



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

# **Kick-start Kanban e Design Centrado no Usuário: Processo de Software de uma Startup**

**Autores: Ailton Aires Amado e Kevin Luis Apolinario Batista**  
**Orientadora: Prof<sup>ª</sup>. Dra. Rejane Maria da Costa Figueiredo**

Brasília, DF  
2024



Ailton Aires Amado e Kevin Luis Apolinario Batista

# **Kick-start Kanban e Design Centrado no Usuário: Processo de Software de uma Startup**

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

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof<sup>ª</sup>. Dra. Rejane Maria da Costa Figueiredo

Brasília, DF

2024

---

Ailton Aires Amado e Kevin Luis Apolinario Batista

Kick-start Kanban e Design Centrado no Usuário: Processo de Software de uma Startup/ Ailton Aires Amado e Kevin Luis Apolinario Batista. – Brasília, DF, 2024- 139 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof<sup>ª</sup>. Dra. Rejane Maria da Costa Figueiredo

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB

Faculdade UnB Gama – FGA , 2024.

1. Metodologia Ágil. 2. Design Centrado no Usuário. 3. *Workshop Kick-start Kanban*. 4. Melhoria de Processo de Software. I. Prof<sup>ª</sup>. Dra. Rejane Maria da Costa Figueiredo. II. III. Universidade de Brasília. IV. Faculdade UnB Gama. V. Kick-start Kanban e Design Centrado no Usuário: Processo de Software de uma Startup

CDU

---

Ailton Aires Amado e Kevin Luis Apolinario Batista

## **Kick-start Kanban e Design Centrado no Usuário: Processo de Software de uma Startup**

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

Brasília, DF

2024

# Agradecimentos

Eu, Kevin, gostaria de iniciar agradecendo aos meus pais, pelo apoio incondicional, pela educação e por sempre acreditarem no meu potencial e no meu sonho de me tornar engenheiro. Suas palavras de encorajamento foram fundamentais para que eu chegasse até aqui. Gostaria de agradecer também a minha companheira, Ana Luiza, por todo o apoio, principalmente na reta final que foi o momento mais difícil.

Agradeço profundamente aos meus amigos e colegas de curso, Victor Samuel, André Lucas, Thiago, João Pedro, Victor Buendia e Ailton, por toda a amizade, pelas discussões enriquecedoras e por tornarem essa caminhada mais leve e divertida. Em especial, ao Ailton, que além de parceiro de trabalho, foi um amigo fiel, sempre ao meu lado nos momentos mais difíceis, oferecendo apoio emocional e intelectual. Sem o apoio e a parceria de todos vocês, essa jornada teria sido muito mais desafiadora.

Agradeço à Universidade de Brasília e a empresa Atena Solutions, que me acolheram e me deram a oportunidade única de aliar o conhecimento teórico a aplicação de soluções práticas e que resolvem problemas do dia a dia.

Por fim deixo meus agradecimentos à minha orientadora, Prof<sup>a</sup>. Rejane Maria da Costa Figueiredo, pela paciência, orientação e pelo compartilhamento de conhecimento que foram essenciais para o desenvolvimento deste trabalho. E estendo o agradecimento também aos membros da banca, Prof<sup>a</sup> Elaine Venson, Prof<sup>o</sup> Marcelo Judice e Thiago Luiz, por suas valiosas contribuições e sugestões que enriqueceram este trabalho.

# Agradecimentos

Eu, Ailton Aires Amado, agradeço à minha família, em especial ao meu pai, Antônio, grande responsável pelo meu brilho, sucesso e permanência até este momento. Dedico toda a dor, gota de suor, esforço e resultados a ele, pois foi a base incondicional para que eu chegasse onde cheguei.

Agradeço também aos meus amigos e companheiros de vida, os quais não consigo listar todos aqui, pois a lista seria interminável. Ter essas pessoas e caminhar ao lado delas me trouxe forças e determinação para chegar onde cheguei. Ao meu saudoso irmão, Alex Boaventura Amado, um homem brilhante e de inteligência incomparável, deixo meu eterno reconhecimento. Ao meu irmão Rafael, onde a sua simples existência me trouxe forças para continuar trilhando a vida. Agradeço a Lucas V. O. S., um homem cheio de bondade e que, com o peito estufado, sempre acreditou e me ajudou nesse processo. Thiago A. Marinho, com um grande coração e sempre acreditando em mim, fez com que eu caminhasse de cabeça erguida. Kevin, um grande companheiro durante toda a escrita desta monografia, foi um pilar fundamental para a qualidade deste trabalho e durante toda a minha trajetória na faculdade. Aproveito também para agradecer a Amanda Lins, uma das pessoas com o coração mais belo e puro que conheço, de uma bondade imensurável e que me inspira muito. Agradeço também a Eduarda de C. N. Stracquadanio, por quem nutro um sentimento imensurável. À professora Isabela, grande apoiadora e incentivadora nos meus primeiros passos de responsabilidade. Aos demais, André Lucas, Victor Samuel, Victor Buendia, Diego Paiva, Juliana Bonfim, Magson W., Janderson G., Luiza F., e a todos os outros, agradeço pela amizade, paciência e compreensão até o presente momento. Espero caminhar ao lado de todos durante toda a minha vida.

Diante dos meus princípios sociais, me curvo à Universidade de Brasília em um gesto humilde de agradecimento e carinho, pois tenho certeza de que, em nenhum outro lugar, conheceria pessoas tão incríveis e também não teria um futuro tão brilhante traçado. Agradeço também à Startup Atena Solutions e seus gestores, pois, sem eles, esta monografia tomaria outros rumos. Eles também me concederam a primeira oportunidade de crescimento profissional.

Agradeço à grande orientadora Prof<sup>a</sup>. Dra. Rejane Maria da Costa Figueiredo, por acreditar e sempre incentivar os caminhos escolhidos durante a escrita deste trabalho, além de contribuir para a construção e sucesso desta monografia. Agradeço também a Deus, à banca e a todos os envolvidos, até mesmo àqueles que tangenciaram quaisquer aspectos positivos da minha vida.

Por fim, deixo uma mensagem autoral que me acompanhou durante todos os anos da faculdade e vai me acompanhar durante toda a vida. A todos, "Que os bons momentos prevaleçam, mas sem esquecer que os ruins poderão existir". Obrigado!



# Resumo

O Desenvolvimento Ágil tem sido empregado no desenvolvimento de *software* por *startups* que buscam eficiência e produtos de excelência para os usuários. Dada a agilidade e o foco nos resultados, muitas abordagens podem não comportar a criação e o desenvolvimento a partir da experiência dos usuários finais, ainda que essa possa agregar valor ao produto final. O *Design* Centrado no Usuário (UCD) visa entender as necessidades, desejos e limitações dos usuários, utilizando *feedback*, observação e iteração contínua, e pode aprimorar o Método Ágil ao proporcionar uma análise mais sistemática das necessidades dos usuários. Por outro lado, o Método Ágil pode beneficiar o UCD, permitindo iterações e entregas mais frequentes, o que leva a avaliações de usabilidade mais regulares e *feedbacks* contínuos. O *workshop* ágil *Kick-start Kanban*, que facilita a rápida implementação do Kanban, se destaca como uma ferramenta versátil que promove a colaboração, mentalidade ágil e que agrega valor a equipe, contribuindo para o sucesso do produto e aceitação pelos usuários finais. Este trabalho propôs a inserção da abordagem UCD no *Workshop Kick-start Kanban* para aprimorar o processo de desenvolvimento de *software* da *startup* Atena Solutions, focando no projeto Groovoo, que enfrenta desafios no processo de desenvolvimento e na satisfação dos usuários. A pesquisa descritiva utilizou a técnica de estudo de caso, analisando e adaptando o *workshop* para incluir técnicas UCD, avaliando a melhoria no processo. Como resultado, houve uma adaptação no Processo de Desenvolvimento do projeto Groovoo, com nova organização do fluxo de trabalho, políticas de trabalho definidas, técnicas UCD aplicadas e maior organização e satisfação da equipe. Conclui-se que a realização do *workshop* e a integração do Kanban com as abordagens UCD permitiram à *startup* Atena Solutions avançar em direção a um desenvolvimento mais centrado no usuário, além de auxiliar na rastreabilidade do trabalho, organização e comunicação da equipe. Futuramente, o *workshop* pode ser ajustado para atender às especificidades de cada *startup* no momento da próxima implementação, incorporando novas técnicas UCD conforme necessário.

**Palavras-chave:** Desenvolvimento Ágil de Software; Design Centrado no Usuário; UCD; Kanban; Kick-start Kanban;

# Abstract

Agile Development has been employed in software development by startups seeking efficiency and excellence in products for users. Given the agility and focus on results, many approaches may not accommodate the creation and development based on the final users' experience, even though this could add value to the final product. User-Centered Design (UCD) aims to understand the users' needs, desires, and limitations, using feedback, observation, and continuous iteration, and it can enhance the Agile Method by providing a more systematic analysis of user needs. On the other hand, the Agile Method can benefit UCD by allowing for more frequent iterations and deliveries, leading to regular usability assessments and continuous feedback. The Kick-start Kanban agile workshop, which facilitates the rapid implementation of Kanban, stands out as a versatile tool that promotes collaboration, an agile mindset, and adds value to the team, contributing to the product's success and user acceptance. This work proposed the inclusion of the UCD approach in the Kick-start Kanban Workshop to improve the software development process of the startup Atena Solutions, focusing on the Groovoo project, which faces challenges in the development process and user satisfaction. The descriptive research employed the case study technique, analyzing and adapting the workshop to include UCD techniques, and evaluating the improvement in the process. As a result, there was an adaptation in the Groovoo project's development process, with a new organization of the workflow, defined work policies, applied UCD techniques, and greater team organization and satisfaction. It is concluded that the workshop's execution and the integration of Kanban with UCD approaches allowed Atena Solutions to advance towards a more user-centered development, in addition to helping with work traceability, team organization, and communication. In the future, the workshop can be adjusted to meet the specific needs of each startup at the time of the next implementation, incorporating new UCD techniques as necessary.

**Keywords:** Agile Software Development; User Centered Desing; UCD; Kanban; Kick-start Kanban.

# Lista de ilustrações

Figura 1 – Exemplo de um quadro <i>Scrumban</i> (BHAVSAR; GOPALAN; SHAH, 2020) . . . . .	26
Figura 2 – Exemplo de um quadro <i>Scrumban</i> (BHAVSAR; GOPALAN; SHAH, 2020) . . . . .	27
Figura 3 – Ciclo de desafios das equipes (ACHOUIANTZ; NORDIN, 2013) . . . . .	31
Figura 4 – Processo para implementação do <i>Kick-start Kanban</i> (ACHOUIANTZ; NORDIN, 2013) . . . . .	33
Figura 5 – Fase 1 do <i>Kick-start Kanban</i> (ACHOUIANTZ; NORDIN, 2013) . . . . .	35
Figura 6 – Fase 2 do <i>Kick-start Kanban</i> (ACHOUIANTZ; NORDIN, 2013) . . . . .	36
Figura 7 – Fase 3 do <i>Kick-start Kanban</i> (ACHOUIANTZ; NORDIN, 2013) . . . . .	38
Figura 8 – Fase 4 do <i>Kick-start Kanban</i> (ACHOUIANTZ; NORDIN, 2013) . . . . .	42
Figura 9 – Etapas do UCD (NORMAN, 2013) . . . . .	48
Figura 10 – Quadro Kanban para design, desenvolvimento e operação (SCHÖN et al., 2016) . . . . .	54
Figura 11 – Cronograma do projeto de pesquisa . . . . .	60
Figura 12 – Processo de desenvolvimento de <i>software</i> do Projeto Groovoo (GOMES; LUCAS, 2023) . . . . .	66
Figura 13 – <i>Scrumban-UCD</i> do Projeto Groovoo (GOMES; LUCAS, 2023) . . . . .	67
Figura 14 – Quadro Kanban do Projeto Groovoo . . . . .	90
Figura 15 – Quadro Kanban do Projeto Groovoo . . . . .	91
Figura 16 – Quadro Kanban do Projeto Groovoo . . . . .	93
Figura 17 – Resposta do Questionário . . . . .	97
Figura 18 – Resposta do Questionário . . . . .	97
Figura 19 – Resposta do Questionário . . . . .	98
Figura 20 – Resposta do Questionário . . . . .	98
Figura 21 – Resposta do Questionário . . . . .	98
Figura 22 – Resposta do Questionário . . . . .	99
Figura 23 – Resposta do Questionário . . . . .	99
Figura 24 – Resposta do Questionário . . . . .	99
Figura 25 – Resposta do Questionário . . . . .	100
Figura 26 – Resposta do Questionário . . . . .	100
Figura 27 – Resposta do Questionário . . . . .	100
Figura 28 – Resposta do Questionário . . . . .	101
Figura 29 – Resposta do Questionário . . . . .	101

# Lista de tabelas

Tabela 1 – Valores do Manifesto Ágil - (BECK et al., 2001) . . . . .	24
Tabela 2 – Aspectos chave do <i>Kick-start Kanban</i> . . . . .	32
Tabela 3 – Fase 1 - Atividades do <i>Kick-start Kanban</i> (ACHOUANTZ; NORDIN, 2013) . . . . .	36
Tabela 4 – Fase 2 - Atividades do <i>Kick-start Kanban</i> . . . . .	37
Tabela 5 – Fase 3 - Atividades do <i>Kick-start Kanban</i> (ACHOUANTZ; NORDIN, 2013) . . . . .	39
Tabela 6 – Fase 4 - Atividades do <i>Kick-start Kanban</i> . . . . .	43
Tabela 7 – Técnicas UCD relatadas na revisão sistemática - Adaptado (SALINAS; CUEVA; PAZ, 2020) . . . . .	49
Tabela 8 – Análise comparativa entre as técnicas UCD I (ARGUMANIS; MOQUILLAZA; PAZ, 2021) - Adaptado . . . . .	51
Tabela 9 – Análise comparativa entre as técnicas UCD II (ARGUMANIS; MOQUILLAZA; PAZ, 2021) . . . . .	52
Tabela 10 – Análise comparativa entre as técnicas UCD III (ARGUMANIS; MOQUILLAZA; PAZ, 2021) . . . . .	52
Tabela 11 – Protocolo de Estudo de Caso (BRERETON et al., 2008) . . . . .	57
Tabela 12 – Análise de técnicas UCD com base nos critérios dos gestores . . . . .	72
Tabela 13 – Etapa 1 - Atividades adicionadas ao workshop. . . . .	75
Tabela 14 – Etapa 2 - Atividades adicionadas ao workshop. . . . .	75
Tabela 15 – Etapa 3 - Atividades adicionadas ao workshop. . . . .	76
Tabela 16 – Fase 1 - Atividades do <i>Kick-start Kanban - Adaptado</i> . . . . .	77
Tabela 17 – Fase 2 - Atividades do <i>Kick-start Kanban - Adaptado</i> . . . . .	78
Tabela 18 – Fase 3 - Atividades do <i>Kick-start Kanban - Adaptado</i> . . . . .	79
Tabela 19 – Fase 4 - Atividades do <i>Kick-start Kanban - Adaptado</i> . . . . .	82

# Lista de abreviaturas e siglas

UnB	Universidade de Brasília
UCD	User-Centered Design
UX	User Experience
ROI	Return Over Investment
WIP	Work-in-Progress
CJM	Customer Journey Map
PO	Product Owner
ASD	Agile Software Development

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
1.1	Considerações Iniciais do Capítulo	17
1.2	Contexto	17
1.3	Problema	19
1.4	Objetivos	20
1.4.1	Objetivo Geral	20
1.4.2	Objetivos Específicos	20
1.5	Metodologia	20
1.5.1	Classificação Metodológica	20
1.6	Organização do Trabalho	21
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>23</b>
2.1	Considerações Iniciais do Capítulo	23
2.2	Metodologias Ágeis	23
2.2.1	Scrumban	26
2.2.1.1	Princípios e Técnicas	26
2.3	Kick-start Kanban	29
2.3.1	História do <i>Kick-start Kanban</i>	30
2.3.2	Aspectos chave do <i>Kick-start Kanban</i>	31
2.3.3	Processo inicial do <i>Kick-start Kanban</i>	32
2.3.4	Aplicação do <i>Kick-start Kanban</i>	34
2.3.4.1	Compreendendo a Equipe	35
2.3.4.2	Planejamento do <i>Kick-start Kanban</i>	36
2.3.4.3	Execução do <i>Kick-start Kanban</i>	37
2.3.4.4	Atingir a linha de base da melhoria contínua	41
2.4	<i>Startups</i> e o emprego de Metodologias Ágeis	44
2.5	Design Centrado no Usuário	44
2.5.1	Atividades do UCD	46
2.5.2	Técnicas UCD	48
2.5.3	Análise comparativa entre as técnicas UCD	50
2.5.4	Design Centrado no Usuário e <i>Kanban</i>	52
2.6	Considerações Finais do Capítulo	55
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>56</b>
3.1	Considerações Iniciais do Capítulo	56
3.1.1	Plano Metodológico	56

<b>3.2</b>	<b>Planejamento da Pesquisa</b>	<b>56</b>
3.2.1	<i>Background</i>	57
3.2.2	<i>Design</i>	57
3.2.3	Seleção de Caso	58
3.2.4	Procedimentos e Papéis do estudo de caso	58
3.2.5	Coleta de Dados	58
3.2.6	Análise	59
3.2.7	Validade do plano	59
3.2.8	Limitações do Estudo	59
3.2.9	Relatórios	59
3.2.10	Cronograma	59
3.2.11	Apêndices	60
<b>3.3</b>	<b>Considerações Finais do Capítulo</b>	<b>61</b>
<b>4</b>	<b>ATENA SOLUTIONS</b>	<b>62</b>
<b>4.1</b>	<b>Considerações Iniciais do Capítulo</b>	<b>62</b>
<b>4.2</b>	<b>Atena Solutions</b>	<b>62</b>
4.2.1	Groovoo	62
<b>4.3</b>	<b>Metodologia Scrumban Groovoo</b>	<b>62</b>
4.3.1	Time de Produto	63
4.3.2	Scrum Master	63
4.3.3	<i>Time de Desenvolvimento</i>	64
4.3.4	Time de Infraestrutura	64
4.3.5	Processos do <i>Groovoo</i>	64
<b>4.4</b>	<b>Trabalhos anteriores envolvendo o processo de desenvolvimento da <i>Atena Solutions</i></b>	<b>66</b>
<b>4.5</b>	<b>Considerações Finais do Capítulo</b>	<b>67</b>
<b>5</b>	<b>DIRETRIZES DE UCD PARA O <i>KICK-START KANBAN</i></b>	<b>68</b>
<b>5.1</b>	<b>Considerações Iniciais do Capítulo</b>	<b>68</b>
<b>5.2</b>	<b>Análise e seleção de abordagens UCD</b>	<b>68</b>
<b>5.3</b>	<b>Análise e proposição do <i>Kick-start Kanban</i> com as técnicas UCD</b>	<b>74</b>
5.3.1	Reunião para escolha das técnicas UCD analisadas	74
5.3.2	Atividades adicionadas ao <i>Kick-start Kanban</i>	74
5.3.3	Proposta de execução do workshop com as técnicas escolhidas e atividades adicionadas	77
<b>6</b>	<b>EXECUÇÃO DO <i>WORKSHOP KICK-START KANBAN</i> COM AS TÉCNICAS UCD SELECIONADAS</b>	<b>83</b>
<b>6.1</b>	<b>Reunião de alinhamento com a Equipe</b>	<b>83</b>

<b>6.2</b>	<b>Atividades da Fase 1 - Compreendendo a equipe</b>	<b>83</b>
6.2.1	Apresentação do <i>Kanban</i>	83
6.2.2	Apresentação do <i>Design</i> Centrado no Usuário (UCD)	83
6.2.3	Contexto da equipe	84
6.2.4	Desafios relatados	84
6.2.5	Expectativas relatadas	84
6.2.6	Preparação do Gerente de Fluxo	85
6.2.7	Decisão <i>GO/NO GO</i>	85
<b>6.3</b>	<b>Atividades da Fase 2 - Planejamento do <i>workshop</i></b>	<b>85</b>
6.3.1	Criação de um Planejamento	85
6.3.2	Configuração do <i>Workshop</i>	85
6.3.3	Planejamento da técnica Questionário	86
6.3.4	Planejamento da técnica Avaliação Heurística	86
6.3.5	Preparativos extras para a equipe	86
<b>6.4</b>	<b>Atividades da Fase 3 - Execução do <i>workshop</i></b>	<b>86</b>
6.4.1	Definir o <i>workshop</i>	86
6.4.2	Definir o UCD	87
6.4.3	Identificação das preocupações atuais	87
6.4.4	Definição de visão compartilhada	87
6.4.5	Descobrir o que a equipe faz	87
6.4.6	Identificar e agrupar os itens de trabalho com base em características comuns	87
6.4.7	Explicar como funciona o <i>Kanban</i>	88
6.4.8	Explicar como funciona a técnica Questionário	88
6.4.9	Criação do Modelo de Questionário	89
6.4.10	Apresentação da técnica Avaliação Heurística	89
6.4.11	Condução da Avaliação Heurística	89
6.4.12	Linguagem Comum	89
6.4.13	Definição de Políticas de visualização	90
6.4.14	Criação e preenchimento do quadro <i>Kanban</i>	90
6.4.15	Definição das políticas de trabalho	90
6.4.16	Limitação do <i>WIP</i>	92
6.4.17	Definição das políticas para reuniões de planejamento	92
6.4.18	Realização da primeira reunião de planejamento	93
<b>6.5</b>	<b>Encerramento do <i>workshop</i></b>	<b>94</b>
<b>7</b>	<b>RESULTADOS OBTIDOS APÓS A EXECUÇÃO DO WORKSHOP</b>	<b>95</b>
<b>7.1</b>	<b>Principais pontos observados</b>	<b>95</b>
<b>7.2</b>	<b>Questionário de <i>feedback</i></b>	<b>97</b>
<b>7.3</b>	<b>Discussão dos Resultados</b>	<b>102</b>

8	CONCLUSÃO . . . . .	104
	REFERÊNCIAS . . . . .	106
	ANEXOS	112
	APÊNDICES	113
	APÊNDICE A – CARTA DE SOLICITAÇÃO DE ACESSO E CITAÇÃO . . . . .	114
	APÊNDICE B – REGRAS PARA MONTAGEM DO QUESTIONÁRIO . . . . .	116
	APÊNDICE C – EXEMPLOS DE PERGUNTAS DO QUESTIONÁRIO . . . . .	119
	APÊNDICE D – AVALIAÇÃO HEURÍSTICA COM A EQUIPE . . . . .	123
	APÊNDICE E – MOTIVAÇÕES PARA A ADIÇÃO DO DESIGN CENTRADO NO USUÁRIO (UCD) AO PROCESSO DE DESENVOLVIMENTO GROOVOO . . . . .	131
	APÊNDICE F – DICIONÁRIO DE TERMOS TÉCNICOS . . . . .	134
	APÊNDICE G – TEMPLATE DE <i>ISSUES</i> . . . . .	138
	APÊNDICE H – COLUNAS DO QUADRO KANBAN . . . . .	139

# 1 Introdução

## 1.1 Considerações Iniciais do Capítulo

Este capítulo apresenta a contextualização deste trabalho, uma breve revisão bibliográfica, a questão de pesquisa, o objetivo geral e os específicos, bem como a metodologia adotada. Finaliza com a apresentação da estrutura dessa monografia.

## 1.2 Contexto

Desde o Manifesto Ágil, em 2001, os métodos ágeis têm sido empregados pela Engenharia de Software e vêm ganhando popularidade ano após ano (SCHÖN *et al.*, 2017). Para Martin, Anslow e Johnson (2017), esses métodos priorizam a flexibilidade, a colaboração e a capacidade de resposta a mudanças, abrangendo uma mentalidade que valoriza a adaptação contínua, a iteração rápida e a comunicação constante.

Nos últimos anos, houve um aumento significativo na utilização de abordagens ágeis no gerenciamento de projetos (BARBOZA *et al.*, 2016). Por meio de práticas como *Scrum* e *Kanban*, os métodos ágeis se concentram no desenvolvimento incremental e iterativo. A agilidade e a flexibilidade utilizadas por diferentes práticas ágeis, como iterações curtas e integração contínua, permitem que as organizações de software abordem as mudanças de forma eficaz (PANTIUCHINA *et al.*, 2017).

O *Scrum* é reconhecido por sua estrutura bem definida, focando em papéis e eventos claros para acelerar o fluxo de informações e reduzir o tempo de execução de projetos (CARVALHO; MELLO, 2009). Seu objetivo central é entregar um *software* útil para o cliente (SCHWABER, 1997). Em contrapartida, o *Kanban* é menos prescritivo e é utilizado para monitorar o fluxo de trabalho em projetos (IKONEN *et al.*, 2010). Seus elementos centrais incluem a visualização do fluxo de trabalho, a limitação do trabalho em andamento e a medição do tempo de execução (KNIBERG, 2009; NIKITINA; KAJKOMATTSSON; STRALE, 2012). Além disso, visa minimizar custos, aumentar a qualidade e reduzir o tempo de ciclo de trabalho (BOUGROUN; ZEAARAOUI; BOUCHENTOUF, 2014).

Os principais motivos de falha em projetos ágeis incluem: falta de experiência com a metodologia, compreensão limitada dentro da organização e os conflitos entre a cultura da empresa e os princípios ágeis (MILLER, 2013).

Embora a abordagem ágil seja conhecida por sua natureza iterativa e preveja usuários durante o teste, há uma percepção de que, na prática, dado o foco nos resultados

e na agilidade, não há necessariamente espaço para a criação a partir da experiência dos usuários finais (ALMUGHAM; ALYAHYA, 2017). Contudo, Patterson et al. (2017) e Choma et al. (2022) ressaltam que trazer a experiência para o processo de desenvolvimento pode agregar valor ao produto final.

A abordagem de *Design Centrado no Usuário* (do inglês *User Centered Design - UCD*) busca compreender as necessidades, desejos e limitações das pessoas para as quais um produto é destinado. Em vez de impor inovações ou recursos que podem não se alinhar com as expectativas ou capacidades do usuário, o UCD se baseia na coleta de *feedback*, na observação e na iteração constante para criar algo que seja verdadeiramente útil e relevante (RITTER; BAXTER; CHURCHILL, 2014).

Para Silva et al. (2013), as *Abordagens Centradas no Usuário* têm o potencial de aprimorar o Método Ágil ao oferecer maneiras sistemáticas de analisar as necessidades dos usuários finais. Por outro lado, o método ágil pode aperfeiçoar o UCD ao permitir iterações e entregas mais frequentes, o que resulta, respectivamente, em avaliações de usabilidade mais regulares e obtenção de *feedbacks* contínuos (SILVA et al., 2013), conceito chave do UCD (MCINERNEY; MAURER, 2005).

No universo dinâmico das *startups de software* muitas se encontram em um contexto desafiador, uma vez que operam sob constantes pressões de tempo e negócio e usualmente caem em problemas de comunicação e colaboração entre as equipes (SAAD et al., 2023). Enquanto a pressão por entregas rápidas é incessante e crucial para a sobrevivência e crescimento dessas empresas, profissionais relatam a importância e a competitividade que as *Abordagens Centradas no Usuário* e *Experiência da Usuário* (UX) trazem ao produto (HOKKANEN; KUUSINEN; VÄÄNÄNEN, 2015), ao mesmo tempo que percebe-se a dificuldade em trabalhar com elas durante o desenvolvimento de *software* (GREGORY et al., 2021).

Nesse cenário, as *startups* de software, conhecidas por sua agilidade e flexibilidade, ao empregar metodologias como *Scrum* ou *Kanban*, conseguem iterar e ajustar suas estratégias, processos e produtos de forma contínua (BANIJAMALI et al., 2017). No entanto, se as abordagens ágeis não forem utilizadas corretamente, podem criar mais problemas do que soluções (OLIVEIRA; GOMES; LIMA, 2014).

O *Kick-start Kanban* surge como uma alternativa a execução de um *Kanban*. É um *workshop* originado da necessidade de lidar com problemas de controle do fluxo de trabalho, estresse da equipe e insatisfação dos clientes. É uma abordagem eficaz para introduzir o método *Kanban* nas equipes de forma rápida e adaptável, além de ser modelado para ser facilmente aceito por gerentes e equipes sob pressão (ACHOUANTZ; NORDIN, 2013). O *Kick-start Kanban* conta com quatro etapas principais, onde cada etapa possui tarefas estruturadas. Essas tarefas ajudam a compreender o contexto da equipe, planejar e executar o *Kick-start Kanban* e também a atingir a linha de base de melhoria contínua.

Ao compreender questões como contexto da equipe, estabelecer expectativas claras e preparar os participantes, a implementação do *Workshop* torna-se mais fluida. Durante a execução do *Kick-start Kanban*, as etapas estruturadas guiam a equipe na definição de políticas de visualização, limitação de trabalho em andamento e criação de um quadro *Kanban*. Essas e outras atividades são essenciais para criar uma base sólida de melhoria contínua, onde a equipe adquire controle sobre seu fluxo de trabalho, identifica problemas atuais e estabelece políticas para otimizar sua eficiência (ACHOUIANTZ; NORDIN, 2013). Para Achouiantz e Nordin (2013), a flexibilidade do *Kick-start Kanban* também permite a aplicação em diferentes contextos e níveis de maturidade das equipes, tornando-o uma ferramenta valiosa para impulsionar a implementação bem-sucedida do *Kanban*.

Este trabalho tem como cenário a *startup* norte-americana *Atena Solutions*, em que objeto de estudo é o Processo de Desenvolvimento de Software do *Projeto Groovoo*. O *Groovoo* é um aplicativo de venda e emissão de ingressos para eventos sediados em estados norte-americanos, que tem seu processo de desenvolvimento fundamentado nas metodologias ágeis *Scrum*, *Kanban* e *Scrumban*.

A *Atena Solutions* é uma *Startup* com foco no desenvolvimento de *software* e *cloud computing*, com operações em São Francisco (CA), Austin (TX), Nova York (NY), São Paulo (SP), Brasília (DF), Manaus (AM) e Vancouver (BC). Atualmente a empresa possui cerca de 18 colaboradores divididos em alguns projetos internos e de empresas parceiras. O projeto *Groovoo*, objeto selecionado para o estudo deste trabalho, está disponível em ambientes *Web* e *Mobile*, e já conta com o processamento de quase 17 milhões de reais em pouco mais de dois anos.

### 1.3 Problema

No projeto *Groovoo*, o ciclo de desenvolvimento ágil adotado (em partes) é o *Scrumban*, que ajuda a gerenciar as tarefas de forma eficiente. No entanto, devido a prazos apertados, às vezes não é possível compreender por completo o que os usuários realmente precisam, tampouco apresentar uma nova proposta de processo de desenvolvimento para corrigir as deficiências. Isso pode resultar em problemas, como a insatisfação dos usuários em relação as novas funcionalidades. Além disso, as políticas de trabalho podem acabar não sendo respeitadas, gerando *Dailys* longas, falta de critérios no gerenciamento das demandas, má priorização, desorganização no fluxo de trabalho, comunicação descentralizada, dentre outros.

Os autores Gomes e Lucas (2023) propuseram diretrizes de *Design Centrado no Usuário* (UCD) para a integração no processo *Scrumban* adotado pela empresa. Na proposta, embora tenham sido delineadas diretrizes detalhadas, a implementação não foi possível devido às restrições de tempo. Diante desse cenário, os gestores anseiam por uma

abordagem mais viável e imediata, visando gerar valor para a empresa. O emprego do *Workshop Kick-start Kanban* pode ser um facilitador, auxiliando as equipes a implementarem de maneira rápida e eficaz o método *Kanban*.

Nesse contexto, a pergunta de pesquisa é *como incorporar atividades centradas no usuário no desenvolvimento de software do projeto Groovoo por meio do Workshop Kick-start Kanban, buscando satisfação do usuário e correção de limitações e deficiências no fluxo atual de trabalho?*

## 1.4 Objetivos

### 1.4.1 Objetivo Geral

O objetivo geral deste trabalho é *investigar o emprego de abordagens UCD no Workshop Kick-Start Kanban para o Processo de Desenvolvimento de Software Groovoo da startup Atena.*

### 1.4.2 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Analisar e selecionar abordagens de UCD a serem empregadas no *Workshop Kick-start Kanban* para o Processo de Desenvolvimento de Software do Groovoo;
- Analisar e propor a execução do *Workshop Kick-start Kanban* com as *Abordagens UCD Selecionadas* para a parte Kanban do Processo de Desenvolvimento de Software do Groovoo;
- Validar a proposta do *Workshop Kick-start Kanban com as Abordagens UCD selecionadas* para o *Processo de desenvolvimento de Software do Groovoo.*

## 1.5 Metodologia

Para o desenvolvimento deste trabalho a metodologia de pesquisa foi classificada e um plano metodológico foi definido.

### 1.5.1 Classificação Metodológica

Conforme o objetivo deste trabalho que é propor, baseado na literatura, diretrizes de UCD para processos que empregam o *Workshop Kick-start Kanban*, esta pesquisa é de natureza *aplicada*, pois tem como objetivo aplicar os resultados obtidos com a investigação na resolução de um problema (GIL, 2002).

A abordagem é *qualitativa*, já que esta pesquisa faz uso da subjetividade e que não pode ser traduzida em números (GIL, 2002). Neste caso, os resultados obtidos são provenientes dos indivíduos, seus interesses, conflitos e demais aspectos considerados subjetivos.

Quanto ao objetivo, a pesquisa é *descritiva*, pois visa descrever características de um determinado fenômeno ou população no momento da investigação e estabelecer relações entre variáveis (GIL, 2002).

A técnica empregada é o *Estudo de caso*. Ela implica no estudo aprofundado de um ou de um pequeno número de objetos, com o intuito de adquirir um conhecimento detalhado e abrangente (GIL, 2002). Para emprego da técnica Estudo de caso, adotou-se o *Protocolo de Estudo de Caso* (BRERETON et al., 2008).

