



NATHÁLIA MARIA RAPUANO DE LYRA NOVAES LEONESSA

**STARTUP KAIZEN: UMA METODOLOGIA ÁGIL PARA DESENVOLVIMENTO
DE SOFTWARE EM STARTUPS**

Sorocaba
2016

UNIVERSIDADE FEDERAL DE SÃO CARLOS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO (PPGCCS)

NATHÁLIA MARIA RAPUANO DE LYRA NOVAES LEONESSA

**STARTUP KAIZEN: UMA METODOLOGIA ÁGIL PARA DESENVOLVIMENTO
DE SOFTWARE EM STARTUPS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCCS) da Universidade Federal de São Carlos, como parte dos requisitos exigidos para obtenção do título de mestre em Ciência da Computação área de concentração: Engenharia de Software e Startups.

Orientação: Prof. Dr. Alexandre Álvaro

Sorocaba
2016

Novaes Leonessa, Nathália

STARTUP KAIZEN: UMA METODOLOGIA ÁGIL PARA
DESENVOLVIMENTO DE SOFTWARE EM STARTUPS. / Nathália Novaes
Leonessa. -- 2016.
151 f. : 30 cm.

Dissertação (mestrado)-Universidade Federal de São Carlos, campus
Sorocaba, Sorocaba

Orientador: Alexandre Álvaro

Banca examinadora: Luciana Zaina, Claudia de Oliveira Melo

Bibliografia

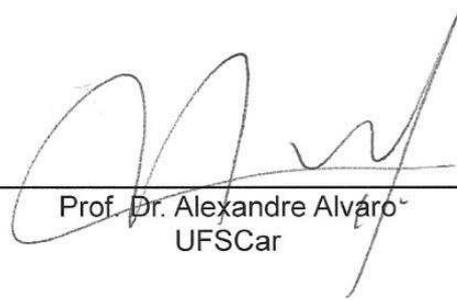
1. Engenharia de Software. 2. Startups. 3. Metodologias de
Desenvolvimento de Software. I. Orientador. II. Universidade Federal de São
Carlos. III. Título.

Ficha catalográfica elaborada pelo Programa de Geração Automática da Secretaria Geral de Informática (SIn). **DADOS
FORNECIDOS PELO(A) AUTOR(A)**



Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a defesa de dissertação de mestrado da candidata Nathália Maria Rapuano de Lyra Novaes Leonessa, realizada em 20/12/2016:



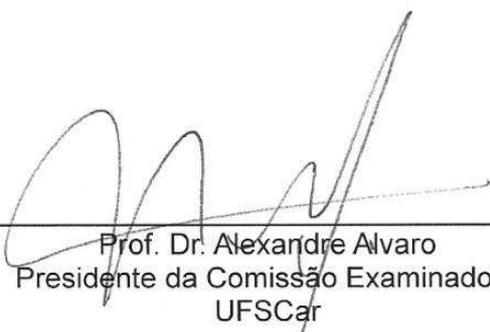
Prof. Dr. Alexandre Alvaro
UFSCar

Profa. Dra. Claudia de Oliveira Melo
UnB



Profa. Dra. Luciana Aparecida Martinez Zaina
UFSCar

Certifico que a sessão de defesa foi realizada com a participação à distância do membro Profa. Dra. Claudia de Oliveira Melo e, depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da comissão examinadora redigido no relatório de defesa da aluna Nathália Maria Rapuano de Lyra Novaes Leonessa.



Prof. Dr. Alexandre Alvaro
Presidente da Comissão Examinadora
UFSCar

DEDICATÓRIA

*À MINHA QUERIDA FAMÍLIA.
AO MEU AMADO MARIDO.
AOS MEUS QUERIDOS AMIGOS.*

AGRADECIMENTOS

Agradeço,

A Deus por sempre me guiar, me dando saúde e força para enfrentar e superar as dificuldades.

Ao meu marido, Raphael Leonessa, por estar sempre ao meu lado me incentivando todos os dias.

A minha família que constituem a base e os alicerces da minha vida, sempre servindo de inspiração e motivação todos os dias.

A nossa filha de quatro patas, Leka, por sempre estar presente e mesmo sem saber, ajudando sempre a alegrar os momentos de tensão e estresse.

Ao meu orientador Professor Dr. Alexandre Álvaro, pelo suporte ao longo de toda essa jornada, pelas suas correções, direcionamentos e incentivos.

A Professora Dra. Luciana Zaina por ter aceitado participar de duas bancas no decorrer do projeto, dando valiosas contribuições.

Aos meus colegas, Carlos Mello e Danielle Vieira, por incentivarem todos os dias a sempre utilizar em diferentes projetos, sempre buscando sua evolução e melhoria.

Aos alunos de mestrado do curso de Pós-graduação em Ciência da Computação, que participaram do projeto através do uso da metodologia.

A UFSCar pela estrutura e ambiente criativo e amigável que proporciona a todos seus alunos.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

RESUMO

As metodologias atualmente utilizadas por *startups* para descoberta de mercado e desenvolvimento de *software* focam no desenvolvimento mais ágil e rápido, visando a obtenção de aprendizado sobre o mercado de atuação. Muitas vezes, essas metodologias deixam de lado boas práticas de desenvolvimento de *software* para que ele seja realizado de forma mais dinâmica e rápida, com constante participação do usuário final.

Se, por um lado, a utilização de metodologias ágeis como *Scrum* e metodologias mais tradicionais como RUP, para o desenvolvimento de *software* pode resultar no desenvolvimento de soluções tecnológicas que não são utilizadas pelos usuários por falta de uma constante participação deles na construção do *software*, por outro lado, elas trazem inúmeros benefícios quando se trata de gerenciamento de projeto e qualidade de *software*. Em contrapartida, o oposto ocorre para as metodologias e ferramentas atualmente utilizadas por *startups*, tais como *Lean Startup*, *Customer Development*, *Design Thinking* e *Business Model Canvas*. Essas metodologias e ferramentas focam na descoberta e validação de mercado, sem a preocupação com a qualidade final do produto desenvolvido para o cliente, impactando-o diretamente. Esse impacto pode também acarretar um desperdício de tempo no desenvolvimento de um produto que ninguém conseguirá utilizar, devido à grande quantidade de falhas, ou até que nunca será finalizado, devido a problemas de gerenciamento.

A falta de preocupação com um bom gerenciamento de um projeto de desenvolvimento, e sua qualidade final, podem trazer tantos prejuízos quanto a falta de aproximação com o potencial cliente. Além disso, no ambiente de extrema incerteza, as variáveis de insegurança devem ser reduzidas ao máximo possível visando a obtenção de sucesso no desenvolvimento como todo, e não apenas em relação a modelos de negócios.

A utilização de boas práticas de engenharia de *software* pode possibilitar a obtenção de mais informações e dados técnicos sobre a solução desenvolvida, que impactam diretamente o usuário. Essas informações podem ser utilizadas como base na tomada de decisão, diminuindo assim, os riscos relacionadas a qualidade final do produto e gerenciamento de projeto, facilitando a utilização por seus potenciais clientes, além de um desenvolvimento mais enxuto.

Neste sentido, este trabalho propõe uma metodologia de desenvolvimento de *software* para *startups*, o *Startup Kaizen* (SK). O SK agrega boas práticas de metodologias como RUP e *Scrum* com metodologias com foco na validação e descoberta de mercado, como *Lean Startup*, *Customer Development*, *Design Thinking* e *Business Model Canvas*. Essa união de boas práticas de engenharia de *software* com a descoberta de mercado, tem como objetivo minimizar as variáveis de risco para a criação de uma nova empresa, sejam elas relacionadas a mercado ou gerenciamento e qualidade de *software*.

Após a criação do *Startup Kaizen*, foi realizado um estudo de caso com sua aplicação com alunos de pós-graduação em Ciências da Computação da Universidade Federal de São Carlos em Sorocaba. Essa aplicação permitiu a coleta de dados para análise dos resultados na forma de estudo de caso.

Palavras-chave: Engenharia de *Software*. *Startup*. *Lean Startup*. *Customer Development*. *Business Model Canvas*. *Design Thinking*. *Scrum*. Metodologia Ágil. RUP.

ABSTRACT

The methodologies currently used by startups for market discovery and software development focus on a more agile and fast development, aiming to obtain learning about the potential market. Often, these methodologies set aside good software development practices to make the process faster and more dynamic, with constant end-user participation.

If, on the one hand, the use of methodologies such as Scrum and RUP for software development can result in the development of technological solutions that are not used by users due to the lack of participation in software construction, on the other hand, they bring many benefits when it comes to project management and software quality. In contrast, the opposite occurs to the methodologies and tools currently used by startups, such as Lean Startup, Customer Development, Thinking Design and Business Model Canvas. These methodologies and tools focus on the discovery and validation of the market, without concern for the final quality of the product developed for the client, impacting directly. This impact can also be a waste of time in developing a product that no one will be able to use because of the large number of failures or even it never be finalized due to management problems.

The lack of concern for good management of a development project, and its final quality, can do as much damage as the lack of approximation with the potential client. Moreover, in an environment of extreme uncertainty, unknown variables should be reduced in order to achieve overall development success, not just business models.

The use of good Software Engineering practices may allow obtaining more information and technical data about the solution developed, which directly impact the user. This information can be used as a basis for decision making, thereby reducing risks related to final product quality and project management, making it easier for potential customers to use, and leaner development.

In this sense, this work proposes a software development methodology for startups, Startup Kaizen (SK). SK integrates good practices of methodologies such as RUP and Scrum with methodologies focused on validation and market discovery, such as Lean Startup, Customer Development, Thinking Design and Business Model Canvas. This union of good software engineering practices with market discovery aims to minimize the risk variables for the creation of a new company, whether related to market or management and software quality.

After the creation of Startup Kaizen, a case study was performed with its application with graduate students in Computer Science at the Federal University of São Carlos in Sorocaba. This application allowed the data collection to analyze the results in the form of a case study.

Keywords: Software Engineering. Startup. Lean Startup. Customer Development. Business Model Canvas. Design Thinking. Scrum. Agile Methodology. Scrum. RUP.

LISTA DE FIGURAS

Figura 1.1 - Etapas da Metodologia de aplicação do estudo	6
Figura 2.1 - Fases e Disciplinas da Metodologia RUP.	8
Figura 2.2 - Modelo de processo do Scrum.	10
Figura 2.3 - Ciclo de processo da metodologia Customer Development.....	11
Figura 2.4 - Ciclo construir-medir-aprender do Lean Startup.....	12
Figura 2.5 - Business Model Canvas	14
Figura 2.6 - Etapas do Design Thinking	15
Figura 2.7 - Exemplo de persona utilizado pelo Design Sprint.....	16
Figura 2.8 - Jornada do usuário.	17
Figura 2.9 - Matriz de análise de ideias.	18
Figura 2.10 - Comparativo metodologias por fase	20
Figura 3.1 - Quantidade de artigos publicados por ano	21
Figura 3.2 - ESSSDM Fase 1: Geração de ideias	23
Figura 3.3 - ESSSDM Fase 2: Validação.....	24
Figura 3.4 - Metodologia Lean UX	26
Figura 4.1 - Ciclo da Metodologia Startup Kaizen.....	31
Figura 4.2 - SW Canvas	33
Figura 4.3 - Exemplo SW Canvas preenchido	33
Figura 4.4 - SW Canvas hipóteses confirmadas e excluídas.....	34
Figura 4.5 - SW Canvas atualizado após validação.....	35
Figura 4.6 - Quadro de personas adaptado do quadro de personas do <i>Design Sprint</i>	36
Figura 4.7 - 8 Momentos-Chave.....	36
Figura 4.8 - Padrão BDD para criação de histórias	37
Figura 4.9 - Matriz Impacto no Cliente X Dificuldade de Implementação adaptado do <i>Design Sprint</i>	37
Figura 4.10 - Guia de Entrevista utilizada na Validação - Parte I adaptado de [BARBOSA; SILVA, 2010].....	38
Figura 4.11 - Guia de Entrevista utilizada na Validação - Parte II adaptado de [BARBOSA; SILVA, 2010].....	38
Figura 4.12 - Guia para planejamento da Validação - Parte I adaptado do RUP.....	39
Figura 4.13 - Guia para planejamento da Validação - Parte II adaptado do RUP.....	40
Figura 4.14 - Matriz de Relacionamento.....	41
Figura 4.15 - Métricas mapeadas na planilha aba Principal.....	43
Figura 4.16 - Planilha aba Principal - parte I	43
Figura 4.17 - Planilha aba Principal - parte II	44
Figura 5.1 - GQM criado para análise dos resultados do estudo de caso	47
Figura 5.2 - Atuação no mercado de TI	49
Figura 5.3 - Resultado das respostas dadas pelos participantes sobre as metodologias estudadas	50
Figura 5.4 - Resultado das respostas dadas pelos participantes sobre o Design Thinking	50
Figura 5.5 - Planilha de Testes	51
Figura 5.6 - Artefatos gerados pela equipe <i>Dover</i> durante a fase de Requisitos.....	54
Figura 5.7 - Exemplo de formato de documento utilizado pela equipe <i>Dover</i>	55
Figura 5.8 - Fase de Requisitos	56
Figura 5.9 – Modelo de documentação utilizada pelo grupo <i>Caregiver</i> na fase de Requisitos.....	56
Figura 5.10 - Modelo de documentação utilizada pelo grupo <i>Dover</i> na fase de Requisitos....	57
Figura 5.11 - Aba Principal criada pela equipe <i>Dover</i>	57

Figura 5.12 - Planilha Startup Kaizen - Aba Principal	58
Figura 5.13 - Modelo utilizado pelo grupo <i>Caregiver</i> para definição dos artefatos na fase de Requisitos	58
Figura 5.14 - Modelo utilizado pela equipe <i>Caregiver</i> para classificação das histórias.....	59
Figura 5.15 - Mapa de Empatia	60
Figura 5.16 - Exemplo de Persona criada pelo grupo <i>Caregiver</i>	61
Figura 5.17 - Exemplo de Persona criada pelo grupo <i>Dover</i>	62
Figura 5.18 - Guia para planejamento da validação do grupo <i>Caregiver</i>	62
Figura 5.19 - Tabela de validação grupo <i>Caregiver</i>	63
Figura 5.20 - Guia para planejamento da validação do grupo <i>Dover</i>	63
Figura 5.21 - Tabela de validação grupo <i>Dover</i>	64
Figura 5.22 - Escala de validação definida pelo grupo <i>Caregiver</i>	64
Figura 5.23 - Escala de validação definida pelo grupo <i>Dover</i>	64
Figura 5.24 - Respostas recebidas durante Validação pela equipe <i>Dover</i>	65
Figura 5.25 - Resultados obtidos durante a Validação pelo grupo <i>Caregiver</i>	65
Figura 5.26 - <i>Backlog</i> de produto gerado pela equipe <i>Caregiver</i>	66
Figura 5.27 - <i>Sprint 0</i> criada pela equipe <i>Caregiver</i>	66
Figura 5.28 - Matriz de Relacionamento gerada pela equipe <i>Dover</i>	67
Figura 5.29 - Aba Principal: Mapeamento das histórias e falhas encontradas pelo grupo <i>Dover</i> - Parte I	68
Figura 5.30 - Aba Principal: Mapeamento das histórias e falhas encontradas pelo grupo <i>Dover</i> - Parte II.....	68
Figura 5.31 - Tela “Produto Retirado” do grupo <i>Dover</i>	69
Figura 5.32 - Tela “Atribuir Medicação” do grupo <i>Caregiver</i> (MVP1).....	69
Figura 5.33 - Planilha de testes de <i>software</i>	70
Figura 5.34 - Planilha de testes criada pelo grupo <i>Factoring Sys</i>	70
Figura 5.35 - Resultado dos testes do sistema <i>Factoring Sys</i>	71
Figura 5.36: Exemplo de perguntas realizadas para os usuários	72
Figura 5.37 - Resultados obtidos com as perguntas realizadas	72
Figura 5.38 - Exemplo de requisitos criados pela equipe Impressão Distribuída	74
Figura 5.39 - Exemplo de requisitos criados pela equipe <i>Dover</i>	74
Figura 5.40 - Exemplo de requisitos criados pela equipe <i>Caregiver</i>	75
Figura 5.41 - Exemplo de requisitos criados pela equipe <i>Factoring Sys</i>	75
Figura 6.1 - Roadmap criado para estruturação da ideia na fase de Requisitos	79

LISTA DE TABELAS

Tabela 1.1 - Comparativo metodologias	4
Tabela 3.1 - Quantidade de artigos encontrados por biblioteca	21
Tabela 3.2 - Pontos Fortes e Pontos Fracos dos trabalhos relacionados.....	27
Tabela 4.1 - Artefatos de entrada e saída da fase de Requisitos.....	32
Tabela 4.2 - Artefatos de entrada e de saída da fase de Validação.....	40
Tabela 4.3 - Artefatos de entrada e saída da fase de Projeto.....	42
Tabela 4.4 - Artefatos de entrada e de saída da fase de Implementação	42
Tabela 5.1 - Entregas da metodologia.....	47
Tabela 5.2 - Grupos responsáveis pelos testes	48
Tabela 5.3 - Definição dos Grupos que utilizaram a Metodologia.....	49
Tabela 5.4 - Quantidade de Histórias criadas e selecionadas para o MVP	59
Tabela 5.5 - Quantidade de artefatos gerados pelos grupos na fase de Requisitos	62
Tabela 5.6 - Quantidade de Histórias criadas, selecionadas e validadas	64
Tabela 5.7 - Quantidade de falhas encontradas por grupo em cada versão de <i>software</i> liberada	67
Tabela 5.8 - Resultado obtido com os testes realizados de cada grupo	71
Tabela 5.9 - Resultado obtido com as notas dadas por usuários	73

LISTA DE ABREVIATURAS, SIGLAS E SÍMBOLOS

AC	Atividades Chave
ACM	Association for Computing Machinery
BDD	Behavior Driven Development
BMC	Business Model Canvas
CC	Canais de Clientes
CD	Customer Development
DT	Design Thinking
EC	Estrutura de Custos
ESSSDM	Early Stage Software Startup Development Model
FR	Fontes de Renda
GQM	Goal Questions Metrics
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IQ	Índice de Qualidade do Software
LS	Lean Startup
MVP	Minimum Viable Product
PC	Parcerias Chave
PO	Product Owner
PV	Proposta de Valor
RC	Relações com Clientes
RCh	Recursos Chave
RUP	Rational Unified Process
SM	Segmentos de Mercado
SK	Startup Kaizen

SW	Software
TAM	Technology Acceptance Model
TDD	Test Driven Development
TI	Tecnologia da Informação
UPO	Usability Product Owner
UX	User Experience
XP	Extreme Programming

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO E PROBLEMA	2
1.2	OBJETIVOS	4
1.3	METODOLOGIA	5
1.4	ORGANIZAÇÃO DO TRABALHO	6
2	FUNDAMENTOS E TÉCNICAS	7
2.1	RUP	7
2.2	SCRUM	8
2.3	LEAN STARTUP E CUSTOMER DEVELOPMENT	10
2.4	BUSINESS MODEL CANVAS	13
2.5	DESIGN THINKING	14
2.5.1	Design Sprint	18
2.6	COMPARATIVO METODOLOGIAS	19
2.7	CONCLUSÃO	20
3	TRABALHOS RELACIONADOS	21
3.1	CONCLUSÃO	29
4	METODOLOGIA STARTUP KAIZEN	30
4.1	FASE DE REQUISITOS	32
4.2	FASE DE VALIDAÇÃO	38
4.3	FASE DE PROJETO	41
4.4	FASE DE IMPLEMENTAÇÃO	42
4.5	CONCLUSÃO	44
5	VALIDAÇÃO DA METODOLOGIA: UM ESTUDO DE CASO	46
5.1	DEFINIÇÃO DOS OBJETIVOS	46
5.1.1	Objetivo Global	46
5.1.2	Objetivo do Estudo	46
5.1.3	Questões	46
5.2	PLANEJAMENTO	47
5.2.1	Descrição da Instrumentação	47
5.2.2	Seleção do Contexto	48
5.2.3	Seleção dos Indivíduos	48
5.2.4	Procedimento de Análise	51
5.2.5	Validade do Procedimento	51
5.3	OPERAÇÃO	52
5.3.1	Aplicação do Estudo Experimental	52
5.3.2	Acompanhamento e Entregas	52
5.4	RESULTADO DO ESTUDO	53
5.4.1	Artefatos Gerados pela Aplicação da Metodologia	53
5.4.2	Resultado dos Testes Realizados	70
5.4.3	Resultado da Avaliação do Usuário	71
5.5	ANÁLISE E INTERPRETAÇÃO DOS RESULTADOS	73
5.5.1	Análise Qualitativa	73
5.5.2	Conclusão	77

6 CONCLUSÃO E TRABALHOS FUTUROS.....	78
REFERÊNCIAS	81
APÊNDICES	85
ANEXOS	128

1 INTRODUÇÃO

Novas empresas de tecnologia estão surgindo no mercado denominando-se como *startups*. As *startups* de tecnologia estão crescendo cada vez mais, trazendo soluções tecnológicas inovadoras para o mercado. Seu principal produto consiste numa solução tecnológica, podendo ser *software*, *hardware* ou ambos, que visa a resolução de um problema através da sua utilização [BLANK, 2013].

As *startups* são caracterizadas como instituições que criarão novos produtos e serviços sob condições de extrema incerteza [RIES, 2011]. Sua origem parte de uma ideia e projeto que não são possíveis de afirmar se darão certo [MOREIRA, 2010]. As *startups* são compostas por um grupo de pessoas que trabalham em prol de uma ideia e que atuam em diferentes setores de uma empresa ao mesmo tempo, como por exemplo, desenvolvimento de *software*, vendas, financeiro e negócios [BLANK, 2013].

São vários os fatores que determinam o sucesso de uma *startup*, dentre eles é possível destacar o mercado no qual ela atuará [RIES, 2011]. Metodologias como *Customer Development* (CD), *Lean Startup* (LS) e *Design Thinking* (DT) foram desenvolvidas para auxiliar os empreendedores na descoberta de potenciais clientes [RIES, 2011]. Essas metodologias têm como objetivo o desenvolvimento de novas tecnologias, com foco no desenvolvimento de *software*, diminuindo o desperdício de tempo e dinheiro. Essa diminuição se dá através de entregas rápidas, aumentando a velocidade na obtenção de *feedback*, e, conseqüentemente, aprendizado sobre seus clientes [BLANK, 2013].

As metodologias CD, LS e DT são conhecidas por terem seu modelo de desenvolvimento baseado no iterativo e incremental, no qual os clientes possuem uma maior participação durante o processo de desenvolvimento do *software* [JINZENJI, 2013]. Em Ries (2011), é feito um comparativo da evolução das metodologias de desenvolvimento de *software*. O autor destaca a diferença entre o desenvolvimento baseado no modelo em cascata, e ao desenvolvimento iterativo e incremental defendido pelas metodologias ágeis, e as metodologias utilizadas por *startups*.

No desenvolvimento em cascata, o usuário fica afastado do processo de construção do *software*. Além disso, o escopo de desenvolvimento é definido no início do ciclo e mantido até o seu final, sem nenhuma interação com o usuário [DYCK; MAJCHRZAK, 2012]. Já, em metodologias ágeis, no *Lean Startup* e *Customer Development*, o escopo de desenvolvimento é definido a cada iteração no ciclo de desenvolvimento, fornecendo um incremento funcional ao final de cada ciclo [POPPENDIECK; CUSUMANO, 2012].

O desenvolvimento iterativo e incremental torna o usuário mais presente, promovendo uma maior interação entre potenciais clientes e time de desenvolvimento durante todas as fases do processo de desenvolvimento [DYCK; MAJCHRZAK, 2012]. Esta interação com possíveis consumidores auxilia na descoberta do mercado antes do esforço de produção em escala, evitando assim o desperdício de tempo na criação de algo que provavelmente não terá público para consumir [RIES, 2011].

A descoberta de mercado é um item fundamental no mundo das *startups* de tecnologia [BLANK, 2013]. Para auxiliá-las na descoberta e definição de um modelo de negócios que seja viável para uma solução proposta, elas utilizam o *Business Model Canvas* (BMC).

O BMC consiste numa ferramenta visual e dinâmica, na qual os membros das *startups* devem preencher com hipóteses de modelo de negócios que devem ser validadas [BLANK, 2013]. Seu principal objetivo é trazer uma reflexão de modelo de negócios através de uma ferramenta de fácil visualização.

Desta forma, o BMC tornou-se uma ferramenta fundamental para as *startups* de tecnologia entenderem se a solução que está em construção é financeiramente viável ou não [BLANK; DORF, 2012]. Sua utilização possibilita assim, uma validação do modelo de negócios. Essa validação faz com que o BMC tenha se tornado uma ferramenta fundamental, já que a solução precisa ser financeiramente possível para poder prosseguir na construção da solução [PROENÇA; NADALI; BORBINHA, 2015].

Além da descoberta de mercado, as *startups* de tecnologia em qualquer estágio, seja inicial ou em tração, também enfrentam a concorrência de produtos já existentes e do rápido desenvolvimento de produtos semelhantes, exigindo uma maior qualidade nos seus produtos / serviços de *software* [GHISI, 2016]. Apesar de não possuírem uma estrutura empresarial sólida, as *startups* de tecnologia, que possuem um *software* como parte da solução desenvolvida, também devem considerar fatores como complexidade no cumprimento dos prazos, atendimento aos requisitos do software identificados pelo cliente, estimativa dos custos e recursos, dentre outros [CROWNE, 2002].

O controle dos processos que envolvem a fabricação do *software*, desde a sua criação até a sua completa instalação no cliente, torna-se necessário. Esse controle visa evitar prejuízos causados por projetos de desenvolvimento que são iniciados e não são terminados, até orçamentos e prazos bem acima do que foi estipulado no início do projeto, além de elevar a qualidade dos produtos em desenvolvimento [LUIZ, 2005].

Segundo CROWNE [2002], além dos fatores citados anteriormente, o insucesso de uma *startup* pode ocorrer devido aos problemas técnicos relacionados a engenharia de *software* e a falta de experiência de seus fundadores. Portanto, este trabalho apresenta uma metodologia de desenvolvimento de *software* em *startups* que une aspectos de negócios e validação recomendada pelas metodologias *Lean Startup*, *Customer Development* e *Design Thinking* e boas práticas de engenharia de *software* existentes no *RUP* e *Scrum*, com o objetivo de medir o impacto gerado por essa união em relação a qualidade do *software* desenvolvido.

1.1 MOTIVAÇÃO E PROBLEMA

Com a crescente demanda na resolução de problemas através de soluções tecnológicas, surgem novas oportunidades de mercado para as empresas de tecnologia. Visando absorver essas demandas de forma rápida e ágil, surgiram as *startups* e metodologias que auxiliam no seu desenvolvimento: *Lean Startup*, *Customer Development* e *Design Thinking*.

Essas metodologias têm como princípio básico o desenvolvimento enxuto, através da redução de desperdício na criação de soluções que ninguém utilizará [RIES, 2011]. Seus métodos e atividades têm como principal objetivo a descoberta de mercado de forma rápida e ágil, antes de iniciar o desenvolvimento de uma solução sem conhecimento de mercado [BLANK, 2013].

Os principais pilares de uma metodologia de desenvolvimento de *software* são as etapas, atividades e papéis que descrevem como a solução deve ser desenvolvida. Seja seu desenvolvimento através de ciclos, como as *sprints* no *Scrum* (seção 2.2) ou as disciplinas do *RUP*. Além disso, Sommerville (2011) diz que para a criação de um processo de desenvolvimento de *software* são necessárias quatro atividades fundamentais: (i) requisitos, (ii) desenvolvimento, (iii) validação e (iv) evolução do *software* (seção 2).

