

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

Diretrizes de Design Centrado no Usuário para um Processo ScrumBan: um estudo de caso

**Autores: Matheus de Cristo Doreia Estanislau e
Lucas Alexandre Fernandes Martins**

Orientadora: Prof^a. Dr^a. Rejane Maria da Costa Figueiredo

Brasília, DF
2022



Matheus de Cristo Doreia Estanislau e Lucas Alexandre Fernandes Martins

Diretrizes de Design Centrado no Usuário para um Processo ScrumBan: um estudo de caso

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^a. Dr^a. Rejane Maria da Costa Figueiredo

Brasília, DF

2022

Matheus de Cristo Doreia Estanislau e Lucas Alexandre Fernandes Martins

Diretrizes de Design Centrado no Usuário para um Processo ScrumBan: um estudo de caso

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

2022

Agradecimentos

Eu, Matheus de Cristo Doreia Estanislau, agradeço meus pais, Alessandra e Edson, não há palavras suficientes para expressar o quanto sou grato pelo amor incondicional, apoio e incentivo que me proporcionaram ao longo desta jornada. Vocês foram minha base sólida, sempre acreditando em mim, mesmo nos momentos em que eu duvidava de mim mesmo. Obrigado por serem a força motriz por trás das minhas conquistas e por me apoiarem incondicionalmente em todas as decisões que tomei.

Agradeço a Professora Dr^a. Rejane Figueiredo por sua orientação e paciência que foram essenciais para o desenvolvimento deste trabalho. Suas valiosas contribuições, sugestões e conhecimentos compartilhados moldaram cada detalhe deste TCC e tornaram possível aprimorar meu entendimento sobre o tema. Obrigado por ser uma fonte inspiradora de sabedoria e por acreditar no meu potencial ao longo de toda a jornada acadêmica.

À minha namorada Myllena Cardoso, meu amor e gratidão não têm limites. Sua presença em minha vida trouxe luz e inspiração. Obrigado por ouvir pacientemente minhas ideias e me encorajar constantemente nos meus sonhos. Seu apoio incondicional e incentivo foram essenciais para a conclusão bem-sucedida deste trabalho.

A todos aqueles que, de alguma forma, contribuíram para o desenvolvimento deste trabalho e me ajudaram ao longo dessa jornada, meu mais profundo agradecimento. Seu apoio fez toda a diferença em minha vida acadêmica e em meu crescimento pessoal.

Eu, Lucas Alexandre Fernandes Martins, agradeço meus pais, Maria Aparecida e Ricardo, por todo o apoio e confiança. Agradeço meu irmão, Ricardo Jr. por sempre me incentivar e ser um exemplo para mim. Agradeço meus amigos que conheci durante o curso, por toda ajuda que recebi. Agradeço aos professores do curso Engenharia da FGA UnB que disseminam seus conhecimentos e contribuem com excelência para a formação de seus discentes.

*"The only way to go fast, is to go well."
(Robert C. Martin)*

Resumo

O final da década de 1990 foi um período marcado por desafios significativos enfrentados pelas práticas tradicionais de desenvolvimento de software e como resposta a esses desafios, um grupo de profissionais se uniu e criou o Manifesto Ágil em 2001, estabelecendo princípios para abordagens mais flexíveis e adaptativas no campo do desenvolvimento de software. Dentre as metodologias ágeis que ganharam destaque nesse contexto, destaca-se o Scrum, desenvolvido por Ken Schwaber e Jeff Sutherland. O Scrum se baseia em ciclos de trabalho denominados *sprints*, nos quais as equipes realizam entregas incrementais do produto em curtos períodos de tempo, promovendo a transparência, inspeção e adaptação contínuas. Outra abordagem ágil relevante é o Kanban, que se concentra na visualização do fluxo de trabalho e na limitação do trabalho em andamento. O Kanban proporciona às equipes de trabalho a capacidade de rastrear e controlar o fluxo de trabalho de forma visual e eficiente. À medida que o tempo avançou, surgiram abordagens híbridas, como o Scrumban, que combinam elementos do Scrum e do Kanban. Tais abordagens oferecem flexibilidade e personalização de acordo com as necessidades específicas da equipe e do projeto em questão. Paralelamente, o User-Centered Design (UCD) emergiu como uma prática reconhecida e relevante no desenvolvimento de software, atribuindo importância primordial à participação ativa dos usuários em todas as etapas do processo de design. O UCD enfatiza a compreensão aprofundada das necessidades, preferências e experiências dos usuários finais, a fim de criar produtos de software que atendam de forma eficiente e eficaz às suas demandas. Diante desse contexto histórico, o objetivo central deste trabalho consiste em propor diretrizes do *Design Centrado no Usuário*, a partir do *framework Scrum-UCD*, para o processo baseado em Scrumban de desenvolvimento de software da *startup Vamos Parcelar*, na qual os autores deste trabalho atuam. A proposta é criar produtos de software que atendam de maneira mais eficiente às necessidades dos usuários finais, combinando os princípios ágeis com a abordagem centrada no usuário. A união entre o UCD e o Scrum, que valoriza entregas iterativas e adaptação contínua, visa alcançar um desenvolvimento de software mais eficiente e orientado às demandas reais dos usuários. Como resultados, espera-se que a integração proposta contribua para aprimorar a qualidade, usabilidade e experiência dos produtos de software desenvolvidos.

Palavras-chave: User-Centered Design, Scrum-UCD, desenvolvimento de software, abordagem centrada no usuário.

Abstract

The late 1990s was a period marked by significant challenges faced by traditional software development practices. In response to these challenges, a group of professionals came together and created the Agile Manifesto in 2001, establishing principles for more flexible and adaptive approaches in software development. Among the prominent agile methodologies that emerged during this period is Scrum, developed by Ken Schwaber and Jeff Sutherland. Scrum is based on work cycles called sprints, in which teams deliver incremental product releases in short periods, promoting continuous transparency, inspection, and adaptation. Another relevant agile approach is Kanban, which focuses on visualizing the workflow and limiting work in progress. Kanban provides teams with the ability to visually and efficiently track and control the workflow.

Over time, hybrid approaches such as Scrumban emerged, combining elements of Scrum and Kanban, offering flexibility and customization according to the specific needs of the team and the project at hand. Simultaneously, User-Centered Design (UCD) emerged as a recognized and relevant practice in software development, giving paramount importance to active user participation in all stages of the design process. UCD emphasizes a deep understanding of the needs, preferences, and experiences of end-users to create software products that efficiently and effectively meet their demands.

Given this historical context, the central objective of this work is to propose User-Centered Design guidelines, based on the Scrum-UCD framework, for the Scrumban-based software development process at the startup Vamos Parcelar, where the authors of this work are involved. The proposal aims to create software products that more effectively meet the needs of end-users by combining agile principles with a user-centered approach. The integration of UCD and Scrum, which values iterative deliveries and continuous adaptation, aims to achieve more efficient software development aligned with the real demands of users. It is expected that the proposed integration will contribute to improving the quality, usability, and user experience of the software products developed.

Keywords: User-Centered Design, Scrum-UCD, software development, user-centered approach.

Lista de ilustrações

Figura 1 – Classificação Metodológica	15
Figura 2 – Etapas UCD	24
Figura 3 – Fase de iniciação	31
Figura 4 – Fase de iniciação - Continuação	32
Figura 5 – Fase de planejamento	33
Figura 6 – Fase de planejamento - Continuação	33
Figura 7 – Cronograma de pesquisa	39
Figura 8 – Processo de desenvolvimento da VP	42
Figura 9 – Valores padrão da VP	44
Figura 10 – Fluxograma Processo VP	47
Figura 11 – Fluxograma Processo VP, parte 2	48
Figura 12 – Fluxograma Processo VP, parte 3	48
Figura 13 – Fluxograma Diretrizes UCD	52
Figura 14 – Gráfico de satisfação	56

Lista de tabelas

Tabela 1 – Comparação entre Scrum, Kanban e Scrumban	22
Tabela 2 – Análise comparativa entre as metodologias UCD	28
Tabela 3 – Análise comparativa entre as técnicas UCD 1	29
Tabela 4 – Análise comparativa entre as técnicas UCD 2	29
Tabela 5 – Análise comparativa entre as técnicas UCD 3	29
Tabela 6 – Estrutura do protocolo de estudo de caso	36

Lista de abreviaturas e siglas

UnB	Universidade de Brasília
UCD	Design centrado no usuário
VP	Vamos Parcelar
UX	Experiência do usuário
PO	<i>Product Owner</i>
QA	<i>Quality Assurance</i>

Sumário

1	INTRODUÇÃO	13
1.1	Contexto	13
1.2	Problema	14
1.3	Objetivos	14
1.4	Metodologia	15
1.4.1	Plano Metodológico	15
1.5	Organização do Trabalho	16
2	REFERENCIAL TEÓRICO	18
2.1	Considerações Iniciais	18
2.2	Frameworks Ágeis	18
2.2.1	Scrum	18
2.2.2	Kanban	20
2.2.3	Scrumban	20
2.2.4	Análise dos frameworks	21
2.3	<i>User-Centered Design</i> - UCD	22
2.3.1	UCD e Metodologias Ágeis	24
2.3.2	Metodologias UCD	24
2.3.3	Técnicas UCD	25
2.4	Framework SCRUM-UCD	27
2.4.1	Análise comparativa entre as metodologias UCD	28
2.4.1.1	Metodologias selecionadas para o Scrum-UCD	28
2.4.2	Análise comparativa entre as técnicas UCD	28
2.4.2.1	Técnicas selecionadas para o Scrum-UCD	30
2.4.3	Fases do Framework Scrum -UCD	30
2.4.3.1	Fase de iniciação	31
2.4.3.2	Fase de Planejamento	32
2.4.3.3	Fase de Implementação	33
2.5	Considerações Finais do Capítulo	34
3	MATERIAIS E MÉTODOS	35
3.1	Considerações Iniciais	35
3.2	Planejamento de Pesquisa	35
3.3	Etapas do Protocolo de Estudo de Caso	36
3.3.1	<i>Background</i>	36
3.3.2	<i>Design</i>	37

3.3.3	Seleção de caso	37
3.3.4	Procedimentos e papéis do estudo de caso	37
3.3.5	Coleta de dados	37
3.3.6	Análise dos Dados	38
3.3.7	Validade do plano	38
3.3.8	Limitações do Estudo	38
3.3.9	Relatório	38
3.3.10	Apêndice	38
3.3.11	Cronograma	38
3.4	Considerações Finais do Capítulo	39
4	STARTUP VAMOS PARCELAR	40
4.1	Considerações Iniciais	40
4.2	Startup Vamos Parcelar (VP)	40
4.3	<i>Scrumban</i> VP	40
4.3.1	Time de Produto	40
4.3.2	Scrum Master	41
4.3.3	Time de desenvolvimento	41
4.3.4	Analista de Qualidade	41
4.3.5	Time de infraestutura	41
4.3.6	Time de <i>Design</i>	42
4.3.7	Processos da VP	42
4.3.8	Planejamento de desenvolvimento	43
4.3.9	Técnicas de estimativa	43
4.3.10	Quadro <i>Kanban</i>	44
4.3.11	<i>Daily</i>	45
4.3.12	Revisão	45
4.3.13	<i>Deploy</i>	45
4.4	Considerações Finais do Capítulo	46
5	DIRETRIZES UCD PARA O PROCESSO STARTUP VP	47
5.1	Considerações Iniciais	47
5.2	Diagrama do Processo VP	47
5.3	Análise do processo da <i>Startup</i> VP quanto a literatura do <i>Design</i> <i>Centrado no Usuário</i>	49
5.4	Integração da <i>Diretrizes UCD</i> criadas para o Processo da <i>startup</i> VP	51
5.4.1	Aplicação das diretrizes UCD selecionadas para o Processo da <i>startup</i> VP .	52
5.4.1.1	Participação dos Usuários	52
5.4.1.2	Compreensão do contexto de uso	53
5.4.1.3	Iteração e <i>feedback</i> contínuos	54

5.4.1.4	Prototipagem	54
5.4.1.5	Avaliação e testes com usuários	55
5.5	Integração do Processo VP com as Diretrizes UCD e o <i>Framework Scrum-UCD</i>	56
5.6	Considerações Finais do Capítulo	58
6	ANÁLISE DOS DADOS	60
6.1	Considerações Iniciais do Capítulo	60
6.1.1	Análise das limitações do processo da Startup VP	60
6.1.2	Identificação de oportunidades de Melhorias	61
7	CONCLUSÃO	63
7.0.1	Proposta para trabalhos futuros	64
	REFERÊNCIAS	65
	APÊNDICE A – CARTA DE SOLICITAÇÃO DE ACESSO E CI- TAÇÃO	68

1 Introdução

1.1 Contexto

Nos últimos anos, as metodologias ágeis têm ganhado destaque na gestão de projetos de TI. Dentre essas metodologias, destacam-se o Scrum, Kanban, Scrumban. Embora cada uma dessas metodologias tenha suas particularidades, todas têm em comum a busca pela melhoria contínua e a ênfase na colaboração entre os membros da equipe.

O Scrum e o Kanban são duas metodologias amplamente utilizadas na gestão de projetos ágeis. Enquanto o Scrum é mais indicado para projetos com um alto grau de complexidade, o Kanban é mais flexível e adaptável a diferentes tipos de projetos (KNI-BERG; SKARIN, 2010). O Scrum divide o projeto em sprints e enfatiza a comunicação e a colaboração entre os membros da equipe (SCHWABER; SUTHERLAND, 2013). Já o Kanban é uma metodologia ágil que tem como foco a otimização do fluxo de trabalho, a eliminação de gargalos e a redução de desperdícios. Nessa metodologia, as tarefas são representadas por cartões que são movidos em um quadro visual, de acordo com o seu status de execução (ANDERSON, 2010).

Algumas equipes enfrentam dificuldades ao tentar escolher um desses dois métodos. Foi então que surgiu o Scrumban, que combina elementos do Scrum e do Kanban. O Scrumban é indicado para equipes que desejam combinar a flexibilidade do Kanban com a estrutura do Scrum. O Scrumban utiliza o quadro Kanban para gerenciar o fluxo de trabalho e adiciona as cerimônias do Scrum, como as reuniões diárias e as retrospectivas. Em outras palavras, o Scrumban é uma abordagem ágil que utiliza os conceitos do Scrum, como as reuniões diárias, o backlog de produtos e as sprints, mas adiciona elementos do Kanban, como o fluxo de trabalho contínuo e a visibilidade das tarefas (LADAS, 2009).

Ainda no contexto de melhorias, um dos temas que tem sido alvo de estudo e de aplicação é o Design Centrado no Usuário (em inglês - *User-Centered Design*, sigla UCD). Essa abordagem valoriza a criação de produtos e serviços que atendam às necessidades dos usuários. O UCD enfatiza a pesquisa e a prototipagem para entender as necessidades dos usuários e testar soluções de forma iterativa. O UCD é particularmente útil em projetos de design de interfaces e de experiência do usuário (UX).

Nesse contexto, observa-se o surgimento do Scrum-UCD, um *framework* que combina os princípios do Scrum e do UCD. Essa abordagem busca integrar a pesquisa e o design centrado no usuário com o processo de desenvolvimento ágil do Scrum. O Scrum-UCD enfatiza a colaboração entre os membros da equipe de desenvolvimento e os designers, com o objetivo de criar produtos que atendam às necessidades dos usuários e que sejam en-

tregues dentro do prazo e do orçamento estabelecidos (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

As metodologias ágeis, como o Scrum, Kanban, Scrumban, UCD e Scrum-UCD, têm em comum a busca pela melhoria contínua, a ênfase na colaboração entre os membros da equipe e a adaptabilidade a diferentes tipos de projetos. Cada uma dessas metodologias tem suas particularidades e pode ser mais indicada para determinados tipos de projetos, dependendo de suas características e objetivos específicos.

O objeto de estudo deste trabalho, a *Startup Vamos Parcelar* (VP), é uma *startup* de desenvolvimento de software, cujos processos de desenvolvimento de software são baseados no *Scrum*, *Kanban* e *Scrumban*.

1.2 Problema

Na *Startup* VP a utilização da metodologia *Scrumban* auxilia e aprimora processos durante o desenvolvimento de software. Contudo, diretrizes do *Design Centrado no Usuário* não estão inseridas nesse processo.

Apesar do *Scrumban* oferecer melhor gerência, comunicação e colaboração nos projetos, sua utilização não garante que o produto correto esteja sendo construído. Uma das demandas é o emprego de processos e atividades iterativas que possam identificar se o produto que está sendo construído atende as necessidades dos usuários ou até mesmo, garanta maior nível de usabilidade e melhora da experiência do usuário.

