, CLF e Designer;*
18. *Scrum Master e Desenvolvedora.*

Após conhecer o profissional que respondeu, sabendo seu nível de experiência sobre o assunto e com qual nome denomina-se um profissional do DT, segue na próxima seção uma análise individual de cada questão apresentada no questionário, inseridas dentro de seus blocos relacionais com as questões de pesquisa do mapeamento sistemático. As questões abaixo podem ser verificadas pelo Apêndice B. Como um todo obteve um total de 158 respostas que foram até o final da pesquisa e 171 respostas que responderam apenas o primeiro bloco, ou seja, 13 profissionais desistiram no meio da pesquisa mas responderam todo o primeiro bloco na qual representa uma parte importante da pesquisa que averigua as questões sobre os modelos, técnicas e ferramentas utilizadas.

4.4 Resultados da Survey

Essa seção está dividida em 3 subseções, organizadas conforme os blocos do questionário mostrados na Figura 4.3 a fim de responder as lacunas apresentadas no Figura 4.2. Todas questões apresentadas a seguir possuem a informação se são FECHADAS, para os casos de questões objetivas, ou ABERTAS, para questões discursivas. As questões discursivas foram analisadas individualmente para entender o conteúdo elaborado pelos participantes e resumir em uma resposta única. O Apêndice C apresenta todas as respostas obtidas no Estudo 2.

A Subseção 4.4.1 tem como objetivo apresentar à análise sobre as questões 1 a 5, representando as lacunas que foram encontradas nas questões QP3, QP4 e QP5.

4.4.1 Bloco 1: Modelos, Técnicas, Ferramentas: Grau de Dificuldades de escolha (QP3, QP4 e QP5)

S1: Existem diversos modelos de processo, que abstraem os espaços de trabalho no uso de Design Thinking. Você utiliza algum desses modelos como referência nas suas atividades? Marque todas opções que se aplicar. (Questão FECHADA)

Nesta questão foram apresentados 11 modelos que foram estudados e trazidos na literatura através do mapeamento sistemático. O objetivo foi identificar quais destes modelos, conforme mostra a questão 1 no Apêndice B, são conhecidos na indústria e/ou se existem outros modelos não descritos e que estão sendo utilizados no processo de desenvolvimento de software. A Figura 4.12 apresenta o resultado dos modelos mais referenciados pela indústria.

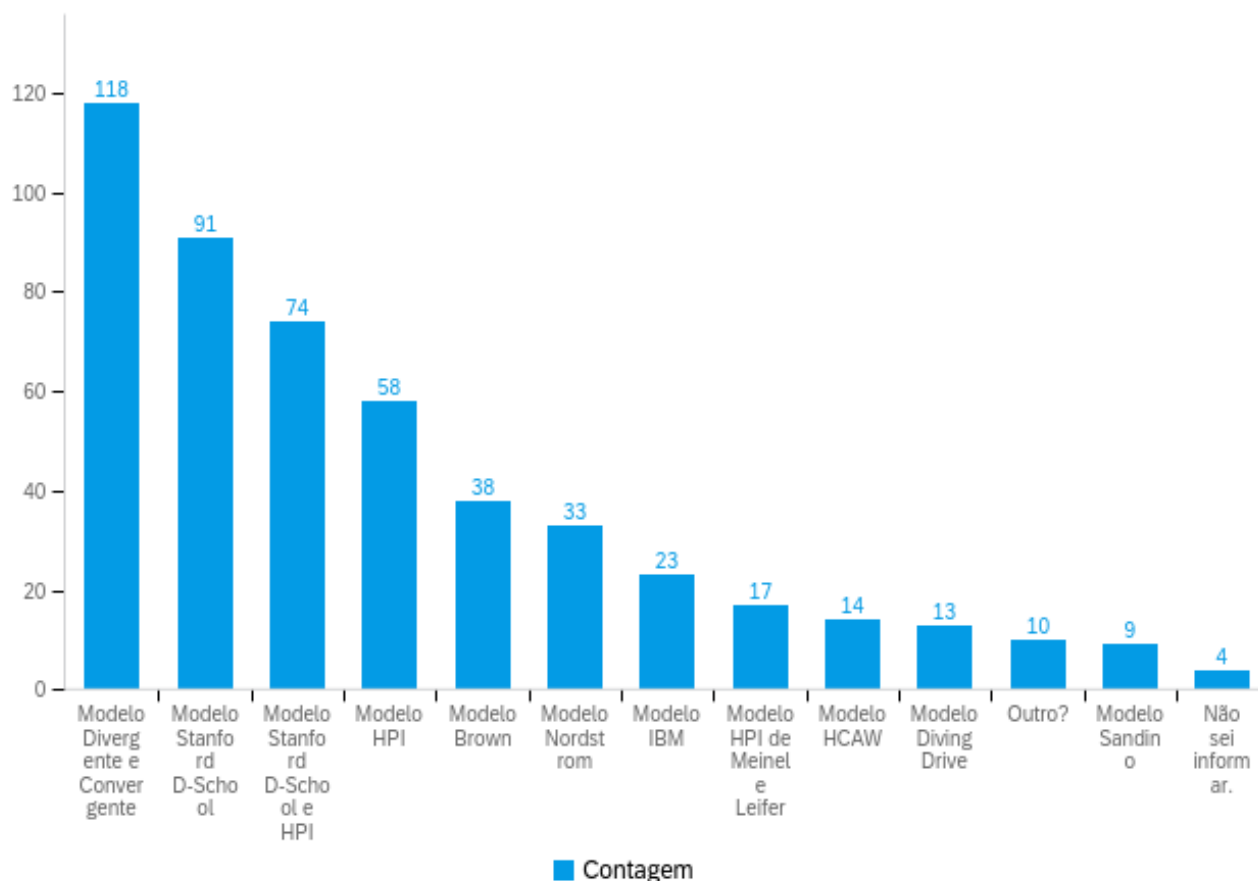
O modelo Divergente e Convergente foi o mais destacado, com 51 dos profissionais utilizam a ideia de divergir e convergir quando se usa o DT no desenvolvimento de software. Os próximos que se destacam é o modelo de *Stanford D-School* e seguido com o modelo HPI. Na opção *Outro?* entrou três novos modelos que a literatura não trouxe mas foram abordados pelos profissionais. São eles: Modelo *Double Diamond*, Modelo usado pela empresa ECHOS e um Modelo criado internamente na CPQD. O Modelo *Double Diamond* já foi discutido no referencial teórico, conhecido também como 4D, possui as etapas de descoberta (*Discover*), definição (*Define*), desenvolvimento (*Develop*) e entrega (*Deliver*), onde sobre as etapas possui a ideia de divergir e convergir sobre sua estrutura de diamante. Se olhar para o modelo mais referenciado, modelo de divergência e convergência, pode-se correlacioná-lo como um modelo *double diamond*, tendo apenas a diferença que o 4D é representado por etapas com objetivos pré-definidos e não tão aberto como o modelo Divergente e Convergente.

Sobre os outros dois modelos, da ECHOS e da CPQD, foi realizando um levantamento na internet para identificar como são estes modelos. A ECHOS⁴ é uma escola de DT, criada em 2012 como um laboratório de inovação com propósito de formar uma geração de inovadores na cultura brasileira [35]. A estrutura do modelo ECHOS pode ser vista pela Figura 4.13.

O modelo representado na Figura 4.13 é semelhante ao modelo criado pela HPI, mostrado na Figura 3.9, contendo duas diferenças, são elas: a segunda etapa é representado pelo nome PESQUISA pela ECHOS e na HPI é conhecida como a etapa de OBSER-

⁴<https://escoladesignthinking.echos.cc/>. Acessado em 29/12/2019

Figura 4.12 – Resultado - Modelos Mais Referenciados



FONTE: Autor, 2020

VAR. A etapa da ECHOS chamado de PONTO DE VISTA pode ser considerada a mesma pela etapa DEFINIR na HPI, visto que na versão inglês do modelo essa etapa na HPI é conhecida como *Point of View*. O modelo ECHOS possui uma etapa a mais da HPI, a etapa ITERAÇÃO, ao final do modelo, após a etapa de TESTE. Essa análise é totalmente superficial sobre as características apresentadas pela estrutura do modelo e não foi realizado um estudo profundo para entender se o trabalho realizado pelo modelo ECHOS se equivale ao trabalho realizado no modelo HPI.

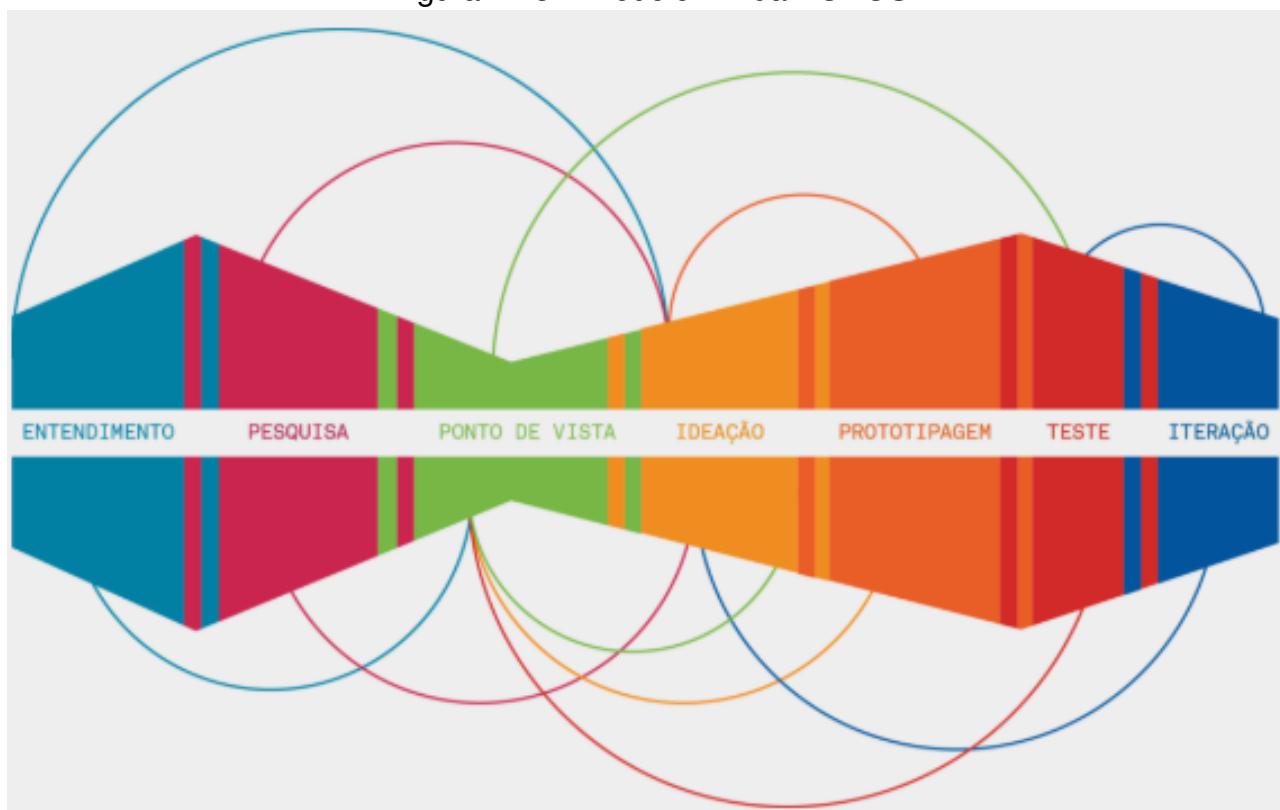
Outro modelo encontrado foi criado internamente na empresa CPQD⁵, porém não foi encontrado mais informações sobre o modelo para descrevê-lo.

O modelo da IBM, na qual foi citado no mapeamento sistemática da literatura e trazido como uma opção para os respondentes, foi dito na pesquisa que não é mais o modelo de DT utilizado pela IBM e que o mesmo foi atualizado pela IBM e é chamado de *LOOP*⁶. Este modelo segundo site da IBM apresenta um ciclo contínuo de observação, re-

⁵<https://www.cpqd.com.br/>

⁶<https://www.ibm.com/design/thinking/page/framework/loop/>

Figura 4.13 – Modelo DT da ECHOS



FONTE: ECHOS [35]

flexão e criação, buscando no entendimento do presente e prever o futuro. Inicia-se por um ciclo chamado *OBSERVE*, que busca o entendimento e identificando novas oportunidades através do conhecimento sobre o usuário em seu habitat natural, ou seja, observando os usuários em seu mundo acaba oferecendo a oportunidade de simpatizar com a sua experiência, entender seu contexto, descobrir necessidades e ouvir seus comentários mais irrestritos. Esse ciclo aponta para alcançar alguns objetivos, como: Conhecer o usuário, entender o contexto, descobrir as necessidades e ouvir os feedbacks. O próximo ciclo é a *REFLECT*, que trabalha com a empatia para entender diversas perspectivas, flexibilidades para responder às mudanças e integridade para o permanecimento fiel aos valores da equipe, ou seja, refletir regularmente com a equipe para sincronizar e sintetizar os pensamentos. Esse ciclo possui o foco de conhecer um ao outro, alinhar as intenções, descobrir novas ideias e planejar com antecedência. Por final, o ciclo que completa o modelo *LOOP* da IBM é o *MAKE*, que é montar uma forma concreta sobre as ideias abstratas, buscando transformar as ideias em formas e lança-las no mundo real, para isso procura-se explorar as possibilidades, comunicar as ideias, ou seja, mostrar as pessoas suas ideias e não contá-las, prototipar para validar as hipóteses e impulsionar os resultados [53].

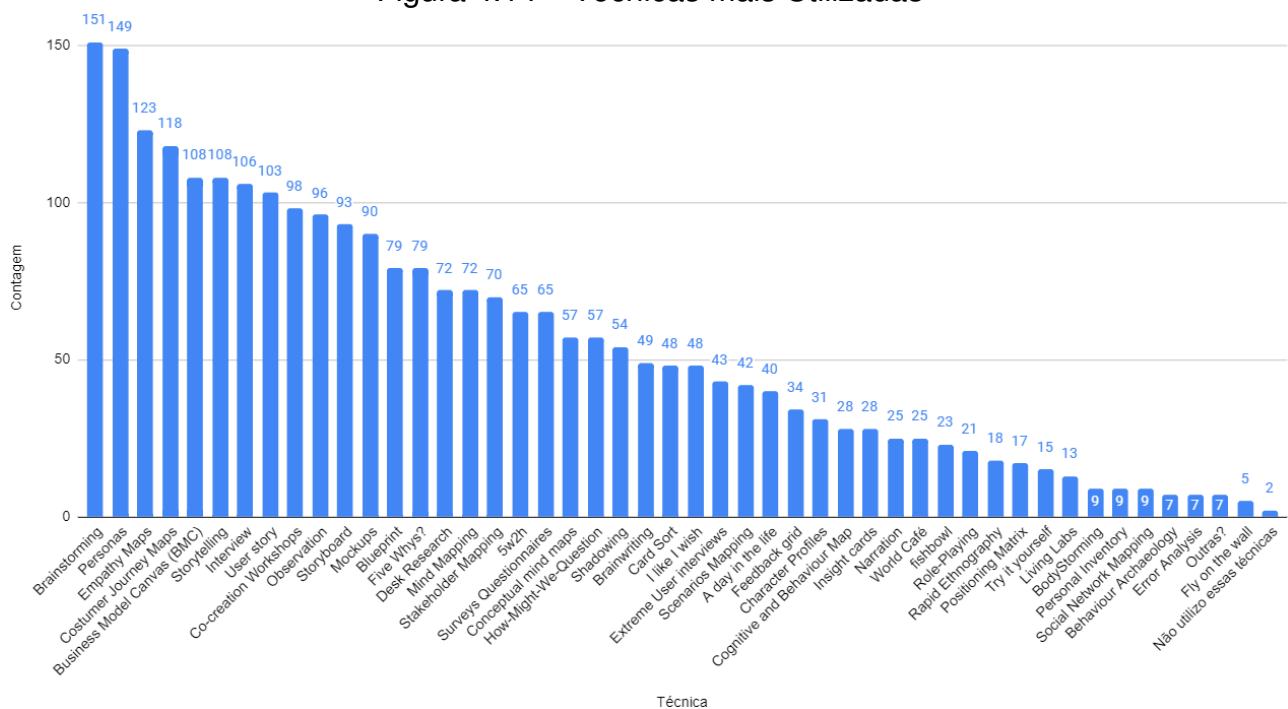
Todos os modelos trazidos da literatura foram identificados pelos profissionais da indústria e todos souberam informar pelo menos um modelo na qual baseia-se para utilizar o DT no dia a dia, mostrando que os modelos podem ser considerados uma estrutura referencial para o desenvolvimento do DT dentro do processo da Engenharia de Software. A ideia do modelo Divergente e Convergente, que propõe criar escolhas no processo de divergir e escolher no processo de convergir, é abordada em todos os modelos em si e será melhor discutido no Capítulo 5 de consolidação dos resultados.

Após a questão dos modelos, foram apresentados 46 técnicas que foram encontradas na literatura e são utilizados durante o processo de DT. As técnicas foram apresentadas através da questão dois para os participantes.

S2: Diversas técnicas podem ser usadas para apoiar o uso de Design Thinking. Quais técnicas você costuma utilizar? Marque todas as opções que se aplicar. (Questão FECHADA)

A Figura 4.14 mostra as técnicas mais utilizadas pelos participantes durante o processo de DT.

Figura 4.14 – Técnicas mais Utilizadas



FONTE: Autor, 2020

As técnicas *Brainstorming*, *Persona*, *Empathy Maps*, *Interview*, *Observation* e *Storyboard* foram consideradas as top 6 técnicas mais citadas, tanto na *survey* como no mapeamento sistemático. A *survey* trouxe uma grande quantidade de uso das outras técnicas que

pouco foram citadas no mapeamento e podem ser relevantes para apoiar o profissional no processo de DT.

Também, surgiram outras técnicas que não foram citadas no mapeamento e foram citadas pelos respondentes. São estas:

- *Problem Tree*;
- *Crazy 8*;
- *Crazy 4*;
- *Dot Voting*;
- *Chain of Undesired Phenomena*;
- *Job Story*;
- *Entrevistas Contextuais*;
- *Dinâmicas de "quebra-gelo"*;
- *Hyphothesys Cards*;
- *Value Proposition Canvas*;
- *Test Cards*;
- *Chain of Undesired Phenomena*;
- *User Story Mapping*.

Concluí-se que houve uma identificação dos modelos e das técnicas citadas pelos respondentes e o que a literatura está trabalhando neste tema está próximo do que realmente a indústria está aplicando. É interessante e necessário fortalecer o ensino das técnicas visto que o resultado trazido na próxima questão mostra que a escolha das técnicas está mais relacionado a experiência do profissional sobre a técnica do que o modelo em si.

S3: Como você geralmente decide quais técnicas utilizar? Marque todas as opções que se aplicar. (Questão FECHADA)

A Tabela 4.1 mostra o motivo na qual são escolhidas as técnicas. As primeiras três maiores escolhas está de acordo com a necessidade do momento e experiência prévia do profissional sobre a técnica que irá utilizar. Logo, a importância de trazer as técnicas para o ensino ao ponto de fortalecer a experiência daqueles que irão atuar com DT no processo de

desenvolvimento de software, agregará maior valor e será mais assertivo durante execução. A quarta maior decisão vai de encontro com os modelos, onde os profissionais destacam que cada técnica está de acordo e dentro de um espaço ou etapa do DT. Com isso abre a necessidade de um estudo empírico na indústria para correlacionar os modelos, a nível de suas etapas, com as técnicas e decidir qual técnica melhor enquadra-se dentro do espaço de cada modelo. Também, percebendo que a experiência prévia e saber qual técnica deve ser executada de acordo com a necessidade do contexto inserido é a decisão mais trazida pelos profissionais é de suma importância a disseminação do conhecimento destas técnicas para apoiar os profissionais, não apenas aqueles com experiência, mas também preparar aqueles que irão adentrar neste universo.

Tabela 4.1 – Resultado Escolha das Técnicas

Descrição	Qtd. de respostas
Quando a técnica se enquadra na minha necessidade.	137
Depende muito do contexto que vou utilizar.	131
Baseado na minha experiência prévia.	128
Escolho as técnicas de acordo com a espaço/etapa do DT, onde cada espaço/etapa possui suas próprias técnicas.	106
Por indicação de um colega.	29
Já tenho o meu catálogo de técnicas que sempre utilizo.	26
Recomendação pela minha empresa.	26
Geralmente preciso estudar as técnicas pois nunca sei qual é a melhor para o momento.	16

FONTE: Autor, 2020

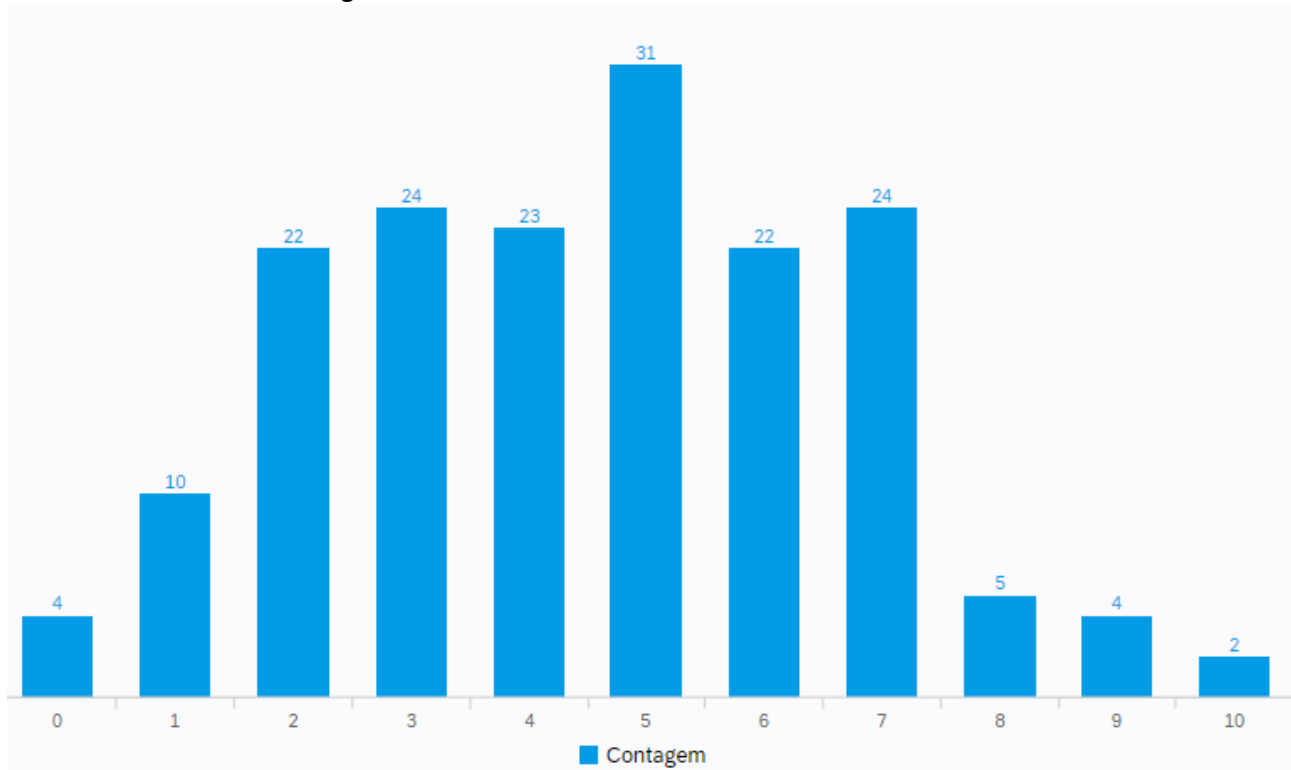
Sabendo que os profissionais vão ao encontro de técnicas de acordo com o contexto e sua prévia experiência, a questão 4 traz entender se estes profissionais possuem dificuldade em escolher a técnica para utilizar. Dentre tantas técnicas citadas, pode-se dizer que todas já foram usadas pelo menos uma vez pelos profissionais, certamente não usadas pelo mesmo mas dentro da amostra coletada de profissionais, todas as técnicas trazidas pela literatura são conhecidas e usadas na indústria. Também, um dos profissionais destaca que: *Não existe sequencia certa. O melhor é combinar mindset e as ferramentas mais adequadas para cada cliente e situação.*, ou seja, sua descrição está mais voltada para opção que baseia-se na experiência prévia e necessidade do contexto inserido.

S4: Em uma escala de 0 (Nenhuma dificuldade) a 10 (Dificuldade extrema), qual o grau de dificuldade que você sente em decidir quais técnicas utilizar em uma determinada situação? (Questão FECHADA)

A tendência encontrada nesta resposta mostra que a amostra possui uma média de 4.50 de grau de dificuldade com um desvio padrão de 2.18, ou seja, significa que os valores amostrais estão próximos a média encontrada. Pelo gráfico apresentado na Figura 4.15 existe um crescimento no lado direito da média, mostrando que os profissionais pos-

suem mais dificuldade na escolha das técnicas para aplicação durante o processo de DT no desenvolvimento de software.

Figura 4.15 – Resultado do Grau de Dificuldade



FONTE: Autor, 2020

Em conjunto com o resultado da questão 2, muitos respondentes escolheram mais de uma técnica, sendo que nenhum escolheu menos de 5 técnicas e teve respondentes que conhece 35 técnicas. A maioria apresentou conhecimento para mais de 20 técnicas. Sobre as 46 técnicas apresentadas, sendo que muitos mostraram que conhecem mais de 20 técnicas, entende-se que pode sim existir tal dificuldade em escolher as técnicas dado um catálogo representativo de muitas opções para selecionar.

Imagine-se que estes profissionais possuem o conhecimento de suas técnicas, e sempre tem que avaliar qual técnica utilizar de acordo com o contexto inserido ou procurar aquela técnica na qual possui mais experiência. Isso nem sempre irá trazer sucesso no processo de DT, visto que, ter que escolher a técnica pois possui uma experiência melhor sobre ela não quer dizer que a técnica é a mais adequada para aquela situação ou contexto. O mesmo serve para o contrário, escolher tal técnica pois é adequada para o processo inserido mas não se sentir confortável em executá-la por falta de experiência pode não ser assertivo e benéfico. A maioria dos profissionais apresentaram uma dificuldade mediana sobre a decisão em quais técnicas escolher. O gráfico apresenta uma quantidade significativa a direita (maior dificuldade), onde mostra que muitos dos profissionais, praticamente

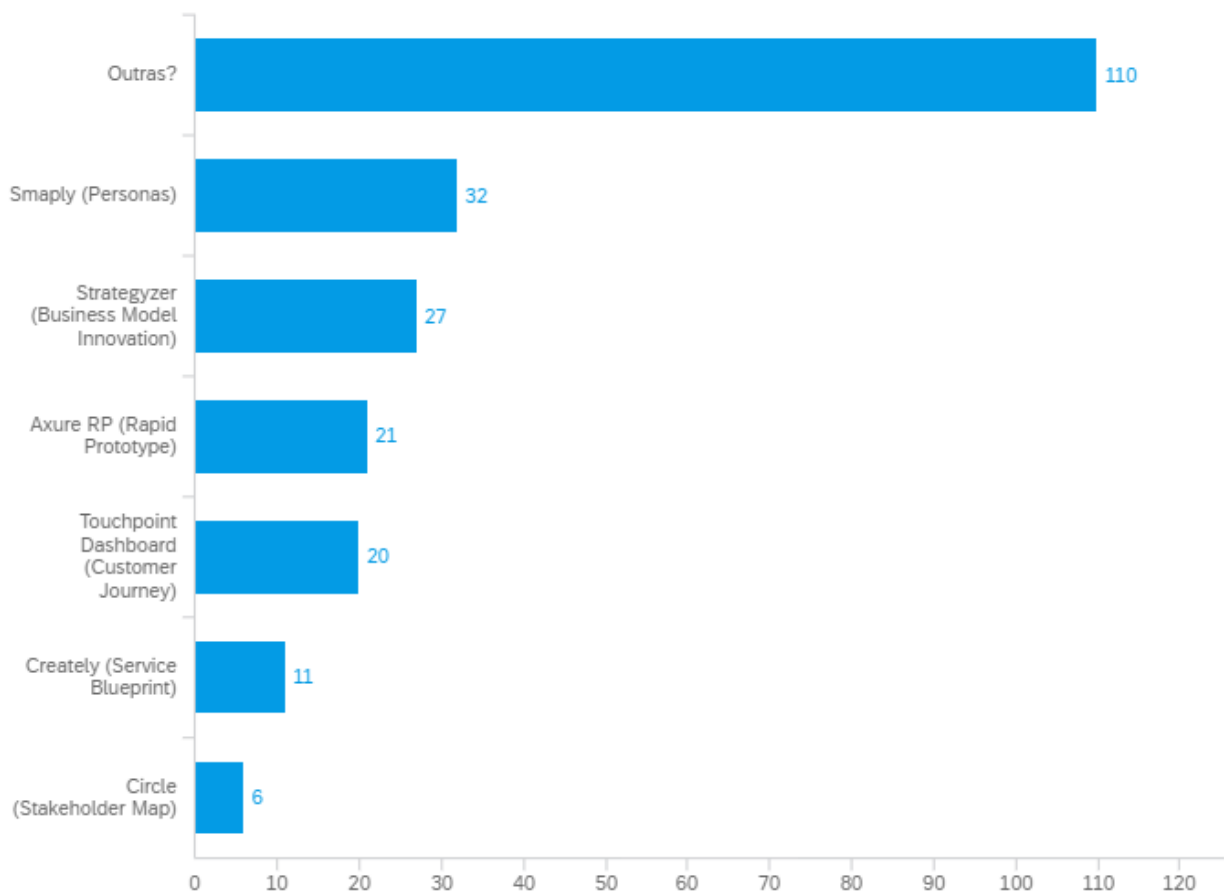
a metade que respondeu este questionário, possui uma dificuldade em escolher e decidir qual técnica aplicar. Aqueles que apresentaram menor dificuldade, foram também os profissionais que apresentaram conhecimento menor que 20 técnicas, pode-se concluir que é mais fácil escolher pois possui um catálogo pequeno de técnicas pela experiência.

A próxima questão traz o conjunto de softwares que apoiam os profissionais no desenvolvimento do DT.

S5: Você utiliza algum software (ou sistema computacional, como preferir denominar) para apoiar o uso das técnicas de Design Thinking? Marque todas as opções que se aplicar. (Questão FECHADA)

Na lista de resposta foram apresentados apenas 6 softwares, visto que no mapeamento sistemático trouxe poucas escolhas. A Figura 4.16 o resultado do conjunto de softwares que apoiam os profissionais da amostra.

Figura 4.16 – Resultado dos Softwares que Apoiam os Profissionais



Todos os softwares apresentados são conhecidos pela amostra e interessante que trouxeram outros softwares que não foram apresentados na literatura. Segue abaixo a lista dos outros softwares. Entre os mais citados na opção outros, encontram-se: Miro, com 17 citações, *Invision* e Figma com 7, Marvelapp com 4 e Adobe XD com 2.

- paint;
- Evrybo;
- Xmind - Stakeholder map;
- Whimsical;
- Figma;
- Mural ou Mural online⁷
- Real Time;
- Muraly;
- Ferramentas de mind map, em geral;
- Build para prototipação;
- Canvanizer para *Business Model*;
- Google Sheets;
- Google presentation;
- POP
- Mindmeister;
- Hotjar;
- *Survey Monkey*;
- *Illustrator*;
- Photoshop;
- SAP Build.

⁷<https://mural.co/>

Foram apresentados 25 novos softwares que podem ser utilizados como apoio durante o processo de DT pelos profissionais. O interessante é que 48 profissionais indicaram que não utilizam nenhuma ferramenta como apoio, nenhum recurso digital, realizando as tarefas geralmente a mão através de *post-its*, um mural e canetas. Um profissional destacou: *Não uso nenhum software específico. Alias, o DT propõe que façamos as coisas em papel e caneta, para estimularmos a criatividade e aí sim os resultados finais podem ser documentados via software.* Logo, as ferramentas foram trazidas pelos profissionais como um apoio mas não é obrigatório e muitas vezes não é bom usar, pois desqualifica o que o DT propõe a fazer.

Com a resposta das 5 primeiras questões, fecha-se o primeiro bloco da *survey*, podendo dessa forma consolidar as informação para fechar as lacunas encontradas na literatura sobre as questões QP3, QP4 e QP5. Conforme a Figura 4.2, estas questões vieram para responder as seguintes lacunas:

- Profissionais conhecem todos esses modelos e técnicas? Eles tem dificuldade em escolher qual usar? A *survey* trouxe que todos os modelos e técnicas são conhecidas sim pelos profissionais e que não existe nenhuma técnica ou modelo na qual eles não conhecem, apenas existem aquelas que são mais utilizadas. Apresentou-se que sim, existe uma dificuldade de escolher qual técnica utilizar, talvez não seja muito representativa, mas existe, mesmo que muitas são escolhidas de acordo com a experiência. Mais de 100 profissionais trouxeram que existe uma relação das técnicas com as etapas dos modelos. A Tabela 4.2 mostra os modelos que foram selecionados por aqueles profissionais que marcaram a opção que as técnicas são escolhidas de acordo com o espaço/etapa do DT, onde cada espaço/etapa possui suas próprias técnicas.

