

# Problemas do Design Thinking para a Engenharia de Requisitos: uma Revisão Sistemática da Literatura

Fábio Avigo de Castro Pinto<sup>1</sup> e Fábio Levy Siqueira<sup>2</sup>

<sup>1</sup> Escola Politécnica da Universidade de São Paulo, Brasil. fabio.avigo@usp.br

<sup>2</sup> Escola Politécnica da Universidade de São Paulo, Brasil. levy.siqueira@usp.br

**Resumo.** Tido como uma estratégia de inovação no campo do Design e trazido como ferramenta à Engenharia de Software, o *Design Thinking* (DT) é uma abordagem promissora em crescente utilização nas etapas da Engenharia de Requisitos (ER), por focar no cliente final do software através de uma técnica iterativa estruturada em fases definidas. Este processo, entretanto, pode trazer um viés ao desenvolvimento de software por desconsiderar outros *stakeholders* que usualmente participam em processos tradicionais e direcionar todos os esforços de requisitos para o usuário. De forma a identificar a existência de problemas trazidos à ER com a aplicação do DT e de desafios ainda existentes na área não endereçados diretamente pela abordagem, este artigo apresenta uma revisão sistemática da literatura de relatos de uso ou de estudos de caso de aplicação do DT a projetos de software na etapa de requisitos. Além dos problemas de ER levantados pelos autores em estudos primários, a revisão também identificou as sugestões de aprimoramento do método. Os resultados apontam para a existência de problemas na área que ainda não são resolvidos pela aplicação do processo de DT.

**Palavras-chave:** *Design Thinking*, Engenharia de Requisitos, Revisão Sistemática da Literatura.

## 1 Introdução

Atualmente, o mercado é influenciado por uma série de fatores complexos – tecnologias emergentes, processos de globalização e alteração constante de requisitos –, o que faz com que o ciclo de vida de um serviço se torne mais curto. Esta situação conduz a um conflito entre as ferramentas e métodos existentes e utilizados no passado, os quais não se adequarão aos novos requisitos de desenvolvimento de produtos e serviços. Para sustentá-los, as companhias deveriam buscar outras possibilidades aplicando novas abordagens para criação de valor [19].

Neste contexto, uma nova ferramenta é o *Design Thinking* (DT), desenvolvido em Stanford, refinado pela IDEO e incentivado por Hasso Plattner, um dos fundadores do SAP [20]. Essa abordagem define um processo de times para o *design* da solução de um problema [21]. Nele, é comum que os participantes desenvolvam empatia pelo usuário final, redefinam o problema, combinem diferentes perspectivas, realizem *brainstorming* de soluções e desenvolvam protótipos. O DT experimentou um ganho

em popularização na economia na última década, dado o seu *framework* prático e que leva a benefícios claros para as companhias [20].

No entanto, apesar de a técnica representar uma potencial solução para algumas questões tradicionais da ER, por elicitar necessidades do usuário, em vez de requisitos [17], sendo considerada, inclusive, como uma “forma moderna de Engenharia de Requisitos” [26], até mesmo entusiastas da abordagem, como Uebnickel e Hehn, apresentam que há desafios ainda a serem superados, como capacidades limitadas de sucesso devido à dependência da experiência individual dos participantes ou ainda problemas de qualidade de requisitos em termos de rastreabilidade, adequação, completude e consistência [27].

Com isso, este trabalho investiga os problemas do *Design Thinking* relatados pela comunidade científica. Essa análise contribui com a área ao indicar pontos para pesquisas futuras que busquem melhorar a integração dessa técnica com as atividades de Engenharia de Requisitos. Inicialmente, realizou-se pesquisa para identificação quanto à existência de Revisão Sistemática de Literatura (RSL) que já contemplasse a temática, e, embora tenham sido encontrados artigos correlatos que são detalhados na seção 2.2, verificou-se a inexistência nos principais portais acadêmicos de uma RSL analisando os problemas e sugestões de melhoria do *Design Thinking*.

A revisão apresentada neste artigo está estruturada da seguinte forma: A Seção 2 apresenta uma breve contextualização a respeito da técnica e a sua relação com a área de Engenharia de Requisitos. A Seção 3 descreve a metodologia da revisão utilizando-se da publicação de Kitchenham et al. [24] como referência. A Seção 4 apresenta uma sumarização e discussão dos resultados, para cada pergunta de pesquisa e, por fim, a Seção 5 conclui o trabalho e introduz possíveis discussões a serem realizadas em trabalhos futuros.

## 2 Contexto – Design Thinking

Tim Brown, CEO da IDEO, uma empresa de consultoria localizada em Palo Alto, Califórnia, define o DT como “uma disciplina que utiliza a sensibilidade e os métodos do *designer* para corresponder as necessidades das pessoas com o que é tecnologicamente factível e o que uma estratégia de negócios viável pode converter em valor para o cliente e oportunidade de mercado” [22]. Os irmãos Kelley, fundadores da IDEO, definem o DT como “uma maneira de encontrar necessidades humanas e criar novas soluções utilizando as ferramentas e *mindsets* de praticantes de *design*” [23].