Contudo, romper de forma abrupta com as metodologias já utilizadas pelo mercado pode ser algo custoso, visto que os ritos utilizados e os artefatos produzidos pela equipe de software muitas vezes já estão inseridos na cultura de trabalho da empresa.

Nesse contexto, surge a pergunta de pesquisa:

- *Como relacionar as práticas do framework Scrum-UCD com as do Scrumban empregado pela VP para que essas possam ser somadas ao processo da empresa, buscando o aperfeiçoamento da construção de software que atenda as necessidades do usuário?*

1.3 Objetivos

O objetivo deste trabalho é propor diretrizes de Design Centrado no Usuário, baseado no *framework Scrum-UCD* para o Processo *Scrumban* de desenvolvimento da *Startup Vamos Parcelar* (VP).

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

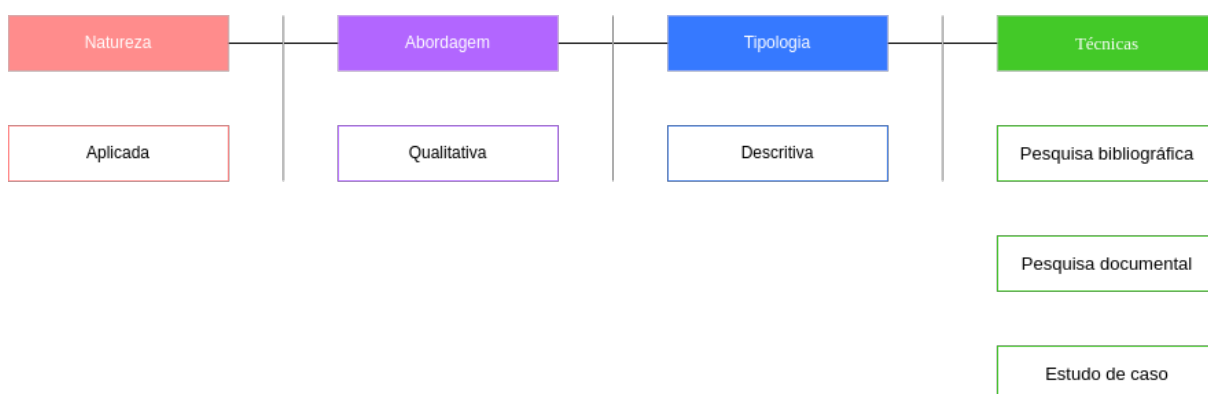
- Analisar o Processo da *Startup VP* quanto aos *frameworks e conceitos de Design Centrado no Usuário* da literatura para a integração de diretrizes de UCD;
- Propor a integração da *Diretrizes UCD selecionadas* para o Processo da *Startup VP* quanto ao Design Centrado no Usuário.

1.4 Metodologia

A metodologia deste trabalho foi definida e classificada de acordo com o seu objetivo e um plano metodológico foi estruturado.

Dado que o objetivo proposto de integração entre o *framework Scrum-UCD* com o *Scrumban* para a *Startup Vamos Parcelar (VP)*, esta pesquisa é de natureza aplicada e sua abordagem é qualitativa. A Tipologia é classificada como descritiva, pois visa descrever métodos e características de um objeto. São empregados os procedimentos de revisão bibliográfica e documental e a técnica Estudo de Caso.

Figura 1 – Classificação Metodológica



1.4.1 Plano Metodológico

Conforme Brereton et al (BRERETON et al., 2008), o processo de um estudo de caso possui quatro fases principais: Planejamento; Coleta de Dados; Análise de Dados; e Relatórios, que compõem o plano metodológico adotado neste trabalho:

- **Planejamento da pesquisa:** Nesta fase foram definidos a questão de pesquisa, os objetivos geral e específicos, e a classificação e o plano da metodologia de pesquisa;
- **Coleta de dados:** Nessa fase são definidos e empregados os procedimentos de revisão bibliográfica e documental, e a técnica estudo de caso;
- **Análise dos Resultados:** Nessa fase, os dados coletados e resultados alcançados são analisados qualitativamente;

- **Relatório:** E finalizado, nessa fase, o relatório constitui a escrita deste documento de *Trabalho de Conclusão de Curso*.

1.5 Organização do Trabalho

Este trabalho de conclusão de curso está organizado nos seguintes capítulos:

- **Capítulo 1 - Introdução:** neste capítulo foram apresentados o contexto do trabalho, o problema de pesquisa, os objetivos deste trabalho, e uma síntese da metodologia planejada;
- **Capítulo 2 - Referencial teórico:** descreve os conceitos que fundamentam este trabalho. O capítulo é subdividido nas seções *Scrum*, *Kanban*, *Scrumban*, *Design Centrado no Usuário (User-Centered Design - UCD)*, e finalizando, o *framework Scrum-UCD*;
- **Capítulo 3 - Materiais e Métodos** apresenta o plano metodológico adotado de forma mais detalhada e caracteriza o objeto de estudo.
- **Capítulo 4 - Startup *Vamos Parcelar*:** apresenta uma breve descrição do objeto de estudo deste trabalho, Startup VP e seu processo de desenvolvimento de software baseados nas metodologias ágeis.
- **Capítulo 5 - Diretrizes UCD para o Processo Startup VP** apresentam-se os procedimentos da avaliação do processo de desenvolvimento de software da *startup* Vamos Parcelar (VP) e a proposição de diretrizes UCD baseadas na revisão bibliográfica. Em seguida, apresenta-se uma proposta de integração da *Diretrizes UCD criadas* para o Processo da *Startup VP*. Finalizando, apresenta-se a análise dos dados resultantes desta pesquisa.
- **Capítulo 6 - Análise de Dados** apresentam-se a análise de dados na integração do processo *Scrumban* da startup Vamos Parcelar com métodos UCD. Analisaremos as limitações do processo da empresa, como a falta de ênfase na experiência do usuário e a ausência de feedback e validação contínua. Em seguida, exploraremos as oportunidades de melhoria ao incorporar o UCD, como compreensão aprofundada das demandas dos usuários, maior colaboração, redução de retrabalho, melhoria da experiência do usuário e aumento da eficiência. A integração do UCD tem o potencial de impulsionar a competitividade e a satisfação do usuário no processo de desenvolvimento da Vamos Parcelar.
- **Capítulo 7 - Conclusões e Trabalhos Futuros** Neste capítulo, são apresentadas as conclusões deste trabalho, abordando os resultados obtidos em relação aos

objetivos propostos. São discutidas as contribuições do estudo para a área de desenvolvimento de software e para a *startup* Vamos Parcelar. Além disso, são apontadas possíveis limitações e sugestões para trabalhos futuros, indicando caminhos para expandir e aprofundar a pesquisa realizada.

Por fim, encontram-se as **Referências Bibliográficas** deste trabalho.

Como **Apêndices**, apresenta-se informações adicionais que complementam o conteúdo do trabalho e fornecem detalhes relevantes para o leitor. Embora essas informações não sejam essenciais para a compreensão do texto principal, elas podem enriquecer a compreensão do assunto abordado.

No apêndice A, é apresentado um modelo de cartão de solicitação de acesso utilizado pela startup Vamos Parcelar (VP). Esse documento ilustra o processo formalizado de solicitação de acesso aos sistemas e recursos da empresa, proporcionando uma visão prática das práticas adotadas pela VP.

2 Referencial Teórico

2.1 Considerações Iniciais

Neste capítulo apresenta-se o referencial teórico deste trabalho. São apresentadas uma pequena introdução dos frameworks *Scrum* e *Kanban*, base do framework adotado pela empresa e objeto de estudo deste trabalho. Em seguida, um detalhamento do *framework Scrumban*, que é o *framework* adotado pela empresa. Em seguida, uma seção introdutória ao tema Design Centrado no Usuário (*User-Centered Design - UCD*), e um detalhamento do *Scrum-UCD, framework* objeto de estudo deste trabalho para a proposição de introdução de UX no processo da empresa, objeto deste estudo.

2.2 Frameworks Ágeis

Nesta seção apresenta-se um resumo dos frameworks Scrum, Kanban e Scrumban, que representam a base deste trabalho. Em seguida, apresenta-se um detalhamento dos frameworks.

2.2.1 Scrum

O *Scrum* é um *framework* lançado por Jeff Sutherland, em 1993, com o objetivo de melhorar a gerência de um desenvolvimento de software. O nome Scrum surgiu da comparação entre desenvolvedores e jogadores de *Rugby*. *Scrum* é a denominação da rápida reunião que ocorre quando os jogadores de *Rugby* vão iniciar um lance (CARVALHO; MELLO, 2012).

Scrum auxilia pessoas, equipes, organizações e empresas gerarem valor por meio de soluções adaptáveis e flexíveis para problemas complexos. O *Scrum* deixa claro a eficácia relativa das práticas de gerenciamento e desenvolvimento de produtos, de modo que você possa melhorá-las. O *Scrum* é dividido em três componentes, cada uma dessas componentes serve a uma finalidade específica e é essencial para aplicar de forma adequada o *Scrum*. São elas: papéis, eventos e artefatos (SCHWABER; SUTHERLAND, 2013).

A equipe do *Scrum* é composta por três papéis, são eles:

- **Scrum Master:** Destaca-se como um papel de fundamental importância dentro do Scrum por, dentre outras atribuições, assegurar o entendimento e a correta aplicação deste processo (BERNARDO; LIMA; CUNHA, 2016).

- **Product Owner:** É um dos papéis mais importantes no Scrum e muitas vezes o mais difícil. Ele é responsável por maximizar o valor do produto durante o seu ciclo de vida e apresenta os requisitos e objetivos do projeto (SVERRISDOTTIR; INGASON; JONASSON, 2014).
- **Time de desenvolvimento:** Consiste em integrantes da equipe que desenvolvem uma parte do produto que pode ser utilizado ao final de cada sprint. Somente integrantes do Time de Desenvolvimento criam incrementos.

Eventos prescritos são usados no *Scrum* para criar uma rotina e minimizar a necessidade de reuniões não definidas no *Scrum*, são eles (SCHWABER; SUTHERLAND, 2013):

- **Sprint:** É um time-boxed a um mês ou menos, no qual uma versão potencialmente utilizável de um produto é desenvolvida. Uma nova *Sprint* inicia imediatamente após a conclusão da *Sprint* anterior
- **Sprint Planning:** O *Sprint Planning* inicia o *Sprint*, estabelecendo o trabalho a ser executado para o *Sprint*. Este plano resultante é criado pelo trabalho colaborativo de todo o Time Scrum.
- **Sprint Review:** O objetivo do *Sprint Review* é inspecionar o resultado do *Sprint* e determinar futuras adaptações. O Time *Scrum* apresenta os resultados de seu trabalho para as principais partes interessadas e o progresso em direção ao Objetivo do Produto é discutido.
- **Sprint Retrospective:** O objetivo da *Sprint Retrospective* é criar uma oportunidade para o Time Scrum inspecionar a si próprio e elaborar um plano para melhorias a serem aplicadas na próxima *Sprint*.

Os artefatos do *Scrum* representam o trabalho ou o valor para o fornecimento de transparência e oportunidades para inspeção e adaptação. O principais artefatos presentes no *Scrum* são (SCHWABER; SUTHERLAND, 2013):

- **Backlog do Produto:** O *Backlog* do Produto é uma lista de histórias de usuário priorizadas
- **Backlog da Sprint:** São tarefas do *Backlog* do Produto a serem realizadas durante a *sprint*.
- **Incremento:** É a soma de todos os itens do *Backlog* do Produto completados durante a *Sprint* e o valor dos incrementos de todas as *Sprints* anteriores.

2.2.2 Kanban

O framework *Kanban* foi desenvolvido nos 50 pela Toyota para auxiliar o controle do seu sistema de produção industrial (OHNO, 1997). *Kanban* não é um sistema de controle de estoque, mas pode ser considerado como um sistema de visualização do trabalho, fluxo, reduzindo o desperdício e maximizando o valor do cliente. Na prática, a configuração de um sistema *Kanban*, incluem os seguintes passos (CORONA; EROS; DIEE, 2013):

- 1 - Realizar o mapeamento do fluxo, encontrando as atividades.
- 2 - Expresse os requisitos através de um conjunto de recursos.
- 3 - Dependendo das atividades e da composição da equipe, estabelecer um limite máximo para os recursos sob trabalho em cada atividade.
- 4 - Configuração do quadro *Kanban*
- 5 - Elaborar a política de atribuição de atividades e tarefas aos desenvolvedores, e para lidar com questões relacionadas ao fluxo.
- 6 - Decida o formato e o agendamento típico das reuniões
- 7 - Elaborar como os lançamentos de recursos únicos e de versões funcionais do sistema são entregues
- 8 - Elaborar as práticas técnicas específicas a serem usadas (design, programação, teste, etc).
- 9 - Decida quais ferramentas, métodos estatísticos e diagramas usar para gerenciar o processo.

Segundo (AHMAD et al., 2016), o *Kanban* é mais indicado para onde exista um alto grau de variabilidade da prioridade, como na manutenção em ciclos curtos. Nesse sentido, *Kanban* contribuiu para benefícios como visibilidade e priorização das tarefas, melhoria na moral da equipe, melhoria na comunicação, proteção das equipes quanto às exigências excessivas e compartilhamento de conhecimento entre os envolvidos.

2.2.3 Scrumban

O Método *Scrumban* é um híbrido entre *Scrum* e *Kanban* e que visa assimilar os pontos positivos de ambas as metodologias para atender as necessidades de um determinado projeto, visto que o *Kanban* e *Scrum* sozinhos não conseguem atender de forma satisfatória todos os projetos, dependendo do time de desenvolvedores, escopo.

Não há práticas específicas para *Scrumban*, porém a equipe precisa entender quais métodos do *Kanban* e do *Scrum* podem mais beneficiar em relação às suas necessidades e seus projetos. Para hibridizar ambos os métodos com sucesso, não há uma única forma

para selecionar o Scrumban, e a formação dele será baseada no problema em questão (ALQUDAH; RAZALI, 2018).

Scrumban é mais adaptável e recomendável especialmente quando há mudanças frequentes nos requisitos do usuário. Foi constatado que quanto maior a familiaridade da equipe ágil com os frameworks Kanban e *Scrum*, mais eles podem se beneficiar do *Scrumban* (JOSHI; SANDEEP, 2014). Os autores (ALQUDAH; RAZALI, 2018) verificaram que o *Scrumban* é mais adaptável especialmente em casos que ocorrem constantes alterações de requisitos.

Segundo (ALQUDAH; RAZALI, 2018), as escolhas para seleção do *Scrumban* são efetuadas de acordo com cada caso específico. Por exemplo, os membros da equipe Agile podem iniciar um projeto adotando o *Scrum* evitando sprints, planejamento, revisão e *backlog* da *sprint* quando a tarefa é pequena/grande para caber em uma *sprint* e quando é difícil estimar o tamanho ou duração das *sprints*. Para esse caso, a equipe manteria as demais práticas do *Scrum* e como práticas do *Kanban* limitaria o WIP (*Work In Progress*) e adotariam a visualização de fluxo de trabalho. Da mesma forma, podem existir muitos cenários e aplicações diferentes, ficando a cargo da equipe ajustar o método Scrumban para cada situação. Para isso, deve-se entender as práticas de ambos os métodos para poder selecionar quais usar.

2.2.4 Análise dos frameworks

Os autores (ALQUDAH; RAZALI, 2018) apresentam uma tabela comparativa entre *Scrum*, *Kanban* e *Scrumban*, conforme apresentado na Tabela 1.