Tabela 4.2 – Modelos com Mais Relações com as Técnicas

Modelo	Qtd. de respostas
Modelo Divergente e Convergente	79
Modelo Stanford D-School	64
Modelo Stanford + HPI	49
Modelo HPI	40
Modelo Brown	28
Modelo Nordstrom	19
Modelo IBM	15
Modelo HPI de Meinel e Leifer	12
Modelo Diving Drive	9
Modelo HCAW	8
Modelo Sandino	7

FONTE: Autor, 2020

Os 5 primeiros modelos apresentaram mais relação com as técnicas. O mais selecionado, conforme mostra a Tabela 4.2, foi o modelo Divergente e Convergente, onde 79 profissionais que selecionaram que a escolha das técnicas é muitas vezes devido ao espaço/etapa do DT, também utilizam este modelo como referência, seguido pelos modelos *Stanford* com 64, *Stanford + HPI* com 49, HPI com 40 e Brown com 28.

Sobre análise das técnicas, *Brainstorming*, *Personas Empathy Maps*, *Customer Journey Maps*, *Business Model Canvas (BMC)*, *Storytelling*, *Interview* e *user story* foram as técnicas mais selecionadas na pesquisa sobre todos os modelos, ou seja, praticamente essas técnicas foram marcadas pelo menos para cada 1 dos modelos selecionados. Percebe-se também que as técnicas focam totalmente em uma representação abstrata do usuário ou cliente, como por exemplo a persona, na qual busca gerar um personagem fictício que representa o usuário. As técnicas mais utilizadas tangem para um foco totalmente em entender e conhecer quem é este usuário ou cliente e como representá-lo e entendê-lo.

Todas as 46 técnicas disponibilizadas aos profissionais na pesquisa foram selecionadas pelo menos 2 vezes, além de 13 novas técnicas terem surgido totalizando 59 técnicas que fornecem a aplicação do DT no desenvolvimento de software, mostrando também haver uma grande variação de técnicas que existem e podem ser utilizadas. Além de que as técnicas mais citadas apontam uma relação mais forte e envolvente com o objetivo de conhecer o usuário, tanto suas características pessoais como seu processo de trabalho, decorrendo por técnicas que ilustram a busca por soluções sobre o contexto inserido. Os profissionais da amostra focam em conhecer o processo de trabalho do usuário ou cliente a fim de explorar o problema e promovendo em conjunto uma solução viável, de valor e válida para os envolvidos.

- Buscar mais ferramentas que apoiam o uso do DT para o desenvolvimento de software. Essa lacuna foi respondida pois além das poucas ferramentas trazidas pela literatura, conseguiu-se mais 25 ferramentas de software que apoiam no desenvolvimento do DT. Muitos respondentes trouxeram a ideia de não utilizar ferramentas no processo de DT e que praticam utilizando apenas materiais como papel e caneta.
- Qual o embasamento dos profissionais para escolher um modelo ou técnica? Essa questão trazida já foi respondida diretamente na resposta da questão S3, onde mostra que a maioria dos respondentes escolhem suas técnicas conforme sua experiência ou necessidade de acordo com contexto. Também, conforme já apresentado, mostra um forte relacionamento das técnicas com as fases dos modelos. Essa questão abre uma margem para estudar detalhadamente como os profissionais trabalham e decidem qual técnica utilizar, cabendo a realização de uma pesquisa mais profunda entendendo como funciona o dia a dia destes profissionais com objetivo de eliminar a incerteza que os profissionais escolhem as técnicas de acordo com a fase do modelo ou escolhem a técnica conforme a situação inserida e sua experiência, qual seria mais

viável? Infelizmente esta pesquisa aponta apenas a escolha e que existe a possibilidade de ambas situações mas não apontam qual possui maior benefício. Existe uma enorme diversificação na escolha das etapas do DT a serem seguidas, nas técnicas a serem usadas e os softwares que apoiam e pelos quais motivos que levam às escolhas de tais técnicas. Também importante salientar que metade dos profissionais que responderam apontam que não possuem dificuldade de escolher a técnica e que o processo do DT aparenta não ser fechado e admite adaptações durante o decorrer de seu uso.

Com o fechamento do bloco 1, na qual aborda um contexto para o entendimento da parte tática do DT, ou seja, entendendo seus modelos, técnicas e ferramentas, passa-se a seguir à analisar os resultados trazidos no bloco 2, que caracterizam o DT na forma do contexto que está sendo aplicado.

4.4.2 Bloco 2: Propósito e Cenários de uso: Benefícios e Dificuldades (QP1, QP2 e QP6)

Esse bloco apresenta as questões que procuraram descobrir os motivos que levaram à escolha da adoção do DT nos processos da empresa, em quais contextos são aplicados e os benefícios e dificuldades na aplicação do DT.

S6: Para qual fim você usa Design Thinking no desenvolvimento de software? Marque todas as opções que se aplicar. (Questão FECHADA)

Essa questão buscou entender como os profissionais estão usando o DT, para qual o propósito. A questão abordou termos resumidos da literatura através do mapeamento para entender para qual fim realmente é utilizado o DT no desenvolvimento de software. Ao listar, de forma resumida, na questão e levar para os profissionais, tiveram 147 respostas abordando que o DT serve para gerar ideias e soluções, 142 para explorar e entender o problema, 109 para criar ideias inovadoras e reduzir as incertezas. Essas foram as 4 respostas mais selecionadas, conforme mostra a Tabela 4.3.

Percebe-se pela Tabela 4.3 que termos mais voltados ao processo de software foram mais classificados a partir da quinta resposta mais votada, onde traz questões de entender e especificar os requisitos, aproximar a equipe de desenvolvimento, melhorar a satisfação do cliente, ganhar a empatia do usuário e, perto das últimas questões, usar DT pois é possui fácil relacionamento com os métodos ágeis. Isso vai contra o que foi trazido pela literatura, onde criou-se uma análise sobre o mapeamento que muitos artigos trouxeram a ideia de usar o DT relacionando diretamente com um método ágil.

Tabela 4.3 – Propósito de Uso do DT

Propósito de uso	Qtd. de Respostas
Para gerar ideias e soluções	147
Explorar e entender o problema	142
Para criar ideias inovadoras	109
Para reduzir as incertezas	109
Entender e especificar os requisitos	103
Melhorar a satisfação do cliente	91
Aproximar a equipe de desenvolvimento com o cliente	87
Fácil relacionamento com os métodos ágeis	69
Ganhar a empatia dos usuários	62
Validação do software	46
Para gerenciar projetos	20
Para o desenvolvimento de jogos	6
Outro	3

FONTE: Autor, 2020

A Figura 3.6, onde aborda os diversos usos de DT trazidos na literatura, mostra que 46,3% dos artigos extraídos utilizam o DT relacionado com o fator ágil trabalhado no desenvolvimento de software. O DT trazido no mapeamento abordou 4 frentes, sendo elas na ordem de mais citados:

- Relacionado com métodos ágeis;
- Entender o problema;
- Inovação;
- Trabalhado na educação.

Entretanto, com a pesquisa realizada com os profissionais através desta *survey*, trouxe uma visão invertida do que realmente o DT vem oferecer. A Tabela 4.3 mostra de forma comparativa o uso do DT no desenvolvimento de software.

Entende-se que à abordagem Educação não foi trazida da literatura para indústria, pois o foco é outro. O DT no mapeamento sistemático trouxe forte relação com os métodos ágeis e já a *survey* trouxe o DT mais voltado para entender o problema e gerar ideias, onde muitas são trabalhadas para uma viés na inovação. Por mais que concluí-se que o DT sim, possui forte relacionamento com os métodos ágeis de acordo com os valores que o DT propõe a trazer, com o foco no usuário, totalmente centrado em entender o usuário, pode-se dizer que em sua essência o DT é dito pela comunidade na qual o utiliza que é um mecanismo gerador de ideias, muitas vezes inovadoras conforme o trabalho realizado para explorar e entender o problema trazido pelo usuário, na qual aborda um valor que o DT não precisa ser ou estar relacionado aos métodos ágeis ou aos valores ágeis para ser usado

pela comunidade que o aplica dentro do desenvolvimento de software, apenas é preciso surgir a necessidade de entender o problema ao ponto de explaná-lo para criar uma solução. Fechando esta questão, coloca-se abaixo dois fatores trazidos por dois profissionais, um designer de formação e outro e outro focado no ágil na computação. Eles citam:

- *Alguns questionamentos que, aparentemente para mim, você trás na pesquisa são um pouco fora do cotidiano. Um exemplo é os modelos de design thinking. Tinha alguns ali que nem conhecia, mas no fim são todos parecidos e muda conforme o seu desafio. Se estou desenvolvendo um produto: a parte de teste, validação e aprendizado são essenciais. Se estou facilitando um grupo para explorar novas ideias: foco mais na pesquisa e em workshops de ideação.*

Para o profissional da computação, cita:

- *O DT para engenharia de software é um método ágil.*

Curioso o olhar de cada profissional. Aquele que traz uma bagagem e nascido na área de formação da Engenharia de Software, possui uma visão que o DT é um método ágil. Por outro lado, o designer de formação desconhecia diversos modelos mas entende de forma macro que todos possuem o mesmo foco, mudando apenas a necessidade de usá-los de acordo com as etapas que os formam. Para focar em um produto, entende-se que de fato as etapas de testes, validação e aprendizado são importantes. Logo os métodos que possuem estas etapas possam ser mais utilizados nesse contexto e quando se usa o DT como um facilitador em um grupo para explorar novas ideias, etapas que focam na pesquisa e workshops de ideação possuem um propósito correspondente.

O DT trazido pelos profissionais com bagagens distintas, mostra que o DT possui um objetivo muito maior sobre explorar o problema e gerar ideias do que apenas usá-lo por estar relacionado com método ágeis. Claro que por traz dos métodos ágeis existem diversos fatores que são usados para gestão de riscos e que os métodos ágeis são formas táticas de execução, aquela que se pratica com os usuários e por traz existem valores estratégicos e de visão sobre o produto e sobre o negócio, não assim desmerecendo o relacionamento e sim se fortalecendo quando se une o DT com os métodos ágeis.

Também 2 profissionais trouxeram outros propósitos para o uso do DT, como: Planejamento Estratégico, Problemas Industriais, Problemas Complexos, Ajustes e melhorias de processos. Resumindo, também trouxeram a questão de explorar o problema.

S7: Quais são os cenários de uso comuns onde você usa o Design Thinking? Marque todas as opções que se aplicar. (Questão FECHADA)

Em conjunto com a questão S6, a questão S7 também busca explorar onde o DT é aplicado com foco no desenvolvimento de software. Sabendo que o DT possui um objetivo,

um propósito maior na geração de ideias, na inovação e no entendimento do problema, a Tabela 4.4 mostra quais são os cenários trazidos por estes profissionais.

Tabela 4.4 – Cenários de Uso mais Selecionados

Cenário de uso	Qtd. de Respostas
Com equipes multiciplinares	135
Utilizado em parceria com os métodos ágeis (Lean, Scrum)	114
Criação de produtos/software inovadores	110
Criar cocriação entre os participantes do projeto	101
Inovação como um todo, desde o processo de desenvolvimento até o software	83
Mudanças e melhorias no desenvolvimento de software	66
Dentro de um processo de desenvolvimento de software diariamente/semanalmente, acompanhado por toda equipe (do cliente até o desenvolvedor)	51
Outro	6

FONTE: Autor, 2020

Percebe-se que toda resposta focado em software, inovação em todo processo de software, diariamente no processo de software, em parceria com os métodos ágeis (que possui uma origem mais para o lado da engenharia da computação) ou mudanças e melhorias no desenvolvimento de software, foram as respostas menos marcadas pela amostra.

O DT mostra que é mais utilizado em cenários que abordam equipes multidisciplinares, isto é, o DT aparenta precisar para o seu uso comum um conjunto de pessoas, tendo elas o mesmo interesse comum e possuindo origens diferentes, podendo ser especialistas na área específica, o usuário ou cliente que aborda e possui o real problema, podendo ele ou eles serem de diversas áreas, onde cada um possui o sua própria visão sobre o negócio (diretores, operadores, analistas, técnicos ou estagiários) e também facilitadores dos métodos e técnicas que serão aplicados durante as sessões de DT para ajudar não apenas nas explicações técnicas mas fortalecer o senso de cocriação entre os participantes e buscar a emergência de inovação para a criação de produtos ou softwares inovadores. Também emergindo na melhoria de processos internos e na concepção de novos projetos.

A seguir as duas últimas questões do questionário que tratam quais são os pontos positivos de realmente se utilizar o DT e quais são as dificuldades encontradas por estes profissionais. Estas questões não apenas mostram as dificuldades e os valores do DT mas ajudam a retratar o propósito de uso e os cenários que melhor se aplica.

S8: Na sua experiência de uso de Design Thinking no desenvolvimento de software, o que você apontaria como benefícios ou pontos positivos trazidos pela adoção da abordagem? (Questão ABERTA)

Essa questão trouxe em profundidade os benefícios que o DT está trazendo para a comunidade da Engenharia de Software e com as respostas consegue-se perceber um relacionamento com a questão número 6, que buscou entender o propósito de uso do DT para o desenvolvimento de software. Os profissionais trouxeram como benefício a ideia que

o DT é de suma natureza qualitativa, onde busca entender com profundidade os usuários e trabalhar através de uma forma esquemática os dados que surgem no processo, sendo muito útil para equipes que muitas vezes desconhecem os usuários e acabam trabalhando com pensamento individual.

A maioria das respostas foram totalmente voltadas ao usuário, onde citam:

- *Manter o usuário no centro do processo sem deixar de lado as necessidades do negócio;*
- *Maior empatia com o usuário;*
- *Foco na necessidade do cliente;*
- *Alta colaboração com usuário;*
- *Entendimento das dores do cliente;*
- *Aproximação da equipe técnica com o cliente.;*
- *Alcançar o usuário da ponta, o Power user.*

Em conjunto com o que o mapeamento na literatura trouxe, onde traz com a maior citação, a integração com métodos ágeis, o entendimento melhor da necessidade do cliente, criação de ideias inovadoras e elucidação dos requisitos, os profissionais muito destacaram também importância sobre quatro frentes: aproximação do cliente/usuário e de toda equipe envolvida, o entendimento do real problema, poder errar mais rápido (validar hipóteses) e gerar ideias de valor. Realizando uma relação entre literatura e o que foi dito pelos profissionais, o DT possui como benefício a exploração do real problema e para isso é necessário uma peça importante para entender este problema, a presença cliente ou usuário. Dessa forma o benefício do DT aumenta visando entregar o que realmente importa através de cocriação entre os participantes dentro de um processo com equipes multidisciplinares, ou seja, um pouco de cada área envolvida junto com o usuário, promovendo assertividade na entrega devido a forma empática e ativa do envolvimento do usuário final com toda equipe através de validações prévias de protótipos oriundos das ideias e soluções que surgiram através do processo do DT, em outras palavras, o DT é trabalhado na indústria como uma forma tática de trazer e promover a possibilidade de errar várias vezes para ser assertivo na solução do real problema do usuário e este torna o maior benefício do DT, o poder de errar.

De acordo com os resultados, o DT é a forma de entregar projetos com alto valor agregado diretamente ao usuário final, com envolvimento direto, ganhando empatia e facilitando o processo de validação de requisitos, descobrindo o valor e aumentando a eficácia do processo de desenvolvimento de software, isto é, aumenta a possibilidade de alcançar um bom resultado e diminuindo o risco de entregar algo sem valor.

Existem diversas formas, como já visto nas questões que abordaram os modelos e as técnicas para alcançar a resolução ou validação do problema que deve ser resolvido, onde através destes modelos e técnicas aderi-se de forma rápida do envolvimento de toda equipe de desenvolvimento em conjunto com os *stakeholders* presentes, principalmente focando na parte mais interessadas de todas, o usuário. A forma que as técnicas trazem estimula a criatividade, melhora o relacionamento entre os envolvidos no projeto, criando dessa forma a possibilidade de errar sem ser feio e aprender a melhorar com isso, evitando o desperdício do tempo e dinheiro através dessa forma humanizada e estruturada de aproximar todos envolvidos, não apenas ajudando o cliente a se próprio entender, conhecendo o real problema, mas também ajudando a equipe de desenvolvimento no esclarecimento de dúvidas que surgem na ponta e, deixando de se preocupar com estas dúvidas, a equipe estimula outros fatores que não teriam tempo para trabalhar, a possibilidade de pensar fora da caixa e promover a inovação para o desenvolvimento do produto, seja ele um produto de software ou um mapeamento de processo.

Entretanto, em conjunto com todos esse benefícios trazidos pelos profissionais existem barreiras, desafios e dificuldade de aplicar o DT. Para isso foi perguntado através da próxima questão quais são estes desafios para promover o DT no desenvolvimento de software.

S9: E quais seriam as dificuldades ou os pontos negativos? (Questão ABERTA)

Como todo processo novo, que busca promover a presença de várias pessoas, sendo todas de grande importância para promover um resultado esperado, existem as mesmas dificuldades de sempre: Resistência dos envolvidos, tempo para que todos participem e mostrar valor na execução do processo. O DT trazido pelos profissionais trouxe a dificuldade de mostrar que o DT sim possui valor e não é apenas uma atividade lúdica onde reúne as pessoas para desenvolver a empatia e aproximá-las através de atividades, muitas vezes divertidas por estarem fora da curva do que é considerado normal no dia a dia. Geralmente as equipes relutam em participar, em mudar o *mindset*, em entender o benefício de algo novo dentro do processo na qual já é de seu costume.

Também, a importância de mostrar que o DT é um tempo necessário para ser usado. Sabe-se que sem usar o DT já é de grande dificuldade achar tempo para que todos estejam presentes em todos os espaços promovidos pelo DT. Muitas vezes o tempo não é ganho devido aquela famosa e cansativa desculpa da mudança cultural. Como conseguir mobilizar equipes, mostrar que é necessário e útil fazer o DT para os mais poderosos da empresa, aqueles que possuem o dinheiro. Como dizer para estes que é necessário usar o dinheiro para errar várias vezes no início.

A falta de efetividade do método pode descrever que o DT é benéfico para aqueles que investem e isso pode transformar a imagem do DT em algo que não é verdade. Muitas

vezes a forma de aplicação pode acabar em um guia de passo a passo, e isso não é DT. Um profissional respondeu: *Adoção do modelo no mundo corporativo versus modelos tradicionais. O mundo corporativo ainda é um pouco quadrado com metodologias centradas nas pessoas, e muito mais orientado a entregas com prazos e escopos bem definidos. Além disso, é um território pouco conhecido e às vezes mal interpretado por pessoas mais técnicas, que julgam ser algo desnecessário.* Em outras palavras comprar a ideia por ser bonita e não saber usá-la pode dificultar o verdadeiro valor que o DT propõe a levar para o ambiente de desenvolvimento de software. Cenários que não são adeptos ao conceito de UX, por exemplo, perde-se no meio do caminho por ninguém saber conduzir e levar adiante.

Além da cultura pesada, dificultando a inserção do DT, a falta de conhecimento das técnicas também promove um pensamento negativo sobre o seu uso e o tempo de todos para a realização dos espaços promovidos pelo DT. Existem outras questões mais pontuais que os profissionais trouxeram como dificuldades. São essas:

- *Alto nível de gerenciamento tem dificuldade para enxergar valor; Nerds não querem participar dos projetos, preferem ficar sentados cumprindo tarefas;*
- *Adequar o projeto ao tempo e escopo;*
- *Adaptação das pessoas em utilizar métodos; Desapego de soluções (colaboradores já vem com a solução e não com entendimento real do problema);*
- *Em projetos de evolução e com escopo muito definido (como migração/modernização), o Design Thinking acaba não sendo muito aplicável... Projetos com arquitetura muito definida (com tecnologias ou ferramentas de terceiros) também dificultam a utilização da abordagem, porque nessas situações há pouco espaço para inovação;*
- *Falta de experiência ao conduzir o design. Pessoas apegadas as ideias;*
- *Transformar dados qualitativos em dados válidos para o ambiente corporativo;*
- *Nem sempre o cliente tem tempo ou disposição para conhecer a fundo o problema, ele sempre acha que sabe o que precisa e a forma como ele pensou é a melhor solução.*

Resume-se como dificuldade: as pessoas, principalmente aqueles com poder do não. Aplicar qualquer método que seja, pode este não ser do DT, sempre haverá problema devido ao envolvimento das pessoas, por saber escalar melhor tal técnica para diferentes estereótipos e achar um tempo de cada um para participar da sessão. Um dos respondentes diz: *É necessário a evangelização do DT.* O mesmo profissional destaca a dificuldade que as empresas têm de autorizar o contato com o público, dificultando dessa forma o entendimento das dores e dos reais problemas. A cultura das pessoas está sempre presente, inseridas no *mindset* das pessoas, mesmo que indiretamente. Não adianta aplicar o DT

se a visão hierárquica é totalmente vertical, isto é, querendo ou não, a cultura influencia sempre em qualquer passo que se deve dar.

Com a resposta destas 3 questões, fecha-se o segundo bloco da *survey*, podendo dessa forma consolidar as informações para fechar as lacunas encontradas na literatura sobre as questões QP1, QP2 e QP6. Conforme a Figura 4.2, estas questões vieram para responder as seguintes lacunas:

- Entender com profissionais qual o propósito de se usar o DT e relacionar com o grupo de 17 já encontrados: Consolidadas as alternativas para se obter uma relação com os grupos encontrados na literatura, entende-se que os profissionais claramente estão de acordo e próximos aos propósitos trazidos na literatura, com uma mudança em priorização de mais usado. O DT mostrou-se pelos profissionais que possui um propósito muito maior que apenas um relacionamento com os métodos ágeis, na qual foi o tópico mais abordado na literatura. A amostra aponta que o DT é ideal para gerar ideias e soluções, sendo elas inovadoras ou não, explorar melhor o problema para redução das incertezas e dessa forma podendo entender e especificar melhor os requisitos que por consequência do processo aproxima a equipe de desenvolvimento e melhora a satisfação do cliente através de uma entrega mais próxima de atender o real problema, ganhando empatia sobre os envolvidos.
- Compreender se o uso do DT está diretamente ligado com os métodos ágeis ou possui outras relações: Não está diretamente ligado aos métodos ágeis, mas aparenta estar sendo usado como um. Pode até ser citado como um método ágil por alguns mas não necessariamente precisa estar ligado a outros métodos ágeis para mostrar seu valor. As respostas trazidas nos benefícios mostra que o DT possui seu valor único e pode ser usado indiferente se junto ou não com um método. Entretanto percebe-se que, conforme o alto índice de respostas mostrados na Tabela 4.4, existe de certa forma uma integração forte entre os métodos ágeis e o DT e está presente no cotidiano dos profissionais de desenvolvimento de software, onde mais de 100 respostas apontam cenários integrados aos métodos ágeis (por exemplo, *Lean Software Development*, Scrum). Esse valor indica que praticamente 72% dos respondentes desenvolvem o DT unindo com os métodos ágeis.
- Entender quais são os benefícios e dificuldades que os profissionais estão tendo na execução do DT: As respostas foram bem esclarecidas sobre as questões dos benefícios e dificuldades. Em resumo, em benefício o DT traz a oportunidade de unir diversas áreas com interesse comum para validar diversas vezes uma solução sobre o real problema do usuário ou do cliente. Com isso permite através de uma estrutura que se erre várias vezes sem gastar muito construindo algo que pode não ser necessário ou que não atenda de verdade. Por outro lado, no momento que se une diversas pessoas traz a tona a dificuldade de ter tempo, aceitação, mostrar o valor de

retorno em executar o DT e transformar diversos dados qualitativos em algo tangível para quem precisa entender e validar. Como tudo também, a forte negação por causa da cultura inserida no contexto pode não levar o DT para um bom caminho caso o profissional que conduz não souber mostrar o verdadeiro valor e executar corretamente as técnicas que compõem o DT, ou seja, saber executar e qual executar estimula a forma que o DT será caracterizado pelo ambiente. Logo, identifica-se que o usuário é definido como o centro das atenções, sendo a equipe de desenvolvimento de software por atender às necessidades desse usuário, mostrando que o desenvolvimento de software utiliza o DT como uma abordagem centrada no usuário.

Entendendo todas questões trazidas na literatura e buscando através desta amostra um resultado mais próximo de como o DT é visto por profissionais dentro da área da Engenharia de Software. Consegue-se assim consolidar estes pensamentos em um resultado para a questão principal deste trabalho. O Capítulo 5 discute os resultados dos dois estudos (MSL e *Survey* a fim de caracterizar o uso do DT sobre o contexto da Engenharia de Software.

5. DISCUSSÃO

Esse trabalho realizou um MSL e *Survey* para identificar as relações entre o DT e o desenvolvimento de software ou a Engenharia de Software como um todo, buscando responder a questão de pesquisa sobre o DT, verificando onde está sendo usado no desenvolvimento de software e quais são os benefícios e desafios do uso, no geral, procurou-se através de todas as questões conhecer, sintetizar, caracterizar o DT no desenvolvimento de software para possibilitar o crescimento do DT na área e possibilitar a criação de ferramentas que possam auxiliar os profissionais no processo. Para fazer isso, conduziu-se um processo sistemático rígido, incluindo uma pesquisa sobre um mapeamento sistemático e uma *survey* para entender como os profissionais trabalham com o DT.

Esse capítulo apresenta um resumo visando entender como é caracterizado o DT na Engenharia de Software. Também, subsequente a esta, divide-se em seções que discutem as questões de pesquisa sobre o entendimento do propósito e o contexto que o DT está sendo usado (QP1, QP2), quais são os modelos, técnicas e ferramentas que caracterizam o DT (QP3, QP4 e QP5) e quais são os benefícios e desafios para usar o DT no desenvolvimento de software (QP6). Dessa forma, cria-se um corpo de conhecimento sobre o fenômeno estudado. Também ao final é destacado um trabalho correlato encontrado na literatura através do Estudo 1 com objetivo de correlacionar e discutir sobre o trabalho e resultados entre as duas pesquisas.

5.1 Caracterização do DT na Engenharia de Software

Para estimular e embasar essa resposta, destaca-se abaixo o trabalho resumido sobre os dois Estudos.

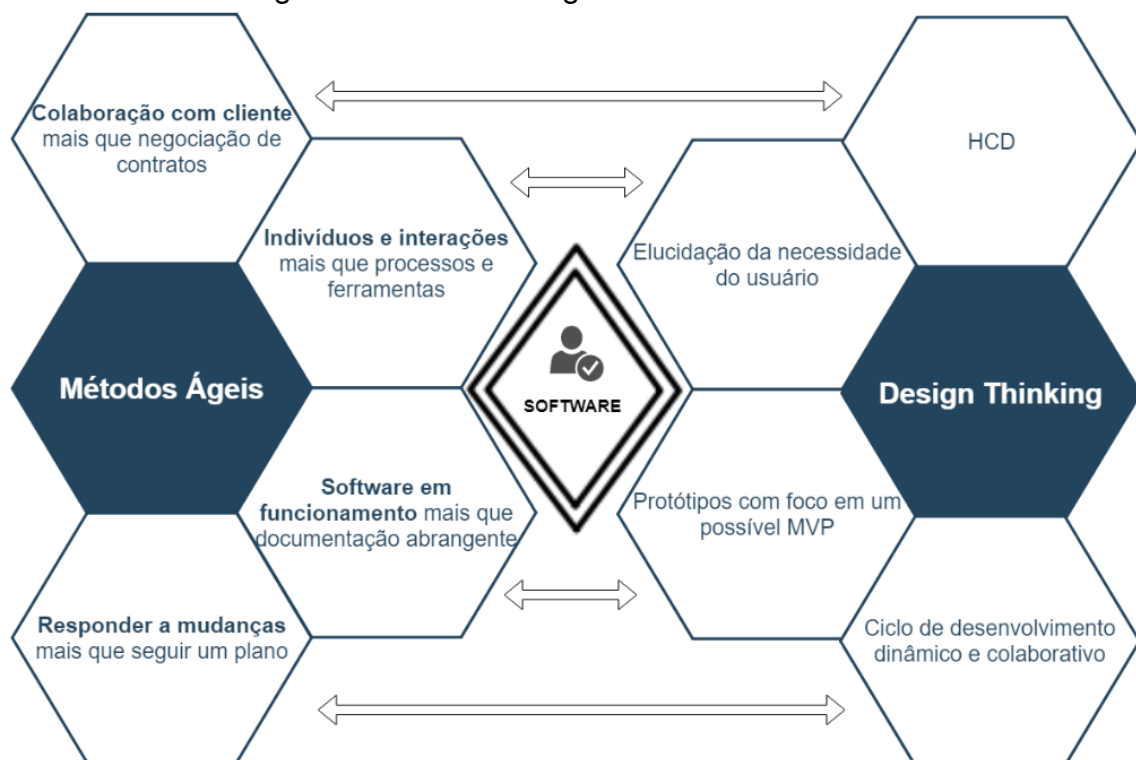
No Estudo 1 (Mapeamento Sistemático), foi realizado um estudo em seis mecanismos diferentes, como o ACM, o IEEE Xplore, o *Science Direct*, o Scopus, o *Springer Database* e o *Wiley Online Library*. Além disso, foi também realizado um processo de seleção completo, cuidadoso e iterativo, extraíndo dados de 78 artigos, de um conjunto inicial de 2588 artigos recuperados. Para garantir a acurácia do acordo, avaliou-se 20% dos trabalhos utilizando o coeficiente de Cohen Kappa, resultando em um índice de 0,85, considerado o mais perfeito.

Além do resultado extraído da literatura, fortalece-se este pensamento sobre DT com a consolidação dos dados capturados no Estudo 2 (*Survey*), da qual preocupou-se em buscar na indústria fatores que caracterizam o DT e para melhor entender como os profissionais estão aplicando sobre esse contexto. Foram enviadas mais de 400 solicitações via plataforma LinkedIn para respondentes que apresentaram características de acordo com o

propósito. Obteve-se 158 respostas das quais foram consolidadas e discutidas no Capítulo anterior. O Estudo 2 trouxe a visão e o conhecimento sobre o DT através do olhar da indústria, fortalecendo o entendimento deste fenômeno nesse trabalho. Com a consolidação destes estudos, criou-se esse resultado da qual abrange um corpo de conhecimento sobre as frentes que juntos representam o DT. São elas: Modelos, técnicas e ferramentas, propósito de uso e contexto inserido, benefícios e dificuldades.

O DT mostrou-se ser uma abordagem benéfica sobre o processo de elucidação dos problemas gerados pela falta de entendimento sobre a real necessidade do usuário. Também percebe-se pelo resumo dos resultados da extração que o DT está sendo utilizado na Engenharia de Software através dos valores de apresentar uma forma adequada para elucidação das necessidades dos usuários e por apresentar conceitos fortes de HCD em sua abordagem. Desta forma também entende-se que existe uma forte relação do que o DT está trazendo para o desenvolvimento de software sobre os valores ágeis, conforme ilustra a Figura 5.1, veja-se algumas semelhanças nos conceitos que cada tópico agrega para a Engenharia de Software.

Figura 5.1 – Valores Ágeis x Características DT



FONTE: Autor, 2019

Estas semelhanças tornam-se expressivas quando se estuda DT. Ressaltar a importância de uma metodologia que, com seus valores, aproxima ainda mais o DT deste mundo, a metodologia ágil, que vem trazendo para o processo de desenvolvimento de soft-

ware através de seus princípios de flexibilidade, agilidade, colaboração e transparência, pontos relevantes que aproxima o usuário e destaca o envolvimento do mesmo como um princípio de sucesso [8]. Realizando um de/para com os valores ágeis [8], percebe-se uma forte correspondência entre os valores ágeis com a natureza do DT e isso traz tal conclusão que o DT pode estar sendo considerado para o universo da Engenharia de Software um *framework* semelhante ao scrum, estando no mesmo grupo e sendo mais uma peça do ágil, mas cada um possui propósitos diferentes.

Gurusamy, Srinivasaraghavan e Adikari [48], trazem em seu artigo semelhanças sobre o DT e os métodos ágeis no que refere-se na compreensão dos problemas, onde são difíceis de serem descritos. Como demais artigos encontrados que também abordam esta semelhança, neste artigo entende-se que aproximação da solução sobre o problema, através de interações mais próximas com as pessoas envolvidas, trocando conhecimentos e opiniões, fortifica o entendimento do contexto e do real problema. Também, por se tratar da mesma forma que os métodos ágeis, a busca de *feedbacks* tornou-se uma forte semelhança por ambas abordagens por apresentarem um desenvolvimento iterativo e incremental. Desta forma, projetos desenvolvidos por estas iterações com abordagens ágeis podem gerar uma oportunidade dos usuários de fornecerem *feedbacks* para ajudar a refinar os requisitos. Porém, integrados com DT, as iterações ágeis podem contribuir para fornecer entregas contínuas e constantes aos clientes.

A questão mais elaborada no mapeamento foi a leitura e tradução de cada modelo encontrado para entendimento do contexto inserido, visto a diversidade de modelos que representaram o uso de DT. Sobre estes o modelo mais citado foi da *Stanford University's D-School*, composto por um método seguido por etapas de Empatia, Definição, Ideação, Prototipação e Teste. Diversos resultados trazidos nesse trabalho apontam o termo etapas, outros fases e espaços de trabalho. Utilizar-se-á nessa discussão os termos fases/etapas para referenciar as etapas do DT. Os modelos possuem etapas diferentes mas acabam seguindo a mesma linha de raciocínio para a execução do DT no contexto inserido. Muitos modelos são adaptados para atender a realidade do contexto e outros seguem alguns modelos já conhecidos, como modelo do Brow, *Stanford University's D-School* ou HPI. Entretanto cabe a dúvida da quantidade de modelos diferentes que aparentam na sua filosofia trazer o mesmo objetivo. Neste caso ressalta-se uma possível necessidade de avaliar individualmente cada modelo e aplicá-lo em diversas situações. Entende-se no geral que muitos modelos foram adaptados para atender a realidade, na mesma forma que hoje é visto os métodos ágeis.