O plano metodológico definido compreende quatro grandes fases: Planejamento; Coleta de Dados; Análise de Dados; e Relatórios (BRERETON et al., 2008), detalhado no **Capítulo 3 - Materiais e Métodos**.

## 1.6 Organização do Trabalho

Este trabalho está organizado em quatro capítulos. Neste, **Capítulo 1 - Introdução**, apresenta-se a contextualização, o problema, objetivos de pesquisa, e a metodologia aplicada. Ademais, os restantes estão estruturados no seguinte formato:

- **Capítulo 2 - Referencial Teórico:** Descreve conceitos acerca de Metodologias Ágeis junto a alguns dos seus principais métodos, bem como os potenciais benefícios e obstáculos relacionados à sua aplicação. O capítulo é subdividido nas seções *Metodologias ágeis*, *Design Centrado no Usuário* e *Kick-Start Kanban*;
- **Capítulo 3 - Materiais e Métodos:** Neste capítulo apresenta-se o planejamento metodológico da pesquisa;
- **Capítulo 4 - Atena Solutions:** Neste capítulo é apresentado a *Startup Atena Solutions* e o projeto Groovoo, objeto de estudo deste trabalho. Apresenta-se ainda, uma proposta acadêmica de Diretrizes UCD ao processo Groovoo.
- **Capítulo 5 - Diretrizes de UCD para o *Kick-start Kanban*:** Apresenta a análise e seleção de técnicas UCD para o projeto Groovoo e insere as escolhas no *workshop Kick-start Kanban*.
- **Capítulo 6 - Execução do *workshop Kick-start Kanban* com as técnicas as UCD selecionadas:** Neste capítulo é apresentado o detalhamento da execução do *workshop Kick-start Kanban* com as técnicas UCD selecionadas e a equipe da Atena Solutions.

- **Capítulo 7 - Resultados** : Neste capítulo são apresentados os resultados obtidos com a execução do *workshop Kick-start Kanban*.
- **Capítulo 8 - Conclusão**: Neste capítulo são apresentadas as conclusões do trabalho alinhadas aos objetivos propostos, bem como as limitações e sugestões para trabalhos futuros.

## 2 Referencial Teórico

### 2.1 Considerações Iniciais do Capítulo

Este capítulo apresenta o referencial teórico deste trabalho.

### 2.2 Metodologias Ágeis

Metodologias ágeis representam um conjunto de abordagens e valores destinados a entregar produtos ou serviços de maneira incremental e iterativa, com uma ênfase contundente na satisfação do clientes.

Em 2001, uma reunião de um grupo de 17 empresários da indústria de software e representantes de metodologias como *Extreme Programming*, *SCRUM*, *DSDM*, *Adaptive Software Development*, *Crystal*, *Feature-Driven Development*, *Pragmatic Programming* (BECK et al., 2001; AYALA; URIBE, 2007), promove um manifesto, denominado Manifesto Ágil.

O manifesto foi criado como uma alternativa as abordagens tradicionais de desenvolvimento de software, marcando o início do movimento ágil na indústria de desenvolvimento de *software* (BECK et al., 2001).

Os empresários se uniram em concordância com a necessidade de uma alternativa aos processos de desenvolvimento de software que eram orientados por documentações extensas e consideradas pesadas (BECK et al., 2001). Críticas a modelos como: modelo cascata; modelo incremental; e modelo espiral (AYALA; URIBE, 2007).

Criado como uma alternativa as abordagens tradicionais de desenvolvimento de software, marcando o início do movimento ágil na indústria de desenvolvimento de *software* (BECK et al., 2001).

O Manifesto Ágil possui quatro valores fundamentais, no qual contrapõem as práticas comuns, conforme descritos na Tabela 1. Os dados apresentados demonstram que, embora exista valor nos itens à direita, os itens à esquerda devem ser **mais** valorizados (BECK et al., 2001).

Tabela 1 – Valores do Manifesto Ágil - (BECK et al., 2001)

<b>Valorizar</b>	<b>Acima de</b>
Indivíduos e interações	Processos e Ferramentas
Software funcionando	Documentação complexa
Colaboração do consumidor	Negociação contratual
Responder às mudanças	Seguir um plano

A partir dos **quatro valores** foram derivados **doze princípios ágeis**:

1. A maior prioridade é satisfazer o cliente a partir da entrega antecipada e contínua de *software* valioso;
2. Receber bem as mudanças nos requisitos, mesmo tarde no desenvolvimento. Os processos ágeis aproveitam as mudanças em prol da vantagem competitiva do cliente;
3. Entregar *software* funcionando com frequência, de algumas semanas a alguns meses, com preferência pelo prazo mais curto;
4. Pessoas de negócios e desenvolvedores devem trabalhar juntos diariamente durante o projeto;
5. Construir projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessários e confie neles para fazer o trabalho;
6. O método mais eficiente e eficaz de transmitir informações para e dentro de uma equipe de desenvolvimento é a conversa face a face;
7. *Software* funcionando é a principal medida de progresso.
8. Processos ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
9. A atenção contínua à excelência técnica e ao bom *design* aprimora a agilidade;
10. Simplicidade - a arte de maximizar a quantidade de trabalho não realizado - é essencial;
11. As melhores arquiteturas, requisitos e *designs* emergem de equipes auto-organizadas;
12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz, ajustando seu comportamento de acordo.

Segundo [Dingsøyr et al. \(2012\)](#), esses princípios geraram várias práticas que agregaram mais valor aos clientes, além deles não serem simplesmente uma definição formal de agilidade, e sim diretrizes que visam fornecer *software* de qualidade de forma ágil.

As primeiras metodologias ágeis surgiram entre as décadas de 80 e 90, com a utilização generalizada dos computadores, visando atender, de forma rápida e sem comprometer a qualidade do *software*, as expectativas e necessidades urgentes e frequentes dos usuários ([AYALA; URIBE, 2007](#)). O termo Ágil refere-se a capacidade em que um projeto responde de forma eficaz dentro de um ambiente em constante mudança ([MAFAKHERI; MOUSAVI, 2008](#)), além de ter as características de "ser dinâmico, específico ao contexto, abraçar mudanças de forma proativa e orientado para o crescimento" ([GOLDMAN; PREISS; NAGEL, 1995](#)) e "a capacidade de uma empresa prosperar em um ambiente de mudanças contínuas e imprevisíveis" ([DOVE; FORUM, 1996](#)).

Após o Manifesto Ágil, seus debates acerca das metodologias, princípios e valores que devem guiar o desenvolvimento de *software* de boa qualidade, obteve-se uma definição e a aceitação do termo Ágil para se referir a novas abordagens metodológicas para o desenvolvimento de *software* de boa qualidade, em prazos curtos e flexíveis para mudanças, respeitando então os princípios do Manifesto ([AYALA; URIBE, 2007](#)).

De acordo com [Beck et al. \(2001\)](#), essas metodologias baseiam-se em princípios essenciais, incluindo a valorização do indivíduo e das interações, a priorização de software plenamente funcional, a colaboração ativa com o cliente, a capacidade de resposta ágil às mudanças, e o compromisso contínuo com a entrega de *software* de alta qualidade ao longo do tempo.

As metodologias ágeis se opõem às abordagens tradicionais de desenvolvimento de *software*, como as do *Modelo em Cascata*. A exemplo, segundo [Carvalho \(2001\)](#), o Modelo em Cascata tende a ser mais rígido e baseado em planos extensos, já que existe um gasto razoável de tempo e esforços para levantar informações, especificá-las e documentá-las dentro de cada etapa para serem utilizadas posteriormente. Em contrapartida, as metodologias ágeis possuem uma abordagem com maior adaptabilidade e flexibilidade quanto as necessidades de mudanças nos requisitos do projeto, com resultados que devem ser entregues ao cliente em curtos espaços de tempo ([SUTHERLAND, 2016](#)).

Além dos princípios, as metodologias ágeis destacam a importância da comunicação aberta e eficaz entre os membros da equipe e os *stakeholders*. Esse foco na comunicação é essencial para compreender as necessidades dos clientes, suas dificuldades e desafios, ajustar prioridades e explorar caminhos alternativos que não são possíveis quando não se trabalha de forma colaborativa ([MAFAKHERI; MOUSAVI, 2008](#)). [Mafakheri e Mousavi \(2008\)](#) ainda ressalta que, o trabalho colaborativo permite explorar possibilidades mais adequadas que o trabalho individual, resultando em soluções que geralmente superam aquilo que é alcançado individualmente.

Nesse sentido, existem várias Metodologias Ágeis que, em maior ou menor grau, seguem o Manifesto Ágil (DINGSØYR et al., 2012), implementando seus valores e princípios. Algumas dessas metodologias incluem o desenvolvimento *eXtreme Programming* (XP), o *Scrum*, o *Kanban*, o desenvolvimento orientado por funcionalidades (FDD), as metodologias *Crystal*, entre outras (AYALA; URIBE, 2007; DINGSØYR et al., 2012).

Neste trabalho, o objeto de estudo, que é a *Startup Atena*, adota a abordagem *ScrumBan*.

### 2.2.1 Scrumban

*Scrumban* é uma abordagem que combina elementos de dois modelos de desenvolvimento de software: *Scrum* e *Kanban* (KHAN, 2014). É uma técnica ágil de gerenciamento de projeto que incorpora a prescrição do *Scrum* para planejamento e controle de projetos, ao mesmo tempo que estimula a melhoria do processo *Kanban*, integrando o gerenciamento visual e possibilitando que as equipes possam melhorar continuamente seus processos (KHAN, 2014; BERTOLINI DAVIDE MEZZOGORI; ZAMMORI, 2023). Essa integração busca aproveitar o melhor de ambos os métodos para melhorar a rotina de trabalho e garantir uma utilização eficaz dos recursos (BERTOLINI DAVIDE MEZZOGORI; ZAMMORI, 2023). A Figura 1 apresenta a combinação do *Scrum* e do *Kanban*.

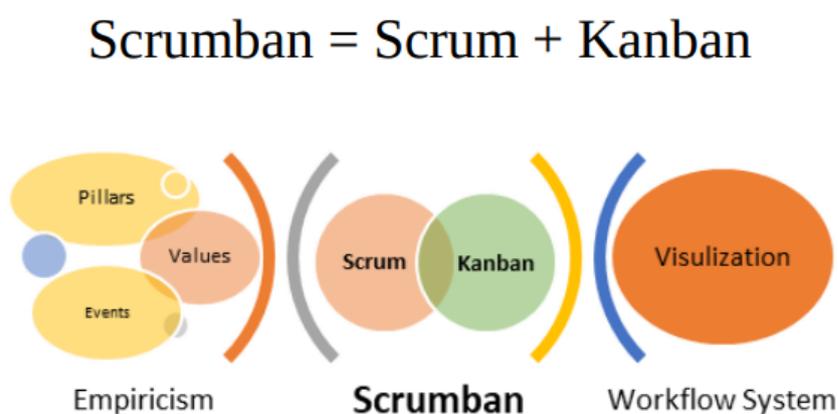


Figura 1 – Exemplo de um quadro *Scrumban* (BHAVSAR; GOPALAN; SHAH, 2020)

Conforme apresentado, o *Scrumban* é uma combinação dos métodos ágeis *Kanban* e *Scrum*, combinando a agilidade com gestão transparente do fluxo de trabalho.

#### 2.2.1.1 Princípios e Técnicas

O *Scrumban* possui sete princípios originados de diretrizes dos *frameworks Scrum* e *Kamban* e ajustados às necessidades e requisitos de uma organização ou equipe. Para Khan (2014), os sete princípios são:

- **Visualizar o fluxo de trabalho** - Esse é um princípio retirado do *Kamban* e tem como objetivo a visualização do fluxo de trabalho por meio de colunas, onde é possível que a equipe visualize os diferentes estágios pelos quais os itens do *backlog* do produto passam até sua conclusão. Com isso, se tornam mais visíveis os pontos de gargalo no fluxo de trabalho, quem está trabalhando em quê e o estado atual de uma determinada tarefa (KHAN, 2014; REDDY, 2015). A figura 2 exemplifica a estrutura de um quadro *Scrumban*.

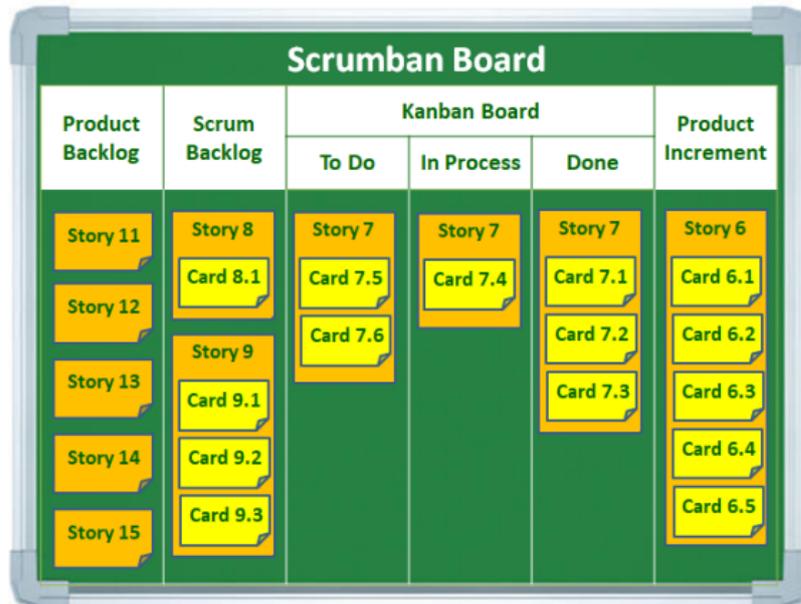


Figura 2 – Exemplo de um quadro Scrumban (BHAVSAR; GOPALAN; SHAH, 2020)

O **Scrumban** implementa novas colunas ao quadro de visualização para adaptar o quadro *Kanban*, de forma que atenda às necessidades e características do *Scrum*. Isso porque o quadro *Kanban* tradicional não inclui colunas referentes ao *Scrum*. Vale ressaltar que a implementação do quadro e as propriedades que o compõem vão de acordo com as necessidades da empresa;

- **Pull Work** - Dentro do *Scrumban*, as tarefas são movimentadas de forma flexível e quando necessário. Reddy (2015) define que os mecanismos de *Pull* evitam sobrecarregar o sistema com muito trabalho. Ou seja, Em vez de uma demanda ser diretamente atribuída para a equipe, é a equipe que seleciona a demanda para incluí-la no seu fluxo de trabalho quando tiver capacidade e disponibilidade para lidar com ela. Essa é uma abordagem que permite que as equipes controlem ativamente com o que estão trabalhando e quando estão trabalhando, em vez de receberem trabalho de forma passiva. Dentro desse contexto de movimentação de tarefas, Ladas (2008) propõe uma melhoria no quadro chamada *Ready Queue*. Basicamente a *Ready Queue* é uma coluna que fica localizada entre as colunas de *Backlog* e de trabalho em andamento. Em resumo, ela contém itens pendentes do *Backlog* e que possuem

maior precedência em relação aos *Backlog* em si, sendo vinculados a um indivíduo conforme sua disponibilidade. Isso acaba desvinculando o processo de atribuição de trabalho do processo de priorização de trabalho, simplificando a atribuição em si (LADAS, 2008);

- **Limite de Trabalho em Andamento (*Work-in-Progress* - *WIP*)** - A aplicação de limites para o trabalho em andamento (WIP) é uma prática importante. Isso significa que as equipes definem quantas tarefas ou itens de trabalho podem estar em andamento em cada estágio do processo de desenvolvimento de *software* (LADAS, 2008; KHAN, 2014; REDDY, 2015). Essa abordagem ajuda a equipe a se concentrar em concluir o trabalho em vez de iniciar novas tarefas constantemente. Quando um estágio atinge seu limite de WIP, a equipe não inicia novos trabalhos nessa etapa. Em vez disso, eles direcionam sua energia para ajudar outros membros da equipe ou resolver problemas em andamento, evitando sobrecarregar a equipe e garantindo que não fiquem presos em gargalos, tornando assim o fluxo de trabalho mais suave (KHAN, 2014);
- **Políticas de Trabalho Explícitas** - Estabelecer políticas de trabalho explícitas é fundamental. Tornar o processo explícito promove a colaboração entre os membros da equipe, que podem se auto-organizar e coordenar de forma mais tranquila (KHAN, 2014). As políticas de equipe explícitas ajudam os membros da equipe "a se gerenciarem, a tomar decisões mais rápidas, sem colocar muito esforço para pensar, e até mesmo a reduzir a probabilidade de ceder a solicitações especiais sob estresse" (KHAN, 2014). Também busca que "todos estejam na mesma página e que as políticas de trabalho possam ser facilmente lembradas e compartilhadas com outras pessoas" (REDDY, 2015). Essas políticas podem estar relacionadas ao processo de trabalho ou à organização como um todo. Elas ajudam a enfrentar situações recorrentes e garantem que todos estejam na mesma página. A equipe deve definir políticas de trabalho explícitas, como: limites de WIP, como lidar com gargalos dentro das etapas, como os bugs e prioridades serão resolvidos dentro da equipe, quem pode criar histórias, políticas de comunicação, e como funcionará a comunicação entre os membros da equipe (KHAN, 2014);
- **Reuniões de Planejamento Objetivas** - As reuniões de planejamento se diferenciam da abordagem tradicional do *Scrum*, já que elas são mais curtas e flexíveis, realizadas para atualizar a fila do *Backlog* à medida que necessário. É importante manter um planejamento de trabalho ideal, fornecendo à equipe a quantidade certa de tarefas a serem realizadas, nem mais nem menos (LADAS, 2008). Para Reddy (2015), a adoção do Scrumban proporciona uma mudança significativa no foco das reuniões. Em vez de se concentrar apenas no status e compromissos, as discussões se tornam mais proativas. O quadro *Kanban* já fornece informações de status e de-

talhes sobre o trabalho em andamento, incluindo quais membros da equipe estão envolvidos nele. Também apresenta os impedimentos que possam surgir durante a fase de desenvolvimento. Por conta disso, as reuniões passam a ser oportunidades colaborativas para identificar possíveis obstáculos, lidar com exceções às políticas estabelecidas e explorar descobertas sobre como o trabalho está sendo executado;

- **Revisão, Retrospectivas e Reuniões *Stand-up* Diárias** - Para o contexto do *Scrumban*, algumas características do *Scrum*, como a Revisão, Retrospectivas e Reuniões *Stand-up* Diárias (também conhecidas como Frequência de Sincronização (REDDY, 2015)), são etapas essenciais (KHAN, 2014). A Revisão permite que a equipe obtenha *feedback* direto dos proprietários do produto e partes interessadas, como gerentes de produto e clientes. Ela frequentemente ocorre em intervalos regulares, como no *Scrum*, onde normalmente é uma preferência dos interessados no produto. Já a Retrospectiva é o espaço onde a equipe aprimora suas regras e processos, buscando oportunidades de melhoria e planejando maneiras de aperfeiçoar seu trabalho ainda mais. Por meio das Reuniões *Stand-up* Diárias (comumente conhecidas como *Dailys*), os membros da equipe se mantêm atualizados sobre o andamento das tarefas, coordenam às atividades, identificam e resolvem os impedimentos que surgem, garantindo um fluxo de trabalho mais eficiente. Segundo Reddy (2015), ao contrário do *Scrum*, o *Scrumban* permite que essas reuniões evoluam para colaborações além do status de trabalho e tomada de decisões. A frequência das sincronizações pode variar, de uma vez por dia a atualizações em tempo real, dependendo das necessidades da equipe e das ferramentas escolhidas.
- **Métricas** - O *Scrumban* propõe uma abordagem mais flexível e eficiente em comparação com o *Scrum* tradicional, especialmente no que diz respeito a métricas e estimativas. Enquanto o *Scrum* depende de métricas como pontos de história e velocidade, o *Scrumban* prefere focar em métricas como tempo de ciclo e tempo de espera (KHAN, 2014). Em vez de estimar cada história individualmente, o *Scrumban* sugere agrupar histórias de tamanho semelhante, onde a equipe pode definir a quantidade de tarefas a serem priorizadas, baseando-se estatisticamente nos itens de trabalhos já concluídos anteriormente.

## 2.3 Kick-start Kanban

O *Kick-start Kanban* é um método em formato de *workshop* proposto por Achouiantz e Nordin (2013) e pouco explorado na literatura. Ele é projetado para ajudar as equipes a implementar eficazmente um sistema *Kanban*. Este *workshop* é estruturado de forma a ser facilmente aceito por gerentes e equipes sob pressão ou com pouca disponibilidade de tempo para investir em novas estratégias. Durante o *workshop*, os membros da

equipe colaboram para entender o que a equipe faz, por que faz, como faz e para quem faz.

Com base nessa compreensão, a equipe desenvolve políticas de visualização para acompanhar o trabalho em andamento usando um quadro *Kanban*, bem como políticas que definem as regras de trabalho. Além disso, são definidas políticas de ação que orientam o planejamento e o gerenciamento do trabalho, com reuniões de planejamento realizadas conforme necessário, e que geralmente ocorrem diariamente.

O método *Kanban* foi escolhido como solução para desenvolver a capacidade de melhoria contínua das equipes. [Anderson \(2010\)](#) descrevem o *Kanban* como um método de mudança evolutivo, onde o processo é evolutivo e incremental, além de ser capaz de otimizar processos com resistência mínima e mantendo um ritmo sustentável. No entanto, [Achouiantz e Nordin \(2013\)](#) identificou o desafio de apresentar esse método às equipes com recursos limitados, principalmente em tempo.

Em essência, o *Kick-start Kanban* ajuda a equipe a obter controle sobre seu fluxo de trabalho, o que é a base para a melhoria contínua de uma equipe ([ACHOUARIANTZ; NORDIN, 2013](#)). Sem um sistema *Kanban* para fornecer esse controle e entendimento, pode ser difícil alcançar melhorias dentro da equipe e do processo. Portanto, o *Kick-start Kanban* é um passo importante para implementar o *Kanban* com sucesso, permitindo que as equipes entendam seu propósito, visualizem o trabalho em andamento, definam políticas de trabalho, planejem seu trabalho diário e otimizem a eficiência e eficácia da equipe.

### 2.3.1 História do *Kick-start Kanban*

O *Kick-start Kanban* não originou-se a partir do *Kanban*, mas sim da necessidade de lidar com problemas de controle do fluxo de trabalho, estresse da equipe e insatisfação dos clientes, com a percepção de que a situação não estaria satisfatória dentro das equipes no *Sandvik IT*. O *Sandvik IT* é "um grupo global de engenharia de alta tecnologia que fornece soluções que aumentam a produtividade, lucratividade e sustentabilidade para as indústrias de manufatura, mineração e infraestrutura" ([Sandvik Group, 2023](#)).

Segundo ([ACHOUARIANTZ; NORDIN, 2013](#)), as equipes não tinham um tempo hábil para dedicar a outras atividades que não estivessem dentro do seu fluxo de trabalho, então surgiu a necessidade de uma abordagem mais evolutiva. Tanto os funcionários quanto os clientes estavam enfrentando desafios, incluindo complexidade técnica crescente, dificuldades na previsão de entregas e insatisfação geral com o trabalho. Foi observado que as pessoas estavam estressadas e sobrecarregadas, pulando de uma tarefa para outra, sem controle sobre o que estava acontecendo. A Figura 3 apresenta o fluxo de trabalho das equipes.

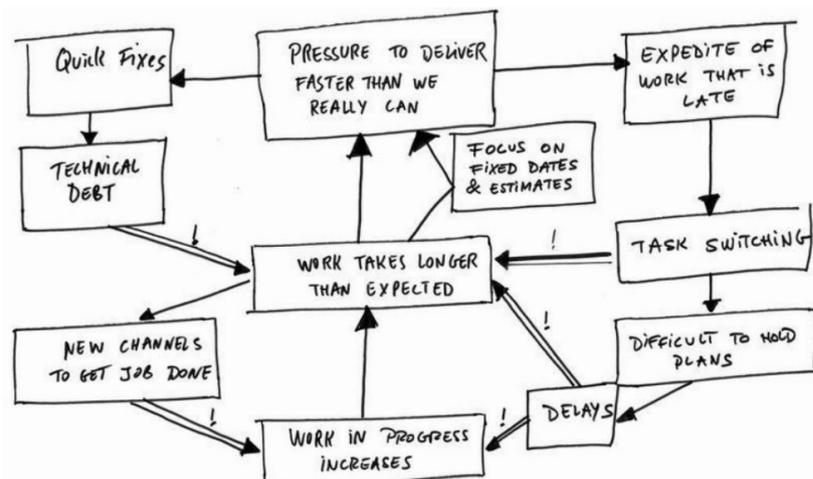


Figura 3 – Ciclo de desafios das equipes (ACHOUANTZ; NORDIN, 2013)

A Figura 3 revela um ciclo de problemas interligados dentro das equipes. A pressão para entregar rapidamente, por exemplo, pode levar a correções rápidas e à criação de débitos técnicos. Esses débitos técnicos, por sua vez, podem aumentar o tempo necessário para realizar o trabalho planejado, causando atrasos. A tentativa de agilizar o trabalho atrasado por vezes resulta em trocas frequentes de tarefas, podendo ser difícil manter planos e causando mais atrasos, o que, por sua vez, pode levar ao aumento do trabalho em andamento. Isso torna o ciclo de trabalho difícil e atrasos podem ocorrer constantemente no ambiente de trabalho.

Segundo Achouiantz e Nordin (2013), após uma análise mais aprofundada, ficou claro que a raiz de todos esses problemas estava na falta de controle sobre o fluxo de trabalho das equipes. Esta situação levou à necessidade de introduzir o *Kickstart Kanban*, uma abordagem que permitiria abordar esses problemas em pequenos passos, melhorando continuamente o controle do fluxo de trabalho e, assim, resolvendo as dores percebidas.

### 2.3.2 Aspectos chave do *Kick-start Kanban*

O *Kick-start Kanban* é uma abordagem específica para introduzir o método *Kanban* nas equipes, e possui algumas características distintas de outros métodos de implementação do *Kanban*. A tabela 2 apresenta essas características.

Tabela 2 – Aspectos chave do *Kick-start Kanban*

Aspectos Chave	Descrição
Equipe como Condu-tora	A equipe é a principal responsável por configurar seu sistema Kanban. O especialista responsável pelo treinamento não projeta o sistema Kanban para a equipe, mas auxilia a equipe a construí-lo
Foco na Evolução	A abordagem não visa necessariamente criar um sistema Kanban ideal. O principal objetivo é aumentar a compreensão da equipe sobre si mesma e seu contexto, permitindo que ela evolua por si só e atinja seu propósito
Entrega de Artefatos	Ao final do workshop, a equipe desenvolve um quadro de visualização, políticas de trabalho explícitas e uma maneira de trabalhar com o quadro desenvolvido
Preparação Anterior	Nem tudo acontece em um único dia; algumas preparações são necessárias antes do workshop, envolvendo os responsáveis, como gerentes e gerente de fluxo
Acompanhamento Con-tínuo	O especialista responsável pelo treinamento deve dar suporte à equipe após o workshop. Isso envolve ajudar a equipe a implementar conceitos do Kanban que podem não ter sido totalmente absorvidos durante o workshop e garantir que a equipe possa manter seu sistema Kanban a longo prazo
Adaptação	O Kick-start Kanban é flexível e pode ser adaptado para equipes com diferentes níveis de maturidade e necessidades específicas
Facilidade de Aceitação por Gestores	O Kick-start Kanban é projetado para ser facilmente aceito por gestores, já que é um workshop de um dia que envolve toda a equipe, onde pode ser uma opção atraente para gestores pressionados e com pouco tempo disponível

Essas características tornam o *Kick-start Kanban* uma abordagem interessante para a implementação do *Kanban*, focada no desenvolvimento da equipe pela própria equipe e na evolução gradual da implementação do sistema *Kanban* em vez de buscar uma solução perfeita desde o início.

### 2.3.3 Processo inicial do *Kick-start Kanban*

Como descrito anteriormente, o *Kick-start Kanban* é um *workshop* destinado a impulsionar uma equipe na utilização do método *Kanban* para apoio do fluxo de trabalho. No entanto, esse termo também é compreendido como um processo capaz de levar uma equipe do ponto em que se encontra para um estágio em que se possa melhorar de forma sustentável e contínua (ACHOUANTZ; NORDIN, 2013). O *Kick-start Kanban* ajuda as equipes a se orientarem quanto as preparações necessárias para a introdução do método *Kanban*, na realização do *workshop* em si e ao suporte posterior, permitindo que as equipes possa trabalhar em direção a suas metas.

Na Figura 4 apresentam-se as etapas do processo de implementação do *Kick-start Kanban*.

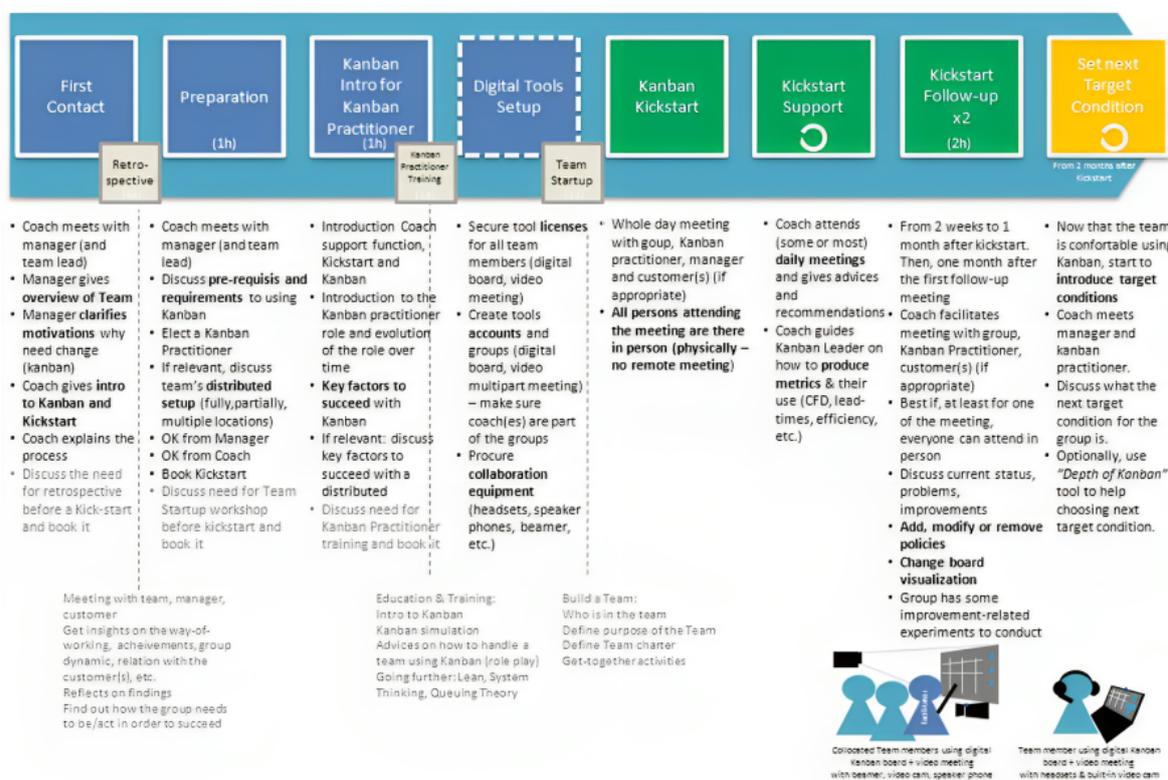


Figura 4 – Processo para implementação do *Kick-start Kanban* (ACHOUANTZ; NORDIN, 2013)

Basicamente é apresentado um *overview* do processo para implementar o *Kanban* de forma rápida e eficaz, compreendendo as limitações da empresa, principalmente de tempo. O objetivo desse processo é "que a equipe assuma o controle sobre sua forma atual de trabalhar e alcance uma linha de base estável de melhoria contínua" (ACHOUANTZ; NORDIN, 2013).

Essa linha de base "é o ponto onde a equipe está verdadeiramente consciente de sua condição atual e tem uma boa base para começar a melhorar" (ACHOUANTZ; NORDIN, 2013). Os autores Achouiantz e Nordin (2013) ainda citam que, sem a compreensão sobre a forma de se trabalhar, qualquer tentativa de melhorar os processos pode ser desorganizada e ineficaz. Em outras palavras, a equipe precisa estar consciente de como as coisas estão funcionando atualmente antes de fazer mudanças para melhorar.

Achouiantz e Nordin (2013) consideram que a equipe possui uma linha de base de melhoria contínua quando os membros:

- Compreendem a condição atual da equipe através de elementos que possam ser visualizados;

- Compreendem sua capacidade atual, utilizando indicadores de atraso, como prazos de entrega, rendimento, taxa de geração de defeitos, desempenho na data de vencimento, entre outros;
- Seguem uma abordagem comum para o trabalho, estabelecendo políticas explícitas (as políticas não precisam ser perfeitas e nem “certas”);
- Demonstram capacidade em manter e ajustar as políticas explícitas sobre o modo de trabalho, seja adicionando, removendo ou as modificando.

Para os autores [Achouiantz e Nordin \(2013\)](#), quando a equipe conseguir atingir essa linha de base, ela finalmente pode dar início às melhorias quanto à forma de trabalhar. E ressaltam que, a linha de base de melhoria contínua é a primeira condição-alvo essencial para a equipe.

### 2.3.4 Aplicação do *Kick-start Kanban*

Para implementar o *Kick-start Kanban*, é importante entender as fases que ocorrem antes, durante e após a sua implementação. Dentro dessas fases, é importante compreender um pouco mais sobre as partes envolvidas no processo, que são: a **equipe**; um **líder de equipe**; um **gerente de fluxo**; e um **especialista** que colocará em prática os conceitos do *Kick-start Kanban*.