Tabela 1 – Comparação entre Scrum, Kanban e Scrumban

Critério	Scrum	Kanban	Scrumban
Funções e Responsabilidades	Estrutura Predefinida	Não predefinido	Com base na decisão da equipe
Tempo de adoção	A transição é um pouco desafiadora. No entanto, as organizações ágeis parecem adotar o Scrum antes do Kanban	A transição para o Kanban é mais fácil, especialmente quando as equipes migram de métodos estruturados como cascata	A experiência da equipe em Scrum e Kanban é necessária para poder combiná-los
Tamanho da equipe	Uma equipe contém de 5 a 11 membros com funções predefinidas. Se a equipe for grande, o dimensionamento é necessário. Assim, para grandes equipes, Kanban pode ser melhor que Scrum.	Mais flexível que o Scrum ao considerar o tamanho da equipe. Os membros da equipe podem ter menos que 5 integrantes ou mais chegar até 14.	Mais flexível quando comparado ao Scrum e de alguma forma semelhante ao Kanban.
Priorização de requisitos	A priorização de requisitos é baseada na duração do sprint.	A priorização de requisitos é feita continuamente, podendo ser diária ou horária.	Baseado na decisão da equipe.
Custo	Ao contrário do Kanban, o Scrum evita a economia de custos, mas foca mais no conhecimento, na experiência e na tomada de decisões com base no que é conhecido.	Concentra-se na redução de custos, especialmente para operações.	Foca mais no corte de custos quando comparado ao Scrum e Kanban.
Qualidade	A reunião de revisão do Sprint é a principal prática para melhorar a qualidade no método Scrum.	O Kanban se concentra mais em melhorar a qualidade quando comparado ao Scrum.	A qualidade será aumentada ao usar Scrumban

Fonte: Adaptado de (ALQUDAH; RAZALI, 2018)

2.3 User-Centered Design - UCD

O termo *Design* Centrado no Usuário (em inglês, *User-Centered Design* - UCD) foi proposto por Norman e Draper em 1986 (NORMAN; DRAPER, 1986). De lá para cá, vem crescendo em várias áreas, inclusive na área de Engenharia de Software.

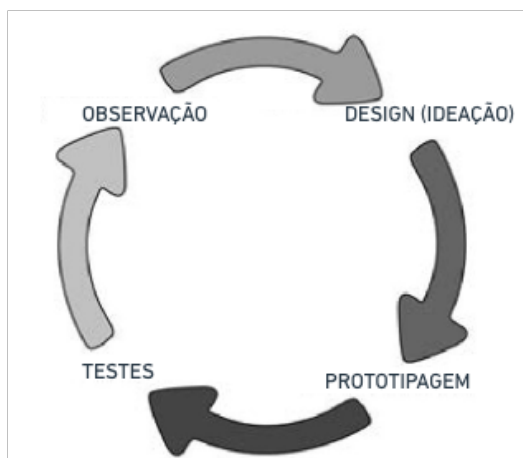
O UCD visa posicionar o usuário no centro das decisões do *design*, a partir de processos iterativos, compostos por 4 etapas (NORMAN, 2013):

- **Observação:** A pesquisa inicial para entender a natureza do problema, faz parte da disciplina de *design research*. Onde o pesquisador irá até os potenciais clientes, observando suas atividades, tentando entender seus interesses, motivações e verdadeiras necessidades.
- **Design (Ideação):** Uma vez que os requisitos de projeto são determinados, através da observação, o próximo passo para a equipe de projeto é gerar potenciais soluções. Existem muitas maneiras de gerar ideias: muitos desses métodos se enquadram no

título de “brainstorming”. Seja qual for o método usado, o autor sugere 2 regras, seguidas de uma terceira complementar:

1. **Gerar inúmeras ideias.** É perigoso ficar obcecado por uma ou duas ideias muito cedo no processo.
 2. **Ser criativo sem levar em conta as restrições.** Evitar a crítica de ideias. Mesmo ideias malucas, muitas vezes obviamente erradas, podem conter percepções criativas que podem ser posteriormente extraídas e bem utilizadas na seleção final da ideia. Evite a rejeição prematura de ideias.
 3. **Questionar tudo.** Uma pergunta estúpida pergunta sobre coisas tão fundamentais que todos assumem que a resposta é óbvia. Mas quando a questão é levada a sério, muitas vezes ela se revela profunda: o óbvio muitas vezes não é nada óbvio. O que supomos ser óbvio é simplesmente a forma como as coisas sempre foram feitas, mas agora que isso é questionado, não sabemos realmente os motivos. Muitas vezes a solução dos problemas é descoberta através de perguntas estúpidas, questionando o óbvio.
- Prototipagem: A única maneira de realmente saber se uma ideia é razoável é testá-la. Construa um protótipo rápido de cada solução potencial. A prototipagem durante a fase de especificação do problema é feita principalmente para garantir que o problema seja bem compreendido. Se o público-alvo já estiver usando algo relacionado ao novo produto, isso pode ser considerado um protótipo. Durante a fase de projeto de solução do problema, protótipos reais da solução proposta são invocados.
 - Testes: Reúna um pequeno grupo de pessoas que correspondam o mais próximo possível aos potenciais clientes, aquelas a quem o produto se destina. Faça com que usem os protótipos o mais próximo possível da maneira como realmente os usariam. Se o dispositivo for normalmente usado por uma pessoa, teste uma pessoa de cada vez. Se for normalmente usado por um grupo, teste um grupo. A única exceção é que mesmo que o uso normal seja por uma única pessoa, é útil pedir a um par de pessoas para usá-lo em conjunto, uma pessoa operando o protótipo, a outra orientando as ações e interpretando os resultados (em voz alta). Usar pares dessa maneira faz com que eles discutam suas ideias, hipóteses e frustrações de forma aberta e natural.

Figura 2 – Etapas UCD



Fonte: (NORMAN, 2013)

2.3.1 UCD e Metodologias Ágeis

A adoção da metodologia ágil para o desenvolvimento de software é crescente. No entanto, em seus ciclos de vida de desenvolvimento, esses métodos ainda carecem da percepção da importância da usabilidade e das interfaces de usuário. Holzinger chama a atenção para a necessidade de conhecimento de várias técnicas de usabilidade por profissionais de software que devem ser aplicadas de acordo com a natureza de um projeto.

Tanto os métodos ágeis quanto as metodologias de *design* centrado no usuário têm muitas semelhanças: ambas as metodologias se concentram na entrega de valor, ambas se concentram nos clientes/usuários, e sua natureza iterativa e testes contínuos são as principais semelhanças para integrá-los (HUSSAIN; SLANY; HOLZINGER, 2009).

2.3.2 Metodologias UCD

Diferentes metodologias foram relatadas para integrar *Scrum* e UCD como: *Design in Parallel to Sprints*, *Design Within Sprints*, *Lean UX* e *Design Thinking* (ARGUMANIS; MOQUILLAZA; PAZ, 2020):

1. *Design in Parallel to Sprint* - A primeira metodologia relatada é o *Design* em Paralelo com *Sprints* (em inglês, *Design in Parallel to Sprints*), onde os designers poderiam apoiar os desenvolvedores na implementação de uma *sprint*, enquanto projeta a interface para a próxima *sprint*;
2. *Design Within Sprint* - Para (ARGUMANIS; MOQUILLAZA; PAZ, 2020) o *Design* em Paralelo com *sprint* pode implicar em conflitos na comunicação, pois o projeto

de desenvolvimento seria dividido em duas equipes. Portanto, o Design dentro da sprint (em inglês, Design Within Sprint) propõe uma metodologia onde o design está inserido dentro dos sprints (M3), o que garantiria uma comunicação eficaz, mas exigiria cooperação da equipe desde o início do projeto;

3. *Lean UX* - O *Lean UX* foca em minimizar risco de desperdício de recursos e tempo pelo lançamento de protótipos rápidos (produto mínimo viável) de forma que sejam facilmente editáveis para serem testados pelos usuários (ARGUMANIS; MOQUILLAZA; PAZ, 2020);
4. *Design Thinking* - Por fim, o *Design Thinking* (DT) é um modelo na inovação centrada no usuário para criar, testar e refinar um produto ou serviço que resolva um problema complexo com uma solução simples para o cliente, podendo envolver toda a equipe na geração de ideias de design (NEDELTCHEVA; SHOIKOVA, 2017).

2.3.3 Técnicas UCD

Com base nos resultados das metodologias levantadas, as seguintes técnicas foram considerada mais importantes (ARGUMANIS; MOQUILLAZA; PAZ, 2020):

1. Protótipos de papel - Protótipos de papel permitem que projetos de interface de usuário sejam testados antes de serem implementados em software. Os protótipos de papel são mais fáceis de criar e alterar do que o software e podem ajudar a identificar os principais problemas de usabilidade antes de sofrer os custos de implementação. Como os protótipos de papel não incluem computação, um dos designers deve desempenhar o papel de “computador” para operar o protótipo de papel (FELKER; SLAMOVA; DAVIS, 2012).
2. Personas - A técnica de Personas desenvolvida pela disciplina de interação humano-computador (IHC) é útil para elicitar requisitos de maneira centrada no uso. Uma Persona é uma descrição de um membro representativo dos usuários. Personas ajudam a focar a análise e o design de software nos recursos e objetivos do usuário final do produto (LOSADA, 2018).
3. Sprint 0 - Essa técnica é usada para criar a estrutura básica de um projeto para que as *sprints* futuras possam ser realmente incrementadas. Inclui a construção de uma estrutura suficientemente flexível que facilite o processo de refatoração. Os designers de *UX* geralmente projetam duas *sprints* à frente para estarem prontos para a equipe de desenvolvimento na *sprint* certa.
4. Design em pares - Design em pares significa que dois designers trabalham no mesmo documento de design no mesmo computador e ao mesmo tempo: o primeiro designer

chamado *condutor* escreve ativamente o documento e o outro, chamado *observador*, revisa. As duas funções podem ser trocadas durante o trabalho conforme necessário: isso geralmente acontece quando o condutor não sabe como proceder, enquanto o observador sabe como resolver o problema. O observador também pode realizar diferentes atividades a partir de revisões que são úteis para atingir o objetivo da tarefa atual. O design de pares pode ajudar os membros de uma equipe a obter rapidamente um conhecimento completo e correto do software (BELLINI et al., 2005).

5. Avaliação heurística - A avaliação heurística é um método informal de análise de usabilidade em que vários avaliadores são apresentados a um design de interface e solicitados a comentar sobre ele. Avaliação heurística é feita olhando para uma interface e tentando chegar a uma opinião sobre o que é bom e ruim sobre a interface. o ideal é que as pessoas conduzam essas avaliações de acordo com certas regras, como as listadas em documentos típicos de diretrizes. Algumas diretrizes possuem cerca de mil regras a serem seguidas e são vistas como intimidadoras pelos desenvolvedores (NIELSEN; MOLICH, 1990).
6. *Card Sorting* - É um método estabelecido para elicitación de requisitos e tem sido amplamente utilizado em várias áreas, como Psicologia, Engenharia do Conhecimento, Engenharia de Software e Design de Sites. No campo da Engenharia de Requisitos, a classificação de cartões é descrita como o método mais eficaz para elicitación domínios de problemas de engenharia de requisitos (NURMULIANI; ZOWGHI; WILLIAMS, 2004). Foi demonstrado que o *Card sorting* têm muitos aspectos positivos que o torna uma ferramenta útil de elicitación. Primeiro, as classificações de cartão podem ser usadas para investigar o conhecimento sobre a entidade de domínio. Em segundo lugar, a classificação de cartões é uma técnica útil para distinguir entre problemas de alto e baixo nível. Em terceiro lugar, o *card sorting* oferece mais informações sobre as opiniões do público alvo sobre o tópico. Quarto, os resultados do *card sorting* podem fornecer uma entrada para outra técnica e análise posterior. Os resultados também podem ser usados como entrada para uma hierarquia ou classificação adicional (por exemplo, arquiteturas de informação para sites da web). Finalmente, o processo de o *card sorting* pode ser feito de forma relativamente rápida, a um custo nominal, e é flexível e fácil de manusear pelo pesquisador (NURMULIANI; ZOWGHI; WILLIAMS, 2004).
7. Investigação contextual - A investigação contextual é um método com raízes na etnografia que examina os contextos dos usuários para garantir que as tecnologias atendam às suas necessidades. Em tarefas de sala de aula, bem como em projetos de pesquisa de UX fora da sala de aula, pode ser difícil incorporar os dois métodos. Ao mesmo tempo, é essencial que estudantes e pesquisadores avaliem o contexto de

um usuário e suas ações ao usar uma tecnologia para aprender como essa tecnologia pode ser aprimorada (GETTO, 2020).

8. *Thinking aloud* (Pensar em voz alta) - O método de pensar em voz alta tem sido um instrumento amplamente utilizado para estudar processos cognitivos, como resolução de problemas, interação humano-computador. Os participantes do *thinking aloud* são solicitados a realizar uma tarefa, enquanto verbalizam seus pensamentos. Os pesquisadores registram todas as verbalizações, anotam-nas em um relato verbal e, em seguida, analisam-nas de uma forma que depende das questões de pesquisa. Os pesquisadores podem, por exemplo, prestar atenção a enunciados que refletem um certo estado de espírito (aborrecimento, confusão) ou a atrasos entre os enunciados. Os dados podem ser analisados de forma qualitativa ou quantitativa (KRAHMER; UMMELLEN, 2004).
9. *Customer journey map* (Mapeamento da jornada do cliente) - A ideia fundamental por trás do mapeamento da jornada do cliente é relativamente simples, é uma representação visual da sequência de eventos por meio dos quais os clientes podem interagir. O Mapeamento da jornada do cliente lista todos os possíveis pontos de contato organizacionais que os clientes podem encontrar durante a utilização dos serviços. Ao compreender claramente os pontos de contato com o cliente, a gerência pode trabalhar com membros da equipe multifuncional para empregar táticas que promovam a inovação do serviço. O objetivo dessas táticas é aprimorar as interações do provedor de atendimento ao cliente, melhorando a experiência do cliente associada a cada ponto de contato (ROSENBAUM; OTALORA; RAMÍREZ, 2017). Segundo (FOUNDATION, 2019) o Design centrado no usuário (UCD) é um processo de design iterativo no qual os designers se concentram nos usuários e em suas necessidades em cada fase do processo de design. No UCD, a equipe de design envolve os usuários em todo o processo de design por meio de uma variedade de técnicas de pesquisa e design, para criar produtos altamente utilizáveis e acessíveis para eles.

2.4 Framework SCRUM-UCD

O *framework Scrum-UCD* busca criar uma estrutura que inclui técnicas e metodologias de *UCD* no *Scrum*. Para que a integração do *scrum* com o *UCD* fosse possível foram selecionadas algumas técnicas para serem incluídas no *framework* (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

2.4.1 Análise comparativa entre as metodologias UCD

De acordo com as metodologias UCD citadas na Seção 2.3.1, os autores ([ARGUMANIS; MOQUILLAZA; PAZ, 2021](#)) realizam uma análise comparativa entre as metodologias analisando a vantagem, desvantagem, onde usar e onde não usar, apresentadas na Tabela 3.

Tabela 2 – Análise comparativa entre as metodologias UCD

	Design em paralelo com sprints	Design dentro da sprint	<i>Lean UX</i>	<i>Design Thinking</i>
Vantagem	Facilita o planejamento do projeto	Design é validado tanto pelo time de desenvolvedores quanto pelo time de design	Desperdício mínimo de recursos	Melhor entendimento das necessidades do usuário
Desvantagem	Dificulta a comunicação do time	Difícil sincronização das atividades	Alto custo	Viável somente para inovações
Onde usar	Grandes projetos	Pequenos projetos	Apenas em projetos novos	Projetos inovadores
Onde não usar	pequenos projetos	Grandes projetos	Redesenho ou melhoria do projeto	Redesenho ou melhoria do projeto

Fonte: ([ARGUMANIS; MOQUILLAZA; PAZ, 2021](#))

2.4.1.1 Metodologias selecionadas para o Scrum-UCD

O *Design* em paralelo com *Sprint* foi selecionado porque facilita o planejamento das iterações e pode ser adaptado a quase todos os projetos. No entanto, considerou-se que os desenvolvedores devem participar da validação dos protótipos, e que ambas as equipes devem participar juntas em todas as cerimônias do *Scrum*. Isso é necessário para melhorar a comunicação entre ambas as partes, especialmente em projetos pequenos ([ARGUMANIS; MOQUILLAZA; PAZ, 2021](#)).

2.4.2 Análise comparativa entre as técnicas UCD

Nas Tabelas 4, 5 e 6 apresentam-se uma análise comparativa entre as técnicas UCD, apresentada pelos pesquisadores ([ARGUMANIS; MOQUILLAZA; PAZ, 2021](#)).