Com o início da economia digital, surgiu uma crescente demanda por sistemas para resolução de problemas. Essa crescente demanda teve como consequência o aumento da

complexidade dos sistemas desenvolvidos devido a sua constante evolução, e também o aumento da velocidade com que essas soluções são desenvolvidas. Além disso, com a evolução da tecnologia, soluções que antes eram consideradas impossíveis, passaram a se tornar realidade [SOMMERVILLE, 2011].

A velocidade requerida no desenvolvimento de novas soluções, trouxe um foco maior na agilidade da construção de novas tecnologias e mercados. Desta forma, as metodologias atualmente utilizadas por *startups* possuem a agilidade no desenvolvimento como um fator fundamental [RIES, 2011].

Esta agilidade acaba trazendo um problema, já que muitas vezes, as boas práticas relacionadas a engenharia e qualidade do *software* são desconsideradas durante o desenvolvimento do produto [CROWNE, 2002]. O desenvolvimento do produto deixou de ser o foco principal nessas metodologias, e a descoberta de mercado e constante interação com os clientes, assumiram um papel fundamental no ciclo de desenvolvimento [BLANK, 2013].

Fazendo um comparativo entre as metodologias utilizadas por *startups* e metodologias de Engenharia de *Software*, as últimas focam nas fases, atividades e ferramentas necessárias para o desenvolvimento de um *software*. Já, as metodologias utilizadas por *startups* possuem como foco principal, a descoberta e validação de mercado [RIES, 2011].

Como exemplos de metodologias de Engenharia de *Software* pode-se citar o *Scrum* e *RUP*. Estas metodologias não possuem uma preocupação com o mercado onde o produto será inserido, ou seja, não possuem diretrizes de como realizar a descoberta de mercado.

Já, as metodologias utilizadas por *startups* para a descoberta e validação de mercado, ressaltam apenas a descoberta do mercado como o fator crucial para o desenvolvimento de uma empresa [CROWNE, 2002]. E, apesar das *startups* utilizarem alguns métodos ágeis, como o *Scrum* por exemplo, esses métodos são utilizados em momentos diferentes.

Startups em estágio inicial, de validação e descoberta de mercado focam no desenvolvimento do *software* apenas após a comprovação de um mercado existente para o seu uso [BLANK, 2013]. Seu principal foco consiste em desenvolver um *software* que terá aceitação para um mercado validado, independente de como ele será desenvolvido, usando ou não uma metodologia de desenvolvimento de *software*.

Como o foco principal das metodologias utilizadas por *startups* consiste no desenvolvimento de produtos funcionais, de forma rápida e com baixo custo, o produto desenvolvido, deve então auxiliar o time desenvolvedor a validar sua solução com potenciais clientes [RIES, 2011]. Já, em metodologias de desenvolvimento de *software*, auxiliam o desenvolvedor a obter um produto com maior qualidade e menor custo, através de boas práticas de Engenharia de *Software*. Essas boas práticas, englobam as atividades defendidas por Sommerville (2011) para a criação de um processo de desenvolvimento de *software*: (i) requisitos, (ii) desenvolvimento, (iii) validação e (iv) evolução do *software* (manutenção).

Portanto, analisando as metodologias de Engenharia de *Software* e as utilizadas por *startups*, é possível perceber que as últimas não possuem descrição de boas práticas que englobem as atividades definidas por Sommerville (2011) para a criação de um processo de desenvolvimento de *software*. No *Lean Startup*, *Customer Development* e *Design Thinking*, apesar de serem metodologias amplamente utilizadas por novas empresas de tecnologia, não possuem descrição de como realizar um gerenciamento de requisitos, quais são as boas práticas para a realização de um desenvolvimento com maior qualidade, bem como descrição do processo de validação e manutenção do *software*.

Na Tabela 1.1 é apresentado um comparativo entre as metodologias de Engenharia de *Software*: *RUP* e *Scrum*, e as metodologias utilizadas por *startups*: *Lean Startup*, *Customer Development*, *Design Thinking* e *Design Sprint*.

Tabela 1.1 - Comparativo metodologias

	Negócios	Eng. SW	Fases bem-definidas	Papeis bem-definidos	Participação do usuário	Capacidade de Adaptação	UX
RUP		X	X	X	Baixa		
<i>Scrum</i>		X		X	Média	X	
<i>Lean Startup</i>	X	X			Alta	X	
<i>Customer Development</i>	X				Alta	X	
<i>Design Thinking</i>			X		Alta	X	X
<i>Design Sprint</i>			X		Alta	X	X

Na Tabela 1.1 é possível perceber que as metodologias possuem focos diferentes. Enquanto RUP e *Scrum* possuem fases e papeis bem-definidos, as metodologias utilizadas por *startups* possuem um foco maior em negócios e descoberta de mercado. Já, as metodologias *Design Thinking* e *Design Sprint*, possuem um foco maior na experiência do usuário (*User Experience – UX*).

Analisando as metodologias de desenvolvimento de *startups*, é possível constatar a grande influência de quatro pilares fundamentais durante a sua execução. São eles: (i) modelo de processo iterativo e incremental (evolutivo), (ii) Produto Mínimo Viável (*Minimum Viable Product - MVP*), (iii) ciclo de aprendizado e (iv) constante participação dos clientes (seção 2.3).

Além disso, Crowne (2002) cita principais problemas de protudo que podem afetar diretamente o sucesso de uma *startup*. Dentre eles, a falta de atendimento aos prazos devido à falta de experiência dos fundadores; até o bloqueio da inserção de novos requisitos devido a grande quantidade de falhas críticas reportadas por clientes.

Desta forma, este trabalho propõe uma metodologia de desenvolvimento de *software* para *startups*, *Startup Kaizen*, fazendo a união de boas práticas de Engenharia de *Software* e descoberta e validação de mercado. Essa união visa realizar um desenvolvimento ordenado e com maior qualidade para *startups* em estágio inicial e de crescimento.

A seguinte pergunta será respondida ao final deste estudo: a inserção de boas práticas de Engenharia de *Software* pode favorecer uma *startup* durante a fase de validação?

1.2 OBJETIVOS

Objetivo geral

Propor uma metodologia de desenvolvimento de *software* que englobe boas práticas de metodologias utilizadas por *startups* para descoberta de mercado e boas práticas de engenharia de *software* verificando se essa união possibilita um desenvolvimento de *software* em *startups* mais estruturado e com maior qualidade durante a fase de busca de mercado.

Objetivos específicos

- Estudar os conceitos, métodos, técnicas e ferramentas defendidos em metodologias para *startups* tais como *Lean Startup*, *Customer Development*, *Design Thinking*, *Business Model Canvas*;

- Estudar os conceitos, métodos e técnicas da engenharia de *software* através do estudo das metodologias *RUP* e *Scrum*;
- Levantar as dificuldades apresentadas pelas *startups* devido à falta de uma metodologia de desenvolvimento com etapas e processos bem definidos;
- Estudar a relação entre *startups* e seus clientes devido a problemas de *software*;
- Propor uma metodologia com etapas bem definidas, ferramentas, artefatos e atividades com base nas metodologias estudadas;
- Aplicar a metodologia proposta em projeto piloto para análise de resultados;
- Validar seus resultados com base nas métricas definidas no estudo, através da aplicação da metodologia proposta em *startups*.

1.3 METODOLOGIA

Para realização deste trabalho foi realizado primeiramente um estudo bibliográfico em bibliotecas digitais: ACM, IEEE, *Google Scholar* e *Scopus* sobre metodologias criadas para auxiliar *startups*: *Customer Development*, *Lean Startup*, *Design Thinking* e a ferramenta *Business Model Canvas*. Essa pesquisa foi realizada através de uma busca *ad hoc* com a utilização de palavras chaves descritas na seção 3.

Juntamente com a pesquisa realizada nas bibliotecas digitais informadas anteriormente, foi feito um estudo através da leitura dos principais livros recomendados para *startups*: *Lean Startup* de Eric Ries (2011), *Four Steps to the Epiphany* de Steve Blank (2013) e *The Startup Owner's Manual* de Steve Blank e Bob Dorf (2012). Para levantar as principais dificuldades de uma *startup* durante o ciclo de desenvolvimento de uma nova solução tecnologia, foi realizado um acompanhamento e um estudo etnográfico¹ de *startups* brasileiras através da aproximação de comunidades e ecossistemas de *startups*.

Além disso foi feito um estudo sobre as ferramentas, metodologias ágeis, modelo CMMI, além de trabalhos relacionados focados na utilização de boas práticas de engenharia de *software* juntamente com as metodologias de *startups*.

Após o levantamento das metodologias e ferramentas de base para esse trabalho, foram realizadas propostas de boas práticas que descritas na metodologia final: *Startup Kaizen*. Essas propostas tiveram como base a análise das metodologias citadas: *Scrum*, *RUP*, *Lean Startup*, *Customer Development*, *Design Thinking* e *Business Model Canvas*, e o entendimento de pontos de intersecção das metodologias, visando a definição de pontos de encaixe para a metodologia proposta.

Finalmente, a metodologia foi aplicada dentro da UFSCar de Sorocaba em grupos formados durante a disciplina de Engenharia de *Software* da Pós-Graduação em Ciência da Computação (PPGCCS) para a realização do estudo de caso. Essa aplicação foi feita com alunos de mestrado que cursavam a disciplina, e tinham como objetivo final desenvolver um projeto de uma *startup*.

No primeiro dia de aula, os alunos foram encorajados a falar suas ideias de negócio para toda a turma. A partir daí os demais alunos escolheram com quais ideias gostariam de trabalhar, formando os 4 grupos de trabalho. Cada grupo ficou responsável por desenvolver apenas uma ideia.

Como resultado dessa formação de grupos, os alunos ficaram divididos em 4 grupos: *Caregiver*, *Impressão Distribuída*, *Factoring Sys* e *Dover*. Após a definição das equipes, dois

¹ O estudo etnográfico consiste na aplicação de técnicas para a obtenção do estudo de um determinado assunto através da vivência direta. Ele auxilia na compreensão em profundidade do comportamento do usuário. O pesquisador se insere no contexto real de estudo, com o objetivo de fazê-lo obter um melhor conhecimento sobre um grupo de usuários [BOTH; LOPES, 2009].

grupos foram sorteados para utilizar a metodologia. Sendo que um desses grupos também cursava a disciplina de Interface Humano-Computador (IHC), que atuou de forma integrada com a disciplina de Engenharia de *Software*, através da utilização de um trabalho em comum, porém com aulas separadas.

Após a definição das equipes que utilizariam a metodologia, foi realizado um *workshop* para demonstração das fases, papéis e artefatos do *Startup Kaizen*. Os grupos trabalhavam em casa, com entregas definidas feitas por e-mail e apresentação em sala de aula. O acompanhamento foi feito com os encontros em sala de aula e reuniões online.

Para a análise dos resultados, foram definidas perguntas de pesquisa visando a exploração dos dados obtidos com a aplicação da metodologia, através de uma análise qualitativa. A Figura 1.1 - Etapas da Metodologia de aplicação do estudo ilustra cada etapa do trabalho desenvolvido.

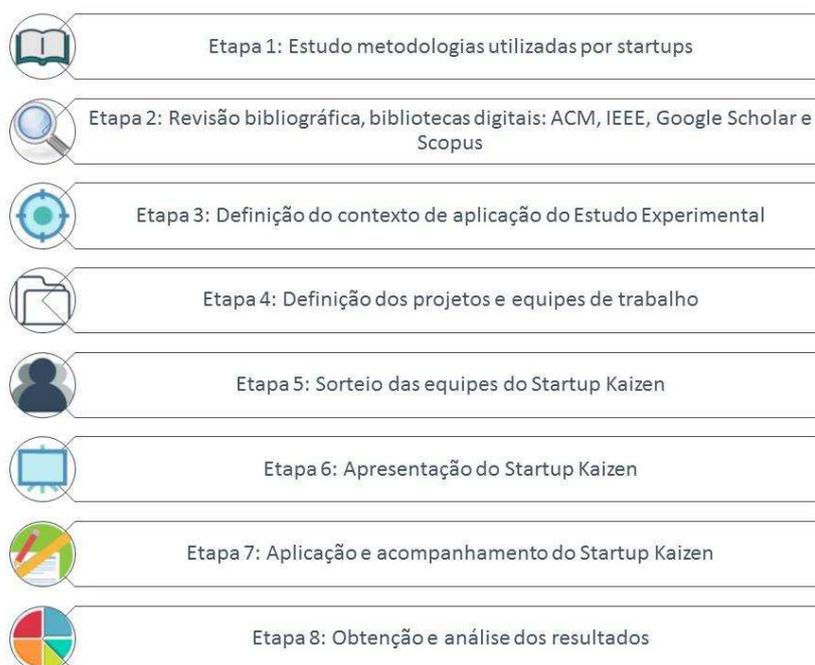


Figura 1.1 - Etapas da Metodologia de aplicação do estudo

1.4 ORGANIZAÇÃO DO TRABALHO

No Capítulo 2 são abordadas as metodologias estudadas para compor a metodologia proposta: *RUP*, *Scrum*, *Lean Startup*, *Customer Development* e *Design Thinking*. Na sequência, no Capítulo 3, são descritos os trabalhos relacionados detalhando suas contribuições nas metodologias estudadas.

No Capítulo 4 é apresentada a metodologia proposta, o *Startup Kaizen*, bem como todas as suas etapas, papéis e artefatos. O estudo de caso definido para validação da metodologia e seus resultados, são descritos no Capítulo 5. Por fim, no Capítulo 6 foi feita uma breve conclusão sobre o estudo, além de sugestões de melhorias e projetos futuros.

2 FUNDAMENTOS E TÉCNICAS

A Engenharia de *Software* aplica conceitos da disciplina de engenharia ao desenvolvimento de *software*; adicionando padrões de desenvolvimento de *software* com o objetivo de trazer um aumento na qualidade da entrega [DYCK; MAJCHRZAK, 2012]. Metodologias de desenvolvimento de *software* foram criadas com o intuito de estruturar essas técnicas e conceitos através da definição de boas práticas que levam a um desenvolvimento mais gerenciável e com maior qualidade [PRESSMAN, 2006].

Segundo Sommerville (2011), uma metodologia de desenvolvimento de *software* consiste numa abordagem sistemática da engenharia de *software* composta por processos e etapas bem-definidos. Um processo é uma sequência de atividades necessárias para o desenvolvimento. Em diferentes metodologias, existem quatro atividades fundamentais comuns. São elas: especificação, desenvolvimento, evolução:

- **Especificação:** é o conjunto de fatores que devem ser considerados para o desenvolvimento de determinada aplicação. Esses fatores, geralmente são definidos pelo time de engenheiros responsável, e pelos seus clientes;
- **Desenvolvimento:** é a etapa na qual o *software* é criado através de programação e criação de um *design*;
- **Validação:** consiste na verificação do *software*, visando a garantia de que ele atenda às necessidades dos clientes;
- **Evolução:** é a evolução do *software* através de sua modificação para refletir as mudanças das necessidades dos clientes e de mercado.

Um processo bem-definido através da sequência de atividades, ferramentas e papéis favorecem mecanismos preventivos. Esses mecanismos ajudam a antecipar a detecção de falhas diminuindo a quantidade de defeitos encontrados posteriormente e muitas vezes liberados para o cliente [PAULA, 2003].

RUP e *Scrum* são exemplos de metodologias de engenharia de *software* que foram desenvolvidas no mercado para auxiliar as empresas a controlarem melhor seu processo de desenvolvimento. Essas metodologias possuem etapas, atividades e papéis bem-definidos, abrangendo também as quatro atividades fundamentais: requisitos, desenvolvimento, validação e evolução.

O *Scrum* e o *RUP* têm como foco o desenvolvimento do *software*, através da definição de uma metodologia composta por processos com fases e etapas bem-definidas. Essas metodologias não fornecem métodos e ferramentas para a descoberta e validação de mercado.

Por outro lado, visando realizar a descoberta e validação de mercado, o *Customer Development*, *Lean Startup* e *Design Thinking* foram criados. Essas metodologias têm como foco principal o conhecimento e aprendizado sobre usuário e futuro cliente, favorecendo a descoberta do mercado de atuação através da criação e da adaptação de uma nova solução tecnológica [BLANK; DORF, 2012].

As próximas seções trazer uma visão geral das metodologias citadas acima.

2.1 RUP

O RUP (*Rational Unified Process*) consiste numa metodologia iterativa e incremental bem definida e estruturada criada pela IBM. A sua estrutura tem como base um ciclo de vida de um projeto através do acompanhamento com marcos essenciais e de pontos de decisão [CORPORATION, 2001]. Ele é dividido em 4 fases (Figura 1): Iniciação,

Elaboração, Construção e Transição e 9 disciplinas: Modelagem de Negócios, Requisitos, Análise e Design, Implementação, Teste, Implantação, Gerenciamento de Configuração e Mudança, Gerenciamento de Projeto e Ambiente.

Em cada fase, são inseridas uma ou mais disciplinas com volume de trabalho diferente [BOENTE; OLIVEIRA; ALVES, 2008]. Por exemplo, na fase de Iniciação, as disciplinas envolvidas são: Modelagem de Negócios, Requisitos, Análise e Design, Implementação, Teste, Gerenciamento de Configuração e Mudança, Gerenciamento de Projeto e Ambiente. Na fase citada como exemplo, a disciplina que possui maior volume de trabalho é a de Modelagem de Negócios. Na Figura 2.1 estão ilustradas as fases e disciplinas do RUP.

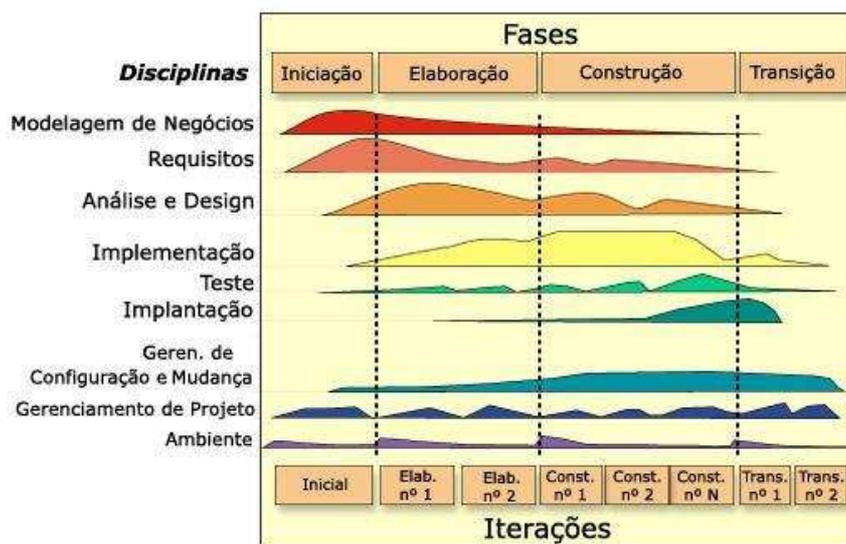


Figura 2.1 - Fases e Disciplinas da Metodologia RUP.

O RUP é uma metodologia bem definida e estruturada, todas as suas disciplinas possuem um *workflow* ou fluxo de trabalho. Cada *workflow* detalha uma sequência de tarefas encadeadas, artefatos de entrada e de saída, além dos papéis que devem ser executados por um determinado indivíduo ou grupo de indivíduos [BOENTE; OLIVEIRA; ALVES, 2008].

As metodologias de desenvolvimento podem trazer inúmeros benefícios para o desenvolvimento de *software* em qualquer estágio de amadurecimento de uma empresa. Nelas existem boas práticas que podem facilmente ser adaptadas para as metodologias de *startups* sem engessá-las, agregando valor à qualidade do produto final.

O RUP, por ser uma metodologia bastante completa e detalhada, possui pontos positivos e negativos. Um dos pontos positivos, é a gestão efetiva do desenvolvimento de um *software*, mapeando todas as suas fases e ferramentas necessárias para rodá-lo. Porém, por possuir todo esse detalhamento, essa metodologia acaba se tornando complexa demais para a execução de projetos com curta duração de desenvolvimento, e com equipes pequenas. Por isso, muitas vezes, as empresas que utilizam o RUP acabam utilizando somente algumas de suas partes, e não a metodologia completa.

2.2 SCRUM

As metodologias ágeis foram criadas com o objetivo de buscar a melhor maneira de tratar as mudanças que ocorrem durante todo o ciclo de desenvolvimento. Essas metodologias visam proporcionar uma maior interação com os clientes a cada ciclo de desenvolvimento

[ABRANTES; TRAVASSOS, 2007]. Elas possuem quatro valores bem-definidos: (i) indivíduos e interações são mais valorizados do que processos e ferramentas, (ii) o *software* em funcionamento, como uma prioridade em relação a uma documentação extensa, (iii) colaboração do cliente, ao invés de apenas uma negociação de contrato e (iv) respostas à mudanças mais do que seguir um plano [BECK et al., 2010]. Alguns exemplos de metodologias ágeis são: o *Scrum*, XP (*Extreme Programming*) e TDD (*Test Driven Development*).

O *Scrum* é uma metodologia que se adapta às mudanças fornecendo flexibilidade, capacidade de respostas rápidas e confiança ao desenvolvimento numa empresa. Nele, o desenvolvimento do *software* é iterativo e incremental, ou seja, ao final de cada iteração, um pedaço funcional do *software* deve ser desenvolvido. Essas iterações, são conhecidas como as *sprints* de desenvolvimento [MUNDRA; MISRA, 2013].

As *sprints* de desenvolvimento regem o *Scrum*, e sua execução ocorre durante um período de tempo pré-determinado. Por exemplo, numa empresa que possui uma *sprint* de uma semana, todas as atividades definidas durante uma reunião de planejamento (*planning meeting*) devem ser realizadas durante este período (1 semana) [SCHWABER; SUTHERLAND, 2014].

Além das *sprints* de desenvolvimento, outros pontos de destaque do *Scrum* são: *backlog* de produto e time de desenvolvimento. O *backlog* de produto é um o conjunto de funcionalidades do *software* que são desenvolvidos durante uma *sprint* [SCHWABER; SUTHERLAND, 2014]. A definição desse *backlog* é feita por uma espécie de gerente de projeto, que fica em contato direto com o cliente para identificação dessas funcionalidades. Este gerente, é conhecido como *Product Owner* (P.O.).

É o P.O. que identifica, juntamente com os clientes, quais as funcionalidades que devem ser desenvolvidas para o *software*, além de separar e priorizá-las em listas de desenvolvimento conhecidas como *backlog* de produto. Seu papel fundamental é fazer a conexão entre o time de desenvolvimento e o cliente final, e a definição e priorização dos *backlogs* de produto [WOLFF, 2012].

Após a definição dos *backlogs* de produto, o time *Scrum* se reúne com o P.O. para realizar a reunião de planejamento (*sprint planning meeting*). Nesta reunião, o P.O. descreve as funcionalidades prioritárias e a equipe as avalia tirando dúvidas quando necessário, visando obter um maior conhecimento sobre as histórias [SCHWABER, 2004].

O principal objetivo da reunião de planejamento, é quebrar o *backlog* de produto priorizado pelo P.O., em atividades que deverão ser executadas pelo time. Além disso, nela também é definido o objetivo da *sprint*, que consiste numa breve descrição do que se tentará alcançar ao final de sua execução [SCHWABER; SUTHERLAND, 2014].

As atividades derivadas da reunião de planejamento são chamadas de *sprint backlog*, e devem ser definidas visando o atingimento do objetivo da *sprint*. Cada membro da equipe deve dizer com quais atividades se comprometem a desenvolver durante a *sprint*, avaliando-as de acordo com sua complexidade de execução [WOLFF, 2012]. Desta forma, ao final da reunião de planejamento, cada membro sairá com uma lista de atividades que devem ser executadas durante a *sprint*.

A Figura 2 ilustra o fluxo de processo da metodologia *Scrum* com um exemplo de uma *sprint* com tamanho de 30 dias. Nela, é possível identificar o *backlog* de produto (*Product Backlog*) como um artefato de entrada para o início do processo, o *backlog* de *sprint* (*Sprint Backlog*) como resultado da reunião de planejamento, além da *sprint* de desenvolvimento resultando num incremento de *software* funcional.

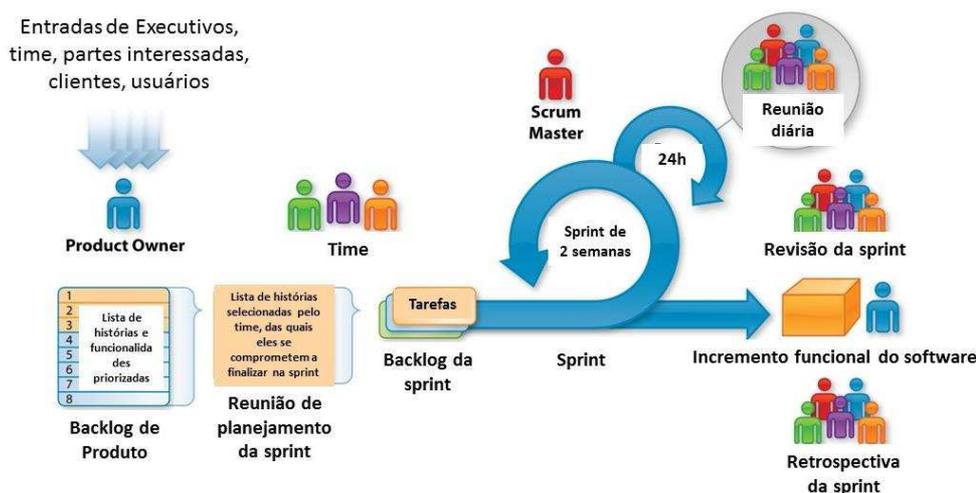


Figura 2.2 - Modelo de processo do Scrum.

Ao final de uma *sprint* de desenvolvimento, é feita uma reunião chamada de retrospectiva. Durante esta reunião, é feita uma revisão do que aconteceu durante o desenvolvimento destacando-se os pontos fortes e fracos. Além dessa identificação, um plano de ação é elaborado visando a melhoria dos pontos fracos, ou seja, o que não funcionou bem [SCHWABER; SUTHERLAND, 2014]. O principal resultado desta reunião é a identificação de pontos falhos, e um plano de ação para corrigi-los.

O *Scrum* é uma metodologia atrativa para as *startups* pois ele não se caracteriza apenas como o desenvolvimento rápido de aplicações. Seus maiores benefícios em relação a *startups* caracterizam-se pela capacidade de adaptação e principalmente na flexibilidade a mudanças no processo, produto e ambiente. Essa flexibilidade e capacidade de adaptação são possíveis devido ao desenvolvimento de iterações com entregas funcionais.

2.3 LEAN STARTUP E CUSTOMER DEVELOPMENT

Além das metodologias ágeis, atualmente existem metodologias novas que são constantemente utilizadas por *startups*, são elas: *Lean Startup* e *Customer Development*. Um dos seus principais benefícios consiste na minimização dos riscos e custos no desenvolvimento de uma *startup*.

Essas metodologias foram criadas com o objetivo de facilitar a descoberta de mercado e desenvolvimento de *software* de forma enxuta reduzindo o desperdício de trabalho em algo que pode não trazer resultados [RIES, 2011]. Segundo MAY [2012], o *Lean Startup* e *Customer Development* utilizam de boas práticas do desenvolvimento ágil e *User Experience (UX)* misturados com conceitos ligados a negócios. Essa mistura visa promover uma forma enxuta e rápida na busca de um conceito de produto que atenda a um mercado existente.

O *Customer Development* e *Lean Startup* atuam através da identificação de um problema para o qual são elaboradas hipóteses de soluções que serão validadas diretamente com possíveis clientes. Essa validação é feita através do desenvolvimento do *Minimum Viable Product (MVP)*, ou produto mínimo viável [RIES, 2011].

O MVP consiste no conjunto mínimo de funcionalidades que entregam valor, em forma de solução, para o mercado. Desta forma, é possível defini-lo como aquele produto que atende às necessidades de seus clientes de forma sucinta, adicionando somente funcionalidades que resolvam o problema abordado [BLANK, 2013].