Essas partes (ou indivíduos) são descritos, com o objetivo de esclarecer quem faz o quê ([ACHOUANTZ; NORDIN, 2013](#)):

- **Equipe** - A Equipe é o conjunto de indivíduos que executam o trabalho. O processo *Kick-start Kanban* é flexível o suficiente para atuar em equipes de desenvolvimento de software, gestão, recursos humanos e muitos outros.
- **Líder de equipe** - O Líder de Equipe desempenha um papel fundamental na melhoria contínua da equipe. O envolvimento dos líderes de equipe é essencial, pois o método *Kanban* torna problemas visíveis, e a falta de ação dos líderes pode desmotivar a equipe.
- **Gerente de Fluxo** - O Gerente de Fluxo lidera a implementação do *Kanban* na equipe, garantindo que as políticas sejam seguidas, criando políticas quando necessário e promovendo soluções criativas. O gerente de fluxo é uma extensão do especialista e assume a liderança da aplicação do *Kanban* após o treinamento.
- **Especialista** - O Especialista é responsável por preparar, executar e acompanhar o *workshop*. Ele auxilia a equipe na implementação do *Kanban* e, posteriormente, treina o Gerente de Fluxo. Esse papel é importante para capacitar as equipes.

### 2.3.4.1 Compreendendo a Equipe

A Figura 5 apresenta os primeiros passos que são pré-requisitos para a implementação do *Kick-start Kanban*. A ideia é que, antes de realizar qualquer atividade relacionada ao *Kick-start Kanban* em si, é importante entender, definir e preparar alguns aspectos sobre a equipe.

<p><b>PREP 1: Present Kanban as a Tool for Evolution</b> Present Kanban to the team manager and flow manager.</p>
<p><b>PREP 2: Study the Context</b> Meet the team manager and the flow manager to understand the team and its context. <i>Purpose   Context   Why Kanban?   Acceptance criteria</i></p>
<p><b>PREP 3: Set Expectations</b> Set expectations with the team manager and flow manager. <i>Manager   Flow Manager   Coach</i></p>
<p><b>PREP 4: Prepare the Flow Manager</b> Prepare the flow manager to be ready for the Kick-start <i>Role   Role over time   Responsibilities</i></p>
<p><b>PREP 5: Get a GO/NO GO Decision</b> You get a GO/NO GO decision from the Team manager to proceed with the Kanban implementation</p>

Figura 5 – Fase 1 do *Kick-start Kanban* (ACHOUANTZ; NORDIN, 2013)

Na Tabela 3 apresenta-se de forma mais detalhada as 5 etapas apresentadas na Figura 5. Antes de partir para a prática em si, é importante preparar a equipe e definir de forma clara as expectativas. Isso envolve apresentar o conceito do *Kanban* aos gerentes, estudar o contexto da equipe, estabelecer as expectativas com os gerentes e preparar o Gerente de Fluxo. Essas etapas buscam a garantia de que a equipe toda esteja alinhada com princípios e objetivos do *Kanban*. Além disso, é importante e crucial obter a decisão *GO/NO GO* do gerente de equipe para determinar se o processo deve prosseguir.

Tabela 3 – Fase 1 - Atividades do *Kick-start Kanban* (ACHOUANTZ; NORDIN, 2013)

Atividade	Descrição	Definição de Concluído	Pode ser ignorado?
Apresentar o <i>Kanban</i>	Apresentar o <i>Kanban</i> ao gerente de equipe e ao gerente de fluxo	Alguém da equipe tem competência para atuar como Gerente de Fluxo; Os gerentes compreender o <i>Kanban</i>	Somente se o os gestores forem especialistas em <i>Kanban</i>
Estudar o contexto da equipe	Conhecer o gerente da equipe e o gerente de fluxo para entender a equipe e seu contexto	O contexto da equipe é compreendido	Não
Definir as expectativas	Defina as expectativas com o gerente de equipe e o gerente de fluxo	As pré-condições necessárias foram atendidas para que a equipe implemente e sustente o <i>Kanban</i>	Não
Preparar o Gerente de Fluxo	Preparar o gerente de fluxo para estar pronto para o <i>Kick-start Kanban</i>	As expectativas são definidas para o gestor, gestor de fluxo e treinador.	Se a pessoa que atua como gestor de fluxo da equipe já possui experiência com o <i>Kanban</i>
Obter uma decisão <i>GO/NO GO</i>	Obter a decisão <i>GO/NO GO</i> do gerente de equipe para prosseguir com a implementação do <i>Kanban</i>	O gerente concorda e decide (ou não) prosseguir com o processo	Não

### 2.3.4.2 Planejamento do *Kick-start Kanban*

Após compreender o contexto da equipe e tornar claras as expectativas acerca do processo, é necessário planejar o *workshop* em si. Na Figura 6 apresentam-se as etapas do planejamento. A fase de planejamento conta com etapas como: *criação* de um planejamento para atender as necessidades da equipe; *configuração* do *workshop* para realização do mesmo; e os *preparativos* extras, para lidar com a comunicação diária.

<p><b>SETUP 1: Create an Agenda</b>            You build an agenda for the Kick-start workshop that fits the team's current situation and maturity.  <i>Select features   Pauses</i></p>
<p><b>SETUP 2: Setup the Workshop</b>            You go through all the little practical details for holding the workshop.  <i>Who attends   The board   Props   Room   Invitation</i></p>
<p><b>SETUP 3: Extra Preparations for a Distributed Team</b>            You have discussed and prepared the tools necessary for the team to communicate distributed.  <i>Collaboration tools</i></p>

Figura 6 – Fase 2 do *Kick-start Kanban* (ACHOUANTZ; NORDIN, 2013)

A Tabela 4 apresenta de forma mais detalhada as três etapas apresentadas na Figura 6 que correspondem a fase de planejamento. Além da descrição de cada atividade, ela indica os critérios para considerar uma atividade como concluída e também se ela

pode ou não ser ignorada. Essas etapas buscam garantir que o *workshop* seja devidamente preparado e esteja pronto para ser executado.

Tabela 4 – Fase 2 - Atividades do *Kick-start Kanban*

Atividade	Descrição	Definição de Concluído	Pode ser ignorado?
Criar um Planejamento	Um planejamento é criado para o <i>workshop</i> e que se adapta à situação atual e à maturidade da equipe	O planejamento foi feito e se adapta à situação atual da equipe	Somente se existir um planejamento
Configurar o <i>Workshop</i>	Definir os detalhes práticos necessários para a realização do <i>workshop</i>	Foi definido horário e local para a equipe participar do <i>workshop</i>	Não
Preparativos extras para a equipe	Definir as ferramentas necessárias para que a equipe se comunique de forma eficaz	A equipe se comunica e resolve problemas de forma eficaz com a utilização das ferramentas	Apenas se a equipe realiza reuniões presenciais

#### 2.3.4.3 Execução do *Kick-start Kanban*

Após entender o contexto da equipe e finalizar o planejamento, é hora de colocá-lo em prática. A Figura 7 apresenta as 14 etapas dentro do processo de execução do *Kick-start Kanban*.

<p><b>KICK 1: Set the Scene</b> Why a workshop? What is the goal? What will happen during the workshop? <i>Welcome   Why   Goals   About Kanban Kick-start   Golden Rule   Agenda</i></p>
<p><b>KICK 2: Share Current Concerns</b> What issues does the team have now? What issues can be resolved today? <i>Gather Inputs   Share   Categorize   Sort</i></p>
<p><b>KICK 3: Define a Shared Vision</b> Help the team discuss and agree on its Purpose. <i>Purpose   Slogan   Criteria for success   Team composition</i></p>
<p><b>KICK 4: Discover What the Team Does</b> Help the team understand what it actually does and for whom. <i>Upstream   Downstream   Responsibility   Workflow   Dependencies</i></p>
<p><b>KICK 5: Identify Work Types</b> Help the team understand its demand on how to categorize it. <i>Gather Data   Analyze</i></p>
<p><b>KICK 6: Explain How the Board Works</b> Help the team understand the mechanics of a Kanban board. <i>Work Item   Work Type   Work Flow   Pull   ...</i></p>
<p><b>KICK 7: Present a Common Language</b> Help the team understand and use a common language.</p>
<p><b>KICK 8: Set Visualization Policies</b> Help the team set policies on how to visualize work. <i>Work Types   Work Items   workflow   What not to visualize   When must be in Sync</i></p>
<p><b>KICK 9: Create &amp; Populate the Board</b> Help the team create and populate a board, according to the team's policies. <i>Draw workflow   Create work items   Populate board   Reflect   DoD   Pull   Blocks   Due Dates   Prio   DoR</i></p>
<p><b>KICK 10: Set Way-of-Working Policies</b> Help the team set policies governing the team's work. <i>What to work on   Side orders   When to pull   Part-times   DoD   DoR   Blocks   Due Dates   Standard work</i></p>
<p><b>KICK 11: Limit WIP</b> Help the team limit the work in progress. <i>Uncontrolled Flow   Value Finishing   Strategies for reducing WIP   Set explicit WIP limits</i></p>
<p><b>KICK 12: Set Planning Meeting Policies</b> Help the team set policies governing planning meetings. <i>The planning meeting   Meeting policies   Policies during the meeting   The after meeting</i></p>
<p><b>KICK 13: Run the First Planning Meeting</b> Help the team run its first planning meeting.</p>
<p><b>KICK 14: Close the Scene</b> Summarize what was done during the workshop and get feedback. <i>What happened   What is next   Loose ends   Feedback</i></p>

Figura 7 – Fase 3 do *Kick-start Kanban* (ACHOUANTZ; NORDIN, 2013)

A fase de execução do *workshop* possui uma série de etapas importantes. Essas etapas abrangem aspectos introdutórias, como apresentação do *workshop* e identificação de problemas atuais na equipe, até a definição de políticas de trabalho. A Tabela 5 apresenta de forma mais detalhada essas etapas.

Tabela 5 – Fase 3 - Atividades do *Kick-start Kanban* (ACHOUANTZ; NORDIN, 2013)

Atividade	Descrição	Definição de Concluído	Pode ser ignorado?
Definir o <i>workshop</i>	Apresentação do <i>workshop</i> e alinhamento de expectativas. Responder a perguntas como: "Por que um <i>workshop</i> ?"; "Qual é o objetivo?"; "O que acontecerá durante o <i>workshop</i> ?"	A equipe compreende o propósito do <i>workshop</i> e seus objetivos associados	Não
Identificar as preocupações atuais	Ajudar os membros da equipe a compartilharem suas preocupações atuais e identificarem aquelas que podem ser resolvidas durante o <i>workshop</i> . Responder a perguntas como: "Quais problemas a equipe tem agora?"; "Que problemas podem ser resolvidos hoje?"	A equipe compreende a situação atual e discute sobre ela; questões que podem (ou não) ser abordadas durante o <i>workshop</i> são definidas	Se você realizou recentemente uma retrospectiva sobre o assunto com a equipe
Definir uma Visão Compartilhada sobre o propósito da equipe	Ajudar a equipe a discutir e chegar a um acordo sobre seu Propósito	A equipe entende seu propósito	Somente se os membros da equipe têm um propósito claro e compartilhado
Descobrir o que a equipe faz	Ajudar a equipe a entender o que realmente faz e para quem	A situação atual da equipe é compreendida (o que a equipe realmente faz, em vez do que ela acha que faz)	Somente se a equipe já tem um bom entendimento sobre a situação/contexto

Identificar e agrupar os itens de trabalho com base em características comuns	Ajudar a equipe a entender sua demanda e como categorizá-la	A equipe categoriza as demandas e compreende se está trabalhando na coisa certas	Se a equipe já possui categorias de trabalho válidas ou se você deseja forçar a equipe a usar categorias predefinidas (certifique-se de que elas façam sentido no contexto da equipe)
Explicar como funciona o <i>Kanban</i>	Ajudar a equipe a entender a mecânica de um quadro <i>Kanban</i>	A equipe compreende como funciona a mecânica de um quadro <i>Kanban</i>	Se todos os membros da equipe já estão familiarizados com o <i>Kanban</i> .
Apresentar uma linguagem comum	Ajude a equipe a entender e usar uma linguagem comum	A equipe utiliza um vocabulário comum que é compartilhado por toda a unidade/departamento/empresa	Se a empresa não tem muitas equipes que interagem entre si ou possui um vocabulário comum
Definir políticas de visualização	Ajudar a equipe a definir políticas sobre como visualizar o trabalho	A equipe concorda com a lista de políticas que regulam a visualização do trabalho para entender o que está acontecendo e tomar decisões apropriadas	Se a equipe já possui políticas de visualização que não precisam ser revisadas
Criar e preencher o quadro	Ajudar a equipe a criar e preencher um quadro, de acordo com as políticas da equipe	Um quadro físico cheio de adesivos coloridos ou um quadro digital cheio de itens de trabalho	Se a equipe já possui um quadro onde é possível visualizar todos os itens de trabalho em andamento de acordo com as políticas atuais
Definir políticas que definem a maneira de trabalhar	Ajudar a equipe a definir políticas que regulam o trabalho da equipe	A equipe segue uma lista de políticas que regulam a forma como o trabalho é conduzido	Se a equipe já possui políticas de trabalho que não precisam ser revisadas

Limitar WIP	Ajudar a equipe a limitar o trabalho em andamento	A equipe possui políticas que definem o quanto de trabalho é ideal/permitido no quadro (por coluna)	Se a equipe já tem uma política de limite de WIP satisfatória ou se a equipe ainda não está madura o suficiente para sustentá-la
Definir políticas para reuniões de planejamento	Ajudar a equipe a definir políticas que regulam as reuniões de planejamento	Políticas que definem quando, como e onde as reuniões serão realizadas	Se a política atual de reuniões existe e é satisfatória
Realizar a primeira reunião de planejamento	Ajudar a equipe a realizar sua primeira reunião de planejamento	Todos os membros da equipe participam e entendem como a reunião de planejamento funciona	Se todos os membros da equipe já estão familiarizados com reuniões de planejamento (diárias)
Encerrar o <i>workshop</i>	Resuma o que foi feito durante o <i>workshop</i> e obtenha o <i>feedback</i>	O <i>workshop</i> é encerrado; a equipe é capaz de comandar o processo; coleta de <i>feedback</i>	Não (exceto se não houver mais tempo)

A Tabela apresentada resume as principais atividades da fase de execução do *workshop*. Essas etapas são cruciais para a compreensão e implementação do sistema *Kanban* em uma equipe. A Tabela também indica quando uma atividade pode ser ignorada, dependendo do contexto e do nível de maturidade da equipe. O objetivo das etapas de execução do *Kick-start Kanban* é capacitar a equipe a compreender questões importantes e melhorar seu processo de trabalho.

#### 2.3.4.4 Atingir a linha de base da melhoria contínua

Após a execução do *workshop*, o próximo passo é alcançar a chamada "linha de base da melhoria contínua", conforme já mencionado. Esse é o ponto onde a equipe compreende bem o sistema *Kanban* e a sua situação atual, permitindo que a equipe identifique áreas passíveis de melhoria e evolução. Portanto, esse é o momento em que a equipe se torna verdadeiramente capaz de melhorar continuamente seu processo e alcançar seus objetivos de forma mais eficaz.

A Figura 8 apresenta as nove etapas que buscam atingir a base de melhoria contínua.

<b>BOOST 1: Attend Planning Meetings</b> Help the team conduct planning meetings as intended.	<i>Observe and point-out problems</i>
<b>BOOST 2: Follow-up</b> Help the team adapt its board and policies to the new understanding.	<i>Build common understanding   Visualization policies   Working policies</i>
<b>BOOST 3: Understand Current Capability</b> Help the team understand its capabilities.	<i>CFD   Control Chart   Create understanding   Investigate</i>
<b>BOOST 4: Limit WIP</b> Help the team limit the work in progress.	<i>Limit WIP</i>
<b>BOOST 6: Set Policies for Demand Management</b> Help the team select the "right" work items to work on.	<i>Understand demand   Manage demand</i>
<b>BOOST 7: Assess Depth of Kanban</b> Help the team assess the depth of its Kanban implementation.	<i>Explain Kanban   Assess depth</i>
<b>BOOST 8: Set Next Target Condition</b> Help the team set a target condition.	<i>Identify direction   Identify target condition   Understand what is in the way   Remove obstacles</i>
<b>BOOST 9: Give Feedback</b>	<i>Contact us</i>

Figura 8 – Fase 4 do *Kick-start Kanban* (ACHOUANTZ; NORDIN, 2013)

A Figura 8 traz as etapas que devem ser realizadas e que estão por trás da base de melhoria contínua da equipe. Essas etapas apresentam aquilo que deve ser realizado pela equipe para que ela consiga melhorar e evoluir a sua condição atual. A Tabela 6 apresenta com mais detalhes essas etapas.

Tabela 6 – Fase 4 - Atividades do *Kick-start Kanban*

Atividade	Descrição	Definição de Concluído	Pode ser ignorado?
Ajustar reuniões de planejamento	Ajudar a equipe a conduzir as reuniões conforme planejado	As reuniões de planejamento são conduzidas da forma correta, seguindo as políticas de reuniões que foram definidas	Apenas se o Gerente de Fluxo já possui grande experiência
Adaptar as políticas ao entendimento atual da equipe	Ajudar a equipe a adaptar seu quadro e políticas ao novo entendimento que ela agora tem sobre o que faz, como faz e para quem faz	O quadro <i>Kanban</i> e as políticas da equipe são adaptados à compreensão da equipe em relação a sua condição atual	Não
Entender a capacidade atual da equipe	Ajudar a equipe, e principalmente o gestor de fluxo, a medir o fluxo de trabalho	A equipe sabe sua capacidade e consegue estipular prazos de entrega	Apenas se a equipe já produz CFD (Cumulative Flow Diagram) e CC (Control Charts)
Limitar a WIP (Work in Progress)	Ajudar a equipe a limitar o trabalho em andamento com base na sua capacidade	A equipe possui um limite de WIP ideal (que não gere estresse sob a equipe)	Apenas se a equipe já tiver limites de WIP adequados
Aumentar o fluxo de trabalho (velocidade e eficiência)	Ajudar a equipe a aumentar o fluxo de trabalho	O fluxo de trabalho da equipe é adequado	Se o fluxo de conclusão das atividades estiver eficiente e rápido
Gerenciar as demandas	Ajudar a equipe a criar políticas para gerenciar a <i>Ready Queue</i>	São definidas políticas para o gerenciamento da <i>Ready Queue</i>	Se a equipe já tiver uma política de gerenciamento da <i>Ready Queue</i>
Avaliar a "profundidade" do <i>Kanban</i> da equipe	Ajudar a equipe a avaliar a "profundidade" de seu sistema <i>Kanban</i> e sugerir melhorias	A equipe tem uma melhor compreensão do <i>Kanban</i> ; possui uma ideia sobre sua condição atual; e algumas informações sobre o que melhorar em seu sistema <i>Kanban</i>	Não
Definir a próxima etapa	Ajudar a equipe a definir uma condição-alvo para impulsionar melhorias	A equipe sabe qual problema resolver para melhorar sua capacidade, de acordo com seus objetivos e desafios	Não
<i>Feedback</i> sobre o guia de campo	Proporcionar <i>feedback</i> sobre o <i>workshop</i> para melhorá-lo.	O <i>feedback</i> possui considerações a fins de melhorá-lo.	Não.

A Tabela apresentada resume as etapas da Fase 4 do *Kick-start Kanban*, que se concentra na otimização do sistema *Kanban* da equipe. Essas etapas visam o aperfeiçoamento do uso do *Kanban* para atingir a base de melhoria contínua, tornando o fluxo de

trabalho mais eficiente e eficaz, além de levar a equipe a um estado de consciência sobre sua condição atual e buscando mantê-lo.

## 2.4 *Startups* e o emprego de Metodologias Ágeis

Para Hart (2011), uma *startup* é uma instituição humana projetada para criar um novo produto/serviço em condições de incerteza. Na mesma linha, Blank (2012) define uma *startup* como uma entidade que visa um ambiente escalável, repetível e que tenha um modelo de negócios rentável e, assim, almeja o crescimento.

As *startups* de desenvolvimento de software representam um fenômeno curioso, já que operam no ambiente desafiador de criar produtos inovadores que dependem amplamente de software, tudo isso enquanto lidam com restrições de tempo e recursos limitados (PATERNOSTER et al., 2014). Além disso, buscam constantemente modelos de negócios sustentáveis e escaláveis e nos últimos anos, essas *startups* têm atraído cada vez mais interesse de pesquisa na comunidade de Engenharia de Software, conforme ressaltado pelos pesquisadores Unterkalmsteiner et al. (2016).

Com o movimento das *startups*, a indústria de software está passando por uma transformação de paradigma, evoluindo de um foco estrito na satisfação dos requisitos do cliente para uma ênfase maior na *criação de valor* para o cliente (NGUYEN-DUC; WANG; ABRAHAMSSON, 2017). O desafio para as empresas de software já não reside primordialmente na execução dos requisitos do cliente, mas sim em *identificar as necessidades dos clientes* e oferecer soluções que realmente acrescentem *valor* a essas necessidades (BLANK, 2020).

Uma das movimentações que vem sendo observada é a mesclagem de metodologias ágeis com UCD (SIGNORETTI et al., 2019; ZORZETTI et al., 2021). Para Signoretto et al. (2019), isso fortalece o engajamento das partes interessadas e dos usuários finais, enquanto viabiliza experimentações ágeis e oferece uma série de outros benefícios. E para assegurar que o software não apenas corresponda às necessidades do negócio, mas também às dos usuários, a aplicação do UCD capacita os desenvolvedores a compreenderem as reais necessidades dos usuários (SALAH; PAIGE; CAIRNS, 2015).

## 2.5 Design Centrado no Usuário

O conceito de Design Centrado no Usuário (do inglês, *User Centered Design* - UCD), foi originalmente introduzido por Don Norman, um especialista em ciência cognitiva e engenharia de usabilidade. O UCD se popularizou, em especial, devido a trabalhos como *User Centered System Design: New Perspectives on Human-Computer Interaction* (NORMAN; DRAPER, 1986) e *The design of everyday things* (NORMAN, 1988).

Diariamente as pessoas interagem com produtos e serviços variados, passando por uma série de experiências envolvendo sensações, pensamentos e emoções (positivas, negativas ou indiferentes) (VENSON et al., 2022). Para Venson et al. (2022), o resultado dessa experiência é um fator crucial para dizer se os usuários voltarão ou não a utilizar e experimentar esses produtos e serviços. Portanto, o UCD pode garantir uma boa e positiva experiência para o usuário, já que este processo estabelece e define uma maneira de desenvolver produtos que sejam realmente úteis, que funcionem bem e sejam fáceis de utilizar pelo usuário final (ANITHA; PRABHU, 2012).

O UCD pode ser entendido como uma série de processos iterativos que priorizam a incorporação das necessidades dos usuários em todas as fases do design e do desenvolvimento de produtos e de serviços (NORMAN, 2013). Essa série de processos tem como principal objetivo adquirir uma compreensão aprofundada dos futuros usuários do produto ou serviço (NOVOSELTSEVA, 2017), melhorando assim as suas experiências. Para isso, um conjunto de métodos e técnicas de pesquisa e de design tem sido empregadas, buscando mapear as necessidades dos usuários de produtos e de serviços (NOGUERRA, 2023).

Para Noguerra (2023), as metodologias centradas no usuário desempenham um papel essencial no processo de *design* e também no desenvolvimento, com ferramentas como o teste de usabilidade, que permite aos usuários interagir com protótipos a fim de identificar e solucionar problemas (QUINTANA; HOWELLS; HETTINGER, 2007), além do desenvolvimento de personas, que possibilita criar perfis detalhados de usuários para direcionar as decisões de *design* de forma mais precisa (SALMINEN et al., 2022).

Nesse contexto de metodologias e padrões, a ISO 13407 (1999), que é uma norma voltada para processos de design centrados no ser humano para sistemas interativos, não define um padrão de projeto de *design*. Mas apresenta alguns princípios que podem ser implementados em qualquer processo existente a partir da perspectiva do usuário:

- **O envolvimento ativo dos usuários e uma compreensão clara dos requisitos dos usuários e das tarefas** - O envolvimento dos usuários no processo de desenvolvimento é crucial para obter informações valiosas sobre o contexto de uso, as tarefas e a forma como os usuários interagirão com o produto ou sistema futuro. A eficácia desse envolvimento cresce com a intensidade da interação entre desenvolvedores e usuários, sendo adaptada conforme as atividades de design em questão;
- **Alocação adequada de funções entre usuários e tecnologia** - Na tomada de decisões, é essencial ponderar diversos fatores, tais como as capacidades e limitações relativas entre seres humanos e tecnologia, abrangendo questões de confiabilidade, velocidade, precisão, força, flexibilidade de resposta, custos financeiros e a impor-

tância da realização eficaz e oportuna de tarefas, bem como o bem-estar do usuário. Não se deve limitar a análise à mera identificação das funções que a tecnologia pode desempenhar. As funções atribuídas aos seres humanos devem ser cuidadosamente planejadas e compor um conjunto significativo de tarefas. É recomendável envolver usuários representativos nesse processo de decisão;

- **Processo de *design* iterativo** - Nas abordagens de design iterativo, o *feedback* dos usuários desempenha um papel crucial, tornando-se uma fonte fundamental de informações. A iteração, quando combinada com o envolvimento ativo do usuário, representa um método eficaz para reduzir o risco de um sistema não atender aos requisitos do usuário e da organização, inclusive aqueles que são complexos ou difíceis de especificar de maneira explícita. Através da iteração, soluções de projeto preliminares são testadas em cenários do mundo real, e os resultados são incorporados em soluções progressivamente refinadas;
- ***Design* multidisciplinar** - O design centrado no ser humano demanda diversas habilidades e envolve uma variedade de profissionais para lidar com os aspectos humanos do design. Isso implica que equipes multidisciplinares devem participar ativamente no processo de design centrado no usuário. Essas equipes podem ser de tamanho reduzido, altamente dinâmicas e com duração limitada ao ciclo de vida do projeto. A composição das equipes deve ser alinhada com o relacionamento entre a organização responsável pelo desenvolvimento técnico e o cliente.

Esses princípios mostram que o UCD se concentra nas necessidades do usuário, envolvendo o consumidor em todo o processo de desenvolvimento. Além disso, o UCD auxilia uma mudança valorizada pelo aprendizado na prática. Ao considerar constantemente as necessidades do usuário, ele incentiva a adaptação e melhoria contínuas, tornando-o uma abordagem dinâmica e eficaz para o desenvolvimento de produtos e serviços.

### 2.5.1 Atividades do UCD

De acordo com [Norman \(2013\)](#), as atividades que compõem o processo UCD são divididas em:

1. **Observação** - Nessa etapa, o foco está na compreensão fundamental do problema em questão. Os membros da equipe de *design* concentram seus esforços em conhecer os usuários do produto em desenvolvimento. Isso é alcançado ao se aproximar dos potenciais clientes, observando suas atividades diárias e buscando um entendimento profundo de seus interesses, motivações e necessidades. A definição do problema de design surge a partir dessa compreensão abrangente dos objetivos que as pessoas buscam alcançar e das barreiras que encontram em suas jornadas.

2. **Geração de ideias (ideação)** - Essa etapa se inicia após a compreensão do problema que será tratado e pode ser definido como processo criativo de geração, desenvolvimento e exploração de ideias. A ideação é frequentemente conduzida por meio de técnicas como o *brainstorming*, que estimula a criatividade e a inovação (KUREK et al., 2023).
  - *Brainstorming* - É uma técnica que, por meio do compartilhamento espontâneo e criativo de ideias, possibilita encontrar soluções para um problema. É recomendado que seja realizada em uma reunião, com pessoas ativas na empresa e com perspectivas diferentes. O *brainstorming* deve ser livre de críticas, pois mesmo as ideias que parecem ineficazes devem ser analisadas, afinal, elas podem ser o ponto inicial para construção de discussões mais profundas.
3. **Prototipagem** - Esta etapa do processo possibilita que a equipe de *design* implemente ideias coletadas em fases anteriores por meio de protótipos, ocasionando um melhor entendimento das necessidades do usuário e garantindo que as lacunas levantadas foram compreendidas. Esta parte do processo pode ser feita em etapas, construindo a partir de um conceito geral para um design completo (GUIMARÃES, 2019).
4. **Teste** - Os protótipos criados devem ser testados e avaliados. Esta etapa visa realizar uma avaliação integral do produto por meio de testes de usabilidade com grupos de usuários reais. Fundamentado em técnicas de avaliação heurística, enquetes, observação, casos de uso, entre outras, essa etapa coloca os usuários em confronto com as interfaces desenvolvidas. Os testes possibilitam entender quais mudanças devem ser realizadas para melhorar a experiência do usuário, a fim de criar um produto final com a menor quantidade de imperfeições possíveis. À medida que um grupo maior de usuários começa a usar o modelo, necessidades, desafios novos ou imprevistos podem surgir e outras mudanças ou ajustes devem ser feitos.

Estas quatro atividades são iteradas, ou seja, elas são repetidas continuamente com cada ciclo produzindo mais *insights* e aproximando-se da solução desejada.

A Figura 9 elucidada a iteração sobre essas atividades. No primeiro estágio está a *observação*, que sugere a importância da análise cuidadosa e da compreensão profunda do problema ou contexto em questão. Em seguida, o *design*, evidenciando a fase de concepção e planejamento, onde ideias e soluções começam a ser moldadas. A terceira atividade, representada pela *prototipagem*, aponta para a materialização e experimentação das ideias concebidas, possibilitando a criação de modelos iniciais para testes. Por fim, a atividade de *testes* surge como o desfecho, indicando a avaliação criteriosa dos protótipos desenvolvidos, a fim de validar, aprimorar e refinar as soluções propostas.

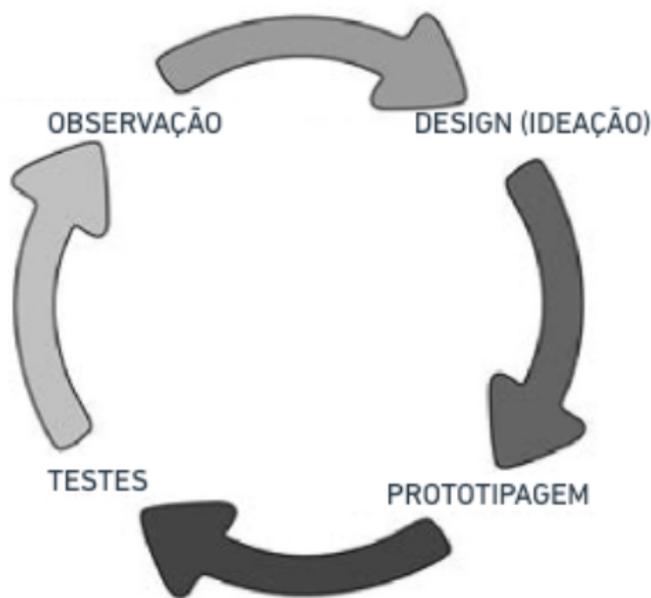


Figura 9 – Etapas do UCD (NORMAN, 2013)

### 2.5.2 Técnicas UCD

Muitos sistemas não apresentam um nível adequado de usabilidade, embora esse possa ser um dos atributos de qualidade mais relevantes nos produtos de software, juntamente com a UX, chegando a ser considerados fatores-chave para o sucesso de um produto de software (QUIÑONES; RUSU, 2019). De acordo com Salinas, Cueva e Paz (2020), é por esse motivo que opta-se por redesenhar esses sistemas usando técnicas que proporcionam resultados positivos nas avaliações de usabilidade. Aguirre, Moquillaza e Paz (2019) ressalta que, tanto a usabilidade quanto a experiência do usuário positiva podem ser alcançadas caso um processo de design centrado no usuário seja seguido durante o desenvolvimento do produto de software.

De acordo com os resultados obtidos da revisão sistemática da literatura realizada por Salinas, Cueva e Paz (2020), foi identificado um número considerável de técnicas usadas para trabalhar seguindo a metodologia UCD, onde as mais relatadas são a prototipagem, a avaliação de usabilidade das interfaces gráficas originais, a identificação de partes interessadas/usuários finais, entrevistas e grupos focais. Outras técnicas também foram encontradas e são apresentadas na tabela 7.

Tabela 7 – Técnicas UCD relatadas na revisão sistemática - Adaptado (SALINAS; CUEVA; PAZ, 2020)

ID	Técnica	Número de Artigos
T01	Prototipagem	12
T02	Avaliação de Usabilidade de Interfaces Gráficas Originais	9
T03	Identificação de Stakeholders/Usuários Finais	6
T04	Entrevistas	5
T05	Grupo Focal	4
T06	<i>Brainstorming</i>	3
T07	Reuniões/Reuniões Face a Face	3
T08	Heurísticas de Usabilidade de Jakob Nielsen/Heurísticas de Pierotti	3
T09	Análise de Tarefas do Usuário	3
T10	<i>Storyboarding</i>	2
T11	<i>Personas</i>	2
T12	Perfis de Usuário	2
T13	Análise de Concorrentes	2
T14	Especificação de Requisitos/Estabelecimento de Requisitos	2
T15	Pesquisas/Questionários	2
T16	Princípios de Fatores Humanos	2
T17	Card Sorting	1
T18	Revisão/Literatura	1
T19	Tag Cloud	1
T20	Análise de Dados	1
T21	Pensamento Visual	1
T22	Observações Não Participativas	1
T23	Mapas de Empatia	1
T24	Histórias de Usuário	1
T25	Diagramas de Caso de Uso	1
T26	Diagramas de Sequência	1
T27	Princípios de Interação Humano-Computador	1
T28	Método TRIZ	1
T29	Design de Layout	1
T30	Princípios de Design de Interação	1

A prototipagem foi a técnica mais relatada nos artigos selecionados (um total de 12 artigos) e é frequentemente usada para criar versões iniciais de sistemas para esboçar um modelo pretendido do produto final, que pode ser mais simples (baixa fidelidade) ou mais próximo da realidade (alta fidelidade) (NELSON et al., 2016). A segunda técnica mais comum é a avaliação de usabilidade de interfaces gráficas originais, aplicada em nove artigos para identificar e corrigir problemas. A terceira técnica mais utilizada é a identificação de partes interessadas/usuários finais, relatada em seis artigos. Entrevistas, a quarta técnica, foram usadas para entender necessidades e requisitos em dois artigos. Grupos focais, reunindo usuários para discutir experiências, foram aplicados em um estudo para entender necessidades e sugestões em relação à usabilidade de aplicativos de aluguel.