Tabela 3 – Análise comparativa entre as técnicas UCD 1

	Protótipo de papel	Personas	<i>Sprint 0</i>	Design em pares
Vantagem	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
Desvantagem	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
Custo	Pequeno	Pequeno	Mediano	Pequeno
Estágio do projeto	Iterativo	Preliminar	Preliminar	Iterativo
Onde usar	Qualquer projeto	Projetos com usuários desconhecidos	Projetos com pequena visão inicial	Projetos complexos
Onde não usar	-	Projetos com usuários conhecidos	Projetos com alta visão inicial	Projetos pequenos

Fonte: (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

Tabela 4 – Análise comparativa entre as técnicas UCD 2

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

Fonte: (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

Tabela 5 – Análise comparativa entre as técnicas UCD 3

	Design conduzido por desenvolvedores	Thinking aloud	<i>Scenarios</i>	Mapa da Jornada do Cliente
Vantagem	Diminui a sobrecarga de trabalho dos designers	Fornecer uma ideia precisa de UX sobre o produto	Definição mais fácil sobre a visão estruturada do projeto	Fornecer conhecimento sobre os processos do usuário para atingir seus objetivos
Desvantagem	Desenvolvedores não são motivados para trabalharem com design	Requer muito planejamento para ser efetivo	A informação recebida não é detalhada	Não fornece informação sobre funcionalidade
Custo	Alto	Mediano	Mediano	Mediano
Estágio do projeto	Preliminar	Iterativo	Preliminar	Preliminar
Onde usar	Projetos pequenos	Qualquer projeto	Qualquer projeto	Qualquer projeto
Onde não usar	Grandes projetos	-	-	-

Fonte: (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

2.4.2.1 Técnicas selecionadas para o Scrum-UCD

A partir das 12 técnicas analisadas, os autores (ARGUMANIS; MOQUILLAZA; PAZ, 2021) selecionaram 9 técnicas:

- a) **Protótipo de papel:** Foi selecionado porque facilita o design e a avaliação de protótipos iterativos, minimizando tempo e recursos.
- b) **Personas:** Foi selecionado para garantir que toda a equipe tenha uma ideia clara dos usuários. Em projetos totalmente novos, as Personas serão criadas do zero e, para reformulações de projetos, as Personas existentes serão atualizadas.
- c) **Sprint 0:** Foi selecionado devido à necessidade dos designers de um espaço para entender as necessidades dos usuários e definir uma visão global das interfaces. Os desenvolvedores podem utilizar este espaço para definir a arquitetura do sistema e as ferramentas que utilizarão, enquanto auxiliam os designers em suas atividades de UCD.
- d) **Pair design:** Foi selecionado para tornar as *sprints* de design mais eficientes, pois os desenvolvedores poderão validar se o design atende às restrições técnicas do projeto, e poderão ajudar na melhoria do design, se necessário.
- e) **Card sorting:** Foi selecionado apenas para projetos que requerem informações estruturadas, pois facilita o processo de definição da estrutura.
- f) **Avaliação Heurística:** Foi selecionado apenas para projetos de redesenho, a fim de permitir que o especialista em IHC consigam identificar a maioria dos problemas de um projeto existente.
- g) **Investigação contextual:** Ele foi selecionado para reunir o conhecimento das atividades do usuário, o que é importante para o processo de definição de requisitos.
- h) **Thinking aloud:** Ele foi selecionado por ser eficaz para obter informações sobre a experiência do usuário ao utilizar o sistema.
- i) **Mapa da Jornada do Cliente:** Foi selecionado porque é eficaz para obter uma ideia do processo de negócios que os usuários vão seguir e, portanto, identificar detalhadamente as necessidades dos usuários.

A partir das *técnicas* e das *metodologias* UCD selecionadas, os autores propuseram um *Framework*.

2.4.3 Fases do Framework Scrum -UCD

Os autores (ARGUMANIS; MOQUILLAZA; PAZ, 2021) propuseram um *Framework SCRUM-UCD* composto por três fases: Iniciação; Planejamento; e Implementação, detalhadas em seguida.

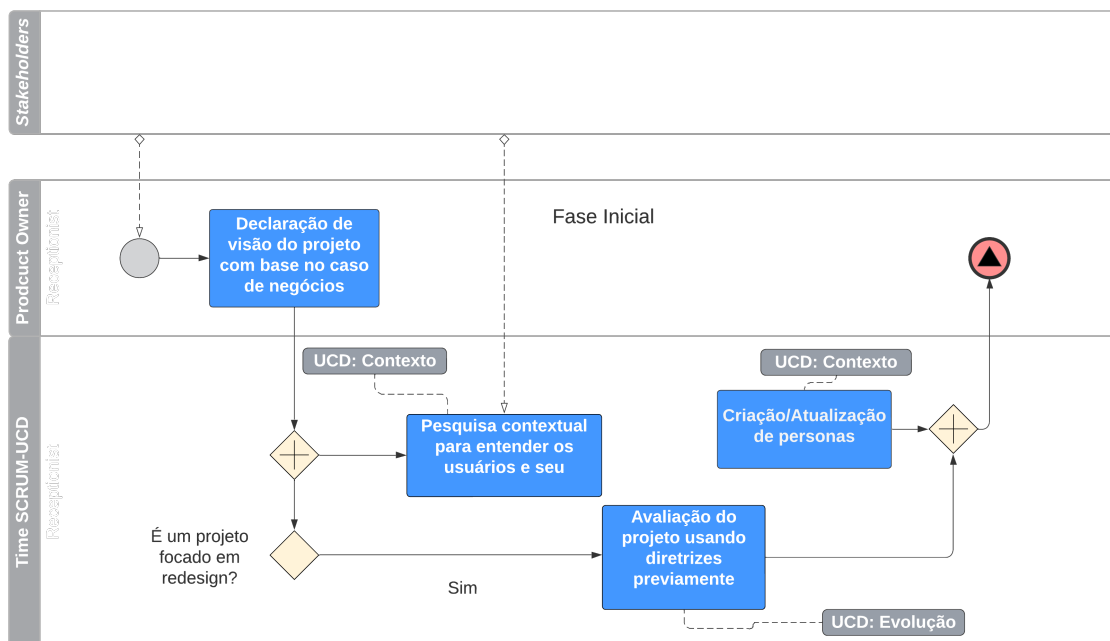
2.4.3.1 Fase de iniciação

A primeira fase do *framework* começa com a declaração da visão do projeto com base nas regras de negócio do produto. O *Product Owner* cria um documento definindo a visão do projeto, que é aprimorada após um processo de consulta contextual que permite que a equipe Scrum-UCD conheça os usuários reais e seu contexto, para poder identificar suas necessidades de maneira transparente (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

Caso o projeto a ser construído seja totalmente novo, os especialistas UCD criam Personas, elaborando perfis de usuários com as informações coletadas durante a consulta contextual. Caso seja um projeto em fase de refatorações os especialistas UCD realizam *design evaluation* utilizando diretrizes previamente estabelecidas, onde a equipe deveria decidir as diretrizes mais adequadas para o projeto. Também atualizam as Personas existentes levando em consideração as novas necessidades dos usuários identificados caso não existam Personas, novas Personas deverão ser criadas (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

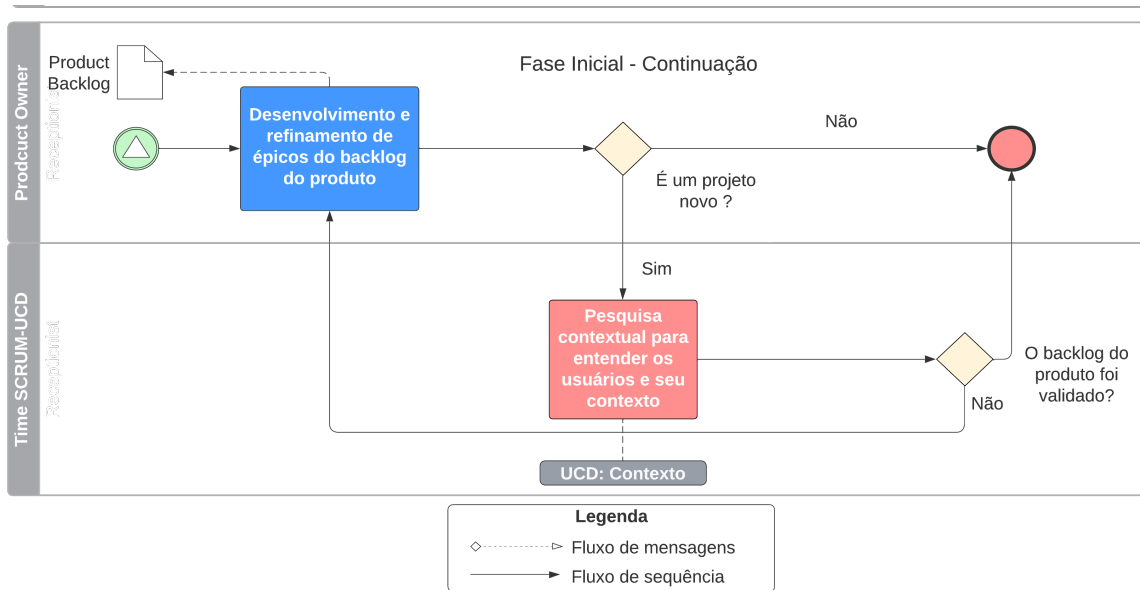
Por ultimo, o backlog do produto é construído listando todos os requisitos do projeto, e após ser validado pelos usuários poderá ser melhorado continuamente durante o desenvolvimento do projeto. Uma vez aceitos os requisitos, inicia-se a fase de planejamento (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

Figura 3 – Fase de iniciação



Fonte: Adaptado de (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

Figura 4 – Fase de iniciação - Continuação



Fonte: Adaptado de (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

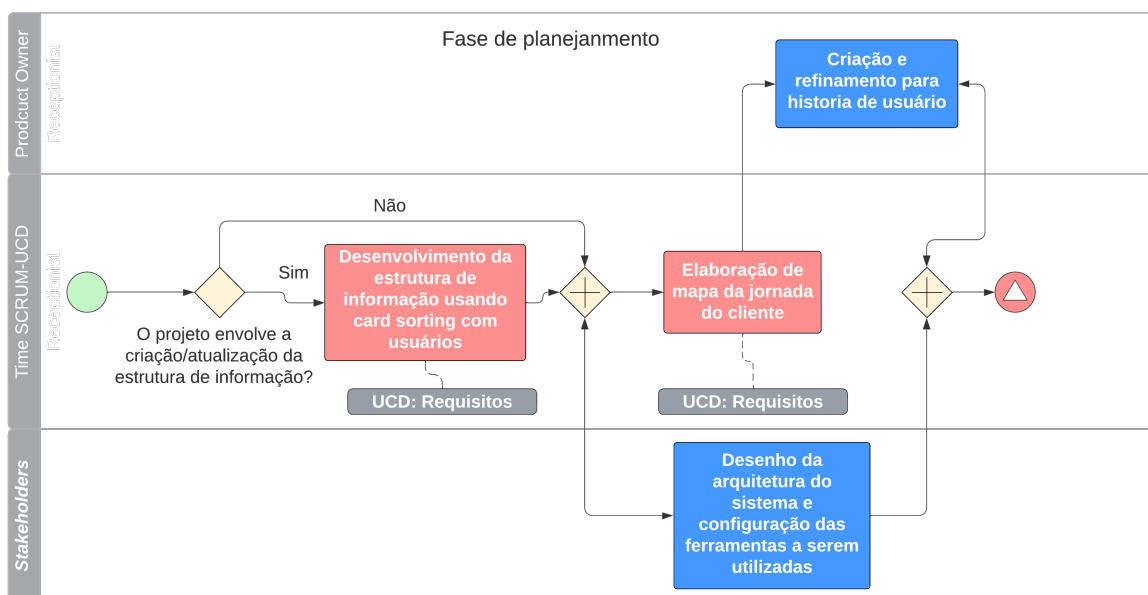
2.4.3.2 Fase de Planejamento

A segunda fase do *framework* envolve o planejamento do projeto. Caso o projeto envolva a criação ou redefinição da estrutura de informação de um sistema, os especialistas UCD utilizam a técnica de *card sorting* com os usuários. Em seguida, independentemente do tipo de software, os especialistas UCD desenvolvem um mapa da jornada do cliente, com o objetivo de definir detalhadamente o fluxo de trabalho que os usuários seguirão ao utilizar o sistema. Ao mesmo tempo em que é realizada o mapa da jornada do cliente a equipe de desenvolvedores realizam as definições arquiteturais do sistema e as configurações das ferramentas de desenvolvimento (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

Com o auxílio dessas técnicas, as histórias de usuários são criadas e estimadas, e a equipe inicia o projeto de protótipos em papel junto com o *Product Owner* e *stakeholders*, definindo uma visão global do sistema. Por fim, as tarefas são identificadas e incluídas no *sprint backlog*, que é criado para as *sprints* de design e desenvolvimento (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

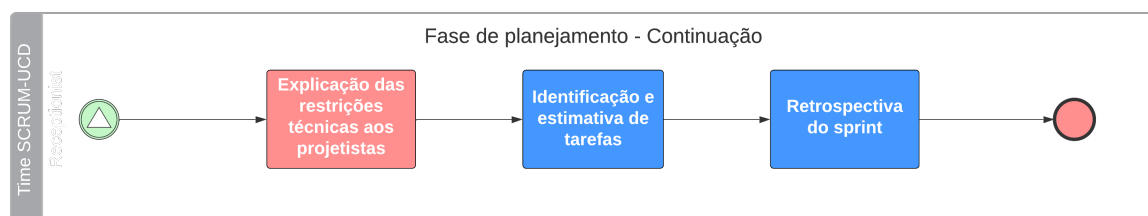
Antes do início da implementação, os desenvolvedores explicam as restrições técnicas do projeto aos especialistas UCD, para que possam ser consideradas durante a concepção dos protótipos. Essas restrições são definidas pelas plataformas nas quais o produto será desenvolvido, pelas limitações de tempo e orçamento e pelas capacidades da equipe de desenvolvimento (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

Figura 5 – Fase de planejamento



Fonte: Adaptado de (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

Figura 6 – Fase de planejamento - Continuação



Fonte: Adaptado de (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

2.4.3.3 Fase de Implementação

O Design *sprint* começa com a criação de protótipos de papel. Em seguida, um designer e o desenvolvedor se sentam juntos com o objetivo de avaliar o design através do processo de design em pares. Caso o desenvolvedor identifique um recurso de design que não possa ser implementado pelos desenvolvedores, ele trabalha em conjunto com o designer para aprimorar os protótipos para que possam estar alinhados com as restrições técnicas do projeto. Em seguida, os especialistas UCD se reúnem com o *Product Owner* para validar e melhorar os protótipos. Em seguida, eles começam a criar protótipos de baixa fidelidade, que são posteriormente validados pelos usuários usando a técnica que a equipe identifica como a mais eficiente levando em consideração as limitações de tempo e recursos. (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

Caso o projeto envolva o desenvolvimento de um grande sistema, o processo anterior é repetido iterativamente até que os protótipos sejam validados com sucesso pelos usuários. Quanto menor o projeto, menor o risco de desperdício de recursos caso uma mudança seja solicitada em etapas posteriores da implementação, assim projetos pequenos podem executar essa validação apenas uma vez e começar a desenvolver os protótipos de alta fidelidade, levando em consideração as observações do usuário que foram apontados durante os testes de protótipos de baixa fidelidade. (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

FALTA O DIAGRAMA!!!!

2.5 Considerações Finais do Capítulo

Nesse capítulo foram descritos os *frameworks Scrum* e *Kanban*, além de um detalhamento do *framework Scrumban*, adotado pela empresa. Uma seção introdutória ao tema Design Centrado no Usuário (*User-Centered Design - UCD*), e um detalhamento do *Scrum-UCD, framework* objeto de estudo deste trabalho. No capítulo seguinte, os materiais e métodos adotados nesta pesquisa.

3

Materiais e Métodos

3.1 Considerações Iniciais

Neste capítulo é retomado o plano metodológico apresentado brevemente no *Capítulo 1*. Esta pesquisa é do tipo Descritiva com o emprego da Técnica Estudo de Caso. Nas seções seguintes apresenta-se um detalhamento da proposta deste trabalho, com o planejamento e seleção de técnicas e atividades.

3.2 Planejamento de Pesquisa

Nesta fase foram definidos o tema de pesquisa, a questão de pesquisa, o objetivo a ser atingido e a classificação metodológica, assim como o Plano de Pesquisa. O Capítulo 1 contempla esta fase. A técnica de *Estudo de Caso* foi utilizada dado que o trabalho é a comparação de um processo de uma *startup* com o *framework Scrum-UCD*.

Neste trabalho foi adotado o Protocolo proposto por Brereton *et al.* (BRERETON *et al.*, 2008), permitindo aos pesquisadores se orientarem no planejamento dos onze itens do Estudo de Caso apresentados na Tabela 6.

