

Thinking about gender: Combinando Design Thinking e GenderMag na elicitação de requisitos para um software de apoio a avaliação de UX.

Elizamara K. Almeida¹[0000-0002-9344-940X], Vinícius Monteiro¹[0000-0002-4983-3260], Franciane Alves¹[0000-0001-7304-2779], Brenda Aguiar¹[0000-0002-4794-4557], Leonardo Marques¹[0000-0002-3645-7606], Bruno Gadelha¹[0000-0001-7007-5209] and Tayana Conte¹[0000-0001-6436-3773]

Universidade Federal do Amazonas, Av. General Rodrigo Octavio Jordão Ramos,
1200 - Coroado I, Manaus - AM, 69067-005
{elizamara.almeida,vfm, faa2, brenda.aguiar, lcm, bruno,
tayana}@icomp.ufam.edu.br,
www.ufam.edu.br

Resumo A elicitação de requisitos é a primeira etapa do processo de Engenharia de Requisitos. Essa etapa é fundamental para o desenvolvimento de *software* visto que, busca entender quais são as necessidades do usuário. *Design Thinking* (DT) é uma abordagem centrada no usuário, buscando identificar as suas necessidades e desejos, viabilizando o desenvolvimento de um *software* mais adequado. Nesse sentido, DT é uma boa alternativa para elicitar requisitos. Apesar de existirem várias abordagens de implementação do DT, não há uma amplamente adotada com ênfase em inclusão de gênero, na fase de elicitação de requisitos. Todavia, existe um método que avalia essa questão de inclusão, chamado *GenderMag* (GM). Este artigo propõe investigar os benefícios do uso conjunto do DT e GM no processo de elicitação de requisitos. Para isso, foi feito o levantamento de requisitos utilizando técnicas de DT, para um *software* de apoio a avaliação UX. Em seguida foi aplicado o GM no protótipo gerado. Os resultados do uso conjunto dessas abordagens consistem numa lista de requisitos obtidos através do DT que apoiam o uso do *software* por quem possui algumas facetas cognitivas, comuns no gênero masculino. Por outro lado, o GM forneceu uma lista de correção que possibilitou a adequação do protótipo incluindo outras facetas cognitivas. A principal contribuição deste trabalho consiste em mostrar como o uso do GM permitiu a concepção de um *software* mais inclusivo, mostrando os benefícios do seu uso conjunto com DT, além de fomentar a importância de levantar requisitos voltados à inclusão de gênero.

Keywords: Elicitação de Requisitos · Design Thinking · GenderMag · UX · Inclusão · Gênero.

1 Introdução

O *Design Thinking* (DT) fornece um conjunto de técnicas inspiradas em design centrado no usuário para resolução de problemas, incitando a criatividade e inspiração para uma solução [28]. Pinto et al. [9] disseram que o DT é uma abordagem promissora em crescente utilização nas etapas da Engenharia de Requisitos (ER) por focar no cliente final do *software*. Afinal, requisitos são a base de todo produto de *software* e, conseqüentemente, a ER desempenha um papel importante no desenvolvimento do *software* [20].

A ER está inserida na Engenharia de software (ES), que tem evoluído durante os anos, porém ainda possui limitações relacionadas à sua aplicação na prática [27]. Uma dessas limitações se refere a questões sociais, como a inclusão de gênero. Nesse sentido, Burnett et al. [7] apontam pesquisas que mostraram que as diferenças individuais em como as pessoas usam recursos de *software* tendem a se agrupar por gênero e, além disso, muitos desses recursos são projetados de forma a favorecer mais um gênero que o outro. A partir dessa observação, Burnett et al. [7] desenvolveram o método *GenderMag* (GM) sendo proposto com o intuito de abordar essas questões de gênero para avaliação de *software* [7].

Portanto, o trabalho apresentado neste artigo busca investigar os benefícios que se possam atingir ao utilizar técnicas de DT e o método GM em conjunto para elicitare requisitos para o desenvolvimento de um *software* mais inclusivo. Para isso, utilizou-se como base o desenvolvimento de uma aplicação cujo objetivo é auxiliar a realização de avaliações de UX no contexto de entretenimento imersivo. O DT e o GM foram utilizados para diferentes objetivos. O primeiro foi usado para levantar os requisitos voltados para os interesses dos *stakeholders*. Já o segundo foi usado para detectar viés de gênero no uso do *software*.

Os resultados revelam que utilizar DT juntamente de GM pode ser uma solução viável para a atividade de elicitação de requisitos e o desenvolvimento de um *software* adequado às questões de gênero. O DT permite identificar as necessidades dos *stakeholders* e transformá-las em requisitos, tornando o *software* mais adequado ao público alvo. Por outro lado, o GM apoia a inclusão de gênero, avaliando se o *software* dá suporte a diferentes formas de aprendizado sobre como usá-lo.