As demais técnicas aparecem com menos frequência, embora algumas sejam comumente conhecidas.

### 2.5.3 Análise comparativa entre as técnicas UCD

Conforme observado na seção anterior, existem muitas técnicas UCD presentes na literatura. Essas técnicas são capazes de auxiliar nas atividades UCD, ajudando na priorização da incorporação das necessidades dos usuários em todas as fases do design e do desenvolvimento de produtos e de serviços (NORMAN, 2013). Para uma visualização minúsciosa de algumas dessas técnicas UCD, as tabelas 8, 9 e 10 contam com uma análise realizada por Argumanis, Moquillaza e Paz (2021)

Na Tabela 8 é feita uma análise comparativa entre Protótipo de papel, Personas, Sprint 0, Design em pares e Questionário, adaptada de Argumanis, Moquillaza e Paz (2021). Ela compara várias características de cada técnica, como vantagens, desvantagens, investimento necessário, estágio do projeto em que é mais adequada e em que tipo de projeto é recomendada ou não.

Tabela 8 – Análise comparativa entre as técnicas UCD I (ARGUMANIS; MOQUILLAZA; PAZ, 2021) - Adaptado

	Protótipo de papel	Personas	Sprint 0	Design em pares	Questionário
<b>Vantagem</b>	Proposta rápida e econômica	Fornecer um claro conhecimento sobre as necessidades dos usuários	Fornecer um espaço de tempo para que haja investigação do usuário e uma elaboração global sobre a visão do produto	Fornecer uma clara ideia sobre as restrições técnicas para os designers	Eficiência em coletar uma grande quantidade de dados; atinge um amplo público, reduzindo custos; padronização e uniformidade; proporciona anonimato, liberdade e a segurança das respostas; minimiza distorções e influências externas (OLIVEIRA et al., 2016)
<b>Desvantagem</b>	Designers são muito abstratos	Não são necessários por todos os tipos de projetos	Perca de tempo se as decisões não forem implementadas	Não aumenta a performance em tarefas pequenas	Pode ter baixa taxa de retorno e perguntas sem respostas; pouca margem para tirar dúvidas; uma pergunta pode influenciar outra; não é possível ter controle sobre quem responde (OLIVEIRA et al., 2016)
<b>Investimento</b>	Baixo	Baixo	Médio	Baixo	Baixo (OLIVEIRA et al., 2016)
<b>Estágio do Projeto</b>	Iterativo	Preliminar	Preliminar	Iterativo	Iterativo
<b>Onde usar</b>	Qualquer projeto	Projetos com usuários desconhecidos	Projetos com pequena visão inicial	Projetos complexos	Qualquer projeto
<b>Onde não usar</b>	Não se aplica	Projetos com usuários conhecidos	Projetos com alta visão inicial	Projetos pequenos	Não se aplica

A Tabela 9 segue a mesma linha da anterior e apresenta uma análise comparativa entre quatro técnicas de Design Centrado no Usuário (UCD): *Card sorting*, Avaliação Heurística, *Big Design Upfront* e Investigação Contextual.

Tabela 9 – Análise comparativa entre as técnicas UCD II (ARGUMANIS; MOQUIL-LAZA; PAZ, 2021)

	<i>Card sorting</i>	Avaliação heurística	<i>Big design upfront</i>	Investigação contextual
<b>Vantagem</b>	Provê conhecimento sobre o modo de pensar do usuário	Ajuda a identificar a maioria dos problemas de design	Facilita a sincronização do projeto	Fornece conhecimento sobre o contexto dos usuários
<b>Desvantagem</b>	É difícil fazer a ordenação de muitos cartões	Alto custo	Mudanças possuem alto custo	Muita dependência de terceiros
<b>Investimento</b>	Alto	Alto	Médio	Médio
<b>Estágio do Projeto</b>	Preliminar	Final	Preliminar	Preliminar
<b>Onde usar</b>	Projetos envolvendo muitas informações estruturadas	Redesign de projetos ou melhorias	Projetos com grande visão inicial	Qualquer projeto
<b>Onde não usar</b>	Projetos envolvendo poucas informações estruturadas	Projetos totalmente novos	Projetos com pouca visão inicial	-

A seguir, a terceira tabela de análise comparativa, também entre quatro técnicas de Design Centrado no Usuário (UCD): Design conduzido por desenvolvedores, *Thinking aloud*, *Scenarios* e Mapa da Jornada do Cliente.

Tabela 10 – Análise comparativa entre as técnicas UCD III (ARGUMANIS; MOQUIL-LAZA; PAZ, 2021)

	Design conduzido por desenvolvedores	<i>Thinking aloud</i>	<i>Scenarios</i>	Mapa da Jornada do Cliente
<b>Vantagem</b>	Diminui a sobrecarga de trabalho dos designers	Fornece uma ideia precisa de UX sobre o produto	Definição mais fácil sobre a visão estruturada do projeto	Fornece conhecimento sobre os processos do usuário para atingir seus objetivos
<b>Desvantagem</b>	É difícil fazer a ordenação de muitos cartões	Alto custo	Mudanças possuem alto custo	Muita dependência de terceiros
<b>Investimento</b>	Alto	Alto	Médio	Médio
<b>Estágio do Projeto</b>	Preliminar	Final	Preliminar	Preliminar
<b>Onde usar</b>	Projetos envolvendo muitas informações estruturadas	Redesign de projetos ou melhorias	Projetos com grande visão inicial	Qualquer projeto
<b>Onde não usar</b>	Projetos envolvendo poucas informações estruturadas	Projetos totalmente novos	Projetos com pouca visão inicial	-

#### 2.5.4 Design Centrado no Usuário e *Kanban*

Para estabelecer um processo de *design* focado nas necessidades do usuário, é fundamental considerar atividades conceituais, como a análise do contexto de utilização, a definição dos requisitos dos utilizadores e a criação de modelos iniciais (SCHÖN et al., 2016). Outro aspecto importante é a avaliação destes modelos para verificar se as soluções de *design* estão alinhadas com os requisitos (UHLENBROK et al., 2015). A partir disso, o objetivo é aprimorar a organização de um processo de desenvolvimento ágil com o uso do método *Kanban*, tornando-o mais orientado para as necessidades do usuário.

Winter et al. (2013) recomendam quatro elementos importantes que otimizam o processo de desenvolvimento:

- **Adição de um quadro de tarefas de design** - O quadro *Kanban* é usado para visualizar o fluxo de trabalho da equipe de desenvolvimento do ponto de vista técnico, e na maioria das vezes, não estão presentes atividades relacionadas ao design centrado no usuário. De acordo com Schön et al. (2016) é comum observar a ausência de tarefas conceituais, como pesquisa de usuários, especificação de requisitos de usuário e testes de usabilidade. Para isso, os autores propõem a adição de um quadro *kanban* para o design antes do quadro de desenvolvimento, este último permanecendo com o mesmo conteúdo como é mostrado na Figura 10. Assim, as tarefas conceituais podem ser integradas ao fluxo de trabalho de maneira organizada, seguindo uma estrutura semelhante à das tarefas de desenvolvimento. Isso representa uma aplicação inicial das práticas fundamentais do *Kanban*, trazendo maior transparência ao fluxo de trabalho global.
- **Colaboração multifuncional** - Um dos valores fundamentais no desenvolvimento ágil é a valorização da interação entre as pessoas (BECK et al., 2001). Para evitar a criação de silos organizacionais, onde especialistas de um domínio específico, como especialistas em UX ou controle de qualidade, trabalham isoladamente, é essencial promover uma colaboração intensa em uma equipe multifuncional. Essa abordagem estimula a sinergia entre os membros da equipe e facilita uma comunicação eficaz (SCHÖN et al., 2016). Assim como na reunião diária do *Scrum*, a equipe pode conduzir uma reunião de alinhamento diária, com o propósito de coordenar o trabalho entre os membros da equipe, incluindo desenvolvedores, especialistas em UX, testadores e gerentes de projeto. Isso promove a sincronização das atividades e uma comunicação eficiente.
- **Avaliação da entrega** - Aproveitando o contexto criado na introdução da subseção anterior, um ponto significativo no UCD é a avaliação da UX a cada iteração ISO 13407 (1999), o *feedback*. Para este propósito, é introduzido uma limitação semelhante ao limite de WIP para a coluna “pronto”, como está representado na Figura 10. Atingir o limite de WIP implica iniciar a avaliação da liberação. Isto tem a vantagem de que a complexidade relativa ao planejamento será reduzida e a avaliação será realizada continuamente. Descobertas significativas podem fluir para o processo como novas tarefas, priorizadas no quadro de design. Alcançar o limite de WIP sinaliza o início da avaliação. Essa abordagem oferece a vantagem de reduzir a complexidade no planejamento, permitindo que a avaliação seja realizada de forma contínua (SCHÖN et al., 2016). Descobertas significativas podem ser integradas ao processo como novas tarefas, sendo priorizadas no quadro de design.

- **Uso de artefatos UX** - Para compartilhar uma visão comum entre a equipe do projeto, são necessários artefatos de requisitos. Esses artefatos devem ser compreensíveis pela equipe do projeto. Abaixo são citados alguns deles:
  - **Personas** - A criação de Personas é uma estratégia eficaz para compreender as necessidades dos usuários, priorizar recursos e funcionalidades, e orientar o design de produtos e serviços digitais (NORMAN, 2013). Personas são derivadas a partir de perfis reais de potenciais usuários para representar perfis típicos. Elas auxiliam as partes interessadas a desenvolver um entendimento compartilhado das necessidades dos usuários.
  - **Histórias de usuários** - As histórias de usuários são um método difundido para engenharia de requisitos em ASD. Eles são usados para definir o escopo (LUCASSEN et al., 2016). Nesse sentido, Persona Stories são um tipo especial de *user stories* em que o papel é preenchido por uma persona (WINTER et al., 2013). Isto traz a vantagem de que os benefícios resultantes da utilização de personas estão mais integrados ao processo de desenvolvimento, como afirmam (SCHÖN et al., 2016).
  - **Protótipos** - Permitem a visualização de conexões complexas entre requisitos. Eles capacitam os membros do projeto a aprender, explorar, gerar e aprimorar ideias. Além disso, os protótipos possibilitam a avaliação se o modelo conceitual do produto está alinhado com as suposições relativas ao modelo mental do usuário (SCHÖN et al., 2016).

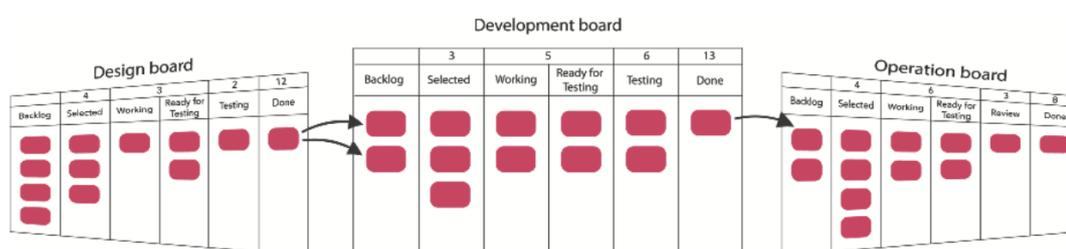


Figura 10 – Quadro Kanban para design, desenvolvimento e operação (SCHÖN et al., 2016)

Na Figura 10 existem três quadros *Kanban* que representam o fluxo de trabalho, abrangendo desde a fase de design até o desenvolvimento e a entrega do produto. Uma tarefa inicialmente apresentada no quadro de design pode ser subdividida em várias tarefas no quadro de desenvolvimento ou operações, visando assegurar um fluxo contínuo tanto dentro de cada quadro quanto entre eles. O fluxo é representado pelas seguintes fases:

- **Lista de pendências (*Backlog*)** - Nesta coluna, são listadas todas as tarefas ou itens que precisam ser feitos, mas ainda não foram priorizados ou agendados. O *backlog*

é uma espécie de repositório de todas as demandas que podem ser incluídas no projeto, e é a partir daqui que as tarefas são selecionadas para serem trabalhadas em uma determinada iteração ou *sprint*;

- Selecionado (*Selected*) - Após priorizar as tarefas do backlog, aquelas que foram escolhidas para serem trabalhadas em um determinado período são movidas para a coluna "Selecionado". Essa coluna representa as tarefas que a equipe se compromete a realizar durante um ciclo de trabalho específico;
- Trabalhando (*Working*) - Quando um membro da equipe começa a trabalhar em uma tarefa, ela é movida para a coluna "Trabalhando". Isso indica que a tarefa está em andamento e alguém está dedicando tempo e esforço para completá-la;
- Pronto para Teste (*Ready for Testing*) - Após a conclusão da implementação de uma tarefa, ela é movida para a coluna "Pronto para Teste". Nesta etapa, a tarefa está pronta para ser avaliada e testada para garantir que atenda aos critérios de aceitação;
- Testando (*Testing*) - Aqui, as tarefas estão sendo testadas quanto à qualidade e funcionalidade. Os testes podem incluir testes de unidade, testes de integração, testes de aceitação, entre outros, dependendo da metodologia de desenvolvimento;
- Pronto (*Done*) - Quando uma tarefa passa com sucesso pelos testes e está pronta para ser entregue ao cliente ou implantada no ambiente de produção, ela é movida para a coluna "Pronto". Isso indica que a tarefa foi concluída com sucesso e está pronta para o próximo passo no processo, como entrega ou implantação.

## 2.6 Considerações Finais do Capítulo

Neste capítulo apresentou-se o objeto de estudo deste trabalho. No capítulo seguinte apresenta-se o Plano Metodológico adotado neste trabalho.

## 3 Materiais e Métodos

### 3.1 Considerações Iniciais do Capítulo

Neste capítulo apresenta-se o Plano Metodológico adotado, conforme classificação da metodologia de pesquisa, apresentada no Capítulo 1. Em seguida, o planejamento das 4 fases do plano: Planejamento; Coleta de Dados; Análise de Dados; e Relatórios, compostas pelos itens do Protocolo de Estudo de Caso.

#### 3.1.1 Plano Metodológico

O plano metodológico compreende quatro grandes fases: Planejamento; Coleta de Dados; Análise de Dados; e Relatórios (BRERETON et al., 2008).

Essas fases compreendem, de forma resumida:

- **Planejamento da Pesquisa:** em que são apresentados o contexto do pesquisa, breve revisão bibliográfica, a pergunta norteadora desta pesquisa, a definição dos objetivos (geral e específicos) e a definição de um plano metodológico, com as escolhas metodológicas;
- **Coleta dos Dados:** são apresentados o levantamento e a aplicação das técnicas de coleta de dados como: revisão documental; revisão bibliográfica e; a técnica de estudo de caso;
- **Análise dos Dados:** são apresentadas a interpretação e análise dos dados coletados, assim como a análise da validade do trabalho, em que se buscará a avaliação da percepção dos gestores da empresa em relação as diretrizes propostas;
- **Relatório:** essa fase caracteriza a materialização deste trabalho de monografia de TCC, apresentando a escrita das 4 grandes fases desta pesquisa.

### 3.2 Planejamento da Pesquisa

Nesta fase de planejamento, conforme apresentado no Capítulo 1, foi realizada a escolha do tema de pesquisa, a questão de pesquisa, e os objetivos, geral e específicos. Para alcançar o objetivo, foi definido um Plano Metodológico de Pesquisa.

A técnica selecionada foi a de Estudo de Caso, uma vez que este trabalho visa examinar a aplicação do *Kick-start Kanban* e de abordagens UCD no contexto real do projeto *Groovoo* da *startup Atena*.

Para execução do Estudo de Caso foi definido a utilização do Protocolo proposto por [Brereton et al. \(2008\)](#). O Protocolo possui 11 itens, e a partir deles, os pesquisadores se orientam no planejamento e na execução da técnica de estudo de caso.

A Tabela 11 apresenta as etapas do Protocolo de [Brereton et al. \(2008\)](#), e a descrição de cada uma delas.

Etapa	Descrição
Background	a) Localizar estudos anteriores relacionados ao tópico b) Estabelecer a principal pergunta de pesquisa que será abordada neste estudo c) Identificar quaisquer perguntas de pesquisa adicionais que serão exploradas
Design	a) Determinar a escolha entre estudos de caso único ou múltiplos b) Detalhar o objeto de estudo, como um novo procedimento de teste ou um recurso recém-introduzido c) Identificar quaisquer proposições ou subquestões que se originem de cada questão de pesquisa
Seleção de Caso	a) Critério para seleção de caso
Procedimentos e papéis do estudo	a) Normas que regulam os processos de campo b) Responsabilidades dos membros da equipe de pesquisa do estudo de caso
Coleta de Dados	a) Identificação dos dados a serem coletados b) Estabelecimento de um plano de coleta de dados c) Definição de como será o armazenamento dos dados
Análise	a) Estabelecimento dos critérios para interpretar os resultados do estudo de caso b) Estabelecer quais dados serão utilizados para responder a questão da pesquisa c) Considerar as diversas possibilidades de resultados e identificar explicações alternativas d) A análise será realizada à medida que o estudo de caso avança
Validade do plano	a) Verificação geral para o projeto e o plano de coleta de dados b) Comprovação da validade interna, demonstrando a relação causal entre os resultados e a intervenção/tratamento c) Definição da validade externa, que identifica o âmbito de generalização dos resultados do estudo d) Garantia da validade de construto, mostrando que as medidas operacionais adequadas estão planejadas para os conceitos sob estudo
Limitações do Estudo	a) Especificação das questões de validade residual, incluindo potenciais conflitos de interesse que sejam inerentes ao problema, não decorrentes do plano
Relatórios	a) Identificação do público-alvo e a relação com estudos mais amplos
Cronograma	a) Fornecer estimativas de tempo para todas as etapas principais: planejamento, coleta de dados, análise de dados, geração de relatórios
Apêndices	a) Atualização contínua durante a condução do estudo, destacando quaisquer desvios em relação às etapas mencionadas anteriormente

Tabela 11 – Protocolo de Estudo de Caso ([BRERETON et al., 2008](#))

### 3.2.1 *Background*

Na etapa de *Background* os pesquisadores exploraram a Pesquisa Bibliográfica e a Pesquisa Documental relacionadas ao tópico que investigado. Isso envolveu uma revisão da literatura para entender o contexto, as descobertas e as lacunas já identificadas no estado da arte. Nessa etapa a principal pergunta de pesquisa foi formulada, assim como a identificação de quaisquer outras questões adicionais que pudessem surgir a partir da revisão da literatura.

### 3.2.2 *Design*

O *Design* deste estudo de caso é a escolha de um caso único. Isso é, uma *startup* e um projeto de desenvolvimento de software dessa *startup*.

No caso, o projeto *Groovoo* foi selecionado para uma análise mais detalhada, no qual será estudada a indicação do *Kick-start Kanban*. O objetivo é definir diretrizes de Design Centrado no Usuário (UCD) para o processo de desenvolvimento de software do projeto *Groovoo* na *startup Atena*.

### 3.2.3 Seleção de Caso

Quanto a seleção de caso, ela é definida pelo projeto *Groovoo* da *Startup Atena Solutions*, descrita no Capítulo 4. Os principais motivos para a seleção do caso são:

1. Os donos da empresa estão dispostos a melhorar o processo de desenvolvimento de seus projetos;
2. Um trabalho de inserção de UCD proposto anteriormente não foi implementado;

### 3.2.4 Procedimentos e Papéis do estudo de caso

Nessa seção são definidos os procedimentos a serem utilizados para conduzir o estudo de caso, assim como as funções e responsabilidades de cada membro da equipe de pesquisa.

Quanto aos papéis, os autores pesquisadores deste trabalho têm vínculo com a *Startup*. Um como desenvolvedor do projeto *Groovoo* e outro, líder técnico de outro projeto da empresa.

Quanto aos procedimentos deste trabalho foram planejadas três atividades. Inicialmente, será realizada uma análise para selecionar abordagens de UCD a serem aplicadas no Workshop Kick-start Kanban, visando aprimorar o processo de desenvolvimento de software Scrumban Groovoo. Em seguida, será proposta a execução do Workshop Kick-start Kanban, integrando as abordagens de UCD selecionadas à parte Kanban do processo de desenvolvimento de software. Por fim, será feita a validação da proposta do Workshop Kick-start Kanban com as abordagens UCD selecionadas para o Processo de desenvolvimento de Software do Groovoo.

### 3.2.5 Coleta de Dados

A coleta de dados para esta monografia engloba tanto a pesquisa bibliográfica quanto a pesquisa documental.

O objetivo é descrever as atividades 1 e 2 como relatórios, no qual são apontadas a proposta do *Workshop Kick-start Kanban com as Abordagens UCD selecionadas* para o *Processo de desenvolvimento de Software do Groovoo* e em que parte do processo de desenvolvimento do projeto *Groovoo* poderiam ser inseridas.

### 3.2.6 Análise

A partir da coleta, o objetivo principal da análise é tirar conclusões sólidas e embasadas a partir das informações coletadas, mantendo uma cadeia clara de evidências que sustente as conclusões alcançadas.

O objetivo é que a análise da proposta do *Workshop Kick-start Kanban com as Abordagens UCD selecionadas* para o *Processo de desenvolvimento de Software do Groovoo*, possa ser apresentada e validada pelos diretores da *startup Atena Solutions*.

### 3.2.7 Validade do plano

O objetivo é validar, junto a direção da startup, os achados deste estudo. E avaliar inclusive, se os achados podem ser generalizados. Inicialmente o foco é o *Projeto Groovoo*, mas espera-se que a abordagem possa ser empregada nos demais projetos da *Atena Solutions*.

### 3.2.8 Limitações do Estudo

O escopo desta pesquisa está limitada ao processo de desenvolvimento do projeto Groovoo da startup Atena Solutions, restringindo a generalização dos resultados.

### 3.2.9 Relatórios

Esta fase é caracterizada pela monografia deste trabalho de dissertação.

### 3.2.10 Cronograma

Na Figura 11 está o cronograma do projeto.

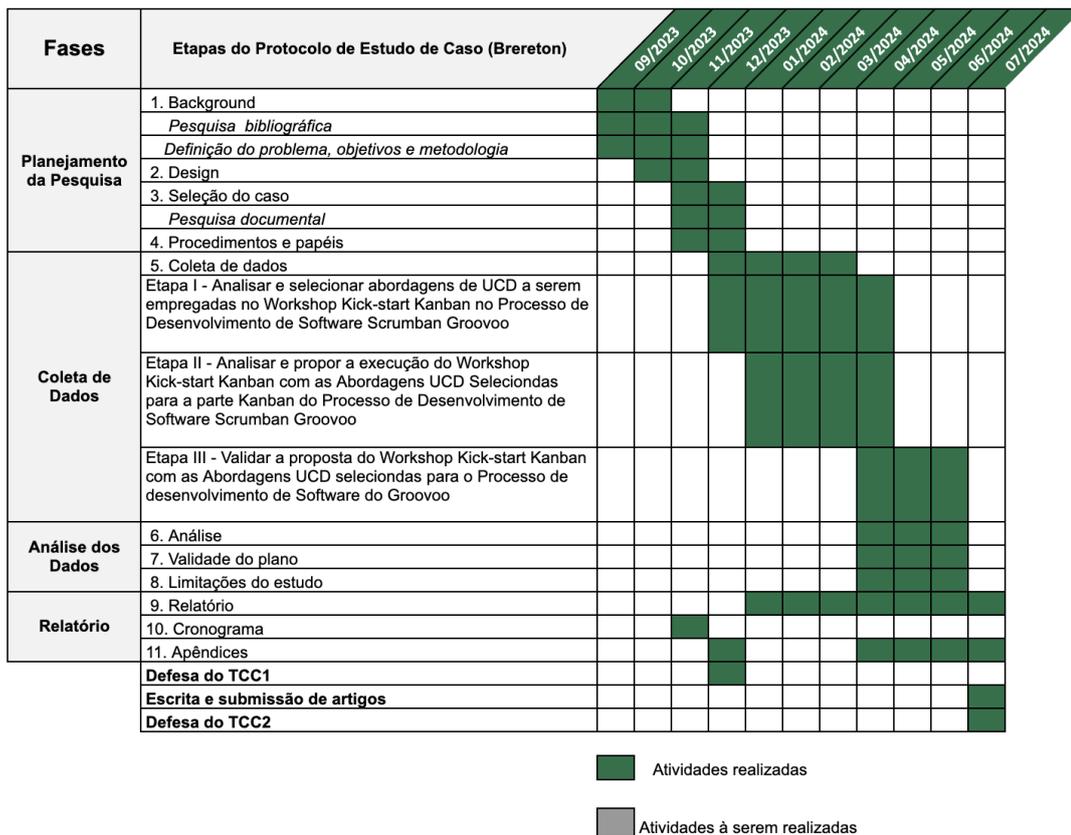


Figura 11 – Cronograma do projeto de pesquisa

### 3.2.11 Apêndices

Os documentos produzidos pelos autores deste trabalho, como a carta de autorização da coleta de dados e citação à empresa e os documentos com diretrizes advindos do planejamento e execução do *workshop* constam como apêndices.

- Apêndice *A* - *Carta de Autorização*
- Apêndice *B* - *Regras para montagem do Questionário*
- Apêndice *C* - *Exemplos de perguntas do Questionário*
- Apêndice *D* - *Avaliação Heurística com a Equipe*
- Apêndice *E* - *Motivações para a Adição do Design Centrado no Usuário (UCD) ao Processo de Desenvolvimento Groovoo*
- Apêndice *F* - *Dicionário de termos técnicos*
- Apêndice *G* - *Template de issues*
- Apêndice *H* - *Colunas do quadro Kanban*

### 3.3 Considerações Finais do Capítulo

Neste capítulo apresentou-se o Plano Metodológico adotado neste trabalho. No capítulo seguinte apresenta-se o objeto de estudo deste trabalho, com uma breve descrição da *startup* e de seu processo de desenvolvimento.

## 4 Atena Solutions

### 4.1 Considerações Iniciais do Capítulo

Neste capítulo apresenta-se uma descrição da *Startup Atena*, apontando as características da empresa, seu modelo de processo baseado no *Scrumban* e detalhes sobre o projeto *Groovoo*, objeto de estudo deste trabalho. Finalizando, apresenta-se uma síntese de um trabalho em que os autores propuserem diretrizes UCD para o projeto *Groovoo*.

### 4.2 Atena Solutions

A *Atena Solutions*, fundada em julho de 2006 por Frederico Castro e Arthur Moraes, é uma startup com operações em várias cidades, incluindo São Francisco, Austin, Nova York, São Paulo, Brasília, Manaus e Vancouver. Ela se concentra no desenvolvimento de software e computação em nuvem, principalmente seguindo o modelo de Software como Serviço (SaaS). Alguns de seus clientes são: *Caixa Econômica Federal*, *Infraero Aeroportos*, *Kikkin Internet Banking*, *Sodexo*, a própria *Groovoo* dentre outros.

#### 4.2.1 Groovoo

O projeto *Groovoo* iniciou as suas atividades no ano de 2021 com o objetivo de ser uma plataforma que possibilita a interação entre os usuários e a venda de ingressos para *shows* realizados nos Estados Unidos, com um foco especial no público brasileiro. O projeto conta atualmente com uma equipe de oito integrantes, dos quais: três *devops*, dois desenvolvedores, um cliente, um gerente de qualidade e um *scrum master*.

### 4.3 Metodologia Scrumban Groovoo

A *Atena Solutions* traz consigo a tentativa de aplicação de uma metodologia de desenvolvimento híbrida para o projeto *Groovoo*, combinando os métodos ágeis Scrum e Kanban, resultando no método *Scrumban* (GOMES; LUCAS, 2023). A escolha desse método, baseado nos próprios princípios do *Scrumban*, visa aproveitar as melhores características de ambos os modelos. O *Scrumban* mantém a natureza normativa do *Scrum*, ao mesmo tempo em que aproveita a capacidade de aprimoramento e a visualização de processos do *Kanban*. A equipe do *Groovoo* busca aplicar essa metodologia com papéis bem definidos, que incluem o Time de Produto, *Scrum Master*, Time de Infraestrutura e o Time de Desenvolvimento.

Embora exista essa tentativa, foi possível notar que uma série de princípios não são respeitados, fazendo com que o processo de desenvolvimento fique completamente desorganizado e sem validade. Atualmente, a *startup* não conta com um processo bem definido, onde princípios básicos como *daily's*, *plannings* e outras estruturas de organização como um quadro *Kanban* não são respeitadas.

### 4.3.1 Time de Produto

O Time de Produto desempenha um papel essencial na gestão e aplicação de estratégias com o objetivo de maximizar o valor do produto. Esse grupo desempenha funções para a coleta de requisitos até a orientação do produto para o público apropriado. O Time de Produto é composto por:

- *Shareholders* - Consistem em um trio de acionistas majoritários da *Groovoo*. Eles estão continuamente em busca de melhorias no sistema, visando atrair um público mais amplo e aumentar o valor do produto;
- *Stakeholders* - São, por definição, qualquer grupo ou indivíduo interessado em um produto e que pode afetar ou ser afetado a partir do alcance dos objetivos e interesses de uma empresa (FREEMAN; MCVEA, 2001). Dentro da *Groovoo*, esse grupo inclui clientes, organizadores de eventos, investidores, desenvolvedores, dentre outros;
- *Product Manager* - O *Product Manager* desempenha um papel multifuncional, atuando em várias áreas do desenvolvimento do produto, como negócios, marketing e desenvolvimento. Sua principal responsabilidade é garantir que a visão do produto esteja alinhada com o que o cliente necessita, sendo um facilitador nas entregas e removendo possíveis obstáculos.
- *Product Owner* - O *Product Owner* (PO) é o membro que geralmente representa o cliente dentro da *Groovoo*. É ele que traduz as necessidades dos clientes e usuários em requisitos e define a prioridade de cada um deles. Sua função é garantir que a visão do produto esteja alinhada com as necessidades do cliente, tornando-se um facilitador nas entregas e auxiliando na remoção de obstáculos (CARVALHO; MELLO, 2012).

### 4.3.2 Scrum Master

O *Scrum Master* possui um papel importante na aplicação e manutenção do processo *Scrum* dentro da *Groovoo*. Ele também atua para que não existam impeditivos dentro da equipe que impeçam que o trabalho seja realizado, como obstáculos apontados nas reuniões diárias (CARVALHO; MELLO, 2012).

### 4.3.3 Time de Desenvolvimento

- Líder Técnico - O Líder Técnico da *Groovoo* é um desenvolvedor com a capacidade de propor mudanças na arquitetura e nas tecnologias utilizadas no desenvolvimento do software. Ele assume as demandas mais complexas e auxilia outros membros da equipe de desenvolvimento a superarem obstáculos técnicos.
- Desenvolvedor - Os desenvolvedores *full stack* atuam nas demandas priorizadas pelo *PO* no *Sprint backlog* e são capazes de lidar com todo o ciclo de desenvolvimento do produto. Eles são responsáveis pelo desenvolvimento técnico, trabalhando no código-fonte tanto no backend quanto no frontend.

### 4.3.4 Time de Infraestrutura

O Time de Infraestrutura é formado por engenheiros *DevOps* que atuam na configuração do ambiente de todos os projetos da empresa. Eles são responsáveis pela manutenção, suporte e evolução dos sistemas de acordo com as necessidades do produto.

### 4.3.5 Processos do *Groovoo*

O processo de desenvolvimento atual do Projeto *Groovoo* busca utilizar o *framework Scrumban*. A equipe em teoria organiza por meio de *sprints* que duram entre duas a quatro semanas, porém raramente são definidos prazos, ou seja, dificilmente é aplicado esse princípio. As demandas são granuladas em atividades menores, chamadas de *tasks*. As *tasks* nada mais são que unidades pequenas de uma determinada demanda. Conforme as *tasks* vão sendo realizadas pela equipe, elas vão sendo reportadas diariamente nas *dailys*, para que os demais *stakeholders* estejam alinhados em relação ao *status* atual da demanda e também das dificuldades encontradas durante o desenvolvimento.

O processo inicia com o *PO* traduzindo as necessidades do cliente em demandas e requisitos para o *software*, os quais são formalizados e registrados no *Product Backlog*. Toda o planejamento e priorização ocorre entre o cliente e o *PO*.

Após a realização do planejamento, as tarefas e requisitos elicitados para cada *Sprint Planning* são incorporadas ao quadro de fluxo de trabalho (*Kanban*) (GOMES; LUCAS, 2023). Isso ocorre para melhorar a visualização das demandas por meio dos cards no quadro e que são categorizados da seguinte forma:

1. Lista de Tarefas Pendentes (*Backlog*) - O *Backlog* é o local que apresenta todas as tarefas que foram definidas pela equipe de desenvolvimento de um projeto e que serão realizadas em algum momento;

2. Planejamento da Atividade (*Sprint*) - A *Sprint* é uma lista de tarefas que a equipe escolhe e que pretende realizar em um período de tempo determinado, geralmente com um número associado (por exemplo, *Sprint 1*, *Sprint 2*);
3. Tarefas em Andamento (*In Progress*) - A categoria *In Progress* inclui as tarefas que a equipe de desenvolvimento está trabalhando. Ao finalizar, as tarefas são encaminhadas ao servidor de homologação para serem validadas;
4. Testes em Andamento (*Testing*) - A categoria *Testing* abrange as tarefas que foram desenvolvidas pela equipe de desenvolvimento e agora estão prontas para serem testadas pela Equipe de Qualidade;
5. Pronto para Implementação (*To Deploy*) - A categoria *To Deploy* é a etapa final do fluxo de trabalho que é visualizado no *Kanban*. Ela inclui as tarefas que já foram testadas em um ambiente de homologação e estão prontas para serem encaminhadas para o ambiente de produção e usadas pelos usuários.