Tabela 6 – Estrutura do protocolo de estudo de caso

	Item	Descrição
a)	Background	Buscar pesquisas anteriores sobre o tema, definir questão de pesquisa principal abordada neste estudo; identificar todas questões adicionais a serem abordadas.
b)	Design	Identificar se serão usados projetos de caso único ou múltiplo; descrever o objeto de estudo; identificar propostas de cada questão de pesquisa.
c)	Seleção de caso	Definir os critérios para seleção de casos.
d)	Procedimentos e papéis do estudo de caso	Definir regras para condução dos procedimentos de campo; definir os papéis de cada membro da equipe de pesquisa.
e)	Coleta de dados	Definir os dados que serão coletados; definir um plano de coleta de dados; definir como os dados serão armazenados.
f)	Análise dos Dados	Identificar os critérios para interpretação dos dados; Os dados para responder as questões de pesquisa; Seguido da gama dos resultados possíveis.
g)	Validade do plano	Verificar a validade do planejamento em relação aos aspectos gerais, a validade do construto, a validade interna e a validade externa.
h)	Limitações do estudo	Especificar as questões de validade residual incluindo potenciais conflitos de interesse.
i)	Relatórios	Identificar o público-alvo e o relacionamento com outros estudos; E descrição dos resultados.
j)	Apêndices	Atualizar durante a execução do estudo observando quaisquer diferenças das etapas acima.

Fonte: Adaptado de (BRERETON et al., 2008)

3.3 Etapas do Protocolo de Estudo de Caso

Nessa seção apresenta-se o planejamento da coleta de dados e análise dos resultados, conforme o Protocolo de Estudo de Caso adotado.

Para cada item, foram detalhados os procedimentos realizados, desde a seleção até o desenvolvimento das etapas.

3.3.1 *Background*

A utilização da pesquisa bibliográfica permitiu a inserção deste estudo em um contexto científico mais amplo, além de auxiliar na formulação da questão de pesquisa e dos objetivos, os quais foram apresentados no Capítulo 1.

Através da pesquisa bibliográfica, foi possível estabelecer o embasamento teórico deste estudo, conforme apresentado no Capítulo 2, onde se procurou identificar abordagens que pudessem auxiliar na otimização dos processos da VP.

3.3.2 Design

No que se refere ao seu *design*, esta pesquisa é categorizada como um estudo de caso único, uma vez que se concentra na análise da *fintech* VP como objeto de estudo.

3.3.3 Seleção de caso

A partir da pesquisa documental, foi possível descrever o objeto de estudo deste trabalho, que é a *startup* VP, e também foi realizada uma descrição detalhada do seu processo de desenvolvimento de software.

O resultante dessa etapa é apresentado no Capítulo 4.

3.3.4 Procedimentos e papéis do estudo de caso

Os autores desta pesquisa são desenvolvedores da organização que está sendo estudada e contam com o apoio do *CEO*, Sr. Pedro Rosa. Os procedimentos adotados neste estudo foram definidos em conformidade com os objetivos específicos e consistem em duas atividades planejadas:

- Analisar o Processo da *Startup* VP quanto aos *frameworks e conceitos de Design Centrado no Usuário* da literatura para a criação de diretrizes de UCD;
- Propor a integração da *Diretrizes UCD criadas* para o Processo da *Startup* VP quanto ao Design Centrado no Usuário.

3.3.5 Coleta de dados

O item Coleta de Dados do Protocolo constitui o emprego dos procedimentos para a realização de atividades que possibilitaram o alcance dos 02 objetivos específicos:

- Analisar o Processo da *Startup* VP quanto aos *frameworks e conceitos de Design Centrado no Usuário* da literatura para a criação de diretrizes de UCD;
- Propor a integração da *Diretrizes UCD criadas* para o Processo da *Startup* VP quanto ao Design Centrado no Usuário.

A descrição desta coleta foi realizada por meio da técnica de Pesquisa Documental, a qual consiste em buscas por documentos referentes a Vamos Parcelar. Dessa forma, a busca instigou questionamentos sobre possíveis melhorias dos processos executados.

A descrição detalhada desta coleta é apresentada nos Capítulos 4 e 5

3.3.6 Análise dos Dados

Neste trabalho, a *Análise dos Dados* definiu-se em expor as dificuldades encontradas na Pesquisa Documental da Vamos Parcelar.

A partir dos resultados, foi realizada uma análise qualitativa para obter um diagnóstico que recomendasse diretrizes para a solução dos contratemplos gerados.

A partir da análise qualitativa, foi examinado o resultado da maturidade da organização avaliada e a identificação das lacunas nos processos de software. Contudo, ao final, as diretrizes apontadas foram apresentadas aos diretores da *startup*.

A descrição da Análise de Dados é apresentada no Capítulo 6

3.3.7 Validade do plano

O objetivo é identificar o domínio para o qual os achados do estudo podem ser generalizados.

3.3.8 Limitações do Estudo

O escopo desta pesquisa está delimitado a um processo de um projeto, restringindo a generalização dos resultados.

3.3.9 Relatório

O Relatório desta pesquisa se materializa como sendo esta monografia de TCC.

3.3.10 Apêndice

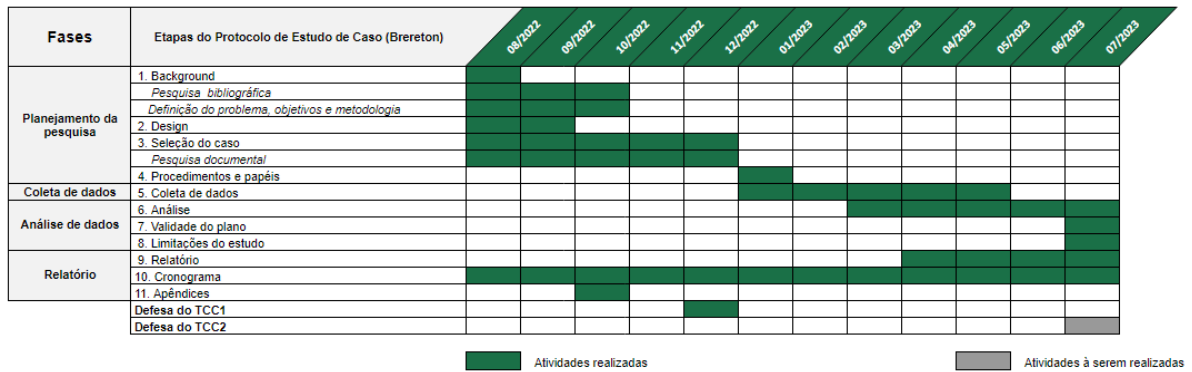
Os documentos produzidos pelos autores deste trabalho, como a carta de apresentação à empresa, constam como apêndices.

- Apêndice A - Carta de solicitação de acesso e citação

3.3.11 Cronograma

O cronograma de atividades deste estudo de caso foi delineado durante a fase de elaboração da monografia e cuidadosamente acompanhado e atualizado ao longo de todo o desenvolvimento desta pesquisa. Na Figura 7, estão descritas as atividades realizadas ao longo desta pesquisa.

Figura 7 – Cronograma de pesquisa



Fonte: Autores

3.4 Considerações Finais do Capítulo

Nesse capítulo foram descritos o plano metodológico e um detalhamento da proposta deste trabalho, com o planejamento e seleção de técnicas e atividades. No Capítulo seguinte apresenta-se um detalhamento da empresa, objeto de estudo deste trabalho, com foco no processo de desenvolvimento de software adotado, baseado no *framework* Scrumban.

4

Startup *Vamos Parcelar*

4.1 Considerações Iniciais

Neste capítulo apresenta-se uma breve descrição da empresa, objeto de estudo deste trabalho, com foco no processo de desenvolvimento de software baseado no *framework Scrumban*, adotado pela empresa.

4.2 Startup *Vamos Parcelar* (VP)

A Startup *Vamos Parcelar* (VP) foi fundada em 2018 pelo sr. Pedro Rosa, que atualmente ocupa o cargo de chefe executivo (*Chief Executive Officer* - CEO). A VP é uma empresa que busca facilitar o pagamento de boletos bancários, utilizando diversos cartões de crédito e permitindo o parcelamento dos pagamentos, auxiliando seus usuários no pagamento de impostos e contas diversas. Nas seções seguintes apresenta-se uma síntese dos processos adotados para desenvolvimento de software pela empresa VP. Esses processos são baseados no Scrum, Kanban e Scrumban. São documentos internos da própria empresa. A empresa apresenta dados públicos em seu site [Vamos parcelar](#) .

4.3 *Scrumban* VP

Na Startup VP foi definido um processo de desenvolvimento baseado no *framework Scrumban*. A equipe é dividida entre Time de Produto, *Scrum Master*, Time de desenvolvimento, Analistas de Qualidade, Time de infraestrutura e Time de *Designers*.

4.3.1 Time de Produto

- Product Owner

O *Product Owner* (PO) é responsável por definir os elementos do produto que serão desenvolvidos, priorizar as funcionalidades do *Backlog* do produto de acordo com o valor de negócio e garantir com que o time de desenvolvimento entenda todos os itens a serem implementados.

- Analista de negócios

O Analista de negócio tem por objetivo analisar demandas e necessidades dos *Stakeholders*, e documentar os requisitos, delimitar escopo a ser desenvolvido e avaliar os processos de negócios. Além disso, deve garantir que os requisitos funcionais e não-funcionais estabelecidos sejam viáveis perante arquitetura técnica adotada e validá-los com o responsável pela área de negócio.

4.3.2 Scrum Master

O *Scrum Master* na VP atua como facilitador das reuniões feitas pelos times, é o responsável por remover os problemas levantados durante essas reuniões, calcula e divulga os prazos das demandas solicitadas para toda a equipe.

4.3.3 Time de desenvolvimento

- Analista desenvolvedor

O Analista desenvolvedor da VP tem como papel realizar a quebra técnica das atividades que serão executadas no processo de desenvolvimento, mapear e modelar o banco de dados dos sistemas e desenvolver os sistemas de acordo com as regras de negócios estabelecidas pelo Time de Produto. Além disso, também são responsáveis por verificar o desempenho dos sistemas já implantados e propor soluções para melhorias.

- Líder técnico de desenvolvimento

O líder técnico é responsável por gerenciar e coordenar as atividades dos analistas desenvolvedores, sendo o principal porta-voz entre o time de desenvolvimento e o demais times da equipe. Além disso, ele organiza o time e quem especifica quem irá executar cada atividade de desenvolvimento.

4.3.4 Analista de Qualidade

O Analista de qualidade, também conhecido como Analista de *Quality Assurance*(QA), possui a função de realizar um processo de casos de testes com o objetivo de garantir o funcionamento correto do sistema. Além disso é responsável por reportar erros para as equipes de desenvolvimento, para que possam solucionar o rápido possível.

4.3.5 Time de infraestrutura

- *DevOps*

O DevOps (Desenvolvedor e Operações) é o responsável por otimizar e garantir a confiabilidade das operações durante o processo de desenvolvimento, como ponto principal a unificação e a automação de processos.

- Analista de infraestrutura

O Analista de Infra Estrutura é o responsável pelo planejamento, implementação, suporte, manutenção e monitoramento de redes de computadores da empresa. Assim, garante a segurança de dados, cria políticas de seguranças e previne a empresa contra invasões.

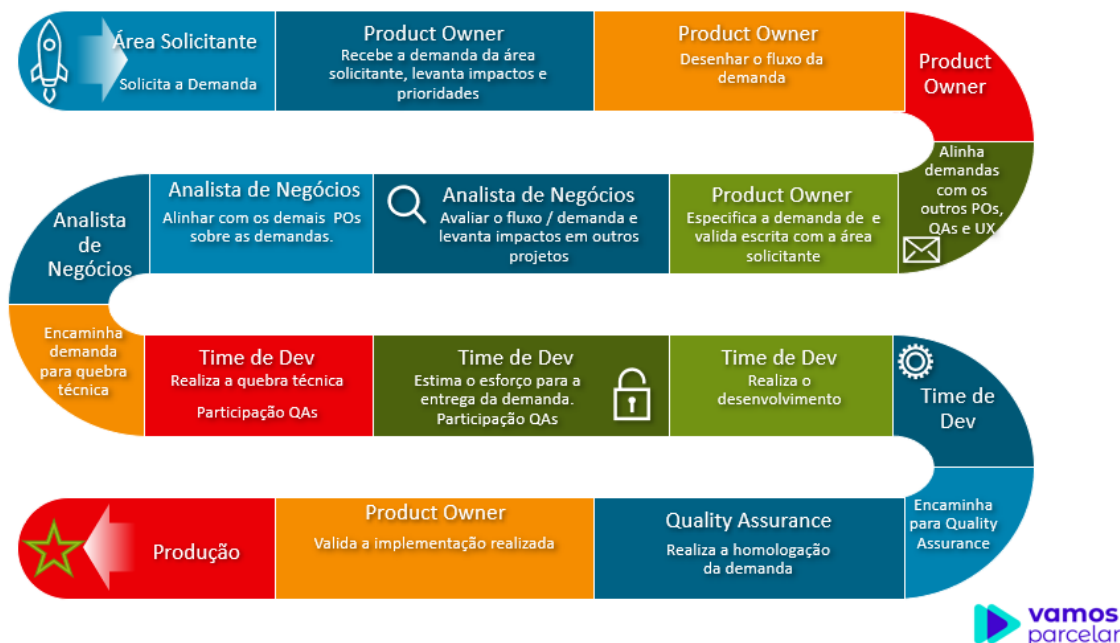
4.3.6 Time de Design

O Time de design tem a responsabilidade de criar protótipos de alta fidelidade de acordo com o que foi proposto pelo Time de produto.

4.3.7 Processos da VP

Descrito as características dos times acima, serão apresentados os ritos e técnicas do *Scrumban* da VP e como se dá a comunicação e demanda entre os times. A figura 8 exemplifica o processo de desenvolvimento

Figura 8 – Processo de desenvolvimento da VP



Fonte: Vamos Parcelar.

4.3.8 Planejamento de desenvolvimento

O Time de produto recebe demandas para criação de um novo projeto ou melhorias de um projeto já existente, um *Product Owner* é selecionado dentro do time de POs que será responsável pelo respectivo produto.

Após isso, o PO identifica o(s) problemas(s) a resolver e os *Stakeholders*, levantam impactos, prioridades, desenha o fluxo da demanda, alinha com outros POs, QAs e Designers, especifica e valida a escrita com área solicitante da demanda.

Na especificação pode conter requisitos funcionais e não-funcionais e até mesmo um diagrama de caso de uso elaborados pelos Analistas de negócio ou prototipação de parte do produto elaborados pelo Time de design. São descritos o passo a passo de cada funcionalidade bem como suas devidas restrições.

Após a escrita de necessidades do cliente/usuário é realizado a validação desses dados com os *stakeholders*, através de uma reunião com intuito de fazer com que os responsáveis assinem o documento para que ele possa ter validade, ou dando ciência da responsabilidade dos requisitos levantados por e-mail. Essa etapa também serve para correções e podem ser descobertos/inclusos outras funcionalidade.

4.3.9 Técnicas de estimativa

Em seguida é alinhado com todo o time de produto participante da demanda e encaminhado para quebra técnica, realizada pelo time de desenvolvimento, definindo assim o *Backlog da sprint*.

Seguidamente é realizado uma estimativa para o desenvolvimento dessas demandas do *Backlog da sprint* através de uma adaptação da técnica de *Planning Poker*. O *Planning Poker* pertence a um conjunto de técnicas de estimativa de grupo que se baseiam no princípio de que mais pessoas pensam melhor do que uma (MAHNIČ; HOVELJA, 2012).

Na VP as histórias de usuários são estimadas em uma reunião com os membros do time de desenvolvimento responsável pela implementação. Para cada história, o Líder Técnico ou outro desenvolvedor com conhecimento técnico e de negócio explica seus respectivos requisitos. Em uma plataforma online e de forma anônima, cada membro leva em consideração o nível de complexidade e tamanho da tarefa e vota em uma sequencia padronizada de valores conforme a Figura 9. A partir desses valores é calculada uma média de todos os votos e estimado o tempo que será gasto na funcionalidade.

Figura 9 – Valores padrão da VP



Fonte: A. Introduction to Planning Poker [Online].

4.3.10 Quadro *Kanban*

Após a quebra técnica e estimativas das atividades, o time de desenvolvimento cria as tarefas detalhadas do que será necessária para concluí-las no Quadro *Kanban* e o PO as prioriza na coluna *Backlog da sprint*. As demais colunas no quadro são representadas por:

- Em Desenvolvimento: Representa as atividades que estavam no *Backlog da sprint* teve seu cartão movimentado para mostrar à equipe que a tarefa está sendo desenvolvida
- Bloqueadas: Demandas com algum impedimento para serem realizadas, o *Scrum Master* é o facilitador responsável por remover quaisquer obstáculos relatados na *Daily*.
- Em Validação técnica: Essa coluna representa as tarefas que foram concluídas pelos desenvolvedores, onde outros desenvolvedores deverão avaliar se todos os requisitos técnicos e regras de negócio forem estabelecidas. Caso as tarefas não estejam de acordo com os critérios estabelecidos, elas retornam para para coluna de Bloqueadas. Se estiver dentro do padrão é movida para coluna Em Publicação(beta).