As metodologias citadas se complementam, podendo ser executadas de maneira isolada ou em conjunto. No *Customer Development* o foco é relacionado a descoberta de mercado, no qual o autor, *Steve Blank*, estabelece formas para realizar uma busca de mercado sem ser necessário o desenvolvimento completo do produto final, através da utilização de um MVP [BLANK; DORF, 2012]. Blank (2013) estabelece métricas e ensina como descobrir potenciais clientes, validá-los, encontrar o mercado alvo. E, somente após essa descoberta e validação, que ele defende a criação uma empresa para atender o mercado validado.

Na Figura 2.3 é ilustrado o processo de *Customer Development*. Este processo é dividido em quatro etapas distintas, que iniciam através da etapa de Descoberta de Clientes, finalizando com a Construção da Empresa. As duas etapas iniciais, Descoberta e Validação de Clientes, devem ser executadas repetidamente até que seja encontrado um modelo de negócios repetível e escalável através da elaboração de hipóteses de mercado. Para essas etapas, faz-se o uso *Business Model Canvas* (descrito na seção 2.4) para estruturação e levantamento das hipóteses que serão validadas [BLANK, 2013].

Somente depois da descoberta de um modelo de negócios repetível e escalável, é possível expandir as vendas do produto e/ou serviço validado, passando para a etapa de Criação de Clientes. Ao final, com um mercado bem definido e estruturado, começa-se a Construção da Empresa, definindo novos através da criação de departamentos e contratação de funcionários para a empresa [BLANK, 2013].

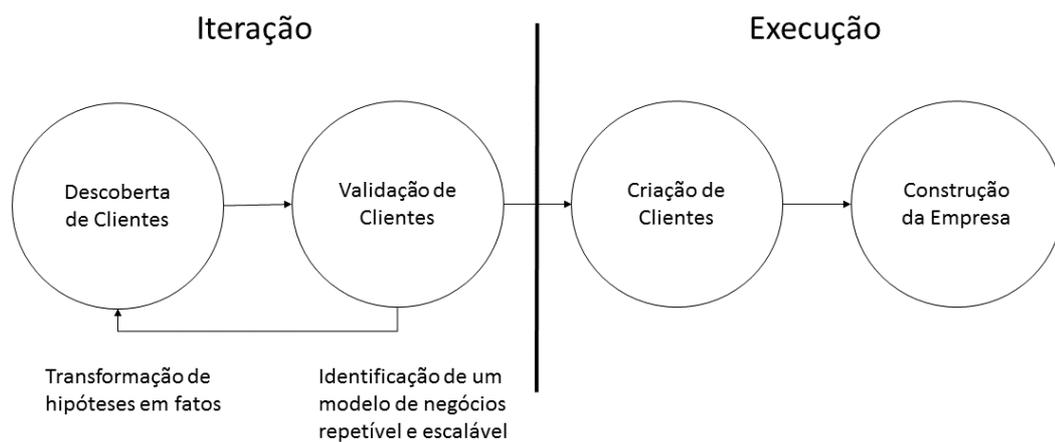


Figura 2.3 - Ciclo de processo da metodologia Customer Development.

O *Lean Startup*, descreve a aplicação do pensamento enxuto na inovação. Ele foi criado com base na manufatura enxuta e tem como objetivo evitar o desperdício gerado durante o desenvolvimento [BLANK; DORF, 2012]. No *Lean Startup*, a cada iteração uma versão funcional do *software* deve ser liberada para validação direta com o cliente. Essas iterações visam aumentar o entendimento sobre o mercado e sua necessidade inicial através de *feedbacks* constantes para a construção de um novo produto [POPPENDIECK; CUSUMANO, 2012].

Assim como o *Customer Development*, o *Lean Startup*, trabalha com a ideia da descoberta de um modelo de negócios repetível e escalável através da criação de produtos que os clientes irão pagar e utilizar [MAY, 2012]. O *Lean Startup*, ou *Startup Enxuta*, está baseada nos pilares de construção através de ciclos de iterações rápidas, na medição de dados para possibilitar o aprendizado, que é o mais importante pilar do *Lean Startup* [RIES, 2011].

Na engenharia de *software*, a metodologia *Lean Startup* é muito comparada a processos baseados no modelo de desenvolvimento em cascata. No último não há quase interação com os clientes e o produto é somente validado ao final, depois do *software*

concluído, já no primeiro, a cada iteração uma versão funcional do *software* deve ser liberada para validação direta com o cliente visando sempre aumentar o entendimento sobre o mercado e sua necessidade inicial [POPPENDIECK; CUSUMANO, 2012].

A Figura 2.4 ilustra o ciclo construir-medir-aprender do *Lean Startup* com três etapas distintas: Ideia, Produto e Dados. Essas três etapas são responsáveis por prover informações suficientes para realizar a construção, medição e aprendizado e em cada ciclo eles devem ser executados [RIES, 2011].

No *Lean Startup*, o MVP também é utilizado como base para realizar uma volta completa no ciclo construir-medir-aprender. É através dele que é possível obter comentários dos clientes sobre a solução proposta, para realizar uma aprendizagem com foco no menor esforço e menor tempo de desenvolvimento [RIES, 2011].

Ao fim do ciclo do *Lean Startup*, o empreendedor deve tomar uma decisão: pivotar ou não, ou seja, continuar no caminho pré-estabelecido na etapa de ideação ou muda-lo (pivotando). E, segundo RIES (2011) essa é a grande vantagem de sua metodologia: ela ajuda a construir uma organização capaz de testar suas suposições sistematicamente e de forma rápida através do *feedback* contínuo.

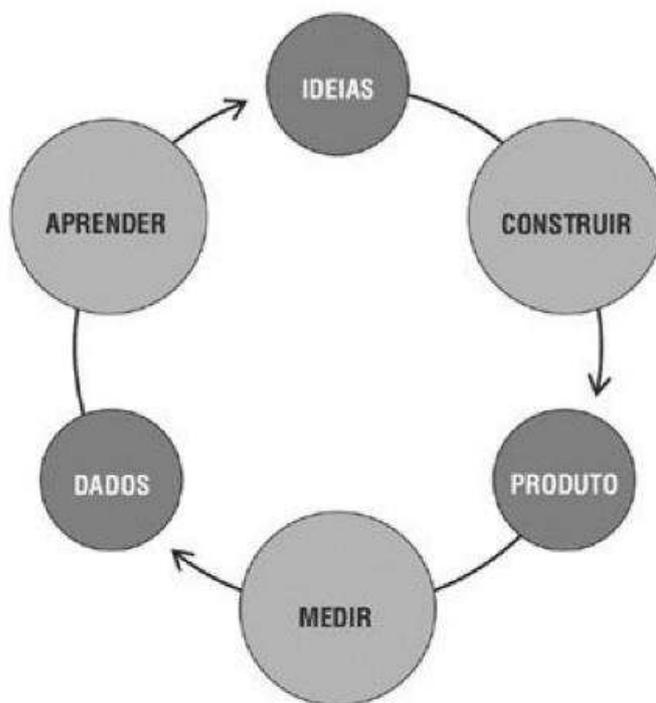


Figura 2.4 - Ciclo construir-medir-aprender do Lean Startup

As duas metodologias citadas, *Lean Startup* e *Customer Development*, possuem semelhanças na forma em que atuam fazendo referência constante ao contato direto com cliente. O *Lean Startup* faz referência a algumas boas práticas de engenharia de *software* como testes unitários e de usabilidade, integração contínua, desenvolvimento incremental, dentre outros. Essas boas práticas são somente referenciadas dentro do ciclo construir-medir-aprender, sem especificar como estas práticas estão inerentes a metodologia.

Em ambas metodologias, não são especificadas de forma detalhada ferramentas, fluxos de processo e procedimentos que podem auxiliar na descoberta e validação de mercado, durante o desenvolvimento do *software*. Por isso, muitas *startups* não realizam as boas práticas de engenharia de *software*, podendo trazer problemas tanto no ciclo de desenvolvimento. Esses problemas podem trazer prejuízos ao projeto em relação a prazos, na

qualidade percebida pelo cliente, além da dificuldade na descoberta e entendimento de um novo mercado.

2.4 BUSINESS MODEL CANVAS

Ao criar uma *startup*, além da preocupação com a parte técnica, e desenvolvimento do produto, outro pilar também se torna fundamental: o modelo de negócios. De acordo com OSTTERWALDER et al. [2009], um modelo de negócios “descreve a lógica de como uma organização cria, entrega, e captura valor” [OSTERWALDER; PIGNEUR, 2016]. O *Business Model Canvas* (BMC) é uma ferramenta utilizada para criação de um modelo de negócios de forma visual e dinâmica [BLANK; DORF, 2012]. Sua aplicação tem como benefício permitir um fácil entendimento de um negócio, através da análise dos pontos que podem positiva e/ou negativamente afetar a proposta de valor de um negócio [PROENÇA; NADALI; BORBINHA, 2015].

O Canvas, como o BMC também é conhecido, é uma ferramenta dinâmica que deve ser constantemente avaliada e alterada. Num primeiro momento, cada quadrante deve ser preenchido com hipóteses sobre o modelo de negócios a ser avaliado. Posteriormente, essas hipóteses serão confirmadas ou descartadas, de acordo com o contato direto com os seus potenciais clientes [BLANK; DORF, 2012].

A Figura 2.5 ilustra o BMC com seus nove quadrantes, utilizados para sumarizar os principais pilares que influenciam um modelo de negócios. São eles: Segmentos de Clientes, Proposta de Valor, Canais, Relações com Clientes, Fontes de Renda, Atividades Chave, Recursos Chave, Parcerias Chave e Estrutura de Custos [OSTERWALDER; PIGNEUR, 2016]. Cada um desses quadrantes deve ser preenchido de acordo com a descrição abaixo:

- **Segmentos de Mercado (SM):** descreve qual grupo de pessoas ou organizações que são os principais clientes do modelo analisado. Quem são as pessoas diretamente impactadas pela proposta de valor e que pagariam pelo produto e/ou serviço oferecido?;
- **Proposta de Valor (PV):** descreve as razões pelas quais os clientes utilizariam a solução proposta. Quais são as necessidades dos clientes, e como atende-las?;
- **Canais (CC):** descreve como a empresa comunica e atinge os segmentos de clientes para entregar a proposta de valor definida. Quais são os principais canais que levam a proposta de valor para os clientes?;
- **Relações com Clientes (RC):** descreve os tipos de relações que a empresa terá com seus segmentos de clientes. Como os clientes serão captados e retidos?;
- **Fontes de Renda (FR):** descreve como a empresa irá ganhar dinheiro. Como será o modelo de pagamento pelo produto e/ou serviço?;
- **Atividades Chave (AC):** descreve as principais atividades que devem ser realizadas para fazer o modelo de negócios rodar. Quais são as principais atividades que a proposta de valor definida necessita para ser executada?;
- **Recursos Chave (RCh):** descreve quais são os principais recursos necessários para fazer o modelo de negócios funcionar. Quais são os principais recursos necessários para criar e oferecer a proposta de valor definida?;
- **Parcerias Chave (PC):** descreve quais parceiros e /ou fornecedores são necessários para fazer o modelo de negócios rodar. Quais são as principais alianças quem devem ser feitas para otimizar, reduzir os riscos ou trazer recursos para o modelo de negócios?; e

- **Estrutura de Custos (EC):** descreve todos os custos necessários para garantir a operação do modelo de negócios. Quais são os principais custos necessários para iniciar e manter a operação do modelo de negócios?

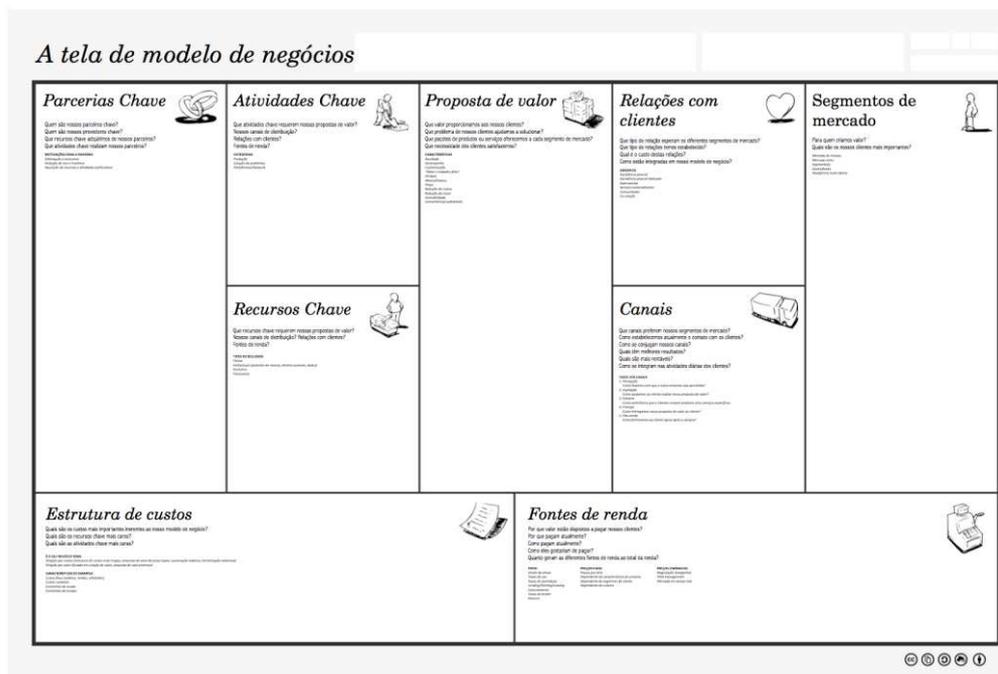


Figura 2.5 - Business Model Canvas

O preenchimento de cada quadrante auxilia a organização e alinhamento das informações relacionadas as atividades deste novo negócio [SHANAND; OLABARRIAGA, 2014]. Geralmente, o BMC é utilizado como uma ferramenta de auxílio às metodologias *Lean Startup* e *Customer Development*, unindo a modelagem de negócios com metodologias de desenvolvimento e avaliação de mercado [BLANK; DORF, 2012].

Desta forma, o seu preenchimento inicia-se juntamente com as primeiras etapas do *Lean Startup* e *Customer Development*, auxiliando os empreendedores a definir suas hipóteses de como criar valor e vendê-lo ao mercado. Após o seu preenchimento inicial, é feito um planejamento de como essas hipóteses serão testadas, com o objetivo de confirmá-las e/ou descartá-las conforme conversa com potenciais clientes [BLANK; DORF, 2012].

Para validar essas hipóteses, pode-se fazer o uso do MVP, descrito na seção 2.3. E, ao final do ciclo de validação, é analisado se as hipóteses descritas foram confirmadas ou não, podendo ser removidas e substituídas por novas hipóteses. Essa alteração, também é conhecida como *pivotar* [BLANK; DORF, 2012].

2.5 DESIGN THINKING

Além do *Customer Development* e *Lean Startup* uma metodologia bastante utilizada é o *Design Thinking*. Como o próprio nome diz, ela tem como princípio o *design* se referindo à maneira do designer de pensar [VIANNA et al., 2011].

Além disso, ela é determinada como uma metodologia que visa a resolução de problemas [PNIEWSKA; ADRIAN, 2013]. Portanto, durante a sua aplicação, busca-se formular questionamentos através da compreensão dos fenômenos para os quais respostas são

obtidas fazendo com que haja um melhor encaixe entre problema e solução [VIANNA et al., 2011]. E, através desse questionamento, busca-se um balanço harmonioso entre três pontos fundamentais: o que é funcionalmente possível de desenvolver num futuro próximo, o que é provável de virar um modelo de negócios sustentável e, por fim, o que faz sentido para as pessoas [BROWN, 2009].

O *Design Thinking* não possui uma sequência de etapas que devem ser seguidas, e sim pontos de destaque e marcos que ajudam no desenvolvimento de um projeto. São eles: inspiração, ideação e implementação [BROWN, 2009].

Esses marcos são obtidos através da realização de cinco etapas distintas: (i) inspiração que utiliza a empatia para a obtenção de entendimento do problema abordado; (ii) definição para selecionar quais dos problemas serão abordados; (iii) ideação com o objetivo de listar soluções que resolvem os problemas abordados; (iv) implementação contemplando as etapas de prototipar, transformando a solução em algum produto e/ou serviço concreto; e (v) testar, para a obtenção de comentários dos usuários [FARNSWORTH; KUMAR; KENNEDY, 2013]. Veja Figura 2.6 - Etapas do Design Thinking.

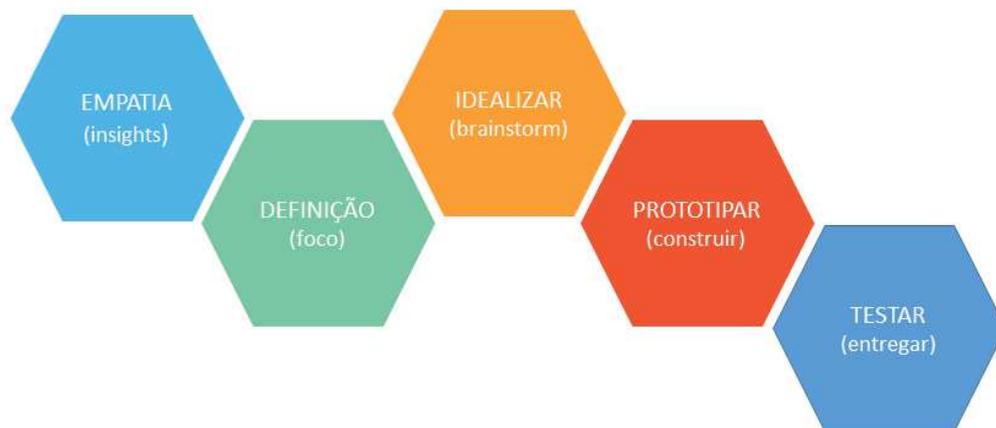


Figura 2.6 - Etapas do Design Thinking

Para cada fase no *design thinking* são definidas ferramentas para auxiliar durante a execução do projeto. Na fase de empatia, por exemplo, são realizadas pesquisas, entrevistas e sessões generativas que visam um maior entendimento e conhecimento do problema e potencial cliente.

A sessão generativa consiste numa imersão em profundidade, na qual grupos de potenciais usuários são reunidos numa sala para discutir determinado tema. A partir dessa discussão, o mediador conseguirá entender melhor seus potenciais clientes, como eles agem, sentem e lidam com o problema abordado [VIANNA et al., 2011].

Um dos principais artefatos de saída da fase de empatia, são as personas. Uma persona é construída visando a obtenção de uma descrição precisa das características de um grupo de usuários, com suas necessidades, objetivos, habilidades, dentre outras características [CHANG; LIM; STOLTERMAN, 2008]. Sua utilização auxilia aos *designers* a entenderem os principais objetivos e padrões de comportamento de um usuário [BODKER et al., 2012].

A persona deve ser obtida através do resultado das entrevistas e pesquisas realizadas durante a fase de empatia [MULDER; YAAR, 2007]. Ao criá-las, é importante definir uma identidade através de nome, idade, sexo e foto da persona criada; qual seu status em relação ao projeto: usuário primário (impactado diretamente), secundário (impactado indiretamente) e/ou um *stakeholder* (uma das partes interessadas). Outros pontos também são necessários saber sobre uma persona: quais são seus objetivos e habilidades, quais tarefas básicas ou críticas que essa persona realiza, além da frequência e importância dessas tarefas [BODKER

et al., 2012]. A Figura 2.7 ilustra um exemplo de uma persona utilizada no *workshop* de *Design Sprint* do *Google*. Nesta figura, é possível destacar quais as características que devem ser levadas em consideração para a criação de uma persona.



Cynthia
29 anos

- Estudante de pós graduação em Design de interiores
- Foi para Nova Iorque sozinha fazer intercâmbio e estágio na área
- Gosta de estudar ao ar livre e é adepta de longas caminhadas em parques

Próxima empreitada: Aproveitar a estada nos EUA para fazer um tour.

Grandes Necessidades

Ela precisa de um jeito de _____
para obter a experiência de _____
porque ela valoriza _____

Figura 2.7 - Exemplo de persona utilizado pelo Design Sprint

Durante a etapa de definição, todas as informações coletadas na fase de empatia são consolidadas e analisadas. Nesta fase, também são utilizadas ferramentas para facilitar essa análise. Dentre elas, é possível citar o mapa de resumo, mapas conceituais e jornada do usuário [VIANNA et al., 2011].

A jornada do usuário é uma ferramenta utilizada para representar graficamente as etapas de relacionamento do cliente com o produto ou serviço o [VIANNA et al., 2011]. Nela, são descritas situações hipotéticas de como o usuário utilizará a solução proposta, seja através de desenhos e/ou alguma descrição sobre a situação representada [BROWN, 2009].

As situações apresentadas na jornada do usuário devem representar todos os estágios de interação entre usuário e ferramenta, desde o início, com o momento de pré-utilização até o final, com a pós-utilização [BROWN, 2009]. Esta ferramenta auxilia a equipe a entender todo o ciclo de relacionamento do cliente, possibilitando uma análise das expectativas do usuário a cada momento de interação [VIANNA et al., 2011]. A Figura 2.8 ilustra um modelo de jornada de usuário.



Figura 2.8 - Jornada do usuário.

2.5.1 Design Sprint

O *Design Sprint* é uma metodologia desenvolvida pelo o *Google Ventures* com base no *Design Thinking* e *Scrum*. Esta metodologia mistura o conceito de agilidade das *sprints* tirados do *Scrum* (descrito na seção 2.2) com o *Design Thinking* (seção 2.5).

Com um ciclo bem-definido e reduzido, as pessoas envolvidas num projeto que roda o *Design Sprint*, possuem apenas 5 dias para responder perguntas críticas relacionadas a negócios, *design* através da prototipação e teste de ideias [GOOGLE VENTURES, 2015].

A metodologia é dividida em 5 etapas: Entender, Divergir, Decidir, Prototipar e Validar. Durante a etapa entender, assim como na etapa empatia do *design thinking*, a equipe envolvida no projeto deve focar em conhecer bem seus usuários e suas necessidades. Durante esta etapa, também são definidas personas, conforme demonstrado na Figura 2.7 [DIREKOVA, 2015].

Na etapa divergir, são geradas ideias (brainstorming), no qual as pessoas envolvidas no projeto devem construir uma lista de novas ideias. Num primeiro momento, nenhum filtro de ideias deve ser feito, para que sejam levantadas a maior quantidade possível de ideias.

Após levantadas, elas serão filtradas e analisadas de acordo com a sua complexidade técnica e valor para usuário [GOOGLE DEVELOPERS, 2016]. A Figura 2.9 ilustra a matriz de complexidade por impacto para análise de ideias.

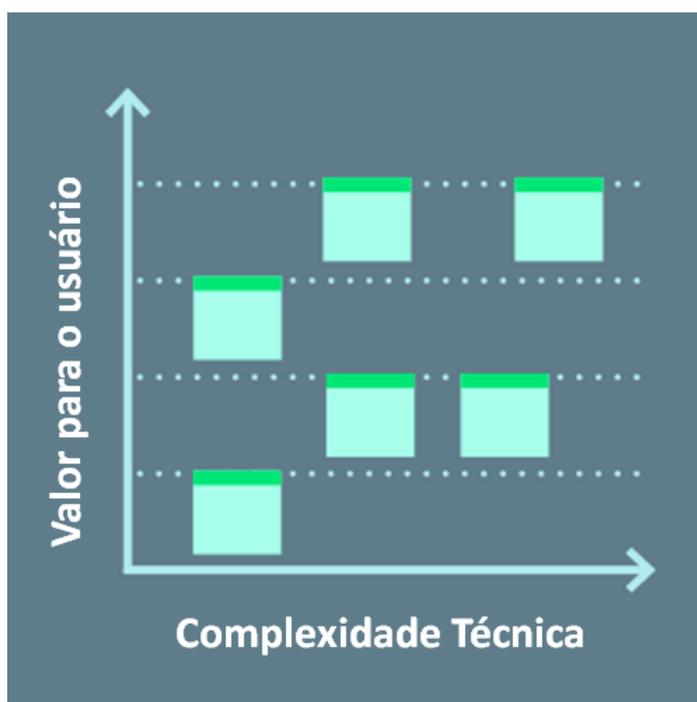


Figura 2.9 - Matriz de análise de ideias.

A matriz de complexidade por impacto deve ser preenchida da seguinte forma: as ideias geradas devem ser preenchidas em pequenos pedaços de papéis auto-adesivos. Após preencher os papéis com as ideias, elas devem ser colocadas nos respectivos quadrantes da matriz, de acordo com a sua complexidade técnica e o valor para o usuário. Desta forma, quanto mais distante do eixo horizontal, o papel com a ideia for posicionado, maior o valor para o usuário. E, quanto mais distante do eixo vertical da matriz, maior a sua complexidade técnica [GOOGLE VENTURES, 2015].

No *Design Sprint*, recomenda-se que as ideias que possuem um maior valor para o usuário e maior complexidade técnica, sejam selecionadas para trabalhar na criação de uma

nova *startup*. Essa seleção deve-se ao fato de uma ideia muito complexa ser difícil de ser copiada, além do alto impacto, fornecer maiores chances de descoberta de mercado [GOOGLE DEVELOPERS, 2016].

Durante a fase prototipar, é necessário transformar as ideias em protótipos. A ferramenta 8 momentos-chave é utilizada neste momento como um protótipo, forçando a equipe a explorar as ideias de forma visual.

Nesses momentos-chave, também são descritos cenários de interação do usuário com a solução proposta, definindo as experiências e interações possíveis que o usuário pode ter com o *software* [GOOGLE DEVELOPERS, 2016]. Esta ferramenta também foi utilizada como uma forma de análise da jornada do usuário, possibilitando uma jornada mais focada e resumida na utilização da ferramenta (seção 4.1 FASE DE REQUISITOS).

2.6 COMPARATIVO METODOLOGIAS

Analisando as metodologias estudadas, é possível destacar pontos em que elas são diferentes e se complementam. Conforme demonstrado na Tabela 1.1 - Comparativo metodologias, percebe-se que as metodologias atualmente utilizadas por *startups* possuem uma maior preocupação com a área de negócios e construção de uma empresa. Essa construção deve-se pelo fato de uma *startup* ter como princípio a possibilidade de criação de uma empresa de forma rápida e ágil, através do desenvolvimento de um produto que seja repetível e escalável.

Já, as metodologias utilizadas por grandes empresas *Scrum* e RUP, possuem um foco maior na Engenharia de *Software*, através da construção de uma solução que seja gerenciável e de fácil manutenção. Portanto, as metodologias *Scrum* e RUP possuem fases, artefatos e papéis bem-definidos, visando um melhor gerenciamento do *software* desenvolvido.

O gerenciamento é realizado para evitar problemas que ocorrem durante o desenvolvimento de um sistema complexo, tais como: complexidade no atendimento dos prazos, falta de atendimento dos requisitos definidos, estimativa de custos e recursos precárias. Esses problemas também trazem inúmeros prejuízos para o desenvolvimento de uma nova empresa, já que muitas vezes o *software* em desenvolvimento acaba não sendo liberado para o cliente, ou até apresentando inúmeras falhas, tornando a sua manutenção precária.

O alto índice de falhas e a complexidade no gerenciamento do projeto traz inúmeros prejuízos financeiros para uma empresa iniciante. Portanto, as metodologias *Scrum* e RUP sugerem atividades que visam facilitar e diminuir os prejuízos causados por um desenvolvimento desorganizado.

Por outro lado, as metodologias utilizadas por *startups* focam na descoberta do mercado através da construção do Produto Mínimo Viável (MVP), visando uma maior agilidade no desenvolvimento e uma grande interação com o usuário. Essa agilidade é necessária, já que o tempo de desenvolvimento se torna menor, e desta forma, o produto chega mais rápido para os clientes testarem e avaliarem se eles atendem a suas necessidades.