Após o planejamento, inicia-se os trabalhos pela equipe de desenvolvimento, onde as atividades definidas para a *Sprint* serão realizadas. Durante a realização das atividades de cada *Sprint*, são realizadas as *Dailys*, que são reuniões diárias e curtas, geralmente realizadas em horários fixos. Essas reuniões contam com a presença do *Product Manager*, *Product Owner*, Time de Desenvolvimento, Time de Qualidade e o Time de *Design*, onde o foco é que cada integrante da equipe apresente no trabalho no dia anterior, falar sobre dificuldades encontradas e falar sobre o que irá trabalhar posteriormente.

Com as tarefas concluídas, a equipe de desenvolvimento as envia para um ambiente de teste chamado homologação. Nesta fase, a equipe de qualidade avalia a qualidade das tarefas, verificam se elas atendem aos critérios estabelecidos e realizam os testes necessários para verificar se tudo funciona bem.

Após os testes, o sistema é automaticamente implantado no ambiente de produção, por meio de uma estrutura criada pela equipe de infraestrutura, garantindo que o sistema esteja seguro e confiável. É a partir desse momento que as novas funcionalidades estarão disponíveis para os usuários.

Finalmente, é realizada uma revisão chamada *Sprint Review*, onde as equipes se reúnem para discutir como foi o progresso durante a *Sprint*. A equipe de desenvolvimento apresenta as tarefas que foram concluídas, destacando o que funcionaram bem, quais desafios enfrentaram e como lidaram com eles. Também é falado sobre as tarefas que não puderam ser realizadas e que serão planejadas para a próxima *Sprint*.

A Figura 12 apresenta o processo de desenvolvimento do projeto *Groovoo*.

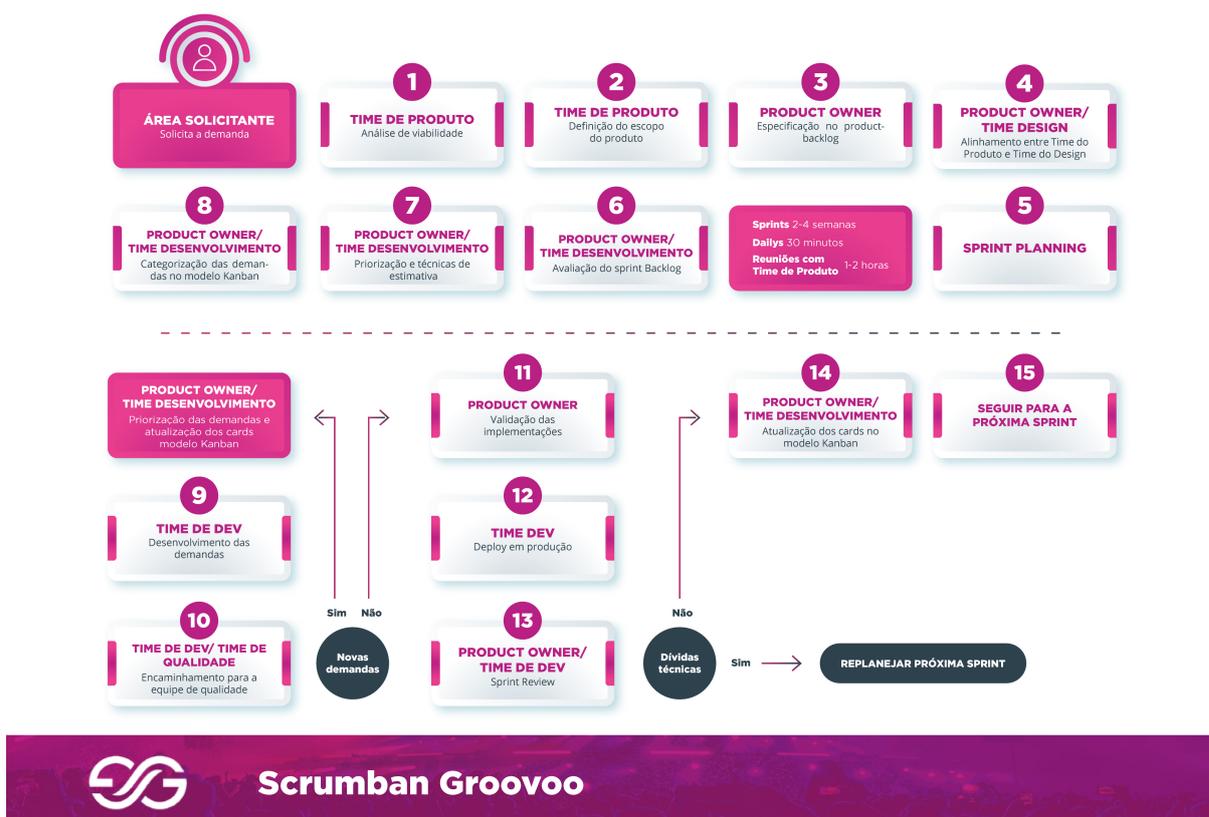


Figura 12 – Processo de desenvolvimento de *software* do Projeto Groovoo (GOMES; LUCAS, 2023)

Embora essa seja a estrutura definida dentro do fluxo de desenvolvimento do *Groovoo*, foi identificado que uma série desses princípios são ignorados. Isso ocorre pela má organização das equipes e uma alta prioridade centrada no desenvolvimento de novas funcionalidades. As *Sprints* e *Planning Review*, por exemplo, são técnicas do *Scrum* que não foram identificadas na prática. No entanto, algumas outras estruturas, como as *daily*s, são seguidas de forma fidedigna. Já o quadro *Kanban* foi outra técnica em que a equipe relatou dificuldades em manter, tornando-a assim uma técnica obsoleta e com um quadro totalmente desatualizado.

#### 4.4 Trabalhos anteriores envolvendo o processo de desenvolvimento da Atena Solutions

No contexto deste estudo, é importante destacar a existência de um trabalho anterior que se dedicou a investigar e propor o uso de diretrizes *UCD* no processo de desenvolvimento atual da empresa. Este trabalho selecionou algumas técnicas para serem utilizadas com as diretrizes *UCD*, entre elas estão grupo focal, *Card Sorting*, design participativo e *thinking aloud* (GOMES; LUCAS, 2023). Além disso, propuseram também

técnicas que não estavam sendo utilizadas no *framework Scrumban*. O resultado final foi o quadro *scrumban-UCD* representado pela Figura 13.

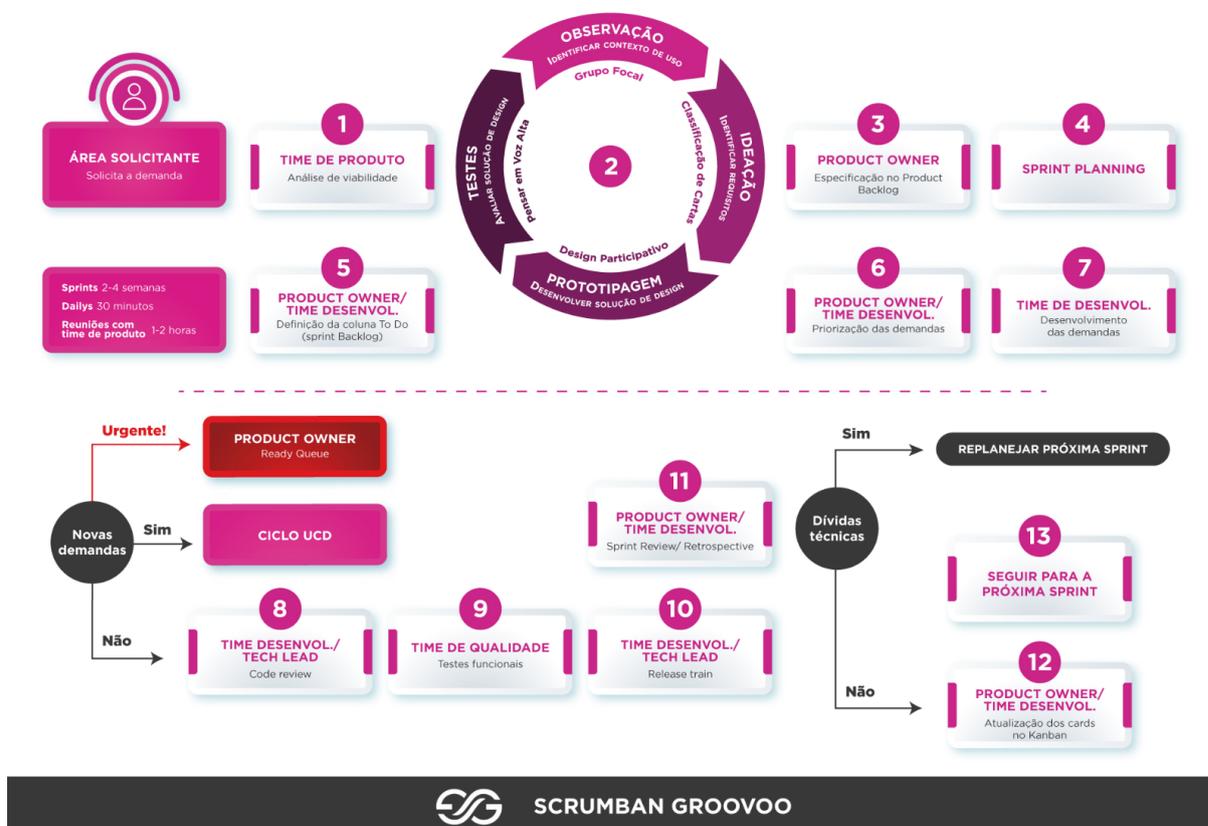


Figura 13 – Scrumban-UCD do Projeto Groovoo (GOMES; LUCAS, 2023)

O processo inicia com uma nova demanda encaminhada ao Time de Produto, que realiza uma análise de viabilidade para determinar se é factível. Em seguida, a demanda entra no ciclo iterativo do *UCD*, onde o time de produto e o time de *design* trabalham para entender as necessidades dos usuários e desenvolver soluções de *design*. Após validação, a demanda vai para o *Product Owner* para especificação no *Product Backlog*. A *Sprint Planning* ocorre para planejar as demandas da próxima *sprint*. O Time de Desenvolvimento inicia a implementação, com possibilidade de novas demandas urgentes indo diretamente para a fila *Ready Queue*. Após o desenvolvimento, há revisões de código e testes de qualidade. A cerimônia de *Release Train* adiciona novas funcionalidades ao ambiente de produção. A *Sprint Review/Retrospective* avalia a *sprint* e planeja a próxima, atualizando o quadro Kanban.

## 4.5 Considerações Finais do Capítulo

Neste capítulo foi apresentada uma caracterização do objeto de estudo deste trabalho. Nos capítulos seguintes apresenta-se o desenvolvimento da proposta deste trabalho. Inicia-se com as Diretrizes UCD para o Kick-start Kanban definidas.

# 5 Diretrizes de UCD para o *Kick-start Kanban*

## 5.1 Considerações Iniciais do Capítulo

Neste capítulo, retoma-se os *Objetivos Específicos* apresentados no Capítulo 1. Primeiramente, é apresentada a seleção e análise das diretrizes UCD baseado na literatura e na viabilidade de aplicação dentro do processo de desenvolvimento do Projeto *Groovoo*. Em seguida, realiza-se a análise e propõe-se a execução do *Workshop Kick-start Kanban* com as *Abordagens UCD Selecionadas* para a parte *Kanban* do Processo de Desenvolvimento do projeto *Groovoo*.

## 5.2 Análise e seleção de abordagens UCD

Nesta seção apresenta-se os procedimentos e etapas adotadas para a conclusão do 1º objetivo específico:

- *Analisar e selecionar abordagens de UCD a serem empregadas no Workshop Kick-start Kanban no Processo de Desenvolvimento de Software Scrumban Groovoo.*

Ao iniciar o processo de seleção de abordagens de Design Centrado no Usuário (UCD) para a *startup* Atena, alguns fatores foram considerados, desde a eficácia da metodologia até a utilização eficiente de recursos da empresa.

A empresa se mostrou disposta a colaborar com o processo de escolha antes de ser feita uma proposta final. O objetivo foi melhorar o processo de desenvolvimento, propondo uma abordagem ágil, que envolvesse o usuário de alguma forma e que gerasse valor. Para isso, os seguintes itens foram usados como base para a escolha, junto de suas justificativas:

**Explorar Abordagens UCD na literatura** - Existem várias abordagens UCD disponíveis, cada uma com suas vantagens e aplicações específicas. Algumas delas são: análise heurística, testes de usabilidade, entrevistas com usuários, prototipagem rápida, análise de jornada do usuário, dentre outras. Cada método oferece *insights* valiosos sobre as necessidades e preferências do usuário, mas é importante escolher aqueles que se alinhem melhor com os objetivos da startup e que possam ser executados com recursos limitados;

**Garantir Embasamento Científico** - Escolher abordagens UCD presentes na literatura para um embasamento científico sólido. Com as pesquisas, tem-se as análises realizadas por [Argumanis, Moquillaza e Paz \(2020\)](#) e [Argumanis, Moquillaza e Paz \(2021\)](#).

Os dados acadêmicos reforçam a confiabilidade no processo de obtenção dos resultados e aumenta a credibilidade da estratégia de *design centrado no usuário* para o processo da *startup*;

**Compreender as Necessidades da Startup** - Tendo como fundamentos entender as necessidades específicas da startup, que permitisse identificar as metas de negócios, o público-alvo e os desafios enfrentados. Uma análise detalhada ajudou a direcionar a escolha da abordagem mais adequada e a garantir que os recursos sejam alocados de forma eficiente.

O público-alvo da startup são empresas de médio porte no setor de sustentabilidade que buscam soluções eficientes e ecologicamente responsáveis:

- **Metas de Negócio:** as metas de negócios do projeto *Groovoo* envolve a expansão do produto de software para novos países, com o Brasil sendo o próximo alvo estratégico. Como consequência e também tendo como meta, busca-se um aumento na base de clientes e almeja a consolidação no mercado como uma das principais plataformas de venda de ingressos para eventos nos próximos anos.
- **Público-Alvo:** O Projeto *Groovoo* é direcionado a indivíduos interessados em participar de shows e eventos presenciais, além de atender a organizadores e patrocinadores que buscam uma plataforma para promover e comercializar seus eventos.
- **Desafios Enfrentados:** os desafios enfrentados incluem a concorrência existente no mercado de tecnologia quanto as plataformas que promovem eventos para venda de ingressos, a necessidade de adaptação contínua às tendências do setor, principalmente quanto às novas tecnologias e a garantia de que a interface do usuário do software atenda aos padrões de usabilidade esperados pelos clientes.

**Potencial de Geração de Valor** - Uma vez que as abordagens UCD tenham sido identificadas, foi importante avaliar seu potencial de geração de valor para a empresa. Isso envolveu considerar como cada método pode contribuir para o aprimoramento do produto ou serviço oferecido pela startup, bem como para a satisfação do usuário.

Uma lista de critérios foi definida levantados pelos gestores:

- O processo de geração de ideias, tomadas de decisão e desenvolvimento não deve ser rígido, evitando uma dependência por parte do usuário;
- Ser capaz de compreender as necessidades e dores dos usuários para que a equipe de produto elicite requisitos adequados para saná-las;

- O foco do desenvolvimento deve ser voltado para a solução dos problemas identificados pelos usuários e para suas necessidades, visando melhorar a experiência de uso do *Groovoo*;
- A interação do usuário ou a introdução das técnicas não deve ser obstáculos para o avanço do processo de desenvolvimento do produto;
- Espera-se a utilização de métodos eficazes para documentar as experiências dos usuários no uso do *software Groovoo*, visando aprimorar sua usabilidade e satisfazer as necessidades dos usuários;
- Considerar técnicas e/ou métodos capazes de atrair novos usuários;
- Reconhecendo a existência de dois grupos distintos de usuários do *Groovoo*, com os produtores sendo o grupo principal, espera-se a abordagem de ambos os grupos para garantir uma visão abrangente das necessidades e expectativas dos usuários;
- Espera-se métodos eficientes e capazes de realizar coleta de feedback dos usuários antes e após o desenvolvimento do produto, com o objetivo de aprimorar o software e atender às demandas dos usuários de forma mais eficaz.

A partir da lista apresentada, os critérios foram resumidos e categorizados, conforme:

- **C1** - o processo inicial de geração de ideias, tomadas de decisão e desenvolvimento não pode partir exclusivamente do usuário;
- **C2** - ajuda a compreender as necessidades e dificuldades dos usuários;
- **C3** - ajuda a identificar problemas enfrentados pelos usuários;
- **C4** - a interação do usuário ou a aplicação da técnica não pode travar o processo de desenvolvimento ou parte dele;
- **C5** - eficaz para documentar e registrar a experiência dos usuários no uso do sistema;
- **C6** - pode ajudar a entender e aprimorar processos que envolvam o usuário, compreendendo e melhorando necessidades e expectativas, assim podendo atrair novos usuários;
- **C7** - é capaz de lidar com assuntos de forma comum ou independente, envolvendo um ou mais grupos de usuários;
- **C8** - é eficaz na coleta de *feedbacks*.

Além dos critérios, os gestores optaram por centralizar as decisões estratégicas e de desenvolvimento das funcionalidades do *software Groovoo* na equipe de produto, ***excluindo a possibilidade de participação direta dos usuários nesse processo.***

Levando em consideração a análise das técnicas UCD realizada por (ARGUMANIS; MOQUILLAZA; PAZ, 2021) e os itens abordados, principalmente os subitens referentes ao tópico de Potencial Geração de Valor, foram definidos quais os critérios e as técnicas são capazes de satisfazer os gestores. Esses são apresentados na Tabela 12.

Tabela 12 – Análise de técnicas UCD com base nos critérios dos gestores

	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>
Protótipo de papel	X	X	-	-	-	X	X	-
Personas	X	X	-	X	-	X	X	-
Questionário	X	X	X	X	X	X	X	X
<i>Card sorting</i>	X	X	X	X	X	X	X	X
Avaliação heurística	X	X	X	X	X	X	X	X
Investigação Contextual	X	X	X	X	X	X	X	X
Thinking Aloud	X	X	X	X	X	X	X	X
Scenários	X	X	X	-	-	X	X	-

Na Tabela 12 foram apresentadas as técnicas UCD abordadas por Argumanis, Moquillaza e Paz (2021) e os respectivos critérios da empresa aos quais aquela técnica é capaz de satisfazer. Vale ressaltar que algumas técnicas foram inicialmente descartadas por não atenderem ao escopo do projeto *Groovoo*. São elas:

- Sprint 0: o projeto já se encontra em desenvolvimento, não existindo muito espaço para a aplicação de uma Sprint 0;
- Design em Pares: a equipe não possui um designer, dificultando a implementação dessa técnica;
- Big Design Upfront: projeto já em andamento e sob uma metodologia ágil, ou seja, essa técnica não se enquadraria no contexto do *Groovoo*;
- Design Conduzido por Desenvolvedores: essa é uma técnica que, quando necessário, já é utilizada, principalmente em demandas de alta complexidade;
- Mapa da Jornada do Cliente: essa é uma atividade que não envolve apenas o uso ou o desenvolvimento da ferramenta, sendo uma técnica mais de análise de negócio do que voltada para UX.

As demais técnicas apresentadas se enquadraram com a ideia inicial de aplicar UCD no contexto Ágil do projeto, ao mesmo tempo que atenderam alguns ou todos os requisitos solicitados pelos gestores.

Dentre essas técnicas, as que mais se enquadraram no contexto do projeto *Groovoo*, atendendo a todos os critérios dos gestores, foram:

- Questionário;
- *Card Sorting*;
- Avaliação Heurística;
- Investigação Contextual; e
- *Think Aloud*.

## 5.3 Análise e proposição do *Kick-start Kanban* com as técnicas UCD

Nesta seção, serão apresentados os procedimentos e estratégias necessárias para alcançar o segundo objetivo específico do trabalho.

- Analisar e propor a execução do *Workshop Kick-start Kanban* com as *Abordagens UCD Selecionadas* para a parte Kanban do Processo de Desenvolvimento de Software Scrumban Groovoo.

### 5.3.1 Reunião para escolha das técnicas UCD analisadas

Esta reunião contou com a participação de um dos gestores. Foram apresentadas todas as técnicas analisadas com base nos critérios estabelecidos. Dentre as técnicas apresentadas, as que mais estão de acordo com os princípios da empresa, segundo o gestor, foram:

- Questionário; e
- Avaliação Heurística.

A escolha das técnicas Questionário e Avaliação Heurística, dentre outras que também poderiam ser aplicadas ao projeto, ocorreu já que o gestor considerou que elas são de fácil compreensão e implementação, além de não engessar o processo de desenvolvimento, principal preocupação relatada desde o início dos trabalhos. Essas técnicas se alinham bem com o escopo do projeto e com as expectativas da empresa em relação a novas introduções no processo de desenvolvimento. Também foi relatado que alguns princípios heurísticos já são intuitivamente utilizados e visualizados de forma implícita no processo de desenvolvimento.

### 5.3.2 Atividades adicionadas ao *Kick-start Kanban*

Com base nas Tabelas 3, 4 e 5, apresentadas na Seção 2.3 e com a necessidade de adaptar o *workshop* com as técnicas UCD selecionadas, foram geradas novas atividades para cada etapa, além das atividades já definidas.

Na primeira etapa, por ser uma etapa de contextualização, foi necessário a adição de uma atividade para conceituar a equipe sobre o UCD. Ela foi aplicada após a apresentação do Kanban. A Tabela 13 apresenta uma breve descrição dessa atividade.

Tabela 13 – Etapa 1 - Atividades adicionadas ao workshop.

Atividade	Descrição	Definição de Concluído	Pode ser ignorado?
Apresentar o UCD	Apresentar os conceitos fundamentais que norteiam o <i>User Centered Design</i> para a equipe, incluindo a importância da empatia com o usuário.	Equipe compreende os princípios e está apta a aplicar técnicas de UCD no projeto.	Não

Essa Etapa 1 - a atividade é para a compreensão do UCD, para apresentar a importância e os principais conceitos do UCD.

Já na Etapa 2, que constitui o planejamento de execução do *workshop*, foram adicionadas duas atividades para planejar a execução das abordagens escolhidas na seção anterior. Estas atividades devem ser concluídas entre a configuração do *workshop* e preparativos extras para a equipe. Na Tabela 14 apresentam-se essas atividades.

Tabela 14 – Etapa 2 - Atividades adicionadas ao workshop.

Atividade	Descrição	Definição de Concluído	Pode ser ignorado?
Planejamento da técnica: Questionário	Buscar modelos de questionários voltados ao desenvolvimento de funcionalidades que envolveram as respostas de usuários	Modelos de questionários relevantes identificados	Não
Planejamento da técnica: Avaliação Heurística	Realizar uma análise especializada para identificar problemas de usabilidade no design de interfaces	Problemas de usabilidade identificados e documentados	Não

A tabela apresenta as atividades adicionais que foram incorporadas ao workshop durante a segunda etapa do projeto. Estas atividades serão planejadas com o objetivo de aprimorar o desenvolvimento de funcionalidades e a usabilidade das interfaces, seguindo os princípios UCD.

Por fim, na etapa três, por se tratar da execução do workshop, foram adicionadas 4 tarefas: explicação de cada técnica selecionada e criação de modelo ou condução para facilitar a implementação da técnica no Processo de Desenvolvimento.

Como ela comporá no futuro o quadro *Kanban*, essas atividades devem ser implementadas no *workshop* após a explicação do mesmo. A tabela 15 apresenta essas atividades.

Tabela 15 – Etapa 3 - Atividades adicionadas ao workshop.

Atividade	Descrição	Definição de Concluído	Pode ser ignorado?
Explicar como funciona a técnica Questionário	Apresentar aos participantes do workshop o funcionamento e os benefícios da utilização de questionários para coletar informações importantes sobre as necessidades e expectativas dos usuários em relação às novas funcionalidades.	Participantes entendem o papel e a importância dos questionários no processo de coleta de informações dos usuários.	Não
Criar um modelo de Questionário	Elaborar um questionário modelo para coletar informações importantes sobre novas funcionalidades	O modelo de questionário está pronto para ser utilizado no futuro pelo time	Não
Explicar como funciona a técnica Avaliação Heurística	Explicar aos membros da equipe os princípios e procedimentos da avaliação heurística para identificar problemas de usabilidade nas interfaces	Membros da equipe compreendem os princípios e procedimentos da avaliação heurística	Não
Conduzir uma Avaliação Heurística	Realizar uma análise detalhada das interfaces de usuário para identificar problemas de usabilidade com base nas heurísticas estabelecidas	Problemas de usabilidade identificados e documentados para futuras melhorias	Não

Essas atividades serão responsáveis por auxiliar na compreensão das técnicas UCD selecionadas, de forma que os membros da equipe compreendam seus objetivos e saibam criar e aplicar tais técnicas.

### 5.3.3 Proposta de execução do workshop com as técnicas escolhidas e atividades adicionadas

O *workshop* é composto por quatro fases, cada uma contendo uma série de atividades que podem ser adaptadas de acordo com as necessidades específicas da equipe. Nesta seção, foram rerepresentadas cada uma das fases e suas respectivas atividades, sendo possível observar a proposta para a execução do *workshop*, que consistiu em cada atividade de cada fase considerando as particularidades do processo de desenvolvimento atual do projeto Groovoo.

A primeira fase do Kick-start Kanban, entendimento das partes envolvidas, busca compreender o contexto da equipe e também definir as expectativas quanto ao *workshop*. A Tabela 16 foi elaborada para registrar as atividades, seus resultados e observações pertinentes de acordo com a Tabela 3 apresentada na seção 2.3.

Tabela 16 – Fase 1 - Atividades do *Kick-start Kanban* - Adaptado

Atividade	Resultados esperados	Observações
Apresentar o <i>Kanban</i>	Compreensão básica do conceito de <i>Kanban</i> e sua aplicação no contexto da equipe	Caso a equipe já utilize ou utilizou uma ferramenta kanban ou tenha conhecimento em alguma, basear a apresentação nela.
Explicar os princípios do UCD	Compreensão dos princípios do <i>User-Centered Design (UCD)</i> e sua relevância para o processo de desenvolvimento	As motivações para a adição do UCD ao processo de desenvolvimento devem ser documentadas e disponibilizadas à equipe.
Estudar o contexto da equipe	Análise do ambiente de trabalho, fluxos de trabalho atuais e desafios enfrentados	-
Definir as expectativas	Identificação das metas e objetivos do workshop, alinhamento das expectativas entre os participantes	-
Preparar o Gerente de Fluxo	Conhecimento das responsabilidades e habilidades necessárias para atuar como Gerente de Fluxo no contexto do <i>Kanban</i>	- Todos devem saber das responsabilidades do gerente de fluxo.
Obter uma decisão <i>GO/NO GO</i>	Decisão foi continuar ( <i>GO</i> ) ou parar ( <i>NO GO</i> ) com a implementação do <i>Kanban</i>	-

Essa tabela apresenta a inserção da atividade de UCD como incremento dentro do processo do workshop.

Após a compreensão do contexto da equipe e uma definição clara das expectativas em relação ao processo, o planejamento do *workshop* deve ser elaborado. Esta fase, conforme apresentado na Figura 6, envolve uma série de etapas que visam garantir o sucesso da execução do *workshop*. Estão descritos na Tabela 17 as atividades, os resultados esperados e observações que servirão para a execução.

Tabela 17 – Fase 2 - Atividades do *Kick-start Kanban - Adaptado*

Atividade	Resultado esperado	Observações
Criar um Planejamento	Documento resumindo agenda, objetivos do workshop, atividades planejadas, materiais necessários, responsáveis e cronograma	Os documentos devem estar disponíveis para todos os membros da equipe
Configurar o <i>Workshop</i>	Ambiente online preparado com as ferramentas necessárias para a realização do workshop (por exemplo, plataformas de videoconferência, documentos compartilhados)	Dar preferência para ferramentas que a equipe já utiliza no dia a dia; os documentos devem permanecer em um repositório compartilhado.
Planejamento da técnica: Questionário	Equipe compreendeu os exemplos que foram encontrados	Preparar um modelo e um questionário junto com a equipe no contexto <i>Groovoo</i>
Planejamento da técnica: Avaliação Heurística	Equipe compreendeu os princípios e procedimentos da avaliação heurística	Preparar um plano para conduzir a avaliação heurística em um cenário do <i>Software Groovoo</i> junto com a equipe
Preparativos extras para a equipe	Atualização da equipe sobre os detalhes do workshop	-

A tabela apresenta a Fase 2 com as técnicas UCD inseridas, contendo as respectivas atividades, resultados esperados e observações, caso existam.

Já na terceira fase do *workshop Kick-start Kanban*, ocorre a aplicação prática dos conceitos previamente definidos na seção anterior. Nela os participantes terão a oportunidade de adaptar os princípios do Kanban ao seu próprio ambiente de trabalho, promovendo uma compreensão mais profunda e a internalização dos conhecimentos adquiridos. Esta etapa será utilizada para encorajar a experimentação e a colaboração entre os membros da equipe, visando otimizar o fluxo de trabalho e identificar oportunidades de melhoria.

A Tabela 18 apresenta um pouco mais os detalhes dessa fase.

Tabela 18 – Fase 3 - Atividades do *Kick-start Kanban* - Adaptado

Atividade	Resultado esperado	Observações
Definir o <i>workshop</i>	Estruturação clara e detalhada do workshop, incluindo agenda, objetivos, atividades e materiais	-
Definir o UCD	Compreensão dos princípios do <i>User-Centered Design (UCD)</i> e sua aplicação no contexto da equipe	-
Identificar as preocupações atuais	Identificação e análise das principais preocupações e desafios enfrentados pela equipe	-
Definir uma Visão Compartilhada sobre o propósito da equipe	Alinhamento da equipe em relação ao propósito comum e metas compartilhadas	-
Descobrir o que a equipe faz	Clarificação das responsabilidades e atividades realizadas pela equipe	Os papéis devem estar definidos para todos os integrantes da equipe
Identificar e agrupar os itens de trabalho com base em características comuns	Agrupamento de tarefas semelhantes para facilitar a gestão e priorização	-
Explicar como funciona o <i>Kanban</i>	Compreensão dos princípios e funcionamento do <i>Kanban</i> como sistema de gestão visual	O <i>Github Projects</i> deve estar configurado para o uso da equipe após o workshop a fim de viabilizar as novas atividades da equipe
Explicar como funciona a técnica Questionário	Compreensão do uso da técnica de questionário como ferramenta de coleta de informações	Disponibilizar exemplos e os documentos gerados devem ser disponibilizados à equipe
Criar um modelo de Questionário	Desenvolvimento de um modelo de questionário para uso da equipe	O modelo gerado deve ser disponibilizado à equipe

Apresentar a técnica Avaliação Heurística	Explicar aos membros da equipe os conceitos e métodos por trás da avaliação heurística para identificar problemas de usabilidade	Disponibilizar materiais complementares e os documentos gerados devem ser disponibilizados à equipe
Conduzir uma Avaliação Heurística	Realizar uma análise detalhada das interfaces de usuário para identificar e documentar problemas de usabilidade com base em princípios heurísticos estabelecidos	A avaliação deve ser documentada e disponibilizada à equipe
Apresentar uma linguagem comum	Estabelecimento de termos e conceitos comuns para facilitar a comunicação dentro da equipe	Conceitos técnicos devem ser documentados em uma espécie de dicionário e disponibilizado à toda equipe
Definir políticas de visualização	Estabelecimento de regras e diretrizes para a visualização das atividades no quadro <i>Kanban</i>	O limite de WIP deve ser definido e as também, além de outras possíveis políticas que a equipe queira adotar
Criar e preencher o quadro	Criação do quadro <i>Kanban</i> e preenchimento inicial com as tarefas identificadas	Esse preenchimento deverá ser no <i>GitHub Projects</i>
Definir políticas que definem a maneira de trabalhar	Estabelecimento de diretrizes para a execução das atividades e gestão do trabalho no <i>Kanban</i>	-
Limitar WIP	Implementação de limites de trabalho em progresso para melhorar o fluxo e a produtividade	Número máximo de tarefas que poderão ser resolvidas ao mesmo tempo
Definir políticas para reuniões de planejamento	Estabelecimento de formatos e objetivos para as reuniões de planejamento no <i>Kanban</i>	Reuniões que ocorrerão durante a sprint pela equipe

Realizar a primeira reunião de planejamento	Condução da primeira reunião de planejamento usando o <i>Kanbanw</i>	-
Encerrar o <i>workshop</i>	Conclusão do workshop com revisão dos resultados e próximos passos definidos	-

Conforme observado, a Tabela 18 apresenta a mais extensa e trabalhosa fase, contando com uma série de atividades a serem realizadas. Dentro dessa tabela, constam também as atividades referentes ao *UCD*, importantes para compreensão e aplicação das técnicas.

A quarta e última tabela desta subseção apresenta um resumo das etapas da Fase 4 do *Kick-start Kanban*, focada na otimização do sistema *Kanban* da equipe. Estas etapas são projetadas para aprimorar a utilização do *Kanban*, visando alcançar a base da melhoria contínua, melhorando a eficiência e eficácia do fluxo de trabalho. O objetivo é levar a equipe a uma maior consciência de sua situação atual e mantê-la nesse estado, buscando constantemente melhorias.