Outro ponto em destaque são as ferramentas. Diversas técnicas foram encontradas e utilizadas em diversas etapas do DT. Como analisado, 46 técnicas foram encontradas e descritas para o entendimento no Estudo 1 e mais 13 técnicas foram citadas no Estudo 2 pelos profissionais, mas pouco surgiu neste mapeamento o uso de ferramentas que apoiam a aplicação destas técnicas com viés em melhorar ou fortalecer o entendimento da técnicas

ou mesmo de uma própria etapa de um método. Conforme retornado no Estudo 2, muito se preocupa na utilização manual das técnicas do que apoio sobre alguma plataforma de software, defendendo a forma de cocriar e participação entre os envolventes. Mas também, desta forma pode-se concluir com estes resultados a carência de ferramentas na área para apoiar a execução do DT, não necessariamente aplicados durante o processo do DT, mas sim auxiliando os profissionais no uso das técnicas sobre diversas perspectivas e exemplos.

Fortalecendo o entendimento do DT, analisa-se os resultados obtidos no Estudo 2, composto por respostas vindas da indústria através de profissionais. A *survey* preencheu o espaço que o mapeamento trouxe para pesquisa, não apenas entendendo como realmente o DT se comporta dentro da área da Engenharia de Software, mas abrindo portas para entender onde o DT pode ser melhorado dentro do processo de desenvolvimento e quais são os tópicos menos citados que podem vir à abrir portas para criação de novas frentes do DT para o desenvolvimento de software.

O DT discutido na *survey* fortaleceu diversos fatores, como: os modelos encontrados no mapeamento são discutidos e trabalhados na indústria, além de serem inspiração para muitos profissionais, possuem suas etapas alocadas em espaços abstratos do DT, que são trabalhados em todos os modelos (espaço divergente e convergente). Além de, dentro de cada espaço, discutam-se tópicos para a descoberta do problema e o surgimento de uma solução, que muitas vezes é praticada com a influência cocriativa e inovadora, tornando o produto entregue muitas vezes mais do que atender o requisito e o real problema do usuário, e sim a criação de um produto inovador.

Em resumo, com a consolidação dos dados obtidos, caracteriza-se o DT no desenvolvimento de software da seguinte forma: O *Design Thinking* é um *framework* que diligencia um processo de descoberta do problema do usuário através da aproximação de todas as partes interessadas do projeto para a criação de um software em comum, de valor e inovador, dentro de um ambiente cocriativo, da qual seguem um conjunto de fases iterativas sobre espaços de trabalhos que divergem, para exploração do problema, e convergem em prol da solução.

Para entender o motivo do DT ser definido como um *framework* para o desenvolvimento de software, realizou-se um pensamento relacionando o DT com o scrum, junto com todo o estudo trabalhado através do mapeamento e a *survey*. Seguindo o conceito do *Scrum Guide* [110], o scrum não é um processo, técnica ou método definitivo, pelo contrário, é um *framework* na qual emprega-se vários processos e técnicas para produzir de forma criativa produtos de maior valor possível. O scrum baseia-se na teoria empírica, ou seja, o conhecimento vem da experiência e da tomada de decisões com base no que já é conhecido, abordando conceitos iterativos e incremental com objetivo de otimizar a previsibilidade e controlar o risco [110].

O scrum é caracterizado como um *framework*. Possui valores, um grupo de pessoas, conhecido como *time*, que atuam dentro desta estrutura, possui pilares que sustentam

sua teoria, seus eventos e artefatos [110]. Porém, em nenhum momento o scrum é apresentado para o desenvolvimento de software como um método seguido com um manual de uso para mostrar como deve ser executado. Percebe-se então que o DT está inserido na mesma linha de raciocínio que o scrum é trabalhado no contexto da Engenharia de Software. Porém não pode-se cair na armadilha que o DT é uma método ágil. O DT de fato possui muitas semelhanças e vem com o mesmo propósito que a gestão ágil está inserida, ou seja, prevenção de risco, uma pré-validação de software com usuário, conhecendo o problema abordado para melhor traduzi-lo em uma solução de software. A ES já possui o costume de inserir os conceitos, as atividades, os métodos em caixas dentro de um processo maior de desenvolvimento de software, por essa maneira veja-se que o DT está sendo considerado por alguns profissionais um método ágil, mas não é. O DT é a forma que o *designer* trabalha para compreender o fenômeno, o problema. Na ES não é muito diferente, pois é a forma que se trabalha para entender o usuário, seu processo, seu problema e tirar o melhor proveito do entendimento do processo do usuário, das atividades, de suas necessidades e por esse motivo que o DT está tão ligado com conceitos de UCD e métodos ágeis. Pelo fato de ser uma forma de trabalho para compreender o usuário, o DT possui uma imagem de *framework* pelo fato também de ser planejado de acordo com o pensamento abduutivo daquele que irá conduzir, ou seja, o DT é planejado dentro do cenário inserido e é executado de acordo com a experiência a bagagem que o *designer* possui para aplicar o processo de DT da melhor forma que o profissional é capaz. Para entender onde o DT está sendo aplicado, a seguir a seção que trata dos propósitos de uso e cenários aplicados do DT.

5.2 Entendendo o Propósito de Uso e Diferentes Cenários (QP1 e QP2)

O DT em sua essência busca a exploração do problema e estuda uma solução para este. Visando o contexto da Engenharia de Software, o DT possui o propósito de elucidar e facilitar o levantamento de requisitos no início do desenvolvimento de software. Para Sohaib et al. [105], o DT busca *feedbacks* dos clientes por meio de protótipos, viabilizando e pré-validando critérios de usabilidade, dessa forma ganhasse uma garantia de entregas rápidas e já pré-validadas pelos usuários. Seguindo estes autores [105], o DT possui semelhanças com *Extreme Programming* (XP), onde focam no usuário, um desenvolvimento iterativo e uma extensa comunicação em equipe. O DT visa explorar radicalmente um novo caminho para solução, visando a experiência do usuário por meio de abordagens ágeis de desenvolvimento de software.

A abordagem centrada no ser humano procurando inovar e buscar a verdadeira solução para o problema do usuário é o maior propósito de uso do DT dentro do desenvolvimento de software. O Estudo 1 trouxe que o DT é melhor aplicado sobre ocasiões que

não se tem o conhecimento do problema, onde o problema não está bem definido. Batista et al [10] apontam que o DT vem para transformar ideias em propostas criativas de soluções através de uma origem do *designer* para suprir as necessidades das pessoas e que possui etapas que seguem uma sequência sobre o problema, a forma de gerar, desenvolver as ideias para alcançar uma solução que pode ser representada como um produto/serviço e para que esse processo ocorra, o DT disponibiliza um conjunto de técnicas para serem aplicadas durante o processo.

O Estudo 2 aponta que, sobre o total de escolha, 14,29% dos profissionais utilizam o DT para gerar ideias e soluções, 13,83% para explorar e entender o problema e 11,43% para criar ideias inovadoras. De fato o DT mostrou que esses são de fato sua essência e sobre uma visão maior é para isso que é utilizado. Porém, dentro do contexto do desenvolvimento de software, o DT pode vir como uma forma de reduzir as incertezas (10,68% apontam), outros 10,38% selecionaram que é para entender e especificar os requisitos, 9,02% para melhorar a satisfação do cliente e 8,12% para aproximar a equipe de desenvolvimento com o cliente, ou seja, de forma macro o DT possui o propósito de uso na exploração do problema para gerar ideias e soluções, muitas vezes inovadoras, mas abaixo disso percebe-se que é usando para redução das incertezas que surgem pela falta de entendimento do real problema e, por consequência, aproximando o entendimento que o DT se preocupa com o usuário, acaba melhorando a satisfação do cliente e aproximando a equipe desenvolvimento com o cliente/usuário. Também com índice menor (6,77% e 4,81%) apontam que ganhar a empatia dos usuários e validar o software, respectivamente, são formas de se usar o DT. No geral, se contar pela quantidade de profissionais que responderam com a quantidade de seleção para cada alternativa, percebe que de 158 profissionais, 32 utilizam para validar o software, 45 para ganhar empatia dos usuários, 60 para melhorar a satisfação do cliente e 69 para entender e especificar os requisitos, lembrando que é uma questão de múltipla seleção, podendo o mesmo optar por mais de uma.

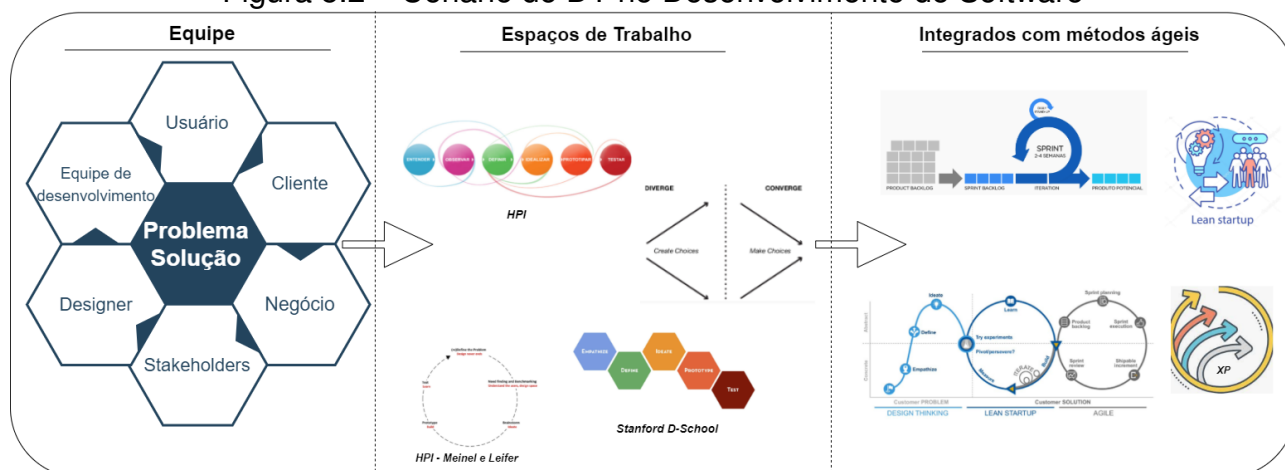
O Estudo 2 também retorna algumas respostas descritivas, que abordam o uso do DT da seguintes forma: "*Para mapear a experiência do cliente ou de um funcionário em suas jornadas ou em treinamentos de mudanças de comportamento.*", "*Para planejamento estratégica, problemas industriais, problemas complexos, avaliação, ajustes e melhoria de processos.*"ou "*Criação de melhoria de processos internos*".

Sobre os cenários que o DT está sendo usado, o Estudo 1 mostra uma grande relação do DT unido com outros métodos e *frameworks*, como por exemplo, o Lean, Lean UX, Lean Startup, Scrum, XP, Kanban, Agile UX. Além da forte presença do DT em conjunto com métodos ágeis, também trouxe em cenários da Engenharia de Requisitos. Em todas situações, o DT mostrou-se ser aplicado sobre uma equipe multidisciplinar, ou seja, o DT aplicado apenas com um profissional e o usuário não existe, pois é necessário uma visão sobre várias perspectivas diferentes no mesmo contexto. O DT no Estudo 2 apontam que 90 respondentes sobre 158 usam o DT em um cenário com equipes multidisciplinares, ou seja, a

ideia trazida na literatura converge com a seleção dos profissionais. Também destacam-se que o DT está em um cenário para criar produtos/software inovadores e que, 72 respondentes, trouxeram que o DT está presente pelos dentro do processo de desenvolvimento de software diariamente/semanalmente e 66 o utilizam em parceria com os métodos ágeis (Lean, Scrum). Em resumo, o DT é um parceiro dos métodos ágeis e está sendo visto pela comunidade da Engenharia de Software como um dos seus métodos/*frameworks* para aproximar o cliente/usuário para o processo de desenvolvimento.

Concluí-se que o DT é seguido por uma equipe multidisciplinar, que seguem espaços de trabalho com intuito de divergir e convergir suas ideias para entender o problema e gerar uma solução que muitas vezes serve como insumo para métodos ágeis já praticados pela indústria de desenvolvimento de software. A Figura 5.2 ilustra esta ideia.

Figura 5.2 – Cenário do DT no Desenvolvimento de Software



FONTE: Autor, 2020

5.3 Modelos, Técnicas e Ferramentas do DT (QP3, QP4 e QP5)

O Estudo 1, sobre mapeamento sistemático na literatura e o Estudo 2 na qual buscou na indústria através de uma *survey*, trouxeram uma imensa quantidade variáveis de modelos, técnicas e ferramentas. Isso mostra como o DT é adaptado e usado em diferentes formas. Porém, buscou-se uma relação sobre todos os modelos encontrados a fim de relatar uma semelhança e buscar a essência da execução do DT. Essa seção é caracterizada pelo resumo e princípios encontrados nos modelos.

O DT é baseado em princípios específicos que diferenciam a abordagem de outros processos criativos e a tornam particularmente útil na TI [15]. Estes princípios são [12]:

1. **Necessidade dos usuários ocupam o centro do processo:** As necessidades dos usuários são colocadas no centro das atenções e o design é constantemente visto da perspectiva de futuros usuários, sem a necessidade de envolvimento direto do usuário;
2. **Mais compreensão em vez de um grande número de casos:** A DT visa descobrir informações interessantes e surpreender, tomá-las como ponto de partida para desenvolver ideias inovadoras em vez de identificar aspectos que são igualmente comuns à maioria dos usuários;
3. **Equipes interdisciplinar:** Idealmente recrutados de diferentes áreas para criar uma diversidade de perspectivas, os membros devem poder apreciar as opiniões e perspectivas de outras pessoas e tentar ousar novas coisas;
4. **Seguir um processo claro:** O processo de design é melhor descrito metaforicamente como um sistema de espaços que demarcar diferentes tipos de atividades relacionadas que juntas formar o continuum de inovação, em vez de uma predefinição série de etapas ordenadas.

A representação mais abstrata do processo de design é a Divergência e Convergência, conhecidas como espaços [59]. O espaço Divergente é baseado na diversidade e esclarece um problema sob diferentes perspectivas e os o espaço de convergente leva à consolidação e fusão de ideias até a solução.

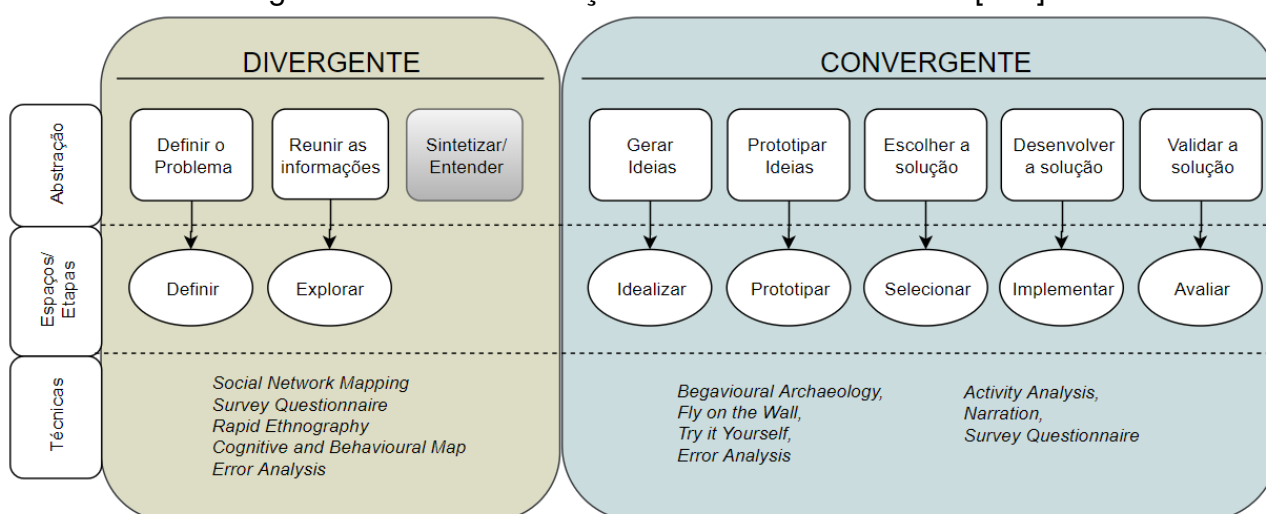
Desde o início do DT, diversos modelos foram surgindo. Para cada modelo encontrado neste trabalho e apresentando durante o Estudo 1 e Estudo 2, foi realizada uma análise para entender de forma macro o que cada espaço de trabalho do DT propõe a fazer e possuem em comum, caracterizando desta forma o respectivo modelo DT, com as suas fases e técnicas utilizadas, agrupados com a ideia de divergência e convergência [59]. Percebe-se que o espaço divergente é principalmente organizado em três atividades principais: configurar o problema, reunir informações sobre ele e sintetizar o entendimento, definido de maneira semelhante em cada estudo e modelo a seguir. O espaço convergente [59], no entanto, varia mais em suas respectivas fases e no ciclo para obter uma solução. Por exemplo, alguns modelos primeiro criam ideias de protótipo para posteriormente validá-las, enquanto outros definem um produto em funcionamento [41] [15]. Igualmente, todos os modelos sugerem três atividades principais: gerar ideias, prototipar e validar as ideias, independente do nível de abstração para representar a solução. Também percebe-se que as fases de prototipação, desenvolvimento e validação são muito mais recorrentes para ideias que resultam em um produto palpável de forma geral (nesse contexto, em um software) e as demais são mais destacadas para melhorias em processo.

Sandino et al. Sandino et al. [101] propõem uma metodologia de DT para o desenvolvimento de aplicativos interativos em tempo real. Começa com a definição de uma série de

restrições (por exemplo, quão fielmente um simulador deve refletir a realidade), que guiará o trabalho subsequente (fase Definir). Em seguida, a equipe reúne informações sobre usuários em potencial, suas necessidades e soluções anteriores para o mesmo problema (Explorar) e, posteriormente, identifica problemas de forma colaborativa e gera o maior número possível de ideias para obter respostas para as necessidades identificadas (Idealizar).

A seguir, as ideias são prototipadas e refinadas por meio de discussões iterativas (Prototipar). Depois de escolher a solução candidata que pode resolver o problema em questão (Selecionar), a equipe o implementa (Implementar) e monitora se realmente se encaixa nos propósitos para os quais foi concebido. Ao final o produto é apresentado aos usuários e identificam-se possíveis áreas de melhoria (Revisão). A Figura 5.3 mostra as etapas abstraídas do modelo divergente e convergente com as etapas existentes no modelo do Sandino et al. [101], seguido pelas técnicas utilizadas em cada espaço. A linha de abstração escura representa que o modelo não a utiliza sobre as suas etapas. A Figura 5.3 apresenta este modelo.

Figura 5.3 – Caracterização: Modelo Sandino et al. [101]

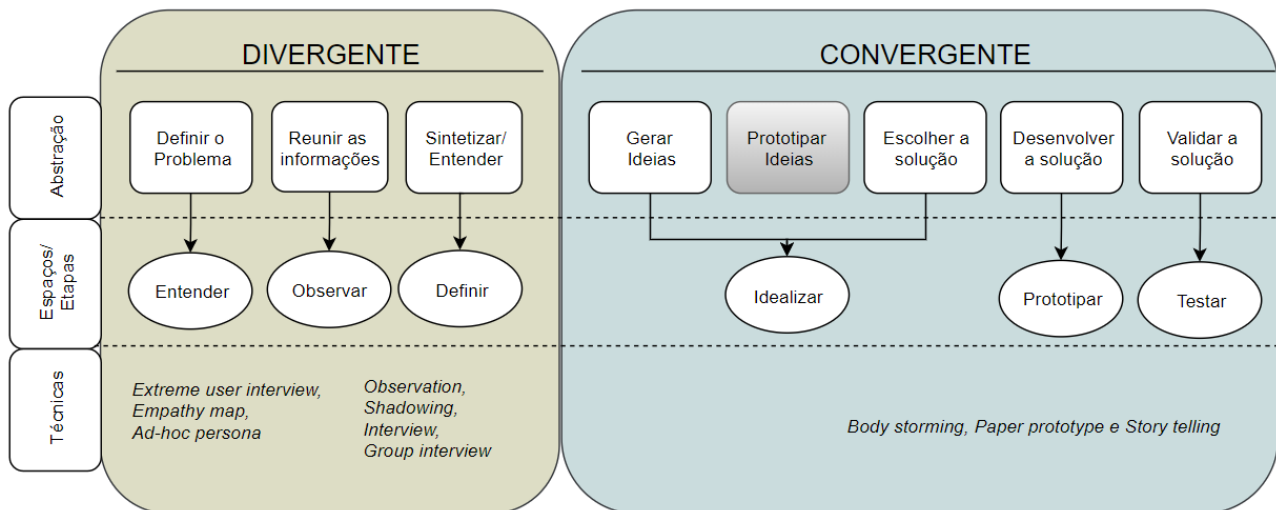


FONTE: Autor, 2020

Hasso-Plattner Institute (HPI). Aplicado ao design de interfaces de usuário de pesquisa [9], o modelo de processo HPI é iterativo e oferece ciclos que levam os participantes a seis fases [49]: Entenda, a equipe define o espaço do problema; Observe, a equipe ganha uma visão externa e simpatiza com os usuários e as partes interessadas; Definir o ponto de vista, serve para definir o ponto de vista e em que o conhecimento adquirido será reunido e resumido, e o desafio reformulado; Idealize, a equipe posteriormente gera uma variedade de possibilidades de solução, depois seleciona um foco; Protótipo, serve no desenvolvimento de soluções concretas que podem ser testadas no grupo-alvo apropriado (fase de teste).

O HPI também foi utilizado para a definição do InnoDev [31], uma abordagem de desenvolvimento de software que integra DT, Lean Startup e Scrum em um único modelo, com o objetivo de superar as limitações do desenvolvimento ágil, como falta de compreensão de problemas e localização de soluções; e como modelo de processo de referência integrado ao ágil para o desenvolvimento de sistemas *cyber-physical* [80]. A Figura 5.4 exibe o modelo com a sua essência.

Figura 5.4 – Caracterização: Modelo HPI



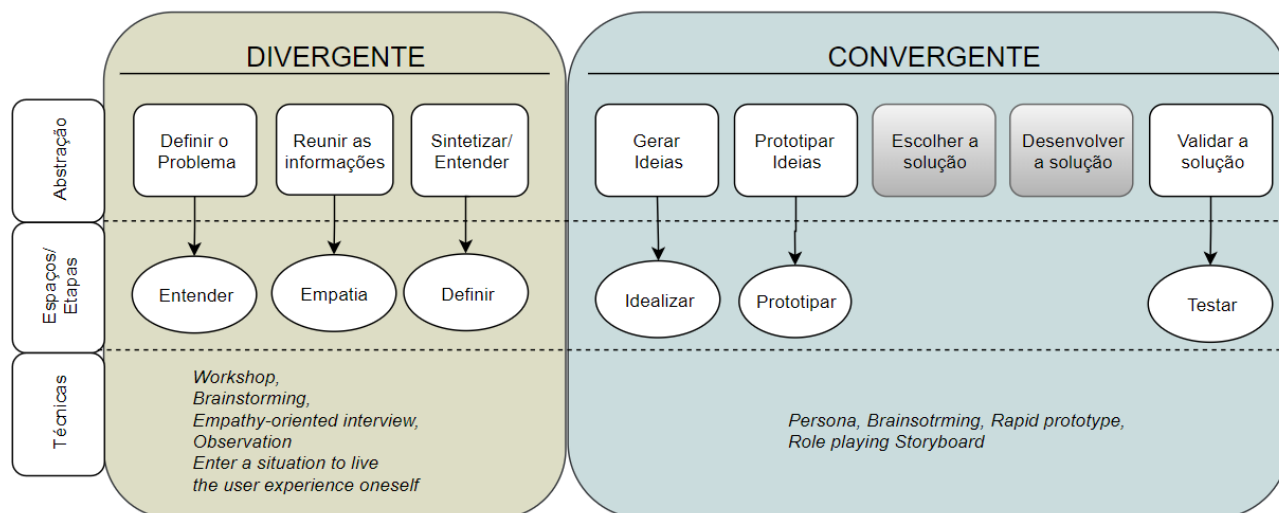
FONTE: Autor, 2020

Stanford University's D.School. O modelo *D.School* visa esclarecer o problema a ser resolvido através da empatia. Isso permite uma imersão no contexto dos usuários, a fim de obter detalhes para apoiar o entendimento das necessidades dos usuários. O modelo [109] é composto por cinco fases: Empatia, refere-se à observação e visualização de usuários e seu comportamento no contexto de suas vidas, interagindo com eles e vivendo suas experiências; Definir, refere-se a descompactar e sintetizar as descobertas da empatia em necessidades e *insights* convincentes e definir um escopo específico desafio; Idealizar, refere-se à geração de design radical alternativas, enquanto Prototipar é a fase em que as ideias são transformadas em artefatos físicos para obter feedback de usuários e soluções de refino na fase Teste.

O modelo *D.School* foi utilizado com uma fase inicial adicional denominada Entender, com o objetivo de entender o problema proposto, para apoiar a engenharia de requisitos de projetos de transformação digital [15]; mesclando suas duas primeiras fases originais de Empatia e Definição em uma única fase chamada Pesquisa como um meio de apoiar o design de sistemas embarcados [4]; e mesclando suas duas últimas fases originais, Protótipo e Teste em uma única fase para aumentar o consumo de dados abertos no Quênia [85]. Também foi usado com novos rótulos para as três primeiras fases originais - (re) Definir o Problema, Necessidade de Encontrar e Benchmarking, e *Brainstorm*, para promover o de-

envolvimento de equipes de TI de alto desempenho [58]. A Figura 5.5 apresenta o modelo e suas ligações.

Figura 5.5 – Caracterização: Modelo *Stanford University's D.School*

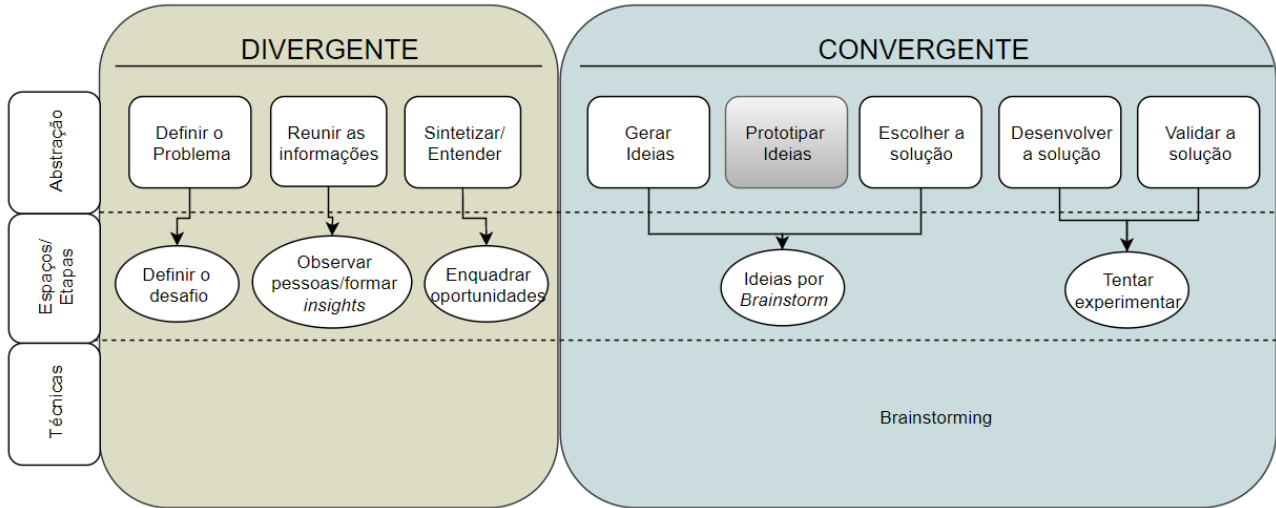


FONTE: Autor, 2020

Modelo Nordstrom. Um modelo para o uso integrado de DT, Lean Startup e *Agile development* [45]. A equipe multidisciplinar de desenvolvedores e designers reúne informações sobre o desafio a ser resolvido e define a sequência de atividades para buscar a solução (*Define the Challenge*). Em seguida, a equipe registra o que as partes interessadas fazem e pensam (*Observe People*), a fim de gerar *insights* para conduzir a discussão de possíveis soluções (*Form Insights*) e cria uma representação visual para entender os dados coletados, destacando os principais relacionamentos e desenvolvendo a estratégia da solução (*Frame Opportunities*). Esse entendimento é usado para debater ideias de soluções (*Brainstorm Ideas*) e testar a solução escolhida em condições reais para aprender como ela funciona na prática (experimentos experimentais). A Figura 5.6 apresente o modelo consolidado com os espaços de trabalho.

Diving Board. O modelo *Diving Board* [87] representa a incorporação da DT em uma estrutura participativa para resolver problemas sociais complexos pouco compreendidos. Ao definir um ciclo iterativo de DT, organizado em quatro fases - Desenvolver, Apresentar e Provocar, Explorar e Refletir, precedido por uma fase de Abordagem, na qual as partes interessadas e a equipe compartilham informações sobre o contexto do problema e as experiências do usuário, seguidas por uma fase de Escape, que permite aos participantes concluir que o problema foi suficientemente explorado e que a lista de requisitos elicitados realmente reflete na solução para o problema identificado, este modelo oferece uma mecanismo poderoso para a geração e desenvolvimento de soluções de software para domínios com problemas subespecificados e abertos. As fases do ciclo visam definir pro-

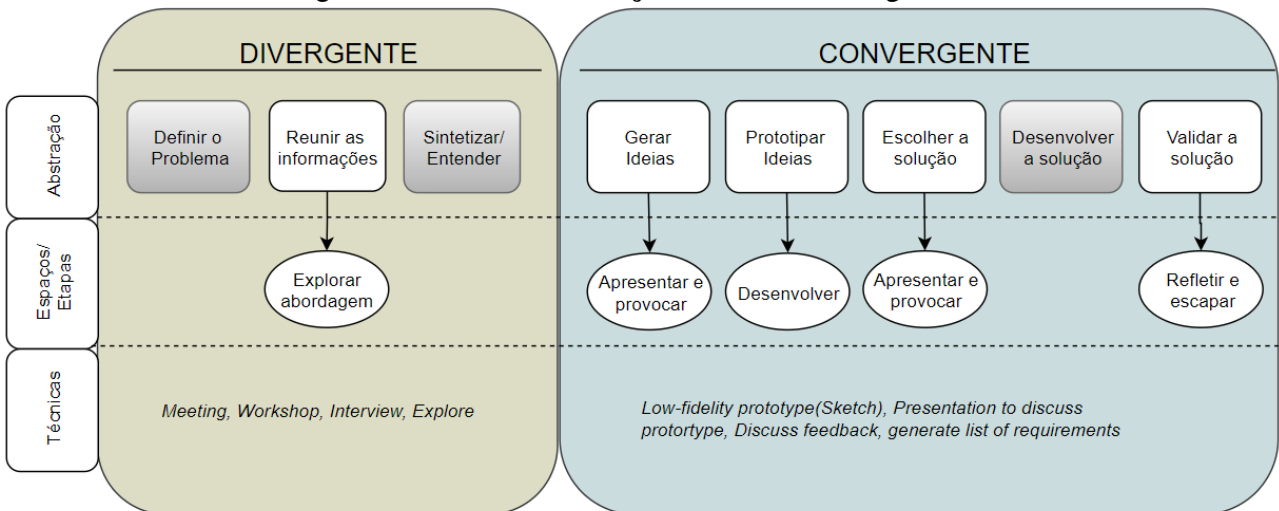
Figura 5.6 – Caracterização: Modelo Nordstrom



FONTE: Autor, 2020

tótipos de baixa fidelidade para dar uma amostra do que poderia ser usado para explorar as ideias discutidas (Desenvolver); apresentar versões mais recentes de protótipos para incentivar mais ideias através do feedback dos participantes (Apresentar e Provocar); use os protótipos como objetos de limite para ajudar a focar o espaço do problema e fornecer um escopo tecnológico com o objetivo de facilitar a exploração adicional do problema (Explorar); e promover a reflexão sobre o feedback fornecido, com o objetivo de melhorar ou propor novos protótipos até que a solução seja alcançado (Refletir). A Figura 5.7 mostra as etapas do modelo em seus espaços de trabalho.