Este artigo está estruturado da seguinte forma: a Seção 2 contém o Referencial Teórico, descrevendo trabalhos relacionados e conceitos relevantes para a compreensão deste artigo. A Seção 3 contém a Metodologia, descrevendo o *Framework* que inspirou a construção do *software*, como o estudo foi conduzido e as técnicas utilizadas em cada fase do ciclo de DT. A Seção 4 apresenta os resultados do estudo. A Seção 5 apresenta a discussão baseada nos resultados obtidos. E, a Seção 6 apresenta as conclusões deste artigo e trabalhos futuros.

2 Referencial Teórico

2.1 Design Thinking

O *Design Thinking* (DT) é a forma como os *designers* pensam e aplicam seus processos mentais para projetar objetos, serviços ou sistemas [13]. Na prática, o DT é uma abordagem centrada no ser humano que gera soluções inovadoras e colabora para solucionar problemas complexos [5]. Essa abordagem usa a sensibilidade e os métodos do *designer* para combinar as necessidades das pessoas com o que é tecnologicamente factível. Também, combinado a uma estratégia de negócios viável, podendo ser convertida em valor para o cliente e oportunidade de mercado [5].

Nesta pesquisa, o modelo base de DT adotado foi proposto por Brown que consiste em um ciclo de três fases: inspiração, ideação e implementação [5]. Cada fase do ciclo será explicada a seguir. Vale lembrar que um processo de DT não é linear, é iterativo, ou seja, o progresso se faz através de tentativas sucessivas de refinamento em que fases podem se repetir quantas vezes necessárias até gerar um produto desejável.

Brown define as três fases da seguinte maneira: a fase de inspiração é onde se aproxima do contexto do problema, é a fase de imersão onde é realizada a coleta e síntese de dados para entender o problema e identificar quem são as pessoas-chave envolvidas no resultado do projeto. A fase de ideação é onde se gera e discute ideias e sugestões de como solucionar os problemas levantados na fase anterior, considerando as necessidades e desejos das pessoas envolvidas. A fase de implementação é o momento em que as ideias abstratas geradas na etapa anterior ganham conteúdo material.

2.2 GenderMag

O *GenderMag* (GM) é um método para detectar e corrigir “*bugs*” de inclusão de gênero (viés de gênero) em *softwares* [26]. Ele integra um método passo a passo cognitivo (Cognitive Walkthrough) [21] com o uso de três personas baseadas em pesquisas de gênero, são elas “Abi”, “Pat” e “Tim”. Cada persona assume determinadas facetas, sendo características relacionadas a aspectos cognitivos em estilos de resolução de problemas.

O GM é baseado em cinco facetas das quais cada um de seus valores estão estatisticamente associados a um gênero [7]. São elas: motivação, estilo de processamento de informações, autoeficácia computacional, atitudes em relação a risco e estilo de aprendizado de tecnologia. A Tabela 1 mostra as características de cada faceta. Vale ressaltar que, para este estudo, o método GM foi aplicado seguindo suas diretrizes para descobrir viés de gênero no protótipo desenvolvido ao final do ciclo de DT.

O *Cognitive Walkthrough* [21] é um método passo a passo para avaliar a capacidade de aprendizagem de *software* sem serem feitos testes em laboratórios de usabilidade de forma empírica. O GM integra o método modificado que consiste em responder 2 perguntas após uma determinada ação do usuário na interface

Tabela 1. Descrição das facetas do método *GenderMag*.

Facetas	Características
Motivação	Se refere a motivos que levam a pessoa a utilizar o software. Pesquisas descobriram que mulheres são mais propensas a serem motivadas a utilizar tecnologias pelos fins que se pode alcançar com ela. Já os homens são mais motivados a utilizar tecnologias por interesses de prazer [6] [8].
Estilo de processamento de informações	Mulheres tendem a coletar informações de forma ampla, obtendo um conhecimento prévio do problema antes de tentar resolvê-lo. Já os homens processam informações seletivamente, seguindo a informação mais promissora e então recuando caso a opção não funcione [17]. Cada estilo tem suas vantagens, mas ambos ficam em desvantagem quando não são suportados pelo software [18].
Autoeficácia computacional	É a confiança que uma pessoa tem sobre seu sucesso em realizar uma tarefa, influenciando em suas estratégias e em como lidar com obstáculos [4]. Estudos empíricos mostraram que mulheres costumam ter menor autoeficácia usando computadores do que homens, e isso pode afetar sua conduta com a tecnologia [8].
Atitudes em relação ao risco	A aversão à riscos utilizando-se um software pode impactar as decisões dos usuários sobre quais recursos usar [10]. Pesquisas apontam que, mulheres tendem a ser mais avessas a riscos do que homens em várias áreas de tomada de decisão [12].
Estilo de aprendizado de tecnologia	Remete a maneira como a pessoa prefere aprender uma tecnologia. Pesquisas relatam que mulheres são menos propensas a explorar recursos de softwares não familiares do que homens. No entanto, se mulheres exploram, são geralmente mais propensas a pensar sobre o que estão fazendo e podem ter vantagens de aprendizado com esse processo mais do que homens [4] [6].

do *software*. A primeira é “O usuário saberá o que fazer nesta etapa?” e a segunda é “Se o usuário fizer a coisa certa, ele saberá que fez a coisa certa e está progredindo em direção a meta?”.