O processo de DT corresponde a uma abordagem estruturada e inovativa para o desenvolvimento de produtos, serviços e modelos de negócios, fornecendo um processo estruturado para a elicitação de requisitos. Embora não exista um consenso formal a respeito das etapas que compõem o método [16], o processo é reconhecido pela utilização de um “Diamante Duplo”, com quatro passos definidos de divergência, convergência, divergência e convergência. A primeira fase (divergente) nasce de um processo de definição do problema, divergindo para buscar os motivos por trás da necessidade. A segunda fase apresenta a convergência desse processo de brainstorming para uma síntese inicial. Daí, a terceira fase (divergente) nasce dessa síntese e parte

para a ideação de soluções que atendam a resolução do problema. Por fim, a quarta e última fase (convergente) se refere à prototipação dos conceitos gerados na fase anterior e à validação.

## 2.1 O Design Thinking e a Engenharia de Requisitos

Vetterli, Brenner e Uebernickel [17] justificaram os motivos pelos quais a área de Engenharia de Requisitos (ER) pode se utilizar do DT para elicitación das necessidades de um cliente, por produzir uma série de protótipos rápidos e simples que convergem em soluções inovadoras. Segundo os autores, dado que a medida primária do sucesso de um sistema de informação corresponde ao grau de atendimento ao propósito original (ou seja, às metas dos *stakeholders*), a ER é um processo de inicial descoberta e definição do propósito, o que a torna inerentemente difícil. Além disso, dado que os analistas de requisitos muitas vezes começam com ideias mal definidas e frequentemente conflitantes, os processos da ER devem ser mais iterativos e envolver um número maior de *stakeholders* que outras atividades da Engenharia de Software [18]. Isto posto, os protótipos resultantes das atividades de DT auxiliam na concretização de diferentes ideias sem a simplificação do ambiente, enquanto focam em necessidades específicas e importantes dentro do espaço de *design*. Embora protótipos também sejam utilizados em metodologias ágeis e na própria área de ER, a grande vantagem do DT se dá na identificação e eliminação de inconsistências técnicas o quanto breve possível no processo [17].

Mais recentemente, Hehn et al. [15] também afirmam que a área de ER enfrenta o desafio de descobrir e satisfazer as necessidades confusas e requisitos voláteis dos vários *stakeholders* envolvidos no processo. Com isso, os autores apresentam que o DT corresponde a uma abordagem promissora de endereçamento deste desafio. O trabalho apresenta três possíveis estratégias para a integração do DT à ER: o *Design Thinking* adiantado, o *Design Thinking* infundido e o *Design Thinking* contínuo.

**Design Thinking Adiantado (*Upfront*).** Para esta situação, o DT é utilizado como um pré-projeto para identificar *features* relevantes a serem implementadas. O resultado é utilizado como base para realizar as atividades de Engenharia de Requisitos. Essa forma de DT é melhor aplicada quando há um nível de incerteza no começo de um projeto a respeito das necessidades do cliente e à solução correspondente [15]. O potencial de entregas é alavancado com um *framework* que serve de referencial para continuamente incentivar a criatividade. O núcleo da solução final possui conexões rastreáveis com as necessidades do cliente, auxiliando a priorizar *features* relevantes. O protótipo funciona como um artefato que apoia a comunicação entre os diferentes grupos de *stakeholders*. Entretanto, a solução requer uma quantidade significativa de esforços e recursos para conduzir um pré-projeto. Há um potencial de perda de conhecimento implícito e de resultados gerados no processo. Além disso, pouca atenção é dada para artefatos cruciais ao desenvolvimento posterior, como requisitos de qualidade, restrições ou modelos de dados.

**Design Thinking Infundido.** Aqui, o DT é utilizado como um conjunto de ferramentas para mesclar um processo de ER existente com artefatos e métodos selecionados. É particularmente adequado quando práticas comumente utilizadas da ER precisam de intervenções específicas, como a necessidade de novas ideias [15]. O caráter intervencionista do DT requer mudanças pequenas às práticas de ER, com baixo gasto adicional de recursos. Essa estratégia possui, entretanto, criatividade limitada se comparada à estratégia anterior. Ela não possui o impacto sustentável dos métodos de DT devido ao caráter de intervenção e também não se atenta aos detalhes subsequentes de artefatos críticos ao desenvolvimento, como no DT adiantado.

**Design Thinking Contínuo.** As atividades e princípios de DT são continuamente integradas às práticas de ER, usualmente em um papel específico do projeto. É recomendável para abordar configurações complexas de problemas, em que consumidores precisam ser continuamente envolvidos dentro do ciclo de vida de desenvolvimento de um produto. Dada a interação contínua, neste caso, leva-se em consideração o desenvolvimento de artefatos críticos ao projeto, e há a integração de um *mindset* centrado no cliente através de todo o projeto, com requisitos mais precisos através da contínua identificação e prototipação. No entanto, a abordagem possui uso mais intensivo de recursos e de suporte organizacional, bem como depende mais do time para a sua condução.