Figura 5.7 – Caracterização: Modelo *Diving Board*

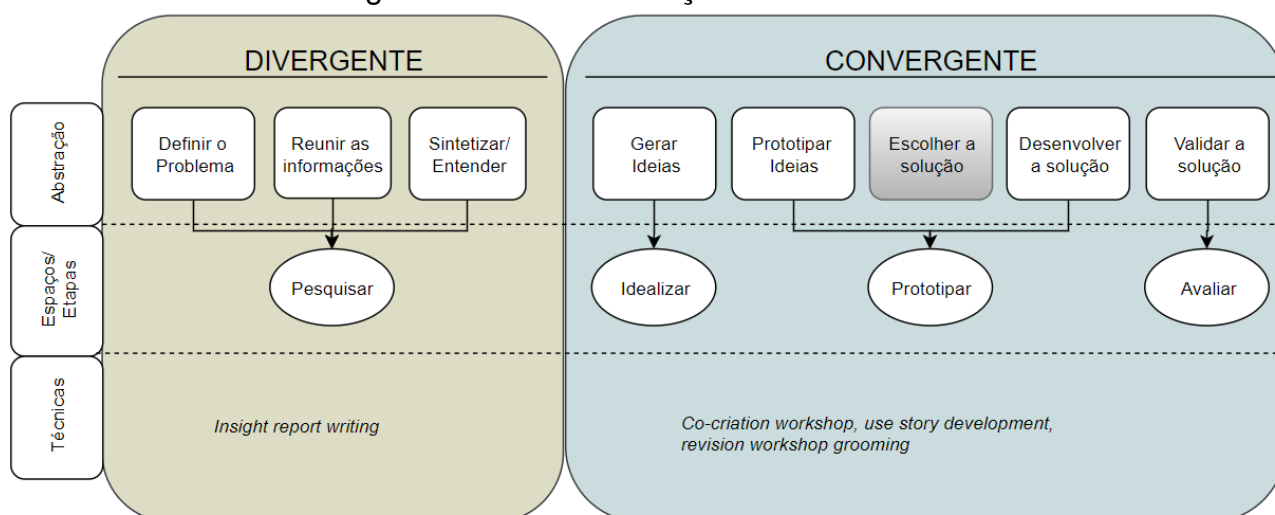


FONTE: Autor, 2020

Human-Centered Agile Workflow (HCAW). O HCAW combina o Scrum com o design centrado no usuário para a concepção de novos produtos de software [41]. Uma concepção inicial do ciclo é proposta, organizada em uma fase de pesquisa que começa por receber da equipe de negócios um cenário de problema usado pela equipe para conduzir a pesquisa para entender melhor o problema. Esse entendimento é registrado em um Relatório de *Insight*, que é a base para fase de Ideação, na qual um Workshop de Co-Criação é hospedado para gerar ideias posteriormente prototipadas na fase de prototipagem e avaliadas na fase de avaliação. O primeiro ciclo abrange todas as fases, enquanto os outros ciclos retornam da fase de ideação ou prototipagem, conforme necessário, e os resultados são usados como entrada para codificar um produto de software final.

Esse modelo utiliza o termo estágio para referências as etapas do DT. O DT discutido por [41] aborda que o DT é semelhante ao HCD, com alguns desvios estruturais: HCD combina duas atividades do DT, "Compreender" e "Observar" em uma, deixando aberto ao contexto do projeto para escolha a escolha de um determinado tipo de método. O DT define claramente esta etapa, a "Observação", da qual realiza uma validação de premissas e obtenções de mais informações. A leitura da etapa "Ponto de Vista" é uma tarefa menos burocrática de definir um requisito de uma solução do que as atividades da especificação no HCD. Esse modelo apresenta uma forma de introduzir um fluxo de trabalho ágil centrado no ser humano através de uma concepção pelos conhecimentos trazidos do DT para realizar uma forma de processo centrada no cliente e no desenvolvimento ágil [41]. A Figura 5.8 apresenta o modelo caracterizado pelos espaços divergentes e convergentes.

Figura 5.8 – Caracterização: Modelo HCAW



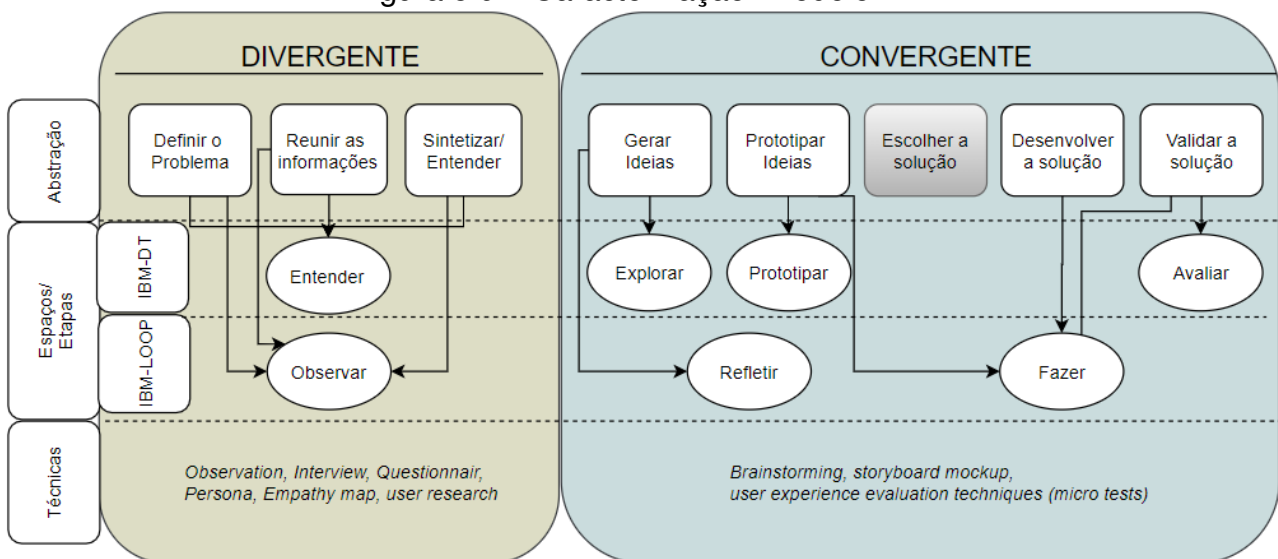
FONTE: Autor, 2020

IBM Design Thinking. O IBM DT fornece uma estrutura para escrever requisitos ou organizar equipes e acompanhar o progresso do projeto, incluindo feedback constante do usuário final [78]. Essa nova metodologia ágil é organizado em quatro fase - Entenda, estabelece

empatia com os usuários para ajudar a entender o contexto deles e de seus problemas, seguido pela definição de uma declaração de problema acionável; Explorar, concentra-se na geração de novas ideias inovadoras; Protótipo, gera artefatos destinados a responder perguntas para resolver o problema; e Avaliar, solicita aos usuários feedback sobre o protótipo criado. A metodologia introduz o conceito *Hill*, que representa uma nova maneira de expressar as necessidades do usuário nos requisitos do projeto. Cada *Hill* articula um objetivo claro e contém uma definição escopo a ser alcançado em um *release* e deve ser escrito para resolver um problema específico e claramente definido do usuário. Os *hills* também descrevem interseções entre as expectativas do usuário e os requisitos de negócios e consistem em três elementos: "Quem": descreve um usuário específico; "O quê": descreve um problema que precisa ser resolvido; e "Como": um mensurável alvo resultante da conclusão da *hill* [78].

Pelo resultado da *survey*, foi apresentada uma nova forma de ver o modelo da IBM, na qual denomina-se como modelo *LOOP* [53]. A Figura 5.9 mostra a visão dos dois modelos, o trazido no Estudo 1 e o *IBM-LOOP* descoberto no Estudo 2.

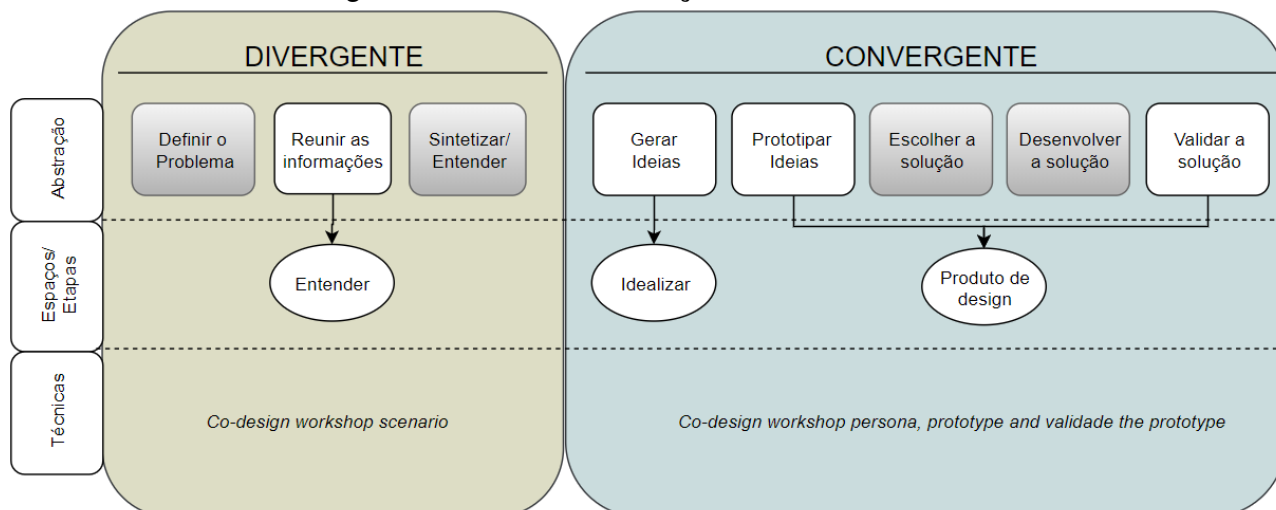
Figura 5.9 – Caracterização: Modelo IBM



FONTE: Autor, 2020

CoDICE: Codesigning Digital Cultural Encounters. Contribuição de um projeto conjunto europeu que visa co-projetar objetos inteligentes para encontros aprimorados com o patrimônio cultural, o processo de co-design do CoDICE [34] é organizado em três espaços de trabalho: Recursos ou Reunião de Recursos Situados, o espaço onde os co-designers podem coletar material útil ou inspirador ao visitar o ambiente físico em que o artefato digital será implantado; Ideias, ou Inquérito divergente e Ideação, destinadas a gerar ideias; e produtos de design ou convergentes Design, focado na produção de soluções para ideias selecionadas. A Figura 5.10 mostra o modelo trabalhado e ligado em sua abstração.

Figura 5.10 – Caracterização: Modelo CoDICE



FONTE: Autor, 2020

Finalizando o entendimento dos modelos e conhecendo que todos possuem um fluxo divergente sobre a concepção do problema e convergente para criação da solução, e possuem ligações na abstração de suas essências, aprende-se também que para a execução de cada etapa dos modelos, o DT utiliza um conjunto de técnicas que muitas vezes possuem um suporte de softwares (ferramentas) para apoio sobre o desenvolvimento da técnica.

A maioria das fases nas quais as técnicas foram citadas foram suportadas pelo uso de várias opções (por exemplo, Sandino et al. [101], coletando informações na fase Explorar). Além disso, várias técnicas foram usadas em várias fases no mesmo estudo, divergindo no espaço do problema (por exemplo, Observações, mapa de empatia [78]) ou convergindo para a definição de uma solução (por exemplo, *Bodystorming* [9]) ou ambos (por exemplo, *Survey* [101]). Também existem técnicas relatadas em estudos e modelos para um único objetivo (por exemplo, cenário para coleta de informações [87], [34]) ou para métodos distintos (Entrevista para coleta de informações [15] ou validação de uma solução [31]).

De acordo com Brenner et al. [11], as técnicas de DT não podem ser facilmente atribuídas a trabalhos individuais. Um profissional experiente usa do repositório aqueles adequados à situação.

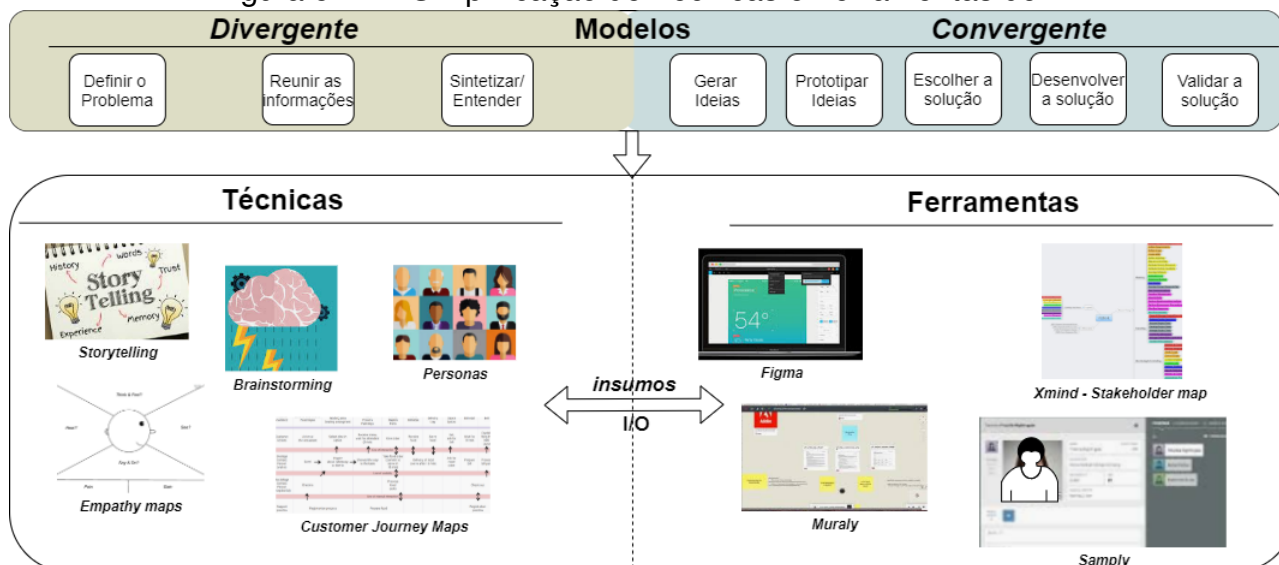
Conforme mostrado no Estudo 2, a escolha das técnicas está muito mais direcionada pela necessidade do profissional que irá executá-la, de acordo com o contexto que está inserido, baseado em uma experiência prévia e relacionada ao espaço de trabalho do modelo em si, ou seja, a capacitação dos profissionais em executar as técnicas provoca a forma que o DT será executado sobre seus espaços de trabalho. Sobre os 171 respondentes do bloco 1 no Estudo 2, a metade aponta uma dificuldade acima de 5 de 10 níveis de

dificuldade e outra metade menos que 5, obtendo uma média de 4.50 o grau de dificuldade. Sobre 10 níveis disponibilizados, acredita-se que os profissionais não possuem um nível de dificuldade mas é um resultado mediano, visto que 33,33% apontaram acima de 5, ou seja, 1/3 dos profissionais possuem dificuldade, 18,13% selecionaram a dificuldade 5 e 48,54% não possuem tanta dificuldade.

As técnicas são a forma de executar as etapas de cada modelo do DT, são a forma de absorver as necessidades do usuário, a empatia de toda equipe, a criação de protótipos que possam ser validados e palpáveis para o usuário final. Os modelos possuem suas abstrações, onde cada um possui um conjunto de etapas ou espaços de trabalho, que são executados por diferentes técnicas. As técnicas mais citadas no Estudo 2 mostram que os profissionais utilizam muito mais técnicas empáticas, ou seja, que visam procurar entender o usuário, conhecer os envolvidos, escutar a opinião de todos da equipe e isso vai de encontro com o objetivo do DT, que prioriza o conceito que o usuário está no centro do processo.

Sobre as ferramentas, existem sim alguns softwares que apoiam o desenvolvimento das técnicas do DT, mas não são obrigatórios para alcançar o objetivo final da execução do modelo. As ferramentas são para o DT um apoio para desenvolver um técnica, analisar melhor dados obtidos de forma quantitativa. O Estudo 2 ajudou a identificar mais ferramentas que podem apoiar o desenvolvimento do DT e trouxe uma questão a ser discutida, visto que muitos respondentes trouxeram a ideia de não utilizar ferramentas para o processo de trabalho do DT pois o mesmo desqualifica a característica que o DT possui de estimular a criatividade e a cocriação entre os participantes. Logo, as ferramentas vem como um apoiador para qualificar o resultado, ajudar a desenvolver um protótipo de alto nível, otimizar os resultados trazidos de alguma técnica, mas não é aconselhável usar ferramentas ou recursos digitais durante uma técnica, visto que o momento da execução da técnica possui o objetivo muito maior do que apenas entregar uma persona criada pelo grupo, mas sim aproximar as pessoas a cocriarem ideias inovadores e soluções para o real problema. A Figura 5.11 exemplifica as técnicas e ferramentas.

Figura 5.11 – Simplificação de Técnicas e Ferramentas do DT



FONTE: Autor, 2020

5.4 Benefícios e Desafios no uso do Design Thinking (QP6)

Identificou-se uma diversidade de benefícios associados ao uso do DT, como: Criar um MVP, elucidar requisitos, fácil integração com métodos ágeis, ou seja, é adaptado para o contexto do desenvolvimento de software ágil, reduz o risco do esforço para o desenvolvimento visto que já realizou um processo de validação através de protótipos, promove soluções sustentáveis, apto para mudanças frequentes, melhora toda a comunicação, não apenas entre a equipe mas sim com todos os participantes e principalmente com o usuário. Burchardt e Maisch [14] trazem que a aproximação do DT com o ambiente ágil proporciona a facilidade do entendimento do problema do usuário, visto que o mesmo encontra-se presente durante as etapas do DT e a presença do designer é importante pois possui uma perspectiva humana sobre as pessoas inseridas no processo do DT.

O DT visto pelos profissionais pelo Estudo 2, compartilha com os envolvidos (TI, *Business*, *Customers*) todos os desafios envolvidos, adequada os processos pela transparência, estimula a inovação tecnológica e aumenta a experiência do usuário, gerando dessa forma um software que agrega valor e atenda as expectativas dos envolvidos. Aumenta a assertividade nas soluções de problemas e entendimento das necessidades. O DT também possui um paradigma, onde o mesmo reduz o tempo de desenvolvimento visto que, quando chega para desenvolver, já se sabe com maior certeza o que é preciso fazer, porém a redução do tempo no desenvolvimento só acontece devido ao tempo gasto executando o processo do DT. O DT é o lado humano do processo na qual sensibiliza os envolvidos com

o problema do usuário, dando dessa forma oportunidades para cocriar dentro da equipe. Além da cocriação o processo de validação do software é mais iterativo e rápido, visto que o usuário já previamente homologou a solução.

O DT é uma forma de engajar a equipe no processo de desenvolvimento e facilitar o consenso sobre as diferentes áreas envolvidas. De certa forma é a maneira de se deixar errar várias vezes no início para acertar no final, ou seja, errar várias vezes até encontrar uma solução viável que apresente maior assertividade para atender o real problema do usuário sem usar muito recurso e custo durante o processo de desenvolvimento. Após entender e conhecer a solução, não se gasta recurso desenvolvendo software que não agregue valor para o usuário final. Entretanto, o DT possui algumas dificuldades e armadilhas, sendo a mais citada no Estudo 2, são as características culturais, ou seja, como mostrar que o processo do DT agregará valor para o usuário e como mostrar para as outras áreas, principalmente à área que possui o recurso, que a realização do DT é benéfica e de valor. O DT no início é difícil mostrar algo palpável e muitas vezes o palpável só se enxerga no final do desenvolvimento de software. Logo deve-se analisar até que ponto o DT é benéfico ou não, visto que o mesmo precisará de tempo entre equipes diferentes, esforço/recurso e mostrar resultado, ou seja, empresas que possuem o *mindset* conservador, provavelmente não aplicarão o DT em seus processos de trabalho.



Também, conheceu-se diversas técnicas e ferramentas para serem aplicadas no DT nesse trabalho. Logo, o desconhecimento da técnica pelos integrantes poderá resultar em um processo falho do DT e não trará valor. Provavelmente por isso muitos dos profissionais no Estudo 2 abordaram a questão que a escolha das técnicas ocorre de acordo com a sua prévia experiência. Sobre a perspectiva de um projeto de software, o DT não possui um roteiro a ser seguido, com as técnicas pré prontas para escolher, com isso corre-se o risco de não escolher a melhor técnica para a execução de alguma das etapas, podendo significar mais tempo de projeto. Em resumo, o DT tira o tempo da incerteza durante o desenvolvimento do software e o utiliza para entender com maior exatidão o real problema do usuário para que se possa criar um software ou aprimorar o processo da melhor forma possível, ou seja, com menos custo, menos tempo e menos retrabalho. Porém esse tempo diminuído no desenvolvimento de software é gasto durante o processo de DT e em alto custo, pois requer conseguir tempo dos participantes de diversas áreas. Não se move apenas a equipe de desenvolvimento para corrigir e melhorar o entendimento, move-se o usuário, o gerente, o designer, a equipe de desenvolvimento e a equipe de negócio. Em suma, deve-se avaliar em que momento o DT deve ser usado e de forma correta. Nesse trabalho defende-se a ideia que o DT é uma abordagem centrada no usuário para facilitar o desenvolvimento de uma solução que seja assertiva e corresponda ao real problema do usuário, mas nem sempre o DT é a melhor opção para ser executado.

O fator psicológico está praticamente 100% presente no DT, visto que a empatia e maturidade das equipes deverão sempre estar a prol e convergindo para o mesmo propó-

sito. Mostrar que o DT é muito mais que um conjunto de técnicas e ferramentas para todos os envolvidos e garantir que todos entendam até o final é difícil. O DT é uma abordagem que deve ser levada a sério e muitas vezes propõe mudanças no dia a dia das pessoas. No caso das mudanças acaba levando a resistência dos participantes em não enxergar o que o DT irá trazer de benefício para o final. Em resumo, a grande dificuldade é evangelizar os processos dentro do *mindset* de todos os envolvidos. No momento que se consegue mostrar o valor do DT, surgem de fato os benefícios, como: Melhor entendimento da jornada do usuário e ajuda no desenvolvimento de soluções diferenciadas para os problemas recorrentes, o ganho da empatia com o cliente e usuário final se torna prazeroso trabalhar e as responsabilidades e atividades acabam fluindo naturalmente, sem cobranças e pressão. Também cria-se um nivelamento de conhecimento sobre todas as áreas, afinidade entre as equipes e prototipações que já passaram pelo processo de validação, que sem o DT, ou abordagem semelhante, a validação ocorreria apenas quando se tivesse o software pronto atendendo os critérios de verificação dos requisitos.

Ao final apresenta-se através da Figura 5.12 os benefícios e desafios encontrados.

Figura 5.12 – Benefícios e Desafios do DT

Pontos Positivos 	Pontos Negativos 
Validação do produto Ajuda na criação de MVP Elucidação de requisitos Integração com métodos ágeis Reduzi risco e esforço para o desenvolvimento Iterativo e apto para mudanças frequentes Mais eficaz em projetos de pequeno a médio porte Engamento da equipe no processo de desenvolvimento Promove uma solução viável dentro do recurso inserido Ideias mais elaboradas / Pensar fora da "caixa" Trabalha em conjunto com o usuário final Soluciona problemas reais e mais próximo da realidade Poder errar várias vezes no início para ser assertivo no final	Disponibilidade de todos os envolvidos Necessidade de tempo para aplica e de todos para participar Difícil escalar para sistemas maiores Mudar o pensamento conservador Necessita de pessoas de diferentes áreas Falta de conhecimento sobre a técnica não gera satisfação e resultado Consenso entre as áreas Grande demanda de tempo Excesso de frameworks a serem utilizados Dificuldade em manter o foco até o final Resistência Investimento

FONTE: Autor, 2020

5.5 Estudo Correlato

Este trabalho abordou muitos contextos sobre o DT na ES e mostrou-se ser um assunto na qual está se expandindo tanto na pesquisa como no seu uso na indústria. Durante o processo realizado no Estudo 1, preocupou-se apenas em buscar estudos primários, ou seja, estudos que relatam o uso do DT na ES, para isso o Estudo 1 é caracterizado por ser um estudo secundário. Entretanto, existem outros estudos secundários que relatam

sobre revisões sistemáticas na literatura que estão pesquisando também sobre o mesmo fenômeno. O viés deste trabalho é agregar valor para a literatura e indústria com os dois estudos realizados para fortalecer o conhecimento do DT através do corpo de conhecimento criado. Nessa seção aborda outro trabalho, encontrado no Estudo 1, que também realizou um estudo secundário na literatura sobre o mesmo fenômeno. O intuito de trazer esse trabalho é objetivar a forma que esta dissertação agregou valor para literatura em complementação com outros trabalhos que também estudam o mesmo fenômeno.

O trabalho encontrado no Estudo 1 foi: Aplicando *Design Thinking* em Engenharia de Software: um Mapeamento Sistemático [30]. Ao final desta seção apresentam-se uma comparação sobre o trabalho encontrado com os resultados obtidos nesta dissertação.

Estudo Correlato. Conte, Ferreira e Souza [30] trazem um mapeamento na literatura com objetivo de investigar e compreender como DT é aplicado na Engenharia de Software. Foram extraídas informações sobre a aplicação de DT no desenvolvimento de software. Este mapeamento apresenta 11 modelos de DT, 55 técnicas e 10 ferramentas de software. O método de pesquisa proposto neste trabalho seguem as orientações de Kitchenham e Charters [60], abordando sua questão de pesquisa, estratégia e processo de busca, critérios de inclusão e exclusão e o processo de extração dos dados.

A partir das orientações de Kitchenham e Charters [60], os autores [30] formularam a seguinte questão de pesquisa: Como *Design Thinking* é aplicado em Engenharia de Software?. Através desta questão, elaboraram três questões específicas:

1. Quais os modelos utilizados em Design Thinking?
2. Quais as técnicas métodos utilizadas dentro dos modelos de Design Thinking?
3. Quais as ferramentas existentes para apoiar a utilização das técnicas de Design Thinking?

Após a elaboração da questão de pesquisa, exibem sua estratégia e processo de busca dos artigos. Utilizaram como fontes digitais de busca a Scopus¹ e a *Engineering Village*². Argumentam que estas duas bases abrangem importantes periódicos na área. Sobre a Scopus citam ser uma das bases que possui os principais periódicos de Engenharia de Software e que a *Engineering Village* agrega informações de diversos bancos de dados bibliográficos na área da Ciência da Computação, possuindo periódicos e conferências da IEEE, ACM, Springer e Elsevier [30]. Com esta fundamentação os autores realizam sua busca através da *string* abaixo:

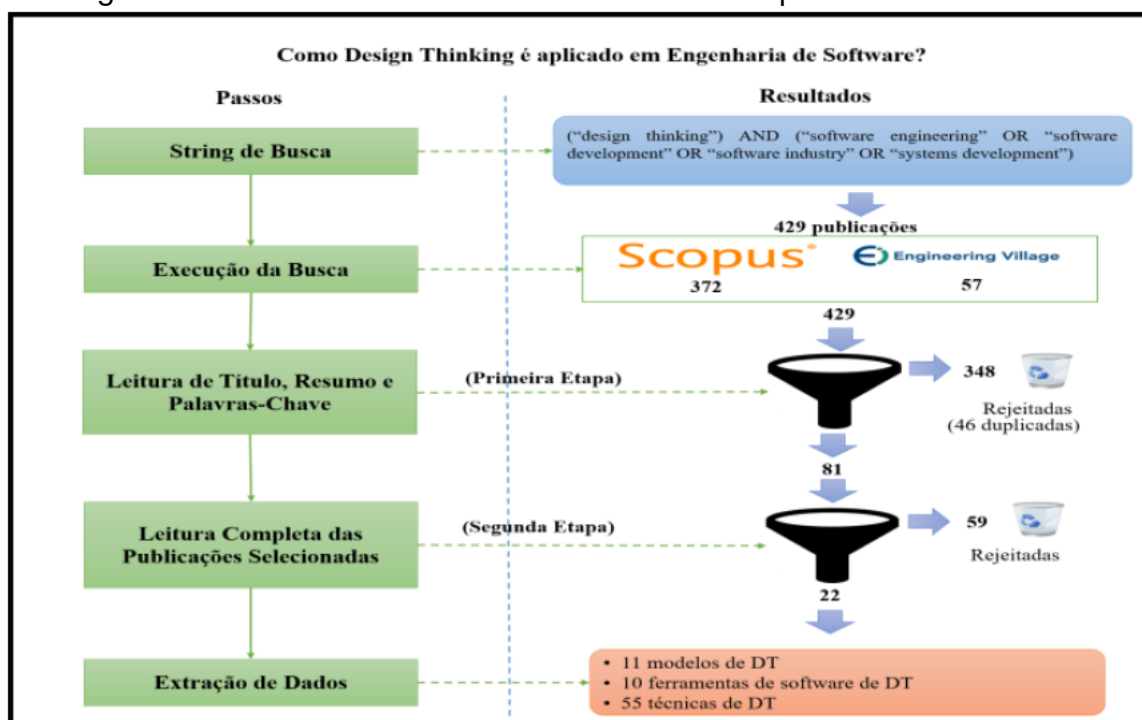
- (“*design thinking*”) AND (“*software engineering*” OR “*software development*” OR “*software industry*” OR “*systems development*”)

¹<https://www.scopus.com/>

²<https://www.engineeringvillage.com>

Para ilustrar o processo e simplificar o entendimento do fluxo do mapeamento sistemática, elaboraram a Figura 5.13.

Figura 5.13 – Trabalho Correlato: Processo do Mapeamento Sistemático



FONTE: Conte et al. [30]

A Figura 5.13 mostra os passos realizados pelos autores e os resultados obtidos em ambas bases escolhidas. Ao final foram encontrados 372 artigos na Scopus e 57 na *Engineering Village*. Realizaram uma primeira etapa de filtragem sobre o total de 429 artigos e rejeitaram 348, sendo 46 destes rejeitados como duplicados. Com o resultado de 81 artigos, novamente fizeram uma triagem e 59 artigos foram rejeitados, resultado ao final 22 artigos [30]. Para os critérios de inclusão, os autores buscaram os artigos com as seguintes características:

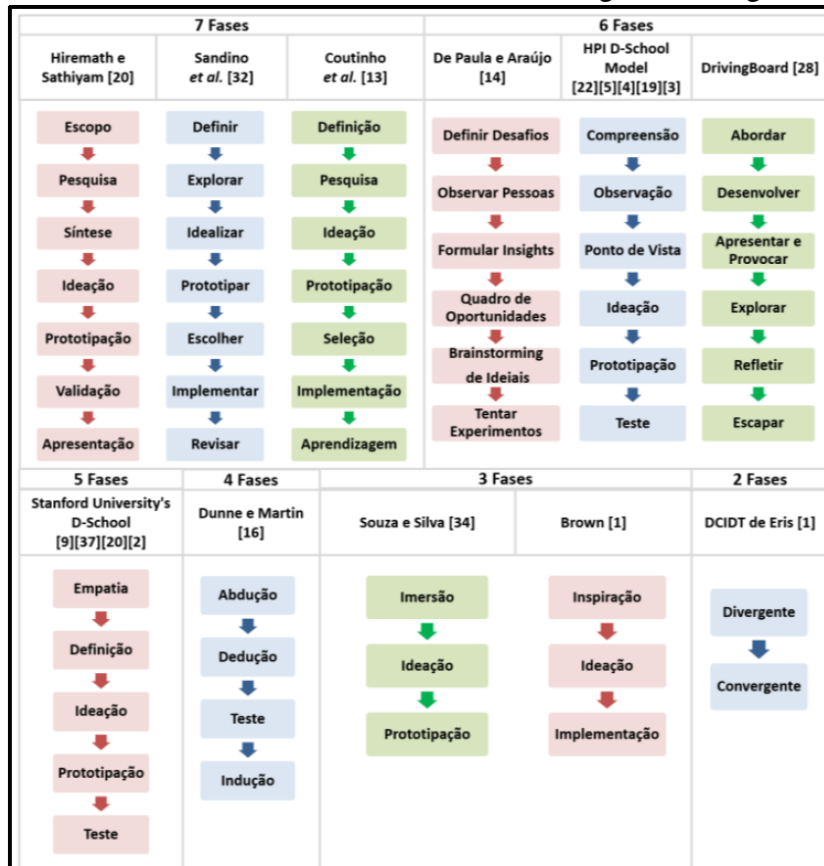
- A publicação apresenta ferramentas computacionais, métodos, modelos ou abordagens baseadas em Design Thinking utilizados em Engenharia de Software.
- A publicação apresenta uma discussão sobre a inclusão de Design Thinking no processo de desenvolvimento de software.
- A publicação apresenta casos de uso de Design Thinking em organizações ou na academia utilizados no contexto de Engenharia de Software.

Os artigos excluídos encontravam-se com os seguintes critérios: Não foram selecionadas publicações que não atenderam os critérios de inclusão, publicações duplicadas,

publicações que não estavam disponíveis para leitura e publicações que não estavam nos idiomas português e inglês.

Após a triagem dos dados e análise dos 22 artigos aceitos, ilustraram os modelos com suas fases e autores através da Figura 5.14.