Para executar o método GM de modo a avaliar um *software*, a equipe de avaliação seleciona uma ou mais personas do conjunto de persona do GM. Em seguida, definem-se objetivos a serem atingidos durante o uso do *software* e uma sequência ideal de ações para realizar cada objetivo. A partir disso, é dado início à análise seguindo o passo a passo especializado em gênero com cada persona selecionada, como mostra o exemplo abaixo:

1. (Objetivo Geral)
2. (Subobjetivo): a *persona* formularia esta meta secundária como uma etapa para seu objetivo geral? Por quê?
 - R: (O avaliador aponta as facetas pertinentes)
3. (Q1 - Ação 1): a *persona* saberá o que fazer nesta etapa? Por quê?
 - R: (O avaliador aponta as facetas pertinentes)
4. (Q2 - Ação 1): se *persona* fizer a coisa certa, ela saberá que o fez e que está progredindo em direção ao seu objetivo? Por quê?

- R: (O avaliador aponta as facetas pertinentes)

2.3 Trabalhos Relacionados

A ER está preocupada em identificar, modelar, comunicar e documentar os requisitos de um sistema e o contexto onde o sistema será usado [11]. O DT é utilizado na ER como processo para produzir prototipação simples e prática até atingir uma solução viável [23]. Mesmo que os protótipos também sejam utilizados em metodologias ágeis e na própria área de ER, a grande vantagem do DT se dá na detecção de possíveis problemas que possam ser resolvidos o quanto breve possível no processo [23].

Hehn e Uebersnickel [15] investigam como o DT e o ER podem trabalhar juntos em projetos de desenvolvimento de *software*. Para isso foi conduzido um estudo de caso explorando um projeto baseado na indústria de serviços públicos na Europa. Os resultados sugerem que o DT tem grande potencial de apoiar as práticas de ER atuais e vice-versa. Mahé et al. [16] apresentam um relato de experiência sobre a integração do DT no processo de desenvolvimento de *software* de uma empresa de tecnologia. Os resultados são positivos, apontando melhoria na compreensão dos requisitos, melhora no desempenho das equipes de desenvolvimento e maior proximidade com o cliente.

Durante o desenvolvimento de um *software*, considerar o que o usuário deseja ajuda a proporcioná-lo uma experiência mais satisfatória ao utilizar o produto final. A *User Experience* (UX) diz respeito a como um usuário se sente sobre um artefato ao usá-lo no mundo real [1]. O foco principal da UX é na criação de uma experiência positiva para o usuário, principalmente através de prazer, alegria e outros sentimentos ligados a sensação de bem-estar [1]. Sendo assim, um conhecimento profundo do usuário e do contexto de uso é importante para se projetar uma UX ideal.

Torrezan et al. [22] integram técnicas de DT com diretrizes de UX para desenvolver um aplicativo móvel intitulado Brasil +9, que permite adicionar o dígito extra para os números existentes em sua lista telefônica. O desafio era oferecer uma experiência agradável ao usuário. Como resultado, o aplicativo teve alto índice de sucesso em comparação a outros aplicativos com o mesmo uso. Paula et al. [19] abordam o desenvolvimento de um sistema de recomendação educacional, chamado *RecOAComp*, que indica objetos de aprendizagem a partir das competências que o usuário necessita. Aqui a abordagem de DT foi utilizada com o intuito de contemplar especificações relacionadas a UX. A versão final do aplicativo obteve bons resultados.

Tratando-se de conhecer profundamente o usuário para fornecer uma boa UX, se torna relevante abordar questões de inclusão de gênero no *software*. De acordo com Burnett et al. [7], o GM “visa fornecer uma maneira sistemática e prática para que os profissionais de *software* (desenvolvedores de *software*, especialistas em UX, etc.) encontrem questões de inclusão de gênero no *software*, mesmo sem experiência em pesquisa de gênero”. Nesse sentido, pode-se dizer que o GM foi elaborado preocupando-se em fazer com que o *software* forneça uma boa UX ao considerar os diferentes aspectos cognitivos, deixando o *software* mais adequado

a uma gama maior de estilos de aprendizado de uso. Dessa forma, o diferencial deste artigo é investigar como o uso do DT e o GM podem complementar um ao outro para se obter um *software* mais completo e inclusivo.

3 Metodologia

3.1 Framework Immersive UX

O *Immersive UX* é um *Framework* de avaliação UX no contexto de entretenimentos imersivos. Esse *Framework* busca avaliar a UX da imersão experienciada por um público a partir de três medidas de UX consideradas importantes para a avaliação de uma experiência imersiva: *Flow*, *Presence* e *Engagement* [3].