Com a popularização da técnica para o desenvolvimento de software, não é difícil encontrar relatos de uso ou estudos de caso com a temática. Hehn e Uebersnickel [11] colocam que, quando um problema é “traçoeiro” (*wicked*), o DT ganhou reconhecimento como uma abordagem estruturada para resolução do problema que se constrói através de trabalho em equipe interdisciplinar, a exploração das necessidades humanas, a rápida prototipação e ciclos iterativos de aprendizagem nos estágios iniciais dos processos de desenvolvimento do produto, serviço ou sistema. Esta técnica é significativamente diferente das práticas tradicionais de ER em projetos de software.

## 2.2 Revisões de Literatura correlatas

Foram identificadas as seguintes revisões de literatura que são correlatas ao projeto:

- *Design Thinking Process Model Review* [16]: o artigo apresenta uma revisão sistemática realizada sobre 35 modelos processuais no campo do DT para aplicação em um ambiente prático, com o objetivo de comparar os modelos de acordo com o número de etapas processuais e específica terminologia. A revisão é reconhecidamente não exaustiva. Concluem que não há um modelo padronizado de processo com fases definidas e, assim, cada autor tem de explicar exatamente o que está incluído em cada fase individual e como elas se relacionam. Os autores concluem que, dez anos passados desde a pesquisa básica sobre DT, a abordagem tem sido subsequentemente desenvolvida através do aprofundamento da comunidade científica dos princípios subjacentes adquiridos de projetos de pesquisa (empíricos). Há diversos modelos distintos atualmente, e a literatura aponta que

alguns pesquisadores trabalharam com modelos processuais diferentes e obtieram resultados similares. Além disso, há uma tendência mais recente para modelos com mais fases.

- *Design Thinking Integrated in Agile Software Development: A Systematic Literature Review* [28]: o artigo apresenta uma RSL a respeito da integração entre o DT e as metodologias ágeis para desenvolvimento de software, sob uma base de 29 artigos. Dentre as conclusões mais relevantes, apresenta que (1) a técnica mais comumente utilizada nesta integração é o *Scrum*; e (2) em alguns casos, a qualidade do software foi significativamente melhorada com reconhecida satisfação de clientes.
- *A Systematic Literature Review for Human-Computer Interaction and Design Thinking Process Integration* [29]: O artigo revisou 72 papéis publicados entre 1972 e 2017 para responder a questão de como a interação humano-computador (IHC) e os processos de DT se sobrepõem, diferenciam e podem melhorar um com o outro. Dentre as conclusões, ambas abordagens compartilham de passos similares e iterativos: compreensão do usuário para identificação de problemas, ideação, prototipação e testes. No entanto, diferem-se nos princípios específicos de cada etapa, determinando as ferramentas a utilizar e objetivos a alcançar. O HCI requer a compreensão dos usuários para elaborar requisitos, aplicando regras e princípios para o *design*, focando sobremaneira na construção do software. O DT, por sua vez, foca na construção de empatia para entender os usuários, projetando atividades que gerem ideias e soluções, encorajando que estas sejam traduzidas na vida do usuário.

### 3 Metodologia da Revisão

#### 3.1 Referência

A Revisão Sistemática de Literatura utilizou como referência o artigo de Kitchenham et al. [24].

#### 3.2 Objetivos e Questões de Pesquisa

Foram elaboradas as seguintes questões de pesquisa para atender aos objetivos da revisão:

*RQ1. Quais problemas de Engenharia de Requisitos foram identificados em projetos de desenvolvimento de software que usaram a abordagem do Design Thinking?*

*RQ2. Quais são as sugestões dos autores para melhorias do Design Thinking?*

#### 3.3 Protocolo de Revisão

A busca por artigos aderentes ao objetivo de pesquisa foi realizada nos quatro principais portais acadêmicos: Springer, ACM, IEEE e Science Direct. Optou-se pela utilização destes por serem os mais comumente utilizados para a consulta de artigos científicos.

O intuito da RSL foi de buscar por estudos de caso ou relatos de experiência do uso do *Design Thinking* em projetos de desenvolvimento de software que tenham apresentado relação direta com a área de Engenharia de Requisitos. Dessa forma, definiu-se a string de busca como: "design thinking" AND ("case study" OR "experience report") AND ("software" OR "development") AND ("requirements engineering" OR "requirements elicitation" OR "requirements activities").

Foram encontrados um total de 155 artigos inicialmente aderentes à seleção nos 4 portais de busca. Após a seleção inicial, realizaram-se dois filtros sequenciais, somente de título e de resumo, removendo-se da lista de artigos os casos de falso-positivo, que não possuíam qualquer relação com a temática de pesquisa. Intencionalmente, não foi realizado um filtro de datas em qualquer momento da revisão, nem quanto ao formato do documento a ser utilizado. Optou-se por não se fazer inicialmente um filtro de língua em que o artigo se encontrava disponível, embora após a etapa de filtro de resumo, todos os artigos já se encontravam em inglês ou português. Essa atividade foi feita pelo primeiro autor do artigo. A Tabela 1 apresenta os resultados obtidos em cada portal.