As metodologias *Lean Startup*, *Customer Development* e *Design Thinking*, partem do pressuposto que o usuário não conhece suas reais necessidades. Portanto é necessário demonstrar um produto (MVP) ou protótipo para a obtenção de um melhor entendimento de quais são as suas necessidades reais. Desta forma, o produto é construído em conjunto com seus usuários, através de comentários e avaliações feitas com o MVP. Esse tipo de abordagem, faz com que as funcionalidades sejam definidas através dos *feedbacks* dos usuários, após um ciclo de desenvolvimento.

A Figura 2.10 representa esse tipo de abordagem, na qual os requisitos são levantados apenas após um primeiro ciclo de utilização do MVP, no LS, CD e DT. Nesta figura, é possível perceber que as metodologias *Scrum* e RUP possuem as fases de Requisitos e Projeto bem definidas, através de artefatos que são construídos para as demais fases. Além disso, esses requisitos são definidos no início do projeto, juntamente com seus usuários.

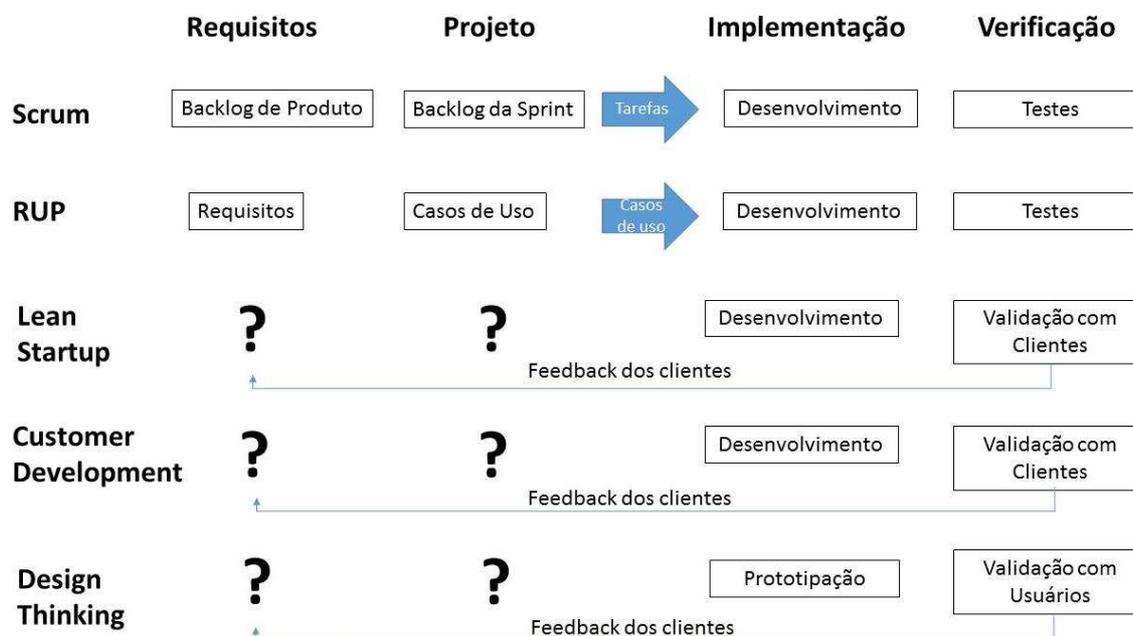


Figura 2.10 - Comparativo metodologias por fase

As metodologias *Lean Startup*, *Customer Development* e *Design Thinking* não possuem artefatos definidos para as fases de Requisitos e Projeto. Isso deve-se ao fato dessas metodologias incentivarem a descoberta desses requisitos através da interação diretamente com o potencial usuário. Portanto, os requisitos são definidos após a primeira interação, seja através de um MVP (que pode ou não ser um *software*), ou através de um protótipo.

Essa abordagem adotada pelas metodologias de *startups* é necessária, já que os potenciais clientes, que serão usuários da ferramenta proposta, não são conhecidos. Portanto, essas metodologias pregam bastante a descoberta de mercado (potenciais usuários). O que não ocorre com as demais metodologias, *Scrum* e RUP, já que os usuários já são conhecidos pela equipe de projeto.

2.7 CONCLUSÃO

Este capítulo apresentou os principais conceitos e técnicas das metodologias de desenvolvimento de *startups*: *Customer Development*, *Lean Startup* e *Design Thinking*, e metodologias de desenvolvimento de *software* *Scrum* e RUP, e a ferramenta *Business Model Canvas*. As metodologias e ferramentas citadas nesse capítulo serviram de base para a criação da metodologia proposta.

3 TRABALHOS RELACIONADOS

O estudo de trabalhos relacionados foi feito através de pesquisa bibliográfica nas bibliotecas digitais: IEEE, ACM, Scopus e Google Acadêmico, além de buscas online por sites e blogs de referência. Foi feita uma busca *ad hoc* com base nas metodologias de base deste estudo: *Lean Startup*, *Customer Development*, *Design Thinking*, *Scrum*, *RUP* e a ferramenta *BMC*. Além dessa busca, foi utilizada a técnica de *snowballing* analisando as referências dos artigos selecionados como base para o estudo.

A busca *ad hoc* foi feita com base na combinação entre metodologias de desenvolvimento de *startups*, com foco em negócios e descoberta de mercado, e as metodologias de desenvolvimento de *software*, com foco nas técnicas e ferramentas para o desenvolvimento de *software*. A seguir estão as palavras-chave buscadas durante o estudo:

("lean startup" OR "customer development" OR "design thinking" OR "design sprint" OR "business model canvas" OR "BMC")

AND

(agile OR "agile method" OR "agile development" OR "scrum" OR "rational unified process" OR "RUP")

Tabela 3.1 - Quantidade de artigos encontrados por biblioteca

Biblioteca	Quantidade Artigos
Scopus	48
ACM	07
IEEE	18
Google Acadêmico	1.520

Os artigos encontrados nas bases Scopus, ACM e IEEE foram filtrados através da análise de seus títulos. O total de artigos relacionados dentro dessas três ferramentas ficou em 17 artigos, ou seja, 23% dos artigos retornados na busca. A Figura 3.1 - Quantidade de artigos publicados por ano ilustra o gráfico gerado com a quantidade de artigos publicados por ano. Nela, é possível perceber um aumento de volume de artigos publicados nos últimos seis anos, reforçando a importância do estudo realizado.



Figura 3.1 - Quantidade de artigos publicados por ano

Durante análise dos artigos no Google Acadêmico, percebeu-se a grande quantidade de artigos de relato da aplicação das novas metodologias de *startups* em grandes empresas e/ou alguns projetos. Além disso, também foi possível encontrar descrições de como aplicar essas novas metodologias, através de um passo-a-passo. Poucos artigos descreviam a união entre metodologias de *startups* e ágeis.

Para a seleção desses artigos, foi feito um filtro por data, considerando apenas os artigos entre o período de 2012 e 2015. Além disso, foi feito o filtro por relevância fornecido pelo Google Acadêmico, além da eliminação de blogs e sites desconhecidos. Após esse filtro, foi realizada uma análise dos títulos e resumos, ficando assim com um total de 12 artigos.

Após o filtro com base nos títulos, foi feito um segundo filtro com base nos resumos dos artigos selecionados. Com este filtro, restaram 9 artigos, dos quais apenas 5 possuem proposta semelhante ao estudo realizado: união de boas práticas de negócios em metodologias de desenvolvimento e *startups*, e técnicas e ferramentas para desenvolvimento de *software* de metodologias ágeis.

Nas pesquisas, foi possível perceber que a maioria dos estudos relacionados fazem união do *Design Thinking* com *Scrum*, focando na inserção de boas práticas de *design* e interação com clientes. Em SINGH (2008), por exemplo, é proposto o “U-*Scrum*” que consiste na aplicação do *Scrum* com um viés de usabilidade. A metodologia propõe a adição de um *Product Owner*, o *Usability Product Owner* (UPO), que teria um foco mais específico em usabilidade e na experiência do usuário (*User Experience - UX*).

O UPO se torna o responsável por implementar uma visão mais focada em UX através da criação de perfis de usuários, ou personas, que descrevem detalhadamente as principais características de um grupo de possíveis usuários para o sistema. Além disso, o *Usability Product Owner* tem como responsabilidade trabalhar diretamente com os clientes, e ajudar na criação do *backlog* de produto com histórias que envolvem características de usabilidade com alta prioridade.

As personas criadas têm um papel muito importante durante o desenvolvimento já que trazem para o desenvolvedor uma visão mais realística sobre o uso da solução desenvolvida. Elas refletem os objetivos, habilidades e atitudes dos clientes do projeto, descrevendo como o usuário quer se sentir ao utilizar a solução proposta.

A criação de personas é utilizada para que todo o time envolvido no projeto conheça as intenções do usuário, como ele age e quais são seus objetivos. Suas informações são derivadas de uma combinação de dados provenientes de pesquisa de mercado, estudos etnográficos e observações anedóticas. O UPO é o responsável por obter todas essas informações com os *stakeholders* envolvidos e fazer a construção das personas com base nas informações obtidas.

Após serem definidas, as personas são apresentadas para a equipe do projeto através de reuniões, e pôsteres descritivos. Desta forma, elas passam a ser utilizadas como base para discussões estratégicas de desenvolvimento, tais como a determinação de viabilidade de uma determinada interface de usuário. Além disso, elas também podem ser utilizadas nas descrições das histórias criadas durante uma *sprint* de desenvolvimento.

Além das personas, outros artefatos de UX e usabilidade são desenvolvidos no decorrer da metodologia U-*Scrum*. São eles: visão de *user experience*, plano de produto, *storyboard*, *wireframes*. Com esses artefatos, o UPO deve fazer validações com os usuários com o objetivo de obter *feedbacks* sobre o produto em desenvolvimento. Os resultados dessas validações são aplicados nas próximas *sprints*.

Dentro do U-*Scrum* é possível destacar a inserção de uma pessoa dentro da equipe focada em entender bem o perfil do usuário. Além disso, há a utilização de ferramentas visando uma melhor experiência de usuário, assim como é feito no *Design Thinking*. Na

Tabela 3.2 estão listados os pontos fortes e fracos do U-Scrum que auxiliaram na criação do *Startup Kaizen*.

Os principais benefícios demonstrados pelo U-Scrum estão relacionados ao desenvolvimento de uma inovação de forma com uma maior preocupação e participação dos usuários, através de técnicas de usabilidade e validação com os usuários. A abordagem feita pelo U-Scrum com a utilização do UPO possibilita obter um conhecimento e uma preocupação maior sobre o possível mercado, assim como é feito no *Design Thinking* descrito na seção 2.5.

Em 2013, BJÖRK et al. (2013) trazem uma abordagem voltada para *startups* em estágio inicial, ESSSDM (*Early Stage Software Startup Development Model*). O ESSSDM consiste numa metodologia criada com base no *Lean Startup* através da utilização do conceito de levantamento de hipóteses e validação. Essa metodologia fornece um passo-a-passo de como iniciar o desenvolvimento de um *software*, que se inicia com o levantamento das hipóteses que serão validadas no decorrer do processo de desenvolvimento de *software*. A partir das hipóteses levantadas, as ideias validadas são formatadas e passam por quatro etapas diferentes. Essas etapas consistem desde o entendimento do problema, até as validações qualitativas e quantitativas [BJÖRK; LJUNGBLAD; BOSCH, 2013].

A metodologia possui duas fases diferentes: na primeira fase é feito um levantamento e gerenciamento das ideias, hipóteses relacionadas a potenciais problemas a serem resolvidos, além de modelos de negócios que devem ser validados. Na segunda fase, uma ideia é selecionada e validada utilizando o ciclo construir-medir-aprender do *Lean Startup* até que a solução seja escalável, ou seja necessário realizar uma mudança de estratégia, ou seja, seja necessário pivotar a ideia [BJÖRK; LJUNGBLAD; BOSCH, 2013].

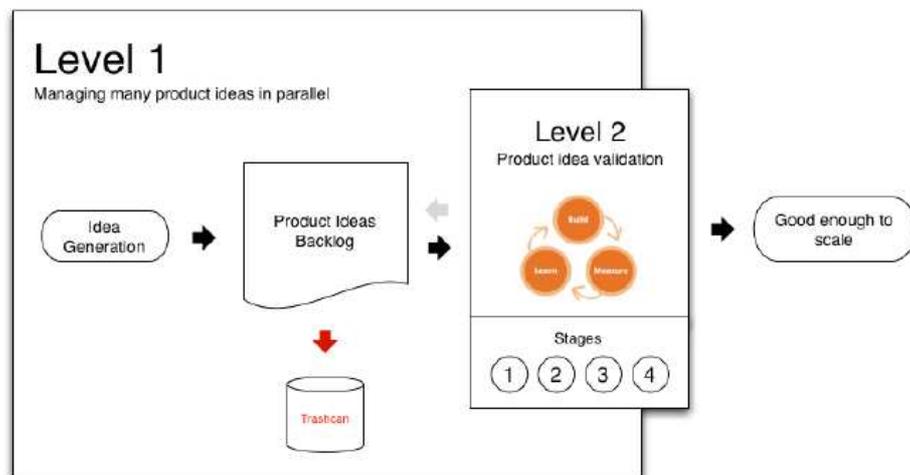


Figura 3.2 - ESSSDM Fase 1: Geração de ideias

A Figura 3.2 o modelo ESSSDM, no qual no primeiro nível são geradas ideias das quais algumas (ou apenas uma) são selecionadas e posteriormente trabalhadas no nível 2. O primeiro passo do nível 1 consiste na geração de ideias, podendo ser genéricas ou mais específicas. Essa geração de ideias é realizada para o levantamento de uma ou mais ideias, que poderão ser validadas individualmente, ou ao mesmo tempo, na fase seguinte [BJÖRK; LJUNGBLAD; BOSCH, 2013].

Após a geração de ideias, as ideias selecionadas são formatadas visando uma fácil comparação entre elas. Conforme a validação ocorre, as ideias devem ser constantemente priorizadas, mantendo sempre as mais promissoras em andamento. E, assim que uma ideia

ganhar tração, e demandar mais recursos, as demais devem ser congeladas momentaneamente [BJÖRK; LJUNGBLAD; BOSCH, 2013].

No segundo nível, após a seleção das ideias que serão trabalhadas, elas devem passar por uma validação sistemática. Essa validação possui 4 sub-níveis diferentes: (i) entendimento do problema, (ii) definição da solução, (iii) validação qualitativa, (iv) validação quantitativa. Ilustrado na Figura 3.3 - ESSSDM Fase 2: Validação. Para cada sub-nível existem riscos diferentes que são priorizados e armazenados num *backlog* [BJÖRK; LJUNGBLAD; BOSCH, 2013].

Após a identificação dos riscos, são identificadas as hipóteses para então iniciar o ciclo construir-medir-aprender do *Lean Startup*. E, a cada aprendizado obtido durante este ciclo, a ideia e seu risco devem ser analisados novamente até que o produto se torne escalável ou a ideia invalidada [BJÖRK; LJUNGBLAD; BOSCH, 2013].

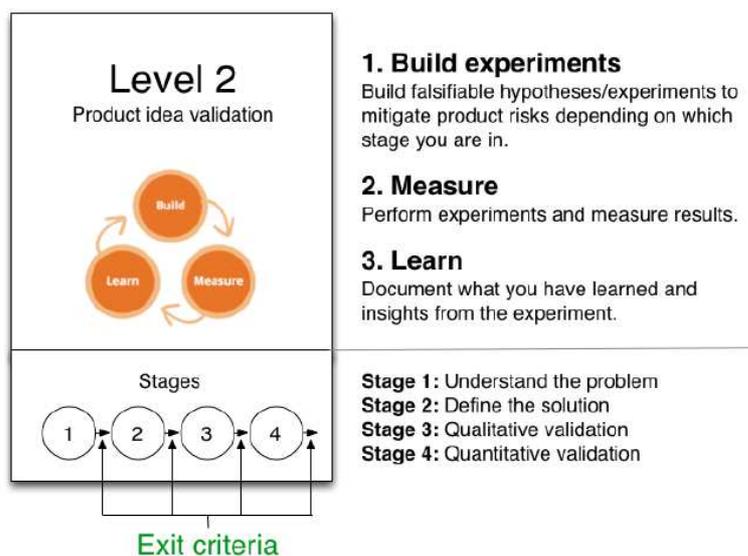


Figura 3.3 - ESSSDM Fase 2: Validação

Durante o passo de entendimento de problema, o foco maior é em obter um conhecimento do problema e das pessoas que sofrem dele. Para prosseguir para o próximo passo, é necessário que pelo menos metade dos potenciais clientes deem uma indicação positiva da ideia do produto [BJÖRK; LJUNGBLAD; BOSCH, 2013].

No segundo passo, definição da solução, seu objetivo principal é a definição de uma solução para o problema identificado anteriormente e comunica-lo para seus potenciais clientes. Após a definição da solução, é definido um MVP que deve ser utilizado para validação com os clientes. Com a confirmação de que clientes estão dispostos a pagar pela solução definida, o MVP definido, e da identificação de clientes que são os adotantes iniciais da solução proposta, é possível dizer que a solução é aceita. Com a solução aceita, a equipe deve prosseguir para o próximo passo: validação qualitativa [BJÖRK; LJUNGBLAD; BOSCH, 2013].

Durante o terceiro passo, o MVP é desenvolvido e liberado para os adotantes iniciais da solução. Com a utilização do MVP é possível validar que a solução definida no passo anterior realmente resolve o problema dos adotantes iniciais, e que eles realmente pagam por essa solução. Além disso, é nesse passo que a proposta de valor oferecida ao usuário é validada juntamente com os clientes adotantes iniciais [BJÖRK; LJUNGBLAD; BOSCH, 2013].

Por fim, no último passo, a solução é liberada para um grupo maior de usuários. Com essa liberação, são validadas hipóteses de que se existe um grupo maior de clientes dispostos

a pagar pela solução e como fazer para alcançá-los através dos canais de vendas [BJÖRK; LJUNGBLAD; BOSCH, 2013].

A vantagem da utilização do ESSSDM está na transformação do *Lean Startup* numa metodologia bem-estruturada através da criação e elaboração de fases tornando-o mais fácil de aplicar. Essa estruturação possibilita uma validação mais estruturada, através da definição e identificação do problema e seus potenciais riscos.

A desvantagem em relação ao ESSSDM é que ele foca somente no *Lean Startup*, deixando de lado as demais metodologias: *Design Thinking*, *Scrum* e RUP, além da ferramenta *Business Model Canvas*. O ESSSDM auxilia a estruturação do pensamento para a realização de uma validação com riscos e objetivos bem definidos. Na Tabela 3.2 estão listados os pontos fortes e fracos do ESSSDM.

Também em 2013, Gothelf et al. defendem uma abordagem que mistura os conceitos de *Lean Startup*, desenvolvimento ágil e *Design Thinking*, o *Lean UX*. A principal característica do *Lean UX* está no rápido desenvolvimento de *software* com foco no usuário. Seu objetivo consiste em produzir um produto que satisfaz as necessidades do usuário, o mais rápido possível, e com os recursos mínimos necessários, evitando desperdícios.

O *Lean UX* não consiste numa metodologia rigorosa na qual todas as atividades e artefatos devem ser rigorosamente seguidos. O *Lean UX* consiste numa forma mais ágil de aplicação dos conceitos de *user experience* viabilizando projetos através da rápida validação de hipóteses, ideias e soluções. Seu fundamento é baseado na utilização de formas simples de entregáveis, esboços, protótipos [CATARINAS, 2014]

Liikkanen et. al (2014) descrevem a criação de um manifesto criado com base no *Lean UX*. Esse manifesto possui seis princípios básicos: (i) preferência pela validação do cliente, ao invés do lançamento de produtos sem usuários finais bem-definidos; (ii) projeto de *design* colaborativo ao invés de um único responsável pelo projeto; (iii) resolução de problemas, ao invés da adição de funcionalidades sem sentido; (iv) definição de métricas-chaves ao invés da não utilização de nenhuma métrica, (v) flexibilidade na utilização de ferramentas apropriadas vs utilização de uma metodologia rigorosa; (vi) *design* eficiente, vs documentação pesada.

Assim como defendido no U-*Scrum*, o *Lean UX* considera os clientes e o *design* como partes inerentes ao processo. Desta forma, o *Lean UX* utiliza as *sprints* do *Scrum* como base no ciclo de desenvolvimento [LIKKANEN et al., 2014].

As *sprints* são divididas em dois momentos principais: planejamento da *sprint* e *sprint* de desenvolvimento. Esses dois momentos possuem 4 etapas diferentes: *design* e *feedback* (planejamento); desenvolvimento e testes de usabilidade (*sprint* de desenvolvimento). As *sprints* de desenvolvimentos, ocorrem geralmente em duas semanas, e nelas são feitos dois ciclos das etapas de desenvolvimento e validação com o usuário. Desta forma, ao final de cada semana, após o desenvolvimento, é feita uma rodada de validação com o usuário. Veja Figura 3.4, que ilustra a metodologia *Lean UX*.



Figura 3.4 - Metodologia Lean UX

Cada *sprint* possui um tema específico, sendo o primeiro relacionado a criação de um protótipo e exercícios. Esses exercícios são utilizados para a geração de ideias que fornecem um esqueleto de *design* que deverá ser trabalhado e validado nas próximas *sprints*.

No início de cada *sprint*, são definidos os objetivos da validação que irá ocorrer. Desta forma, a cada *sprint* esses objetivos podem ser confirmados ou não ao final do processo. Essa análise é feita com base nos *feedbacks* diretos de usuários e/ou potenciais clientes [LIKKANEN et al., 2014].

Os dados obtidos com os *feedbacks* dos usuários, são analisados e utilizados como base para a criação das histórias para o desenvolvimento da próxima *sprint*. Essa abordagem força a participação do usuário em cada ciclo, fazendo com que as funcionalidades desenvolvidas sejam planejadas de acordo com as necessidades e *feedbacks* do usuário [LIKKANEN et al., 2014].

Além das *sprints* do *Scrum*, o *Lean UX* utiliza também personas para identificação e definição de potenciais usuários. Essas personas são criadas no início do processo como hipóteses que também devem ser validadas (*protoperonas*). Além disso, elas são utilizadas como atores das histórias criadas para o desenvolvimento da *sprint*. Desta forma, uma história é descrita através de ações que são realizadas pelas personas, visando atingir seus objetivos [LIKKANEN et al., 2014].

O principal benefício apresentado pelo *Lean UX* deve-se ao fato de estar diretamente relacionado a uma metodologia de desenvolvimento de *startups*, o *Lean Startup*. Nessa abordagem foram inseridas boas práticas de UX respeitando o ciclo de construir-medir-aprender do *Lean Startup*.

Neste caso, o *Lean UX* foca na inserção de boas práticas relacionadas a UX em metodologias ágeis, como por exemplo, no *Scrum*. Diferente da metodologia defendida em SINGH [2008], o *Lean UX* traz a participação dos usuários para dentro das *sprints* de desenvolvimento. A Tabela 3.2 lista os principais pontos fortes e pontos fracos do *Lean UX*.

Em 2014, CATARINAS et. al definem uma metodologia que faz a união de *Lean Startup* e *Lean UX*. A metodologia é dividida em 7 passos: (i) definição de hipóteses, (ii) validação externa, (iii) entrevistas, (iv) criação de uma página de lançamento, (v) prototipação, (vi) testes de usabilidade e (vii) aprender. Os passos são distintos, e em cada um deles, são indicadas as atividades que devem ou não devem ser feitas em cada etapa.

Os três primeiros passos consistem na definição das hipóteses através da indicação de possíveis usuários e ir até os possíveis clientes para fazer entrevistas. Durante as entrevistas, a metodologia descreve alguns pontos críticos para a sua realização, tais como: focar em perguntas que contenham informações relevantes, atentar para não dar respostas às perguntas definidas antes de fazer entrevistas, evitar perguntas cujas respostas são apenas

“sim” ou “não”, entender melhor as respostas através de perguntas que levem a respostas mais completas [CATARINAS, 2014].

Após as entrevistas, é criada uma página de lançamento antes do produto estar pronto. Essa página de lançamento deve simular a solução, medindo o nível de interesse dos potenciais usuários. O conteúdo dessa página de lançamento deve passar uma informação como se a solução já estivesse pronta e disponível no mercado, incentivando seus usuários a já fazerem uma compra e/ou contratação da solução disponível. O objetivo da página de lançamento consiste na obtenção de usuários interessados para quando a solução estiver pronta [CATARINAS, 2014].

Com a página de lançamento criada, é construído um protótipo podendo ser de baixa ou alta fidelidade. Com a página de lançamento e o protótipo criados, são realizados testes de usabilidade. Esses testes de usabilidade visam a obtenção de aprendizado que, por sua vez, deverá ajudar na decisão de continuar ou pivotar [CATARINAS, 2014].

Além dos estudos citados anteriormente, pode-se citar também o *Design Sprint* descrito na seção 2.5.1 Criada em 2015, esta metodologia mistura o conceito de agilidade das *sprints* tiradas do *Scrum* com o *Design Thinking*. Ela consiste num ciclo ágil com um tempo pré-definido, utilizando o conceito de *timebox* do *Scrum* (2.2 SCRUM). O *Design Sprint* é utilizado para responder perguntas críticas relacionadas a negócios, *design* através da prototipação e teste de ideias [GOOGLE VENTURES, 2015].

Baseado nos trabalhos relacionados, foi possível perceber que existem estudos focados em unir as metodologias de *startups* e metodologias ágeis, principalmente o *Scrum*. Na maioria dos trabalhos estudados, nota-se uma preocupação na inserção de boas práticas relacionadas a UX, através da união de atividades do *Design Thinking* dentro do processo *Scrum*.

Através do estudo das metodologias propostas, é possível destacar a importância da constante participação do cliente dentro do processo. O cliente ocupa um papel de destaque ainda maior do que acontece nas metodologias ágeis, tornando-se um fator fundamental na evolução do desenvolvimento.

Outro fator importante, que deve ser destacado, consiste no fato da maioria das metodologias estudadas não se preocuparem com a validação de mercado. Apesar de algumas reforçarem a constante validação com os usuários, nenhuma delas se preocupa, conforme apresentado na Tabela 3.2.

Tabela 3.2 - Pontos Fortes e Pontos Fracos dos trabalhos relacionados

	Pontos Fortes	Pontos Fracos
U- <i>Scrum</i>	União de <i>Scrum</i> com boas práticas de UX e usabilidade; Adição do papel do UPO; Utilização de personas; Utilização das personas na descrição de histórias.	Não faz referência a modelo de negócios; Não utiliza o BMC; Não faz referência a boas práticas de qualidade e testes de <i>software</i> ; Não utiliza o MVP; Não detalha as atividades e ferramentas de cada fase.
ESSSDM	Levantamento de hipóteses e validação; Utilização de análise qualitativa e quantitativa; Estruturação da ideia antes da validação; Levantamento e priorização de riscos durante a validação;	Não faz referência a modelo de negócios; Não utiliza o BMC; Não faz referência a boas práticas de qualidade e testes de <i>software</i> ; Preocupa-se somente com as etapas iniciais de uma <i>startup</i> ; Possui foco em <i>startups</i> iniciantes;

	Realização de uma validação sistemática.	Não detalha as atividades e ferramentas em cada fase.
<i>Lean UX</i>	Utilização de métricas; Inserção de etapas de validação dentro das <i>sprints</i> de desenvolvimento; Cada <i>sprint</i> possui um tema específico; Utilização de <i>protopersonas</i> ; Utilização de personas na descrição de histórias.	Não faz referência a modelo de negócios; Não utiliza o BMC; Não faz referência a boas práticas de qualidade e teste de <i>software</i> ; Não detalha as ferramentas e atividades por etapa.
<i>Lean UX + Lean Startup</i>	Define técnicas e ferramentas que devem ser utilizadas durante a validação com o usuário; Utiliza uma página de lançamento para validação; Detalha as atividades que devem ser realizadas em cada fase; Detalha boas práticas de entrevistas com usuários; Define o tipo de MVP que deve ser utilizado.	Não faz referência a modelo de negócios; Não utiliza o BMC; Não faz referência a boas práticas de qualidade e testes de <i>software</i> ; Preocupa-se somente com as etapas iniciais de uma <i>startup</i> ; Possui foco em <i>startups</i> iniciantes.
<i>Design Sprint</i>	União das <i>sprints</i> do <i>Scrum</i> com boas práticas do <i>Design Thinking</i> ; Definição de um limite de tempo (<i>timebox</i>) para a realização do desenvolvimento; Utilização de personas; Utilização de <i>wireframes</i> (8 momentos-chave).	Não faz referência a modelo de negócios; Não utiliza o BMC; Não faz referência a boas práticas de qualidade e testes de <i>software</i> ; Possui foco maior no <i>design</i> ;

Conforme demonstrado na Tabela 3.2, cada metodologia apresentada neste capítulo possui seus pontos fortes e fracos. Nenhuma das metodologias faz referência a modelagem de negócios e utilização do BMC. Além disso, nenhuma delas apresenta boas práticas de Engenharia de *Software* com foco em qualidade e testes.