O *Framework* é constituído de três etapas: expectativa, desempenho e satisfação [3]. Durante essas etapas, o avaliador coleta diferentes dados do público participante do entretenimento, em seguida calcula cada medida e, assim, avalia a UX da experiência imersiva. Com o intuito de facilitar a avaliação de UX através do *Framework*, decidiu-se desenvolver um software que contemplasse a avaliação do *Framework*. Sendo assim, a construção desse software visa apoiar o avaliador na aplicação do *Framework Immersive UX*.

A equipe de desenvolvimento responsável pela elicitação dos requisitos e construção do protótipo, além da aplicação do DT e GM, é composta por quatro pesquisadores da área de UX voltada para *softwares* inovadores. Para o desenvolvimento desse *software*, a equipe utilizou o ciclo de DT em três fases, descrito na Seção 2.1, com o objetivo de descobrir as necessidades dos usuários, levantar os requisitos e assim construir seu protótipo. Nas próximas seções será relatado como ocorreu cada fase. Além disso, a equipe aplicou o método GM no protótipo navegável de modo a investigar se o *software* fornecia suporte a diferentes estilos cognitivos de uso. Este ocorrido também será relatado posteriormente.

3.2 Inspiração

Coleta de dados: Para a coleta foram utilizadas as técnicas Mapa de *Stakeholders* e a Matriz de Motivação. O Mapa de *Stakeholders* [14] é uma técnica que permite identificar quem são as pessoas-chave que podem ter uma participação nos resultados do projeto. Ele é um ponto de referência visual para a equipe no planejamento das atividades de pesquisa com os usuários e promove a comunicação adequada com os *stakeholders* ao longo do processo de desenvolvimento [14]. Para construí-lo, a equipe listou primeiro quem são os possíveis *stakeholders* e suas necessidades. Em seguida discutiu-se quais relações eles tinham entre si, para visualizar se os *stakeholders* descritos faziam sentido para a proposta do *software*. A equipe também consultou potenciais usuários, e descobriu a existência de mais um *stakeholder*.

A Matriz de Motivação [24] visa identificar as motivações dos *stakeholders* envolvidos no sistema. Ela representa a solução do serviço do ponto de vista da motivação individual de cada *stakeholder* relacionada uns aos outros. Ao cruzar

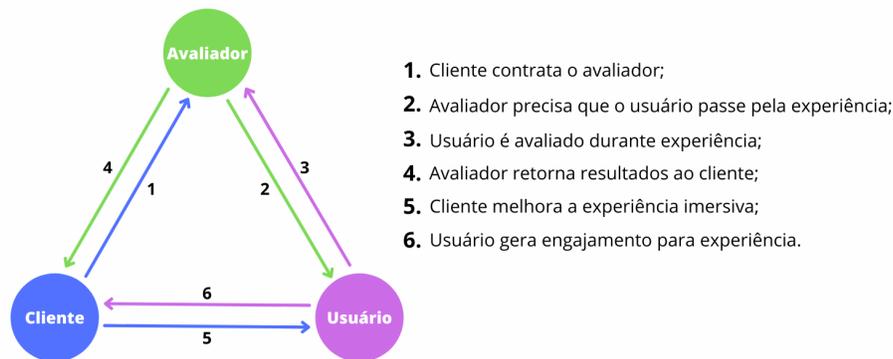


Figura 1. Mapa de *Stakeholders*.

suas várias motivações, é possível entender os motivos, as contribuições potenciais e os benefícios esperados do sistema [24]. Tanto o Mapa de *Stakeholders* quanto a Matriz de Motivação foram construídos após consultas feitas com os potenciais usuários para esta etapa inicial.

Síntese de dados: Ocorre após a coleta de dados em que é necessário sintetizar e analisar esses dados para obter informações e ideias para aplicar. As técnicas utilizadas foram Mapa conceitual e Personas.

O Mapa conceitual [14] é um diagrama que indica a relação entre os conceitos ou palavras que representam esses conceitos. Dessa forma, a equipe construiu um mapa que ajudasse a compreender as relações entre os conceitos utilizados pelo *Framework* para o *software*. Além de poder compreender as relações é possível compreender o usuário. A técnica usada que foca nessa questão é a técnica Personas. Personas [25] é uma técnica na qual personagens fictícios são criados como uma representação do cliente ideal. A partir dos dados coletados foram criadas duas personas pela equipe de desenvolvimento. As personas foram revisadas por um especialista em DT e podem ser consultadas com detalhes no Relatório Técnico (RT)[2].