Tabela 19 – Fase 4 - Atividades do *Kick-start Kanban* - Adaptado

Atividade	Resultado esperado	Observações
Ajustar reuniões de planejamento	Reuniões mais eficientes e focadas, alinhadas com as necessidades atuais da equipe	-
Adaptar as políticas ao entendimento atual da equipe	Políticas atualizadas e ajustadas conforme a evolução do entendimento e práticas da equipe	-
Entender a capacidade atual da equipe	Conhecimento claro da capacidade de trabalho da equipe para melhor planejamento	-
Limitar a WIP (Work in Progress)	Rever os limites para reduzir sobrecarga e melhorar o fluxo de trabalho	-
Aumentar o fluxo de trabalho (velocidade e eficiência)	Melhoria na velocidade de entrega e eficiência do processo de trabalho	-
Gerenciar as demandas	Estratégias eficazes para lidar com as demandas e garantir priorização adequada	-
Avaliar a profundidade do <i>Kanban</i> da equipe	Análise da eficácia e abrangência do quadro <i>Kanban</i> atual da equipe	-
Definir a próxima etapa	Planejamento dos próximos passos e melhorias a serem implementadas	-
<i>Feedback</i> sobre o guia de campo	Avaliação e ajustes no guia de campo do <i>Kanban</i> conforme experiência da equipe	-

Essa tabela apresenta um resumo das atividades da Fase 4 do Kick-start Kanban.

## 6 Execução do *workshop Kick-start Kanban* com as técnicas UCD selecionadas

O objetivo deste capítulo é apresentar as atividades adotadas para a produção do 3º objetivo específico. Será abordada a execução do *workshop Kick-start Kanban*, detalhando cada etapa do processo e fornecendo informações de suas atividades. Após essa descrição das etapas, será feita uma análise sobre os resultados obtidos.

### 6.1 Reunião de alinhamento com a Equipe

Para a execução do *workshop*, foram realizadas duas reuniões com todos os integrantes da equipe Groovoo. Na primeira reunião, foram abordadas as atividades relacionadas à definição de papéis, explicação das expectativas do *workshop*, compreensão do contexto da equipe e a decisão de continuar ou não, conforme a Fase 1 do Kick-start. Após a primeira reunião, foi montado um planejamento do *workshop* pelos especialistas. E na segunda reunião, foram realizadas as atividades práticas da Fase 3 do Kick-start.

### 6.2 Atividades da Fase 1 - Compreendendo a equipe

#### 6.2.1 Apresentação do *Kanban*

Foi feita uma apresentação completa do *Kanban*, abordando seus princípios fundamentais, suas práticas, e os benefícios esperados de sua implementação. Todos os integrantes entenderam os objetivos do *Kanban*, e principalmente como ele pode melhorar a eficiência do fluxo de trabalho e a importância da colaboração contínua para o sucesso da equipe e do produto.

Para esta apresentação, foi utilizado o *GitHub Projects* como ferramenta de exemplo, devido à familiaridade da equipe com ela.

#### 6.2.2 Apresentação do *Design Centrado no Usuário (UCD)*

Na mesma reunião, também foi apresentada a abordagem do *Design Centrado no Usuário (UCD)*. Foi discutido a importância de compreender as necessidades e expectativas dos usuários finais durante todo o processo de desenvolvimento. Além disso, destacou-se como a integração do UCD pode resultar em produtos mais intuitivos, eficientes e satisfatórios para os usuários, contribuindo para a redução de erros e retrabalhos.

A equipe foi incentivada a adotar uma mentalidade empática, colocando-se no lugar dos usuários para tomar decisões mais informadas e centradas no usuário.

Alguns pontos relevantes sobre o UCD estão disponíveis no apêndice [E](#).

### 6.2.3 Contexto da equipe

Quanto ao contexto da equipe, trata-se de uma equipe pequena, que atualmente desempenha um papel importante na manutenção do produto de *software Groovoo*. A ideia principal, de acordo com um dos gestores, e em comum acordo com o restante da equipe, é obter uma ferramenta de divulgação de eventos e compra de ingressos que seja reconhecida mundialmente por suas tecnologias, eficiência e inovação no mercado.

Além disso, a equipe destacou valores essenciais, como a singularidade para se diferenciar no mercado, a inovação contínua para manter a competitividade e a colaboração eficiente para garantir um trabalho em equipe harmonioso e produtivo, fundamentais para o sucesso do projeto.

### 6.2.4 Desafios relatados

Os integrantes relataram que a equipe inteira enfrenta diariamente uma série de desafios, entre eles destacam-se:

- Falta de organização no processo de desenvolvimento;
- Dificuldade em validar se as demandas atendem às necessidades dos usuários;
- Baixa aderência aos ritos do Scrum e a metodologia ágil;
- Dificuldade na visualização e gestão do fluxo de trabalho;
- Comunicação e alinhamento ineficazes dentro da equipe, que resultam em gargalos e dificuldades na compreensão das atividades.

A ideia é mitigar esses desafios com base no novo processo *Kick-start Kanban*.

### 6.2.5 Expectativas relatadas

Após discutir os problemas comuns enfrentados pelo time, as expectativas para o *workshop* elencadas pela equipe foram:

- Melhoria da visibilidade do trabalho em andamento;
- Redução de gargalos e os tempos de espera entre as etapas do processo;

- Solucionar o problema de sobrecarga de trabalho e de falta de priorização das demandas;
- Identificar e solucionar problemas quanto ao fluxo de trabalho.

### 6.2.6 Preparação do Gerente de Fluxo

Posteriormente, os especialistas do *workshop* solicitaram à equipe que escolhesse a pessoa mais preparada e experiente no gerenciamento de um quadro Kanban. Após avaliação do time, decidiu-se que a pessoa mais adequada para assumir as atividades de gerente de fluxo seria o gerente de projeto. Essa escolha se deve à sua experiência no uso do Kanban, além de sua atuação direta com as demandas e requisitos do produto diariamente.

### 6.2.7 Decisão *GO/NO GO*

Após apresentar as características e objetivos do *Kick-start* juntamente com as técnicas UCD, a equipe e os gestores decidiram dar continuidade ao processo de aplicação do *workshop* devido ao bom andamento da reunião e ao interesse mútuo da equipe em melhorar o processo.

## 6.3 Atividades da Fase 2 - Planejamento do *workshop*

Após a decisão de continuar com a execução do *workshop*, foi feita uma reunião entre os especialistas do *workshop* para planejá-lo. Essa reunião teve como objetivo concluir a Fase 2 do *Kick-start* e englobou toda a organização da execução da oficina.

### 6.3.1 Criação de um Planejamento

Para garantir que o workshop fosse eficaz e atingisse os objetivos desejados, foi criado um roteiro que envolveu a definição dos objetivos do workshop e também a definição do contexto em que a equipe estava inserida. Foi decidido que o workshop seria aplicado em uma sessão, focada em aspectos específicos do processo de desenvolvimento onde a equipe apresentava carências e também em trazer as técnicas escolhidas para dentro do processo.

### 6.3.2 Configuração do *Workshop*

O *workshop* foi configurado para ser realizado online, utilizando a plataforma *Discord*, após a concordância de toda a equipe. Esta abordagem foi escolhida para facilitar a participação de todos os membros da equipe, independentemente de sua localização

geográfica. E como material de apoio, foram utilizados slides, que foram apresentados durante a sessão para ilustrar os conceitos e métodos discutidos.

### 6.3.3 Planejamento da técnica Questionário

O planejamento da técnica Questionário incluiu uma pesquisa na literatura para identificar um conjunto de regras que pudessem orientar a criação de um questionário eficaz, capaz de coletar feedbacks de forma fácil e que os usuários respondessem de maneira que a equipe de desenvolvimento pudesse obter *insights* valiosos.

Além disso, o planejamento da técnica de questionário envolveu a criação de perguntas voltadas ao software *Groovoo*, incluindo aquelas que poderiam ser utilizadas de forma eficaz e outras que deveriam ser evitadas para garantir a qualidade das respostas obtidas.

### 6.3.4 Planejamento da técnica Avaliação Heurística

O planejamento da técnica de Avaliação Heurística seguiu as heurísticas de Nielsen como referência central. Tendo isso como base, foram disponibilizados exemplos de avaliações e uma avaliação será feita em conjunto com a equipe, baseada em cenários típicos enfrentados pelos usuários ao utilizarem o aplicativo *Groovoo*. O objetivo é que a equipe seja capaz de entender na teoria e na prática a criação e condução de uma avaliação heurística.

### 6.3.5 Preparativos extras para a equipe

A atividade de Preparativos extras contou com a reconfiguração do *GitHub Projects*, pois o quadro *Kanban* que a equipe possuía estava inativo e desatualizado. Fora o quadro *Kanban* do projeto, as ferramentas de comunicação interna utilizadas pela equipe foram reaproveitadas.

## 6.4 Atividades da Fase 3 - Execução do *workshop*

### 6.4.1 Definir o *workshop*

Durante o *workshop*, essa etapa foi essencial para que a equipe entendesse claramente os motivos e os objetivos dele, além dos artefatos e dos resultados que serão gerados. Além disso, a equipe pôde compreender o "quê" e o "porquê" do projeto que eles atualmente estão inseridos.

## 6.4.2 Definir o UCD

Após as definições do workshop, a equipe discutiu a respeito do UCD e a sua relevância para o momento atual do projeto *Groovoo*. A equipe entendeu que abordagem do UCD coloca o usuário no centro do processo de *design*, garantindo que todas as decisões de desenvolvimento sejam guiadas pelas necessidades, expectativas e experiências dos usuários finais.

## 6.4.3 Identificação das preocupações atuais

Esta atividade não foi executada já que pode ser ignorada, conforme previsto no workshop. Ela foi ignorada uma vez que a primeira reunião já havia abordado uma retrospectiva sobre os desafios enfrentados pela equipe.

## 6.4.4 Definição de visão compartilhada

Esta etapa também foi ignorada uma vez que foi abordada anteriormente na primeira reunião, em "Contexto da Equipe".

## 6.4.5 Descobrir o que a equipe faz

Essa etapa foi capaz de auxiliar a equipe na compreensão da sua situação atual, bem como transparecer o papel da equipe e o que ela realmente faz. Os próprios integrantes demonstraram uma certa dificuldade em relatar o que a equipe faz, uma vez que alguns integrantes desempenham papéis mútuos em virtude do tamanho da equipe.

As demandas que a equipe de desenvolvimento recebe vêm diretamente do time de produto e são focadas principalmente no desenvolvimento de novas funcionalidades e na correção de bugs.

As entregas finais são direcionadas para o ambiente de teste pela equipe de desenvolvimento para que sejam testadas e aprovadas pelas mesmas pessoas envolvidas no time de produto.

Embora tenha sido uma etapa difícil para a equipe identificar como as coisas ocorrem, o fluxo de trabalho da equipe segue as etapas de recebimento da demanda, desenvolvimento, testes e validação, e finalmente entrega.

## 6.4.6 Identificar e agrupar os itens de trabalho com base em características comuns

O objetivo desta etapa foi auxiliar a equipe a compreender se está focando nas atividades certas e equilibrando o fluxo de trabalho. Basicamente foram elencadas algumas

demandas que estavam no radar da equipe de produto para serem desenvolvidas e juntamente com os demais integrantes da equipe, todos foram capazes de agrupar e categorizar, por meio do quadro Kanban, a urgência e importância de cada demanda.

#### 6.4.7 Explicar como funciona o Kanban

Nesta etapa foi apresentado aos integrantes da equipe o funcionamento de um quadro *Kanban* na prática. Para isso, foi recriado o quadro para visualização do fluxo de trabalho na ferramenta *GitHub*, conhecida como *GitHub Projects*.

Após a finalização do quadro, os integrantes da equipe de produto, juntamente com o gerente de fluxo foram capazes de cadastrar algumas demandas e compreenderem um pouco melhor a dinâmica de funcionamento do atual processo. As colunas do quadro se encontram no Apêndice H.

Durante o desenvolvimento, algumas dúvidas surgiram e rapidamente foram sanadas. Todas elas foram posteriormente levadas e abordadas na seção *Definir políticas de visualização*.

#### 6.4.8 Explicar como funciona a técnica Questionário

Nesta etapa, foi apresentado aos integrantes da equipe como a técnica Questionário pode ser utilizada como uma ferramenta eficaz para a coleta de informações e *feedback* de usuários. A apresentação incluiu uma explicação sobre os diferentes tipos de questionários (abertos e fechados), como elaborar perguntas claras e objetivas, e a importância de garantir a anonimidade dos respondentes para obter respostas mais sinceras.

Os principais pontos abordados foram:

- **Tipos de Perguntas:** Foi discutida a diferença entre perguntas abertas e fechadas, e como cada tipo pode ser utilizado para obter diferentes tipos de informações.
- **Estrutura do Questionário:** Explicação sobre a importância de uma boa estrutura para garantir que o questionário seja fácil de entender e de responder. Incluiu a ordem das perguntas e o fluxo lógico entre elas.
- **Objetividade:** Foi enfatizada a importância de elaborar perguntas claras e diretas para evitar ambiguidades e interpretações erradas.
- **Análise de Dados:** Orientações sobre como coletar, compilar e analisar os dados obtidos com os questionários para transformar as respostas em informações úteis para a tomada de decisões.

Durante a apresentação, foi mostrado um questionário com perguntas relacionadas a funcionalidade de validação de *tickets* da ferramenta que utiliza a tecnologia NFC (Near Field Communication). Até o presente momento, o formulário com o questionário não foi aplicado, mas consta nas atividades a serem realizadas pela empresa.

#### 6.4.9 Criação do Modelo de Questionário

A criação do modelo de um Questionário foi elaborado com base nas regras definidas e constantes no Apêndice B, composto por regras baseadas em alguns autores e também algumas definidas entre a equipe durante o workshop. Essa etapa foi importante para que os membros da equipe compreendessem como e por que seguir essas regras na montagem de um questionário. Alguns exemplos de perguntas errôneas foram apresentadas para a equipe visualizar quando uma pergunta não seguia regras comuns e o quanto isso poderia impactar na qualidade das respostas. Estes exemplos estão documentados no Apêndice C.

#### 6.4.10 Apresentação da técnica Avaliação Heurística

Esse parte iniciou-se com a apresentação das 10 Heurísticas de Nielsen, importante para que a equipe compreendesse os fundamentos e a base de uma avaliação heurística. Posteriormente foi explicado como fazer uma avaliação heurística (definição de cenários, identificação e gravidade dos problemas e sugestões de melhoria).

#### 6.4.11 Condução da Avaliação Heurística

Com o objetivo de auxiliar na compreensão por parte da equipe quanto a prática de uma avaliação heurística, os especialistas no *Kick-start* elaboraram uma pequena avaliação, tendo como cenário a compra de tickets na plataforma *Groovoo*. Essa avaliação consta no Apêndice D desta monografia e foi realizada juntamente com a equipe.

#### 6.4.12 Linguagem Comum

Um dicionário de dados foi criado e inserido no documento *Readme* do quadro Kanban no *GitHub Projects*. Ele contém termos técnicos e foi criado com o objetivo principal de facilitar a compreensão dos termos comumente utilizados pela equipe de desenvolvimento entre as demais equipes.

Esse documento é de cunho incremental e não se restringe a termos apenas da equipe de desenvolvimento. Isso é necessário para que as equipes sejam capazes de entender umas as outras sem dificuldades. Esse dicionário de termos técnicos consta no Apêndice F.

### 6.4.13 Definição de Políticas de visualização

Durante o workshop, foram apresentadas algumas políticas de visualização, dentre elas um template previamente criado para abertura de issues, disponível no Apêndice G. Além disso, outros pontos foram definidos juntamente com a equipe, como cores e organização dos cards, labels de identificação das issues, estimativa de tempo de execução, etc.

### 6.4.14 Criação e preenchimento do quadro Kanban

O quadro Kanban foi montado com a participação de toda a equipe e ele pode ser visualizado nas Figuras 14 e 15

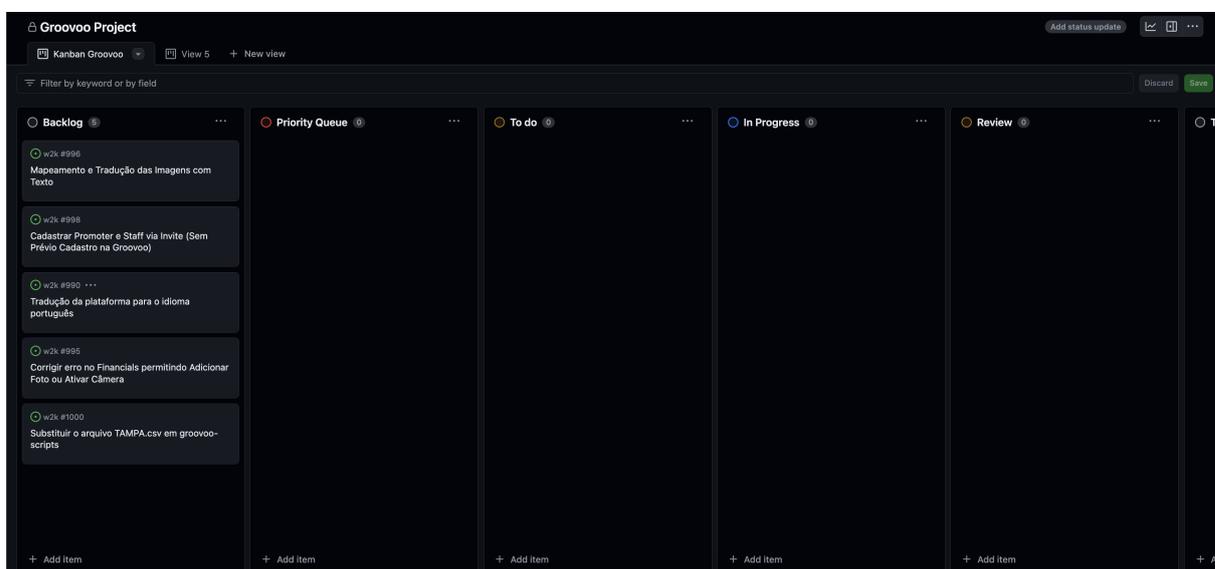


Figura 14 – Quadro Kanban do Projeto Groovoo

Conforme apresentado na figura acima, os cards foram inseridos com base em demandas já existentes e com outras que ainda não estavam sendo produzidas.

As demais colunas encontram-se vazias, já que até o presente momento nenhuma demanda havia sido finalizada ou se enquadrava na coluna *To Redo*, coluna essa destinada para atividades que foram consideradas finalizadas e testadas, porém não foram validadas por não passarem em algum dos testes feito pela equipe competente. A descrição de todas as colunas se encontram no apêndice H.

### 6.4.15 Definição das políticas de trabalho

Algumas políticas de trabalho foram definidas durante a reunião do workshop. São elas:

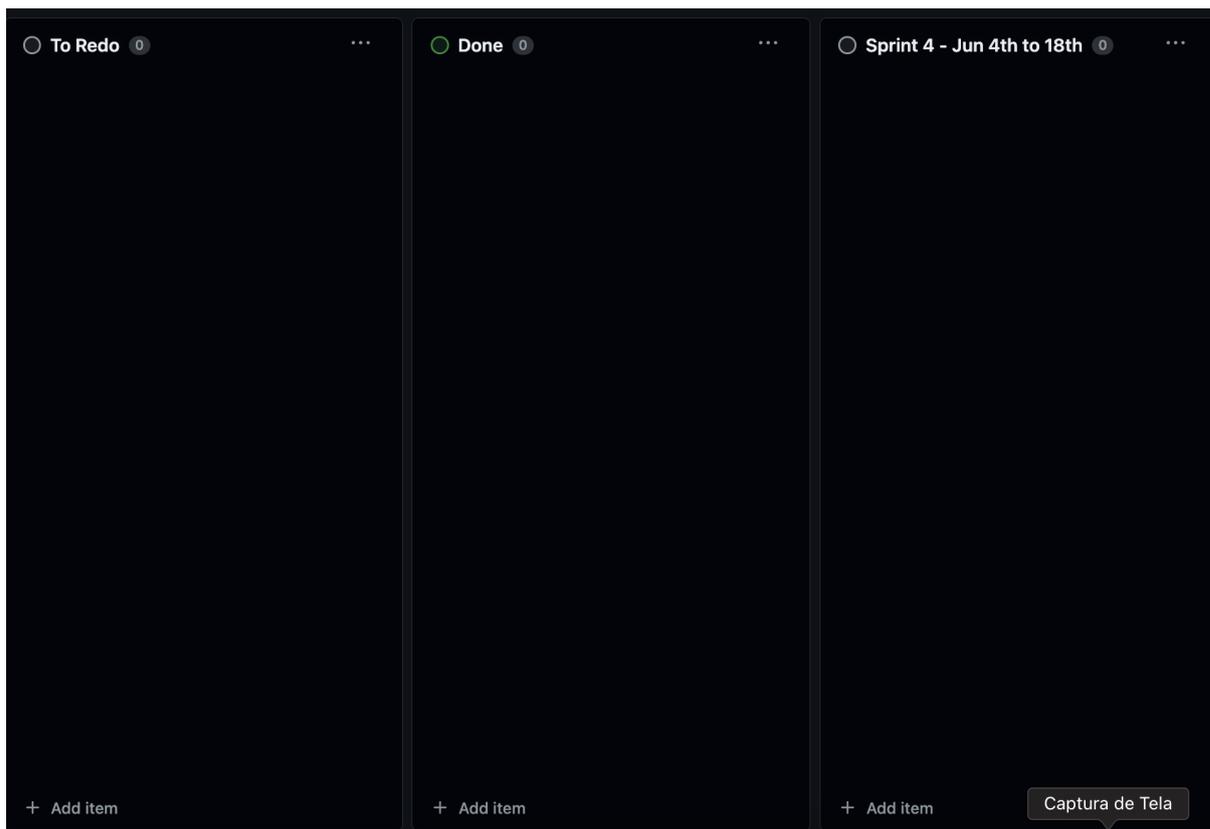


Figura 15 – Quadro Kanban do Projeto Groovoo

- **Discussão de Demandas:** Todas as demandas não compreendidas nos cards devem ser discutidas pela equipe para garantir um entendimento comum e eficaz.
- **Compartilhamento de Soluções:** A equipe de desenvolvimento compartilha a modelagem de soluções complexas para promover o aprendizado e a colaboração.
- **Mapeamento de Alterações:** Após o desenvolvimento, a equipe de desenvolvimento elabora um mapeamento de alterações para a equipe de testes, detalhando o que deve ser testado.
- **Retorno para To Redo:** Demandas que não passarem nos testes devem retornar para a coluna "To Redo" para revisão e correção. Os casos de teste realizados pela equipe de testes devem estar bem descritos nas issues e, se possível, com imagens do problema. Isso pode ser colocado como comentário na issue em questão.
- **Status das Dailys:** O status das atividades é compartilhado diariamente em texto ou áudio para manter a transparência e facilitar a comunicação entre os membros da equipe.
- **Limite de WIP:** Os desenvolvedores só poderão ter duas issues assigned, sendo uma demanda do Backlog e outra de Priority Queue. Nunca poderá haver duas demandas de mesma característica com o mesmo desenvolvedor.

- **Demandas externas ao fluxo:** todas as demandas externas ao fluxo de trabalho que foi definido deverão ser direcionadas imediatamente ao gerente de fluxo. Em caso de demandas vindas direto de produção, como bugs e outras necessidades, deverão ser realizadas de imediato, priorizando o desenvolvimento e depois a organização do trabalho.
- **Demandas em paralelo:** Cada membro da equipe de desenvolvimento só poderá pegar duas demandas em paralelo em caso de surgir alguma em produção, fazendo assim com que o desenvolvedor priorize a demanda em produção.
- **Priorização dos cards:** A priorização dos cards que estão na fila de pronto (To Do) ocorre pela ordem em que são apresentados, de cima para baixo. Ou seja, as demandas que devem ser priorizadas são as que estão mais ao topo da coluna To Do.
- **Itens pendentes:** Esses itens são aqueles em que o desenvolvedor já começou as análises/desenvolvimento, mas por algum motivo depende da intervenção de algum membro da equipe. Nesse caso, foi de comum acordo que o desenvolvedor tem total liberdade para entrar em contato direto com o integrante que pode ajudar a esclarecer as dúvidas e destravar o processo.

Conforme é previsto no Kick-start, a inserção, alteração ou remoção de qualquer política pode ocorrer a qualquer momento, com aviso prévio aos membros da equipe.

#### 6.4.16 Limitação do WIP

Conforme definido na seção anterior, cada desenvolvedor só pode trabalhar em duas tarefas ao mesmo tempo: uma do Backlog e outra da Priority Queue. Nunca poderá ter duas tarefas iguais com o mesmo desenvolvedor.

#### 6.4.17 Definição das políticas para reuniões de planejamento

As reuniões de planejamento poderão ocorrer de 7 em 7 dias ou de 14 em 14 dias, a depender da quantidade de dias que foi definido para a sprint. Elas ocorrerão no período da noite e contará com todos os membros, conforme combinado com toda a equipe.

O gerente de fluxo será o facilitador dessas reuniões (exceto na primeira vez, que serão especialistas no kick-start). Em caso de atraso no desenvolvimento das demandas da sprint, a reunião de planejamento também deverá ser postergada.

As reuniões devem ter duração máxima de 30 minutos, onde a agenda deve ser passada anteriormente para todos os interessados.

Vale lembrar que essa é uma reunião de planejamento, não havendo margem para resolução de quaisquer questões. Os problemas identificados devem ser anotados e discutidos posteriormente a reunião.

#### 6.4.18 Realização da primeira reunião de planejamento

Durante a execução, foi apresentado o objetivo da reunião de planejamento, que serve para alinhar a equipe sobre as tarefas que serão priorizadas e definir as ações para o próximo período de trabalho.

Posteriormente, as seguintes atividades foram realizadas:

- Revisão do quadro Kanban;
- Análise do backlog de tarefas, priorização dos itens mais importantes para a próxima sprint (To Do) e quantidade de itens selecionados;
- Seleção dos responsáveis pelos itens de trabalho (respeitando o limite de WIP);
- Revisão dos itens selecionados e atribuídos
- Esclarecimento sobre dúvidas
- Movimentação dos cards para coluna In Progress

Após essa reunião, o quadro ficou conforme a Figura 16 apresenta:

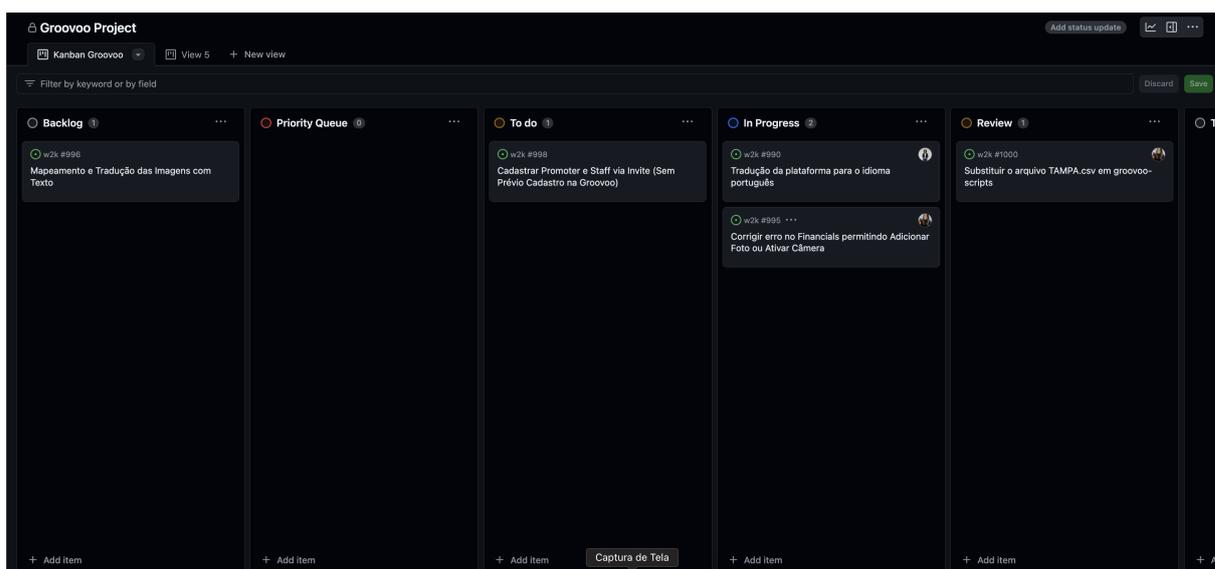


Figura 16 – Quadro Kanban do Projeto Groovoo

Os cards foram movimentados após serem inseridos, conforme apresentado na figura 14. Neste caso, como o quadro foi preenchido com o desenvolvimento já em andamento, existiam algumas demandas em desenvolvimento e uma disponível para teste. Duas

novas atividades foram discutidas e inseridas, conforme constam nas colunas Backlog e To Redo.

## 6.5 Encerramento do *workshop*

O *workshop* foi encerrado após a revisão das atividades realizadas e a organização final do quadro *Kanban*. Durante a reunião de encerramento, foram discutidos os próximos passos a serem seguidos pela equipe. Contudo, as atividades previstas para a Etapa 4 do *workshop* não foram realizadas. A decisão foi tomada para dar à equipe o tempo necessário para absorver os conhecimentos adquiridos até o momento e alcançar a chamada "linha de base da melhoria contínua".

# 7 Resultados Obtidos Após a Execução do Workshop

A realização da primeira reunião de planejamento marcou o fim do *workshop*. Nela a equipe revisou o quadro Kanban, priorizou tarefas, selecionou responsáveis, esclareceu dúvidas e iniciou as atividades conforme o planejado.

Dito isto, neste capítulo serão apresentados e analisados os resultados obtidos do *workshop*. Além da percepção de um dos especialistas que está presente no processo de desenvolvimento atual do projeto Groovoo, será apresentado os resultados de um questionário de feedback do *workshop*.

## 7.1 Principais pontos observados

Após a execução do workshop, alguns resultados puderam ser observados, destacando os impactos positivos e os desafios enfrentados pela equipe da *startup* no contexto da adaptação do processo de desenvolvimento para algo que envolva os usuários. A seguir, são apresentados os principais resultados positivos obtidos:

1. **Definição do Fluxo de Trabalho:** O fluxo de trabalho da equipe foi claramente definido. Durante o workshop, ajustes importantes foram realizados, permitindo que a equipe adaptasse suas práticas às necessidades específicas do projeto;
2. **Quadro Kanban Estruturado:** A equipe conseguiu estabelecer um quadro Kanban bem definido, facilitando a visualização das tarefas, o acompanhamento do progresso e a identificação de gargalos no processo de desenvolvimento;
3. **Políticas de Trabalho Definidas:** Uma lista de políticas de trabalho foram definidas junto com a equipe e disponibilizada para todos os integrantes;
4. **Melhoria na Comunicação:** A comunicação entre os membros da equipe melhorou após a reunião de execução. O workshop incentivou a troca de informações e a colaboração, promovendo um ambiente de trabalho mais bem definido;
5. **Organização do Processo de Desenvolvimento:** Houve uma melhoria na organização do processo de desenvolvimento. A equipe adotou práticas mais estruturadas, resultando em maior clareza e eficiência na execução das tarefas;
6. **Modelo de Avaliação Heurística:** A Avaliação Heurística foi implementada para avaliar a usabilidade e a eficiência do sistema ao longo das *sprints*. Um modelo e

exemplos foram disponibilizados a equipe para que pudessem desenvolver a avaliação minimizando os erros. Essa atividade ajudou a equipe a identificar problemas no design e usabilidade que não eram evidentes antes da execução do *workshop*, e também gerou novas tarefas para os desenvolvedores trabalharem.

7. **Documentações:** Diversos documentos e materiais auxiliares foram gerados após a aplicação do *workshop*. Alguns constam na seção de apêndices e outros no repositório da equipe, inclusive um dicionário de termos técnicos que foi elaborado por conta de algumas dificuldades percebidas durante o *workshop* em relação a alguns termos de linguajar mais restrito.

Foi observada uma certa resistência por parte dos gestores quanto a algumas políticas que foram definidas, bem como o desrespeito delas por parte de alguns membros da equipe. Abaixo são relatados as dificuldades e limitações identificadas durante o fluxo de desenvolvimento:

1. **Adaptação às Políticas de Trabalho:** Algumas políticas definidas durante o workshop não puderam ser implementadas de imediato devido à necessidade de adaptação da equipe. Isso indica que, embora as políticas sejam valiosas, a transição para novos métodos pode exigir tempo e treinamento adicional. Essas políticas são: *definição de Sprints, Reunião de Planejamento e Review*;
2. **Início de Demandas sem Cadastro no Quadro Kanban:** Foi observado que algumas demandas começaram a ser trabalhadas sem serem previamente cadastradas no quadro Kanban. Esse comportamento foi identificado, corrigido e relatado para a equipe as consequências, podendo gerar falta de visibilidade e controle sobre o progresso das tarefas;
3. **Desconsideração do Limite de WIP (Work In Progress):** O limite de WIP, crucial para evitar sobrecarga de trabalho e melhorar o foco da equipe, não foi inicialmente considerado. Isso foi prontamente relatado para todos os membros da equipe e corrigido, relatando como isso pode resultar em multitarefas e queda na produtividade;
4. **Dificuldade no Cadastro de Cards:** O gerente de projeto em meio a todo o processo de desenvolvimento acabou tendo dificuldades de tempo para poder cadastrar as novas demandas no quadro, tendo esta atividade sido feita então pelos desenvolvedores.
5. **Desafios da Técnica Questionário:** A técnica do questionário, apesar de sua utilidade, mostrou-se demorada para ser aplicada. Embora tenha sido criado um questionário de interesse da empresa, a equipe não conseguiu utilizá-lo devido à

falta de tempo. Este fato evidenciou a necessidade de se planejar melhor o tempo disponível para a aplicação de técnicas mais detalhadas.

## 7.2 Questionário de *feedback*

O objetivo deste questionário foi avaliar a percepção dos participantes sobre diversos aspectos do workshop. Ele foi estruturado com perguntas objetivas e abertas para capturar informações. As respostas foram analisadas qualitativamente e quantitativamente.