- Em Publicação (beta): Tarefas que estão prontas para serem publicadas e validadas pelo Time de Qualidade
- Em Homologação: Funcionalidades que foram publicadas em beta e estão sendo testadas e validadas pelo Time de Qualidade.
- *Review*: Após serem validadas pelos QAs, o PO valida se o fluxo de todo o desenvolvimento está de acordo com as regras estabelecidas antes e durante o desenvolvimento. Caso haja requisitos incompletos a tarefa específica daquela funcionalidade volta para a coluna Em Desenvolvimento e realiza-se todo o processo novamente.
- Em produção: Validado pelo PO, o desenvolvimento deve ser publicado em produção para o usuário final utilizar. Neste caso, caso haja algum erro ou bug, é criada uma nova tarefa no quadro onde um desenvolvedor irá solucionar o problema.

4.3.11 *Daily*

A *daily* é um dos ritos do *Scrum* utilizado na VP, porém adaptado. O *Scrum Master* combina alguns dias da semana para fazer a reunião e inspecionar o progresso do desenvolvimento e retirar impedimentos.

4.3.12 Revisão

Após o fim do desenvolvimento ocorre a Revisão, onde toda a equipe participante do produto deve estar presente. O PO tem o objetivo de apresentar quais histórias foram entregues, validar se os critérios de aceitação foram satisfeitos, obter feedback sobre as funcionalidades demonstradas. Já o time de desenvolvimento deve demonstrar as funcionalidades desenvolvidas para o produto.

4.3.13 *Deploy*

Com o desenvolvimento testado, identificadas as falhas, melhorias implementadas e validado pelo Time de Qualidade, PO e *stakeholders* é realizado o *deploy* para produção e disponibilizado para o cliente/usuário poder utilizar. O *deploy* é completamente automatizado usando recursos de ponta. Assim, a segurança, qualidade e eficiência quando se faz atualizações e melhorias nos serviços é de maior garantia.

4.4 Considerações Finais do Capítulo

Nesse capítulo foi descrita brevemente a organização Startup VP e seu processo de desenvolvimento de software, detalhando cada passo até chegar no usuário final. Recentemente a empresa foi adquirida por uma grande instituição financeira.

5

Diretrizes UCD para o Processo Startup VP

5.1 Considerações Iniciais

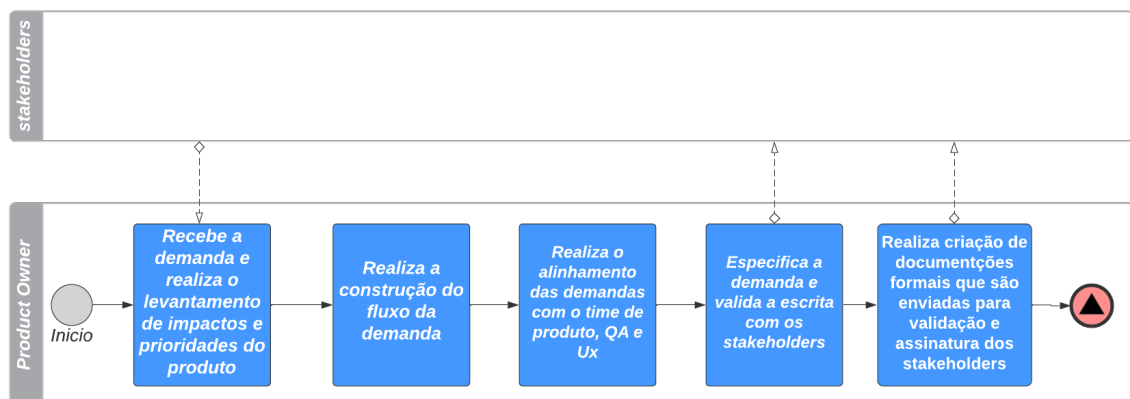
Dado o objetivo geral desta monografia, apresenta-se neste capítulo os procedimentos da análise do processo de desenvolvimento de software da *startup* Vamos Parcelar (VP). Foi criado um *Diagrama do Processo VP*. Em seguida, foi realizada uma *análise das diretrizes UCD*, baseadas na revisão bibliográfica, em relação ao Processo VP. Seguida da *proposição da integração das Diretrizes UCD criadas* para o Processo da *Startup VP*. É apresentado o novo *Diagrama do Processo VP com Diretrizes UCD*. Finalizando, apresenta-se a análise dos dados resultantes desta pesquisa.

5.2 Diagrama do Processo VP

A partir da análise do Processo da VP foi realizada a criação de um fluxograma desse processo.

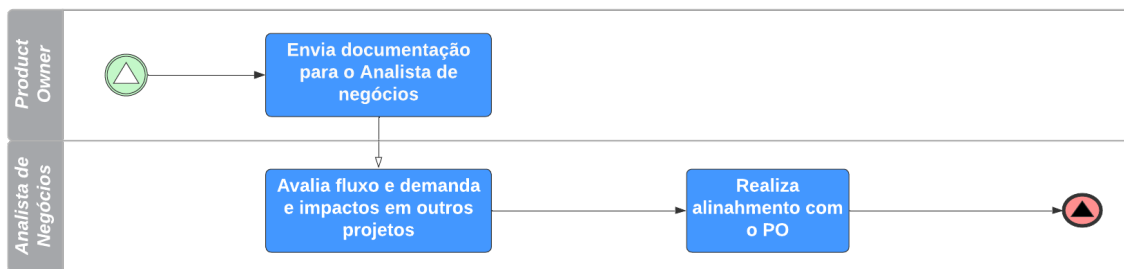
O diagrama do Processo VP é apresentado nas Figuras 10, 11, 12

Figura 10 – Fluxograma Processo VP



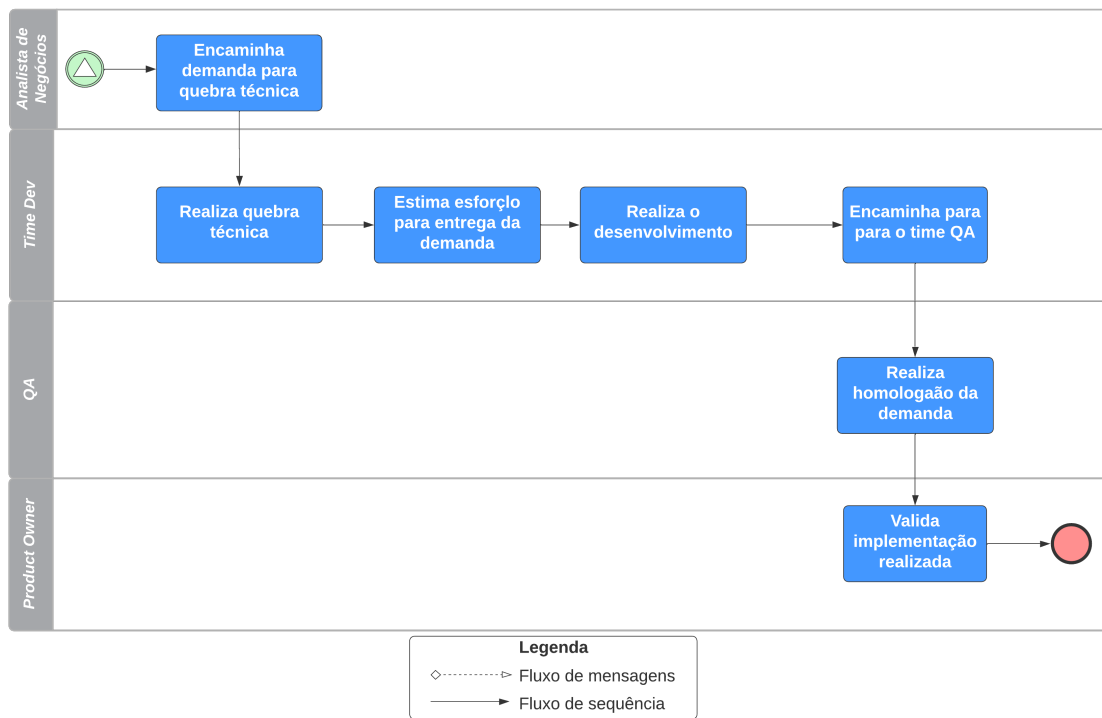
Fonte: Autores.

Figura 11 – Fluxograma Processo VP, parte 2



Fonte: Autores.

Figura 12 – Fluxograma Processo VP, parte 3



Fonte: Autores.

O diagrama BPMN representa o fluxo de trabalho relacionado ao desenvolvimento de um produto, desde o Product Owner recebendo a demanda e conduzindo a análise de impacto e prioridades até a validação da solução implementada pelo Product Owner ao final.

O processo começa com o Product Owner, que recebe as demandas e realiza a análise de impacto e prioridade do produto. Em seguida, o Product Owner constrói o fluxo da demanda e alinha-o com as equipes de produto, QA e UX. Posteriormente, as

demandas são especificadas, e sua redação é validada com as partes interessadas, criando documentação formal que é enviada para validação e aprovação.

A próxima função é a do Business Analyst (Analista de Negócios), que avalia o fluxo e a demanda, e verifica os impactos em outros projetos. O *Business Analyst* alinha-se com o Product Owner e encaminha a demanda para a equipe técnica realizar a quebra técnica.

O Time de Desenvolvimento realiza a quebra técnica, estima o esforço necessário e prossegue com o desenvolvimento. Em seguida, eles repassam a demanda para o Time de QA (Garantia de Qualidade) para homologação.

Após a etapa de homologação, o Product Owner valida a implementação realizada, garantindo que as expectativas foram atendidas. Em caso de aprovação, o produto é encaminhado para o próximo ciclo de desenvolvimento ou para sua disponibilização no mercado. Em caso de ajustes necessários, o fluxo retorna para a equipe de desenvolvimento para as devidas correções.

O diagrama BPMN apresentado neste trabalho proporciona uma visão abrangente e clara do fluxo de desenvolvimento do produto da VP, evidenciando a colaboração entre os diferentes atores envolvidos.

O processo de desenvolvimento da VP não inclui a utilização das diretrizes UCD (*User-Centered Design*), o que pode acarretar desafios e impactos negativos na qualidade e aceitação do produto. Sem essa metodologia, há um maior risco de criar soluções que não atendam completamente às expectativas dos usuários, resultando em produtos menos intuitivos, funcionais e agradáveis de usar. Além disso, a ausência do UCD pode levar a falhas na validação de requisitos e na identificação de oportunidades de inovação, prejudicando a qualidade geral do produto e sua competitividade no mercado. Incorporar o UCD no processo de desenvolvimento é fundamental para garantir que o produto satisfaça verdadeiramente as necessidades dos usuários e proporcione uma experiência positiva.

5.3 Análise do processo da *Startup* VP quanto a literatura do *Design Centrado no Usuário*

Nesta seção, apresentam-se os procedimentos desenvolvidos para atingir um dos objetivos específicos desta monografia :

- Analisar o Processo da *Startup* VP quanto aos *frameworks e conceitos de Design Centrado no Usuário* da literatura para a seleção de diretrizes de UCD para a VP;

Nesse seção, apresentam-se algumas das diretrizes UCD citadas no Capítulo 2, que conforme os autores, são diretrizes a serem consideradas em processos de desenvolvimento.

Segundo os autores, o Design Centrado no Usuário (User Centered Design - UCD) é um processo de design que coloca os usuários no centro do desenvolvimento de produtos ou serviços. Isso é, envolve compreender as necessidades, expectativas e comportamentos dos usuários e utilizar essas informações para orientar o design e o desenvolvimento.

Neste trabalho apresenta-se um levantamento da relevância de algumas diretrizes a serem selecionadas para inserção no processo da VP:

- a) Participação dos Usuários: O UCD requer uma abordagem colaborativa, na qual os usuários são envolvidos desde as fases iniciais do processo. Isso pode ser feito por meio de pesquisas, entrevistas, observações de usuários reais e coleta de feedback. Os usuários devem ser considerados como especialistas no uso do produto ou serviço, e sua participação ativa é essencial.
- b) Compreensão do contexto de uso: É importante entender o contexto no qual o produto ou serviço será utilizado. Isso envolve compreender as tarefas, o ambiente, as restrições e as necessidades dos usuários. Essa compreensão permite projetar soluções que sejam adequadas ao contexto específico e que atendam às necessidades reais dos usuários.
- c) Iteração e feedback contínuos: O UCD é um processo iterativo, no qual o design é refinado com base no feedback dos usuários. Isso significa que é necessário incorporar mecanismos para coletar e analisar o feedback dos usuários ao longo do processo de design e desenvolvimento. A iteração contínua permite ajustes e melhorias, garantindo que o produto final seja otimizado para atender às necessidades dos usuários.
- d) Prototipagem: A prototipagem é uma parte essencial do UCD. Ela permite criar versões preliminares do produto ou serviço para testar e validar com os usuários. Os protótipos podem variar em fidelidade, desde protótipos de baixa fidelidade, como desenhos ou wireframes, até protótipos de alta fidelidade, que se assemelham ao produto final. A prototipagem ajuda a obter feedback dos usuários e a identificar problemas ou oportunidades de melhoria.
- e) Avaliação e testes com usuários: O UCD requer a realização de avaliações e testes com usuários para validar o design e a usabilidade do produto ou serviço. Esses testes podem ser conduzidos em diferentes estágios do processo, desde os estágios iniciais de concepção até a versão final. Os testes com usuários ajudam a identificar problemas de usabilidade, entender as preferências dos usuários e garantir que o produto atenda às suas necessidades.

O processo de desenvolvimento de software utilizado pela *startup* VP (Capítulo 4), foi utilizado como base para realizar a análise quanto aos requisitos que atendem as diretrizes do Design Centrado no Usuário.

Os resultados obtidos apontam a ausência da implementação para a maioria das diretrizes, salvo alguns casos. As análises são apresentadas:

- a) Participação dos Usuários: Ao longo do processo da *startup* Vamos parcelar, em todas as fases, não é estabelecido o envolvimento dos usuários.
- b) Compreensão do contexto de uso: O Time de Produto desempenha um papel crucial na compreensão do contexto de uso ao desenvolver um produto ou serviço. Trabalhando em colaboração com o Analista de Negócios, eles são responsáveis por identificar e documentar os requisitos funcionais e não funcionais do produto, considerando o contexto específico em que será utilizado.
- c) Iteração e feedback contínuos: O processo não é iterativo devido à ausência de análises e coleta de dados do feedback dos usuários.
- d) Prototipagem: O Time de Design cria apenas protótipos de alta fidelidade, porém, esses protótipos não são submetidos a consultas com os usuários para análise de feedbacks.
- e) Avaliação e testes com usuários: A falta de realização de avaliações e testes com usuários impede a identificação de problemas, limita melhorias na usabilidade e compromete a garantia de que o produto atenda adequadamente às necessidades dos usuários.

5.4 Integração da *Diretrizes UCD criadas* para o Processo da *startup* VP

Nesta seção, apresentam-se os procedimentos desenvolvidos para atingir o segundo objetivo específico desta pesquisa:

- Propor a integração da *Diretrizes UCD* selecionadas para o Processo da *startup* VP quanto ao Design Centrado no Usuário.

Dessa forma, apresenta-se algumas das aplicações das diretrizes UCD selecionadas de maneira a priorizar as de maior importância para o processo da *startup* VP.

5.4.1 Aplicação das diretrizes UCD selecionadas para o Processo da *startup* VP

Figura 13 – Fluxograma Diretrizes UCD



Fonte: Autores.

Como proposto pelo fluxograma da figura 13 serão descritos nesta seção os tópicos de Participação dos Usuários, a Compreensão do contexto de uso, as quais se perduram desde o início do desenvolvimento da aplicação das diretrizes até a última etapa, a Iteração e *feedback* contínuos, a Prototipagem e a Avaliação e testes com usuários serão desenvolvidas ao longo da trajetória da aplicação.

5.4.1.1 Participação dos Usuários

No contexto do desenvolvimento de software centrado no usuário, a participação de forma ativa e a compreensão das necessidades dos usuários são aspectos cruciais para o sucesso de um produto. Dessa forma, a Vamos Parcelar poderia iniciar sua abordagem implementando pesquisas de opinião com o objetivo de obter uma compreensão aprofundada dos usuários e suas necessidades.