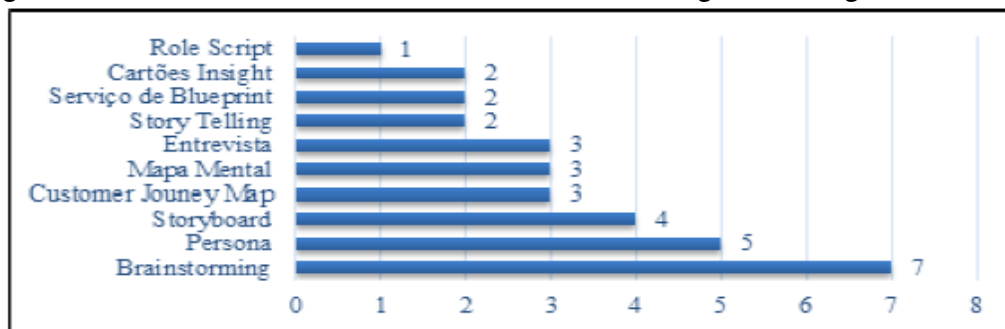
Figura 5.14 – Trabalho Correlato: Modelos de *Design Thinking* e suas Fases



FONTE: Conte et al. [30]

A segunda questão específica foi respondida mostrando um gráfico com as técnicas mais citadas dentre os 22 artigos selecionados. A Figura 5.15 mostra a quantidade de citação sobre cada técnica. A técnica *Role Script* encontra-se apenas com 1 citação e entra na categoria das técnicas que foram encontradas e citadas apenas uma vez, resultando no total em 55 técnicas encontradas. Os autores mostram as demais técnicas em uma tabela. Também, concluíam que as técnicas podem ser usadas em uma ou mais fases do DT e uma única fase pode usar uma ou mais técnicas, dependendo do contexto do problema [30].

Para a resposta da terceira questão específica, encontraram 10 ferramentas de apoio. Estas ferramentas foram exibidas sobre qual possível fase do DT podem ser utilizadas e em qual técnica de DT se enquadram. Entretanto, as ferramentas Balsamiq e Nvivo não foram especificadas em qual técnica podem ser utilizadas e Stakeholder Circle, Creately e Scenes não especificaram em qual fase de DT podem ser usadas [30].

Figura 5.15 – Trabalho Correlato: Técnicas de *Design Thinking* mais Citadas

FONTE: Conte et al. [30]

Ao final, Conte, Ferreira e Souza [30] discutem os resultados a partir da perspectiva da relação observada por eles sobre o uso dos modelos de DT mapeados na Engenharia de Software e sobre as semelhanças entre suas fases, bem como o uso das técnicas de DT. Após mostrarem cada fase, suas semelhanças e o ambiente que podem ser utilizadas, concluem a descoberta de 11 modelos de DT usados na engenharia de software mas não conseguem afirmar se os modelos mais citados são os mais usados. Também, fecham o artigo mostrando que conseguiram observar que as técnicas podem ser usadas em conjunto em uma mesma fase e que ainda não há um grande número de ferramenta de software disponíveis de apoio ao processo de DT. Como trabalho futuro, os autores pretendem realizar um estudo para verificar qual o melhor conjunto de técnicas para uma determinada fase, além de também, realizarem um estudo comparativo entre algumas ferramentas que apoiam o uso das técnicas de DT.

Após estudo sobre este trabalho, foi realizado um levantamento considerando a convergência de valores de seus resultados e metodologias utilizadas, podendo assim elencar alguns pontos correlatos entre esta dissertação e esse trabalho, junto com uma comparação entre os mapeamentos realizados e resultados obtidos

Existem diversos artigos que abordam o uso do DT na prática ou direcionado em revisões ou mapeamentos na literatura, buscando entender de forma quantitativa ou qualitativa como o DT está emergindo no ambiente de desenvolvimento de software. Sobretudo o objetivo maior de fato é aproximar o que o entendimento do uso do DT na ES para os pesquisadores e profissionais, almejando um conhecimento mais claro sobre esse fenômeno entre diversos pontos de vistas trabalhados na literatura. A Tabela 5.1 apresenta uma comparação sobre o Estudo 1 e o trabalho correlato descrito.

Foi colocado em comparação com este trabalho apenas o Estudo 1, visto que ambos realizaram uma pesquisa na literatura a fim de responder suas questões de pesquisa. Com o entendimento e discussão sobre os contextos, consolidação das informações para a criação deste corpo de conhecimento e abordando a forma que esse trabalho está vindo

para agregar valor para outros que já estão na literatura, o próximo Capítulo apresenta a conclusão, as limitações e trabalhos futuros.

Tabela 5.1 – Estudo 1 x Trabalho Correlato

	Estudo 1	Trabalho Correlato	De/Para
Objetivo do Trabalho	O objetivo geral é caracterizar o uso de DT no desenvolvimento de software. Essa caracterização envolve o entendimento de aspectos relacionados ao uso da abordagem, seus benefícios e dificuldades enfrentadas.	O objetivo deste trabalho é apresentar os resultados de um mapeamento sistemático onde foram identificadas publicações que especificam modelos de DT em ES, bem como técnicas e ferramentas que auxiliam na aplicação de DT	Estudo 1 procurou conhecer, além dos modelos, técnicas e ferramentas, os contextos trabalhados e os cenários de uso. Com isso também procurou-se os benefícios e desafios na aplicação do DT. O trabalho correlato trabalhou apenas com os modelos, técnicas e ferramentas, procurando relacionar o uso dos modelos com as técnicas e suas ferramentas.
Revisão / Mapeamento	Mapeamento Sistemático	Mapeamento Sistemático	Ambos trabalhos realizados um mapeamento sistemático
Qtd. de Questões de Pesquisa	1) Qual o propósito de se usar DT na ES? 2) Em qual contexto de desenvolvimento de software o DT está sendo usado? 3) Quais modelos, métodos, técnicas e/ou abordagens utilizadas do DT na ES? 4) Quais ferramentas são utilizadas para apoiar a realização de DT no desenvolvimento de software? 5) Como estes modelos, métodos, técnicas, abordagens e ferramentas são selecionados para o uso? 6) Quais são as lições aprendidas, boas práticas, recomendações e dificuldades experimentadas no DT no desenvolvimento de software?	1) Quais os modelos utilizados em Design Thinking? 2) Quais as técnicas métodos utilizadas dentro dos modelos de Design Thinking? 3) Quais as ferramentas existentes para apoiar a utilização das técnicas de Design Thinking?	O Estudo 1 trouxe um objetivo em entender o propósito de uso do DT sobre diversos cenários encontrados e atuando sobre diversos modelos, técnicas e/ou ferramentas. Além disso, buscou entender como essas técnicas são selecionadas. O trabalho correlato buscou encontrar os modelos, técnicas e ferramentas que estão sendo usadas, não procurando o motivo da escolha destas, os cenários de uso e os benefícios e desafios encontrados na aplicação destas técnicas.
Qtd. de bases de dados	ACM IEEE Xplore Science Direct Scopus Springer Wiley	Scopus Engineering Village	Estudo 1 procurou em 6 bases de dados. O trabalho correlato procurou em apenas 2 bases. A base SCOPUS busca artigos de outras bases. Porém, o Estudo 1 preocupou-se em buscar em outras bases que compõem os resultados da SCOPUS para garantir que não se perdesse artigos importados para serem analisados.
Anos dos artigos	2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019	2011, 2013, 2014, 2015, 2016	Estudo 1 encontrou artigos em mais anos que o trabalho correlato.
Qtd. de Artigos retornados das bases	2588 artigos	429 artigos	Estudo 1 teve 2.159 artigos a mais para aplicar o mapeamento sistemático.
Qtd. de Artigos Extraídos	78 publicações extraídas	22 publicações extraídas	Estudo 1 teve 56 artigos extraídos a mais que o trabalho correlato.
Utilizou KAPPA?	Sim, sobre 20% dos artigos selecionados na segunda etapa do trabalho (selecionados aleatoriamente)	Sim, sobre 10% dos artigos (selecionados aleatoriamente)	Estudo 1 realizou 3 etapas para seleção dos artigos. A etapa 2 do Estudo 1, obteve um total de 783 artigos, onde sobre estes artigos foi realizado o KAPPA. O trabalho correlato realizou o KAPPA sobre 10% de todos os trabalhos encontrados (429 artigos).
Contextos de DT Utilizados	Encontrados 12 contextos usados. Posteriormente classificados em 4 grupos: Educação, Inovação, Explorar e entender o problema e semelhanças ágeis.	Não citado	Estudo 1 abordou o tema de propósito de uso do DT na ES.
Cenários de Uso de DT	Mais de 10 cenários encontrados, sendo o seu grande uso em 46,3% sobre integração com métodos ágeis.	Não citado	Estudo 1 buscou cenários de uso do DT na ES.
Qtd. de Modelos	19 modelos encontrados	11 modelos encontrados	Estudo 1 encontrou 8 modelos a mais que o trabalho correlato.
Qtd. de Técnicas	46 técnicas encontradas	55 técnicas encontradas	Trabalho correlato encontrou 9 técnicas a mais que o Estudo 1.
Qtd. de Ferramentas	7 ferramentas encontradas	10 ferramentas encontradas	Trabalho correlato encontrou 3 ferramentas a mais que o Estudo 1.
Qtd. de Benefícios	13 benefícios encontrados	Não citado	Estudo 1 apresentou 13 benefícios para se usar o DT na ES.
Qtd. de Desafios	18 desafios encontrados	Não citado	Estudo 1 apresentou 18 dificuldades/desafios de uso do DT na ES.

FONTE: Autor, 2020

6. CONSIDERAÇÕES FINAIS

Este Capítulo possui a resolução final e conclusões sobre os estudos realizados neste trabalho, dividido em quatro seções: a conclusão, as limitações encontradas durante o processo de desenvolvimento e os trabalhos futuros.

6.1 Conclusão

Este trabalho apresentou um corpo de conhecimento através de uma pesquisa realizada no Mapeamento Sistemático da Literatura (Estudo 1) para entender como o DT está sendo trabalhado na literatura e buscar uma quantidade máxima de abordagens, modelos, técnicas, cenários de uso, benefícios e desafios que o DT está viabilizando para a comunidade de software. Com o resultado consolidado do Estudo 1 realizou-se uma Survey (Estudo 2), que trouxe o propósito de suprir as lacunas encontradas na literatura e maximizar o conhecimento do DT na visão de profissionais praticantes desta abordagem na indústria.

O DT pode ser usado em uma ampla gama de atividades em Engenharia de Software. No mapeamento realizado neste trabalho foram encontrados diversos modelos, técnicas e ferramentas de DT, pesquisadores mais ativos e mecanismos de publicações para este tópico, com mais de 50 artigos extraídos com abordagens do DT, ou seja, o seu uso está sendo muito explorado na área da computação. Mais pesquisas poderão ser conduzidas para coletar informações mais detalhadas de outras perspectivas sobre a integração de DT com a Engenharia de Software. Com o valor trazido neste trabalho, unificou-se diversas formas, visões e perspectivas de enxergar o DT e usá-lo no contexto da Engenharia de Software.

Na maioria dos trabalhos analisados no Estudo 1, o embasamento mais citado sobre o uso do DT foi de Brown [12], conceituando que o DT é uma abordagem de solução de problemas que traz princípios de design centrado no usuário em sua essência. Impulsionado pela busca por inovação, ele se encaixa no desenvolvimento de novos produtos, serviços e processos em vários contextos, em startups a ambientes grandes e complexos. Entende-se com esse conceito que o DT é útil e de interesse para as equipes de desenvolvimento de software, principalmente para as equipes ágeis, onde está muito presente nos dias de hoje pelos profissionais da TI e que aborda termos e princípios muito semelhantes do DT.

Outro ponto na qual cabe destacar é possível análise sobre os dados extraídos com o processo real na indústria a ponto de correlacionar informações entre DT e métodos ágeis, fortalecendo o fato que o resultado trazido neste trabalho mostrou claramente que

o DT é muito usado em contextos de desenvolvimento de software ágil e este é seu maior cenário dentro da Engenharia de Software. A investigação apresentada esclarece que a união dos métodos ágeis com o DT pode acarretar em uma melhora no entendimento geral do contexto em determinadas situações, principalmente em trazer o usuário para o centro do processo de desenvolvimento e entendê-lo melhor. Dentre várias informações obtidas sobre o olhar de diferentes bases, destaca-se o tema menos discutido: o uso de ferramentas. Foram poucos os resultados trazidos sobre ferramentas e poucos artigos citam sobre. Também, o Estudo 2 realizado mostra que os profissionais não se preocupam muito com as ferramentas e que estas seriam apenas uma forma de apoiar a concretização de resultados obtidos nas técnicas. Dessa forma surge uma possível lacuna para ser estudada com maior profundidade ou uma oportunidade de criação de ferramentas que auxilia o DT em suas diversas etapas dentro do processo de desenvolvimento de software, sem eliminar a forma de se aplicar o DT e que venha com a visão de cocriar, maior interação e facilitação das aplicações das técnicas.

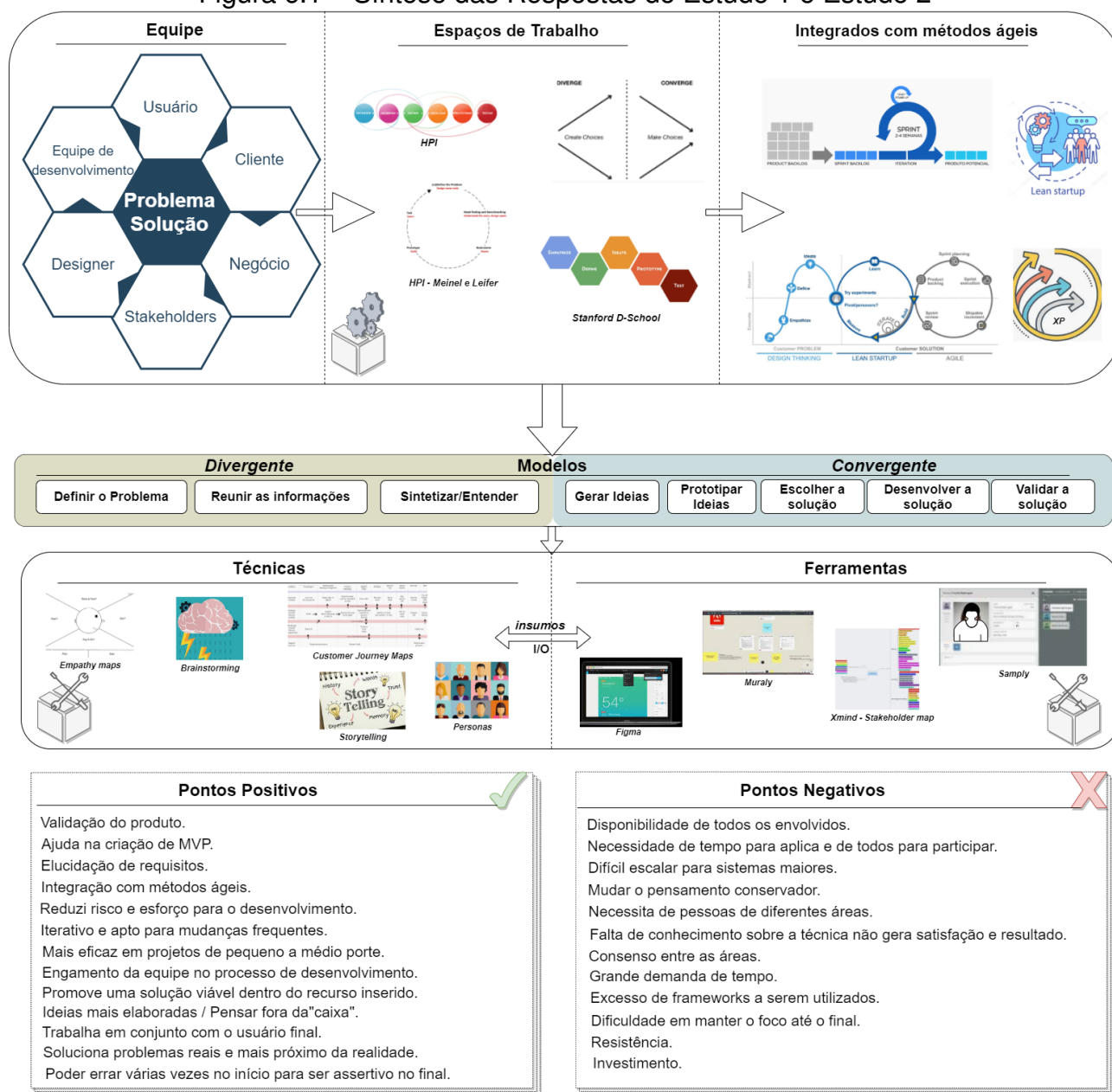
Também encontrou-se neste estudo uma relação do DT com UX [78], onde existe a possibilidade da utilização de UX na etapa de avaliação sobre modelo de DT. Pelo entendimento da aplicação do DT, entende-se que o mesmo é utilizado em contextos nas quais existem um problema a ser explorado, sendo este problema muito citado como melhorar o entendimento da necessidade do usuário. Com isso o DT acaba-se sendo usado em situações que não se sabe qual a melhor solução ou se realmente haverá um software ou um serviço envolvido. Por outro lado o UX vem sobre a perspectiva de avaliar o uso de um software ou serviço, ou seja, já se existe algo para ser avaliado onde muitas vezes surgiu de um processo de DT.

Embora os estudos apresentados não forneçam detalhes sobre como ou por que quais dos modelos e técnicas de DT foram escolhido (por exemplo, por que a observação foi usada em vez de narrativa no modelo IBM DT [78]), recomenda-se da mesma forma o uso do DT. O mapeamento sistemático e a survey revelaram que os modelos se ajustam a diferentes cenários de desenvolvimento de software e essas técnicas devem ser vistas como um repositório de recursos para escolha do profissional. Se os profissionais da Engenharia de Software vestissem a camisa do designer e comesçassem à agir de acordo com os mesmos princípios, poderia de certa forma almejar mais o DT no processo de desenvolvimento. O DT não é uma receita pronta para ser aplicada conforme a situação, por mais que existam diversos exemplos de uso de sucesso, deve-se aprender a construir sua própria forma com base na experiência prévia, no conhecimento sobre as técnicas e das pessoas, das equipes multidisciplinares. Contata-se que o DT pode ser denominado como uma framework, ou seja, ele possui suas próprias características e o mínimo para sua utilização (usuário presente pois ele denomina o problema, o processo sempre no início será divergente e terminará como convergente, ou seja, existem espaços de trabalho que devem ser trabalhados, participantes de todas as áreas possíveis, deverá vir com propósito de so-

lucionar o problema e não gerar um software tangível mas sim em um escopo próximo e pré validado pelo usuário sobre o verdadeira problema explorado e a melhor solução possível. Percebe-se então que o DT possui diversas estruturas para estudar e conhecer problema ao ponto de encontrar a melhor solução para ele, porém entende-se que existe um mínimo do DT que deve ser respeitado para que haja-se sucesso em sua aplicação.

A Figura 6.1 apresenta uma síntese de acordo com as características encontradas através do Estudo 1 e Estudo 2 sobre o DT no desenvolvimento de software.

Figura 6.1 – Síntese das Respostas do Estudo 1 e Estudo 2



FONTE: Autor, 2020

6.2 Limitações

Todo trabalho está sujeito a sofrer mudanças e limitações que são encontradas ao decorrer do trabalho. Sobre as limitações, destacam-se:

- A realização de levantamento de respostas do Estudo 2 durante o período de recesso de ano novo e natal. Conforme apresentado no Estudo 2, por mais que tenha-se alcançado um resultado atendível para o propósito deste trabalho, o recesso trouxe uma grande diminuição de respostas durante o período da coleta e que possivelmente, se a coleta ocorresse em um período normal, tivesse um aumento considerável para análise no Estudo 2.
- Houve limitações por parte do pesquisador para entender e consolidar uma imensa quantidade de artigos retornados no Estudo 1, de forma que foi necessário construir um cronograma e aplicá-lo ao longo de um semestre e meio. Este tempo gasto sobre análise dos artigos retornados, mesmo com apoio de outros pesquisadores, prejudicou na atualização dos resultados do Estudo 1, não sendo possível a coleta de novos estudos sobre o ano de 2019.

6.3 Trabalhos Futuros

Entende-se que existe um grande potencial de crescimento sobre o DT no desenvolvimento de software e que através dos dados analisados neste trabalho, surgem oportunidades como: (i) Criar, aprimorar e entender as diferentes ferramentas que apoiam a aplicação de cada técnica utilizadas nos espaços de trabalho do DT. (ii) A realização de um estudo empírico na indústria, aplicando o DT através de um estudo de caso com profissionais e extraindo informações que apenas pelo conhecimento adquirido de forma empírica consegue-se agregar valor para o estudo. (iii) Criar *guidelines* para se usar o DT sobre diversas situações. Acredita-se que este estudo trouxe uma gama enorme de informações benéficas para que se possa trabalhar para iniciar uma construção de direções de apoio ao desenvolvimento do DT. (iv) Levar este estudo para fora do Brasil, almejando a qualificação de um estudo internacional que ofereça valor de retorno para o desenvolvimento de software no Brasil.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Adikari, S.; Keighran, H.; Sarbazhosseini, H. “Embed design thinking in co-design for rapid innovation of design solutions”. In: *Proceedings of the International Conference of Design, User Experience, and Usability*, Toronto, Canada: Springer, 2016, pp. 12.
- [2] Adikari, S.; McDonald, C.; Campbell, J. “Reframed contexts: design thinking for agile user experience design”. In: *Proceedings of the International Conference of Design, User Experience, and Usability*, Las Vegas, United States: Springer, 2013, pp. 10.
- [3] Ahmad, R.; Chyi, W. A.; Sarlan, A.; Kasbon, R. “Guiding novice database developers in database schema creation”. In: *Proceedings of the Conference on e-Learning, e-Management and e-Services*, Melbourne, Australia: IEEE, 2014, pp. 6.
- [4] Araújo, R.; Anjos, E.; Silva, D. R. “Trends in the use of design thinking for embedded systems”. In: *Proceedings of the International Conference on Computational Science and Its Applications*, Banff, Canada: IEEE, 2015, pp. 5.
- [5] Asante, G. “Effective design methodologies”, *Design Management Review*, vol. 29, Jun 2018, pp. 10–15.
- [6] Avalos, M.; Larios, V. M.; Salazar, P.; Maciel, R. “Hackathons, semesterathons, and summerathons as vehicles to develop smart city local talent that via their innovations promote synergy between industry, academia, government and citizens”. In: *Proceedings of the International Smart Cities Conference*, Wuxi, China: IEEE, 2017, pp. 6.
- [7] Barbosa, A. F. P. “Design thinking na especificação de requisitos: O caso i2s – informática, sistemas e soluções”, *Dissertação de mestrado*, Católica Porto Business School, 2016, 23p.
- [8] Beck, K.; Beedle, M.; Bennekum, A. v.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marick, B.; Martin, R. C.; Mellor, S.; Schwaber, K.; Sutherland, J.; Thomas, D. “Manifesto for agile software development”. Capturado em: <http://agilemanifesto.org/>, Fevereiro 2020.
- [9] Berger, A. “Design thinking for search user interface design”, *Central Europe Workshop Proceedings*, vol. 763, Jan 2011, pp. 1–4.
- [10] Borba, A. W. T.; Batista, G. H. C.; Souza, R. A. C. “Innostartup-a toolbox for innovation in software development process”, *Latin America Transactions*, vol. 14, Dez 2016, pp. 3875–3885.

- [11] Brenner, W.; Uebernickel, F.; Abrell, T. "Design thinking as Mindset, Process, and Toolbox". Springer, 2016, cap. 1, pp. 3–21.
- [12] Brown, T. "Design thinking", *Harvard Business Review*, vol. 86, Jun 2008, pp. 84–95.
- [13] Brown, T. "Design thinking: uma metodologia poderosa para decretar o fim das velhas ideias". Alta Books Editora, 2018, 272p.
- [14] Burchardt, C.; Maisch, B. "Advanced agile approaches to improve engineering activities", *Procedia Manufacturing*, vol. 25, Mai 2018, pp. 202–212.
- [15] Carell, A.; Lauenroth, K.; Platz, D. "Using Design Thinking for Requirements Engineering in the Context of Digitalization and Digital Transformation: A Motivation and an Experience Report". Springer, 2018, cap. 7, pp. 107–120.
- [16] Carlgren, L.; Rauth, I.; Elmquist, M. "Framing design thinking: The concept in idea and enactment", *Creativity and Innovation Management*, vol. 25, Mar 2016, pp. 38–57.
- [17] Carroll, N.; Richardson, I. "Aligning healthcare innovation and software requirements through design thinking". In: Proceedings of the International Workshop on Software Engineering in Healthcare Systems, Austin, United States: ACM, 2016, pp. 7.
- [18] Castro, J. W.; Acuña, S. T.; Juristo Juzgado, N. "Enriching requirements analysis with the personas technique". In: Proceedings of the Workshop on the Interplay between Usability Evaluation and Software Development, Pisa, Italy: CEUR-WS, 2008, pp. 6.
- [19] Chasanidou, D.; Gasparini, A. A.; Lee, E. "Design thinking methods and tools for innovation". In: Proceedings of the Design, User Experience, and Usability: Design Discourse, Los Angeles, United States: Springer, 2015, pp. 12.
- [20] Cockton, G. "Integrating Both User-Centered Design and Creative Practices into Agile Development". Springer, 2016, cap. 11, pp. 249–276.
- [21] Cockton, G.; Lárusdóttir, M.; Gregory, P.; Cajander, Å. "Integrating User-Centred Design in Agile Development". Springer, 2016, 291p.
- [22] Cohen, J. "A coefficient of agreement for nominal scales", *Educational and Psychological Measurement*, vol. 20, Abr 1960, pp. 37–46.
- [23] Corral, L.; Fronza, I. "Design thinking and agile practices for software engineering: An opportunity for innovation". In: Proceedings of the Special Interest Group on Conference on Information Technology Education, Fort Lauderdale, United States: ACM, 2018, pp. 6.

- [24] Costa Valentim, N. M.; Silva, W.; Conte, T. “The students’ perspectives on applying design thinking for the design of mobile applications”. In: Proceedings of the International Conference on Software Engineering: Software Engineering Education and Training Track, Buenos Aires, Argentina: IEEE, 2017, pp. 10.
- [25] Coutinho, E. F.; Gomes, G. A. M.; José, M. A. “Applying design thinking in disciplines of systems development”. In: Proceedings of the Euro American Conference on Telematics and Information Systems, Cartagena, Colombia: IEEE, 2016, pp. 8.
- [26] da Cruz Júnior, G. G.; do Nascimento, R. L. S. “Desenvolvendo um objeto de aprendizagem virtual para o ensino da programação web mobile com o design thinking”. In: Anais do Congresso Regional sobre Tecnologias na Educação, Natal, Brasil: CEUR-WS, 2016, pp. 12.
- [27] Darrin, M. A. G.; Devereux, W. S. “The agile manifesto, design thinking and systems engineering”. In: Proceedings of the International Systems Conference, Montreal, Canada: IEEE, 2017, pp. 5.
- [28] de Carvalho Souza, C. L.; Silva, C. T. “Use of design thinking in requirements elicitation of mobile learning in virtual environments”. In: Proceedings of the Workshop on Requirements Engineering, Pucón, Chile: PUC-Rio, 2014, pp. 14.
- [29] de Paula, D. F.; Araújo, C. C. “Pet empires: Combining design thinking, lean startup and agile to learn from failure and develop a successful game in an undergraduate environment”. In: Proceedings of the International Conference on Human-Computer Interaction, Toronto, Canada: Springer, 2016, pp. 5.
- [30] de Souza, A. F. B.; Ferreira, B. M.; Conte, T. “Applying design thinking in software engineering: A systematic mapping”. In: Proceedings of the Ibero-American Conference on Software Engineering, Buenos Aires, Argentina: ClbSE, 2017, pp. 14.
- [31] Dobrigkeit, F.; de Paula, D.; et al.. “The best of three worlds-the creation of innodev a software development approach that integrates design thinking, scrum and lean startup”. In: Proceedings of the International Conference on Engineering Design, Vancouver, Canada: Design Society, 2017, pp. 10.
- [32] Dobrigkeit, F.; Wilson, M.; Nicolai, C.; et al.. “Adding scrum-style project management to an advanced design thinking class”. In: Proceedings of the NordDesign, Linköping, Sweden: Design Society, 2018, pp. 12.
- [33] Dunne, D.; Martin, R. “Design thinking and how it will change management education: An interview and discussion”, *Academy of Management Learning & Education*, vol. 5, Dez 2006, pp. 512–523.

- [34] Díaz, P.; Aedo, I.; Cubas, J. “Codice: Balancing software engineering and creativity in the co-design of digital encounters with cultural heritage”. In: Proceedings of the Workshop on Advanced Visual Interfaces, Como, Italy: ACM, 2014, pp. 4.
- [35] ECHOS. “Design thinking | echos laboratório de inovação”. Capturado em: <https://escoladesignthinking.echos.cc>, Fevereiro 2020.
- [36] El-Sharkawy, S.; Schmid, K. “A heuristic approach for supporting product innovation in requirements engineering: A controlled experiment”. In: Proceedings of the International Conference on Requirements Engineering: Foundation for Software Quality, Essen, Germany: Springer, 2011, pp. 16.
- [37] Fabbri, S.; Silva, C.; Hernandez, E.; Octaviano, F.; Di Thommazo, A.; Belgamo, A. “Improvements in the start tool to better support the systematic review process”. In: Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, Limerick, Ireland: ACM, 2016, pp. 21.
- [38] Fernandes, J. C.; Neto, V. V. G.; Santos, R. P. d. “Interoperability in systems-of-information systems: A systematic mapping study”. In: Proceedings of the 17th Brazilian Symposium on Software Quality, Curitiba, Brazil: ACM, 2018, pp. 10.
- [39] Ferreira, B.; Conte, T.; Barbosa, S. D. J. “Eliciting requirements using personas and empathy map to enhance the user experience”. In: Proceedings of the Brazilian Symposium on Software Engineering, Belo Horizonte, Brazil: IEEE, 2015, pp. 10.
- [40] Fleiss, J. L.; Levin, B.; Paik, M. C. “Statistical methods for rates and proportions”. John Wiley & Sons, 2003, 800p.
- [41] Glomann, L. “Introducing ‘human-centered agile workflow’(hcaw)—an agile conception and development process model”. In: Proceedings of the International Conference on Applied Human Factors and Ergonomics, Los Angeles, United States: Springer, 2017, pp. 10.
- [42] González, C. S. G.; González, E. G.; Cruz, V. M.; Saavedra, J. S. “Integrating the design thinking into the ucd’s methodology”. In: Proceedings of the Engineering Education Conference, Madrid, Spain: IEEE, 2010, pp. 4.
- [43] Goodspeed, R.; Riseng, C.; Wehrly, K.; Yin, W.; Mason, L.; Schoenfeldt, B. “Applying design thinking methods to ecosystem management tools: Creating the great lakes aquatic habitat explorer”, *Marine Policy*, vol. 69, Jul 2016, pp. 134–145.
- [44] Goudsouzian, L. K.; Riola, P.; Ruggles, K.; Gupta, P.; Mondoux, M. A. “Integrating cell and molecular biology concepts: Comparing learning gains and self-efficacy in corresponding live and virtual undergraduate laboratory experiences”, *Biochemistry and Molecular Biology Education*, vol. 46, Jul 2018, pp. 361–372.

- [45] Grossman-Kahn, B.; Rosensweig, R. “Skip the silver bullet: driving innovation through small bets and diverse practices”, *Leading Through Design*, vol. 18, Ago 2012, pp. 815.
- [46] Gumienny, R.; Meinel, C.; Gericke, L.; Quasthoff, M.; LoBue, P.; Willems, C. “Teleboard: enabling efficient collaboration in digital design spaces across time and distance”. Springer, 2011, cap. 9, pp. 147–164.
- [47] Gürdür, D.; Törngren, M. “Visual analytics for cyber-physical systems development: Blending design thinking and systems thinking”. In: Proceedings of the NordDesign, Linköping, Sweden: Design Society, 2018, pp. 15.
- [48] Gurusamy, K.; Srinivasaraghavan, N.; Adikari, S. “An integrated framework for design thinking and agile methods for digital transformation”. In: Proceedings of the International Conference of Design, User Experience, and Usability, Toronto, Canada: Springer, 2016, pp. 9.
- [49] Hasso-Plattner-Institut, D.-S. “What is design thinking?” Capturado em: <https://hpi-academy.de/en/design-thinking/what-is-design-thinking.html>, Fevereiro 2020.
- [50] Hehn, J.; Uebernicketel, F. “The use of design thinking for requirements engineering: An ongoing case study in the field of innovative software-intensive systems”. In: Proceedings of the International Requirements Engineering Conference, Banff, Canada: IEEE, 2018, pp. 6.
- [51] Hehn, J.; Uebernicketel, F.; Stöckli, E.; Brenner, W. “Designing human-centric information systems: Towards an understanding of challenges in specifying requirements within design thinking projects”. In: Proceedings Of The Multikonferenz Wirtschaftsinformatik, Lüneburg, Germany: Leuphana University of Lüneburg, 2018, pp. 12.
- [52] Hiremath, M.; Sathiyam, V. “Fast train to dt: a practical guide to coach design thinking in software industry”. In: Proceedings of the International Federation for Information Processing, Cape Town, South Africa: Springer Nature, 2013, pp. 8.
- [53] IBM. “Enterprise design thinking”. Capturado em: <https://www.ibm.com/design/thinking/page/framework/loop>, Fevereiro 2020.
- [54] IP, M. E. “Localizar ip”. Capturado em: <https://www.meuenderecoip.com/localizar-ip.php>, Fevereiro 2020.
- [55] Jensen, M. B.; Lozano, F.; Steinert, M. “The origins of design thinking and the relevance in software innovations”. In: Proceedings of the International Conference on Product-Focused Software Process Improvement, Trondheim, Norway: Springer, 2016, pp. 4.