**O quadro Kanban foi bem definido e está facilitando a visualização das tarefas e o acompanhamento do progresso?**

5 respostas

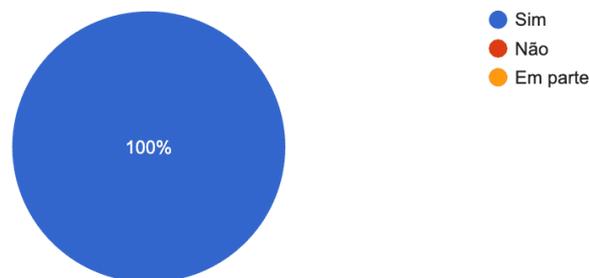


Figura 17 – Resposta do Questionário

**Você considera que o fluxo de trabalho da equipe ficou bem definido após o workshop?**

5 respostas

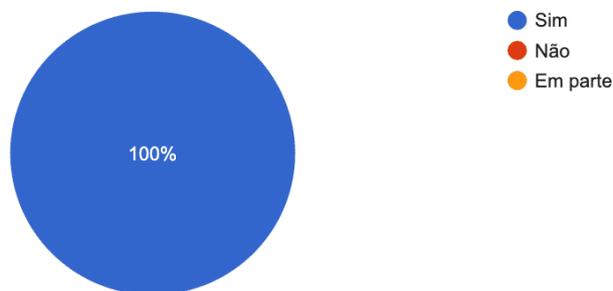


Figura 18 – Resposta do Questionário

**A equipe enfrentou dificuldades para implementar alguma das políticas de trabalho definidas durante o workshop?**

5 respostas

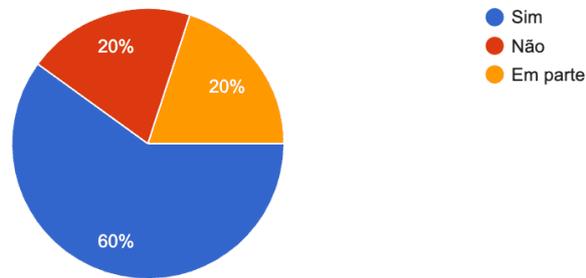


Figura 19 – Resposta do Questionário

Se você respondeu "Sim" à pergunta anterior, por favor, indique quais políticas foram mais difíceis de seguir:

[Copiar](#)

4 respostas

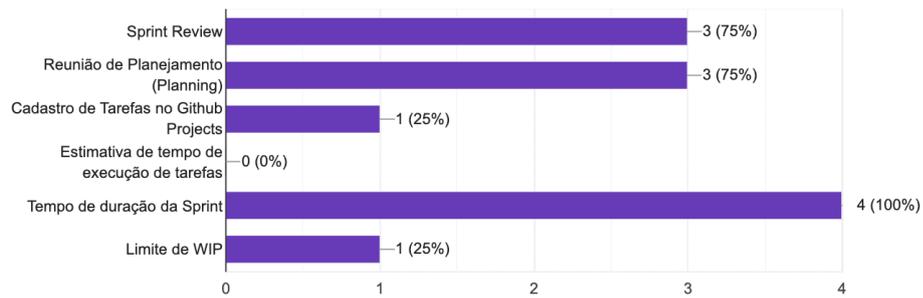


Figura 20 – Resposta do Questionário

**"A comunicação da equipe melhorou."**

O quanto você concorda com a frase acima?

5 respostas

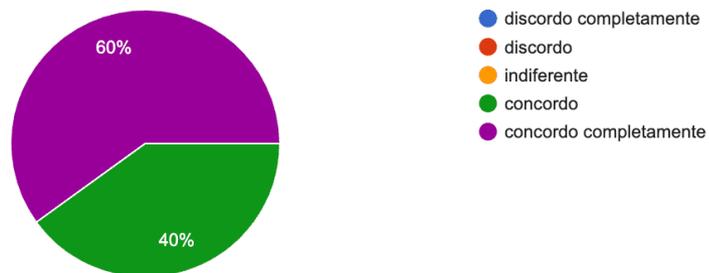


Figura 21 – Resposta do Questionário

**"A organização do processo de desenvolvimento melhorou após o workshop."**

O quanto você concorda com a frase acima?

5 respostas

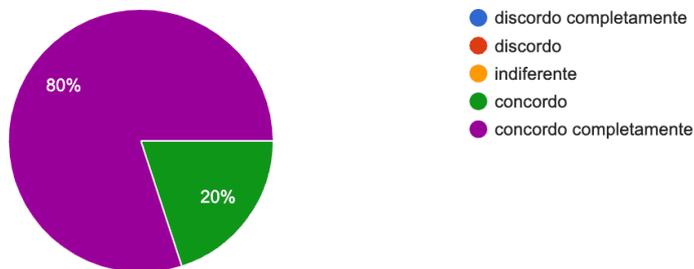


Figura 22 – Resposta do Questionário

**O material com o modelo e as regras para elaboração de questionários está sendo utilizado pela equipe?**

5 respostas

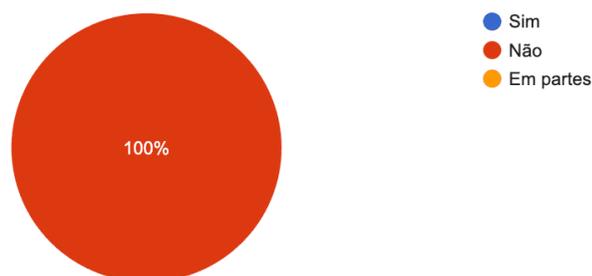


Figura 23 – Resposta do Questionário

**Se você respondeu "Não" ou "Em parte" à pergunta anterior, por favor, relate as dificuldades encontradas:**

4 respostas

- O material ficou bem definido, porém ainda não houve demanda que necessitasse o uso da técnica para as regras serem consultadas.
- No momento ainda não houve demandas que necessitassem do uso de Questionários
- Não teve demanda para ser utilizado o questionário
- Não houve demandas que necessitassem do questionário

Figura 24 – Resposta do Questionário

**Sobre a técnica de Questionário, o quão útil foi o material disponibilizado pelos especialistas do workshop?**

5 respostas

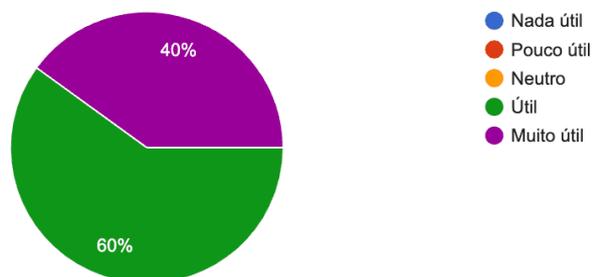


Figura 25 – Resposta do Questionário

**O modelo de Avaliação Heurística foi útil para avaliar a usabilidade e eficiência do sistema?**

5 respostas

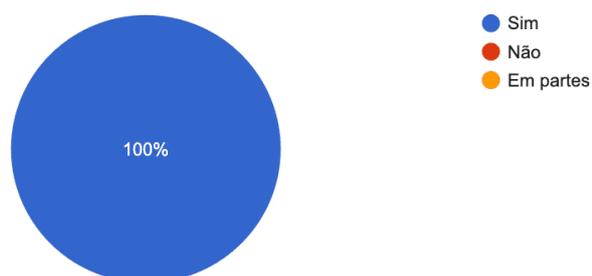


Figura 26 – Resposta do Questionário

**Se você respondeu "Não" ou "Em parte" à pergunta anterior, por favor, relate as dificuldades encontradas:**

0 resposta

Ainda não há respostas para esta pergunta.

Figura 27 – Resposta do Questionário

A configuração do Github Projects para a execução do processo de desenvolvimento foi útil e ajudou a equipe a definir melhor as tarefas?

5 respostas

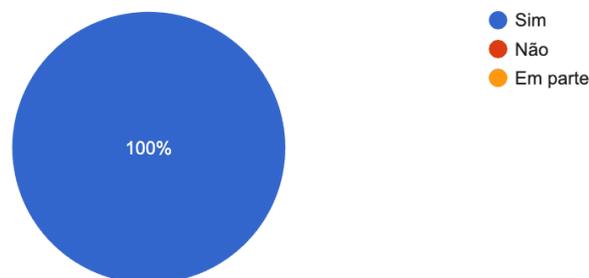


Figura 28 – Resposta do Questionário

Você tem algum comentário sobre algo que não foi mencionado nas perguntas sobre o workshop e o processo de desenvolvimento?

4 respostas

Gostaria de destacar a Análise Heurística que rolou durante o workshop. Foi ótimo saber dessa técnica e esse método nos ajudou a identificar rapidamente alguns problemas em alguns fluxos do nosso aplicativo.

A análise heurística ajudou a equipe a perceber alguns problemas que existiam no app. Isso gerou novas tarefas e foi muito fácil de executar

Em alguns momentos o limite de WIP não foi considerado para os desenvolvedores, que tiveram que trabalhar em algumas demandas urgentes ao mesmo tempo de outras não tao urgentes

O limite de WIP não foi levado em consideração no inicio.

Figura 29 – Resposta do Questionário

Considerando um total de cinco participantes no questionário, os seguintes pontos principais foram observados:

- Os participantes que responderam ao questionário consideraram que houve melhora no acompanhamento do progresso devido ao novo quadro Kanban do projeto e que a comunicação no geral também melhorou de acordo com as Figuras 17, 18 e 21.
- Na Figura 19, foi possível observar que quatro dos cinco participantes tiveram alguma dificuldade para implementar algumas políticas definidas no workshop. Sendo que desses quatro, três tiveram dificuldade para implementar a política de *Sprint Review* e *Sprint Planning*, e todos consideraram que houve dificuldade para seguir o tempo de duração das *Sprints*, como exposto na Figura 20.
- As Figuras 23, 24 e 25 demonstram que apesar de a técnica Questionário e seus materiais serem considerados úteis, ela não foi testada no processo de desenvolvimento do projeto e a equipe ainda não conseguiu validar de fato a técnica.

- Todos os participantes consideraram que a técnica Avaliação Heurística e a configuração do quadro Kanban foram úteis para o projeto, como mostrado nas Figura 26 e 28. Além disso, houve um relato na Figura 29 de que surgiram novas tarefas a partir de análises feitas com a técnica Avaliação Heurística.
- A Figura 29 mostra que dois participantes relataram que o limite de WIP não foi levado em consideração em alguns casos.

### 7.3 Discussão dos Resultados

A realização do *workshop* trouxe uma série de impactos positivos e desafiadores para a equipe da startup. Os resultados indicam que algumas das estratégias adotadas se mostraram bem consolidadas. Já em outras, evidenciou-se a necessidade de um ajustamento gradual às novas práticas e métodos introduzidos, embora existam também alguns obstáculos que têm de ser continuamente tratados, conforme previsto no *workshop*.

A clareza no novo fluxo de trabalho surge como o primeiro grande avanço da aplicação do *workshop*, e é seguida pela configuração do quadro Kanban, que se mostra satisfatória do ponto de vista da equipe. Ser capaz de ver claramente o fluxo de trabalho ajudou a equipe a adaptar suas práticas ao que o projeto exigia, elevando assim a eficiência da equipe. O quadro Kanban, uma representação visual das tarefas a serem realizadas diariamente, foi útil para representar de forma clara o progresso, a rastreabilidade e identificar áreas onde os processos de desenvolvimento poderiam estar embargados ou onde havia tarefas redundantes, além de todos os membros serem capazes de consultar a qualquer momento as tarefas e seus respectivos status, sem precisar de um contato direto com a equipe de desenvolvimento.

Os documentos produzidos tiveram um papel importante no fornecimento de mais recursos à equipe. Esses materiais auxiliaram na padronização de processos e no esclarecimento de terminologias técnicas, o que auxiliou na facilitação da comunicação e na execução das atividades propostas.

A mudança e a implementação da proposta inicialmente não foi fácil, já que a equipe precisou seguir uma série de políticas e um fluxo totalmente novo, tirando assim todos os membros de uma zona totalmente diferente. A oposição inicial dos gestores a determinados protocolos de trabalho mostraram a necessidade de um período maior de adaptação, bem como a reestruturação e remoção temporária de algumas políticas, com o objetivo de não tornar o processo burocrático do ponto de vista da empresa e por consequência a rejeição da proposta. A aplicação direta de políticas, como a criação de sprints ou reuniões de planejamento e revisão enfrentou obstáculos, pois a equipe mostrou que precisava de uma adaptação em fases, uma vez que a equipe teve dificuldade em implantar essas e algumas outras alterações de forma imediata.

A identificação das demandas feitas antes de qualquer registro no quadro Kanban, bem como a falta de consideração pelo limite de WIP revelam áreas com requisitos de atenção significativos. Embora essas práticas tenham sido pontuadas pelos especialistas e corrigidas pela equipe, indicam a necessidade de monitoramento contínuo para garantir a adesão a todos os processos estabelecidos durante o *workshop*.

Os obstáculos enfrentados pelo gestor do projeto no registro dos cartões e os desafios encontrados na implementação do método do questionário enfatizam a necessidade de, respectivamente, repensar a estratégia quanto ao cadastro de *cards* e outras atividades ligadas ao quadro Kanban na ausência do gestor, e na viabilidade da técnica Questionário, cogitando até uma alteração da técnica escolhida.

## 8 Conclusão

O desenvolvimento deste estudo possibilitou uma análise de como incorporar atividades centradas no usuário no desenvolvimento de software do projeto *Groovoo* por meio do *workshop Kick-start Kanban*. Além disso, permitiu a investigação de técnicas focadas no usuário que geram valor para a *startup*, buscando a satisfação do usuário e a correção de limitações e deficiências no fluxo de trabalho atual.

Para alcançar esse objetivo, foram analisadas e selecionadas abordagens de UCD em colaboração com o gestor do projeto, que contribuiu ativamente com o trabalho. Posteriormente, foi proposta e executada a implementação do *Workshop Kick-start Kanban* com base nessas abordagens.

Durante a execução do *workshop*, foram observados resultados que destacaram tanto os impactos positivos quanto os desafios enfrentados pela equipe da *startup* na adaptação do processo de desenvolvimento para envolver mais os usuários. O fluxo de trabalho da equipe foi definido e ajustado durante o *workshop*, permitindo adaptações necessárias às práticas específicas do projeto. A equipe estabeleceu um novo quadro Kanban mais organizado, facilitando a visualização das tarefas, o acompanhamento do progresso e a identificação de gargalos. A comunicação entre os membros da equipe melhorou, conforme relatados pelos mesmos, após a reunião de execução, promovendo um ambiente de trabalho mais coeso e eficiente. Houve uma melhoria geral na organização do processo de desenvolvimento, resultando em maior clareza e eficiência na execução das tarefas.

No entanto, algumas políticas definidas durante o *workshop* não puderam ser implementadas imediatamente devido à necessidade de adaptação da equipe, indicando a importância de tempo e treinamento adicional. Isso fica evidente nos resultados quando existe uma certa resistência no uso da técnica Questionário que, embora útil, mostrou-se demorada para ser aplicada regularmente e também mostrou-se não ser prioritária no momento devido ao tempo escasso. Essa resistência por parte da equipe pode ser justificada pela mudança em adotar um processo que envolvia muitas etapas (novas) em relação ao que estavam acostumados.

Outro ponto que indica a necessidade de tempo para a adaptação ao novo processo é a observação de que algumas demandas começaram a ser trabalhadas sem serem cadastradas no quadro Kanban, o que pode gerar dificuldade de visibilidade no trabalho atual e um menor controle sobre o progresso das tarefas. Esse desvio do processo estabelecido pode resultar em várias consequências negativas, como a duplicação de esforços, a perda de prioridades e a incapacidade de identificar gargalos no fluxo de trabalho.

Portanto, no contexto de adaptação de um novo processo de desenvolvimento para uma *startup*, o *workshop Kick-start Kanban* se mostra um grande aliado devido a sua estrutura adaptável e iterativa. Essa abordagem permite identificar erros na execução anterior e aprimorá-los em iterações subsequentes. Além disso, a adição de técnicas de Design Centrado no Usuário (UCD) pode ser um diferencial nas próximas iterações já que o *workshop* proporciona um ambiente propício para a introdução e refinamento de novas estruturas.

Dada a importância do tema, é necessário continuar desenvolvendo formas de agilizar e melhorar o processo de desenvolvimento centrado no usuário, especialmente em *startups* que ainda não adotaram práticas estruturadas. Para futuras pesquisas, recomenda-se aplicar uma segunda rodada do *kick-start Kanban* e implementar a quarta etapa do *workshop*. Isso permitirá uma avaliação constante e a realização de ajustes necessários para melhorar ainda mais o processo e garantir que as práticas adotadas tragam benefícios constantes, garantindo a base de melhoria contínua.

## Referências

- ACHOUANTZ, C.; NORDIN, J. *The Kanban Kick-start Field Guide*. [S.l.], 2013. Citado 15 vezes nas páginas 10, 11, 18, 19, 29, 30, 31, 32, 33, 34, 35, 36, 38, 39 e 42.
- AGUIRRE, J.; MOQUILLAZA, A.; PAZ, F. A user-centered framework for the design of usable atm interfaces. In: MARCUS, A.; WANG, W. (Ed.). *Design, User Experience, and Usability. Design Philosophy and Theory*. Cham: Springer International Publishing, 2019. p. 163–178. ISBN 978-3-030-23570-3. Citado na página 48.
- ALMUGHRAM, O.; ALYAHYA, S. Coordination support for integrating user centered design in distributed agile projects. In: *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*. [S.l.: s.n.], 2017. p. 229–238. Citado na página 18.
- ANDERSON, D. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010. ISBN 9780984521401. Disponível em: <<https://books.google.com.br/books?id=RJ0VUkfUWZkC>>. Citado na página 30.
- ANITHA, P.; PRABHU, B. Integrating requirements engineering and user experience design in product life cycle management. In: IEEE. *2012 First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)*. [S.l.], 2012. p. 12–17. Citado na página 45.
- ARGUMANIS, D.; MOQUILLAZA, A.; PAZ, F. Challenges in integrating scrum and the user-centered design framework: A systematic review. In: \_\_\_\_\_. [S.l.: s.n.], 2020. p. 52–62. ISBN 978-3-030-66918-8. Citado na página 68.
- ARGUMANIS, D.; MOQUILLAZA, A.; PAZ, F. A framework based on ucd and scrum for the software development process. In: SOARES, M. M.; ROSENZWEIG, E.; MARCUS, A. (Ed.). *Design, User Experience, and Usability: UX Research and Design*. Cham: Springer International Publishing, 2021. p. 15–33. ISBN 978-3-030-78221-4. Citado 7 vezes nas páginas 11, 50, 51, 52, 68, 71 e 73.
- AYALA, L. E. V.; URIBE, E. H. Del manifiesto ágil sus valores y principios. *Scientia et technica*, Universidad Tecnológica de Pereira, v. 2, n. 34, p. 381–385, 2007. ISSN 0122-1701. Citado 3 vezes nas páginas 23, 25 e 26.
- BANIJAMALI, A. et al. Empirical investigation of scrumban in global software development. In: \_\_\_\_\_. [S.l.: s.n.], 2017. p. 229–248. ISBN 978-3-319-66301-2. Citado na página 18.
- BARBOZA, L. F. et al. Análise comparativa entre as abordagens ágil e tradicional de gestão de projetos: Um estudo de caso no setor industrial. In: PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS. *V SINGEP SIMPÓSIO INTERNACIONAL DE GESTÃO DE PROJETOS, INOVAÇÃO E SUSTENTABILIDADE*. [S.l.], 2016. Citado na página 17.
- BECK, K. et al. *Manifesto for Agile Software Development*. 2001. Disponível em: <<https://agilemanifesto.org/>>. Citado 5 vezes nas páginas 11, 23, 24, 25 e 53.

BERTOLINI DAVIDE MEZZOGORI, M. N. M.; ZAMMORI, F. A scrumban board-based approach to improve material flow in engineering to order (eto) companies: an industrial application based on action research. *Production Planning & Control*, Taylor Francis, v. 0, n. 0, p. 1–23, 2023. Disponível em: <<https://doi.org/10.1080/09537287.2023.2248940>>. Citado na página 26.

BHAVSAR, K.; GOPALAN, S.; SHAH, V. Scrumban: An agile integration of scrum and kanban in software engineering. In: . [S.l.: s.n.], 2020. v. 9. Citado 3 vezes nas páginas 10, 26 e 27.

BLANK, S. *The startup owner's Manual: The Step-by-step guide for building a great company*. [S.l.]: K amp; S Ranch Press, 2012. Citado na página 44.

BLANK, S. *The Four Steps to the Epiphany: Successful Strategies for Products that Win*. Wiley, 2020. ISBN 9781119690375. Disponível em: <<https://books.google.com.br/books?id=6yrfDwAAQBAJ>>. Citado na página 44.

BOUGROUN, Z.; ZEAARAOUI, A.; BOUCHENTOUF, T. The projection of the specific practices of the third level of cmmi model in agile methods: Scrum, xp and kanban. In: *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*. [S.l.: s.n.], 2014. p. 174–179. Citado na página 17.

BRERETON, P. et al. Using a protocol template for case study planning. In: *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*. Swindon, GBR: BCS Learning & Development Ltd., 2008. (EASE'08), p. 41–48. Citado 4 vezes nas páginas 11, 21, 56 e 57.

CARVALHO, A. *Introdução à engenharia de software*. Ed. da Unicamp, 2001. (Títulos em Engenharia de Software). ISBN 9788526805323. Disponível em: <<https://books.google.com.br/books?id=uRnlZwEACAAJ>>. Citado na página 25.

CARVALHO, B. V.; MELLO, C. H. P. Revisão, análise e classificação da literatura sobre o método de desenvolvimento de produtos ágil scrum. In: *Anais do XII Simpósio de Administração da Produção, Logística e Operações Internacionais - SIMPOI*. São Paulo/SP: [s.n.], 2009. Citado na página 17.

CARVALHO, B. V. d.; MELLO, C. H. P. Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica. *Gestão Produção*, Universidade Federal de São Carlos, v. 19, n. 3, p. 557–573, 2012. ISSN 0104-530X. Disponível em: <<https://doi.org/10.1590/S0104-530X2012000300009>>. Citado na página 63.

CHOMA, J. et al. Influences of ux factors in the agile ux context of software startups. *Information and Software Technology*, v. 152, p. 107041, 08 2022. Citado na página 18.

DINGSØYR, T. et al. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, v. 85, n. 6, p. 1213–1221, 2012. ISSN 0164-1212. Special Issue: Agile Development. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0164121212000532>>. Citado 2 vezes nas páginas 25 e 26.

DOVE, R.; FORUM, A. *Tools for Analyzing and Constructing Agile Capabilities*. Agility Forum, 1996. (Perspectives on Agility Series). ISBN 9781885166081. Disponível em: <<https://books.google.com.br/books?id=oSFNAAAACAAJ>>. Citado na página 25.

FREEMAN, R.; MCVEA, J. A stakeholder approach to strategic management. *SSRN Electronic Journal*, 01 2001. Citado na página 63.

GIL, A. *Como elaborar projetos de pesquisa*. Atlas, 2002. ISBN 9788522431694. Disponível em: <<https://books.google.com.br/books?id=X4uvAAAACAAJ>>. Citado 2 vezes nas páginas 20 e 21.

GOLDMAN, S. L.; PREISS, K.; NAGEL, R. N. *Agile Competitors and Virtual Organizations*. [s.n.], 1995. Disponível em: <[http://www.alibris.co.uk/booksearch?qwork=170399&matches=16&author=Goldman%2C+Steven+L.&browse=1&cm\\_sp=works\\*listing\\*title](http://www.alibris.co.uk/booksearch?qwork=170399&matches=16&author=Goldman%2C+Steven+L.&browse=1&cm_sp=works*listing*title)>. Citado na página 25.

GOMES, T. L. d. S.; LUCAS, V. S. d. S. *Desenvolvimento Centrado no Usuário para um processo Scrumban: um estudo de caso*. 108 p. — Faculdade do Gama, Universidade de Brasília, Brasília, 2023. Citado 6 vezes nas páginas 10, 19, 62, 64, 66 e 67.

GREGORY, P. et al. (Ed.). *Agile Processes in Software Engineering and Extreme Programming: 22nd international conference on agile software development, xp 2021, virtual event, june 14–18, 2021, proceedings*. 1. ed. [S.l.]: Springer Cham, 2021. X, 211 p. (Lecture Notes in Business Information Processing). ISBN 978-3-030-78097-5. Citado na página 18.

GUIMARÃES, F. M. *O que é Design Centrado no Usuário?* 2019. Disponível em: <<https://medium.com/aela/o-que-%C3%A9-design-centrado-no-usu%C3%A1rio-11a9c13c3a2f>>. Citado na página 47.

HART, M. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses* eric ries. new york: Crown business, 2011. 320 pages. us26.00. *Journal of Product Innovation Management*, v. 29, 052011. Citado na página 44.

HOKKANEN, L.; KUUSINEN, K.; VÄÄNÄNEN, K. Early product design in startups: Towards a ux strategy. In: ABRAHAMSSON, P. et al. (Ed.). *Product-Focused Software Process Improvement*. Cham: Springer International Publishing, 2015. p. 217–224. ISBN 978-3-319-26844-6. Citado na página 18.

IKONEN, M. et al. Exploring the sources of waste in kanban software development projects. In: *2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications*. [S.l.: s.n.], 2010. p. 376–381. Citado na página 17.

ISO 13407. *Human-centred design processes for interactive systems*. Geneva, CH, 1999. Disponível em: <<https://www.sis.se/api/document/preview/615069/>>. Citado 2 vezes nas páginas 45 e 53.

KHAN, Z. A. *Scrumban - Adaptive Agile Development Process*. 2014. Citado 4 vezes nas páginas 26, 27, 28 e 29.

KNIBERG, H. *Kanban vs. scrum: How to make the most of both*. 2009. Citado na página 17.

- KUREK, J. et al. Sustainable Business Models Innovation and Design Thinking: A Bibliometric Analysis and Systematic Review of Literature. *Sustainability*, v. 15, n. 2, 2023. ISSN 2071-1050. Disponível em: <<https://www.mdpi.com/2071-1050/15/2/988>>. Citado na página 47.
- LADAS, C. *crumban: Essays On Kanban Systems For Lean Software Development*. [S.l.: s.n.], 2008. Citado 2 vezes nas páginas 27 e 28.
- LUCASSEN, G. et al. The use and effectiveness of user stories in practice. In: . [S.l.: s.n.], 2016. p. 205–222. ISBN 978-3-319-30281-2. Citado na página 54.
- MAFAKHERI, F. N. F.; MOUSAVI, M. Project agility assessment: an integrated decision analysis approach. *Production Planning & Control*, Taylor Francis, v. 19, n. 6, p. 567–576, 2008. Disponível em: <<https://doi.org/10.1080/09537280802360884>>. Citado na página 25.
- MARTIN, A.; ANSLOW, C.; JOHNSON, D. Teaching agile methods to software engineering professionals: 10 years, 1000 release plans. In: . [S.l.: s.n.], 2017. p. 151–166. ISBN 978-3-319-57632-9. Citado na página 17.
- MCINERNEY, P.; MAURER, F. Ucd in agile projects: Dream team or odd couple? *Interactions*, Association for Computing Machinery, New York, NY, USA, v. 12, n. 6, p. 19–23, nov 2005. ISSN 1072-5520. Disponível em: <<https://doi.org/10.1145/1096554.1096556>>. Citado na página 18.
- MILLER, G. Agile problems, challenges, failures. In: . [S.l.: s.n.], 2013. Citado na página 17.
- NELSON, S. D. et al. Software prototyping a case report of refining user requirements for a health information exchange dashboard. *Applied clinical informatics*, Thieme Medical Publishers, STUTTGART, v. 7, n. 1, p. 22–32, 2016. ISSN 1869-0327. Citado na página 49.
- NGUYEN-DUC, A.; WANG, X.; ABRAHAMSSON, P. What influences the speed of prototyping? an empirical investigation of twenty software startups. In: BAUMEISTER, H.; LICHTER, H.; RIEBISCH, M. (Ed.). *Agile Processes in Software Engineering and Extreme Programming*. Cham: Springer International Publishing, 2017. p. 20–36. ISBN 978-3-319-57633-6. Citado na página 44.
- NIKITINA, N.; KAJKO-MATTSSON, M.; STRALE, M. From scrum to scrumban: a case study of a process transition. *2012 International Conference on Software and System Process, ICSSP 2012 - Proceedings*, 06 2012. Citado na página 17.
- NOGUERRA, C. User-centered design principles for effective information system development. *International Journal of Advanced Research in Science, Communication and Technology*, p. 826–830, 07 2023. Citado na página 45.
- NORMAN, D. A. *The design of everyday things*. [S.l.]: MIT Press, 1988. Citado na página 44.
- NORMAN, D. A. *The design of everyday things*. [S.l.: s.n.], 2013. Citado 6 vezes nas páginas 10, 45, 46, 48, 50 e 54.

- NORMAN, D. A.; DRAPER, S. W. *User centered system design : new perspectives on human-computer interaction*. [S.l.: s.n.], 1986. Citado na página 44.
- NOVOSELTSEVA, E. *User-Centered Design: An Introduction*. 2017. Section: Terminology. Disponível em: <<https://usabilitygeek.com/user-centered-design-introduction/>>. Citado na página 45.
- OLIVEIRA, J. C. P. D. et al. O questionário, o formulário e a entrevista como instrumentos de coleta de dados: vantagens e desvantagens do seu uso na pesquisa de campo em ciências humanas. In: *Anais III CONEDU*. Campina Grande: Realize Editora, 2016. Acesso em: 12/04/2024 11:34. Disponível em: <<https://www.editorarealize.com.br/artigo/visualizar/21719>>. Citado na página 51.
- OLIVEIRA, L. R. d.; GOMES, G. S.; LIMA, F. P. d. Análise de riscos pelo uso de métodos Ágeis na gestão de projetos de desenvolvimento de software. *Revista de Gestão e Projetos*, v. 5, n. 2, p. 90–101, ago. 2014. Disponível em: <<https://periodicos.uninove.br/gep/article/view/9598>>. Citado na página 18.
- PANTIUCHINA, J. et al. Are software startups applying agile practices? the state of the practice from a large survey. In: . [S.l.: s.n.], 2017. p. 167–183. ISBN 978-3-319-57632-9. Citado na página 17.
- PATERNOSTER, N. et al. Software development in startup companies: A systematic mapping study. *Information and Software Technology*, v. 56, 10 2014. Citado na página 44.
- PATTERSON, M. et al. Towards an Agile User Experience Virtual Assistant and Management Platform. p. 1, jul. 2017. Publisher: BCS Learning & Development. Disponível em: <<https://www.scienceopen.com/hosted-document?doi=10.14236/ewic/HCI2017.77>>. Citado na página 18.
- QUINTANA, V.; HOWELLS, R.; HETTINGER, L. User-centered design in a large-scale naval ship design program: Usability testing of complex military systems—ddg 1000. *Naval Engineers Journal*, v. 119, p. 25 – 33, 10 2007. Citado na página 45.
- QUIÑONES, D.; RUSU, C. Applying a methodology to develop user experience heuristics. *Computer Standards Interfaces*, v. 66, p. 103345, 2019. ISSN 0920-5489. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0920548919300303>>. Citado na página 48.
- REDDY, A. *The Scrumban [R]Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban*. 1st. ed. [S.l.]: Addison-Wesley Professional, 2015. ISBN 013408621X. Citado 3 vezes nas páginas 27, 28 e 29.
- RITTER, F.; BAXTER, G.; CHURCHILL, E. User-centered systems design: A brief history. In: \_\_\_\_\_. [S.l.: s.n.], 2014. p. 33–54. ISBN 978-1-4471-5133-3. Citado na página 18.
- SAAD, J. et al. The impact of ux work on communication and collaboration in software startups. In: *Anais do VIII Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software*. Porto Alegre, RS, Brasil: SBC, 2023. p. 41–50. ISSN 2763-874X. Disponível em: <<https://sol.sbc.org.br/index.php/washes/article/view/24774>>. Citado na página 18.

- SALAH, D.; PAIGE, R.; CAIRNS, P. Patterns for integrating agile development processes and user centred design. In: . [S.l.: s.n.], 2015. p. 1–10. Citado na página 44.
- SALINAS, E.; CUEVA, R.; PAZ, F. A systematic review of user-centered design techniques. In: MARCUS, A.; ROSENZWEIG, E. (Ed.). *Design, User Experience, and Usability. Interaction Design*. Cham: Springer International Publishing, 2020. p. 253–267. ISBN 978-3-030-49713-2. Citado 3 vezes nas páginas 11, 48 e 49.
- SALMINEN, J. et al. Use cases for design personas: A systematic review and new frontiers. In: . [S.l.: s.n.], 2022. p. 1–21. Citado na página 45.
- Sandvik Group. *Quem somos*. 2023. Acessado em 12 de outubro de 2023. Disponível em: <<https://www.home.sandvik/br/quem-somos/>>. Citado na página 30.
- SCHWABER, K. SCRUM Development Process. In: SUTHERLAND, J. et al. (Ed.). *Business Object Design and Implementation*. London: Springer London, 1997. p. 117–134. ISBN 978-3-540-76096-2 978-1-4471-0947-1. Disponível em: <[http://link.springer.com/10.1007/978-1-4471-0947-1\\_11](http://link.springer.com/10.1007/978-1-4471-0947-1_11)>. Citado na página 17.
- SCHÖN, E.-M. et al. Key challenges in agile requirements engineering. In: . [S.l.: s.n.], 2017. p. 37–51. ISBN 978-3-319-57632-9. Citado na página 17.
- SCHÖN, E.-M. et al. Enterprise experience into the integration of human-centered design and kanban. In: . [S.l.: s.n.], 2016. Citado 4 vezes nas páginas 10, 52, 53 e 54.
- SIGNORETTI, I. et al. Boosting agile by using user-centered design and lean startup: A case study of the adoption of the combined approach in software development. In: . [S.l.: s.n.], 2019. p. 1–6. Citado na página 44.
- SILVA, T. S. da et al. Understanding the ux designer’s role within agile teams. In: MARCUS, A. (Ed.). *Design, User Experience, and Usability. Design Philosophy, Methods, and Tools*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 599–609. ISBN 978-3-642-39229-0. Citado na página 18.
- SUTHERLAND, J. *Scrum: a arte de fazer o dobro do trabalho na metade do tempo*. Leya Brasil, 2016. ISBN 9788544104712. Disponível em: <<https://books.google.com.br/books?id=Q1G2DQAAQBAJ>>. Citado na página 25.
- UHLENBROK, J. et al. User experience in kanban. In: \_\_\_\_\_. [S.l.: s.n.], 2015. p. 84–94. ISBN 9783110443882. Citado na página 52.
- UNTERKALMSTEINER, M. et al. Software startups - a research agenda. *E-Informatica Software Engineering Journal*, v. 2016, p. 89–124, 10 2016. Citado na página 44.
- VENSON, E. et al. *Experiência do usuário e engenharia de software: interações, atividades e produtos: relatório técnico*. [S.l.], 2022. 32 p. Disponível em: <<https://repositorio.unb.br/handle/10482/43289>>. Citado na página 45.
- WINTER, D. et al. User experience in kanban. In: . [S.l.: s.n.], 2013. Citado 2 vezes nas páginas 53 e 54.
- ZORZETTI, M. et al. Improving agile software development using user-centered design and lean startup. *Information and Software Technology*, v. 141, p. 106718, 08 2021. Citado na página 44.