3.3 Ideação

Após a fase de descobrimento e pesquisa sobre o usuário, a fase de ideação se propõe a discutir problemas e ideias de como resolvê-los. Para isso, foi utilizado a técnica *Brainstorming* [25], pois nessa fase é preciso gerar ideias para criar soluções. O *Brainstorming* é uma técnica, feita a partir de uma dinâmica em grupo, para estimular a geração de um grande número de ideias em um curto espaço de tempo. Essa técnica foi escolhida com o objetivo de levantar as necessidades e desejos dos usuários delimitados na fase anterior, assim definindo quais requisitos seriam necessários para a construção do *software*.

Primeiramente, a equipe de desenvolvimento discutiu quais problemas o avaliador poderia ter ao utilizar o *Framework*, visto que os membros da equipe também já haviam sido avaliadores e utilizado o *Framework* para realizar uma avaliação de UX [3]. Sendo assim, foram agrupadas ideias de como resolver esses problemas, gerando uma lista de requisitos para o *software*. Essa lista foi analisada com os artefatos resultantes da fase de inspiração, visando confirmar se os requisitos atendem as necessidades e desejos dos *Stakeholders*. Esses artefatos podem ser consultados no RT [2] detalhadamente.

Em sequência, a lista de requisitos foi encaminhada a um especialista em ER para ser avaliada. O resultado dessa avaliação foi a lista de requisitos encontrada na Tabela 2, contendo os requisitos que foram considerados para a construção do protótipo, na fase de implementação.

3.4 Implementação

A implementação trata do desenvolvimento de design do projeto. Então, com base nos dados obtidos das fases de inspiração 3.2 e ideação 3.3, foi produzido um protótipo navegável do *software*. Esse protótipo contém as três etapas do *Framework* citado na Seção 3.1. Através das telas iniciais mostradas na Figura 2, o avaliador pode adicionar ou criar uma nova avaliação. Ao clicar numa das avaliações, ele pode navegar entre as etapas de Expectativa, Desempenho e Satisfação.



Figura 2. Telas Iniciais do Protótipo.

Na tela de Expectativa, o avaliador pode criar e editar um questionário conforme o entretenimento que ele pretende avaliar, além de visualizar os gráficos das respostas. Essas mesmas funcionalidades também estão na tela de Satisfação, com a diferença de que o questionário não é editável, como proposto pelo *Framework Immersive UX*. Na tela de Desempenho é sugerida uma ou mais técnicas

de observação que podem ser aplicadas para esta etapa do *Framework*, auxiliando o avaliador na escolha da técnica adequada para a avaliação.

3.5 GenderMag

A avaliação foi aplicada individualmente por cada integrante da equipe de desenvolvimento. As personas escolhidas foram Abi e Tim, que representam um extremo do outro (feminino e masculino) para verificar se o protótipo era inclusivo para ambas as facetas. Foi definido um objetivo a ser atingido pelas personas e as ações a serem realizadas no passo a passo cognitivo. O cenário ficou da seguinte forma:

Objetivo Geral: Persona quer criar um questionário de expectativa.

1. Subobjetivo: Adicionar pergunta.
 - I Clicar em “editar questionário”.
 - II Clicar em “adicionar nova pergunta”.
 - III Preencher o campo de pergunta.
 - IV Dizer o tipo da pergunta e, caso necessário, preencher os demais campos.
2. Subobjetivo: Gerar link de compartilhamento.
 - I Clicar em “gerar link compartilhável”.

4 Resultados

Nesta seção serão mostrados os resultados das fases do *Design Thinking* (DT) e da aplicação do *GenderMag* (GM). Os resultados do DT (Subseção 4.1) mostram, principalmente, os requisitos obtidos para o *software* referente ao *Framework Immersive UX*. Já na Subseção 4.2 são apresentados os problemas encontrados através do GM, relacionados à inclusão de gênero considerando cada persona.

4.1 Design Thinking

Os resultados das fases de inspiração e ideação foram utilizados para o levantamento de requisitos. Na fase de inspiração foi construído o “Mapa de *Stakeholders*” onde foram identificados os seguintes *stakeholders*: o avaliador, que é a pessoa que realiza a avaliação de UX do entretenimento imersivo; o usuário, que é a pessoa que participa do entretenimento imersivo; e o cliente, sendo o dono do entretenimento imersivo. Com a “Matriz de Motivação”, foi obtido uma tabela em que é possível observar a motivação individual de cada *stakeholder*, por exemplo, o avaliador espera que o usuário passe pela experiência e retorne um *feedback*, e o usuário espera do avaliador que a avaliação não seja invasiva ou incômoda.

Na fase de ideação, foi possível listar os principais problemas encontrados pelo *stakeholder* avaliador, levando a equipe a pensar em soluções tecnológicas viáveis. Dessa forma, os requisitos para o *software* foram levantados e construiu-se um protótipo navegável. A Tabela 2 lista os requisitos levantados.

Tabela 2. Requisitos levantados na fase de ideação.