Todas as 5 metodologias desse capítulo destacam a importância de uma maior aproximação com o usuário, visando obter um melhor entendimento sobre a forma como ele irá interagir com a solução proposta. Apesar dessa preocupação com a validação diretamente com o usuário, o ESSSDM não faz o uso de personas para descrição do grupo de usuários que será trabalhado. Todas as outras metodologias utilizam personas para descrição do grupo de usuários.

Outro ponto importante de se destacar, é que as metodologias *Lean UX*, *Lean UX + Lean Startup* e ESSSDM apresentam diferentes formas de validação diretamente com o usuário. Porém todas utilizam do MVP como uma forma de primeiro contato com o potencial usuário, podendo ser através da criação de uma página de lançamento (*Lean UX + Lean Startup*) ou através de uma validação qualitativa através de entrevistas diretamente com potenciais usuários.

Por fim, é importante destacar que das 5 metodologias descritas neste capítulo, somente a ESSSDM não utiliza de *sprints* de desenvolvimento. As demais metodologias: *U-Scrum*, *Lean UX*, *Lean UX + Lean Startup* e *Design Sprint* utilizam o conceito de *timebox* (descrito na seção 2.2 **SCRUM**) das *sprints* do *Scrum* para a realização de um ciclo de desenvolvimento.

3.1 CONCLUSÃO

Com base nos trabalhos estudados, foi possível perceber que poucas metodologias fazem referência as boas práticas de engenharia de *software*. As metodologias não fornecem uma conexão entre descoberta de mercado e as boas práticas no desenvolvimento de *software*, deixando de lado pontos fundamentais para a gestão do desenvolvimento.

Além dos trabalhos estudados, existem outras propostas de metodologias aplicáveis em *startups*, porém com um foco muito maior à descoberta de mercado e validação da ideia através da prototipação. Através desse estudo foi possível perceber a tendência na união de metodologias ágeis de desenvolvimento de *software*, como o *Scrum*, e as demais metodologias atualmente utilizadas por *startups*: *Design Thinking*, *Lean Startup* e *Customer Development*.

4 METODOLOGIA STARTUP KAIZEN

Com o objetivo de responder à pergunta levantada durante esse estudo a inserção de boas práticas de Engenharia de *Software* pode favorecer uma *startup* durante a fase de validação, foi criado o *Startup Kaizen* (SK).

O *Startup Kaizen* foi criado através da união de boas práticas de Engenharia de *Software*, boas práticas de metodologias utilizadas por *startups* através da maior participação do usuário no ciclo de desenvolvimento, e a utilização do MVP, além de modelagem de negócios com a utilização do BMC. As fases bem-definidas do *Startup Kaizen*, tiveram como base o *Scrum* e *RUP*.

As fases que compõem o *Startup Kaizen* abrangem as atividades de: (i) especificação, representada pela fase de **Requisitos**, (ii) desenvolvimento, representada pela fase de **Implementação**, (iii) *sprint planning* do *Scrum*, representadas na fase de **Projeto**, (iv) validação, representado pela fase de **Validação** (Figura 4.1). Além das fases bem-definidas, a evolução pregada pelo *Lean Startup* e *Customer Development*, foi representada através do ciclo completo de aprendizagem, que tem como resultado o incremento da solução desenvolvida.

Em relação às metodologias de *startups*, os quatro pilares citados na seção 1.1 MOTIVAÇÃO E PROBLEMA, serviram como base para a criação do *Startup Kaizen*. São eles: (i) processo iterativo e incremental, através do desenvolvimento de um *software* funcional a cada rodada do ciclo de desenvolvimento, (ii) MVP, como o produto inicial que deve ser desenvolvido para validação de mercado, (iii) ciclo de aprendizado, através das reuniões de retrospectiva descritas na seção 2.2 e (iv) participação de clientes, com a validação constante de todos os requisitos que devem ser inseridos na solução.

Além dos pilares citados como base, boas práticas dos trabalhos relacionados também foram aplicadas para a definição do SK. As atividades inseridas são descritas nas próximas seções, nas quais são destacadas de quais metodologias elas originaram.

A primeira fase do SK consiste na identificação e definição do produto mínimo viável (MVP), retirado das metodologias *Lean Startup* e *Customer Development*. Por fim, a última fase, Implementação, finaliza o ciclo de desenvolvimento com testes, inserção de requisitos no *software* desenvolvido, somente após sua aprovação pelo cliente, e uma reunião de retrospectiva do *Scrum*.

Dentro da metodologia, além das atividades e artefatos de entrada e saída, foram definidos dois papéis, inspirados no *Scrum*, com base em perfis complementares: (i) equipe técnica, responsável pelo desenvolvimento do *software*; e (ii) negócios, representado pelo *Product Owner* (PO) da solução desenvolvida.

O PO é o principal responsável pelas fases iniciais da metodologia (Requisitos e Validação), identificando a ideia, definindo o MVP e descrevendo as histórias da solução a ser validada, definidas na seção 4.1. Já o time técnico, assume o papel principal nas duas últimas fases de desenvolvimento da metodologia: Projeto e Implementação.

Apesar da metodologia proposta possuir os papéis bem-definidos, assim como ocorre nas metodologias de desenvolvimento de *software* (seções 2.1 RUP e 2.2 SCRUM), a equipe toda deve trabalhar próxima. Desta forma, os envolvidos devem participar de todas as fases do *Startup Kaizen*, facilitando assim a comunicação interna. Na Figura 4.1 - Ciclo da Metodologia Startup Kaizen, é possível perceber os papéis responsáveis por cada fase do *Startup Kaizen*.

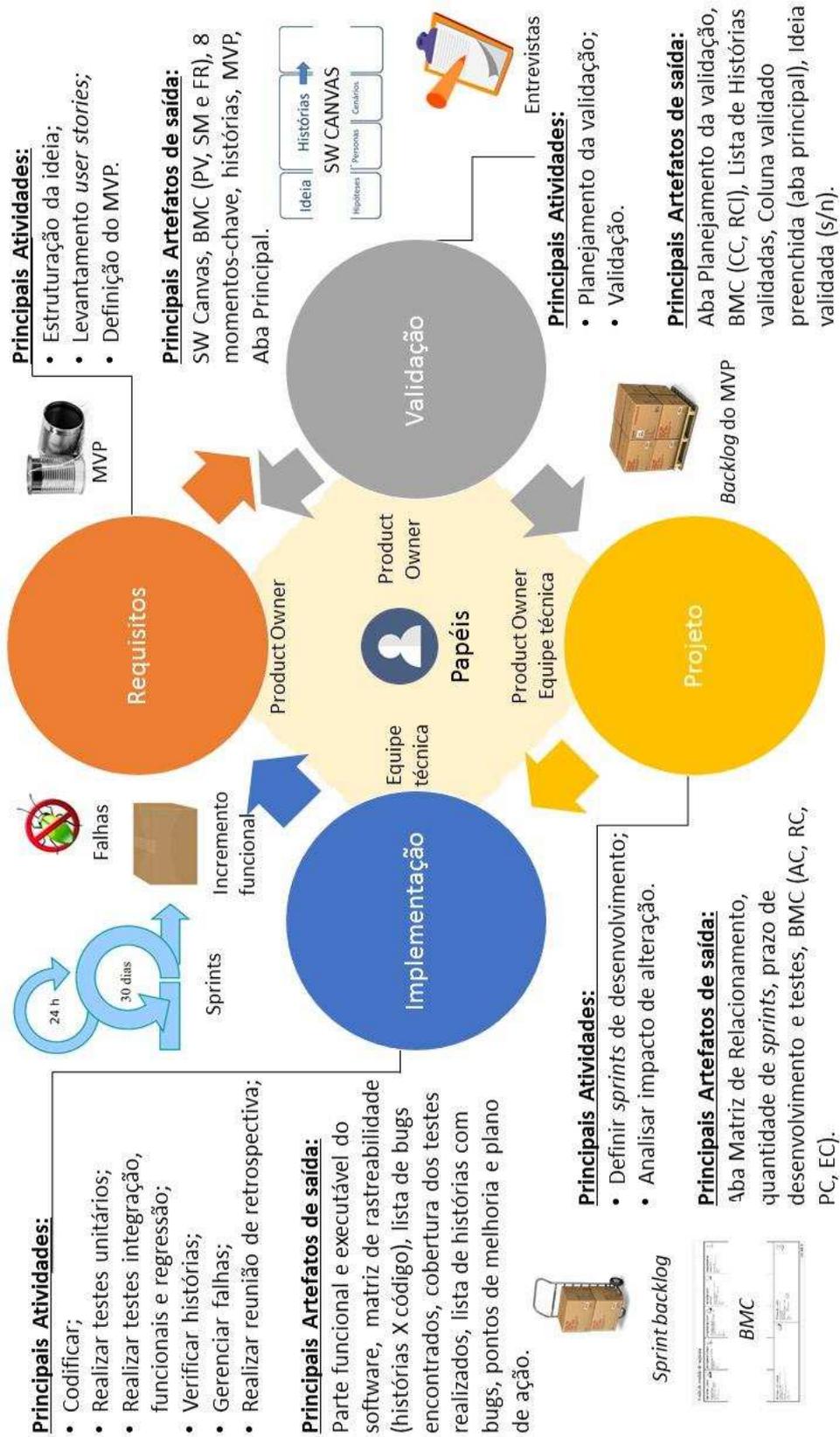


Figura 4.1 - Ciclo da Metodologia Startup Kaizen

O processo inicia-se pela fase de Requisitos, dando sequência para a fase de Validação, passando pela fase de Projeto, e por fim, fase de Implementação. As duas primeiras fases são focadas em estruturação da ideia para validação. Somente após a validação das histórias levantadas na fase de Requisitos, que é possível prosseguir para a próxima etapa de Projetos. Caso nenhuma história seja validada, é necessário retornar para a fase de Requisitos, conforme demonstrado na Figura 4.1 - Ciclo da Metodologia Startup Kaizen.

Após o ciclo completo, com a finalização da fase de Implementação e realização da Retrospectiva, retorna-se para a fase de Requisitos. Este ciclo deverá ser sempre repetido, fazendo com que a inserção de novas funcionalidades seja feita somente após a sua validação, mesmo após a descoberta de mercado.

As próximas seções descrevem detalhadamente cada fase do *Startup Kaizen*, bem como a origem de cada boa prática inserida na metodologia proposta.

4.1 FASE DE REQUISITOS

A fase de Requisitos consiste na análise e transformação da ideia com o objetivo de entendê-la melhor, e verificar sua viabilidade. Blank et al. (2012) descreve que todas as ideias são conjunto de hipóteses que devem ser validadas. Nesta fase, todas as atividades foram estruturadas visando a utilização de ferramentas que auxiliem ao empreendedor a preparar e definir hipóteses a serem validadas.

Durante essa preparação, o empreendedor assume o papel de P.O., devendo responder uma sequência de perguntas sobre a sua ideia, com o objetivo de mapear pontos essenciais ao modelo de negócios, tais como: proposta de valor, segmento de mercado, personas, fontes de renda e histórias. Essa sequência de perguntas é apresentada no APÊNCICE C – Aba 4: Requisitos.

Os artefatos de saída da fase de Requisitos são: três quadrantes do BMC preenchidos (proposta de valor, segmentos de mercado e fontes de renda), personas, cenários (chamados de 8 momentos-chave), histórias e o MVP, conforme demonstrado na Tabela 4.1 - Artefatos de entrada e saída da fase de Requisitos. Os artefatos de entrada para o início do ciclo são: Ideia de Negócio.

No decorrer do processo, após seguir para a fase de Validação, caso a ideia não seja validada, é necessário retornar para a fase de Requisitos. Desta forma, no segundo ciclo, os artefatos de entrada para a fase de Requisitos passam a ser: Ideia Não Validada e Lista de Histórias não Validadas, conforme descritos na Tabela 4.1.

Tabela 4.1 - Artefatos de entrada e saída da fase de Requisitos

REQUISITOS	
ARTEFATOS DE ENTRADA	ARTEFATOS DE SAÍDA
Ideia de negócio;	SW Canvas preenchido: Ideia, Cenários, Personas, Hipóteses, Histórias, MVP;
Ideia validada = NÃO;	Quadrantes do BMC: Proposta de Valor, Segmentos de Mercado, Fontes de renda;
Lista de histórias não validadas.	Abas 8 Momentos-chave, Personas, Classificação das Histórias preenchidas; Problema classificado; Aba principal colunas: Histórias, Descrição, Classificação, MVP.

Para facilitar esse processo, foi criada uma ferramenta para validação de requisitos de desenvolvimento do *software*. Essa ferramenta, chama-se SW Canvas, ilustrada na Figura 4.2. O SW Canvas (Figura 4.2) teve como inspiração o BMC (Figura 2.5), porém no lugar de aspectos de negócios, as informações inseridas devem ser utilizadas para validação de hipóteses relacionadas a histórias, cenários de utilização e personas.



Figura 4.2 - SW Canvas

O intuito do SW Canvas é auxiliar no mapeamento de todas as hipóteses relacionadas ao desenvolvimento de *software*, de forma visual e de fácil alteração. Assim como é proposto pelo BMC, ao utilizar o SW Canvas, o mesmo deve ficar visível para toda a equipe (recomenda-se colocá-lo na parede).

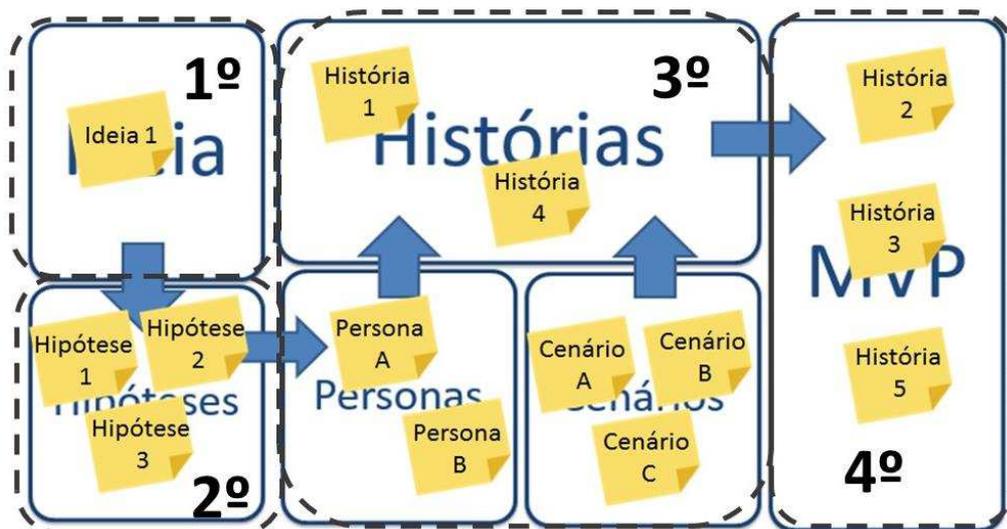


Figura 4.3 - Exemplo SW Canvas preenchido

No primeiro ciclo da metodologia, o SW Canvas deve ser todo preenchido com as hipóteses levantadas, além de cenários, histórias e personas possíveis para a solução proposta, conforme ilustrado na Figura 4.3. O quadrante de hipóteses deve ser preenchido com as

premissas que a equipe possui em relação a sua ideia. Neste caso seriam pontos que são considerados como verdadeiros pela equipe, por exemplo: pessoas pagariam por um sistema para organização de despesas pessoais.

Os demais quadrantes devem ser preenchidos com informações que serão validadas. Por exemplo, o quadrante Personas, deve ser preenchido com hipóteses de usuários (personas) que utilizarão a ferramenta desenvolvida. Enquanto no quadrante Cenário, devem ser descritas as situações nas quais as pessoas interagem com a solução proposta. Já no quadrante Histórias, são inseridas as histórias que deverão ser validadas com os clientes. Essas histórias consistem num combinado das personas com os cenários, seguindo o padrão BDD² para definição de histórias. Por fim, no quadrante MVP são inseridas as histórias que serão utilizadas para a construção do MVP. A escolha dessas histórias é feita com a utilização da matriz de complexidade *versus* impacto, apresentada posteriormente na seção 4.3.

Após a fase de validação, as informações que foram confirmadas devem permanecer, enquanto as outras devem ser removidas, veja Figura 4.4 e Figura 4.5. Todas as informações levantadas, derivam da ideia inicial, dando sequência a levantamento de hipóteses que gerarão as possíveis personas e cenários de utilização.

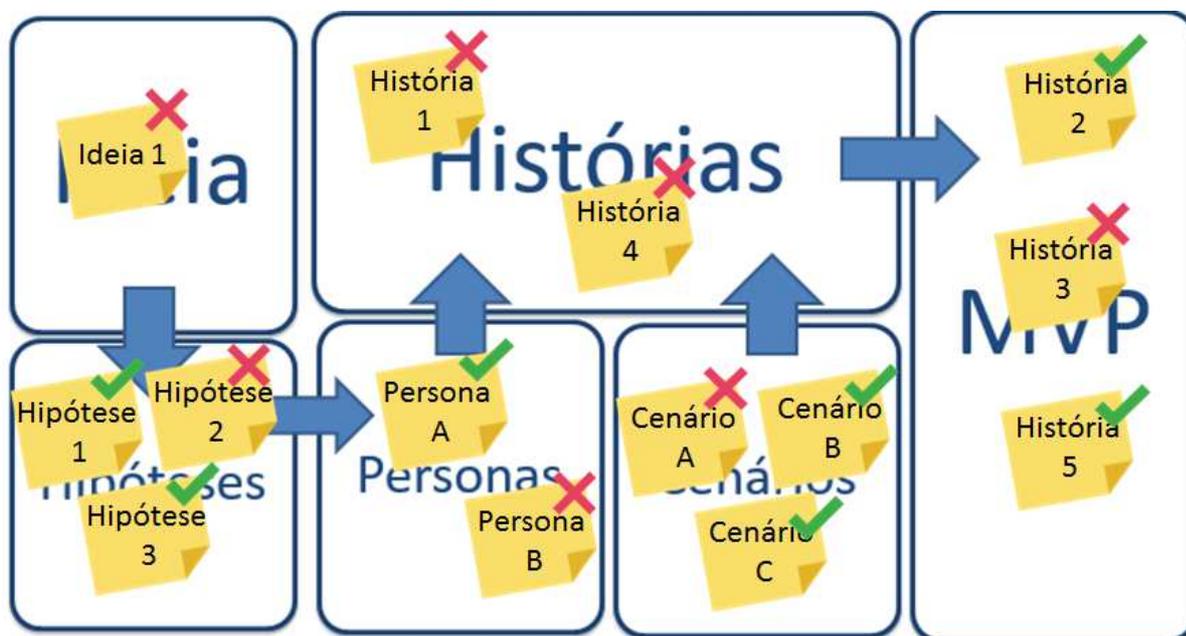


Figura 4.4 - SW Canvas hipóteses confirmadas e excluídas

² O *Behavior Driven Development* consiste numa metodologia de desenvolvimento ágil desenvolvida para facilitar a comunicação das equipes de projeto, através da aplicação da linguagem natural na descrição de métodos [NORTH, 2006].

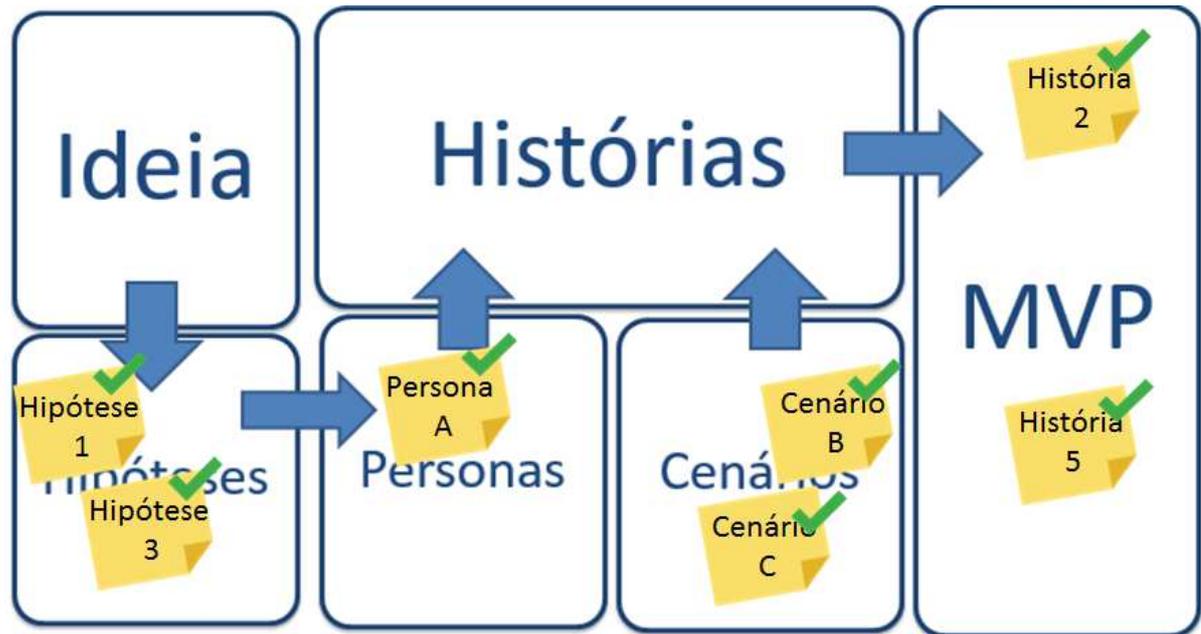


Figura 4.5 - SW Canvas atualizado após validação

Seguindo a sequência do SW Canvas, o primeiro passo consiste na descrição da ideia, seguida pela definição de hipóteses que darão sequência às personas, cenários, histórias e por fim, MVP, conforme representado na Figura 4.3. As hipóteses são descritas como questões que devem ser validadas com base na ideia. Por exemplo, para criação de um aplicativo para contato com taxista, é necessário validar primeiramente se as pessoas estão dispostas a utilizar um aplicativo para este tipo de atividade.

Após a definição das hipóteses, serão criadas as personas, conforme apresentado na Figura 4.6. Este quadro foi inspirado no quadro de personas do *Design Sprint* (Figura 2.7), conforme descrito na seção 2.5.1. A utilização das personas auxilia no detalhamento das informações de identidade, status, objetivos e habilidades pessoais visando um melhor entendimento e conhecimento dos potenciais usuários do sistema.

Design Sprint	
Identidade:	Nome, Idade, Sexo, Foto
Status:	primária, secundária, outro stakeholder
Hobbie:	O que essa persona gosta de fazer?
Atividades:	Qual é a especialidade da persona? Isso inclui educação, treinamento e competências específicas.
O que a persona gosta de fazer no dia-a-dia?	Em linhas gerais, quais as tarefas básicas ou críticas que a persona gosta de realizar? Qual é a frequência, importância e duração dessas tarefas?
Qual a característica que se destaca nessa persona?	Se fosse resumir essa persona em uma frase, como você
O que você destacaria sobre essa persona?	Qual seu próximo desafio? O que ela mais gosta? O que é um dia perfeito para ela?
Grandes Necessidades	Ela precisa de um jeito de _____ Para obter a experiência de _____ Porque ela valoriza _____



Figura 4.6 - Quadro de personas adaptado do quadro de personas do *Design Sprint*

Após criadas as personas, são definidos os cenários. Para criá-los de forma mais visual, é proposta a utilização dos 8 momentos-chave com base na metodologia *Design Sprint* do *Google* (descrito na seção 2.5.1). Esses momentos-chave devem considerar o ambiente ou contexto de utilização da solução proposta, detalhando situações que motivam ou explicam os objetivos, ações e reações dos atores nos cenários. Esses cenários podem ser descritos, ou representados através de desenhos, veja Figura 4.7 - 8 Momentos-Chave.

Cenário 1	Cenário 2	Cenário 3	Cenário 4
Cenário 5	Cenário 6	Cenário 7	Cenário 8

Figura 4.7 - 8 Momentos-Chave

Com as personas e cenários de utilização definidos, é possível seguir para o levantamento de histórias. Para sua criação, foi feita uma adaptação no modelo proposto por Cohn (2004). O modelo estabelecido por Cohn (2004) consiste numa descrição simples da história, envolvendo um ator, uma necessidade, ação e funcionalidade.

Para uma melhor aplicabilidade do padrão definido por Cohn (2004), foi necessário realizar uma adaptação do modelo proposto por ele. Essa adaptação tem como objetivo tornar as histórias criadas mais próximas da aplicação real e validação necessária. Para isso, foi feita a reutilização de artefatos criados anteriormente: personas e cenários. Desta forma, no *Startup Kaizen*, a história criada deve englobar as personas e cenários definidos no processo. Essa utilização deve ser feita conforme apresentado abaixo na Figura 4.8 - Padrão BDD para

criação de histórias. Este padrão foi tirado do modelo *Behaviour Driven Development*, uma metodologia de desenvolvimento de *software* focada no comportamento do usuário, através da utilização de linguagem não técnica para definição e criação de artefatos do processo. Ao utilizar o modelo proposto por Chon, as personas geradas no SK assumem o papel de ator, e o cenário estabelece a descrição de uso (ação) de uma funcionalidade.



Figura 4.8 - Padrão BDD para criação de histórias

Após a definição dessas histórias, elas são classificadas de acordo com dois aspectos: impacto *versus* dificuldade de implementação. Esse tipo de classificação é utilizado na metodologia *Design Sprint*, com o objetivo de classificar e filtrar ideias (veja seção 2.5.1).