**Tabela 1 – Resultados obtidos da busca nos portais acadêmicos**

Portal	Resultados	Filtro de Título	Filtro de Resumo
Springer	57	32	6
ACM	18	7	4
IEEE	37	13	4
Science Direct	43	22	0
<b>Total</b>	<b>155</b>	<b>74</b>	<b>14</b>

Feita a seleção dos 14 artigos, a revisão foi conduzida através da leitura e análise de cada artigo de forma independente. Para responder às perguntas de pesquisa, realizou-se a catalogação e posterior categorização dos itens de interesse a partir das declarações dos pesquisadores em seu texto original. Buscou-se não se fazerem inferências a respeito de cada experimentação conduzida. Essa etapa foi realizada por apenas o primeiro autor do artigo, com a revisão pelo segundo autor das declarações e da categorização.

## 4 Resultados

Este capítulo contempla o resultado da revisão, apresentando primeiro uma sumarização dos estudos selecionados e depois tratando das questões de pesquisa.

### 4.1 Sumarização de Estudos Selecionados

Esta seção apresenta os 14 artigos selecionados, divididos a partir da forma como o *Design Thinking* foi aplicado (conceito apresentado na seção 2.1). Os artigos são selecionados foram: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], e sua referência completa se apresenta na seção de Referências Bibliográficas.

A abordagem adiantada do DT foi o processo mais comum, verificando-se 9 artigos que se enquadram nesta categoria. O artigo [1] tem como contexto dois exemplos de

caso em uma Consultoria de Tecnologia (Adesso); [2] trata do uso de DT para a modernização de um sistema; [12] relata um workshop realizado em companhia de desenvolvimento de software e [13] relata um estudo de caso de DT e gamificação em uma ferramenta de priorização de requisitos. Os demais trabalhos são relatos de uso do DT em projetos de software específicos: para marinheiros profissionais (competitivos) [4], para uma farmácia digital (*e-pharmacy*) [8], para a gestão de energia em uma pequena comunidade [9] e um sistema de gerenciamento acadêmico [10].

Dentre os artigos estudados, somente um deles foi categorizado como DT infundido, o [5] que trata de um estudo de caso de elaboração de um aplicativo de armazenagem de dados em que se utilizou a técnica nomeada de “Converge”, em que o DT é utilizado em fases específicas ao longo do desenvolvimento como ferramenta de inovação.

Por fim, 4 artigos se adequam à abordagem do DT contínuo: o [6] que trata da criação de protótipos virtuais e a experiência de uso do DT ao longo de um projeto de 3 anos de duração; o [7] que descreve um projeto voltado ao entendimento das manifestações do DT por diferentes usuários em uma companhia de desenvolvimento de software; o [11] que é um estudo a respeito da integração contínua do DT com a ER no escopo de um sistema industrial de utilidades; e o [14] que discute a migração dos processos de uma companhia de desenvolvimento de software para a utilização do DT de forma contínua.

## 4.2 RQ1

A primeira pergunta de pesquisa (RQ1) teve o objetivo de identificar problemas associados com o DT para a ER. Dentre os 14 artigos, foram identificados um total de 29 pontos levantados pelos autores como problemas ou desafios da abordagem. Como forma de categorização adotada, os itens foram organizados segundo a classificação de categorias da área de conhecimento de Requisitos de Software do SWEBOK [25]. A **Tabela 2** apresenta as categorias com o resultado da RQ1. O detalhamento dos problemas encontrados dentro de cada categoria é apresentado na sequência.

**Tabela 2 – Número de ocorrências de problemas com a técnica do *Design Thinking* nos artigos de interesse por categoria da área de conhecimento de Requisitos de Software segundo o SWEBOK e artigos onde a temática se verificou.**

<b>Categoria</b>	<b>Número de Ocorrências</b>	<b>Artigos com a temática</b>
Processos de requisitos	9	[4, 5, 6, 7, 11]
Elicitação de requisitos	3	[1, 6, 11]
Análise de requisitos	3	[7, 11, 4]
Especificação de requisitos	3	[5, 9, 11]
Validação de requisitos	1	[6]
Outros	10	[7, 11, 14]

**Processos de requisitos.** Dentro de processos de requisitos, as seguintes questões são abordadas:

- Com o uso da abordagem, não é óbvio diferenciar entre cenários “*as-is*” e cenários “*to-be*” da perspectiva do usuário para compor o mapa de histórias [4].
- A correta aplicação da abordagem e a obtenção de bons resultados são diretamente dependentes da experiência e conhecimento das pessoas que participam da atividade de DT, tanto a respeito da abordagem e métodos de Engenharia de Software e de processos quanto do contexto da informação [5, 6, 7, 11].
- O DT enfrenta desafios referentes à disponibilidade dos clientes no momento adequado. Enquanto um estágio está em construção, é natural que esteja sendo discutida outra questão, e os resultados não são imediatamente relevantes [7, 11].