	Requisitos do Software
1	Ao entrar no <i>software</i> o usuário tem a opção de “adicionar avaliações” ou ver “avaliações criadas”.
2	O usuário é capaz de criar e editar um questionário para a etapa de expectativa.
3	Ao editar um questionário, o usuário é capaz de modificar o tipo de pergunta (fechada ou aberta).
4	Ao concluir a criação do questionário, o usuário é capaz de gerar um link de compartilhamento para o questionário de expectativa e para o questionário de satisfação.
5	Dado que o questionário recebe respostas, o usuário é capaz de visualizar às respostas desse questionário.
6	O usuário é capaz de visualizar os gráficos gerados pelo <i>software</i> referentes às respostas recebidas pelo questionário de expectativa.
7	O usuário é capaz de visualizar os gráficos gerados pelo <i>software</i> referentes às métricas de <i>flow</i> , <i>presence</i> e <i>engagement</i> , resultantes das respostas do questionário de satisfação.
8	O usuário é capaz de visualizar uma nuvem de palavras gerada pelas palavras que mais se repetiram no questionário de expectativa.

4.2 GenderMag

Os resultados da aplicação do *GenderMag* foram obtidos através da avaliação do protótipo com as personas Abi e Tim. A persona Tim não encontrou problemas ao navegar pelo protótipo, já a persona Abi encontrou os seguintes problemas:

- O botão “editar questionário” poderia se chamar “criar questionário”, pois condiz melhor com a ação do botão.
- O botão “adicionar pergunta” deve vir antes do botão “editar pergunta”. Essa forma é mais intuitiva.
- Ao criar uma pergunta para o questionário de expectativa, Abi não sabe se realizou esta ação corretamente. Logo, o *software* precisa mostrar algo que indique que a ação feita foi concluída com êxito.
- O botão “gerar link” poderia ser ativado ao preencher ao menos uma questão.

Após listar os problemas encontrados por Abi, foram contabilizados quantos “Sim”, “Talvez” e “Não” que Abi e Tim responderam para cada atividade do passo a passo cognitivo (Tabela 3). No método GM, as respostas “Sim” são consideradas boas, pois indicam que a persona realizou um percurso bem-sucedido, percorrendo os passos esperados pelo *software*. As respostas “Talvez” indicam que a persona pode realizar o percurso bem-sucedido ou não, dependendo de suas facetas. Já as respostas “Não”, indicam que a persona falhou ao realizar o percurso, indicando uma limitação do *software* ao dar suporte as características da persona.

As respostas foram contabilizadas com base nas 4 avaliações feitas pela equipe de desenvolvimento, como citado na Seção 3.5. Cada coluna da Tabela 3 demonstra

quantas vezes cada faceta se repetiu para cada resposta dada por Abi e Tim. A última coluna representa a quantidade total de “Sim”, “Talvez” e “Não” contabilizados pela avaliação da equipe para cada atividade do passo a passo cognitivo, sem considerar as repetições de cada faceta.

Tabela 3. Contabilidade das Respostas das personas Abi e Tim.

		Motivações	Estilos de Processamento de Informações	Estilo de aprendizagem de tecnologia	Atitudes em relação ao risco	Autoeficácia computacional	Total
Abi	Sim	21	8	2	5	5	28
	Talvez	8	2	4	2	4	11
	Não	3	3	4	3	1	7
Tim	Sim	14	28	17	10	6	44
	Talvez	1	1	1	0	0	3
	Não	1	0	0	0	0	1

5 Discussão

O objetivo principal do *software*, conforme os dados levantados, é dar suporte ao avaliador no uso do *Framework Immersive UX*. Mas, segundo os resultados do GM, foi identificado viés de gênero no protótipo inicial (Subseção 4.2). Esse viés pode ter sido influenciado pela própria equipe de desenvolvimento. Considerando que a equipe já havia atuado como avaliador utilizando o *Framework*, os requisitos foram implementados no protótipo da maneira que a equipe achou mais adequado, sem considerar questões de gênero. Isso é um indício de que, caso o propósito seja a construção de um *software* mais inclusivo, a equipe de desenvolvimento precisa estar atenta a essa questão durante a fase de implementação, para que os requisitos levantados sejam implementados de forma inclusiva.

Os resultados do GM mostraram que a persona Abi encontrou dificuldades em navegar no protótipo, onde obteve 7 respostas “Não”, enquanto Tim obteve apenas 1 (referentes a coluna “Total” da Tabela 3). Isso demonstra uma carência no apoio do *software* para o público com as características de Abi, o que o *GenderMag* considera um “*bug*” de gênero na aplicação. Porém, o ciclo de DT pode ser visto como uma vantagem nesse sentido, já que não é uma abordagem linear (como descrito por Brown [5]). Caberia nessa situação realizar novos ciclos e fazer os refinamentos necessários, investigando de que modo o *software* poderia considerar melhor as características da persona Abi.