- [56] Kasunic, M. "Designing an Effective Survey". Carnegie Mellon University, 2018, 143p.
- [57] Kaur, R.; Sengupta, J. "Software process models and analysis on failure of software development projects", *International Journal of Scientific & Engineering Research*, vol. 2, Jun 2013, pp. 1–4.
- [58] Keighran, H.; Adikari, S. "Developing high-performing teams: A design thinking led approach". In: *Proceedings of the International Conference of Design, User Experience, and Usability*, Toronto, Canada: Springer, 2016, pp. 12.
- [59] Kelley, T.; Littman, J. "The Art of Innovation: Lessons in Creativity from IDEO, America's Leading Design Firm". Harper Collins Business, 2001, 202p.
- [60] Kitchenham, B.; Charters, S. "Guidelines for performing systematic literature reviews in software engineering", *Software Engineering Group Department of Computer Science Keele University*, vol. 2, Jan 2007, pp. 512–523.
- [61] Kitchenham, B.; Pfleeger, S. L. "Principles of survey research part 4: questionnaire evaluation", *Software Engineering Notes*, vol. 27, Jan 2002, pp. 20–23.
- [62] Kitchenham, B.; Pfleeger, S. L. "Principles of survey research: part 5: populations and samples", *Software Engineering Notes*, vol. 27, Set 2002, pp. 17–20.
- [63] Kitchenham, B.; Pfleeger, S. L. "Principles of survey research part 6: data analysis", *Software Engineering Notes*, vol. 28, Mar 2003, pp. 24–27.
- [64] Kitchenham, B. A.; Pfleeger, S. L. "Principles of survey research part 2: designing a survey", *Software Engineering Notes*, vol. 27, Mai 2002, pp. 18–20.
- [65] Kitchenham, B. A.; Pfleeger, S. L. "Principles of survey research: part 3: constructing a survey instrument", *Software Engineering Notes*, vol. 27, Jan 2002, pp. 20–24.
- [66] Kitchenham, B. A.; Pfleeger, S. L.; Pickard, L. M.; Jones, P. W.; Hoaglin, D. C.; El Emam, K.; Rosenberg, J. "Preliminary guidelines for empirical research in software engineering", *Transactions on Software Engineering*, vol. 28, Set 2002, pp. 721–734.
- [67] Kloeckner, A. P.; Libânio, C. d. S.; Ribeiro, J. L. D.; et al.. "Design thinking methods and techniques in design education". In: *Proceedings of the International Conference on Engineering and Product Design Education, Building Community: Design Education for a Sustainable Future*, Oslo, Norway: IED, 2017, pp. 6.
- [68] Kolko, J. "Design thinking comes of age", *Harvard Business Review*, vol. 93, Set 2015, pp. 66–77.
- [69] Landis, J. R.; Koch, G. G. "The measurement of observer agreement for categorical data", *Biometrics*, vol. 33, Fev 1977, pp. 159–174.

- [70] Lee, C.-S.; Wong, K. D. "An entrepreneurial narrative media-model framework to knowledge building and open co-design for smart cities". In: Proceedings of the Computing Conference, London, England: IEEE, 2017, pp. 7.
- [71] Levy, M. "Promoting the elicitation of usability and accessibility requirements in design thinking: Using a designed object as a boundary object". In: Proceedings of the International Requirements Engineering Conference Workshops, Lisbon, Portugal: IEEE, 2017, pp. 4.
- [72] Levy, M. "Educating for empathy in software engineering course". In: Proceedings of the International Conference on Requirements Engineering: Foundation for Software Quality, Utrecht, Holland: CEUR-WS, 2018, pp. 9.
- [73] Levy, M.; Hadar, I. "The importance of empathy for analyzing privacy requirements". In: Proceedings of the International Workshop on Evolving Security & Privacy Requirements Engineering, Banff, Canada: IEEE, 2018, pp. 5.
- [74] Liikkanen, L. A.; Kilpiö, H.; Svan, L.; Hiltunen, M. "Lean ux: the next generation of user-centered agile development?" In: Proceedings of the Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, Helsinki, Finland: ACM, 2014, pp. 6.
- [75] Liikkanen, L. A.; Laakso, M.; Björklund, T. "Foundations for studying creative design practices". In: Proceedings of the Conference on Creativity and Innovation in Design, Eindhoven, Holland: ACM, 2011, pp. 7.
- [76] Lima, J.; Simões, A.; Vieira, A.; Braga, L.; Ferreira, B. "Using design thinking to design an application for recognizing the work of it professionals". In: Proceedings of the Brazilian Symposium on Software Quality, Curitiba, Brazil: ACM, 2018, pp. 9.
- [77] Linåker, J.; Sulaman, S. M.; Maiani de Mello, R.; Höst, M. "Guidelines for conducting surveys in software engineering v. 1.1". Capturado em: <https://portal.research.lu.se/portal/files/6062997/5463412.pdf>, Dezembro 2019.
- [78] Lucena, P.; Braz, A.; Chicoria, A.; Tizzei, L. "Ibm design thinking software development framework". In: Proceedings of the Brazilian Workshop on Agile Methods, Curitiba, Brazil: Springer, 2016, pp. 12.
- [79] Luchs, M. G. "A brief introduction to design thinking". John Wiley & Sons, 2015, cap. 1, pp. 1–12.
- [80] Luedeke, T. F.; Köhler, C.; Conrad, J.; Grashiller, M.; Sailer, A.; Vielhaber, M.; et al.. "Cpm/pdd in the context of design thinking and agile development of cyber-physical systems". In: Proceedings of the NordDesign, Linköping, Sweden: Design Society, 2018, pp. 11.

- [81] Malins, J.; Maciver, F. "From the Real to the Virtual: Developing Improved Software Using Design Thinking". Springer, 2016, cap. 16, pp. 337–349.
- [82] Matz, A.; Germanakos, P. "Increasing the quality of use case definition through a design thinking collaborative method and an alternative hybrid documentation style". In: Proceedings of the International Conference on Learning and Collaboration Technologies, Toronto, Canada: Springer, 2016, pp. 12.
- [83] McMahon, E. "From product development to innovation". In: Proceedings of the International Conference of the American Society for Engineering Management, Virginia Beach, United States: ASEM, 2014, pp. 11.
- [84] Murugesan, L. K.; Hoda, R.; Salcic, Z. "Identifying design features using combination of requirements elicitation techniques". In: Proceedings of the International Workshop on Design and Innovation in Software Engineering, Buenos Aires, Argentina: IEEE, 2017, pp. 7.
- [85] Mutuku, L. N.; Colaco, J. "Increasing kenyan open data consumption: A design thinking approach". In: Proceedings of the International Conference on Theory and Practice of Electronic Governance, New York, United States: ACM, 2012, pp. 4.
- [86] Nedeltcheva, G. N.; Shoikova, E. "Coupling design thinking, user experience design and agile: Towards cooperation framework". In: Proceedings of the International Conference on Big Data and Internet of Thing, London, England: ACM, 2017, pp. 5.
- [87] Newman, P.; Ferrario, M. A.; Simm, W.; Forshaw, S.; Friday, A.; Whittle, J. "The role of design thinking and physical prototyping in social software engineering". In: Proceedings of the International Conference on Software Engineering, Florence, Italy: IEEE, 2015, pp. 9.
- [88] Palacin-Silva, M.; Khakurel, J.; Happonen, A.; Hynninen, T.; Porrás, J. "Infusing design thinking into a software engineering capstone course". In: Proceedings of the Conference on Software Engineering Education and Training, Savannah, United States: IEEE, 2017, pp. 10.
- [89] Parizi, R.; Moreira da Silva, M.; de Souza Couto, I.; Pavin Trindade, K.; Plautz, M.; Marczak, S.; Conte, T.; Candello, H. "Design thinking in software requirements: What techniques to use? a proposal for a recommendation tool". In: Proceedings of the Ibero-American Conference-American on Software-American Engineering, Curitiba, Brazil: Curran Associates, 2020, pp. 14.
- [90] Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. "Systematic mapping studies in software engineering". In: Proceedings of the International Conference on Evaluation

and Assessment in Software Engineering, Bari, Italy: BCS Learning and Development Ltd, 2008, pp. 10.

- [91] Pfleeger, S. L.; Kitchenham, B. A. “Principles of survey research: part 1: turning lemons into lemonade”, *Software Engineering Notes*, vol. 26, Jan 2001, pp. 16–18.
- [92] Pham, Y. D.; Fucci, D.; Maalej, W. “A first implementation of a design thinking workshop during a mobile app development course project”. In: *Proceedings of the International Workshop on Software Engineering Education for Millennials*, Gothenburg, Sweden: IEEE, 2018, pp. 8.
- [93] Prasad, W. R.; Perera, G.; Padmini, K. J.; Bandara, H. D. “Adopting design thinking practices to satisfy customer expectations in agile practices: A case from sri lankan software development industry”. In: *Proceedings of the Moratuwa Engineering Research Conference*, Moratuwa, Sri Lanka: IEEE, 2018, pp. 6.
- [94] Pressman, R.; Maxim, B. “Engenharia de Software-8ª Edição”. McGraw Hill Brasil, 2016, 968p.
- [95] Prestes, M.; Parizi, R.; Marczak, S.; Conte, T. “On the use of design thinking: A survey of the brazilian agile software development community”. In: *Proceedings of the International Conference on Agile Software Development*, Copenhagen, Denmark: Springer, 2020, pp. 13.
- [96] Radatz, J.; Geraci, A.; Katki, F. “Ieee standard glossary of software engineering terminology”, *The Institute of Electrical and Electronics Engineers Standards*, vol. 610121990, Dez 1990, pp. 3.
- [97] Rajashekharaiyah, K. M. M.; Pawar, M.; Patil, M. S.; Kulenavar, N.; Joshi, G. “Design thinking framework to enhance object oriented design and problem analysis skill in java programming laboratory: An experience”. In: *Proceedings of the International Conference on MOOCs, Innovation and Technology in Education*, Madurai, India: IEEE, 2016, pp. 6.
- [98] Reddy, P. D.; Iyer, S.; Sasikumar, M. “Fathom: Tel environment to develop divergent and convergent thinking skills in software design”. In: *Proceedings of the International Conference on Advanced Learning Technologies*, Timisoara, Romania: IEEE, 2017, pp. 5.
- [99] Rhinow, H.; Meinel, C. “Design thinking: expectations from a management perspective”. Springer, 2014, cap. 15, pp. 239–252.
- [100] Rösel, A. “Are We Ready for Disruptive Improvement?” Springer, 2016, cap. 5, pp. 77–91.

- [101] Sandino, D.; Matey, L. M.; Vélez, G. "Design thinking methodology for the design of interactive real-time applications". In: Proceedings of the International Conference of Design, User Experience, and Usability, Las Vegas, United States: Springer, 2013, pp. 10.
- [102] Santos, H. R. M.; Alves, C. "A2bp: A method for ambidextrous analysis of business process". In: Proceedings of the International Conference on Enterprise Information Systems, Porto, Portugal: SCITEPRESS, 2017, pp. 12.
- [103] Sarbazhosseini, H.; Adikari, S.; Keighran, H. "Design thinking framework for project portfolio management". In: Proceedings of the International Conference of Design, User Experience, and Usability, Toronto, Canada: Springer, 2016, pp. 8.
- [104] Seyff, N.; Ollmann, G.; Bortenschlager, M. "irequire: Gathering end-user requirements for new apps". In: Proceedings of the International Requirements Engineering Conference, Trento, Italy: IEEE, 2011, pp. 2.
- [105] Sohaib, O.; Solanki, H.; Dhaliwa, N.; Hussain, W.; Asif, M. "Integrating design thinking into extreme programming", *Ambient Intelligence and Humanized Computing*, vol. 10, Jun 2019, pp. 2485–2492.
- [106] Soledade, M. P. J.; Freitas, R. S.; Peres, S. M.; Fantinato, M.; Steinbeck, R.; Araújo, U. F. "Experimenting with design thinking in requirements refinement for a learning management system". In: Anais do Simpósio Brasileiro de Sistemas de Informação, João Pessoa, Brasil: SBC, 2013, pp. 12.
- [107] Sosnin, P. "Question-answer analysis in design thinking at the conceptual stage of developing a system with a software". In: Proceedings of the International Conference on Information, Intelligence, Systems Applications, Larnaca, Cyprus: IEEE, 2017, pp. 6.
- [108] Souza, A. F.; Ferreira, B.; Valentim, N.; Conte, T. "An experience report on teaching multiple design thinking techniques to software engineering students". In: Proceedings of the Brazilian Symposium on Software Engineering, São Carlos, Brazil: ACM, 2018, pp. 10.
- [109] Stanford, D.-S. "Design thinking - bootcamp bootleg compilation". Capturado em: <https://dschool.stanford.edu/resources/the-bootcamp-bootleg>, Fevereiro 2020.
- [110] Stanford, D.-S. "The scrum guide™". Capturado em: <https://www.scrumguides.org/scrum-guide.html>, Fevereiro 2020.
- [111] Steinbeck, R. "Building creative competence in globally distributed courses through design thinking", *Comunicar*, vol. 19, Out 2011, pp. 27–34.

- [112] Tellioğlu, H. “Models as bridges from design thinking to engineering”. In: Proceedings of the International Conference on Interfaces and Human Computer Interaction, Madeira, Portugal: IADIS, 2016, pp. 4.
- [113] Tschimmel, K. “Design thinking as an effective toolkit for innovation”. In: Proceedings of the International Society for Professional Innovation Management, Barcelona, Spain: ISPIM, 2012, pp. 1.
- [114] Unnikrishnan, R.; Hebert, M. “Measures of similarity”. In: Proceedings of the Workshops on Applications of Computer Vision, Washington D.C., United States: IEEE, 2005, pp. 1.
- [115] Vetterli, C.; Brenner, W.; Uebernickel, F.; Petrie, C. “From palaces to yurts: Why requirements engineering needs design thinking”, *Internet Computing*, vol. 17, Mar 2013, pp. 91–94.
- [116] Vianna, M. “Design thinking: inovação em negócios”. Design Thinking, 2012, 195p.
- [117] Viera, A. J.; Garrett, J. M.; et al.. “Understanding interobserver agreement: the kappa statistic”, *Family Medicine*, vol. 37, Jun 2005, pp. 360–363.
- [118] Weigel, L. “Design thinking to bridge research and concept design”. John Wiley & Sons, 2015, cap. 5, pp. 59–70.

APÊNDICE A – LISTA DE ARTIGOS SELECIONADOS

#id	Classificação	Base	Estudos Seleccionados/Extraídos	Publicado em?	Ano	Autores
1	Semelhanças ágeis	ACM	Design Thinking and Agile Practices for Software Engineering: An Opportunity for Innovation	SIGITE '18 Proceedings of the 19th Annual SIG Conference on Information Technology Education	2018	Luis Corral e Ilenia Fronza
2	Gerar Soluções/Ideias	ACM	Increasing Kenyan Open Data Consumption: A Design Thinking Approach	ICEGOV '12 Proceedings of the 6th International Conference on Theory and Practice of Electronic Governance	2012	Leonida N. Mutuku e Jessica Colaco
3	Explorar o problema, Gerar Soluções/Ideias	IEEE	Integrating the Design Thinking into the UCD's methodology	IEEE EDUCON 2010 Conference	2010	Carina S. González González, Evelio González González, Vanesa Muñoz Cruz e José Sigut Saavedra
4	Explorar o problema, Gerar Soluções/Ideias	IEEE	The Role of Design Thinking and Physical Prototyping in Social Software Engineering	ICSE '15 Proceedings of the 37th International Conference on Software Engineering	2015	Peter Newman, Maria Angela Ferrarioy, Will Simm, Stephen Forshawz, Adrian Friday e Jon Whittle
5	Educação	IEEE	Design Thinking Framework to Enhance Object Oriented Design and Problem Analysis Skill in Java Programming Laboratory: An Experience	MITE IEEE 4th International Conference on MOOCs, Innovation and Technology in Education	2016	K.M.M Rajashekaraiah, Manjula Pawar, Mahesh S Patil, Nagaratna Kulenavar e Dr. G H Joshi
6	Explorar o problema, Olhar no cliente, Melhorar a satisfação do	IEEE	Can Metamodels Link Development to Design Intent?	BRIDGE IEEE/ACM 1st International Workshop on Bringing Architectural Design Thinking Into Developers' Daily Activities	2016	M. Todd Gamble
7	Elucidar/Especificar Requisitos, Inovação	IEEE	Trends in the Use of Design Thinking for Embedded Systems	ICCSA 15th International Conference on Computational Science and Its Applications	2015	Rodolfo Araújo, Eudisley Anjos e Danielle Rousy Silva
8	Semelhanças Ágeis	IEEE	Promoting the Elicitation of Usability and Accessibility Requirements in Design Thinking: Using a Designed Object as a Boundary Object	REW IEEE 25th International Requirements Engineering Conference Workshops	2017	Meira Levy
9	Redução de incertezas	IEEE	Question-answer analysis in design thinking at the conceptual stage of developing a system with a software	IISA 8th International Conference on Information, Intelligence, Systems Applications	2017	P. Sosnin
10	Olhar no cliente, Semelhanças Ágeis	IEEE	The Agile Manifesto, design thinking and systems engineering	SysCon'17 Annual IEEE International Systems Conference	2017	M. Ann Garrison Darrin e William S. Devreux
11	Aumento da Empatia	IEEE	The Importance of Empathy for Analyzing Privacy Requirements	ESPRE IEEE 5th International Workshop on Evolving Security & Privacy Requirements Engineering	2018	Meira Levy e Irit Hadar
12	Indústria/Educação, Inovação	IEEE	Hackathons, semesterathons, and summerathons as vehicles to develop smart city local talent that via their innovations promote synergy between industry, academia, government and	ISC2 International Smart Cities Conference	2017	Avalos, Manuel e Larios, Victor M e Salazar, Pablo e Maciel, Rocio

13	Semelhanças Ageis	IEEE	An entrepreneurial narrative media-model framework to knowledge building and open co-design for smart cities	Computing Conference	2017	Lee, Chien-Sing e Wong, K Daniel
14	Elucidar/Especificar Requisitos, Redução de incertezas	IEEE	Policy Recommendations to Induce Behavioural Changes through Interactive Energy Visualisation	ISGT Asia IEEE Innovative Smart Grid Technologies-Asia	2018	Murugesan, Latha e Hoda, Rashina e Salcic, Zoran e Verma, Piyush
15	Elucidar/Especificar Requisitos, Semelhanças Ageis	IEEE	From Palaces to Yurts: Why Requirements Engineering Needs Design Thinking	Journal: IEEE Internet Computing	2013	Vetterli, Christophe e Brenner, Walter e Uebernickel, Falk e Petrie, Charles
16	Educação	IEEE	The Students' Perspectives on Applying Design Thinking for the Design of Mobile Applications	ICSE-SEET '17 Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track	2017	Valentim, Natasha M Costa e Silva, Williamson e Conte, Tayana
17	Educação, Semelhanças Ageis	IEEE	Infusing Design Thinking into a Software Engineering Capstone Course	CSEE&T IEEE 30th Conference on Software Engineering Education and Training	2017	Palacin-Silva, Maria e Khakurel, Jayden e Haponen, Ari e Hynninen, Timo e Porras, Jari
18	Educação	IEEE	Applying design thinking in disciplines of systems development	EATIS 8th Euro American Conference on Telematics and Information Systems	2016	Coutinho, Emanuel F e Gomes, George Allan M e José, ML Antonio
19	Aumento da Empatia, Inovação	IEEE	Eliciting Requirements Using Personas and Empathy Map to Enhance the User Experience	SBES '15 29th Brazilian Symposium on Software Engineering	2015	Bruna Ferreira, Tayana Conte e Simone Diniz Junqueira Barbosa
20	Elucidar/Especificar Requisitos, Validação, Inovação	IEEE	Identifying Design Features Using Combination of Requirements Elicitation Techniques	DISE IEEE/ACM 1st International Workshop on Design and Innovation in Software Engineering	2017	L. K. Murugesu, R. Hoda e Z. Salcic
21	Educação	IEEE	FATHOM: TEL Environment to Develop Divergent and Convergent Thinking Skills in Software Design	ICALT IEEE 17th International Conference on Advanced Learning Technologies	2017	P. D. Reddy, S. Iyer e M. Sasikumar
22	Educação, Gerar Soluções/Ideias	SCOPUS	A first implementation of a design thinking workshop during a mobile app development course project	SEEM IEEE/ACM International Workshop on Software Engineering Education for Millennials	2018	Y. D. Pham, D. Fucci e W. Maalej
23	Elucidar/Especificar Requisitos, Inovação	SCOPUS	A heuristic approach for supporting innovation in requirements engineering	A Heuristic Approach for Supporting Innovation in Requirements Engineering. CibSE	2011	El-Sharkawy, Sascha e Schmid, Klau
24	Processos de negócio	SCOPUS	A2BP: A method for ambidextrous analysis of business processes	ICEIS International Conference on Enterprise Information Systems	2017	Higor Monteiro Santos e Carina Alves

25	Elucidar/Especificar Requisitos, Redução de incertezas, Semelhanças	SCOPUS	Adopting design thinking practices to satisfy customer expectations in agile practices: A case from Sri Lankan software development industry	MERCon Moratuwa Engineering Research Conference	2018	Prasad, WMD Ruchira e Perera, GIUS e Padmini, KV Jeeva e Bandara, HMN Dilum
26	Explorar o problema, Olhar no cliente, Melhorar a satisfação do	SCOPUS	Aligning healthcare innovation and software requirements through design thinking	SEHS'16 Proceedings of the International Workshop on Software Engineering in Healthcare Systems	2016	Carroll, Noel e Richardson, Ita
27	Elucidar/Especificar Requisitos, Redução de incertezas	SCOPUS	An experience report on teaching multiple design thinking techniques to software engineering students	SBSE Proceedings of the XXXII Brazilian Symposium on Software Engineering	2018	Souza, Anderson Felipe e Ferreira, Bruna e Valentim, Natasha e Conte, Tayana
28	Ferramenta de software voltado ao DT	SCOPUS	CoDICE: Balancing software engineering and creativity in the co-design of digital encounters with cultural heritage	International Working Conference on Advanced Visual Interfaces	2014	Díaz, Paloma e Aedo, Ignacio e Cubas, Jaime
29	Explorar o problema, Inovação, Semelhanças Ágeis	SCOPUS	Coupling design thinking, user experience design and agile: Towards cooperation framework	Proceedings of the International Conference on Big Data and Internet of Thing	2017	Nedeltcheva, Galia Novakova e Shoikova, Elena
30	Educação, Semelhanças Ágeis	SCOPUS	CPM / PDD in the context of Design Thinking and Agile Development of Cyber-Physical Systems: Use cases and methodology	DS 91: Proceedings of NordDesign 2018	2018	Luedeke, Tobias F e Kohler, Christian e Conrad, Jan e Grashiller, Michael e Sailer, Andreas e Vielhaber, Michael
31	Inovação	SCOPUS	Design thinking for search user interface design	Euro HCIR 2011 European Workshop on Human-Computer Interaction and Information Retrieval	2011	Berger, Arne
32	Explorar o problema, Gerar Soluções/Ideias	SCOPUS	Design thinking methodology for the design of interactive real-time applications	HCI International conference of design, user experience, and usability	2013	Sandino, Diego e Matey, Luis M e Velez, Gorka
33	Gerenciamento de projeto	SCOPUS	Design thinking methods and techniques in design education	E&PED International Conference on Engineering and Product Design Education	2017	KloECKner, Ana Paula e Libânio, Cláudia de Souza e Ribeiro, José Luis Duarte
34	Inovação	SCOPUS	Design thinking methods and tools for innovation	HCI International conference of design, user experience, and usability	2015	Chasanidou, Dimitra e Gasparini, Andrea Alessandro e Lee, Eunji
35	Gerenciamento de projeto	SCOPUS	Design thinking: Expectations from a management perspective	Book: Design Thinking Research: Building Innovation Eco-Systems Springer International	2014	Rhinow, Holger e Meinel, Christoph
36	Educação, Gerar Soluções/Ideias, Elucidar/Especificar	SCOPUS	Designing human-centric information systems: Towards an understanding of challenges in specifying requirements within design thinking projects	MKWI 2018	2018	Hehn, Jennifer a e Ueberrickel, Falk e Stockli, Emanuel e Brenner, Walter

37	Educação, Inovação	SCOPUS	Developing a virtual learning object for teaching mobile web programming with design thinking	Ctrl+E 2016 Congresso Regional sobre Tecnologias na Educação	2016	da Cruz Júnior, Geraldo Gomes do Nascimento, Rafaella Leandra Souza
38	Educação, Empatia, Elucidar/Especificar Requisitos, Redução de	SCOPUS	Educating for empathy in software engineering course	REFSQ Workshops	2018	Levy, Meira
39	Experiência do usuário, Inovação	SCOPUS	Fast train to DT: A practical guide to coach design thinking in software industry	IFIP Conference on Human-Computer Interaction	2013	Hiremath, Muktha and Sathiyam, Visvapriya
40	Inovação, Tradicional para Ágil (Semelhanças Ágeis)	SCOPUS	From product development to innovation	American Society for Engineering Management (ASEM)	2014	McMahon, Ed.
41	Semelhanças ágeis, Elucidar/Especificar Requisitos, Validação	SCOPUS	Guiding novice database developers in database schema creation	IEEE Conference on e-Learning, e-Management and e-Services (IC3e)	2014	Ahmad, Rohiza, et al.
42	Olhar no cliente, Semelhanças Ágeis, Satisfação com Cliente	SCOPUS	IBM design thinking software development framework	Brazilian Workshop on Agile Methods. Springer, Cham, 2016.	2016	Lucena, Percival, et al.
43	Semelhanças Ágeis, Inovação, Explorar o problema, Gerar	SCOPUS	InnoStartup - A toolbox for innovation in software development process	IEEE Latin America Transactions	2016	Borba, Alexandre Willams Torres, Glauber Henrique Camelo Batista, and Ricardo Andre Cavalcante Souza.
44	Inovação, Semelhanças Ágeis, Explorar o problema, Olhar no cliente	SCOPUS	Integrating design thinking into extreme programming	Journal of Ambient Intelligence and Humanized Computing	2019	Osama Sohaib1, Hiralkumari Solanki1, Navkiran Dhaliwa1, Walayat Hussain1, Muhammad Asif2
45	Redução de incertezas, Inovação, Gerenciamento de projeto, Semelhanças	SCOPUS	Introducing 'Human-Centered Agile Workflow' (HCAW) – An Agile Conception and Development Process Model	Conference: International Conference on Applied Human Factors and Ergonomics	2017	Leonhard Glomann
46	Educação	SCOPUS	Juicing the game design process: towards a content centric framework for understanding and teaching game design in higher education	Educational Media International	2018	Lasse Juel Larsen
47	Semelhança Ágil	SCOPUS	Lean UX - The next generation of user-centered Agile development?	NordiCHI '14 Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational	2014	Lassi A. Liikkanen, Harri Kilpiö, Lauri Svan, and Miko Hiltunen
48	Educação, Semelhança Ágil	SCOPUS	Adding Scrum-style project management to an advanced Design Thinking class	DS 91: Proceedings of NordDesign 2018, Linköping, Sweden. Publicada pela Hasso Plattner Institute (Modelo DT - HPI)	2018	Franziska Dobrigkeit, Molly Wilson, Claudia Nicolai

49	Educação, Inovação	SCOPUS	LODPRO: learning objects development process	Journal of the Brazilian Computer Society	2016	Leandro Marques Queiros ^{1*} , Denis Silva da Silveira ² , Jorge da Silva Correia-Neto ¹ and Guilherme Vilar ¹
50	Educação, Inovação	SCOPUS	Models as bridges from design thinking to engineering	Editora- Katherine Blashki and Yingcai Xiao	2016	Hilda Telloğlu
51	Elucidar/Especificar Requisitos, Explorar o problema, Olhar no	SCOPUS	OnTimeCargo: A smart transportation system development in logistics management by a design thinking approach	PACIFIC ASIA CONFERENCE ON INFORMATION SYSTEMS (PACIS)	2016	Ahmed Azab, Noha Mostafa e Jaehyun Park
52	Semelhanças Ágeis, Gerar Soluções/Ideias, Inovação	SCOPUS	Pet empires: Combining design thinking, lean startup and agile to learn from failure and develop a successful game in an undergraduate environment	International Conference on Human-Computer Interaction	2016	Danielly F.O. de Paula e Cristiano C. Araújo
53	Experiência do usuário, Inovação, Semelhanças Ágeis	SCOPUS	Reframed contexts: Design thinking for agile user experience design	Conferencia paper - International Conference of Design, User Experience, and Usability	2013	Sisira Adikari, Craig Mcdonald e John Campbell
54	Semelhanças Ágeis	SCOPUS	Software 4.0: 'How' of building 'Next-Gen' systems	Proceedings of the 11th Innovations in Software Engineering Conference. ACM	2018	Pendse, Abhay, and Haresh Amre
55	Gerar Soluções/Ideias, Inovação	SCOPUS	The best of three worlds -The creation of innodev a software development approach that integrates design thinking, scrum and lean startup	DS 87-8 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 8: Human Behaviour in Design, Vancouver, Canada	2017	Dobrigkeit, Franziska, and Danielly de Paula
56	Elucidar/Especificar Requisitos, Gerar Soluções/Ideias, Inovação	SCOPUS	The origins of design thinking and the relevance in software innovations	International Conference on Product-Focused Software Process Improvement	2016	Jensen, Matilde Bisballe, Federico Lozano, and Martin Steinert
57	Olhar no cliente, Melhorar a satisfação do cliente, Elucidar/Especificar	SCOPUS	The use of design thinking for requirements engineering: An ongoing case study in the field of innovative software-intensive systems	IEEE 26th International Requirements Engineering Conference (RE).	2018	Hehn, Jennifer, and Falk Uebernickel
58	Educação, Elucidar/Especificar Requisitos, Inovação, Olhar no	SCOPUS	Use of design thinking in requirements elicitation of mobile learning in virtual environments [Uso do design thinking na elicitação de requisitos de ambientes virtuais de	WER. 2014.	2014	Cynara Lira de Carvalho Souza, Carla Silva
59	Gerar Soluções/Ideias, Olhar no Cliente	SCOPUS	Using design thinking to design an application for recognizing the work of IT professionals	Proceedings of the 17th Brazilian Symposium on Software Quality. ACM	2018	Juliana Lima, Adriano Simões, Andréia Vieira, Luis Braga e Bruna Ferreira
60	Elucidar/Especificar Requisitos	SCOPUS	Visual analytics for cyber-physical systems development: Blending design thinking and systems thinking	NordDesign 2018	2018	Gürdür, Didem, and Martin Törngren.

61	Explorar o problema, Gerar Soluções/Ideias	Springer	MELCA - Customizing Visualizations for Design Thinking	Domain-specific conceptual modeling.	2016	Hawryszkiewicz, Igor Titus, and Christoph Prackwieser
62	Olhar no cliente, inovação, Semelhanças Ágeis, Inovação	Springer	An Integrated Framework for Design Thinking and Agile Methods for Digital Transformation	International Conference of Design, User Experience, and Usability. Springer, Cham,	2016	Gurusamy, Kavitha, Narayanan Srinivasaraghavan, and Sisira Adikari
63	Semelhanças ágeis	Springer	Are We Ready for Disruptive Improvement?	Managing Software Process Evolution, Springer, Cham	2016	Rösel, Andreas
64	Semelhanças Ágeis	Springer	Communication Breakdowns in the Integration of User-Centred Design and Agile Development	Integrating User-Centred Design in Agile Development. Springer	2016	Bordin, Silvia, and Antonella De Angeli.
65	Semelhanças Ágeis	Springer	Increasing the Quality of Use Case Definition Through a Design Thinking Collaborative Method and an Alternative Hybrid Documentation Style	International Conference on Learning and Collaboration Technologies. Springer	2016	Matz, Alexandra, and Panagiotis Germanakos.
66	Elucidar/Especificar Requisitos, Melhorar a satisfação do cliente,	Springer	Developing High-Performing Teams: A Design Thinking Led Approach	International Conference of Design, User Experience, and Usability. Springer, Cham,	2016	Keighran, Heath, and Sisira Adikari.
67	Explorar o problema, Empatia, Gerar Soluções/Ideias, Inovação	Springer	Embed Design Thinking in Co-Design for Rapid Innovation of Design Solutions	International Conference of Design, User Experience, and Usability. Springer, Cham,	2016	Adikari, Sisira, Heath Keighran, and Hamed Sarbazhosseini.
68	Gerar Soluções/Ideias, Elucidar/Especificar Requisitos	Springer	From the Real to the Virtual: Developing Improved Software Using Design Thinking	Collaboration in Creative Design. Springer,	2016	Malins, Julian, and Fiona Maciver.
69	Elucidar/Especificar Requisitos,	Springer	Using Design Thinking for Requirements Engineering in the Context of Digitalization and Digital Transformation: A Motivation and an Experience Report	The Essence of Software Engineering. Springer, Cham, 2018	2018	Carell, Angela, Kim Lauenroth, and Dirk Platz
70	Elucidar/Especificar Requisitos	Springer	Design Thinking Framework for Project Portfolio Management	International Conference of Design, User Experience, and Usability	2016	Sarbazhosseini, Hamed, Sisira Adikari, and Heath Keighran
71	Inovação, Semelhanças Ágeis, Explorar o problema, Gerar Soluções/Ideias	Springer	Integrating Both User-Centered Design and Creative Practices into Agile Development	Integrating User-Centred Design in Agile Development	2016	Cockton, Gilbert
72	Semelhanças Ágeis	Springer	Integrating User-Centred Design in Agile Development	Integrating User-Centred Design in Agile Development. Springer,	2016	Cockton, G., Lárusdóttir, M., Gregory, P., & Cajander, A

73	Semelhanças Ágeis	Wiley	A Brief Introduction to Design Thinking	Design thinking: New product development essentials from the PDMA	2015	Luchs, Michael G
74	Inovação, Educação	Wiley	Framing Design Thinking: The Concept in Idea and Enactment	Creativity and Innovation Management	2016	Carlgren, Lisa, Ingo Rauth, and Maria Elmquist
75	Semelhanças Ágeis, Inovação	Wiley	Effective Design Methodologies	Design Management Review	2018	By Dr Geanie Asante
76	Elucidar/Especificar Requisitos, Olhar no Usuário, Explorar o	Wiley	Integrating cell and molecular biology concepts: Comparing learning gains and self - efficacy in corresponding live and virtual undergraduate laboratory experiences	Biochemistry and Molecular Biology Education	2018	Goudsouzian, Lara K., et al.
77	Gerar Soluções/Ideias	Science Direct	Applying design thinking methods to ecosystem management tools: Creating the Great Lakes Aquatic Habitat Explorer	Marine Policy 69 (2016)	2016	Goodspeed, Robert, et al.
78	Semelhanças Ágeis	Science Direct	Advanced agile approaches to improve engineering activities	Procedia Manufacturing 25	2018	Burchardt, Carsten, and Bettina Maisch

APÊNDICE B – SURVEY - INSTRUMENTO DE COLETA

28/12/2019

Qualtrics Survey Software

Bloco de perguntas padrão

oiá!