**Elicitação de requisitos.** A etapa de eliciação de requisitos foi caracterizada pelos seguintes problemas:

- Não é sempre possível identificar as necessidades dos usuários através de perguntas, sendo necessário de alguma forma rastreá-las [1].
- O DT possui dificuldades de rastreabilidade da informação. Os requisitos elicitados são capturados via post-its ou protótipos, de forma desestruturada para agilizar a colaboração do time e dar velocidade ao processo [11].

**Análise de requisitos.** Na análise de requisitos verificaram-se os pontos destacados:

- Há problemas quando os resultados do DT conflitam com as expectativas da gestão ou com o foco pré-definido do produto (priorização de requisitos) [7, 11].
- Não são feitas expectativas a respeito de quais requisitos levantados em um workshop de DT serão endereçados futuramente [14].

**Especificação de requisitos.** Verificaram-se as seguintes questões na etapa de especificação de requisitos:

- Os entrevistados possuem diferentes necessidades e tarefas para endereçar. É bastante difícil a construção com altos padrões de qualidade e usabilidade que atendessem às expectativas de todos os usuários, particularmente considerando a restrição de tempo [5].
- O processo de DT negligencia requisitos não-funcionais (RNF) como requisitos de segurança ou desempenho [9, 11].

**Validação de requisitos.** Dentro da validação de requisitos, houve uma manifestação:

- Para sistemas complexos multi-usuários, a criação de protótipos (fundamental ao DT) que capturem todos os processos subjacentes e fluxos de dados é muito custosa, tornando-se proibitiva [6].

**Outros.** Nesta categoria foram listadas as questões encontradas que não se encaixavam diretamente em nenhuma das definidas pelo SWEBOK:

- O compartilhamento de resultados das atividades de DT e a garantia de um bom resultado no produto final são uma barreira. Os resultados precisam ser compartilhados com toda a equipe de maneira eficiente no tempo, e são só interessantes para a maioria assim que a *feature* já está em desenvolvimento [7].
- Devido ao pequeno tempo de planejamento, o DT pode levar a uma arquitetura inapropriada [11].
- O DT pode mostrar problemas quanto a estimativas imprecisas de esforço [11].
- Os passos iniciais da abordagem de DT aparentam ser demorados e ineficazes. Nos subsequentes, há problemas de flutuação de moral nos times envolvidos [14].
- Devido à natureza não convencional colaborativa das atividades de DT e à necessidade de mudar o *mindset* dos clientes, um desafio é o de preparar os clientes antecipadamente para as atividades do primeiro encontro [14].
- O alto grau de interação durante os workshops de DT indica que a logística destes deve ser preparada cuidadosamente e já de forma antecipada [14].
- Se um time de DT é composto de pessoas de hierarquias muito diferentes, é possível que se perca o efeito global (de inovação), pois haverá um problema de comunicação e as pessoas tenderão a participar menos [14].
- Os problemas subjacentes identificados durante uma atividade de DT por vezes são obscurecidos pelas muitas soluções sugeridas [14].
- O foco do DT no futuro facilita a criação de expectativas que não são realistas. Os clientes podem ter a sensação de que posteriormente somente os pontos mais fáceis são endereçados, e as promessas de elementos vistos em *workshop* são deixadas de lado [14].

**Discussão.** Os artigos apresentam que há, efetivamente, problemas com a abordagem do DT para a ER em todas as subetapas do SWEBOK.

O maior número de ocorrências se deu na etapa de processos de requisitos. Esta questão é também reflexo da maneira como o DT é abordado (processo adiantado) na maioria dos artigos.

Uma questão comum dentre vários artigos é a de que o DT não se trata de um processo rígido, podendo ser aplicado em vários formatos e com diferentes níveis de experiência. Há evidências para a existência de uma curva de aprendizado, que conduz a melhores resultados quando o processo é aplicado de forma mais evoluída. [7] sugere por exemplo que a implementação do DT depende da experiência e conhecimento dos participantes. Dessa forma, há indícios de que bons resultados estão diretamente correlacionados a conhecimentos sobre o contexto da informação e sobre o próprio método de DT [5, 6, 7, 11].

Alguns dos problemas encontrados empiricamente já são endereçados diretamente pelos próprios pesquisadores e podem ser mitigados com boas práticas de planejamento antecipado às rodadas de DT. Um tema comum a [7, 11] foi, por exemplo, a disponibilidade dos clientes, um problema resolvido de forma simples com um planejamento antecipado [11].

Dentre os problemas, alguns são de natureza organizacional da empresa onde o processo é conduzido: a presença de pessoas de hierarquias muito distintas em um

mesmo grupo inibindo a comunicação [14] , a priorização de requisitos e o potencial conflito gerencial [7, 11].