Essas pesquisas podem ser conduzidas por meio de questionários online, com a finalidade de coletar informações relevantes sobre o perfil demográfico dos usuários, seus objetivos ao utilizar o software e os desafios enfrentados durante a interação com o sistema.

Ao empregar questionários online, a Vamos Parcelar poderia elaborar um conjunto de perguntas estruturadas que abordem questões específicas relacionadas ao desenvolvimento do software. Por exemplo, questões demográficas como idade, gênero e formação acadêmica podem fornecer uma visão sobre a diversidade do público-alvo, assim como auxiliar a definir padrões de interesse.

Sendo assim, as indagações voltadas aos interesses dos usuários, tais como as tarefas que eles pretendem realizar por meio do software e os resultados esperados, podem auxiliar a *startup* a compreender as necessidades funcionais e os benefícios almejados pelos usuários. Adicionalmente, investigar os problemas e obstáculos encontrados pelos usuários no uso do software atual pode proporcionar *insights* sobre as deficiências e áreas de melhoria do sistema.

Esses *insights* fornecerão uma base sólida para direcionar os esforços de desenvolvimento, adaptando recursos, funcionalidades e interfaces do usuário de acordo com as necessidades e expectativas identificadas. Além disso, a inclusão dos usuários no processo de desenvolvimento, desde as fases iniciais até o *feedback* pós-lançamento, pode promover uma maior aceitação e adoção do software, aprimorando a satisfação do usuário e a qualidade geral do produto final.

Portanto, por meio da realização de questionários de opinião detalhados com os usuários, a *startup* estará fortalecendo seu compromisso com a abordagem centrada no usuário, permitindo a criação de soluções mais eficazes e direcionadas às necessidades e expectativas dos usuários finais.

5.4.1.2 Compreensão do contexto de uso

A Compreensão do Contexto de Uso no Design Centrado no Usuário desempenha um papel essencial no desenvolvimento de produtos e serviços que atendam às necessidades e expectativas dos usuários. Nesse sentido, a VP deve adotar uma abordagem cautelosa e abrangente para integrar os *insights* obtidos durante a Participação dos Usuários.

Com base nas percepções obtidas, a Vamos Parcelar poderá criar requisitos de design específicos que atendam às necessidades dos usuários. Esses requisitos devem ser fundamentados nas informações coletadas durante a etapa de Participação dos Usuários, garantindo que o produto ou serviço seja projetado de forma a atender às demandas reais dos usuários. Além disso, é importante definir metas de usabilidade claras, que estabeleçam critérios mensuráveis para avaliar a eficácia, eficiência e satisfação do usuário em relação ao produto ou serviço.

No entanto, a compreensão do contexto de uso não se limita apenas às características dos usuários. A Vamos Parcelar também deve considerar as restrições e requisitos do ambiente em que o produto será utilizado. Isso pode incluir questões técnicas, restrições legais, limitações físicas ou culturais, entre outros aspectos. Ao incorporar esses elementos no processo de design, a *startup* poderá desenvolver soluções mais adequadas e viáveis, a fim de garantir as condições reais de uso.

Por conseguinte, uma abordagem colaborativa é essencial para garantir o sucesso da percepção do contexto de uso no processo de design. Os designers da VP devem trabalhar em estreita colaboração com os usuários e outras partes interessadas, como engenheiros, especialistas em marketing e equipes de suporte ao cliente. Esse envolvimento colaborativo permite uma troca contínua de informações, *feedback* e ideias, garantindo que o produto ou serviço seja desenvolvido levando em conta uma ampla gama de perspectivas.

Ao aplicar efetivamente a compreensão do contexto de uso, a Vamos Parcelar poderá desenvolver produtos e serviços que se destacam no mercado, oferecendo uma experiência aprimorada aos usuários e atendendo às suas expectativas. Além disso, essa abordagem centralizada no usuário pode contribuir para a fidelização dos clientes, o fortalecimento da marca e a conquista de vantagens competitivas. Portanto, a *startup* pode considerar a compreensão do contexto de uso como um componente essencial de sua estratégia de design, investindo tempo, recursos e esforços para incorporar efetivamente os *insights* adquiridos durante a pesquisa de contexto de uso em seu processo de *design*.

5.4.1.3 Iteração e *feedback* contínuos

É fundamental que a equipe de gerenciamento da Vamos Parcelar realize uma análise minuciosa do processo de desenvolvimento atual. Essa avaliação tem o propósito de identificar oportunidades para incorporar a iteração e o *feedback* contínuos. Durante esse processo, é possível identificar lacunas, como o fato de o *feedback* dos usuários ser coletado apenas no final do ciclo de desenvolvimento. Com base nessa observação, é necessário estabelecer etapas específicas para revisar e ajustar o design do produto, levando em consideração o *feedback* dos usuários. Essa abordagem permitirá melhorias contínuas e garantirá que as necessidades dos usuários sejam atendidas de forma eficaz.

Ao implementar essas abordagens, a *startup* estará em uma condição vantajosa para fornecer produtos e serviços que respondam de forma eficaz às necessidades dos usuários, garantindo uma experiência de excelência.

5.4.1.4 Prototipagem

A implementação da prototipagem no contexto da Vamos Parcelar requer uma abordagem cautelosa e estruturada, fundamentada em princípios do design que priorizam o usuário. Inicialmente, é essencial estabelecer uma compreensão precisa dos propósitos e vantagens que a prototipagem pode oferecer para a VP. Esses propósitos podem abranger desde a validação de conceitos até a identificação antecipada de desvantagens, além do aprimoramento da qualidade do produto ou serviço final.

O processo de prototipagem deve ser realizado de maneira iterativa com ciclos de *feedback* contínuos e refinamento dos protótipos. Primeiramente, são criados protótipos de baixa fidelidade permitindo a exploração e a validação preliminar das ideias. Em seguida, esses protótipos são testados com usuários reais, visando coletar *feedback* valioso sobre a usabilidade, a experiência do usuário e a adequação às necessidades e expectativas dos clientes.

Com base nesse *feedback*, os protótipos são aprimorados e refinados em iterações subsequentes, incorporando as necessidades adquiridas durante o processo. É primordial a adoção de uma abordagem colaborativa, envolvendo uma equipe multidisciplinar composta por profissionais especializados em áreas como design, desenvolvimento, *marketing* e usabilidade, para garantir que as decisões dos usuários e interessados sejam fundamentadas em dados e necessidades realísticas.

Além disso, é importante documentar e registrar todas as versões e iterações dos protótipos desenvolvidos, a fim de rastrear o progresso do projeto e facilitar a análise comparativa das diferentes soluções propostas.

Ao implementar a prototipagem de forma adequada, a VP poderá obter benefícios significativos, como a redução de riscos e incertezas, a economia de recursos, a melhoria da qualidade do produto ou serviço final, além de uma maior satisfação do usuário final. A prototipagem se revela uma abordagem determinante no processo de desenvolvimento, permitindo uma validação precoce das percepções, assim como uma abordagem iterativa e orientada ao potencial cliente.

5.4.1.5 Avaliação e testes com usuários

A avaliação e testes com usuários no contexto do Design Centrado no Usuário é uma prática fundamental para que os produtos e serviços desenvolvidos atendam as expectativas e necessidades dos usuários. Essas atividades fornecem informações de como os usuários interagem com o sistema e auxiliam a empresa a identificar possíveis dificuldades de usabilidade e coletam *feedbacks* para aprimorar a usabilidade do sistema.

Dessa forma, a Vamos parcelar deverá iniciar o processo identificando e definindo claramente os objetivos a serem alcançados com a avaliação e os testes. Com a identificação dos objetivos facilita o processo de resultados esperados com foco na resolução dos desafios. Portanto, nesses objetivos a empresa deve selecionar criteriosamente os participantes adequados, representando o público alvo e possuindo características relevantes para o estudo.

Durante a execução dos testes, os participantes deverão executar tarefas pertinentes aos casos de uso do sistema e os possíveis desafios encontrados durante a jornada do usuário. É necessário criar um ambiente propício para a execução das atividades para que os participantes sintam liberdade para expressar suas opiniões, emoções e dificuldades.

Os designers e avaliadores selecionados para executarem os testes devem sempre estar atentos para registrar todas as adversidades de usabilidade encontrados pelos usuários, as interações dos usuários e comentários relevantes. Além de dados

quantitativos, como tempo para conclusão das tarefas e taxa de sucesso deverão ser considerados para uma análise mais concisa.

Após a conclusão dos testes, os dados coletados serão analisados para identificar os obstáculos de usabilidade e priorizá-los com base no impacto causado para a experiência do usuário, classificando em escala de maior para menor gravidade, como pode ser ilustrado na figura 14. Os resultados obtidos serão comparados com os objetivos estabelecidos na fase inicial do processo, assim identificando melhorias a serem realizadas e determinando o sucesso do sistema.

Figura 14 – Gráfico de satisfação



Fonte: Autores.

5.5 Integração do Processo VP com as Diretrizes UCD e o *Framework Scrum-UCD*

A execução do processo de desenvolvimento de software, baseado nas metodologias ágeis, como o *Scrum*, juntamente com as práticas de Design Centrado no Usuário (UCD), requer uma abordagem colaborativa e integrada entre diferentes atores e equipes. Essa abordagem visa atender os objetivos do projeto, garantindo a valorização e o alto desempenho do software.

Inicialmente, a etapa de identificação do problema ou necessidade do cliente é conduzida pelo analista de negócios. Esse profissional realiza uma análise detalhada do contexto e das demandas dos usuários, utilizando técnicas de pesquisa e análise. Com base nessas informações, ele define os objetivos do projeto e avalia a viabilidade técnica e recursos disponíveis.

Paralelamente, a equipe de UX desempenha um papel fundamental na fase de identificação do problema ou necessidade envolvendo o projeto. Com a utilização de métodos de pesquisa, como entrevistas e observações, busca-se explorar o ambiente, as tarefas e os fluxos de trabalho dos usuários. Permitindo, assim, a identificação de pontos de impasses, requisitos e oportunidades ocasionando a melhoria do sistema.

Com a coleta das informações, a equipe de UX desenvolve personas que busquem representar perfis típicos de usuários. As personas são baseadas em características,

necessidades, comportamentos e propósitos dos usuários identificados durante a pesquisa. Além disso, pode ser produzido o *card sorting*, que consiste em uma técnica dos usuários em distribuir as informações relacionadas em grupos lógicos, auxiliando na definição da estrutura e organização do software.

Outra atividade importante realizada pela equipe de UX é a elaboração do mapa da jornada do usuário. O mapa contempla as interações e experiências dos usuários ao longo do ciclo de uso do software, destacando os pontos de contato e as emoções dos usuários em cada etapa. Assim, proporcionando *insights* valiosos sobre as ações, sentimentos e percepções do usuário, colaborando com o design de uma experiência do usuário mais eficaz..

Ao mesmo tempo, o analista de negócios continua seu trabalho, avaliando os requisitos do cliente e identificando as necessidades do projeto. Com a colaboração da equipe de desenvolvimento, há uma maior garantia de que os requisitos do projeto sejam devidamente considerados durante o processo de desenvolvimento.

Após as informações serem coletadas e a definição dos requisitos, o processo avança para a fase de planejamento. O *Product Owner*, com a colaboração da equipe de desenvolvimento, produz o *Product Backlog*, priorizando os itens se baseando nas necessidades do cliente e do usuário, além dos requisitos do projeto. Na reunião de *Sprint Planning*, a equipe de desenvolvimento seleciona as histórias de usuário relevantes para a *Sprint* atual e as divide em tarefas menores, estimando o esforço necessário para cada uma delas.

A equipe de desenvolvimento, juntamente com a equipe de UX, executa as atividades de desenvolvimento de acordo com as práticas do *Scrumban*. Durante o processo, a equipe de UX desempenha um papel fundamental na elaboração de protótipos de baixa ou alta fidelidade, permitindo a visualização das interfaces e funcionalidades do software. Os protótipos são validados com os usuários, por meio de testes de usabilidade, obtendo *feedback* sobre a usabilidade, funcionalidades e fluxos do sistema.

A fim de garantir a satisfação do cliente, a participação contínua dos usuários é importante durante todo o processo de desenvolvimento durante todo o processo de desenvolvimento. A equipe de UX realiza entrevistas, testes de usabilidade e outras atividades de pesquisa, buscando entender as necessidades e expectativas dos usuários. O *feedback* é incorporado nas iterações subsequentes do desenvolvimento, garantindo que o software atenda às demandas dos usuários e proporcione uma experiência satisfatória.

A equipe de desenvolvimento, em colaboração com o analista de qualidade, realiza testes unitários, de integração e de aceitação para garantir a qualidade do software.

Os problemas identificados durante os testes são corrigidos e refinamentos são feitos para melhorar a qualidade geral do software.

Ao final de cada *Sprint*, a equipe realiza duas importantes cerimônias: a Revisão da *Sprint*, onde os incrementos são apresentados ao cliente ou usuário para obtenção de *feedback*, e a Retrospectiva da *Sprint*, onde são identificados pontos positivos e negativos, discutindo melhorias para a próxima iteração. As práticas definidas são fundamentais para o aprendizado contínuo da equipe e para garantir o aprimoramento constante do processo de desenvolvimento ágil.

O ciclo de *Sprints* é repetido ao longo do projeto, com a equipe refinando continuamente o *Product Backlog* com base nos *feedbacks* do cliente/usuário, nas personas, no *card sorting*, no mapa da jornada do usuário e nos requisitos do negócio. No encerramento do projeto, uma revisão final é realizada com o usuário, documentando as lições aprendidas e as melhores práticas, e o produto final é entregue.

Em geral, a execução do processo de desenvolvimento de software, com base no *Scrum*, *Scrumban*, UCD e práticas de participação do usuário, requer a colaboração efetiva entre diferentes atores e equipes.

5.6 Considerações Finais do Capítulo

O capítulo apresentou a aplicação das diretrizes UCD no processo da *startup* VP, com o objetivo de aprimorar o desenvolvimento de software centrado no usuário. Foram identificadas as etapas do processo VP e, em seguida, realizada uma análise das diretrizes UCD com base na revisão bibliográfica, apontando a falta de implementação de várias dessas diretrizes na abordagem atual da empresa.

Com base na análise, foram propostas estratégias para a integração das diretrizes UCD no processo da *startup* VP. Foram destacadas a importância da participação ativa dos usuários, a compreensão do contexto de uso, a adoção de iteração e *feedback* contínuos, a prática de prototipagem e a realização de avaliações e testes com usuários.

Além disso, foi proposta uma integração do processo VP com as diretrizes UCD e o *framework Scrum-UCD*. Essa abordagem colaborativa e iterativa permitirá que a equipe de desenvolvimento atenda de forma mais efetiva às necessidades e expectativas dos usuários, proporcionando uma experiência de usuário mais satisfatória e melhorando a qualidade geral do produto ou serviço final.

Ao adotar essa abordagem centrada no usuário, a *startup* VP estará em uma posição vantajosa para desenvolver produtos e serviços que se destaquem no mercado, conquistem a fidelidade dos clientes e alcancem vantagens competitivas. Essa integração

entre as práticas ágeis e as diretrizes UCD fortalecerá a estratégia de desenvolvimento da empresa, impulsionando o sucesso de seus projetos e a satisfação de seus clientes.

6

Análise dos Dados

6.1 Considerações Iniciais do Capítulo

Este capítulo tem como objetivo examinar a análise de dados no contexto da integração do processo *Scrumban* da Vamos Parcelar com métodos UCD (User-Centered Design). A análise de dados fornece informações relevantes para aprimorar a eficácia do processo, identificar áreas de melhoria e promover a entrega de produtos de maior qualidade.

6.1.1 Análise das limitações do processo da Startup VP

A compreensão das limitações é fundamental para a identificação das oportunidades de melhoria e considerar abordagens complementares, como o User-Centered Design (UCD). A análise contribui para uma visão crítica e informada sobre práticas ágeis de desenvolvimento de software, sendo relevante para profissionais e pesquisadores na área de Engenharia de Software.