Sou aluno do Pós-Graduação da PUCRS, pesquisando sobre o uso de Design Thinking no Desenvolvimento de Software. Nosso estudo busca entender quais modelos de processo, técnicas e ferramentas estão sendo utilizados nesse contexto e em quais situações.

Se você é um profissional da indústria que usa Design Thinking no desenvolvimento de software, gostaríamos de saber a sua opinião sobre o assunto. Deve levar de 5 a 10 minutos para você expressar a sua opinião para as 11 perguntas que temos. Estamos divulgando nosso convite publicamente, então pode ser pela recomendação de um colega que nosso convite chegue até você.

É importante que você saiba que os resultados serão anonimizados, ou seja, não temos nenhum identificador que revele a sua identidade. Também não usaremos os resultados de forma individualizada. Nos interessa a visão consolidada entre os participantes. Não temos interesse comercial nos resultados. Nosso estudo é acadêmico.

Enviaremos os resultados para você caso você expresse o interesse em receber a visão consolidada uma vez que encerrarmos a coleta.

Se durante a resposta do questionário você mudar de ideia e não quiser mais participar, você pode simplesmente encerrar a página ou abandoná-la e seus dados não serão considerados.

Desde já agradecemos muito a sua participação e pedimos que você encaminhe para colegas que atendam o perfil se assim achar adequado.

Será de grande valia a sua ajuda!

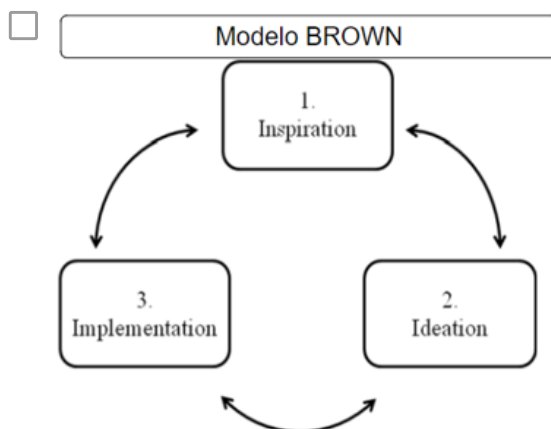
Matheus Plautz Prestes (Aluno de Mestrado) - Contato: matheus.plautz@edu.pucrs.br

Rafael Parizi (Aluno de Doutorado) - Contato: rafael.parizi@edu.pucrs.br

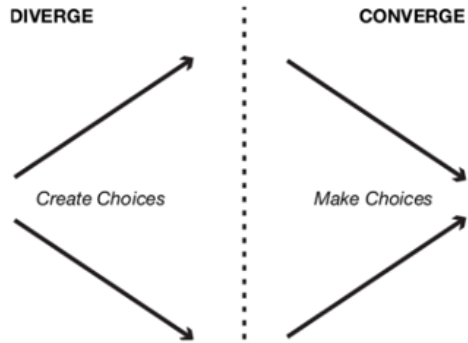
Sabrina Marczak (Orientadora, Professora Adjunta) - Contato: sabrina.marczak@pucrs.br

1) Existem diversos modelos de processo, que abstraem os espaços de trabalho no uso de Design Thinking. Você utiliza algum desses modelos como referência nas suas atividades?

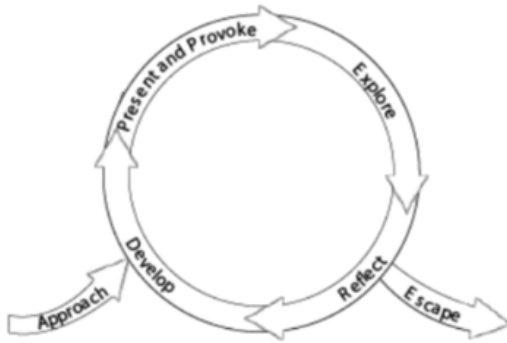
Marque todas opções que se aplicar.



Modelo Divergente e Convergente



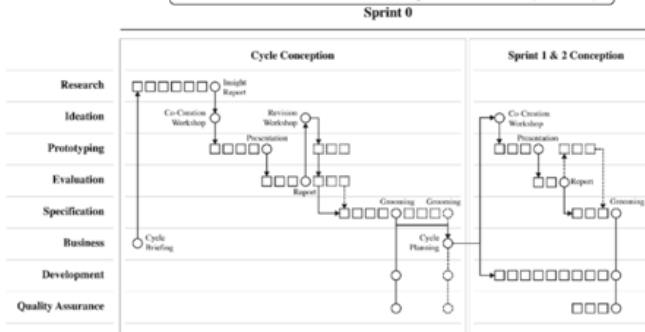
Modelo Diving Drive

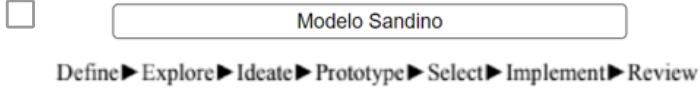
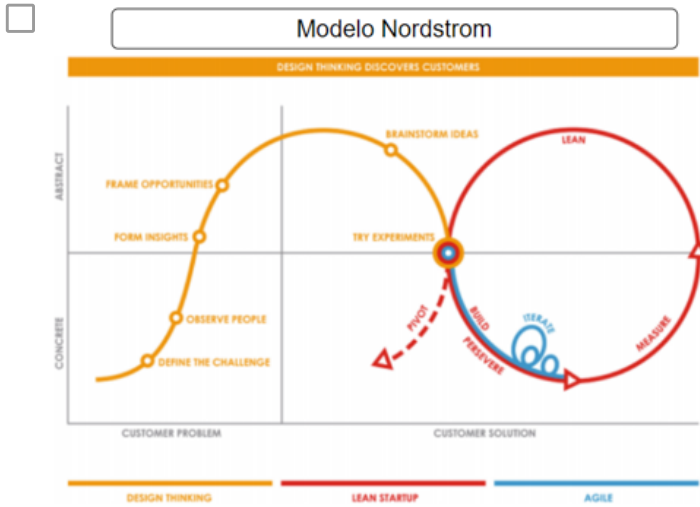
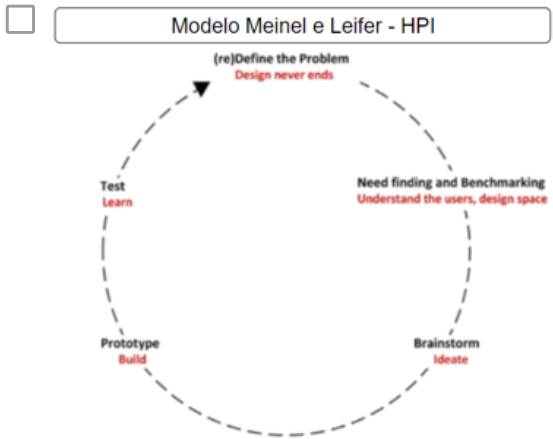
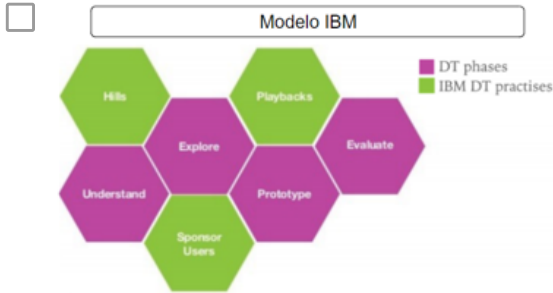


Modelo Hasso-Plattner-Institut (HPI)

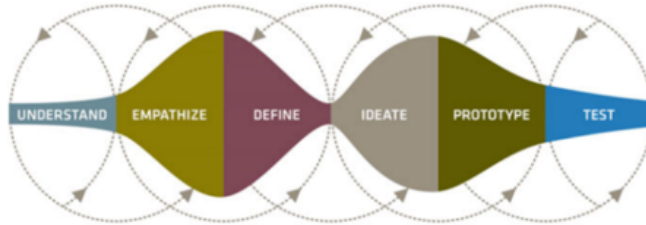


Modelo Human Centered Agile Workflow (HCAW)

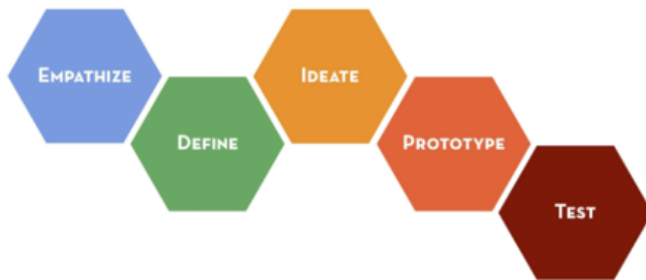




Modelo Stanford D-School e HPI



Modelo Stanford D-School



Outro?

Não sei informar.

2) Diversas técnicas podem ser usadas para apoiar o uso de Design Thinking. Quais técnicas você costuma utilizar?

Marque todas as opções que se aplicar.

- 5w2h
- A day in the life
- Behaviour Archaeology
- Blueprint
- BodyStorming
- Brainstorming

- Brainwriting
- Business Model Canvas (BMC)
- Card Sort
- Character Profiles
- Co-creation Workshops
- Cognitive and Behaviour Map
- Conceptual mind maps
- Costumer Journey Maps
- Desk Research
- Empathy Maps
- Error Analysis
- Extreme User interviews
- fishbowl
- Five Whys?
- Fly on the wall
- How-Might-We-Question
- I like I wish
- Insight cards
- Interview
- Living Labs
- Mind Mapping
- Mockups
- Narration
- Observation
- Personal Inventory
- Personas
- Positioning Matrix
- Rapid Ethnography
- Role-Playing

- Scenarios Mapping
- Shadowing
- Social Network Mapping
- Stakeholder Mapping
- Storyboard
- Storytelling
- Surveys Questionnaires
- Try it yourself
- Feedback grid
- User story
- World Café
- Não utilizo essas técnicas
- Outras?

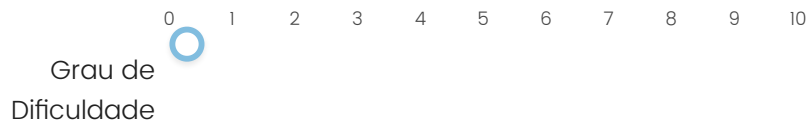
3) Como você geralmente decide quais técnicas utilizar?

Marque todas as opções que se aplicar.

- Baseado na minha experiência prévia
- Por indicação de um colega
- Recomendação pela minha empresa
- Escolho as técnicas de acordo com a espaço/etapa do DT, onde cada espaço/etapa possui suas próprias técnicas
- Quando a técnica se enquadra na minha necessidade
- Já tenho o meu catálogo de técnicas que sempre utilizo

- Depende muito do contexto que vou utilizar
- Geralmente preciso estudar as técnicas pois nunca sei qual é a melhor para o momento
- Outro motivo?

4) Em uma escala de 0 (Nenhuma dificuldade) a 10 (Dificuldade extrema), qual o grau de dificuldade que você sente em decidir quais técnicas utilizar em uma determinada situação?



5) Você utiliza algum software (ou sistema computacional, como preferir denominar) para apoiar o uso das técnicas de Design Thinking?
Marque todas as opções que se aplicar.

- Smaply (Personas)
- Circle (Stakeholder Map)
- Touchpoint Dashboard (Customer Journey)
- Creately (Service Blueprint)

- Strategyzer (Business Model Innovation)
- Axure RP (Rapid Prototype)
- Outras?

Bloco 2

6) Para qual fim você usa Design Thinking no desenvolvimento de software?

Marque todas as opções que se aplicar.

- Ganhar a empatia dos usuários
- Para o desenvolvimento de jogos
- Entender e especificar os requisitos
- Explorar e entender o problema
- Para gerar ideias e soluções
- Para gerenciar projetos
- Para criar ideias inovadoras
- Aproximar a equipe de desenvolvimento com o cliente
- Melhorar a satisfação do cliente
- Para reduzir as incertezas
- Fácil relacionamento com os métodos ágeis
- Validação do software

Outro?

7) Quais são os cenários de uso comuns onde você usa o Design Thinking?

Marque todas as opções que se aplicar.

- Com equipes multidisciplinares
- Criação de produtos/softwarees inovadores
- Dentro de um processo desenvolvimento de software diariamente/semanalmente, acompanhado por toda equipe (do cliente até o desenvolvedor)
- Inovação como um todo, desde o processo de desenvolvimento até o software
- Criar cocriação entre os participantes do projeto
- Mudanças e melhorias no desenvolvimento de software
- Utilizado em parceria com os métodos ágeis (Lean, Scrum)
- Outro?

8) Na sua experiência de uso de Design Thinking no desenvolvimento de software, o que você apontaria

como benefícios ou pontos positivos trazidos pela adoção da abordagem?

9) E quais seriam as dificuldades ou os pontos negativos?

Bloco 3

10) Qual a sua experiência, em anos, nos seguintes critérios?

	Menos de 1 ano de experiência	De 1 a 3 anos	De 4 a 7 anos	Acima de 8 anos
No uso de Design Thinking no desenvolvimento de software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Na área de TI de forma geral	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

11) Qual o seu papel ou função organizacional atual?

- Facilitador
- Pesquisador
- Design de interação
- UX/UI Designer
- Desenvolvedor
- Especialista
- Analista
- Agile Coach
- Consultor
- Product Owner
- Engenheiro
- Outro?

Bloco FINAL

(Opcional) Caso tenha interesse, por favor informe o seu e-mail para que possamos enviar lhe enviar os resultados consolidados desta pesquisa. Obrigada!

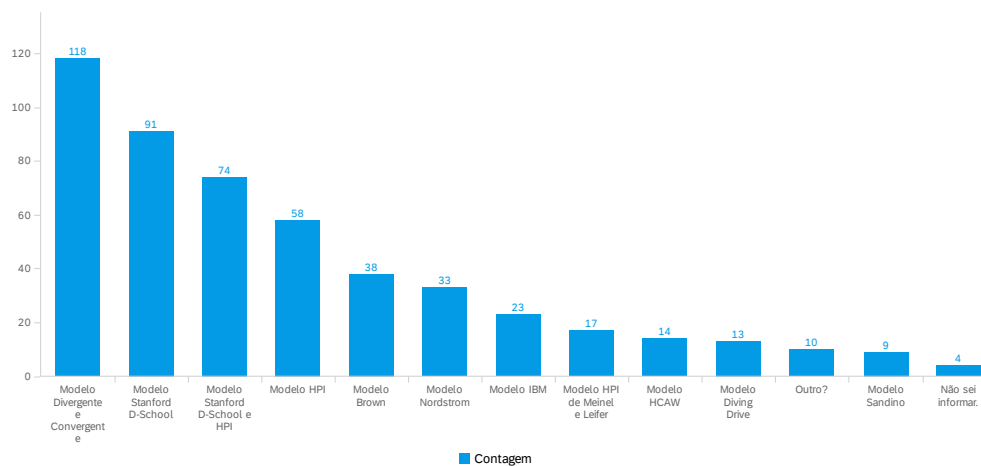
APÊNDICE C – SURVEY - RESPOSTAS

Relatório de Respostas

[PRD] Estudo Exploratório sobre Design Thinking no Desenvolvimento de Software

20 de Fevereiro de 2020 às 00:23 ART

S1 - 1) Existem diversos modelos de processo, que abstraem os espaços de trabalho no uso de Design Thinking. Você utiliza algum desses modelos como referência nas suas atividades? Marque todas opções que se aplicar.



Outro?

Outro?
▼

Cara, não há um modelo formal que é seguido. O que se faz é no mercado é muito mais parecido com o modelo Sandino, contudo não há problema algum em inserir no modelo um pouco de Lean Startup, Lean UX. Vale ressaltar que a todo instante usa-se dados para tomada de decisão, desse modo é possível se tomar decisões quantitativas coerentes com os insights ou hipóteses definidas;

Circular design e double diamond

Design Council

Design Sprint - Google Venture

Double Diamond

MIT approach

Outro?
▼

Modelo Fractal não linear: Discover -> Build -> Measure -> Learn, ou seja, em cada uma dessas etapas existe a possibilidade de todas estarem presentes, e não necessariamente acontecer de forma linear (como demonstrado nos modelos de Stanford e Hasso)

Modelo usado pela empresa ECHOS e Modelo criado internamente (CPQD)

Utilizo no dia a dia o modelo da Standford School com Convergência e Divergência nas etapas Definir e Idear. Ambas possuem o momento de divergência e depois convergência.

double diamond

S2 - 2) Diversas técnicas podem ser usadas para apoiar o uso de Design Thinking.

Quais técnicas você costuma utilizar? Marque todas as opções que se aplicar.

#	Campo	Contagem da escolha
6	Brainstorming	5,77% 151
32	Personas	5,70% 149
16	Empathy Maps	4,70% 123
14	Customer Journey Maps	4,51% 118
8	Business Model Canvas (BMC)	4,13% 108
41	Storytelling	4,13% 108
25	Interview	4,05% 106
45	User story	3,94% 103
11	Co-creation Workshops	3,75% 98
30	Observation	3,67% 96
40	Storyboard	3,56% 93
28	Mockups	3,44% 90
4	Blueprint	3,02% 79
20	Five Whys?	3,02% 79
15	Desk Research	2,75% 72
27	Mind Mapping	2,75% 72
39	Stakeholder Mapping	2,68% 70
1	5w2h	2,48% 65
42	Surveys Questionnaires	2,48% 65
13	Conceptual mind maps	2,18% 57
22	How-Might-We-Question	2,18% 57
37	Shadowing	2,06% 54
7	Brainwriting	1,87% 49
9	Card Sort	1,83% 48
23	I like I wish	1,83% 48

#	Campo	Contagem da escolha
18	Extreme User interviews	1,64% 43
36	Scenarios Mapping	1,61% 42
2	A day in the life	1,53% 40
44	Feedback grid	1,30% 34
10	Character Profiles	1,19% 31
12	Cognitive and Behaviour Map	1,07% 28
24	Insight cards	1,07% 28
29	Narration	0,96% 25
46	World Café	0,96% 25
19	fishbowl	0,88% 23
35	Role-Playing	0,80% 21
34	Rapid Ethnography	0,69% 18
33	Positioning Matrix	0,65% 17
43	Try it yourself	0,57% 15
26	Living Labs	0,50% 13
5	BodyStorming	0,34% 9
31	Personal Inventory	0,34% 9
38	Social Network Mapping	0,34% 9
3	Behaviour Archaeology	0,27% 7
17	Error Analysis	0,27% 7
92	Outras?	0,27% 7
21	Fly on the wall	0,19% 5
91	Não utilizo essas técnicas	0,08% 2

2616

Exibindo linhas 1 - 49 de 49

Outras?

Is / Isn't / Do / Don't; User Story Mapping;

Outras?

Diversas dinâmicas de "quebra-gelo"

Value proposition Canvas, Hypothesis Cards, Test Cards, Ecosystem Maps e etc

Entrevistas contextuais

Problem Tree, Crazy 8, Crazy 4, Dot Voting.

Chain of Undesired Phenomena

Job Story

S3 - 3) Como você geralmente decide quais técnicas utilizar? Marque todas as opções que se aplicar.

#	Campo	Contagem da escolha
5	Quando a técnica se enquadra na minha necessidade	22,72% 137
7	Depende muito do contexto que vou utilizar	21,72% 131
1	Baseado na minha experiência prévia	21,23% 128
4	Escolho as técnicas de acordo com a espaço/etapa do DT, onde cada espaço/etapa possui suas próprias técnicas	17,58% 106
2	Por indicação de um colega	4,81% 29
3	Recomendação pela minha empresa	4,31% 26
6	Já tenho o meu catálogo de técnicas que sempre utilizo	4,31% 26
8	Geralmente preciso estudar as técnicas pois nunca sei qual é a melhor para o momento	2,65% 16
9	Outro motivo?	0,66% 4
		603

Exibindo linhas 1 - 10 de 10

Outro motivo?

Outro motivo?

Depende do objetivo do projeto, tempo disponível e recursos financeiros

Disponibilidade de ter um designer/ alguém que ja tenha experimentado as técnicas na primeira vez que vou usar.

Geralmente isto é definido junto com a Agile Master do time, que costuma facilitar essas interações todas.

Não existe sequencia certa. O melhor é combinar mindset e as ferramentas mais adequadas para cada cliente e situação.

S4 - 4) Em uma escala de 0 (Nenhuma dificuldade) a 10 (Dificuldade extrema), qual o grau de dificuldade que você sente em decidir quais técnicas utilizar em uma determinada situação?

#	Campo	Mínimo	Máximo	Média	Desvio padrão	Variância	Contagem
1	Grau de Dificuldade	0.00	10.00	4.50	2.18	4.74	171

S5 - 5) Você utiliza algum software (ou sistema computacional, como preferir denominar) para apoiar o uso das técnicas de Design Thinking? Marque todas as opções que se aplicar.

#	Campo	Contagem da escolha
7	Outras?	48,46% 110
1	Smapply (Personas)	14,10% 32
5	Strategyzer (Business Model Innovation)	11,89% 27
6	Axure RP (Rapid Prototype)	9,25% 21
3	Touchpoint Dashboard (Customer Journey)	8,81% 20
4	Creately (Service Blueprint)	4,85% 11
2	Circle (Stakeholder Map)	2,64% 6
		227

Exibindo linhas 1 - 8 de 8

Outras?

Outras?

Não

Não

Nenhum

Bizagi

Miro, Sketch, Figma, Mural.co, Mindmeister

Mural

Adobe XD Prototype

Nao utilizo

Nenhuma

Miro

Outras?

Costumo utilizar o Illustrator para os canvas

Não utilizo software

nenhuma das opções

Miro

Mural.ly só para sessões remotas

Nope

Nenhum

Mural.ly, Miro

figma (protótipos)

Nenhum

Nenhum software. Utilizo papel, quadro e canetas.

não

Mural para atividades online (<https://mural.co/>)

Nao uso

Não utilizo

Não utilizo.

Balsamiq

Miro, Sketch

Miro

Invision, Mural, Sketch

Não costumamos utilizar ferramentas de mercado. Temos os Canvas necessários impressos em folha A0, e acaba sendo tudo no post-it e canetinha mesmo. Se mostrou mais eficiente no nosso contexto. Parece que gerou mais engajamento e envolvimento do time.

Microsoft office, Google drive, invision, figma, skype, XMind,

Não utilizo, tudo no papel, porém faço um report digital depois com fotos, etapas e até transcrição de post-its.

Invision e Miro

Outras?

Nao utilizo

miro

Nenhum

não utilizo

uso o Miro, caso tenha que documentar ou mover o que fiz em post its

build.me

mural.ly

Utilizo outros sistemas para criar Wireframes e Mockups (Adobe XP). Utilizo também papelão e outros materiais para fazer os protótipos. Tudo depende da etapa e do contexto do time que está na sessão de DT.

Não utilizo

InVision

Invision

Não

As vezes Mural

e o velho e bom post it

Salvo raros casos, utilizo ferramentas simples como pencil, paint e ppt.

postit :)

Não utilizo.

Sketch

Miro, Skecth, Google Presnetation. De modo geral os assets dos DTs são perecíveis

Não uso nenhum sistema computacional, papel canetas e post ir.

Não utilizo nenhum software.

Miro

Mural.co

Miro

Outras?

Excell, Power point

Invision e Figma

Miro, Figma

Nenhuma

Não, normalmente uso modelos analógicos.

Não

Mural

Xmind - Stakeholder Map

Miro, Whimsical, Figma

Mural

não

Real Time

Não utilizo

Muraly

Mural

Não uso software, geralmente faço na mão

Marvelapp (rapid prototype)

Ferramentas de mind map, em geral.

Utilizo 'Build' para prototipação 'Canvanizer' for Business Model

Não

Marvel app para prototipação, Paint, Evybo...

Não costumo utilizar

Não utilizo

Mural, Miro

Outras?

Nenhum recurso digital

NDA

Google Sheets

Não uso

Nenhum

Não utilizo

Nenhuma

Não uso softwares

Nao

não uso nenhum software específico. Aliás, o DT propõe que façamos as coisas em papel e caneta, para estimularmos a criatividade e aí sim os resultados finais podem ser documentados via software.

Nao uso software

SAP build

Geralmente faço manual, gostei das referencia citas.

não uso

Costumo usar post-it e caneta para a execução. Raramente durante a dinâmica uso alguma ferramenta digital, mas já usei para documentar o X-mind.

Não utilizo

Ferramentas de mind maps

nenhuma

Nenhuma dessas. Gostei das dicas ;)

Marvel

Adobe XD e google presentation

Marvel/POP

minminster, miro.com, whimshical

Figma para transformar as informações em conteúdos visuais. (gráficos, diagramas, etc)

Outras?

XD e Invision - Prototype Miro - MindMap Google, Hotjar e Survey Monkey - Surveys

Illustrator e Photoshop

S6 - 6) Para qual fim você usa Design Thinking no desenvolvimento de software?

Marque todas as opções que se aplicar.

#	Campo	Contagem da escolha
5	Para gerar ideias e soluções	14,79% 147
4	Explorar e entender o problema	14,29% 142
7	Para criar ideias inovadoras	10,97% 109
10	Para reduzir as incertezas	10,97% 109
3	Entender e especificar os requisitos	10,36% 103
9	Melhorar a satisfação do cliente	9,15% 91
8	Aproximar a equipe de desenvolvimento com o cliente	8,75% 87
11	Fácil relacionamento com os métodos ágeis	6,94% 69
1	Ganhar a empatia dos usuários	6,24% 62
12	Validação do software	4,63% 46
6	Para gerenciar projetos	2,01% 20
2	Para o desenvolvimento de jogos	0,60% 6
13	Outro?	0,30% 3
		994

Exibindo linhas 1 - 14 de 14

Outro?

Outro?

Para mapear a experiência do cliente ou de um funcionário em suas jornadas ou em treinamentos de mudança de comportamento

Planejamento Estratégico Problemas Industriais Problemas Complexos Avaliação, ajustes e melhoria de processos Entre outras

Criação e melhoria de processos internos

S7 - 7) Quais são os cenários de uso comuns onde você usa o Design Thinking?

Marque todas as opções que se aplicar.

#	Campo	Contagem escolt
1	Com equipes multidisciplinares	20,27%
7	Utilizado em parceria com os métodos ágeis (Lean, Scrum)	17,12%
2	Criação de produtos/softwarees inovadores	16,52%
5	Criar cocriação entre os participantes do projeto	15,17%
4	Inovação como um todo, desde o processo de desenvolvimento até o software	12,46%
6	Mudanças e melhorias no desenvolvimento de software	9,91%
3	Dentro de um processo desenvolvimento de software diariamente/semanalmente, acompanhado por toda equipe (do cliente até o desenvolvedor	7,66%
8	Outro?	0,90%

Exibindo linhas 1 - 9 de 9

Outro?

Trabalho em ciclos trimestrais, onde a cada ciclo nós paramos e refletimos sobre o andamento do trabalho. Neste momento, utilizamos algumas técnicas de DT para apoiar a organização do trabalho da equipe.

No geral, utilizo no começo de projetos.

Planejamento estratégico.

Quando o problema/oportunidade não está completamente claro. Em momentos que precisa-se de uma investigação para detalhar o caso.

Concepção de novos projetos

Melhoria de processos internos.

S8 - 8) Na sua experiência de uso de Design Thinking no desenvolvimento de software, o que você apontaria como benefícios ou pontos positivos trazidos pela adoção da abordagem?

8) Na sua experiência de uso de Design Thinking no desenvolvimento de softw...

Não trabalho diretamente com projetos de desenvolvimento de software - meu trabalho é sempre com foco na mudança de comportamento organizacional e transformação cultural.

O maior benefício é trazer o cliente para a rotina de criação de software e oferecer a oportunidade da equipe de desenvolvimento imergir na rotina de negócio do cliente. Assim o produto será exatamente o que o cliente necessita e o trabalho da equipe será totalmente motivado em alcançar este objetivo.

Entendo que o DT compartilha com todos os envolvidos, (TI , Business, Customers) todos os desafios envolvidos. Adequação de processos, inovação tecnológica, experiência do usuário. Com isso o software entregue de fato "agrega valor" , atendendo as expectativas dos envolvidos.

User centric, empatia

Entender as reais necessidades dos usuários

Aumenta o espírito de equipe e colaboração; Maior assertividade na solução de problemas e entendimento das necessidades

Redução do tempo de desenvolvimento, Entendimento dos problemas reais

Engajamento do time que vê a solução não como algo externo, mas feito por ele mesmo.

Identificar e experimentar erros e corrigi-los mais rápido.

Comunicação e melhor entendimento e alinhamento dos problemas

Soluciona um problema.

Dar voz a todos os envolvidos; conhecer diferentes interpretações sobre um problema; criar diferentes hipóteses e identificar a maneira mais simples possível para testa-las

Desenvolvimento focado nos problemas reais do usuário. Sensibilização, através da co-criação, de toda a cadeia de desenvolvimento do que gerou aquele software/programa. Prototipação mais rápida, o que permite o teste com usuários reais para que haja as iterações necessárias.

Facilidade de consenso entre áreas.

o principal benefício para mim é a aproximação da equipe com o business/problema do cliente.

Co-criação a partir do mapeamento de problemas e oportunidades reais

Engajamento do time no processo de desenvolvimento.

8) Na sua experiência de uso de Design Thinking no desenvolvimento de softw...

Ideias mais bem elaboradas, pensamento fora da "caixa" e assertividade de entregas

nnn

Um aprofundamento do assunto ajuda a encontrar soluções ou desvendar problemas que não estavam tão na cara. Descobrir as necessidades para inovar o melhor para o cliente final

Validar as dores reais dos usuários

Agilidade na coleta e definição dos dados

Proximidade com clientes, redução de riscos, facilidade no processo de vendas (quando usado para esse propósito)

Identificação prévia de cenários não mapeados, Facilita na priorização de features e backlog pois traz a prioridade do usuário para o centro da mesa, melhora a tomada de decisões relacionadas a fluxo e layout pois traz dados de usuários reais para análise, Traz olhos frescos ao produto e evita problemas no fluxo que geralmente são ocasionados por olhares viciados

Co-criação

Entendimento da perspectiva do usuário, definição do problema, ideação, classificação de potenciais soluções.

melhor entendimento de todos os envolvidos no projeto e maior engajamento da equipe

Diminui os riscos e estimular a participação da equipe

Entendimento das reais necessidades e dores do consumidor, engajamento e empatia por parte dos colaboradores e soluções mais criativas e que resolvem problemas complexos.

- Participação de pessoas com skills e background diferentes na abordagem de um problema. - Partir de um problema real antes de pensar na solução

A base do Design é melhorar algo, e para isso temos que entender para quem é, as suas necessidades, cenários de uso o que podemos construir. Ele nos ajuda a abordar todo esse processo de exploração, alternativas, construção e acompanhamento do uso.

O uso do Design Thinking facilita muito a coleta de requisitos e faz com que a solução seja construída em conjunto com o cliente (não só a pessoa que assina o contrato em si, mas com diversos cargos que serão afetados pela solução desenvolvida) e o time de desenvolvimento. No Scrum também fazemos pedaços de sessões de DT em retrospectivas, para que o time consiga entender o problema, chegar na raiz, e então gerar soluções para isto.

Processo colaborativo reforçando a responsabilidade e o sentido de dono de todos os envolvidos.

Permite uma facilitação onde todos participam e saem da base de conhecimento existente, permitindo soluções mais simples e focadas no problema que precisa ser resolvido

Soluções inovadoras (diferentes e fora da caixa), integração dos participantes e democracia

O cliente pode visualizar, previamente, o resultado do desenvolvimento.

Alinhamento entre a equipe e com o cliente.

Redução de incertezas ou de certezas equivocadas. Aprendizado rápido

8) Na sua experiência de uso de Design Thinking no desenvolvimento de softw...

Maior clareza e assertividade no desenvolvimento de a nova solução

Proximidade e geração de empatia com o cliente e usuário final. O trabalho fica mais prazeroso e as conversas entre a equipe fluem mais naturalmente, pois é muito mais fácil criar situações hipotéticas para imaginar a utilização do produto que está sendo desenvolvido. A equipe fica mais entrosada e interessada pelo que está sendo feito.