Na metodologia proposta, essa classificação foi utilizada para a definição de histórias que irão compor o MVP, como pode ser visualizado na Figura 4.9 - Matriz Impacto no Cliente X Dificuldade de Implementação. Desta forma, as histórias que são classificadas com um impacto alto e baixa dificuldade de implementação são adicionadas ao MVP; já as que possuem baixo impacto e alta dificuldade de desenvolvimento são automaticamente descartadas. Essa matriz foi retirada do *Design Sprint* e descrita na seção 2.5.1

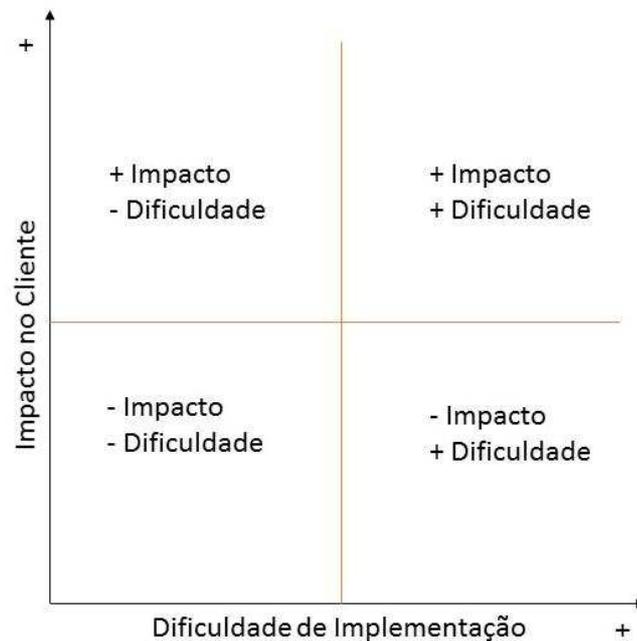


Figura 4.9 - Matriz Impacto no Cliente X Dificuldade de Implementação adaptado do *Design Sprint*

Os detalhes da fase de Requisitos estão representados no APÊNDICE B – Aba Requisitos. Nele é possível identificar quais são as atividades obrigatórias, artefatos e papéis dessa fase, bem como de quais metodologias essas atividades são originadas. Após definição das histórias que compõem o MVP, o empreendedor segue para a fase de validação.

4.2 FASE DE VALIDAÇÃO

Após a fase de Requisitos, a próxima fase a ser executada é a de Validação. O objetivo dessa fase consiste na validação diretamente com o cliente, visando obter *feedbacks* sobre as hipóteses e histórias levantadas, confirmando ou excluindo-as. Nessa fase, também é realizado todo o planejamento da validação através de técnicas baseadas em *Design Thinking* e planejamento de testes de *software* do RUP, conforme descrito, respectivamente, nas seções 2.5 e 2.

Guia de Entrevista
<p>Durante a entrevista, identifique os pontos a seguir:</p> <p>Qual a relação do cliente com tecnologia? Qual o seu conhecimento do domínio do produto? Qual seu conhecimento das tarefas que deverá realizar? Quais suas motivações e valores?</p> <p>Próprio usuário</p> <p><input checked="" type="checkbox"/> dados demográficos: idade, sexo, status socioeconômico; <input checked="" type="checkbox"/> educação: grau de instrução, área de formação, cursos realizados, alfabetismo.</p> <ul style="list-style-type: none"> ◦ O quão bem o usuário lê? ◦ Prefere aprender com outras pessoas? ◦ Prefere aprender fazendo? <p><input checked="" type="checkbox"/> idiomas e jargões:</p> <p>Que idiomas o usuário conhece e utiliza fluentemente? Possui: ◦ jargão profissional particular ◦ vocabulário próprio da empresa, da sua atividade ou de algum grupo social relevante para o seu projeto</p>

Figura 4.10 - Guia de Entrevista utilizada na Validação - Parte I adaptado de [BARBOSA; SILVA, 2010]

Guia de Entrevista
<p>Relação com a tecnologia</p> <p><input checked="" type="checkbox"/> experiência com computadores: alfabetismo computacional, habilidade com computadores, anos de experiência.</p> <ul style="list-style-type: none"> ◦ Que sistemas computacionais o usuário conhece? ◦ Quais deles costuma utilizar? ◦ Que hardware costuma utilizar? <p><input checked="" type="checkbox"/> experiência com um produto específico ou ferramentas semelhantes: experiência com produtos concorrentes e outros produtos específicos do domínio, hábitos de uso, preferências e descontentamentos</p> <p><input checked="" type="checkbox"/> tecnologia disponível: hardware (tamanho e resolução do monitor, velocidade do processamento etc.), software e outras ferramentas aos quais tem acesso</p> <p><input checked="" type="checkbox"/> é importante investigar:</p> <ul style="list-style-type: none"> ◦ Quem utilizará o sistema? ◦ Quem será afetado por ele? ◦ Quem é responsável por decidir quais objetivos o sistema deve apoiar e quais funcionalidades ele deve ter? ◦ Quem definiu os processos a serem apoiados pelo sistema?

Figura 4.11 - Guia de Entrevista utilizada na Validação - Parte II adaptado de [BARBOSA; SILVA, 2010]

Suas atividades principais foram desenvolvidas para obter um planejamento detalhado de como a validação será realizada. Desta forma, é necessário realizar um planejamento da validação que será executada juntamente com os potenciais usuários. Este planejamento é feito através da aba “Planejamento da Validação” (Apêndice C – Aba 13: Planejamento da Validação). Essa aba foi criada com base no artefato Plano de Testes ³do RUP.

Nesta aba, é necessário informar quais são pré-requisitos, ambientes, ferramentas, objetivos e salto de fé que são necessários para realizar a validação e confirmar, ou não, as hipóteses levantadas. Após a informação desses pré-requisitos, todas as histórias são listadas, juntamente com o resultado esperado da validação.

Depois do planejamento realizado, é feita a validação juntamente com um ou mais usuários que tenham as mesmas características listadas pela persona criada. A forma que essas histórias são ou não confirmadas, fica a critério da equipe de execução. Ela pode ser feita através de entrevistas diretas com os potenciais usuários ou o uso de outros métodos, tais como *card sorting* ⁴para a simulação da solução que será desenvolvida.

Durante a validação, as histórias devem atender ou não o resultado esperado. Caso elas atendam, a história está confirmada, portanto na coluna “Comprovado (s/n)” da aba “Planejamento da Validação” deve ser marcada como “Sim”. Caso contrário, ela não foi confirmada e não deve ser acrescentada no desenvolvimento do *software*, e a coluna “Comprovado” deve ser marcada como “Não”.

Desta forma, após a realização da validação, a equipe de projeto possuirá uma lista de histórias validadas, que serão implementadas. Além de terem em mãos também dados que auxiliem na tomada de decisão na priorização das histórias que serão desenvolvidas e também um dado quantitativo da validação realizada.

Os principais artefatos de saída esperados são: guia de entrevista (Figura 4.10 e Figura 4.11), guia para planejamento da validação (Figura 4.12 e Figura 4.13), dois quadrantes do BMC (relação com clientes e canais) e lista de histórias validadas. Veja Tabela 4.2 - Artefatos de entrada e de saída da fase de Validação.

Guia para planejamento da validação	
Pré-requisitos:	Quais são os pré-requisitos necessários para que a validação seja executada de forma correta, visando atingir seu objetivo?
Ambiente:	Qual ambiente será necessário para realizar os testes? Uma sala, laboratório, computador? Qual sistema operacional?
Ferramentas:	Quais ferramentas serão utilizadas para a validação: protótipo de papel, protótipo de baixa fidelidade, computador?
Objetivo da Validação:	Qual o objetivo da sua validação?
Salto de fé:	Qual o resultado esperado da sua validação? Defina em forma de números. Ex: monte um formulário para que o usuário diga o quão ele gostou da funcionalidade

Figura 4.12 - Guia para planejamento da Validação - Parte I adaptado do RUP

³ http://www.funpar.ufpr.br:8080/rup/webtmpl/templates/test/rup_tstpln.htm

⁴ A técnica de *Card Sorting* é uma técnica aplicada diretamente com o usuário, e utilizada para descobrir como ele classifica uma informação. Desta forma, o usuário recebe cartões e devem separá-los em grupos que consideram mais apropriados [MOULIN, 2011].

História	Resultado Esperado	Resultado obtido	Comprovado (s/n)	Comentários
Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]			--	
			--	
			--	
			--	
			--	

Figura 4.13 - Guia para planejamento da Validação - Parte II adaptado do RUP

Durante a fase de validação, o P.O. deve estabelecer o plano de validação direta com potenciais usuários, definindo informações que o ajudará a realiza-la de forma mais eficiente, e executá-lo. Nesse plano, as informações: pré-requisitos, ambiente, ferramentas, objetivos e o salto de fé devem ser preenchidos.

Tabela 4.2 - Artefatos de entrada e de saída da fase de Validação

VALIDAÇÃO	
ARTEFATOS DE ENTRADA	ARTEFATOS DE SAÍDA
SW Canvas preenchido: Ideia, Cenários, Personas, Hipóteses, Histórias, MVP;	Aba Planejamento da validação;
Quadrantes do BMC preenchidos: Proposta de Valor, Segmentos de Mercado, Fontes de renda;	Quadrantes do BMC preenchidos: Canais de Comunicação, Relações com o cliente;
Abas 8 Momentos-chave, Personas, Classificação das Histórias preenchidas;	Lista de histórias validadas;
Problema classificado;	Aba principal coluna validado preenchida;
Aba principal colunas: Histórias, Descrição, Classificação, MVP.	Ideia validada = SIM/Não.

Os pré-requisitos são todas as informações e condições necessárias para a realização da validação, tais como: agendamento e reserva de um horário para a entrevista. O ambiente e as ferramentas, formam o conjunto de definição de onde essa validação ocorrerá, através das informações que façam com que o P.O. consiga simular uma situação semelhante aos cenários de uso, ou até mesmo aplica-las num cenário real de uso.

Os objetivos e salto de fé possuem um papel fundamental na fase de validação. São eles que guiarão os empreendedores para que eles possam, no final, dizer que a ideia foi ou não validada. O objetivo deve ser expresso de forma mais abrangente, e sempre relacionado às hipóteses levantadas na etapa de requisitos (seção 2.3). O objetivo é utilizado para validar a hipótese principal de uma ideia, verificando se ela foi comprovada ou não.

Após a definição dos objetivos, é necessário escrever o salto de fé. Ele consiste no resultado esperado da validação, definido de forma mais específica através de números e métricas. Ao final da validação, deve ser feita uma análise em relação ao objetivo e salto de fé, verificando se os resultados obtidos foram ou não alcançados, ou seja, validados.

Ao final da realização dessa fase, caso nenhuma história e/ou os objetivos e salto de fé não sejam validados, o ciclo obriga o retorno para a fase de requisitos novamente. Desta forma, o P.O. deverá analisar e refazer todas as atividades referentes a fase de requisitos, podendo alterar apenas alguns pontos definidos, ou até todos eles.

Esse tipo de mudança após a validação, é conhecido como pivotar. Quando um ato de pivotar é realizado, são analisados todos os aspectos relacionados ao desenvolvimento da ideia, com o objetivo de executar mudanças para uma nova fase de validação (descrito na seção 2.3). Essas mudanças podem ser baseadas em apenas alguns aspectos do negócio: proposta de valor, segmento de mercado, fontes de renda, MVP, etc., ou em todos os seus aspectos, mudando inclusive a ideia a ser trabalhada.

Os detalhes da fase de Validação estão representados no APÊNDICE B – Aba Validação. Nele é possível identificar quais são as atividades obrigatórias, artefatos e papéis dessa fase, bem como de quais metodologias essas atividades são originadas. Caso alguma história for confirmada durante a fase de validação, deve-se seguir para a fase de projeto.

4.3 FASE DE PROJETO

Na fase de projeto, é feito o planejamento de desenvolvimento com os resultados obtidos na validação. Nesta fase, as histórias confirmadas serão priorizadas, divididas em *sprints* de entregas, além de mapeadas conforme impacto gerado, com a criação de uma matriz de relacionamento.

		IMPACTADOS																		
		História 1	História 2	História 3	História 4	História 5														
IMPACTA	História 1	x		x																
	História 2		x			x														
	História 3	x		x																
	História 4				x															
	História 5		x			x														

Figura 4.14 - Matriz de Relacionamento

Para o planejamento da fase do projeto, é necessário primeiro definir o tamanho das *sprints* de desenvolvimento. Essa definição de tamanho deve ser feita conforme a necessidade de cada projeto. Porém, para projetos iniciantes e com grande volume de atualizações, recomenda-se que o tamanho dela seja, no máximo de duas semanas.

Desta forma, após definir o tamanho da *sprint*, é necessário entender quais histórias entrarão em cada uma. Para isso, recomenda-se a técnica de reunião de planejamento (ou *planning*) utilizada no *Scrum* detalhado na seção 2.2.

A matriz de relacionamento (Figura 4.14 - Matriz de Relacionamento) deve ser preenchida para auxiliar na identificação de pontos de atenção quando houver alguma alteração no *software*. Além da matriz, os demais artefatos gerados por esta fase são: quantidade e tempo de cada *sprint*, prazos e quatro quadrantes do BMC (conforme descrito na seção 2.4): atividades, recursos e parcerias chave, estrutura de custos. Todos os artefatos de entrada e saída estão listados na Tabela 4.3 - Artefatos de entrada e saída da fase de Projeto.

Tabela 4.3 - Artefatos de entrada e saída da fase de Projeto

PROJETO	
ARTEFATOS DE ENTRADA	ARTEFATOS DE SAÍDA
Lista de Histórias validadas; Ideia validada = SIM;	Aba Matriz de Relacionamento;
	Quantidade de <i>sprints</i> que serão realizadas;
	Prazo para finalização do desenvolvimento e testes;
	Quadrantes do BMC: atividades, recursos e parcerias chave, estrutura de custos.

Para a realização dessa fase, recomenda-se a utilização de ferramentas para gerenciamento de projeto de *software* já existentes no mercado, tais como: *GitHub*⁵, *Visual Studio*⁶, *Jira*⁷, dentre outras. A escolha da ferramenta deve ser feita de acordo com as preferências de cada equipe.

Os detalhes da fase de Projeto estão representados no APÊNDICE B – Aba Projeto. Nele é possível identificar quais são as atividades obrigatórias, artefatos e papéis dessa fase, bem como de quais metodologias essas atividades são originadas. Após o planejamento, é realizada a fase de Implementação.

4.4 FASE DE IMPLEMENTAÇÃO

Nesta fase é feito o desenvolvimento e testes, visando obter um incremento funcional do *software* e com uma qualidade mínima. Suas principais atividades são: desenvolvimento do *software*, realização de testes (unitários, integração, funcionais e regressão), registro e gerenciamento de falhas e retrospectiva das *sprints* realizadas.

A escolha de ferramentas para realização de testes unitários, integração fica a critério da equipe do projeto, de acordo com as necessidades técnicas do desenvolvimento (linguagem de programação, *framework* de desenvolvimento, etc.). A realização de testes funcionais e de regressão deve ser feita seguindo o roteiro de testes criado na aba “Principal” com as histórias definidas para o MVP.

Tabela 4.4 - Artefatos de entrada e de saída da fase de Implementação

IMPLEMENTAÇÃO	
ARTEFATOS DE ENTRADA	ARTEFATOS DE SAÍDA
Aba Matriz de Relacionamento;	Parte funcional e executável do <i>software</i> ;
Quantidade de <i>sprints</i> que serão realizadas;	Matriz de rastreabilidade (histórias x código);
Prazo para finalização do desenvolvimento e testes;	Lista e quantidade de bugs encontrados;
Quadrantes do BMC: atividades, recursos e parcerias chave, estrutura de custos.	

Durante os testes, recomenda-se que todas as histórias levantadas sejam testadas, garantindo uma qualidade mínima do produto. Essa qualidade mínima dá-se através da confirmação do fluxo-principal do *software* (Figura 4.16 - Planilha aba Principal - parte I), passando pelo caminho básico do desenvolvimento. Ao testar determinada história, seu status “Testado (s/n)” deve ser alterado para “Sim”.

⁵ <https://github.com>

⁶ <https://www.visualstudio.com/>

⁷ <https://br.atlassian.com/software/jira>

O status “Passou (s/n)” deve ser alterado para “Sim” no caso da história ser executada sem nenhuma falha encontrada, e “Não” caso contrário (Figura 4.17 - Planilha aba Principal - parte II). Quando uma falha é encontrada, ela deve ser listada na coluna “Bugs” com uma breve descrição e um número para identificá-la. Esse registro deve ser feito todas as vezes que uma falha for encontrada, sem exceções.

Para realização do teste de Regressão, recomenda-se a verificação das falhas corrigidas e histórias impactadas com a correção. Essa análise deve ser feita através da utilização da matriz de relacionamento (Figura 4.14 - Matriz de Relacionamento), identificando quais histórias impactam ou são impactadas umas pelas outras.

Nessa matriz, nas colunas são colocadas as histórias que são impactadas, e as linhas as histórias que impactam. Dessa forma, a leitura que deve ser feita na matriz é a seguinte: História 1 impacta Histórias 1 e 3, História 2 impacta Histórias 2 e 5, conforme exemplo na Figura 4.14. Além disso, vale ressaltar que uma história sempre será impactada por ela mesma, ou seja, qualquer alteração que ela sofrer, ela será impactada.

Quando os testes são finalizados, na aba “Principal”, são monitoradas algumas métricas de qualidade de *software*: Porcentagem de histórias validadas (número de histórias validadas/quantidade total de histórias), Cobertura de Testes (número de histórias testadas/quantidade total de histórias), Índice de Qualidade (número de histórias que passaram sem nenhuma falha/número total de histórias).

O Índice de Qualidade é uma métrica muito importante para acompanhamento durante a evolução nos demais ciclos de desenvolvimento. Com ele, é possível medir a relação da qualidade entregue para o cliente e a sua satisfação em relação a ferramenta.

Os artefatos de saída esperados para essa fase são: incremento funcional e executável, versão do *software* liberado, rastreabilidade através do mapeamento de histórias, código, falhas encontradas, cobertura e resultado dos testes executados, lista de funcionalidades impactadas com alterações, pontos de melhoria e plano de ação. Muitos desses artefatos podem ser derivados automaticamente pelas ferramentas sugeridas, porém foi criada também uma aba “Principal” para mapeamento de todos os pontos trabalhados durante todo o ciclo de desenvolvimento (Figura 4.15, Figura 4.16 e Figura 4.17).

Porcentagem Histórias Validadas	0%	Quantidade de Histórias do MVP	0%
Cobertura de Testes	0%	Índice de qualidade	0%

Figura 4.15 - Métricas mapeadas na planilha aba Principal

#	História	Descrição	Classificação	Prioridade	MVP
1	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				
2	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				
3	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				

Figura 4.16 - Planilha aba Principal - parte I

Validado (s/n)?	Sprint	Módulo	Testado (s/n)?	Passou (s/n)?	Bugs
--			Sim	--	- Bug 1 - Bug 2 - Bug 3
--			Sim	--	
--			Sim	--	
--			Sim	--	
--			--	--	

Figura 4.17 - Planilha aba Principal - parte II

A fase de implementação finaliza somente após a realização da retrospectiva da *sprint* realizada. Essa retrospectiva deve ser realizada da mesma forma que acontece no *Scrum*, no qual são levantados todos os pontos positivos e negativos durante o desenvolvimento do projeto. Na seção 2.2 é detalhada a reunião de retrospectiva.

Ao final de todo o ciclo de desenvolvimento, passando por cada fase: requisitos, validação, projeto e implementação, como resultado espera-se obter um incremento funcional do *software*. Desta forma, o *software* vai adquirindo novas funcionalidades, crescendo de forma incremental a partir do MVP definido.

Além disso, a metodologia proposta visa um desenvolvimento organizado e estruturado, fornecendo informações que auxiliem o empreendedor na tomada de decisões através de métricas de desenvolvimento. Essas informações impactam diretamente a qualidade final do *software*, já que é possível entender o ponta-a-ponta do ciclo de desenvolvimento, desde o levantamento de requisitos até o gerenciamento de falhas.

Os detalhes da fase de Implementação estão representados no APÊNDICE B – Aba Implementação. Nele é possível identificar quais são as atividades obrigatórias, artefatos e papéis dessa fase, bem como de quais metodologias essas atividades são originadas.

4.5 CONCLUSÃO

Neste capítulo foi apresentada a metodologia *Startup Kaizen*, com todas as suas fases, atividades, artefatos e papéis detalhados. A metodologia é composta por quatro fases diferentes: Requisitos, Validação, Projeto e Implementação e cada uma delas foi descrita apresentando todos os seus artefatos de entrada e de saída.

O *Startup Kaizen* tem como principal benefício a união de boas práticas ligadas à Engenharia de *Software*: rastreabilidade de requisitos, gerenciamento de requisitos e falhas, planejamento e execução de testes, dentre outras. Essas boas práticas foram unidas às boas práticas defendidas por metodologias utilizadas por *startups*, tais como: modelagem de negócios, utilização de MVP, ciclos de desenvolvimento curtos e rápidos, constante validação com o cliente, fácil adaptação e agilidade. Através das atividades empregadas no SK, é possível estruturar um projeto inicial, através da definição de requisitos que, posteriormente são validados diretamente com potenciais clientes.

Com a sua utilização, é possível obter um desenvolvimento ágil, estruturado e com constante interação com seus usuários. Além disso, diferente dos trabalhos relacionados estudados, os requisitos de *software* são implementados somente após aprovação realizada através de uma validação estruturada.

As duas primeiras fases do *Startup Kaizen* auxiliam na obtenção de um projeto estruturado, com a possibilidade de utilização de métricas de qualidade para monitoramento

da evolução da solução desenvolvida. Através da estruturação das fases de Requisitos e Validação, é possível realizar uma análise quantitativa profunda dos resultados esperados e obtidos antes da criação de qualquer linha de código. Esse tipo de validação faz com que haja um menor desperdício de tempo no desenvolvimento de funcionalidades que nenhum cliente irá utilizar, diminuindo assim os desperdícios ocasionados desenvolvimento de algo inútil.

Além disso, as duas primeiras fases do SK consistem nos pilares fundamentais para dar a continuidade ao processo de desenvolvimento. São essas fases que possibilitam todo o mapeamento através da rastreabilidade de requisitos, bem como gerenciamento de falhas e controle de mudanças. Essas boas práticas possibilitam a realização de melhoria contínua no desenvolvimento do *software*, e também de aperfeiçoamento do processo de desenvolvimento.

A rastreabilidade dos requisitos de *software* possibilita uma redução no desperdício de tempo e esforço para análises de impacto provocadas por alterações realizadas no *software*. Já o gerenciamento de falhas e controle de mudanças favorecem um maior conhecimento do impacto de alterações realizadas, bem como quais falhas são mais impactantes, e até na tomada de decisão em relação a liberação do *software* para o cliente, através da análise da quantidade de falhas encontradas *vs* resolvidas.

Desta forma, diferente das metodologias estudadas, o *Startup Kaizen* foi criado com o objetivo de prover mais informações e um maior controle no desenvolvimento de *software* ágil e sob condições de extrema incerteza. Esse maior controle auxilia a tomada de decisão, fazendo com que os riscos, e até o desperdício de desenvolvimento sejam menores.

5 VALIDAÇÃO DA METODOLOGIA: UM ESTUDO DE CASO

Para analisar o impacto da utilização da metodologia em *startups* foi realizado um estudo de caso. Nesse estudo, foram selecionados grupos de trabalho na disciplina de Engenharia de *Software* do Programa de Pós-graduação em Ciência da Computação na UFSCar – Sorocaba.

Durante o período de experimentação, as *startups* selecionadas utilizaram a metodologia e também tiveram um acompanhamento durante os 4 meses de trabalho. Para medição do resultado, foram definidas perguntas de pesquisa para análise dos artefatos gerados durante a utilização da metodologia.

Neste capítulo, está descrito o planejamento do estudo de caso realizado, bem como suas premissas e resultados obtidos. Ao final do capítulo, as perguntas levantadas no estudo foram analisadas, de forma a obter um entendimento do impacto da inserção de boas práticas de Engenharia de *Software* em *startups* em estágio inicial (descoberta de mercado).

5.1 DEFINIÇÃO DOS OBJETIVOS

5.1.1 Objetivo Global

Definir uma metodologia de desenvolvimento de *software* que englobe boas práticas de metodologias utilizadas por *startups* para descoberta de mercado, e boas práticas de engenharia de *software* visando o aumento da qualidade final do produto.

5.1.2 Objetivo do Estudo

Analisar a aplicação de técnicas de Engenharia de *Software* em metodologias de desenvolvimento de *startups*;

Com o propósito de verificar o impacto da inserção de técnicas de Engenharia de *Software* em *startups* em estágio inicial;

Com respeito à falta de utilização de ferramentas de apoio para desenvolvimento de *software* e descoberta de mercado em empresas iniciantes;

Do ponto de vista do empreendedor e desenvolvedor de *software*;

No contexto de alunos de mestrado da disciplina de Engenharia de *Software* da UFSCar em Sorocaba.

5.1.3 Questões

Para a análise dos resultados obtidos, foram levantadas hipóteses e perguntas que foram investigadas durante o processo de aplicação do estudo de caso. Essas perguntas foram definidas visando responder à pergunta principal desse estudo de caso: a inserção de boas práticas de Engenharia de *Software* pode favorecer uma *startup* durante a fase de validação?

Para definição dessas perguntas foi utilizado o método GQM (*Goal Question Metrics*). O GQM consiste num método criado para auxiliar no planejamento, definição de objetivos e avaliação de um processo e/ou experimento. O método requer a elaboração de um objetivo de avaliação, seguido por questões que auxiliem na análise do objetivo estipulado.

Por fim, a partir das questões, são levantadas métricas para medição dos resultados definidos [SOUSA et al., 2003].

Desta forma, com base no objetivo inicial: avaliar o impacto de inserção de boas práticas de Engenharia de *Software* durante o ciclo de desenvolvimento de uma *startup* na fase de validação, foram levantadas algumas perguntas para serem analisadas durante o experimento. As seguintes perguntas foram levantadas:

Q1: A utilização da metodologia proporcionará um desenvolvimento do software mais estruturado?

Q2: A utilização da metodologia proporcionará métricas para tomada de decisão na liberação do software para os usuários?

Q3: A utilização da metodologia influenciará na qualidade final do software desenvolvido?

Após definidas as questões de avaliação, foram definidas métricas para análise. Além dessas métricas, outros pontos também foram levados em consideração, conforme descrito na seção Procedimento de Análise. A Figura ilustra o GQM criado para análise do estudo de caso.

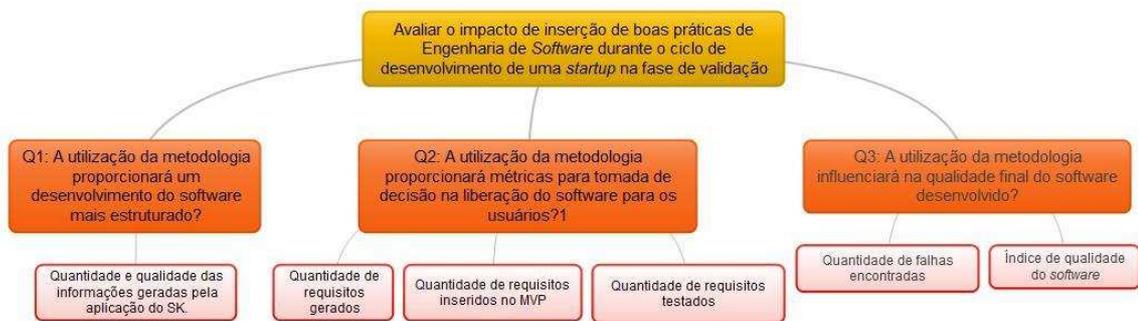


Figura 5.1 - GQM criado para análise dos resultados do estudo de caso

5.2 PLANEJAMENTO

5.2.1 Descrição da Instrumentação

Para a realização da instrumentação, fez-se necessário um treinamento com as equipes que irão utilizar a metodologia proposta. Esse treinamento consistiu em quatro horas, passando por todas as fases e ferramentas utilizadas na metodologia.

O período de acompanhamento foi de quatro meses, dos quais foram determinadas entregas conforme a evolução de cada equipe. Essas entregas foram divididas da seguinte maneira:

Tabela 5.1 - Entregas da metodologia

Entrega 1	<p>Aba Requisitos: descrição da ideia, problema que ela resolve, proposta de valor, classificação do problema, possíveis clientes, personas, 8 momentos-chave, fontes de renda, histórias iniciais, classificação das histórias (impacto X complexidade), salto de fé, histórias selecionadas para MVP;</p> <p>BMC: segmentos de mercado, proposta de valor e fontes de renda;</p> <p>Aba 8 momentos-chave</p> <p>Aba Principal: História, descrição, classificação, prioridade, MVP;</p> <p>Aba Personas, Aba SW Canvas.</p>
Entrega 2	Aba Validação: planejamento da validação, BMC: Relações com clientes e canais.