Há também indícios de que exista um tamanho ideal de projeto para o qual o DT apresente melhores resultados, embora a margem de tolerância possa ser razoavelmente grande. Em atividades muito pequenas ou onde a solução já é conhecida, a técnica não se justifica (por exemplo para a resolução de *bugs* ou para o planejamento de atividades [7]), e em projetos muito grandes ou muito complexos, a prototipação é custosa e se torna proibitiva [6].

### 4.3 RQ2.

A segunda pergunta de pesquisa (RQ2) buscou identificar as propostas dos autores para melhoria do processo de DT. Foram identificadas 10 sugestões de melhoria que são apresentadas a seguir:

- [1] Além dos papéis clássicos de gestão de projeto, é interessante ter-se uma pessoa com responsabilidade estabelecida de projetar o software de acordo com os requisitos do usuário, de forma a garantir o foco necessário dos projetos no *design* do ponto de vista do usuário e evitar conflitos de interesse na medida do possível, dado que um arquiteto pode, em caso de dúvida, decidir contra o usuário a favor de uma solução técnica mais simples.
- [2] Os autores sugerem a participação de um designer para facilitar o processo de DT, e apontam para uma correlação direta quanto à qualidade dos protótipos criados.
- [4] Na primeira parte divergente do processo de DT, os autores sugerem a utilização de uma "pesquisa 360 graus", se referindo à observação e entrevistas com usuários e fontes secundárias, como analistas, pensadores e competidores em domínios análogos e adjacentes (buscar além de informações diretas com o usuário, benchmarks).
- [4] Mapas de história de usuários podem servir como uma ferramenta para preencher a lacuna entre a empatia e *insights* adquiridos com práticas de DT e o *backlog* para construir a solução.
- [5] O processo de DT pode ser abordado em rodadas.
- [6] Para permitir que todas as pessoas envolvidas no projeto desenvolvam um entendimento comum, os autores utilizaram protótipos de modelos formais combinados com animações interativas específicas do domínio, de forma a facilitar com que os usuários finais fornecessem *feedback* sobre o modelo durante a simulação iterativamente.
- [9] Os autores apontam que a utilização de artefatos físicos podem ser eficientes para representar uma ideia ou conceito abstratos, simplificando os objetivos do projeto para os participantes.
- [14] Assim como para mudanças em uma companhia, o suporte intenso da gestão para o DT é essencial como forma de evitar os passos iniciais que podem parecer ineficazes ou demorados, e as eventuais flutuações de moral nas equipes.

- [14] Para preparar os participantes de forma antecipada, somente mandar uma agenda por e-mail não é suficiente, pois pode levar a reações de indiferença ou negativas. É especialmente importante explicar com clareza à gestão o que será requerido dos colaboradores, e ativamente envolver a gestão na organização de *workshops* de DT.
- [14] É importante possuir um tomador de notas dedicado, dado que os problemas subjacentes identificados na atividade são ofuscados pelas muitas sugestões sugeridas. Seria necessário gravar não somente as palavras ditas, mas até o comportamento não-verbal dos participantes, por exemplo ao decidir em prioridades para *features*. Além disso, a riqueza de informação produzida em artefatos durante o *workshop* deve ser cuidadosamente arquivada e indexada para dar suporte a acompanhamentos futuros.

**Discussão.** As sugestões de melhorias ao método foram compiladas de verificações práticas pelos autores para obtenções de melhores resultados. Não é razoável supor que somente com a aplicação destas seria factível a mitigação de todos os problemas compilados dentre os artigos. Podem, entretanto, ser pontos interessantes para evitar principalmente alguns dos problemas que advêm da eventual pequena experiência de aplicantes do método (em especial os que realizam pela primeira vez), como a necessidade de estruturar o processo de antemão e de fomentar uma estrutura organizacional que permita a aplicação eficiente do método e alinhada com os interesses de longo prazo da própria empresa.

As recomendações são, em sua maioria, voltadas para a aplicação do DT adiantado, até dado que correspondeu à maior parte dos artigos. Também não são recomendações formais e com efetividade comprovada, o que se justifica pelo formato dos artigos que foram utilizados para a RSL: o seu intuito (para todos) não é o de buscar a melhoria do processo, mas sim relatar o seu uso dentro da Engenharia de Software.

## 5 Conclusões e Trabalhos Futuros

A RSL, que possuía o intuito principal de identificar a existência na prática de problemas associados à técnica de Design Thinking para a Engenharia de Requisitos, foi bem sucedida, uma vez que os artigos identificados no filtro de seleção apresentado na metodologia de pesquisa mostraram de forma clara pontos de fraqueza da técnica ainda não solucionados pelo estado atual de sua abordagem e com as ferramentas disponíveis dentro da ER. A seção 4 apresentou conclusões relevantes para cada pergunta de pesquisa.