Porém, Abi apresenta uma alta motivação, sendo essa a sua faceta com mais repetições para as respostas “Sim”. Esse dado pode indicar que sua motivação faz com que ela supere suas dificuldades, pois entre suas características é citado que ela pode aprender uma ferramenta nova se for necessário. Apesar disso, é preciso avaliar as demais facetas, porque se o *software* se demonstra difícil de usar

para pessoas com características da Abi, a motivação não se torna o suficiente para o usuário permanecer utilizando o *software*.

As demais facetas que apontam “*bugs*” de gênero, comparando as características de Abi e Tim, são o “Estilo de Aprendizagem de tecnologia” e “Estilo de Processamento de Informações”, podendo ser observadas na Tabela 3. Em relação à faceta “Estilo de aprendizagem”, Abi se encontra em desvantagem quanto a Tim. Isso demonstra a falta de suporte da aplicação quanto ao estilo de aprendizagem para pessoas com estilos cognitivos da Abi. Em relação ao estilo de processamento de informação, como mostrado na Tabela 1, cada estilo tem suas vantagens, mas ambos ficam em desvantagem quando não são suportados pelo *software*.

No contexto de engenharia de *software*, o DT se torna vantajoso para a elicitación de requisitos, visto que é uma abordagem centrada no ser humano e que busca entender as necessidades e desejos do usuário, podendo contribuir, assim, para a qualidade do *software*. Durante a construção do *software*, caso o objetivo também seja torná-lo mais inclusivo em questões de gênero, o uso do método GM no ciclo de DT pode ser vantajoso. Dado que o DT permite realizar novos ciclos, ao detectar bugs de gênero com o GM, é possível realizar melhorias no *software* ainda em desenvolvimento.

6 Conclusões e Trabalhos Futuros

Neste artigo, foi descrito o processo de levantamento de requisitos para a construção de um *software* visando auxiliar um avaliador no uso do *Framework Immersive UX*, citado na Subseção 3.1. Para isso, foi utilizado o ciclo de DT proposto por Tim Brown (inspiração, ideação e prototipação). Com o protótipo feito, foi aplicado o método GM de modo a investigar o quão inclusivo (relacionado aos aspectos cognitivos) a aplicação se mostra. Do modo que a equipe de desenvolvimento implementou os requisitos no protótipo, observou-se que o protótipo dava mais suporte aos aspectos cognitivos de Tim, indicando um viés de gênero.

O uso do GM serviu para resolver essa questão. A equipe de desenvolvimento pôde enxergar aspectos na aplicação durante seu uso em que não incluíam as características cognitivas da Abi, possibilitando que a equipe ampliasse a inclusão de gênero no *software*. Este artigo mostra indícios que o uso conjunto do DT com o GM é vantajoso para o desenvolvimento de um *software* mais inclusivo, pois se torna possível considerar questões de gênero durante o desenvolvimento do *software*.

Cabe considerar que o artigo apresenta indícios limitados ao contexto do *Framework Immersive UX*, mas pode ser vantajoso para outros contextos. Também é necessário realizar pesquisas futuras para investigar possíveis soluções para evitar “*bugs*” de gênero mais cedo, por exemplo, analisar se a construção de personas se adéquam às facetas do GM.

Agradecimentos

Esta pesquisa, realizada no âmbito do Projeto Samsung-UFAM de Ensino e Pesquisa (SUPER), nos termos do artigo 48 do Decreto nº 6.008/2006 (SUFRAMA), foi financiada pela Samsung Eletrônica da Amazônia Ltda., Nos termos do Lei Federal nº 8.387/1991, por meio do convênio 001/2020, firmado com a Universidade Federal do Amazonas e a FAEPI, Brasil. Também apoiado pela CAPES - Código Financiamento 001, processo CNPq 314174/2020-6, e processo FAPEAM 062.00150/2020 e POSGRAD.