Entrega 3	Aba Validação: Lista de histórias validadas com o cliente; Aba Projeto: Matriz de relacionamento; Aba Principal: <i>Backlog</i> de produto: histórias MVP (mesmas histórias validadas na aba validação), classificação de histórias MVP, prioridade histórias MVP, <i>sprints</i> , histórias x <i>sprints</i> .
Entrega 4	MVP implementado; Aba Principal: módulo do <i>software</i> , testado (s/n), passou (s/n), lista de <i>bugs</i> .
Entrega 5	Revisar Aba Requisitos: histórias, cenários, personas; Revisar BMC: proposta de valor, segmentos de mercado, fonte de renda.
Entrega 6	Aba Projeto: matriz de relacionamento; Aba Principal: <i>backlog</i> do produto, classificação histórias MVP, prioridade histórias MVP, <i>sprints</i> , histórias x <i>sprints</i> .
Entrega 7	MVP atualizado; Aba Principal: módulo de <i>software</i> , testado (s/n), passou (s/n), lista de <i>bugs</i> .

O acompanhamento foi feito através de reuniões *online* para esclarecimento de dúvidas, e presenciais para apresentação dos entregáveis de cada etapa definidos acima.

Para obtenção dos dados comparativos, foram estipulados dois critérios: (i) índice de qualidade do *software* testado e (ii) índice de impacto das falhas encontradas nos clientes. Para o primeiro critério, uma equipe deveria testar o *software* de outra equipe. Cada grupo enviou uma lista de requisitos para outra equipe testá-la. A definição dos grupos que iriam testar foi feita através de sorteio, e ficou de acordo com a Tabela 5.2 - Grupos responsáveis pelos testes.

Tabela 5.2 - Grupos responsáveis pelos testes

Grupo	Testado por
<i>Caregiver</i>	Impressão Digital
<i>Dover</i>	<i>Caregiver</i>
<i>Factoring Sys</i>	<i>Dover</i>
Impressão Digital	<i>Factoring Sys</i>

O segundo critério, índice de impacto das falhas encontradas nos clientes, foi obtido através de um formulário com perguntas sobre avaliação do *software*. Essas perguntas possuem uma escala de 1 a 10, onde 1 era a nota menor e 10 a maior.

As perguntas feitas envolviam tantos aspectos de atendimento às necessidades dos clientes, quanto a qualidade percebida pelos usuários através de problemas e falhas encontradas durante sua utilização.

5.2.2 Seleção do Contexto

A instrumentação foi realizada durante a disciplina de Engenharia de *Software* no Mestrado em Ciências da Computação na UFSCar em Sorocaba. Optou-se por realiza-la com alunos que possuem um conhecimento na área de computação, que já possuem uma familiaridade com desenvolvimento de *software* e metodologias.

5.2.3 Seleção dos Indivíduos

Para seleção dos indivíduos, a turma foi dividida em quatro grupos, dos quais foram sorteados dois grupos para a utilização da metodologia, e dois que não a utilizaram.

Dentro dos grupos, existiam participantes que, além da disciplina de engenharia de *software*, cursavam também a disciplina de IHC (Interface Humano-Computador). Sendo assim, dos grupos selecionados, um cursava as duas disciplinas, enquanto o outro grupo selecionado cursava apenas engenharia de *software*. O mesmo critério foi definido para os grupos que não utilizaram a metodologia. Veja os grupos de trabalho na Tabela 5.3 - Definição dos Grupos que utilizaram a Metodologia.

Tabela 5.3 - Definição dos Grupos que utilizaram a Metodologia

Nome do Grupo	Descrição	Metodologia (s/n)
<i>Caregiver</i>	Ferramenta para lar de idosos, utilizada para controle de estoque de remédios e lembrete.	Sim
<i>Dover</i>	Aplicativo para realização de frete por pessoas físicas.	Sim
Impressão Distribuída	Ferramenta para compra de impressão através de gráficas.	Não
<i>Factoring Sys</i>	Ferramenta para negociação de <i>factoring</i> entre empresas.	Não

Para definição do perfil do participante foi criado um formulário, com 8 perguntas, sendo as 4 primeiras para identificação do participante, seguidas por 4 perguntas para entendimento do perfil. Para finalizar, o formulário possuía uma tabela com as metodologias, técnicas e ferramentas utilizadas na metodologia. Nessa tabela, o participante deveria classificar cada metodologia seguindo os seguintes critérios: “não conhecia”, “possuía conhecimento teórico apenas”, “possuía conhecimento prático apenas”, “possuía conhecimento teórico e prático” e “já havia aplicado o conhecimento em projetos no mercado de trabalho”. O formulário completo pode ser visualizado no Apêndice D.

Todos os participantes das equipes responderam o formulário, com um total de 11 respostas. Dessas 11 respostas, somente uma pessoa não atua e nunca atuou no mercado de TI. Dentre os demais, 82% diz ter atuado e/ou atuar na área de desenvolvimento e sistemas, enquanto 55% atua e/ou atuou na área de infraestrutura. Somente 36% atuou na área de processos, e apenas um participante atuou na área de qualidade de *software*, conforme ilustrado na Figura 5.2.

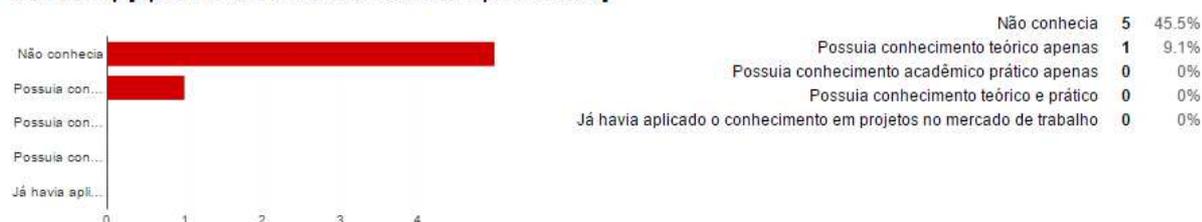


Figura 5.2 - Atuação no mercado de TI

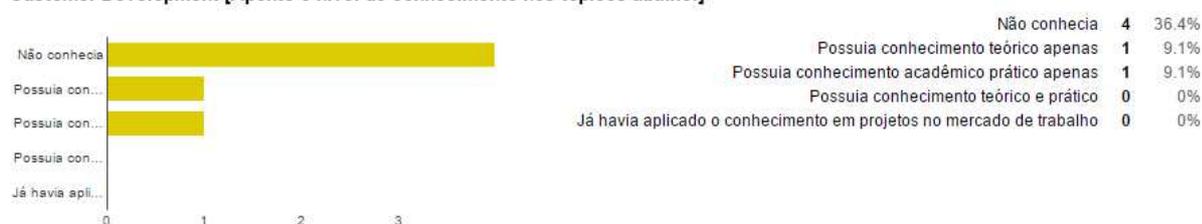
A maioria dos participantes possui um tempo de atuação maior do que 5 anos (63%), além de estarem atuando e realizando uma pós-graduação ao mesmo tempo. Essas informações mostram que a grande maioria das pessoas possui uma experiência no mercado de TI, em diversas áreas de conhecimento.

Em relação as metodologias utilizadas como base para este estudo: *Lean Startup* (5 pessoas – 45%), *Customer Development* (4 pessoas – 36%), *Design Thinking* (5 pessoas – 45%) e BMC (5 pessoas – 45%), a maioria não possui conhecimento prático. Apenas uma pessoa possuía conhecimento teórico e prático no BMC, as demais possuíam somente conhecimento teórico nas demais metodologias. As figuras Figura 5.3 e Figura 5.4 mostram o resultado das respostas obtidas com o formulário aplicado.

Lean Startup [Aponte o nível de conhecimento nos tópicos abaixo:]



Customer Development [Aponte o nível de conhecimento nos tópicos abaixo:]



Business Model Canvas [Aponte o nível de conhecimento nos tópicos abaixo:]

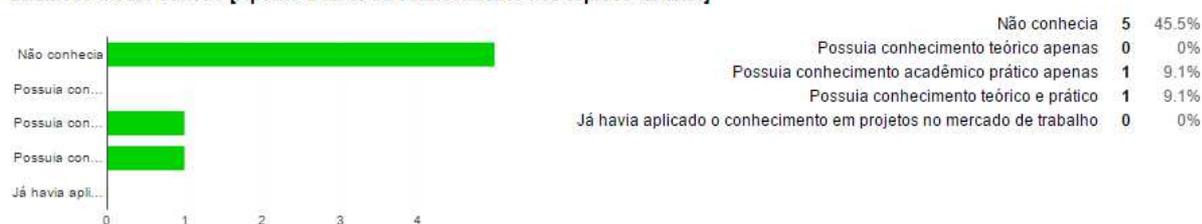


Figura 5.3 - Resultado das respostas dadas pelos participantes sobre as metodologias estudadas

Design Thinking [Aponte o nível de conhecimento nos tópicos abaixo:]

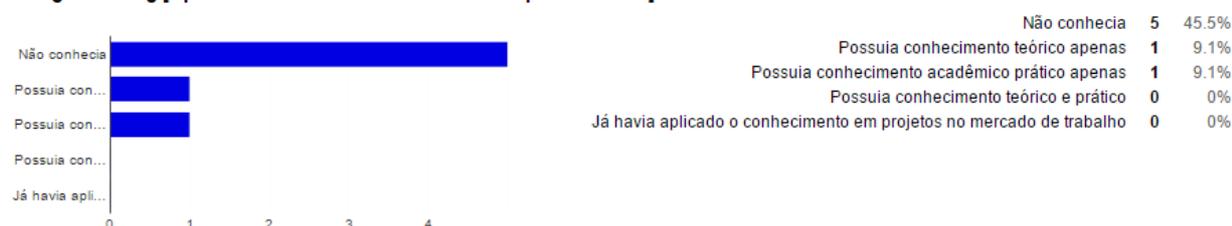


Figura 5.4 - Resultado das respostas dadas pelos participantes sobre o Design Thinking

Com base nos dados coletados sobre o perfil dos participantes, foi possível perceber que a grande maioria atua e/ou atuou no mercado de trabalho como desenvolvedor, poucos possuíam o perfil de processos e qualidade.

Além disso, a maioria dos participantes não conhecia as metodologias utilizadas por *startups* para validação da ideia e descoberta de mercado. É importante ressaltar também, que nenhum membro das equipes já havia aplicado as metodologias.

5.2.4 Procedimento de Análise

Segundo Boente et al. (2008), a qualidade de *software* não deve ser medida de forma isolada. A sua obtenção ocorre a partir do momento que as definições realizadas no projeto de *software*, ou seja no seu planejamento, são realizadas até a conclusão do projeto [BOENTE; OLIVEIRA; ALVES, 2008].

Além disso, para alcançar uma qualidade maior, é necessário realizar um conjunto de atividades planejadas e sistemáticas que forneçam dados o suficiente afim de fornecer maiores informações sobre o projeto desenvolvido. Essas informações servem como base para uma maior compreensão e confiança de que a qualidade do produto satisfaz o planejamento definido [BOENTE; OLIVEIRA; ALVES, 2008]. Desta forma, não é possível afirmar que um *software* possui uma qualidade maior que outro tomando somente como base somente a quantidade de *bugs* encontrados durante seus testes.

Portanto, para afirmar quais grupos tiveram melhor resultado em relação à qualidade final do *software*, é necessário fazer uma análise qualitativa de todo o processo de desenvolvimento. Essa análise faz-se necessária já que as equipes que não utilizaram o *Startup Kaizen*, não realizaram um planejamento prévio do que seria desenvolvido e entregue ao usuário final.

A obtenção dos resultados e análise foi feita por uma pessoa, durante a aplicação e acompanhamento do estudo de caso. A cada fase os artefatos gerados foram analisados e comentados para as equipes que utilizaram o *Startup Kaizen*.

O processo de análise durante a execução do estudo de caso foi feito de forma comparativa das apresentações realizadas por todas as equipes que cursavam a disciplina de Engenharia de *Software*. Ao final do processo de aplicação do estudo de caso, foi recolhido um artefato enviado por todas as quatro equipes que cursavam a disciplina. Este artefato consiste numa planilha de testes, conforme apresentada na **Figura 5.5 - Planilha de Testes**.

Esta planilha, foi criada com base na planilha “Principal” do SK, porém com foco em testes. Portanto, suas colunas eram compostas por: ID (número de identificação), História, Descrição, Testado (S/N), Passou (S/N), Falhas Encontradas (**Figura 5.5 - Planilha de Testes**).

#	Histórias ou Requisitos	Descrição	Testado (s/n)?	Passou (s/n)?	Bugs
1			-	-	
2			-	-	
3			-	-	
4			-	-	
5			-	-	
6			-	-	

Figura 5.5 - Planilha de Testes

Esses dados foram utilizados como base de análise dos resultados da aplicação do *Startup Kaizen*. Alguns pontos foram levados em consideração durante a análise, são eles:

- (1) Formato e padronização das informações obtidas;
- (2) Quantidade de informações fornecidas;
- (3) Resultado dos testes realizados (número de falhas encontradas e cobertura de testes).

5.2.5 Validade do Procedimento

O experimento foi realizado dentro de um ambiente controlado num contexto acadêmico. Para a obtenção de resultados mais realistas, faz-se necessário a aplicação em *startups* que realmente estão desenvolvendo novos mercados.

O fato desse experimento ter rodado dentro de um ambiente acadêmico influencia nos resultados, já que os participantes estão inseridos num contexto de aprendizado. Esse tipo de contexto não é necessariamente verdadeiro para os criadores de uma nova *startup*. Além disso, os alunos possuem apenas interesse acadêmico relacionado as *startups*, podendo ou não focar na criação de uma nova empresa, gerando um novo mercado de atuação.

Durante a realização desse estudo de caso, não foi requerido que os participantes iniciassem processos de venda diretamente com seus clientes. O fato de ser um trabalho acadêmico, influencia a forma com que os participantes interagem diretamente com seus potenciais clientes, já que os mesmos acabam tendo apenas o interesse acadêmico de conhecimento das metodologias de desenvolvimento de *software*.

Portanto, para obtenção de melhores resultados, faz-se necessária a aplicação da metodologia juntamente com *startups* fora do contexto acadêmico. Essa aplicação favorecerá tanto a análise do estudo de caso, como a melhoria da metodologia através da identificação de pontos fortes e fracos, que precisam ser melhorados.

5.3 OPERAÇÃO

5.3.1 Aplicação do Estudo Experimental

Durante a disciplina de Engenharia de *Software* do Programa de Pós-Graduação em Ciência da Computação da UFSCar Sorocaba, os alunos são incentivados a criar uma *startup* no decorrer de um semestre. Durante esse semestre, nas primeiras aulas, os alunos devem apresentar as ideias que possuem. Após essa apresentação, cada aluno deve selecionar qual ideia gostaria de trabalhar, e, desta forma, fazer uma formação de equipes.

Após a formação, durante a disciplina, são intercaladas aulas de conteúdo e de apresentação dos resultados. Para a aplicação da metodologia, foram selecionadas algumas aulas de conteúdo, nas quais o conteúdo específico da metodologia seria passado somente para os grupos selecionados.

Desta forma, houveram 4 encontros em sala de aula: no primeiro encontro, foi feita uma apresentação explicativa com as etapas e as principais ferramentas da metodologia. Esta apresentação está presente no Apêndice A. Nesse dia, além do conteúdo, estava programada também uma dinâmica que simulasse a utilização do SW Canvas. Essa dinâmica, por questões de espaço, não foi possível ser aplicada.

Os demais encontros, consistiram em apresentações sobre a evolução dos projetos de todas as equipes. Na primeira apresentação, as equipes deviam focar em mostrar um primeiro BMC, estruturado a partir da ideia selecionada. A segunda apresentação foi para mostrar os resultados obtidos com a evolução do projeto e um primeiro MVP. Por fim, a última apresentação consistia na demonstração dos resultados obtidos com o primeiro MVP, e a sua evolução, através da construção de uma melhoria do MVP.

Durante esse período, além das apresentações, foi feito um acompanhamento e definidas entregas de acordo com as datas das aulas, e conforme descrito na Tabela 5.1. O acompanhamento foi feito durante os encontros presenciais em sala de aula, além de conversas *online* agendadas entre as entregas. Esse acompanhamento foi feito para esclarecimento de dúvidas sobre a metodologia, suas ferramentas e entregáveis.

5.3.2 Acompanhamento e Entregas

O acompanhamento realizado com os grupos que utilizaram a metodologia foi feito através de reuniões via *Skype* para esclarecimento de dúvidas. Essas reuniões aconteceram

conforme necessidade dos grupos. Desta forma, ocorreram 2 reuniões com o grupo *Caregiver* e três com o grupo *Dover*.

Essas reuniões ocorreram separadamente com cada grupo, e suas entregas eram feitas via *e-mail*. Além das reuniões de dúvidas, durante os encontros em sala de aula, também aconteceram momentos de esclarecimento de dúvidas e definição de próximos passos.

Os próximos passos foram sempre definidos conforme o cronograma geral de entregas descrito na seção 5.2.1 Descrição da Instrumentação. As entregas dos artefatos foram feitas com a utilização de um *excel* criado para inserção dos dados e artefatos gerados.

A princípio, este documento ficou disponível online, via *Google Drive*, porém as equipes optaram por ter versões próprias em seus computadores. Desta forma, a cada nova entrega, eles dispararam *e-mails* com os artefatos gerados.

O documento criado para criação e mapeamento dos artefatos está presente no Apêndice C.

5.4 RESULTADO DO ESTUDO

Os seguintes resultados foram obtidos durante o processo de experimentação da metodologia. As Tabelas 7 e 8 contêm os resultados obtidos através da realização dos testes e aplicação do formulário de avaliação do usuário.

5.4.1 Artefatos Gerados pela Aplicação da Metodologia

No decorrer do estudo de caso, os grupos selecionados para utilização da metodologia: *Caregiver* e *Dover*, receberam um material para auxiliá-los durante o processo. Nesse material continha a apresentação realizada no primeiro encontro (Apêndice A), um arquivo *excel* detalhando passo-a-passo a metodologia, com seus artefatos de entrada e de saída (Apêndice B) e outro arquivo *excel* como ferramenta para mapeamento dos dados criados a partir da utilização do SK (Apêndice C).

Este arquivo foi dividido em diferentes abas, num total de 16 abas ligadas através de *links* para navegar entre elas. A ordem das abas criadas ficou da seguinte forma:

1. Processo – apresenta uma visão geral do processo, conforme ilustrado na Figura 4.1;
2. SW Canvas - Figura 4.2;
3. *Business Model Canvas* - Figura 2.5;
4. Requisitos – apresenta uma lista de atividades, técnicas e ferramentas utilizadas durante a fase de Requisitos;
5. Validação – apresenta uma lista de atividades, técnicas e ferramentas utilizadas durante a fase de Validação;
6. Projeto – apresenta uma lista de atividades, técnicas e ferramentas utilizadas durante a fase de Projeto;
7. Implementação – apresenta uma lista de atividades, técnicas e ferramentas utilizadas durante a fase de Implementação;
8. Principal – aba principal, apresenta um resumo dos principais dados inseridos nas demais abas, é uma espécie de *dashboard* da metodologia;
9. Classificação das Histórias – apresenta a matriz (impacto X dificuldade) para qualificação das histórias definidas (Figura 4.9);
10. Personas – apresenta modelos de criação de personas, além de uma lista das personas criadas pelos grupos;
11. 8 Momentos-chave – apresenta a ferramenta 8 Momentos-chave (Figura 4.7);

12. Guia de Entrevista – apresenta uma guia de sugestão de roteiros para entrevistas;
13. Planejamento da Validação – concentra todas as informações relacionadas ao planejamento da validação, com uma breve descrição dos requisitos necessários para a validação: pré-requisitos, ambiente, ferramentas, objetivo e salto de fé. Nesta guia possui também a lista de histórias criadas, com seus resultados esperados, obtidos, comentários e indicador: comprovado (s/n);
14. Criação de Histórias – apresenta a lista de histórias criadas através do conjunto de atores e cenários definidos;
15. Matriz Relacionamento – apresenta a matriz de relacionamento (impacta X é impactada) das histórias (ou requisitos) definidos (Figura 4.14);
16. Retrospectiva – apresenta uma descrição do que é a retrospectiva, como ela deve ser realizada, além da listagem dos itens que deram certo, errado e um plano de ação para corrigir o que deu errado.

5.4.1.1 Artefatos Gerados na fase de Requisitos

A primeira entrega estava relacionada a fase de Requisitos. Nela, os grupos apresentaram dificuldades na utilização do documento *excel* para inserção dos artefatos gerados durante a aplicação do *Startup Kaizen*. Portanto, os grupos optaram por modelos que facilitassem a sua utilização.

O grupo *Dover*, por exemplo, utilizou documentos *word* para mapeamento dos artefatos gerados pela fase de requisitos. Para cada entrega foi criado um novo documento, ficando desta forma: Requisitos, BMC, SW Canvas, Personas, Momentos, Principal, Cartesiano, Validação, Planejamento – cada nome representa um documento diferente conforme ilustrado na **Figura 5.6**.

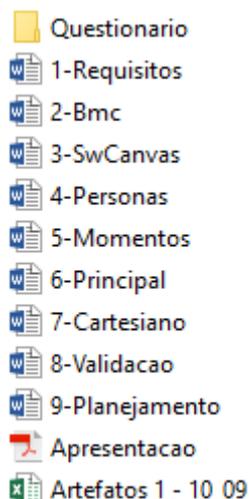


Figura 5.6 - Artefatos gerados pela equipe *Dover* durante a fase de Requisitos

Conforme a lista ilustrada na **Figura 5.6**, no documento Momentos, por exemplo, estão os 8 Momentos-chave criados pela equipe, já em Cartesiano, estão as informações relacionadas a classificação da história (impacto X dificuldade). A Figura 5.7 ilustra uma parte do documento utilizado pela equipe *Dover* para classificação das histórias criadas.

```

=====
Id {Requisitos} -> 01
Impacto / Dificuldade {Requisitos} -> + / +
=====
Id {Requisitos} -> 02
Impacto / Dificuldade {Requisitos} -> - / -
=====
Id {Requisitos} -> 03
Impacto / Dificuldade {Requisitos} -> + / +
=====
Id {Requisitos} -> 04
Impacto / Dificuldade {Requisitos} -> - / -
=====
Id {Requisitos} -> 05
Impacto / Dificuldade {Requisitos} -> - / -
=====
Id {Requisitos} -> 06
Impacto / Dificuldade {Requisitos} -> + / -
=====

```

Figura 5.7 - Exemplo de formato de documento utilizado pela equipe *Dover*

O primeiro artefato a ser preenchido, consistia na aba Requisitos, com algumas perguntas a serem respondidas:

- (1) Descreva a sua ideia;
- (2) Qual problema a sua ideia resolve?
 - (a) Defina sua proposta de valor;
- (3) Como você classificaria esse problema em relação aos possíveis clientes?
 - (a) Blocante;
 - (b) Grande;
 - (c) Pequeno;
 - (d) Trivial;
- (4) Quais são os possíveis clientes?
 - (a) Defina o segmento de clientes (BMC);
 - (b) Crie personas;
- (5) Como os seus clientes usam/interagem com a sua ideia?
- (6) Como você ganhará dinheiro com ela?
 - (a) Defina fontes de renda (BMC);
- (7) Descreva os requisitos iniciais da sua solução em formato de histórias com base nas personas e momentos-chave criados;
- (8) Classifique as histórias iniciais da sua solução conforme impacto X complexidade de desenvolvimento;
- (9) Defina o Salto de fé;
 - (a) O que você quer testar com essa ideia?
 - (b) Quais histórias você quer validar?
- (10) Defina as histórias principais que estarão no MVP.

A **Figura 5.8** ilustra a aba Requisitos com as perguntas, técnicas e ferramentas utilizadas para definição dessa fase.

Objetivo: Como identificar e definir MVP

Atividades	Técnicas	Ferramentas
1- Descreva a sua idéia:		SW CANVAS
2- Qual problema a sua idéia resolve?		
- Defina proposta de valor		Business Model CANVAS
3- Como você classificaria esse problema em relação aos possíveis clientes?	Blocante; Grande; Pequeno; Trivial	
4- Quais são os possíveis clientes?	CANVAS - Definir Grupo de Foco, Criar Personas	Business Model CANVAS, Personas
- Defina o segmento de clientes		Business Model Canvas
- Crie Personas		Personas
5- Como os seus clientes usam/interagem com a sua ideia?	8 Momentos Chave	8 Momentos Chave
6- Como você ganhará dinheiro com ela?		
- Defina fontes de Renda		Business Model Canvas
7- Descreva os requisitos iniciais da sua solução em formato de histórias com base nas personas e momentos chave criados		Principal
8- Classifique as histórias iniciais da sua solução conforme impacto X complexidade de desenvolvimento		Classificação das Histórias
9- Defina o salto de fé:		Planejamento da Validação
- O que você quer testar com essa ideia?		
- Quais histórias você quer validar?		
10- Defina as histórias principais que entrarão no MVP		Principal

Figura 5.8 - Fase de Requisitos

Para estruturação inicial da ideia, através das respostas dadas às perguntas definidas, cada grupo usou um formato diferente. O grupo *Caregiver* utilizou uma apresentação elaborada na ferramenta *prezi*⁸ conforme ilustrado na **Figura 5.9**.

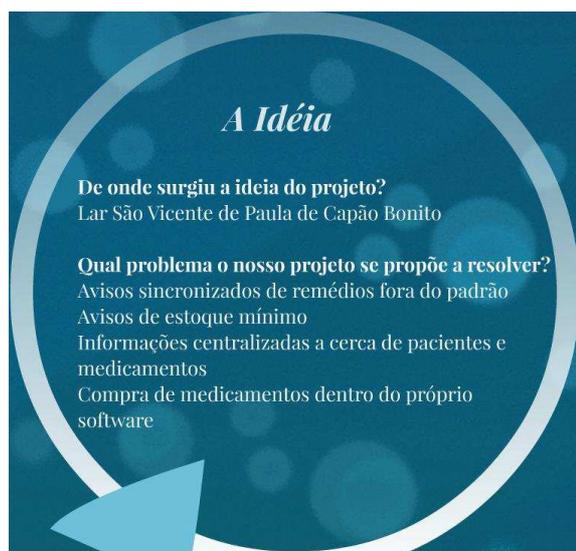


Figura 5.9 – Modelo de documentação utilizada pelo grupo *Caregiver* na fase de Requisitos

Já, o grupo *Dover* utilizou um documento com o nome Requisitos, que consistia na lista de perguntas, nas quais foram respondidas uma por uma. O grupo optou por essa forma

⁸ <https://prezi.com/>

de estruturação devido a incompatibilidade de ferramentas disponibilizadas pelo *Startup Kaizen* em relação às ferramentas que os participantes das equipes possuíam. A Figura 5.10 ilustra o modelo de documentação utilizada pela equipe *Dover* para responder as perguntas da fase de Requisitos.

[] -> Artefato Gerado
 () -> Resposta de uma pergunta
 ? -> Ponto de desvio
 S: Saída R: Resposta E: Entrada

1-Qual a sua ideia?
 S: [Ideia, SwCanvas]
 R: Aplicativo para o transporte de encomendas entre pontos definidos com valor competitivo para quem envia e oportunidade de renda para quem transporta.

2-Qual problema ideia resolve?
 S: [Proposta de Valor, BMC]
 R: Achar quem queira transportar encomendas entre dois pontos definidos. Redução de custo no envio de encomendas.