# Anexos

# Apêndices

# APÊNDICE A – Carta de solicitação de acesso e citação



UNIVERSIDADE DE BRASÍLIA - UNB  
FACULDADE DO GAMA - FGA  
CURSO DE ENGENHARIA DE SOFTWARE

A Atena Soluções Informática Ltda. pessoa jurídica devidamente inscrita no CNPJ nº 08.189.785/0001-02, com endereço Rua da Imprensa 560, bairro Ipiranga, na cidade de São Paulo, Estado de São Paulo, telefone para contato (11)97070-1925 neste ato representada por seu responsável legal Frederico José Pelúcio de Castro, profissão administrador, portador da R.G nº 63137154-01 e inscrito no CPF/MF 057648118-13 por intermédio da presente autoriza a realização, em suas dependências e fora delas, do Projeto de Pesquisa do trabalho de conclusão de curso intitulado **Kick-start Kanban e Design Centrado no Usuário para um Processo de Software de uma Startup** que tem por objetivo analisar a inserção de procedimentos de *UX Design* no modelo de processo de software da empresa. Autoriza expressamente a divulgação da pesquisa, do nome da empresa, dos modelos de processo, dados e do resultado. Declara que tem conhecimento e que concorda plenamente que a participação da empresa que representa se dá à título gratuito, não recebendo, portanto nenhum honorário ou gratificação referente ao projeto de pesquisa. Concorda com a possibilidade de as informações relacionadas ao estudo serem inspecionadas pelo orientador da pesquisa.

São Paulo, dia 15 de Junho de 2024.

-----  
(Assinatura)

Empresa: Atena Soluções Informática Ltda  
Nome completo do responsável legal: Frederico José Pelúcio de Castro  
Telefone p/ contato: (11)97070-1925  
CPF/MF: 057648118-13

## APÊNDICE B – Regras para montagem do Questionário

- As perguntas devem ser claras, simples e precisas para todos os entrevistados. Deve ir direto ao ponto. Isso é essencial para garantir resultados confiáveis entre os diferentes participantes.
- Evitar ambiguidade quanto às perguntas apresentadas para os entrevistados.
- Evite perguntas emocionalmente carregadas, a fim de evitar preconceitos.
- Busque transmitir um pensamento de cada vez.
- Utilizar um linguajar mais simples possível.
- O processo de desenvolvimento do questionário deve atender-se a diferentes pontos: tipo de aplicação (presencial, virtual, autopreenchido), níveis de alfabetização dos entrevistados, taxas de respostas esperadas e características do público alvo.
- Apresentar o questionário com um design visual amigável e adequado, principalmente os autopreenchidos
- Caso necessário, apresentar instruções adequadas para os usuários que irão preencher
- Não faça mais de uma pergunta na mesma questão. Isso pode confundir o entrevistado e gerar inconsistência nos dados coletados. Caso precise de uma explicação acerca de uma resposta dada anteriormente, abra uma nova pergunta para isso. Deixar tudo em uma questão apenas pode fazer com que o usuário não dê uma explicação.
- Ropa e Rani (2012) - descreve três aspectos principais de um questionário que são, em geral, sequência das perguntas, formulação e redação das perguntas.
- A sequência das perguntas é importante para o entrevistado não se confundir. Crie uma sequência adequada. (Não comece falando de banana e na pergunta seguinte falando de maçã. Seja razoável). A sequência de perguntas dentro de um questionário desempenha um papel crucial para garantir a clareza e coerência. O sequenciamento adequado minimiza a probabilidade de mal-entendidos. Para alcançar isto, opte por perguntas mais diretas e menos invasivas, optando por uma progressão geral consultas específicas
- Para aplicação de questionário online automatizado, ou seja, sem um entrevistador, opte por questionários estruturados. Questionários estruturados consistem em perguntas definidas, concretas e pré-determinadas apresentadas uniformemente a todos os entrevistados.
- Pré-teste do Questionário: Realizar um pré-teste (ou piloto) do questionário com um pequeno grupo de pessoas que representem o público-alvo pode ajudar a identificar e corrigir problemas de compreensão ou redação antes da aplicação em larga escala.
- Variedade de Tipos de Perguntas: Embora as perguntas estruturadas sejam recomendadas, incorporar uma variedade de tipos de perguntas (por exemplo, perguntas abertas, escalas Likert, múltipla escolha) pode proporcionar dados mais ricos e permitir uma análise mais profunda.
- Neutralidade na Redação das Perguntas: Além de evitar perguntas emocionalmente carregadas, assegure-se de que as perguntas sejam neutras e não induzam a respostas específicas. Isso ajuda a reduzir o viés e a obter respostas mais genuínas.

- **Motivação para Participação:** Considere oferecer incentivos ou explicar a importância da pesquisa para aumentar a taxa de resposta. Um simples agradecimento ou a explicação de como os dados serão utilizados pode ser motivador.
- **Equilíbrio na Duração:** Certifique-se de que o questionário não seja muito longo, para evitar a fadiga dos entrevistados, o que pode levar a respostas menos precisas. Um questionário ideal deve ser curto o suficiente para manter o interesse, mas abrangente o suficiente para coletar os dados necessários.
- **Feedback dos Participantes:** Inclua uma seção opcional ao final do questionário para que os participantes possam fornecer feedback sobre a experiência de preenchimento. Isso pode fornecer insights valiosos para futuras pesquisas.

# APÊNDICE C – Exemplos de perguntas do Questionário

## Execução – Exemplos de perguntas

### Questionário sobre o Uso de Sistema de Check-in de Tickets com NFC

5. Quais dificuldades você encontrou ao utilizar o sistema de check-in de tickets com NFC? (Marque todas as que se aplicam)

- Não consegui ativar o NFC no meu celular
- O leitor de NFC não funciona corretamente
- Demorou muito para reconhecer o ticket
- Problemas com a conexão de internet
- Não tive nenhuma dificuldade
- Outros (por favor, especifique): \_\_\_\_\_

Fonte: Os autores

## Execução – Exemplos de perguntas

### Questionário sobre o Uso de Sistema de Check-in de Tickets com NFC

11. Você recomendaria o uso do check-in com NFC na Groovoo para outras pessoas?

- Sim
- Não
- Nunca utilizei

18. Gostaria de fornecer feedback sobre o design ou usabilidade do questionário? (Pergunta aberta) \_\_\_\_\_

Fonte: Os autores

## **Execução – Perguntas que devem ser evitadas**

### **Exemplo 1**

- **Você achou a experiência de usar o sistema de check-in boa?**

### **Problema:**

- **A pergunta é ambígua porque "boa" pode ter diferentes significados para diferentes pessoas. Não especifica claramente quais aspectos da experiência estão sendo avaliados.**

Fonte: Os autores

## **Execução – Perguntas que devem ser evitadas**

### **Exemplo 2**

- **Você conseguiu ativar o NFC facilmente e o check-in foi rápido?**

### **Problema:**

- **Esta pergunta combina duas perguntas em uma, o que pode confundir o entrevistado. Se a resposta for "não", não fica claro a qual parte da pergunta se refere.**

Fonte: Os autores

## **Execução – Perguntas que devem ser evitadas**

### **Exemplo 3**

- **Você conseguiu realizar a leitura do seu ingresso utilizando o Near Field Communication sem nenhuma intercorrência técnica?**

### **Problema:**

- **A linguagem é técnica e complicada, o que pode ser difícil de entender para entrevistados que não são familiarizados com o termo "Near Field Communication" ou "intercorrência técnica".**

Fonte: Os autores

## **Execução – Perguntas que devem ser evitadas**

### **Exemplo 4**

- **Considerando todas as vezes que você usou tecnologia NFC em diversos contextos, incluindo check-in de eventos, pagamentos móveis e acesso a transporte público, como você avaliaria sua experiência geral com essa tecnologia em termos de facilidade de uso, confiabilidade e satisfação?**

### **Problema:**

- **A pergunta é muito longa e cobre muitos aspectos ao mesmo tempo, o que pode ser cansativo e confuso para o entrevistado responder.**

Fonte: Os autores

## **Execução – Perguntas que devem ser evitadas**

### **Exemplo 5**

- Como você classificaria sua experiência com o sistema?

### **Problema:**

- Falta de instruções claras sobre como responder à pergunta (por exemplo, usando uma escala de 1 a 5).

Fonte: Os autores

## **Execução – Perguntas que devem ser evitadas**

### **Exemplo 6**

- Você prefere o sistema de check-in com NFC ao invés dos métodos tradicionais porque é mais rápido e eficiente, certo?

### **Problema:**

- A pergunta é direcionada e sugere a resposta, o que pode introduzir viés.

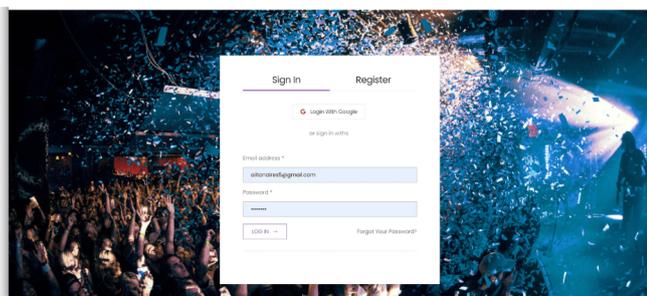
Fonte: Os autores

# APÊNDICE D – Avaliação Heurística com a Equipe

## Execução – Heurísticas

Cenário: compra de tickets na plataforma Groovoo

### 1. Visibilidade do status do sistema



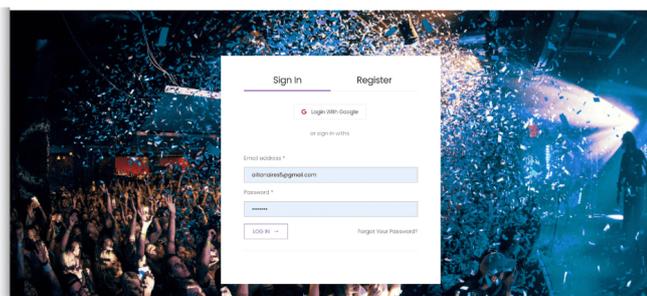
O sistema deve manter os usuários informados sobre o que está acontecendo, através de feedback apropriado e oportuno em relação às ações do usuário e ao estado do sistema.

Fonte: Os autores

## Execução – Heurísticas

Cenário: compra de tickets na plataforma Groovoo

### 1. Visibilidade do status do sistema



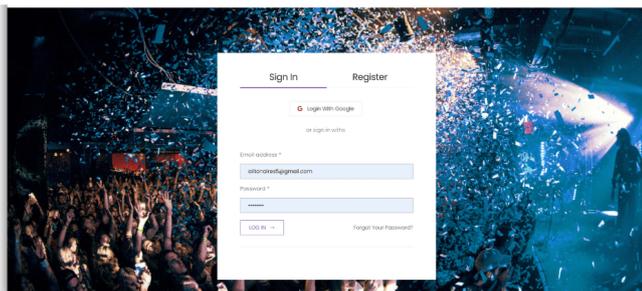
Ao clicar em fazer login, nenhum componente de loading ou informação é claramente apresentada ao usuário sobre a sua requisição para efetuar login.

Fonte: Os autores

## Execução – Heurísticas

Cenário: compra de tickets na plataforma Groovoo

### 1. Visibilidade do status do sistema



**Classificação: moderada**

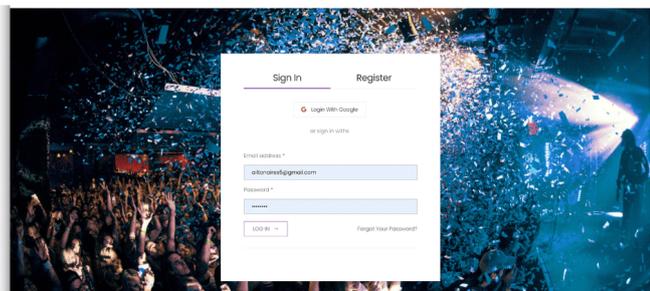
Caso a requisição demore, o usuário pode compreender que houve um erro, quando na verdade não necessariamente isso ocorreu.

Fonte: Os autores

## Execução – Heurísticas

Cenário: compra de tickets na plataforma Groovoo

### 1. Visibilidade do status do sistema



**Sugestão**

apresentar um componente informativo em linguagem verbal ou não-verbal de carregamento da requisição.

**Exemplo:**

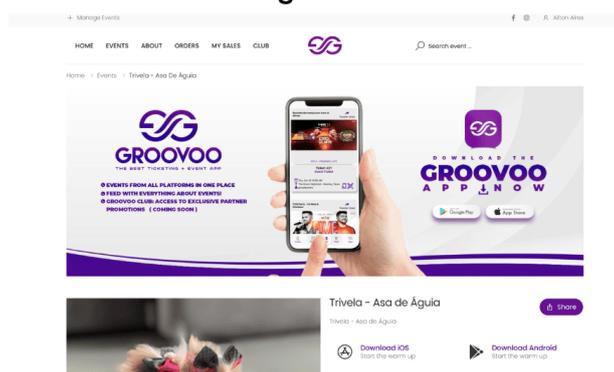


Fonte: Os autores

## Execução – Heurísticas

Cenário: compra de tickets na plataforma Groovoo

### 2. Estética e design minimalista



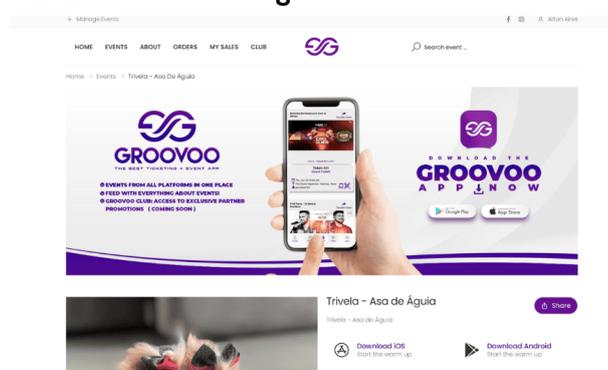
O sistema deve ser esteticamente agradável e simples, sem elementos desnecessários que possam distrair o usuário da tarefa em questão.

Fonte: Os autores

## Execução – Heurísticas

Cenário: compra de tickets na plataforma Groovoo

### 2. Estética e design minimalista



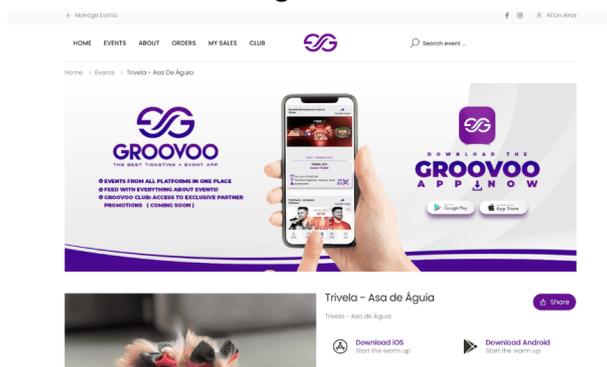
Ao clicar em um banner de evento, as informações que o usuário busca é em relação ao evento. A única informação referente ao evento que aparece após o clique é a do título do evento. O banner pode acabar desviando o foco do usuário

Fonte: Os autores

## Execução - Heurísticas

Cenário: compra de tickets na plataforma Groovoo

### 2. Estética e design minimalista



**Classificação: leve**

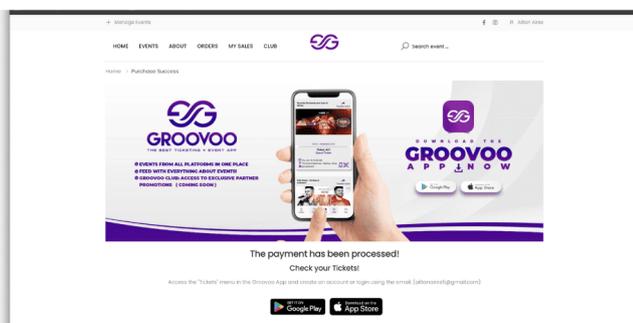
O usuário pode perder o foco da compra de ticket (o que gera receita) e acabar realizando as ações referidas ao que é apresentado no banner

Fonte: Os autores

## Execução - Heurísticas

Cenário: compra de tickets na plataforma Groovoo

### 2. Estética e design minimalista



**Sugestão**

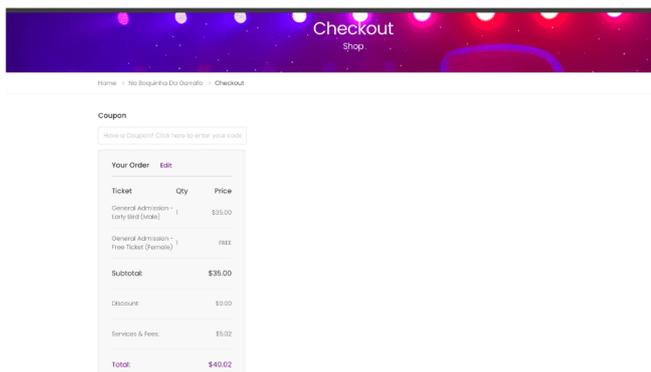
O banner de apresentação que aparece no checkout pode aparecer após a finalização da compra, já que os objetivos do usuário já foram atingidos.

Fonte: Os autores

## Execução - Heurísticas

**Cenário: compra de tickets na plataforma Groovoo**

### 3. Visibilidade do status do sistema



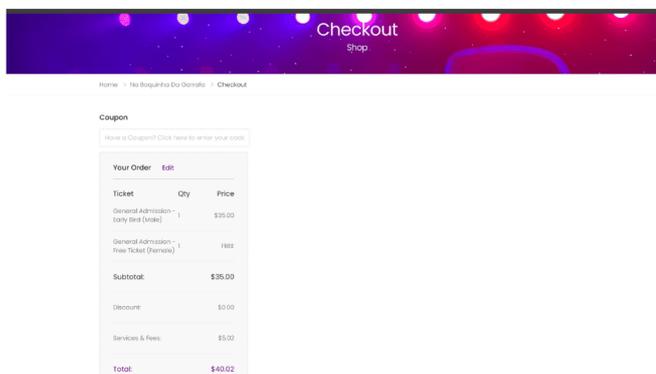
O sistema deve manter os usuários informados sobre o que está acontecendo, através de feedback apropriado e oportuno em relação às ações do usuário e ao estado do sistema.

Fonte: Os autores

## Execução - Heurísticas

**Cenário: compra de tickets na plataforma Groovoo**

### 3. Visibilidade do status do sistema



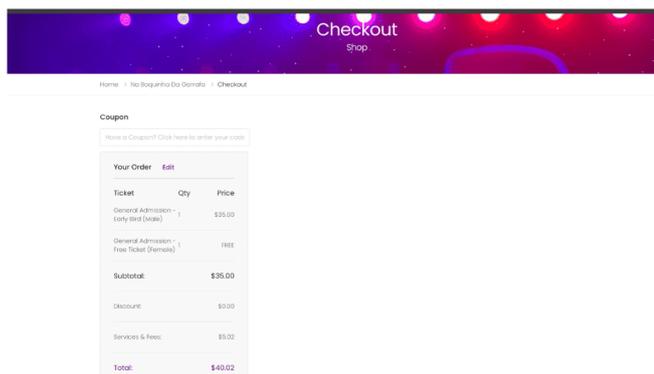
Ao clicar no botão place order, a parte da direita da tela fica em branco por alguns segundos até que o usuário tenha algum feedback do sistema.

Fonte: Os autores

## Execução - Heurísticas

**Cenário: compra de tickets na plataforma Groovoo**

### 3. Visibilidade do status do sistema



**Classificação: moderada**

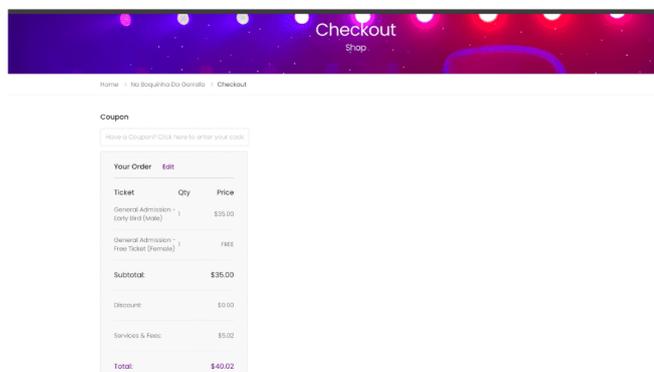
O usuário pode não finalizar a compra, o que pode gerar insatisfação no usuário e também impacta diretamente na receita do produto no caso de a compra não ser finalizada

Fonte: Os autores

## Execução - Heurísticas

**Cenário: compra de tickets na plataforma Groovoo**

### 3. Visibilidade do status do sistema



**Sugestão**

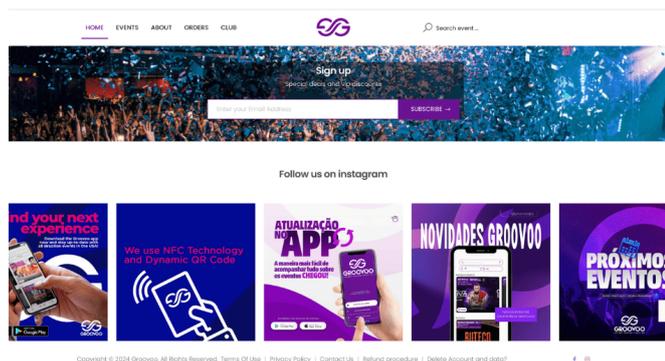
Adicionar um componente de loading enquanto os dados não são carregados

Fonte: Os autores

## Execução - Heurísticas

### Outros problemas

#### 4. Consistência e padronização:



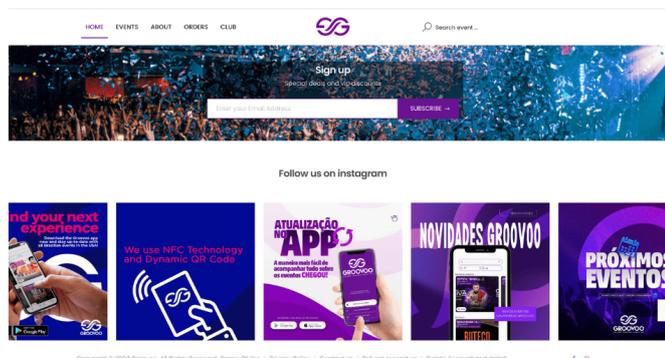
O sistema deve ser consistente em sua aparência e comportamento, e seguir padrões reconhecidos, para evitar confusão e aprendizado desnecessário.

Fonte: Os autores

## Execução - Heurísticas

### Outros problemas

#### 4. Consistência e padronização:



Nesse cenário o usuário está logado no sistema e mesmo assim um componente de cadastro aparece para ele

Fonte: Os autores

## Execução - Heurísticas

### Outros problemas

#### 4. Consistência e padronização:



#### Classificação: leve

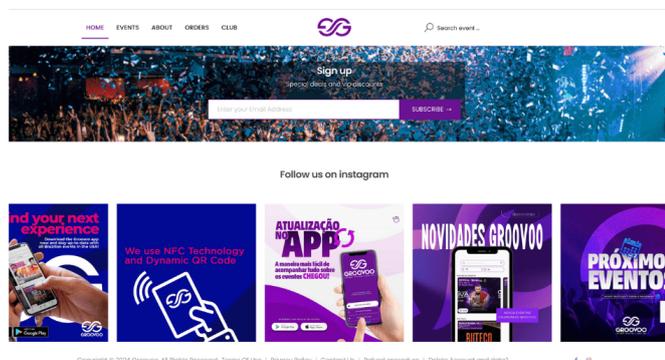
O usuário pode clicar no botão por engano e ter que passar por mais passos para voltar ao lugar que gostaria de acessar

Fonte: Os autores

## Execução - Heurísticas

### Outros problemas

#### 4. Consistência e padronização:



#### Sugestão

Filtrar os componentes que podem aparecer para usuários logados e para usuários sem sessão

Fonte: Os autores

# APÊNDICE E – Motivações para a Adição do Design Centrado no Usuário (UCD) ao Processo de Desenvolvimento Groovoo

# Motivações para a Adição do Design Centrado no Usuário (UCD) ao Processo de Desenvolvimento

## Introdução

O Design Centrado no Usuário (UCD) é uma abordagem que visa compreender as necessidades dos usuários e colocá-las no centro do design e desenvolvimento. Durante a reunião, foi discutida a importância do UCD. Este documento traz o motivo da implementação do UCD e suas contribuições.

## Compreensão das Necessidades dos Usuários

A principal razão para que uma equipe adote o UCD é a garantia de que o produto final atenda às necessidades dos usuários. Ao entender bem as necessidades dos usuários, a equipe de desenvolvimento pode criar soluções mais adequadas. Isso ajuda a:

- Identificar problemas e oportunidades que podem passar despercebidos sem a perspectiva UCD.
- Desenvolver funcionalidades que realmente agregam valor para os usuários.

## Melhoria da Intuitividade e Eficiência

Produtos com foco no usuário tendem a ser mais intuitivos durante a sua utilização. Usar o UCD pode resultar em interfaces mais amigáveis e fluxos de trabalho mais adequados, reduzindo a curva de aprendizado e aumentando a eficiência. Os principais benefícios são:

- Aumento da satisfação do usuário, já que terá uma melhor experiência de uso.
- Redução do tempo necessário para os usuários aprenderem a usar o produto.

## Redução de Erros e Retrabalhos

Integrar o UCD ao processo de desenvolvimento pode reduzir erros e retrabalhos. Ao considerar as necessidades dos usuários como base para o desenvolvimento, pode-se evitar problemas que normalmente só seriam descobertos após o produto estar disponível para os usuários. Isso resulta em:

- Menos bugs e problemas de usabilidade identificados durante os testes e após o lançamento.
- Menor necessidade de correções e ajustes pós-lançamento, o que faz com que economize tempo e recursos

## Mentalidade Empática

Adotar o UCD incentiva a equipe a desenvolver uma mentalidade empática, colocando-se no lugar dos usuários para tomar decisões mais adequadas e centradas neles. Isso ajuda a:

- Melhorar a comunicação interna e a colaboração, alinhando todos em relação ao objetivo de criar um produto que realmente atenda às necessidades dos usuários.

- Tomar decisões mais fundamentadas, sempre considerando o impacto no usuário final.

# APÊNDICE F – Dicionário de termos técnicos

# Dicionário de Termos da Equipe de Desenvolvimento

- API (Application Programming Interface): Conjunto de rotinas e padrões estabelecidos por uma aplicação para que outras aplicações possam utilizar suas funcionalidades.
- Deploy: Processo de colocar uma aplicação em produção, tornando-a acessível para os usuários finais.
- Endpoint: URL específica onde um serviço ou recurso da API pode ser acessado.
- Framework: Conjunto de ferramentas e bibliotecas que ajudam no desenvolvimento de software de forma mais rápida e organizada.
- Library: Coleção de funções e rotinas que podem ser reutilizadas em diferentes programas.

## GitHub

- Repository (Repositório): Local onde o código-fonte de um projeto é armazenado e gerenciado.
- Commit: Registro de alterações feitas no código.
- Branch (Ramo): Linha de desenvolvimento paralela, usada para trabalhar em funcionalidades separadas sem afetar o código principal.
- Merge: Processo de unir mudanças de diferentes branches em uma única branch.
- Pull Request (PR): Solicitação para revisar e mesclar alterações de uma branch para outra.
- Fork: Cópia de um repositório que permite fazer alterações sem afetar o repositório original.
- Issue: Ferramenta de rastreamento de tarefas, bugs e melhorias.
- GitHub Projects (Quadro Kanban): Ferramenta de gerenciamento de projetos que permite organizar e priorizar tarefas.
- GitHub Actions: Ferramenta de automação de fluxos de trabalho baseada em eventos.
- Assignment (Atribuição): Ato de designar uma tarefa ou responsabilidade a um membro da equipe.
- Code Review: Processo de revisão do código por outros membros da equipe para garantir qualidade e consistência.
- Master: Branch principal de um repositório, onde o código estável é mantido.
- Develop: Branch de desenvolvimento onde as novas funcionalidades são integradas antes de serem mescladas na branch master.
- Labels (Etiquetas): Marcadores usados para categorizar e organizar issues e pull requests.

# Docker

- Container: Instância isolada de uma aplicação ou serviço que compartilha o mesmo kernel do sistema operacional.
- Image: Snapshot de um sistema de arquivos e suas dependências, usado para criar containers.
- Dockerfile: Arquivo de configuração utilizado para definir os passos necessários para criar uma imagem Docker.
- Volume: Mecanismo para persistir dados gerados e utilizados por containers Docker.
- Compose: Ferramenta para definir e gerenciar multi-containers Docker em um ambiente de desenvolvimento.
- Infraestrutura (Infra)
- Cloud Computing: Uso de serviços de computação (armazenamento, servidores, banco de dados, rede) através da internet.
- Load Balancer (Balanceador de Carga): Distribui o tráfego de rede ou aplicação entre vários servidores.
- Serverless: Modelo de execução em que o provedor de cloud gerencia automaticamente a infraestrutura necessária para executar o código.
- Virtual Machine (VM): Emulação de um sistema computacional que fornece a funcionalidade de um computador físico.
- Kubernetes: Sistema de orquestração de containers que automatiza a implantação, escalabilidade e gerenciamento de aplicações em containers.

# Desenvolvimento Front-end

- HTML (HyperText Markup Language): Linguagem de marcação utilizada para estruturar o conteúdo da web.
- CSS (Cascading Style Sheets): Linguagem de estilo utilizada para descrever a apresentação de documentos escritos em HTML ou XML.
- JavaScript: Linguagem de programação utilizada para criar interatividade em páginas web.
- React: Biblioteca JavaScript para construção de interfaces de usuário.
- Angular: Framework JavaScript para construção de aplicações web robustas.

# Desenvolvimento Mobile

- Responsive Design (Design Responsivo): Técnica de design que ajusta o layout de acordo com o tamanho da tela do dispositivo.
- Native App (Aplicativo Nativo): Aplicativo desenvolvido especificamente para um sistema operacional móvel, como iOS ou Android.
- Hybrid App (Aplicativo Híbrido): Aplicativo desenvolvido usando tecnologias web que é executado dentro de um container nativo.

- Progressive Web App (PWA): Aplicações web que utilizam capacidades modernas da web para fornecer uma experiência similar a aplicativos nativos.
- SDK (Software Development Kit): Conjunto de ferramentas que facilita o desenvolvimento de aplicações para uma plataforma específica.

## Outras Áreas Relevantes

- Microservices (Microserviços): Arquitetura de software que estrutura uma aplicação como um conjunto de serviços independentes.
- CI/CD (Continuous Integration/Continuous Deployment): Práticas de desenvolvimento que buscam integrar e entregar código continuamente e de forma automatizada.
- ORM (Object-Relational Mapping): Técnica de conversão de dados entre sistemas incompatíveis usando linguagens de programação orientadas a objetos.
- REST (Representational State Transfer): Estilo arquitetural para construir APIs usando HTTP.
- GraphQL: Linguagem de consulta para APIs que permite solicitar exatamente os dados necessários.

## APÊNDICE G – Template de *issues*

The image shows a GitHub issue template form with a dark theme. It consists of the following elements:

- Add a title:** A text input field labeled "Title".
- Add a description:** A section with two tabs: "Write" (active) and "Preview".
- Rich text editor:** A toolbar with icons for bold (H), italic (I), strikethrough, code (<>), link, list, ordered list, table, image, mention (@), share, undo, and redo.
- Template content:**
  - # Titulo
  - ## 📄 Descrição
  - ## 🛠️ Como solucionar
  - ## ✅ Critérios de aceitação
  - [ ] Critério 1
  - [ ] Critério 2
  - [ ] Critério 3
- Footer:** Two informational boxes: "Markdown is supported" and "Paste, drop, or click to add files".
- Submit button:** A green button labeled "Submit new issue".

Fonte: Os autores

## APÊNDICE H – Colunas do quadro Kanban

- **No Status:** Quanto às demandas, elas serão cadastradas como *issue* no GitHub e, através da integração, todas serão automaticamente direcionadas para a coluna 'No Status' no quadro Kanban do GitHub Projects;
- **Backlog:** Coluna destinada as demandas a serem desenvolvidas;
- **Priority Queue:** Coluna destinada a demandas urgentes, estritamente relacionadas ao ambiente de produção;
- **To Do:** Coluna que apresenta o quadro de demandas que devem ser produzidas pelos desenvolvedores, em ordem de prioridade
- **In Progress:** Coluna que apresenta as demandas atualmente em desenvolvimento;
- **Review:** Coluna que apresenta as demandas já desenvolvidas e enviadas para o ambiente de teste;
- **Done:** Coluna que apresenta as demandas já testadas e que podem ser enviadas para o ambiente de produção.