Verificou-se que a presença de diversos dos problemas relatados está condicionada pela presença de alguns fatores externos à técnica e que não são diretamente contornados por ela, dos quais destacam-se: a estrutura empresarial, a limitação de recursos disponíveis, a experiência dos participantes do processo, a inadequação do tamanho do projeto, a inexistência de mecanismos para tomada de decisão a respeito da priorização de requisitos.

Dentre os problemas identificados destaca-se a problemática da negligência de requisitos não-funcionais (RNF). Uma das percepções é de que a ênfase exclusivamente no cliente e no problema que se está tentando resolver foca em alguns RNF em especial de usabilidade, tomando como garantidos (e ignorados) outros RNF, como segurança, performance e manutenibilidade.

Uma questão que também surge diz respeito à definição formal da abordagem. Como colocado por [16], não há um processo rigoroso de aplicação do DT, e assim, é de se esperar que novos aplicantes da técnica sofram de alguns dos problemas aqui relatados devido à inexperiência. De forma a contornar este fator, a revisão busca justamente identificar os problemas advindos da técnica de forma a fornecer insumos para estudos futuros voltados à sua melhoria.

### 5.1 Limitações

A RSL buscou por identificar questões problemáticas e sugestões de melhoria da técnica do Design Thinking associada a projetos de Engenharia de Software que possuíssem relação direta com a Engenharia de Requisitos. Apesar de a abordagem se manifestar sobremaneira nesta etapa, é possível que existam também potenciais impactos em outras etapas do ciclo de vida do software que não são capturadas por esta revisão. Desta forma, reconhece-se que a revisão seja não exaustiva, dado que se buscou por trabalhos que falavam explicitamente sobre a Engenharia de Software, mas é possível que outros artigos sobre o tema tenham sido ignorados. Também ressalta-se que, apesar de seguir as recomendações de Kitchenham et al. [24] para a condução de revisão sistemática com a redução de vies dos pesquisadores, a revisão individual de cada dado extraído dos artigos não passou por uma atividade de *peer review*.

Ademais, é importante ressaltar que a principal questão desta revisão é a RQ1, referente aos problemas associados. Assim, as sugestões de melhoria podem ser vistas como um subproduto desta, e poderia ser razoável a criação de uma RSL específica para responder a RQ2 de forma mais abrangente.

Ainda, uma limitação natural da RSL é a de que nem sempre é possível capturar todas as informações de interesse para responder uma pergunta de pesquisa somente através do artigo publicado, dado que a revisão corresponde a um estudo secundário e, portanto, se limita nos elementos que os autores de cada artigo julgaram mais relevantes de serem relatados.

### 5.2 Trabalhos Futuros

A contribuição deste trabalho é indicar pontos para pesquisas futuras que busquem melhorar a integração do *Design Thinking* com as atividades de Engenharia de Requisitos. Assim sendo, e tendo em mente o caráter não-exaustivo da revisão, trabalhos futuros podem tratar da identificação de questões complementares que por vezes não são relatadas em um artigo e não podem ser capturadas somente com uma RSL.

Adicionalmente, outros trabalhos poderiam se aprofundar nos problemas encontrados, de forma a verificar se eles são de fato frequentes, relevantes e/ou já endereçados de alguma forma por outros autores.

Além disso, embora este estudo permita uma melhor compreensão a respeito da técnica, julga-se que se fará necessária a realização de estudo empírico que possa avaliar os aprimoramentos que possam surgir, de forma a verificar a sua eficácia.

## Referências

1. 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. DOI: 10.1007/978-3-319-73897-0\_7.
2. Canedo, E., Costa, R.: The Use of Design Thinking in Agile Software Requirements Survey: A Case Study. Em: Design, User Experience, and Usability: Theory and Practice, pp.642-657. 2018. DOI: 10.1007/978-3-319-91797-9\_45.
3. Braz, R., Merlin, J., Trindade, D., Ribeiro, C.: Design Thinking and Scrum in Software Requirements Elicitation: A Case Study. Em: Design, User Experience, and Usability. Design Philosophy and Theory, pp.179-194. 2019. DOI: 10.1007/978-3-030-23570-3\_14
4. Schimmer, T., Meyer, J.: Intertwining Lean and Design Thinking: Software Product Development from Empathy to Shipment. Em: Software Usability in Small and Medium Sized Enterprises in Germany: An Empirical Study (pp.217-237). 2012. DOI: 10.1007/978-3-642-31371-4\_13.
5. Ximenes, B., Alves, I., Araújo, C.: Software Project Management Combining Agile, Lean Startup and Design Thinking. Em: A Posture HCI Design Pattern for Television Commerce Based on User Experience (pp.356-367). 2015. DOI: 10.1007/978-3-319-20886-2\_34.
6. Gabrysiak, G., Giese, H.: Virtual Multi-User Software Prototypes III. Em: Design Thinking Research - Measuring Performance in Context, Understanding Innovation, pp. 263–284, Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-31991-4\_15.
7. Dobrigkeit, F., Paula, D.: Design Thinking in Practice: Understanding Manifestations of Design Thinking in Software Engineering. 27th ACM Joint Meeting, 2019. DOI: 10.1145/3338906.3340451.
8. Carroll, N., Richardson, I.: Aligning Healthcare Innovation and Software Requirements through Design Thinking. International Workshop on Software Engineering in Healthcare Systems, Austin, TX, USA, 2016. DOI: 10.1145/2897683.2897687.
9. Newman, P., Ferrario, M., Simm, W., Forshaw, S.: The Role of Design Thinking and Physical Prototyping in Social Software Engineering. 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE). 2015. DOI: 10.1109/ICSE.2015.181.
10. Ferreira, B., Barbosa, S., Conte, T.: Creating Personas Focused on Representing Potential Requirements to Support the Design of Applications. 17th Brazilian Symposium, 2018. DOI: 10.1145/3274192.3274207.
11. Hehn, J., Uebernickel, F.: The Use of Design Thinking for Requirements Engineering: An Ongoing Case Study in the Field of Innovative Software-Intensive Systems. 26th IEEE International Requirements Engineering Conference, Banff, Canada, 2018. DOI: 10.1109/RE.2018.00-18.
12. Levy, M., Huli, C.: Design Thinking in a Nutshell for Eliciting Requirements of a Business Process: A Case Study of a Design Thinking Workshop. 2019 IEEE 27th International Requirements Engineering Conference (RE), 2019. DOI: 10.1109/RE.2019.00044.