Engajamento da equipe (sentimento de cocriação traz accountability), clareza entre o time de desenvolvimento e stakeholders, especialmente pq o Design Thinking traz um espaço propício para assuntos delicados e importantes serem tratados de maneira despersonalizada e tranquila.

Facilita o entendimento da jornada do usuário. Ajuda no desenvolvimento de soluções diferenciadas para problemas recorrentes.

Entendo que independente da dinâmica e/ou modelo utilizado, as técnicas nos permitem tangibilizar/objetivar ideias muitas vezes vagas que temos em mente, bem como manter pensamentos registrados, até que percebamos que podem ser utilizados, combinados com outros ou até, que percebamos que não façam sentido. Na prática, as técnicas ajudam a criarmos modelos mentais coletivos, incrementais e dinâmicos.

Integração e alinhamento com outros times, acelerar entrega, ser mais preciso na entrega

Prototipação rápida, afinidade entre a equipe, equipe com o cliente, nivelar conhecimento, remover incertezas, com esses benefícios a consequência é normalmente mais engajamento e entregas melhores e mais pontuais. Também faço uso de outras abordagens após o DT como complemento, como Lean Inception / Sprint e após isso com processo SCRUM.

Maior engajamento do time de desenvolvimento

Entendimento compartilhado, co-criação trazendo sentimento de dono pelos envolvidos, visão do produto, entendimentos das necessidades do cliente, minimizar outputs e maximizar outcomes

principalmente a conexão entre usuário e desenvolvimento, quebrando a barreira entre "nós e eles", e mostrando a sequência lógica da construção do produto

Real entendimento do problema e qual solução se torna mais aderente.

Benefício: experimental e experiencial... O que o torna um risco e possivelmente árduo para adoção nas culturas menos inovadoras e avessas a riscos e imprevisibilidade.

Acertar na solução mais adequada às necessidades dos clientes.

Melhora a comunicação com o cliente. Facilita o entendimento do time de desenvolvimento.

método estruturado para envolvimento de todos na equipe, trazendo a visão mais ampla sobre o problema e as possíveis soluções.

Um dos Pontos positivos é exercitar a empatia e a colaboração entre os membros da equipe

Grau de assertividade junto dos usuários

Otimização do trabalho e ideias inovadoras.

Aumento na satisfação dos clientes para com as entregas. Cliente se sente dono da solução entregue, com isso o engajamento do cliente aumenta.

.

8) Na sua experiência de uso de Design Thinking no desenvolvimento de softw...

Além dos pontos citados na pesquisa, o fato de ouvir todas as ideias sem julgamento e depois através de um processo claro a todos, priorizar as melhores. Isso faz com que todos acreditem no processo, do mais alto escalão até os times da base.

Entendimento da situação.

O DT muda a perspectiva para "Estudar o problema" ao invés de ir direto pra solução. Tradicionalmente somos programados para solucionar problemas, mas essa abordagem não permite explorar ao máximo a criatividade com o que realmente dará certo e gerará valor. O estudo da perspectiva do problema e sua definição (etapa Define), adotando um POV (ponto de vista) faz com que toda a ideação e as demais fases sejam produzidas a partir desse POV. Feedback rápido, criatividade, empatia, cultura da experimentação, aprendizados constantes e inovação são outros benefícios explorados pelo DT.

Identificação de necessidades e criação de ideias que não costumam surgir de outra forma.

alinhamento da necessidade entre os envolvidos, cocriação, teste de hipóteses

Entender a perspectiva do usuário e aplicar o action plan de melhoria

Empatia e alinhamento na co-criação de soluções inovadoras com times multidisciplinares.

Criar empatia para com as pessoas que usaram a solução.

Entendimento do negócio

A abordagem, assim como citado acima, aproxima o patrocinador que de fato está interessado, junto a equipe de desenvolvimento e também aproxima a equipe do cliente. Com essa equipe próxima, o entendimento do problema pode ser melhor esclarecido, assim como a definição dos requisitos. Depois disso tudo esclarecido, o processo ainda facilita o desenvolvimento de ideias.

Colocar o usuário no centro, resolver o pain point, ter visão holística do problema e ser mais assertivo devido ao fato de ter equipe multidisciplinar, preparar o ambiente para liberar a criatividade

Engajamento dos times

Creio que os pontos positivos sejam: criar uma cultura de human centric, visando entregar a melhor solução. Abrir espaço para inovação e também criar um movimento de co-criação no time e empatia

Maior assertividade, melhoria guiada para o time

Possibilidade de compreender o problema certo e fazer as perguntas certas antes de ficar nas soluções. Isso dá uma compreensão melhor do que queremos de fato, dos nossos wicked problems, aumentando a assertividade das soluções.

A identificação correta do problema.

A participação de todos os envolvidos no projeto, tendo uma visão multidisciplinar.

A facilidade de compreender as dores do usuário

Entendimento da necessidade do cliente e aproximação com times de desenvolvimento.

eficácia

8) Na sua experiência de uso de Design Thinking no desenvolvimento de softw...

A inclusão da visão multidisciplinar ao processo, a percepção multistakeholders ao longo do processo e diminuição (ou previsibilidade) de impactos não previstos, visão mais sistêmica e humanizada.

Simplemente funciona. É facilmente acoplado à processos já estabelecidos, não é engessado, não demanda certificação, não demanda profissionais especializados (qualquer um que se interesse pode utilizar), no final sempre converge, há liberdade para se inovar nos processos de DT, existe muitas referências em artigos e livros, é uma linguagem comum no mercado e é de graça

Assertividade e melhor entendimento e entrega

Os pontos positivos são a estimulação a criação e inovação de produtos e na resolução de problemas.

Foco no cliente. Gerar ideias inovadoras Aprimorar soluções

Para mim alguns dos principais benefícios são: ter um entendimento mais claro sobre o que realmente deve ser resolvido, validar/ajustar antes de codar poupa muuuuto tempo.

Foco no usuário para gerar valor

Alcançar o usuário da ponta, o Power user. E não ficar preso e definir exclusividade do gestor do sistema.

Hoje consigo conceber nenhum desenvolvimento de produto ou serviço sem a utilização

Empolga em usar uma técnica divertida de pensar diferente

Melhor o entendimento do problema, e tu se torna mais próximo de quem irá usar o produto no final, fazendo assim a criação de um produto mais próximo do esperado.

Integração da equipe, geração de novas ideias e validação de hipóteses.

Aproximação da equipe técnica com o cliente, empatia entre as equipes envolvidas

Envolvimento maior das pessoas em comparação com outros métodos.

Não tenho experiência com DT

Hgg

Agilidade na tomada de decisões Facilidade em teste de hipóteses Consistência dos resultados Retorno à equipe Além de ser muito "gostoso" de se trabalhar

Rápida adesão do time por ser uma abordagem bastante disseminada no mercado, fácil utilização, praticidade.

Entender o problema

.

Construção de novas possibilidades através da empatia, com isso, trazendo mais input de ideias inovadoras para o desenvolvimento de um produto ou serviço

8) Na sua experiência de uso de Design Thinking no desenvolvimento de softw...

A adoção do design thinking promove a criação de softwares que se aproximam mais da realidade das necessidades do cliente, levando em consideração como ele realmente trabalhava antes da existência de um sistema, isso ajuda o mesmo a não ter dificuldades em seu fluxo de trabalho.

Co criação, integração com o cliente desde a definição do problema até a materialização dos problemas tendo como centro da discussão o ser humano(personas)

Desenvolvimento de empatia.

Assertividade no desenvolvimento

Falhar mais rapidamente (alta colaboração com usuário)

Empatia com usuário e seus problemas; Definição conjunta da dor a ser trabalhada;

Entendimento total do problema e da demanda e auxílio no processo de UX

Aproximação e profundidade com o assunto a ser abordado... integrar times para construção de soluções e validar hipóteses antes de seguir para construção ou desenvolvimento...

Entendimento das dores do cliente e ajuda o time a entender como entregar valor

Entregar oq realmente importa e evoluir a partir disto

Trazer para a equipe de desenvolvimento a real necessidade do usuário para determinado problema ou desafio apresentado. Geralmente há um desconhecimento da causa do problema, que geralmente traz por solução o desenvolvimento de algum problema digital, sem de fato explorar o problema real. Por isso, o Design Thinking soma na hora de apresentar os achados do campo para a equipe de desenvolvimento.

O uso de DT permite a criação colaborativa a partir de uma equipe multidisciplinar, o que enriquece o projeto de pontos de vistas e soluções inovadoras.

Agilidade, melhor tomada de decisões, criação de múltiplos caminhos adaptáveis que facilitam a solução de problemas e comunicação com o usuário, que traz frequentes insights sobre melhorias.

Uma solução que realmente faz sentido, e traz benefícios para o usuário.

N/a

Ajuda a equipe de desenvolvimento e cliente no aprofundamento da problemática, para esclarecer as dores que a solução precisa focar e para gerar ideias inovadoras. Tivemos uma experiência de reformular um produto existente e o Design Thinking contribuiu para tirar o vício da equipe com a solução existe e nos permitiu criar uma solução totalmente inovadora, "fora da caixa" mesmo.

Creio que o benefícios mais evidente é a validação das reais necessidades do cliente. Assim, não precisa "empurrar" o produto para o cliente depois.

Melhor entendimento das necessidades principais do usuário final e portanto melhores requisitos

.

Exploração de novas oportunidades ainda não percebidas, oriundas de necessidades dos clientes

Engajamento, diversidade ponto de vista e abrangência de respostas

8) Na sua experiência de uso de Design Thinking no desenvolvimento de softw...

O design thinking é uma abordagem de natureza qualitativa, ou seja, busca entender com profundidade os usuários e trabalhar esquematicamente com dados de fonte quali. Assim, é muito útil para equipes que normalmente não conhecem seus usuários e trabalham em cima de vieses próprios.

Agilidade no encontro de soluções e buy in dos diferentes stakeholders

A segurança e o empoderamento que as ferramentas entregam para as equipes envolvidas é o ponto central. Quando as equipes conseguem elas mesmas se colocar no lugar do usuário e resolver as incertezas do processo de inovação por conta própria. O impacto a nível humano que o Design Thinking causa nos designers pode ajudar a criarmos um futuro melhor.

Ter o cliente/usuario participando

Descobrimto de novas soluções criativas para os principais problemas encontrados nas rotinas dos usuários.

Empatia e sensibilizar o olhar para o problema

A aproximação das equipes de desenvolvimento ao cliente e às áreas de negócio, aliada a uma forte mudança cultural tem sido um dos maiores benefícios

Como os demais modelos empíricos, buscar a inovação por tentativa e coleta de feedback.

Incentivo de criatividade

Aumenta a aderência das entregas a real necessidade do cliente.

A busca por horizontalização dos conhecimentos e das mentalidades

Resolução ou validação do real problema a ser resolvido.

Mais do que entregar projetos, o DT entrega alto valor agregado direto ao usuário final

Exploração de ideias, co-criação, melhor compreensão do público alvo e engajamento dos envolvidos.

Melhor user experience e aceleração no tempo de entrega

A empatia gerada com o cliente durante o desenvolvimento da solução.

profundo entendimento sobre o contexto, prototipagem rápida e feedback ágeis.

Maior empatia com o usuário.

Aproximação da equipe com o produto, eles entende o importância, necessidades que ele vai suprir e o impacto que o produto pode gerar no negocio e nas pessoas!

A criação de um mindset no time, que facilmente soluciona problemas e/ou propõe soluções sem necessidade da participação de gestores.

Fácil iteração e a forma em melhorar o MVP

o grande benefício do DT está em abrir espaço para questionar o briefing para entender qual é o real problema, pela perspectiva das pessoas envolvidas do processo.

8) Na sua experiência de uso de Design Thinking no desenvolvimento de softw...

São um conjunto de práticas e métodos fácil de aplicar em dinâmicas rápidas. Participantes inexperientes conseguem utilizar e trazer muitos insights. Outra vantagem, é ter um processo especificado com etapas que se complementa ajudando a manter o foco no processo.

Compreender melhor o modelo de negócio do cliente, seus desafios, os problemas que ele quer resolver, identificação se os ativos da companhia sanam essas necessidades e esses problemas, conhecer o dia a dia do cliente para entender como prover uma solução, identificar os atores envolvidos e a atuação deles dentro da solução.

Múltiplos pontos de vista. Foco na necessidade do cliente. Foco no problema e não na solução.

Hipóteses de negócio melhor validadas

Várias pessoas com pontos de vista e experiências variadas criando uma solução inovadora e viável.

falar com o usuário final

Melhora o entendimento sobre o problema x solução.

A meu ver os processos de UX e Ui design se baseiam em design thinking e user centered design e os benefícios disso é compreender para quem se está projetando e para qual contexto de uso e isso direciona o foco de atuação e de solução do problema para designers e desenvolvedores.

Estimula a criatividade, melhora relacionamento e engajamento entre stakeholders do projeto, criação de espaços onde vc pode "errar" e está tudo bem pois precisa do erro para aprender e melhorar (conceito do fail-fast). Maior assertividade nas entregas quando se ouve empática e ativamente o usuário final e com a validação de idéias/soluções através dos protótipos.

minimiza riscos; facilita o entendimento e priorização

Colaboração e visão sistêmica

Processo formal e funcional de descoberta de valor, ajudando a aumentar a eficácia do processo de desenvolvimento de software.

democratiza a solução, envolve o time tecnico o que reduz incertezas e frustrações

Manter o usuário no centro do processo sem deixar de lado as necessidades do negócio.

Integrar todos os participantes no assunto abordado, fazer com que todos se sintam parte dos projetos. Ter ideias mais disruptivas.

Aproxima todos os envolvidos no desenvolvimento de um produto do usuário, humaniza o processo e evita desperdício de tempo e dinheiro com um processo de descoberta e melhorias estruturado e centrado nas necessidades das pessoas - seja usuário ou stakeholders

S9 - 9) E quais seriam as dificuldades ou os pontos negativos?

9) E quais seriam as dificuldades ou os pontos negativos?

Para mim a maior dificuldade é fazer as pessoas aprenderem a se conectar com a necessidade real do outro, a se desapegar das suas ideias e entender que o processo deve existir para diminuir o esforço do usuário e não criar.

Imposição de prazos antes do início do projeto e clientes de empresas de mindset fixo, funcionais e hierárquicas que não possuem a informação necessária mas não envolvem os funcionários que as tem, gerando vários obstáculos.

As dificuldades quando o assunto é DT, assim como com abordagens de desenvolvimento de software ágeis, são características culturais. Algumas oportunidades permitem ir adotando gradativamente os processos e capacitando as pessoas, mudando a empresa. Em outros casos, quando o tempo é escasso, é necessário workshops para identificar o que efetivo serão as equipes neste processo.

Muito aberto, depende muito da experiência, totalmente sem roteiro

Pode ser que a empresa que esteja desenvolvendo o software não entenda a necessidade do processo.

Cliente enxergar valor; Agenda com a maior participação de membros;

Adoção por parte de corporações que nunca trabalham com DT.

Muitas vezes, se não feito corretamente, os resultados ficam muito alto nível ou intangíveis.

A equipe entender que o processo leva mais tempo do que o esperado.

Engajamento do time. Principalmente dos que não conhecem DT

Interação de todos os envolvidos.

Quando temos grande quantidade de pessoas envolvidas, a quantidade de informação pode ser ingerenciável, assim como a possibilidade de alcançar um consenso.

Em projetos de concepção, existe uma dificuldade em tangibilizar as ideias geradas para criação de um backlog acionável para a equipe de desenvolvimento, o que muitas vezes gera retrabalho.

Consenso entre áreas.

A obtenção de informações precisas por vezes é um problema na aplicação do DT.

Comprometimento da equipe, acesso a dados reais, filtragem de insights que dão match com o cliente

Exceção de frameworks a serem utilizados.

Grande demanda de tempo e pesquisa prévia, normalmente as empresas não estão dispostas a investir este tempo e tão pouco enxergam valor nisso. O DT está mais para o tema "da moda" que um método efetivo na maioria das empresas

mmn

9) E quais seriam as dificuldades ou os pontos negativos?

O entendimento dos participantes, em alguns casos vira um muro de lamentações. Ok, tem problemas e quais seriam as soluções? mas muitos ficam girando em cima do problema atual sem enxergar algo fora caixa, por exemplo

contato com os usuários

quando os participantes nao sao colaborativos ou nao estao engajados na dinamica o resultado é ruim

Tempo necessário para uma sessão profunda. É muito difícil que aceitem na disponibilidade recomendada, que no meu caso seria de 1 dia a até 4 dias.

Não é um processo simples e linear, portanto para ser bem executado e produtivo necessita de pensamento crítico, conhecimento de negócios e os mindset corretos, como por exemplo mindset de beggíner, característica difícil de encontrar em um profissional, principalmente aqueles com cargos de alta senioridade.

Trazer pessoas extremamente práticas (desenvolvedores) para um contexto mais lúdico

É comum não se ter o tempo adequado, dificuldade de diversidade de participantes incluindo o ponto de vista do usuário.

A falta de credibilidade no processo por parte de algumas pessoas envolvidas

Não sei

Engajamento por parte do cliente na etapa de discovery e pesquisa

Planejamento, agenda dos participantes, engajamento

A dificuldade que vejo não é no processo, mas na forma como alguns grupos/pessoas se utilizam do Design Thinking, querendo ter muita previsibilidade ou certeza quanto a dores, hipóteses e seus resultados. O Design é um processo cíclico de Build -> Test -> Learn

Em um mundo ágil onde ainda existem prazos (deadlines), é difícil achar tempo para fazer o processo de DT com calma e realmente voltar etapas quando necessário. Infelizmente ainda não conseguimos aplicar o DT da forma correta a ponto de ter todos os seus benefícios.

A facilitacao pode ser uma dificuldade, as pessoas previamente entenderem a importancia de trabalhos assim, influencia ou indução de alguns sobre outros, foco na solucao e nao no problema.

Pouca gente capacitada que possui um bom skill de facilitação para contratar

Demora muito para executar o processo (normalmente mais de 6h) e pouca pessoas conhecem o método fora da minha empresa

Definição da equipe do cliente que participará da ideação.

Falta de tempo.

Dificuldade no convencimento de que o tempo utilizado irá gerar valor

Aceitação e conhecimento dos times sobre as técnicas

9) E quais seriam as dificuldades ou os pontos negativos?

A minha maior dificuldade hoje é conseguir agenda do cliente para realizar as cerimônias. Colocar várias pessoas de áreas diferentes em uma sala no mesmo período de tempo (geralmente longo) para realizar as atividades propostas pelas técnicas é um grande problema no mundo corporativo. As pessoas são muito ocupadas e não tem tempo disponível. Geralmente pedem para quebrar as atividades do processo escolhido em várias reuniões de tempo menor e isso acaba prejudicando e atrasando bastante a equipe de desenvolvimento. Assim sendo, na maioria das vezes, opto por não realizar todas as atividades e fico apenas com as mais básicas.

Se a moderação/ facilitação das práticas dentro do ciclo do design thinking é fraca, pode-se ter um produto sem pé nem cabeça no final ou expectativas diferentes vindo de pontos diferentes do grupo de trabalho, o que acaba resultando em frustração ou do time de dev ou dos clientes mais pra frente.

A técnica é muito interessante, mas se não houver um facilitador experiente, os resultados não são satisfatórios -> Muita ideia e pouca ação.

Com exceção dos momentos em que utilizamos técnicas mais informais, com pouco apego aos frameworks, o processos de Design Thinking quando não bem facilitados são massantes, inconclusivos e nada efetivos. Considerando que essas práticas são ferramentas, ou seja, meios para se alcançar algo, é como dar um martelo pra quem não sabe usar, ou seja, acaba que não serve pra nada.

Engajar todas as pessoas da empresa, trazer o decisor para a tomada de decisão

Na minha corporação, não consigo utilizar ao longo de um projeto, sempre pensamos como momento inicial do projeto, por isso adotamos workshops como abordagem. Talvez em projetos muito longos eu teria dificuldade em manter o DT rolando (mas já vi funcionar também). Outra dificuldade é que o time que for participar deve ter ideia do valor do DT, pra isso eu faço uma apresentação de cases de sucesso no início e também foco muito no mindset correto para o workshop. Sem dúvida o facilitador apesar de estar apenas lá pra "facilitar" uma vez que o time não chegue a um entregável de valor no final, isso queima o filme da metodologia e do facilitador, portanto é um trabalho "tenso" de certa forma, onde você não põe a mão diretamente na solução mas ao mesmo tempo tem que gerar resultados.

Know how de como facilitar as dinâmicas para que todos entendam o seu objetivo, forma de fazer, e a própria utilidade no projeto.

Conseguir com que os envolvidos tenham agenda e se comprometam a estar 100% presentes durante todo o processo

mediar os diferentes atores, mostrar a todos que o problema que está se tentando resolver pode não ser o verdadeiro problema a ser resolvido

Ocasionalmente o não compartilhamento desta visão.

Já apontado acima.

nenhum

Conduzir o design thinking. Conseguir tempo e atenção necessários com todos os envolvidos.

tempo para aplicação e retroalimentação dos modelos/métodos. depende de uma mudança de cultura.

A dificuldade de usar DT é a mesma que qualquer abordagem ou mudança dentro de uma empresa ou área... é o conhecimento e a utilidade. Eu sempre sugiro fazer um workshop prático com a equipe e os envolvidos para conseguir alinhar conhecimento e assim progredir de forma positiva. Se as pessoas não veem valor na mudança, se elas não são engajadas de forma positiva... a resistência é bem maior

Conexão entre os participantes

Nem todos conhecem a metodologia, tornando o processo moroso.

Em empresas menores ou com um mindset conservador é difícil de vender para o negócio o tempo necessário para que as sessões de dt ocorram. Além disso, o tempo é localização das diferentes equipes pode ser um fator a ser considerado em sessões de dt. (tempo necessário X esforço X resultado)

.

9) E quais seriam as dificuldades ou os pontos negativos?

Quando deturpam o processo, utilizando o design thinking e o agile apenas pra justificar decisões que já estão pré estabelecidas ou já possuem soluções.

Alguns pontos da prototipagem.

Não vejo pontos negativos, pois até errar, mas errar rápido é permitido.

Competição pelo tempo dos colegas com outras tarefas e falta de engajamento de alguns durante a participação.

ter o grupo multidisciplinar amplo junto, pensar que para tudo se deve usar DT

Engajamento da equipe para trabalhar com o design thinking e entender o problema do usuário pelo método de observação ou entrevista

A dificuldade esta na quebra de silos e engajamento de todos envolvidos no projeto e/ou solução.

Falta de conhecimento e pratica em como Manter equipe focada nas pessoas envolvidas e em como resolver para elas e não nos processos ou softwares que deverão ser criados

Envolvimento das pessoas

Em minha opinião o processo precisa ser bem realizado. Caso alguma parte (principalmente a da pesquisa) seja feita as pressas, o desenvolvimento final pode ser falho.

alguns artefatos complexos..nao entendemos bem para qual cenário utilizar..Demora para fazer as sessões, medo de não conseguir identificar e resolver o problema correto.

Maturidade dos times

Acho que o processo é um pouco demorado para quem não está engajado. É difícil também, isolar os participantes durante os 2-3 dias do DT e fazê-los ficarem focados, sempre estão trabalhando durante o DT...

nem todos estão sempre dispostos a colaborar, mas isso vem mudando

Dificuldade em quebrar a inércia inicial e trazer engajamento para a equipe. Associar estruturas libertadoras pode ser muito bom para trabalhar nas reuniões e workshops

Acho que principalmente nos momentos de divergência, é comum que as pessoas do time se sintam perdidas e isso acabe as desmotivando.

Muitas pessoas ainda preferem os métodos tradicionais por acreditarem ser mais objetivo.

conseguir unir uma equipe multidisciplinar

Muitas melhorias sugeridas não são possíveis de implementar , o que pode causar frustração no cliente em não ver o valor agregado.

desconhecimento da técnica por outros integrantes do time

Lidar com o emergente e o complexo.

Do ponto de vista de gestão de produto e projetos não há pontos negativos.

9) E quais seriam as dificuldades ou os pontos negativos?

-

Reunir todos os envolvidos no projeto

Mobilizar equipe

Reunir as pessoas e mantê-las focadas por um tempo maior.

Engajamento de todos participantes

Resistência da equipe de desenvolvimento

Mudança de cultural da empresa, mostrar necessidade, mobilizar equipe

Falta de domínio sobre o assunto e resistência de pessoas céticas

Não encontrei nenhuma atenuante; o momento

Primeiras facilidades e varia a dificuldade de acordo com entrosamento com a equipe

Nenhum

A maior dificuldade é conseguir agenda das pessoas.

Primeiramente conhecer mais sobre UX e DT

Hhg

Receio dos participantes e resistência à métodos inovadores. No final todo mundo gosta dos resultados, durante o processo pode haver algum espaço de dúvida, o que obriga os condutores a serem ainda mais seguros.

Dependendo de como for utilizada, pode acabar sendo um guia passo a passo

Adaptação das pessoas em utilizar métodos; Desapego de soluções (colaboradores já vem com a solução e não com entendimento real do problema)

.

Conhecimento dos participantes para utilização correta das ferramentas

A maior dificuldade é a implementação da cultura e do jeito de se criar um software com design thinking entre toda equipe de desenvolvimento, bem como o convencimento do cliente que é necessário um certo convívio para entender suas necessidades.

Tempo dos nossos clientes para participarem de toda a jornada.

conhecer o Impacto do design thinking.

manter o foco

9) E quais seriam as dificuldades ou os pontos negativos?

Alto nível de gerenciamento tem dificuldade para enxergar valor; Nerds não querem participar dos projetos, preferem ficar sentados cumprindo tarefas.

Talvez a única dificuldade é fazer os desenvolvedores pensarem de forma incremental, construindo e evoluindo aos poucos, pois ouviram um caminhão de problemas dos usuários.

O tempo do projeto precisa ser um pouco maior para que se saia com todas as validações. O que apesar de ser um ponto negativo, não impede de agregar ainda mais valor no processo

Devido a velocidade demandada área e estratégicas levam o DT como uma parte que pode ser retirada do projeto... ter efetivamente a participação de multidisciplinar e pessoas chaves nas construções e testes

Resistência

entendimento dos participantes. Cultura da empresa e profissionais. Vender a ideia não é fácil dependendo do ambiente

Praticamente em todos os projetos é necessária a evangelização do DT, as pessoas confundem muito o que DT de fato se propõe a fazer, e o que de fato é. Há muita glamurização por traz dos workshops, e algumas empresas têm dificuldade de autorizar o contato com o seu público, o que dificulta bastante o entendimento das dores e problemas reais. Alguns pensam que o DT é uma modinha, ou algo não aplicável no mundo real, fora dos escritórios da IDEO.

A dificuldade muitas vezes está no alinhamento das equipes em diferentes projetos e a condição produtiva entre as etapas de divergência e convergência.

Adaptação relativamente complexa da equipe quanto às técnicas e ao fluxo extremamente ágil proposto pelo DT.

1- Sair do papel e executar de fato o que for definido no DT. 2- Tempo para que seja executada a sessão. Muitas vezes os stakeholders não querem dar tempo para que seja executado o processo de DT.

N/a

Em projetos de evolução e com escopo muito definido (como migração/modernização), o Design Thinking acaba não sendo muito aplicável... Projetos com arquitetura muito definida (com tecnologias ou ferramentas de terceiros) também dificultam a utilização da abordagem, porque nessas situações há pouco espaço para inovação.

Como não existe um roteiro, corremos o risco de não ter escolhido a melhor técnica ou ferramenta, sendo necessários ajustes e iterações. Isso pode significar mais tempo de projeto.

Resistência inicial da alta direção: acreditam que é só uma "modinha" que consome tempo e recursos

.

Resistência do time em fazer TBC

Normalmente as empresas ainda enxergam o DT como atividade lúdica

Normalmente, as equipes relutam em fazer campo, que é a parte mais importante e mais custosa do processo. É difícil convencer os atores de que o conhecimento deles sobre o usuário é limitado.

Manter o hábito de utilizar a técnica com recorrência

Transformar dados qualitativos em dados válidos para o ambiente corporativo

9) E quais seriam as dificuldades ou os pontos negativos?

Muitas técnicas e ideias podem surgir. Tomar decisão é essencial

Nem sempre a solução encontrada é viável.

Há quem não tenha paciência e consequentemente não acredite na efetividade do método

Encontrar participantes com o perfil adequado (do negócio aos desenvolvedores), questões contratuais com desenvolvimento terceirizado, manter a equipe dedicada durante o projeto (a maioria desenvolve dezenas de projetos em paralelo), bem como controlar a expectativa da alta gestão quanto aos projetos de inovação,

Não é do Design Thinking mas sim da cultura das empresas. A Técnica diz como fazer mas sempre a empresa quer modificar pra fazer na sua realidade e depois diz que não funciona

por ser sistemático pode causar desgaste emocional a profissionais "detalhistas"

Entender o problema e causa dele.

Silos hierárquicos que os participam trazem de suas empresas

Falta de conhecimento de ferramentas ou técnicas para extrair mais vantagens da técnica.

Adoção do modelo no mundo corporativo versus modelos tradicionais. O mundo corporativo ainda é um pouco quadrado com metodologias centradas na pessoa, e muito mais orientado a entregas com prazo e escopos bem definidos. Além disso, é um território pouco conhecido e às vezes mal interpretado por pessoas mais técnicas, que julgam ser algo desnecessário.

Convergir as divergências.

Mentalidade do time no início

A própria empatia pode se tornar uma dificuldade, no sentido de entender e compreender o lado do cliente, para encontrar a solução que o mesmo necessita.

Facilitação de grupos ainda com pouca maturidade da abordagem e do conhecimento técnico do processo de DT

Adequar o projeto ao tempo e escopo

A maior dificuldade é adaptação da abordagem, pensar como adaptar processo para cada projeto e ter um processo definido que seja assertivo é minha maior dificuldade na aplicação da abordagem. Pois nem todo projeto dá para utilizar a abordagem completa.

Quando o cliente questiona a importância do método, a julgar pelo tempo que se leva para descobrir e validar uma solução. Muitas vezes fica difícil provar para o cliente uma solução e que foi descoberta com o uso dos métodos de Design Thinking.

Cocriação entre algumas equipes.

A dificuldade está em separar tempo para este tipo de prática no dia a dia corrido das empresas.

Algumas etapas demandam tempo, quorum, agendamento; e isso nos dias de hoje costuma dificultar a execução das tarefas.

Nem sempre o cliente tem tempo ou disposição para conhecer a fundo o problema, ele sempre acha que sabe o que precisa e a forma como ele pensou é a melhor solução.

9) E quais seriam as dificuldades ou os pontos negativos?

Dificuldade do grupo em se distanciar da busca imediata pela solução e nas limitações técnicas e financeiras.

Os participantes as vezes não acham que tenha valor

Falta de experiência ao conduzir o design. Pessoas apegadas as ideias

buy-in da liderança

A cultura tradicional por vezes inibe a potência do método e a falta de profissionais experientes para a facilitação do método.

Trabalhar com pessoas de áreas de conhecimentos mais exatas que não compreendem os valores de cada dinâmica e cada etapa do processo.

a dificuldade é promover a mudança de mindset, pois o design thinking é muito mais que um conjunto de ferramentas, ele é uma abordagem, uma forma de se trabalhar que requer grande mudança de mindset principalmente por parte de gestores/líderes (também encontro bastante resistência com os colegas programadores).

tempo disponível dos participantes

Participação e investimento

Engajamento dos clientes para presença nos workshop's.

A grande dificuldade está em implantar o trabalho colaborativo principalmente em ambientes com cultura de performance e que buscam a meritocracia

Evangelizar os processos e internaliza-los na equipe e na empresa como um todo.

A empresa comprar a ideia do uso de técnicas e pesquisa. E num cenário de pouca cultura de UX, fazer com que os desdobramentos aconteçam, às vezes você faz todo um projeto e ele morre pq ninguém levou adiante.

O processo de compilação e introspecção para gerar os protótipos levam um tempo, que normalmente não é respeitado, assim como etapa de testes nunca é respeitada pelos integrantes e stakeholders

S10 - 10) Qual a sua experiência, em anos, nos seguintes critérios?

#	Campo	Menos de 1 ano de experiência	De 1 a 3 anos	De 4 a 7 anos	Acima de 8 anos	Total
1	No uso de Design Thinking no desenvolvimento de software	19,62% 31	48,10% 76	26,58% 42	5,70% 9	158
2	Na área de TI de forma geral	12,03% 19	18,35% 29	20,25% 32	49,37% 78	158

Exibindo linhas 1 - 2 de 2

S11 - 11) Qual o seu papel ou função organizacional atual?

#	Campo	Contagem da escolha
1	Facilitador	10,13% 16
2	Pesquisador	0,63% 1
3	Design de interação	0,00% 0
4	UX/UI Designer	13,29% 21
5	Desenvolvedor	3,80% 6
6	Especialista	6,96% 11
7	Analista	3,80% 6
8	Agile Coach	17,09% 27
9	Consultor	8,23% 13
10	Product Owner	8,23% 13
11	Engenheiro	3,16% 5
12	Outro?	24,68% 39

158

Exibindo linhas 1 - 13 de 13

Outro?

Head de Transformação Digital

Gerente de Design e Inovação

Service Designer

Product manager

Gerente

Coordenador

CEO

Coordenador de times de desenvolvimento

UX Designer, Pesquisador e Facilitador