Referências

1. Adikari, S., McDonald, C., Campbell, J.: Reframed contexts: Design thinking for agile user experience design. pp. 3–12 (07 2013)
2. Almeida, E., Monteiro, V., Alves, F., Aguiar, B., Marques, L., Gadelha, B., Conte, T.: Material de apoio para thinking about gender: Combinando design thinking e gendermag na elicitação de requisitos para um software de apoio a avaliação de ux (2021). <https://doi.org/10.6084/m9.figshare.14331281>
3. Alves, F., Aguiar, B., Monteiro, V., Almeida, E., Marques, L., Gadelha, B., Conte, T.: Immersive ux: a ux evaluation framework for digital immersive experiences in the context of entertainment. *Proceedings of the 23rd International Conference on Enterprise Information Systems* **17**(3) (2021)
4. Beckwith, L., Kissinger, C., Burnett, M., Wiedenbeck, S., Lawrance, J., Blackwell, A., Cook, C.: Tinkering and gender in end-user programmers' debugging. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. pp. 231–240 (2006)
5. Brown, T.: *Design Thinking: Uma metodologia poderosa para decretar o fim das velhas ideias*. Alta Books (2020), <https://books.google.com.br/books?id=AKb5DwAAQBAJ>
6. Burnett, M., Fleming, S.D., Iqbal, S., Venolia, G., Rajaram, V., Farooq, U., Grigoreanu, V., Czerwinski, M.: Gender differences and programming environments: Across programming populations. In: *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*. pp. 1–10 (2010)
7. Burnett, M., Stumpf, S., Macbeth, J., Makri, S., Beckwith, L., Kwan, I., Peters, A., Jernigan, W.: GenderMag: A Method for Evaluating Software's Gender Inclusiveness. *Interacting with Computers* **28**(6), 760–787 (10 2016). <https://doi.org/10.1093/iwc/iwv046>, <https://doi.org/10.1093/iwc/iwv046>
8. Burnett, M.M., Beckwith, L., Wiedenbeck, S., Fleming, S.D., Cao, J., Park, T.H., Grigoreanu, V., Rector, K.: Gender pluralism in problem-solving software. *Interacting with computers* **23**(5), 450–460 (2011)
9. de Castro Pinto, F.A., Siqueira, F.L.: Problemas do design thinking para a engenharia de requisitos: uma revisão sistemática da literatura. In: *WER20 - Workshop em Engenharia de Requisitos* (2020)
10. Charness, G., Gneezy, U.: Strong evidence for gender differences in risk taking. *Journal of Economic Behavior & Organization* **83**(1), 50–58 (2012)
11. Curcio, K., Navarro, T., Malucelli, A., Reinehr, S.: Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software* **139**, 32–50 (2018)

12. Dohmen, T., Falk, A., Huffman, D., Sunde, U., Schupp, J., Wagner, G.G.: Individual risk attitudes: Measurement, determinants, and behavioral consequences. *Journal of the European Economic Association* **9**(3), 522–550 (2011)
13. Dunne, D., Martin, R.: Design thinking and how it will change management education: An interview and discussion. *Academy of Management Learning and Education* **5**, 512–523 (2006)
14. Hanington, B., Martin, B.: *Universal methods of design: 100 ways to research complex problems*. Rockport (2012)
15. Hehn, J., Uebersnickel, F.: The use of design thinking for requirements engineering: An ongoing case study in the field of innovative software-intensive systems. In: 2018 IEEE 26th International Requirements Engineering Conference (RE). pp. 400–405 (2018). <https://doi.org/10.1109/RE.2018.00-18>
16. Mahé, N., Adams, B., Marsan, J., Templier, M., Bissonnette, S.: Migrating a software factory to design thinking: Paying attention to people and mind-sets. *IEEE Software* **37**(2), 32–40 (2019)
17. Meyers-Levy, J., Loken, B.: Revisiting gender differences: What we know and what lies ahead. *Journal of Consumer Psychology* **25**(1), 129–149 (2015)
18. O'Donnell, E., Johnson, E.: Gender effects on processing effort during analytical procedures. *Int. J. Auditing* **5**, 91–105 (2001)
19. de Paula, D.F., Menezes, B.H., Araújo, C.C.: Building a quality mobile application: A user-centered study focusing on design thinking, user experience and usability. In: *International Conference of Design, User Experience, and Usability*. pp. 313–322. Springer (2014)
20. Schön, E.M., Thomaschewski, J., Escalona, M.J.: Agile requirements engineering: A systematic literature review. *Computer Standards & Interfaces* **49**, 79–91 (2017)
21. Spencer, R.: The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. pp. 353–359 (2000)
22. Torrezan, C.A.W., Sparremberger, A.S., da Silva, K.K.A., Behar, P.A.A., Félix, J.A.L., de Carli, M.F.: Sistema de recomendação educacional para mobile: um foco na experiência do usuário. *RENTE-Revista Novas Tecnologias na Educação* **17**(3) (2019)
23. Vetterli, C., Brenner, W., Uebersnickel, F., Petrie, C.: From palaces to yurts - why requirements engineering needs design thinking. *IEEE Internet Computing* **17**(2), 91–94 (2013)
24. Vezzoli, C.A., Delfino, E., Lorraine, A.: *System design for sustainable energy for all. a new challenging role for design to foster sustainable development* (2014)
25. Vianna, M.: *Design thinking: inovação em negócios*. MJV Press (2012)
26. Vorvoreanu, M., Zhang, L., Huang, Y.H., Hilderbrand, C., Steine-Hanson, Z., Burnett, M.: From gender biases to gender-inclusive design: An empirical investigation. p. 1–14. CHI '19, Association for Computing Machinery (2019)
27. Wazlawick, R.: *Engenharia de software: Conceitos e práticas*. Elsevier Editora Ltda. (2019), <https://books.google.com.br/books?id=d1qnDwAAQBAJ>
28. Weigel, L.: *Design Thinking to Bridge Research and Concept Design*, pp. 59–70 (2015). <https://doi.org/10.1002/9781119154273.ch5>