13. Piras, L., Dellagiacoma, D., Perini, A.: Design Thinking and Acceptance Requirements for Designing Gamified Software. 2019 13th International Conference on Research Challenges in Information Science (RCIS), 2019. DOI: 10.1109/RCIS.2019.8876973.
14. Mahé, N., Adams, B., Marsan, J., Templier, M., Bissonnette, S.: Migrating a Software Factory to Design Thinking - Paying attention to people and mindsets. *IEEE Software*, 2019. DOI: 10.1109/MS.2019.2958646.
15. Hehn, J., Mendez, D., Uebernickel, F., Brenner, W., Broy, M.: On Integrating Design Thinking for a Human-centered Requirements Engineering. *IEEE Software*, 2019.
16. Waidelich, L., Richter, A., Kölmel, B., Bulander, R.: Design Thinking Process Model Review. *IEEE Software*, 2018.
17. Vetterli, C., Brenner, W., Uebernickel, F.: From Palaces to Yurts: Why Requirements Engineering Needs Design Thinking. *IEEE Internet Computing* 17. 03/2013.
18. Chen, B., Atlee, J.: Research Directions in Requirements Engineering. *Proc. Future Software Eng. (FOSE 07)*. *IEEE CS*, 2007. pp. 285-303.
19. Volkova, T., Jakobson, I.: Design Thinking as a Business Tool to Ensure Continuous Value Generation. *Intellectual Economics* 10 (10 (2016)). pp. 63-69.
20. Brenner, W., Uebernickel, F.: Design Thinking for Innovation: Research and Practice. 1<sup>st</sup> ed., Springer International Publishing, Cham, s.l., 2016.
21. Plattner, H., Meinel, C., Leifer, L.: Design Thinking: Understand – Improve – Apply. Springer-Verlag Berlin Heidelberg, 2011.
22. Brown, T. Design Thinking. *Harv Bus Rev*. 2008.
23. Kelley, T., Kelley, D. Creative confidence: Unleashing the creative potential within us all. Crown Business, 2013.
24. Kitchenham, B., Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, J.: Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology* 51, 2009.
25. Bourque, P., Fairley, R.: Guide to the Software Engineering Body of Knowledge, Version 3.0. IEEE Computer Society, 2014.
26. Beyhl, T., Giese, H.: Connecting Designing and Engineering Activities III. In: Plattner, H., Meinel, C., and Leifer, L. (eds.) *Design Thinking Research: Making Design Thinking Foundational*. pp. 265–290. Springer International Publishing, Cham (2016). [https://doi.org/10.1007/978-3-319-19641-1\\_16](https://doi.org/10.1007/978-3-319-19641-1_16).
27. Uebernickel, F., Hehn, J.: Towards an Understanding of the Role of Design Thinking for Requirements Elicitation - Findings from a Multiple-Case Study. Presented at the August 16 (2018).
28. Pereira, J.C., Russo, R. de F.S.M.: Design Thinking Integrated in Agile Software Development: A Systematic Literature Review. *Procedia Computer Science*. 138, 775–782 (2018). <https://doi.org/10.1016/j.procs.2018.10.101>.
29. Park, H., McKilligan, S.: A Systematic Literature Review for Human-Computer Interaction and Design Thinking Process Integration. In: Marcus, A. and Wang, W. (eds.) *Design, User Experience, and Usability: Theory and Practice*. pp. 725–740. Springer International Publishing, Cham (2018). [https://doi.org/10.1007/978-3-319-91797-9\\_50](https://doi.org/10.1007/978-3-319-91797-9_50).