Este estudo analisa as limitações do processo da *Startup VP* como método de desenvolvimento de software. O processo adotado pela VP apresenta alguns desafios descritos abaixo:

- a) Foco limitado na experiência do usuário: Uma das limitações reside na sua falta de ênfase explícita nas necessidades e expectativas dos usuários finais. Embora o processo utilize o *scrumban*, que é projetado para a entrega de incrementos funcionais de software, ele pode não abordar adequadamente elementos críticos da experiência do usuário, como usabilidade, design e satisfação geral.
- b) Ausência de *feedback* por parte dos usuários durante as iterações: Outro aspecto importante a ser considerado é a ausência de engajamento efetivo por parte dos usuários finais durante as iterações do processo. A inexistência do *feedback* pode privar a equipe de informações estratégicas para o direcionamento do desenvolvimento do serviço ou produto e garantir que as necessidades do usuários sejam atingidas de forma satisfatória.
- c) Comunicação indireta com os usuários: Embora o processo enfatize a comunicação eficaz entre a equipe, a falta de interação direta com os usuários finais pode gerar lacunas na compreensão de suas necessidades. Isso pode levar à

decisões de projeto inadequadas e uma desconexão entre o produto final e as expectativas dos usuários.

- d) Ausência de validação constante com os usuários: O Processo não inclui mecanismos para a inclusão adequada da validação contínua com os usuários finais. A ausência de testes de usabilidade, pesquisa de satisfação, impossibilitam a coleta de informações indispensáveis para a evolução ininterrupta do serviço ou produto.

Após a descrição das limitações, é imprescindível ressaltar a relevância da incorporação de abordagens complementares ao processo da Vamos Parcelar para garantir produtos e serviços que atendam as expectativas e demandas de seus usuários finais. Sendo assim, é notável a identificação de oportunidades de melhorias a fim de assegurar que os produtos e serviços satisfaçam os anseios dos seus clientes finais.

6.1.2 Identificação de oportunidades de Melhorias

A identificação de oportunidades de melhorias consiste na representação dos dados da abordagem *UCD* e o *framework scrum-UCD* realizados durante a pesquisa bibliográfica e a integração do processo da Vamos parcelar com algumas diretrizes do *UCD*. Assim, pode-se expressar algumas das propostas para melhorias do processo de desenvolvimento de software da Vamos Parcelar:

- Compreensão aprofundada das demandas dos usuários: A aplicação de diretrizes de *UCD* permite uma melhor compreensão dos requisitos, projeções e anseios dos usuários. Demonstrando em um aperfeiçoamento no desenvolvimento de software, visando atender de forma mais assertiva às demandas práticas dos usuários finais.
- Maior colaboração dos usuários: A integração de práticas de *UCD* incentiva um envolvimento mais ativo dos usuários durante todo o processo de desenvolvimento. Os usuários podem fornecer *feedbacks* contínuos, participar de testes e validações, contribuir com ideias e auxiliar na definição de prioridades. Isso resulta em um produto final alinhado com as necessidades do público-alvo.
- Redução de retrabalho e desperdício: Ao incorporar práticas de *UCD* no processo, os desafios de usabilidade e as lacunas nas expectativas dos usuários são identificados e resolvidos com antecedência. Isso reduz a necessidade de retrabalho resultando em um processo de desenvolvimento mais eficiente.
- Melhoria da experiência do usuário: A abordagem centrada no usuário do *UCD* coloca a experiência do usuário em destaque. Isso leva a uma interface mais

amigável, intuitiva e agradável, resultando em uma melhor experiência geral para os usuários. Contribuindo para maior satisfação e retenção de usuários.

- **Aumento da eficiência e produtividade:** Ao entender melhor as necessidades dos usuários e envolvê-los no processo de desenvolvimento, as equipes de projeto podem tomar decisões concisas e direcionadas. Contribuindo com um fluxo de atividades otimizado, com menos retrabalho, menos iterações desnecessárias e maior produtividade geral.
- **Maior competitividade do produto:** Ao utilizar uma abordagem centrada no usuário, o produto ou serviço tende a ser competitivo no mercado. Um produto que atende de forma eficaz às necessidades dos usuários e proporciona uma excelente experiência do usuário tem maiores chances de se destacar e atrair mais usuários.

De forma breve, a integração de uma estratégia centrada no usuário, em um processo de desenvolvimento de software, pode resultar em evoluções consideráveis. As melhorias apresentadas possuem o potencial de proporcionar benefícios em todas as etapas do desenvolvimento. Ao considerar a integração de práticas de UCD, é possível alcançar resultados positivos no desenvolvimento de software, buscando sempre atender e superar as expectativas dos usuários.

7

Conclusão

Este estudo teve como objetivo analisar e propor a integração das Diretrizes de Design Centrado no Usuário (UCD) no processo de desenvolvimento de software da startup Vamos Parcelar (VP), por meio do framework Scrum-UCD. Durante a pesquisa, ficou evidente que a VP ainda não adota integralmente práticas centradas no usuário, o que representa uma oportunidade de melhoria para a organização, como foi demonstrado no Capítulo 4.

Assim, foram apresentadas as principais etapas do processo de pesquisa, uma revisão bibliográfica no Capítulo 2 e uma análise do contexto da startup VP no Capítulo 4. Com base nesse embasamento, foram selecionadas algumas Diretrizes de UCD, abrangendo aspectos essenciais, como a participação dos usuários, a compreensão do contexto de uso, a iteração contínua, a prototipagem, a avaliação e os testes com usuários.

A análise do processo da VP, detalhada na Seção 5.2, em relação às diretrizes, revelou que a empresa ainda não incorpora completamente essas práticas, o que representa uma oportunidade de melhoria para a organização. A proposta de integração feita na Seção 5.4 do processo VP com o *framework Scrum-UCD* demonstra que uma abordagem colaborativa e iterativa, centrada no usuário, pode fortalecer a competitividade da startup no mercado.

Realizou-se uma análise das limitações do processo da startup VP no contexto da integração do processo Scrumban com métodos de *User-Centered Design* (UCD), apresentada no capítulo 6. Essa análise revelou-se crucial para identificar oportunidades de melhoria e promover a entrega de produtos de maior qualidade. No processo da VP, foram identificados desafios, tais como o foco limitado na experiência do usuário, a falta de *feedback* dos usuários durante as iterações, a comunicação indireta com os usuários e a ausência de validação constante, conforme abordado na Seção 6.1.1.

Espera-se que este estudo contribua para uma maior compreensão da metodologia Scrum-UCD e sua relevância no contexto do desenvolvimento de software, especialmente em *startups*. Acredita-se que a implementação dessas diretrizes possa inspirar outras empresas a adotarem abordagens centradas no usuário e melhorarem a qualidade de seus produtos e serviços.

Por fim, ao se comprometer com a integração do *framework Scrum-UCD*, a VP estará avançando em direção a um desenvolvimento de software mais alinhado com as necessidades dos usuários, o que pode impactar positivamente suas experiências e contribuir para o sucesso da empresa. Que este estudo seja um passo importante em direção a uma jornada contínua de aperfeiçoamento e satisfação dos usuários.

7.0.1 Proposta para trabalhos futuros

Sugere-se alguns direcionamentos para trabalhos futuros que possam estender e aprofundar a pesquisa realizada neste estudo de caso:

Análise comparativa de metodologias ágeis e abordagens centradas no usuário: Realizar um estudo comparativo entre diferentes metodologias ágeis, como *Scrum*, *Kanban*, *Lean*, e outras abordagens centradas no usuário, como *Design Thinking* e *User-Centered Design*. Investigar como cada uma delas se complementa e pode ser integrada de forma sinérgica para melhorar a qualidade dos produtos desenvolvidos.

Avaliação do impacto financeiro das melhorias: Realizar uma análise detalhada dos benefícios financeiros resultantes da integração entre metodologias ágeis e abordagem centrada no usuário. Isso pode incluir a redução de custos de retrabalho, o aumento da satisfação do cliente levando a maior retenção de clientes, e o impacto positivo nas receitas devido a produtos mais bem adaptados ao mercado.

Adaptação das práticas para diferentes contextos: Investigar como as práticas propostas neste estudo de caso podem ser adaptadas para diferentes tipos de empresas, projetos ou setores de atuação. Cada contexto organizacional possui desafios específicos, portanto, é fundamental entender como as práticas podem ser ajustadas para melhor se adequar a cada realidade.

Exploração de métricas de usabilidade e satisfação do usuário: Pesquisar e identificar métricas específicas para medir a usabilidade e a satisfação do usuário ao longo do processo de desenvolvimento. A utilização de métricas adequadas permitirá que as equipes de desenvolvimento acompanhem de forma mais objetiva a evolução da qualidade do produto em relação às necessidades dos usuários.

Referências

- AHMAD, M. O. et al. Transition of software maintenance teams from scrum to kanban. *2016 49th Hawaii International Conference on System Sciences (HICSS)*, p. 5427–5436, 01 2016. Citado na página 20.
- ALQUDAH, M.; RAZALI, R. An empirical study of scrumban formation based on the selection of scrum and kanban practices. *International Journal on Advanced Science, Engineering and Information Technology*, v. 8, p. 2315–2318, 12 2018. Disponível em: <<http://www.insightsociety.org/ojaseit/index.php/ijaseit/article/view/6566>>. Citado 2 vezes nas páginas 21 e 22.
- ANDERSON, D. J. *Kanban : successful evolutionary change for your technology business*. [S.l.]: Blue Hole Press, 2010. Citado na página 13.
- ARGUMANIS, D.; MOQUILLAZA, A.; PAZ, F. Challenges in integrating scrum and the user-centered design framework: A systematic review. In: AGREDO-DELGADO, V.; RUIZ, P. H.; VILLALBA-CONDORI, K. O. (Ed.). *Human-Computer Interaction*. Cham: Springer International Publishing, 2020. p. 52–62. ISBN 978-3-030-66919-5. Citado 2 vezes nas páginas 24 e 25.
- ARGUMANIS, D.; MOQUILLAZA, A.; PAZ, F. A framework based on ucd and scrum for the software development process. *Design, User Experience, and Usability: UX Research and Design - 10th International Conference, DUXU 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Proceedings*, p. 15–33, 2021. Disponível em: <<https://cris.usil.edu.pe/es/publications/a-framework-based-on-ucd-and-scrum-for-the-software-development-p>>. Citado 9 vezes nas páginas 14, 27, 28, 29, 30, 31, 32, 33 e 34.
- BELLINI, E. et al. Pair designing as practice for enforcing and diffusing design knowledge. *Journal of Software Maintenance and Evolution: Research and Practice*, v. 17, p. 401–423, 2005. Citado na página 26.
- BERNARDO, J. H. J. A.; LIMA, J. J. D. J.; CUNHA, J. A. O. G. *Atribuições e Competências Individuais do Scrum Master: Um Estudo Exploratório*. SBC, 2016. 17-20 p. Disponível em: <<https://sol.sbc.org.br/index.php/sbsi/article/view/5996>>. Citado na página 18.
- BRERETON, P. et al. Using a protocol template for case study planning. 06 2008. Citado 3 vezes nas páginas 15, 35 e 36.
- 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*, v. 19, p. 557–573, 2012. Citado na página 18.
- CORONA, E.; EROS, F.; DIEE, P. *A Review of Lean-Kanban Approaches in the Software Development*. 2013. 3 p. Disponível em: <<http://www.wseas.us/journal/pdf/information/2013/5709-110.pdf>>. Citado na página 20.
- FELKER, C.; SLAMOVA, R.; DAVIS, J. Integrating ux with scrum in an undergraduate software development project. 02 2012. Citado na página 25.

- FOUNDATION, I. D. *What is User Centered Design?* UX courses, 2019. Disponível em: <<https://www.interaction-design.org/literature/topics/user-centered-design>>. Citado na página 27.
- GETTO, G. The story/test/story method: A combined approach to usability testing and contextual inquiry. *Computers and Composition*, v. 55, p. 102548, 03 2020. Citado na página 27.
- HUSSAIN, Z.; SLANY, W.; HOLZINGER, A. Investigating agile user-centered design in practice: A grounded theory perspective. In: . [S.l.: s.n.], 2009. v. 5889, p. 279–289. ISBN 978-3-642-10307-0. Citado na página 24.
- JOSHI, A.; SANDEEP, M. *Our Journey into Scrumban*. 2014. Disponível em: <<https://www.arkgroup.com/thought-leadership/our-journey-into-scrumban/>>. Citado na página 21.
- KNIBERG, H.; SKARIN, M. *Kanban and Scrum - Making the Most of Both*. [S.l.]: Lulu.com, 2010. Citado na página 13.
- KRAHMER, E.; UMMELN, N. Thinking about thinking aloud: A comparison of two verbal protocols for usability testing. *IEEE Transactions on Professional Communication*, v. 47, p. 105–117, 06 2004. Citado na página 27.
- LADAS, C. *Scrumban - Essays on Kanban Systems for Lean Software Development*. [S.l.: s.n.], 2009. Citado na página 13.
- LOSADA, B. *Flexible Requirement Development through User Objectives in an Agile-UCD Hybrid Approach*. 2018. Disponível em: <<https://dl.acm.org/doi/10.1145/3233824.3233865>>. Citado na página 25.
- MAHNIČ, V.; HOVELJA, T. On using planning poker for estimating user stories. *Journal of Systems and Software*, v. 85, p. 2086–2095, 09 2012. Citado na página 43.
- NEDELTCHEVA, G. N.; SHOIKOVA, E. Coupling design thinking, user experience design and agile. *Proceedings of the International Conference on Big Data and Internet of Thing - BDIOT2017*, 2017. Citado na página 25.
- NIELSEN, J.; MOLICH, R. *HEURISTIC EVALUATION OF USER INTERFACES*. 1990. Disponível em: <<http://concretaux.com/wp-content/uploads/nielsenheuristicsCHI.pdf>>. Citado na página 26.
- NORMAN, D. A. *The design of everyday things*. [S.l.]: Basic Books, 2013. Citado 2 vezes nas páginas 22 e 24.
- NORMAN, D. A.; DRAPER, S. W. *User centered system design : new perspectives on human-computer interaction*. [S.l.]: Lawrence Erlbaum, 1986. Citado na página 22.
- NURMULIANI, N.; ZOWGHI, D.; WILLIAMS, S. *Using card sorting technique to classify requirements change*. 2004. 240–248 p. Disponível em: <<https://ieeexplore.ieee.org/document/1335681>>. Citado na página 26.
- OHNO, T. *O sistema Toyota de produção : além da produção em larga escala*. [S.l.]: Bookman, 1997. Citado na página 20.

ROSENBAUM, M. S.; OTALORA, M. L.; RAMÍREZ, G. C. How to create a realistic customer journey map. *Business Horizons*, v. 60, p. 143–150, 01 2017. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0007681316300982>>. Citado na página 27.

SCHWABER, K.; SUTHERLAND, J. *Guia do Scrum™ Um guia definitivo para o Scrum: As regras do jogo*. 2013. 3-12 p. Disponível em: <<https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Citado 3 vezes nas páginas 13, 18 e 19.

SVERRISDOTTIR, H. S.; INGASON, H. T.; JONASSON, H. I. The role of the product owner in scrum-comparison between theory and practices. *Procedia - Social and Behavioral Sciences*, v. 119, p. 257–267, 03 2014. Citado na página 19.

APÊNDICE A – Carta de solicitação de acesso e citação



SOLICITAÇÃO DE ACESSO E CITAÇÃO

Brasília, 21 de novembro de 2022 Organização: Vamos Parcelar

A/c Responsável: Pedro Rosa

Por meio desta, solicitamos a empresa Vamos Parcelar acesso e citação dos dados do *Processo Scrumban de desenvolvimento da Startup Vamos Parcelar (VP)* para a realização do Trabalho de Conclusão de Curso dos estudantes, também funcionários desta empresa.

A presente pesquisa representa parte dos requisitos do trabalho de conclusão de curso de graduação de *Engenharia de Software* pela FGA-UnB (Faculdade do Gama – Universidade de Brasília).

Trata-se de um trabalho acadêmico cujo objetivo é a caracterização do *Processo Scrumban de desenvolvimento da Startup Vamos Parcelar (VP)* com intuito de, a partir das pesquisas nas bases científicas, propor diretrizes de *Design Centrado no Usuário* baseado no *framework* Scrum-UCD para o processo da VP.

Gostaríamos de ressaltar que todas as informações relacionadas e liberadas serão consideradas em comum acordo com a organização. Quando da conclusão deste trabalho, a organização receberá uma cópia do estudo.

Desde já, agradecemos a colaboração.

Cordialmente,

Lucas Alexandre e
Matheus de Cristo

CURSO:	ENGENHARIA DE SOFTWARE
DISCIPLINA:	TRABALHO DE CONCLUSÃO DE CURSO (TCC)
SEMESTRE:	REFERENTE ao 2º. de 2022
ESTUDANTES:	Lucas Alexandre Fernandes Martins e Matheus de Cristo Doreia Estanislau