3-Como os clientes usam a ideia?
 S: [Cenários, SwCanvas] e [Cenário, Aba 8 momentos]
 R: Por meio de uma aplicação no smart phone. Detalhe dos cenários estão no artefato Momentos.

4-Definir os possíveis clientes?
 S: [Personas, SwCanvas] e [Segmentos de Mercado, BMC] e [Personas, aba Personas]
 R: Quem deseja transportar encomendas, para reduzir custo da viagem ou como fonte de renda. Quem deseja enviar ou receber encomendas de modo mais customizado.

Figura 5.10 - Modelo de documentação utilizada pelo grupo *Dover* na fase de Requisitos

Além dos documentos utilizados, posteriormente a equipe *Dover* consolidou os artefatos gerados numa planilha criada por eles com base na planilha definida pelo *Startup Kaizen*. A Figura 5.11 ilustra a aba Principal criada pela equipe e a **Figura 5.12** ilustra a aba Principal criada para mapeamento dos principais artefatos durante aplicação do SK.

ID	HISTORIA	DESCRICAO	IMPACTO/DIFICULDADE	PRIORIDADE	MVP
1	Registrar rotas-horarios por transportadores	Cadastro de endereços de origem e destino, meio de transporte e horarios aos quais o trajeto será percorrido pelo transportador, sera as informações de rotas e horarios existentes.	[+/-]		sim
2	Modelos de entrega	Quem transporta pode cadastrar as formas de entrega que aceita, elas são entrega a local definido, postagem em correio local, agendamento de local-horario com quem vai receber.	[-/-]		nao
3	Busca por quem transporta	Formulario que lista as possiveis rotas de envio e quem faz essa rota com em diferentes horarios. Permitindo a quem deseja enviar realizar um agendamento. As rotas tem origem e destino definidos.	[+/-]		sim
4	Rastreamento por check-in	Registro da posição momentanea do transportador. Posição que só disponível aos envolvidos em algum transporte dele naquele horario	[-/-]		nao

Figura 5.11 - Aba Principal criada pela equipe *Dover*

#	História	Descrição	Classificação	Prioridade	MVP
1	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				
2	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para				
3	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para				
4	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para				
5	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para				

Figura 5.12 - Planilha Startup Kaizen - Aba Principal

Conforme evolução das atividades, o grupo *Caregiver* realizou as demais entregas utilizando como base o documento de apresentação do *Startup Kaizen* feita para eles no primeiro encontro, e enviada como material de apoio (Apêndice A). Eles utilizaram cada *slide* criado para apresentação do *Startup Kaizen* como base para definição dos artefatos desenvolvidos. A **Figura 5.13** ilustra o documento enviado pela equipe durante as atividades iniciais da fase de Requisitos.



Figura 5.13 - Modelo utilizado pelo grupo *Caregiver* para definição dos artefatos na fase de Requisitos

Cada quadrado do lado esquerdo possui os artefatos gerados durante a fase de Requisitos do SK com um total de 60 *slides* utilizados. Por exemplo, o *slide* número 32 representa a classificação das histórias criadas pelo grupo, conforme ilustrado na Figura 5.14.

CLASSIFICAÇÃO DAS HISTÓRIAS



Figura 5.14 - Modelo utilizado pela equipe *Caregiver* para classificação das histórias

Conforme descrito na seção 4.1, a matriz impacto no cliente *versus* dificuldade de implementação foi utilizada para definição das histórias que seriam desenvolvidas no MVP. Cada grupo ficou responsável por definir quais seriam os quadrantes que eles utilizariam como referência para a construção desses MVPs. Desta forma, o grupo *Caregiver* selecionou os dois quadrantes que possuíam maior impacto, descartando as histórias que não possuíam impacto no cliente e que estavam nos dois quadrantes da parte inferior da matriz.

O grupo *Dover*, assim como o *Caregiver*, também optou pela escolha das histórias que possuíam mais impacto no desenvolvimento do MVP. A **Figura 5.7** ilustra a forma de classificação utilizada pelo grupo *Dover*.

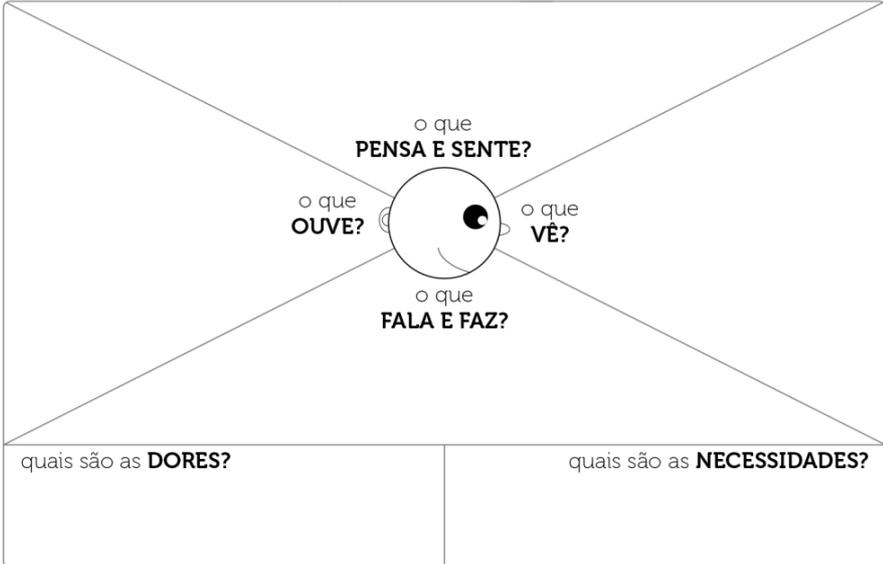
Na Tabela 5.4 estão listadas as quantidades de histórias criadas e selecionadas para o MVP. Nela, é possível perceber que ambos os grupos focaram apenas em algumas histórias para desenvolverem seus MVPs diretamente com seus clientes. Essa seleção trouxe benefícios já que os grupos conseguiram desenvolver suas primeiras versões de *software* de forma mais rápida, focando apenas no que era essencial para sua validação.

Tabela 5.4 - Quantidade de Histórias criadas e selecionadas para o MVP

Histórias	Criadas	Selecionadas (MVP)
<i>Caregiver</i>	7	3
<i>Dover</i>	17	6

As duas equipes entregaram todos os artefatos solicitados na primeira entrega, e a equipe *Caregiver* incluiu o mapa de empatia (Figura 5.15) para auxílio na criação das personas. Esta inserção deu-se por influência da disciplina de Interface Humano-Computador (IHC).

Nome: _____ Idade: _____



The diagram is a large rectangle divided into four quadrants by two diagonal lines that intersect at a central point. In the center of this intersection is a simple line drawing of a person's face, showing a nose, a smiling mouth, and two eyes. Each eye is positioned near a question: 'o que OUVE?' on the left and 'o que VÊ?' on the right. Above the face, centered, is the text 'o que PENSA E SENTE?'. Below the face, centered, is the text 'o que FALA E FAZ?'. The bottom-left quadrant of the rectangle is labeled 'quais são as DORES?' and the bottom-right quadrant is labeled 'quais são as NECESSIDADES?'.

o que
PENSA E SENTE?

o que
OUVE?

o que
VÊ?

o que
FALA E FAZ?

quais são as **DORES?**

quais são as **NECESSIDADES?**

Figura 5.15 - Mapa de Empatia

A disciplina de IHC e o Mapa de Empatia tiveram uma grande influência na criação das personas por parte do grupo *Caregiver*. Através da disciplina, e da utilização da ferramenta, o grupo apresentou uma maior riqueza de detalhes nas personas do que o *Dover*. Veja exemplos na Figura 5.16 e Figura 5.17.

	<p>Nome: Andréia Azevedo</p> <p>Dados demográficos e pessoais: 32 anos, Auxiliar de Enfermagem</p>	<p>Perfil: Andréia é formada em Auxiliar de Enfermagem pelo SENAC. Trabalha atualmente em uma Intuição de Longa Permanência para idosos no interior de São Paulo.</p>
<p>Conhecimento competências e habilidades: Andréia trabalha como auxiliar de enfermagem a seis anos. Sua grande preocupação e objetivo é cuidar do bem-estar dos idosos. Ela gosta de saber que seu trabalho esta sendo bem feito e ter certeza que não esqueceu de cumprir nenhuma tarefa, considerando que ela é muito responsável por ter consciência que outros seres humanos dependem dos seus cuidados. Ao longo do dia ela visita e ministra medicações em muitos idosos e preocupa-se em ser eficiente</p>		
<p>Metas, Motivos e Preocupações Gostaria de poder automatizar o controle do estoque para que seja possível rapidamente repor um medicamento quando o mesmo esta acabando. Ela acharia interessante que isso fosse feito de maneira simples e que ela pudesse ser avisada quando o medicamento estivesse se esgotando. Ela gostaria também de uma maneira automatizada de ser avisada quando fosse necessário administrar uma medicação "fora de padrão" em um determinado idoso. Ela se preocupa em atrasar a medicação pois são muitos idosos em diferentes quartos e ela precisa memorizar todos eles e inclusive memorizar as fichas para</p>		
<p>Padrões de utilização Andréia utiliza seu smartphone , cujo sistema operacional é Android, todos os dias para acessar as redes sociais e ficar por dentro do que esta havendo com seus amigos e família, assim como, gosta de utilizar o whats app para trocar mensagens com seu marido e filho. Não sente muita dificuldade para acessar a internet e utilizar aplicações com as quais já esta habituada. Já ao acessar alguns aplicativos ela admite sentir dificuldade média em determinadas situações, como ao lidar com as configurações dos aplicativos.</p>		
<p>Ela precisa de um jeito de: Controlar o estoque de medicamentos e lembrar-se de ministrar os medicamentos fora do padrão com eficiência. Para: Realizar o seu trabalho mais eficientemente e de maneira mais fácil e organizada. Porque ela valoriza: O trabalho bem feito e o bem estar dos idosos</p>		

Figura 5.16 - Exemplo de Persona criada pelo grupo *Caregiver*

Na Figura 5.16 é possível notar uma descrição mais detalhada da persona criada, além de possuir uma imagem que represente a persona. Já, na Figura 5.17, percebe-se a ausência de uma imagem que reflita as personas criadas, além de não possuírem muitos detalhes.

NOME	IDADE	SEXO	STATUS	OBJETIVOS	HABILIDADES	TAREFAS	EXPECTATIVAS
Carlos	30	Masculino	Primario	Enviar uma encomenda no sabado de manha para um comercio no centro de uma cidade vizinha	Contador	Envia encomendas a clientes para o qual trabalha como contador	
Mara	25	Feminina	Secundario	Receber os relatorios semanais da loja onde trabalha	Auxiliar de escritorio. Nao dirige	Receber e catalogar as encomendas enviadas a loja	Receber as encomendas em horarios programados
Rafael	23	Masculino	Primario	Ir ao centro no sabado comprar uma camiseta	Estudante de curso Tecnico	Vai ao centro com frequencia aos sabados para fazer compras e a passeio	Gastar menos com as idas ao centro

Figura 5.17 - Exemplo de Persona criada pelo grupo *Dover*

Durante essa fase, foram gerados artefatos que foram analisados e validados nas fases posteriores. A Tabela 5.5 apresenta a quantidade de artefatos gerados para cada item por cada grupo.

Tabela 5.5 - Quantidade de artefatos gerados pelos grupos na fase de Requisitos

	Histórias	Personas	Momentos-Chave
<i>Caregiver</i>	5	2	6
<i>Dover</i>	17	3	8

5.4.1.2 Artefatos Gerados na fase de Validação

As entregas 2 e 3 foram relacionadas a etapa de Validação. Nelas, os grupos se prepararam para realizar uma validação direta com o usuário, indicando as histórias que seriam validadas, como elas seriam validadas e o resultado esperado.

Para o planejamento foi criada uma aba no documento enviado: “Planejamento da Validação” (Apêndice C – Aba 13: Planejamento da Validação). Nesta aba contém um Guia para Planejamento da Validação, com as seguintes informações: Pré-Requisitos, Ambiente, Ferramentas, Objetivo da Validação, Salto de Fé, e uma tabela contando as histórias criadas com o resultado esperado, resultado obtido, comprovado (s/n) e comentários. A Figura 5.18 ilustra a Guia para Planejamento da Validação e a Figura 5.19 ilustra a tabela de planejamento da validação geradas pelo grupo *Caregiver*.

Guia para planejamento da validação

Pré-requisitos:	Será necessário a disponibilidade dos enfermeiros e administradores, permissão da instituição e os protótipos prontos para serem validados.
Ambiente:	Os testes serão realizados dentro da ILPI no ambiente de trabalho dos enfermeiros de administradores da instituição. Para fazer os teste será utilizado o notebook do avaliador e o mesmo levará um celular android para testar a aplicação mobile.
Ferramentas:	Foi feita uma validação suja utilizando um protótipo de baixa fidelidade (com mockups e POP) e uma segunda validação para entrega 2 utilizando um protótipo de alta fidelidade que englobou 40% do sistemas, apresentando as funcionalidades principais.
Objetivo da Validação:	Verificar se o que esta sendo implementado esta atendendo a expectativa do usuário e se será necessário alterar alguma coisa nas funcionalidades principais para entrega final da sprint 2.
Salto de fé:	https://drive.google.com/drive/folders/0B1Ym_QFjctdW1ZdUxQWZlUTA

Figura 5.18 - Guia para planejamento da validação do grupo *Caregiver*

História	Resultado Esperado	Resultado obtido	Comprovado (s/n)?	Comentários
Como Bianca, técnica de administração, preciso estar de olho no estoque de medicamentos para poder fazer novos pedidos.	Conforme os enfermeiros ministrarem a medicação e apontarem isso no mobile o remédio é decrementado do banco de dados e quando o estoque chegar em determinados níveis alguns avisos são disparados para que Bianca possa fazer novos pedidos.	Apesar de Bianca poder receber os avisos de estoque baixo sobre um determinado medicamento, a auxiliar de enfermagem Andreia consegue somente visualizar na aplicação mobile uma tela com os próximos avisos de medicações a serem ministradas. Consequentemente, Andreia ainda não recebe o aviso efetivo de quando deve medicar um idoso, e com isso, o remédio ainda não é decrementado do banco de dados.	Sim	Sem comentários.
Como Andreia, auxiliar de enfermagem, preciso estar de olho no estoque de medicamentos para garantir que sempre que for ministrar um medicamento em um determinado idoso este medicamento estará disponível.	Conforme os enfermeiros ministrarem a medicação e apontarem isso no mobile o remédio é decrementado do banco de dados e quando o estoque chegar em determinados níveis alguns avisos são disparados. Dessa maneira o administrativo pode fazer novos pedidos e Andreia não vai se preocupar com falta de medicamentos em estoque	Bianca já recebe os avisos no sistema web sobre os medicamentos que estão em baixa no estoque, e pode visualizar a quantidade atual e quantidade mínima que deve ter em estoque, de cada medicamento. Todavia, Andreia ainda não consegue indicar que uma medicação foi aplicada a um idoso, pois ainda não está recebendo os avisos de aplicação de medicação, e consequentemente, não há decréscimo da quantidade de um medicamento no banco de dados.	Sim	Sem comentários.

Figura 5.19 - Tabela de validação grupo *Caregiver*

Ambas equipes utilizaram a ferramenta fornecida para o planejamento da validação. As figuras Figura 5.20 e Figura 5.21 ilustram a planilha de validação gerada pela equipe *Dover*. Além da aba criada, as equipes utilizaram outras ferramentas para a validação, tais como: formulários *online* para entrevistas, formulários de papel e protótipos de baixa e/ou alta fidelidade.

Pré-requisitos:	Ter uma aplicação no sistema operacional Android com as telas relacionadas às histórias desenvolvidas para usuários que tenham conhecimentos básicos no uso de aplicativos móveis e com acesso à internet. O usuário deverá ter sido previamente convidado para utilizar a aplicação
Ambiente:	Com um smartfone com sistema operacional android o usuário pode utilizar a aplicação em qualquer lugar desde que esteja conectado com a internet
Ferramentas:	- Convite para testar a aplicação via email com link para baixar o instalador - Uso de um formulário com questões simples para validar as histórias com notas Sendo: 1 nada útil, 2 pouco útil, 3 normal, 4, útil, 5 muito útil - Observação relacionado ao uso de um protótipo funcional de telas em uma aplicação mobile
Objetivo da Validação:	Validar se a aplicação auxilia o usuário a encontrar pessoas que possam entregar encomendas de maneira confiável
Salto de fé:	O registro de novos usuários através de convites, a qualificação dos usuários, o cadastro de rotas de pessoas interessadas em realizar transportes são pontos suficientes para que as pessoas se interessem em utilizar o aplicativo?

Figura 5.20 - Guia para planejamento da validação do grupo *Dover*

H ID	Histórias	RESULTADO ESPERADO	RESULTADO OBTIDO	COMPROVADO (S/N)	COMENTARIOS
1	Eu [Munilo, estudante universitário] gostaria de [ACESSAR] o cadastro de rotas para que eu possa [REGISTRAR ROTAS E HORÁRIOS] para transportar encomendas	Espera-se que usuário entenda a opção para cadastrar rotas e horários e seja capaz de cadastrá-la de maneira rápida e fácil	Nada útil 10% Pouco útil 10% Normal 10% Útil 70% Muito útil 0%	S	não houve dificuldades para entender a história porém houve comentários sobre a interface
3	Eu [Joana, vendedora] preciso [PROCURAR] pessoas que tenham disponibilidade para [TRANSPORTAR ENCOMENDAS] toda semana	Espera-se que o usuário realize uma busca e encontre facilmente as rotas para transportar sua encomenda de maneira eficiente	Nada útil 10% Pouco útil 10% Normal 30% Útil 30% Muito útil 20%	S	Apesar de haver poucas rotas cadastradas, o sistema ainda precisa de melhorias relacionadas à detalhes da busca e dos usuários que oferecem transporte e opção para múltiplos destinos
6	Eu [Carlos, contador] quero ter a opção de [REGISTRAR] o meu grau de satisfação com a entrega contribuindo para o [RANKING DE USUÁRIOS] da aplicação	Espera-se que os usuários possam, ao finalizar a entrega de uma encomenda qualificar o outro de maneira intuitiva e rápida	Nada útil 10% Pouco útil 10% Normal 0% Útil 0% Muito útil 80%	S	Poucos experienciaram esta tarefa mas acharam a ideia do ranqueamento válida
10	Eu [Fernanda, estudante universitária] gostaria de [REALIZAR] o [REGISTRO DE ENTREGA] de uma encomenda que eu transporte	Espera-se que o usuário que transporta uma encomenda seja capaz de registrar a entrega do produto de maneira satisfatória	Nada útil 0% Pouco útil 20% Normal 10% Útil 50% Muito útil 20%	S	Poucos puderam experienciar esta funcionalidade por não haver encontrado rotas que eles pudessem enviar, mas no geral se apresentou útil desde que o aplicativo passe a receber mais usuários com o tempo

Figura 5.21 - Tabela de validação grupo *Dover*

Durante a validação, as duas equipes confirmaram todas as histórias selecionadas para o MVP. O grupo *Dover* selecionou 6 histórias para o MVP, enquanto o *Caregiver* selecionou 3. Os dois grupos validaram as histórias utilizando diferentes escalas: escala de utilidade (*Dover*) e de prioridade (*Caregiver*). Na Tabela 5.6 estão os dados relacionados as histórias criadas, selecionadas e validadas. Em ambos os grupos, os clientes validaram as histórias geradas utilizando um protótipo de alta fidelidade criado, e, após sua análise, os usuários classificaram na escala definida por cada grupo. A Figura 5.22 ilustra a escala definida pelo grupo *Caregiver*, e a Figura 5.21 ilustra a escala definida pelo grupo *Dover*.

Tabela 5.6 - Quantidade de Histórias criadas, selecionadas e validadas

Histórias	Criadas	Selecionadas (MVP)	Validadas
<i>Caregiver</i>	7	3	6
<i>Dover</i>	17	6	6

- (1) menor prioridade
- (2) pouca prioridade
- (3) prioridade média
- (4) alta prioridade
- (5) maior prioridade

Figura 5.22 - Escala de validação definida pelo grupo *Caregiver*

- Nada útil
- Pouco útil
- Normal
- Útil
- Muito útil

Figura 5.23 - Escala de validação definida pelo grupo *Dover*

Após a validação, os grupos analisaram os resultados quantitativamente e definiram quais as histórias que deveriam ser inseridas na fase de Projeto para planejamento de *sprints*

de desenvolvimento. A Figura 5.24 ilustra algumas respostas obtidas pelo grupo *Dover* durante a fase de Validação. Ao total, eles obtiveram 32 respostas no formulário.

Indicação de data e hora	Idade	Sexo	Qual a frequência com qu	Utiliza redes sociais? (
8/31/2015 12:57:03		33 Masculino	1 vez por mês	Facebook, Instagram,
8/31/2015 13:04:44		34 Masculino	1 vez por mês	Whatsapp
8/31/2015 13:07:28		32 Masculino	2 ou mais vezes por sem	Facebook, Instagram,
8/31/2015 13:08:46		33 Masculino	1 vez por mês	Facebook
8/31/2015 13:11:30		34 Masculino	Não viajo com frequencia	Facebook, Instagram,
8/31/2015 13:13:18		23 Masculino	Todos os dias	Facebook, Whatsapp,
8/31/2015 13:14:58		27 Masculino	1 vez por semana	Facebook, Whatsapp,
8/31/2015 13:16:51		33 Masculino	Todos os dias	Whatsapp, LinkedIn
8/31/2015 13:19:37		30 Masculino	Todos os dias	Whatsapp, LinkedIn
8/31/2015 13:33:11		25 Masculino	Todos os dias	Facebook, Whatsapp
8/31/2015 13:39:56		27 Masculino	1 vez por mês	Facebook, Instagram,
8/31/2015 13:43:24		27 Feminino	Todos os dias	Facebook, Instagram,

Figura 5.24 - Respostas recebidas durante Validação pela equipe *Dover*

Enquanto a equipe *Dover* optou por realizar uma validação mais qualitativa, com base na quantidade de respostas obtidas, o grupo *Caregiver* optou por uma avaliação qualitativa. Desta forma, o *Caregiver* realizou entrevistas diretas com potenciais usuários da solução que eles estavam trabalhando, obtendo um número menor de respostas, porém com mais riqueza de informações. A Figura 5.24 ilustra o formulário de resultados da avaliação aplicada pela equipe *Caregiver*.

História	Resultado Esperado	Resultado obtido	Comprovado (s/n)?	Comentários
Como Bianca, técnica de administração, preciso estar de olho no estoque de medicamentos para poder fazer novos pedidos.	Conforme os enfermeiros ministrarem a medicação e apontarem isso no mobile o remédio é decrementado do banco de dados e quando o estoque chegar em determinados níveis alguns avisos são disparados para que Bianca possa fazer novos pedidos.	Apesar de Bianca poder receber os avisos de estoque baixo sobre um determinado medicamento, a auxiliar de enfermagem Andreia consegue somente visualizar na aplicação mobile uma tela com os próximos avisos de medicações a serem ministradas. Conseqüentemente, Andreia ainda não recebe o aviso efetivo de quando deve medicar um idoso, e com isso, o remédio ainda não é decrementado do banco de dados.	Sim	Sem comentários.

Figura 5.25 - Resultados obtidos durante a Validação pelo grupo *Caregiver*

5.4.1.3 Artefatos Gerados na fase de Projeto

Após a validação das histórias geradas, na fase de Projeto, foi realizado o planejamento de como elas seriam implementadas. Neste planejamento, as equipes informaram em quais *sprints* as histórias seriam desenvolvidas.

A equipe *Caregiver* selecionou as histórias e subdividiu as histórias em histórias menores, gerando um total de 11 histórias. Além disso, a equipe listou todas as atividades relacionadas ao projeto, dividindo-as também nas *sprints*.

No total, o grupo *Caregiver* rodou 3 *sprints* (0 a 2), com um total de 50 tarefas incluindo tarefas associadas a gerenciamento de projeto e validação. Já, as histórias criadas foram divididas nas *sprints* 1 e 2, sendo a *sprint* 0 focada apenas nas tarefas de validação e gerenciamento de projetos. A Figura 5.26 - *Backlog* de produto gerado pela equipe *Caregiver* ilustra o *backlog* gerado, e a Figura 5.27 um exemplo de *sprint* rodado pela equipe.

BACKLOG		
Prioridade	Item	Versão
Alta	Cadastro, atualização e exclusão de medicamentos no estoque da instituição.	Web (Enfermeiro)
Alta	Atribuição (cadastro, atualização e exclusão) de medicamentos para os idosos.	Mobile e Web (Enfermeiro)
Alta	Recebimento de avisos sobre o valor mínimo de determinado medicamento no estoque.	Mobile e Web (Administrativo e Enfermeiro)
Alta	Recebimento de avisos disparados pelo próprio aplicativo, no horário programado, ao qual informa ao enfermeiro o	Mobile

Figura 5.26 - *Backlog* de produto gerado pela equipe *Caregiver*

Sprint 0		
Sprint / Itens	Prioridade	Data de finalização da task
Definir as principais características do domínio.	Alta	27/ago
Delimitar o escopo.	Alta	27/ago
Pesquisar sistemas semelhantes.	Alta	28/ago

Figura 5.27 - *Sprint* 0 criada pela equipe *Caregiver*

O grupo *Dover* realizou três *sprints* de desenvolvimento (1 a 3), sendo na primeira trabalhadas as 6 histórias selecionadas no MVP. Na segunda *sprint* foram trabalhadas 3 histórias, e por fim, na terceira foram trabalhadas 4 histórias. Desta forma, do total de 17 histórias levantadas, 13 foram desenvolvidas e liberadas para seus usuários.

Além da definição das *sprints*, os grupos também deveriam entregar nesta etapa a Matriz de Relacionamento preenchida. Porém somente o grupo *Dover* entregou a matriz preenchida, indicando quais histórias impactam e/ou são impactadas umas pelas outras. Na Figura 5.28 contém uma parte da Matriz de Relacionamento gerada pela equipe *Dover*.

		Criar	Deletar	Agendar	Localizar	Registrar
IMPACTA	Criar Rotas			X	X	
	Deletar Rotas			X		
	Agendar envio	X	X		X	
	Localizar Rotas	X		X		
	Registrar usuario					
	Enviar convites					X
	Login					X
	Consultar Agendamentos		X	X	X	
	Qualificar Envolvidos					X
	Visualizar Rotas	X	X		X	
	Criar Itinerarios		X			
	Deletar Itinerarios		X			
	Atualizar estado do		X		X	
	Foto Status			X		
	Modelo de Entrega			X		
	Acesso pelo Facebook				X	X

Figura 5.28 - Matriz de Relacionamento gerada pela equipe *Dover*

5.4.1.4 Artefatos Gerados na fase de Implementação

Durante a fase de implementação, as equipes tiveram que desenvolver o *software* de acordo com os requisitos levantados, realizando as *sprints* planejadas e testes. As principais entregas dessa fase são: parte funcional e executável do *software*, matriz de rastreabilidade (histórias X código), lista e quantidade de *bugs* encontrados.

O grupo *Caregiver* fez dois *releases* de *software*, classificando-as como MVP1 e MVP2, já o grupo *Dover* fez três *releases* de *software*: 1.0.0, 1.1.0 e 1.2.0. Na versão MVP1 do grupo *Caregiver*, foi encontrado apenas uma falha durante os testes. Já, na versão MVP2 foram encontradas duas falhas.

Nas versões realizadas pelo grupo *Dover*, foram encontradas um total de 33 falhas. Na versão 1.0.0 foram encontrados 18 *bugs*, já na versão 1.1.0 foram encontrados 10 e, por fim, na versão 1.2.0 foram encontrados 5 *bugs*. Desta forma, foi possível perceber que a quantidade de falhas diminuiu conforme a evolução do desenvolvimento. A Tabela 5.7 apresenta a quantidade de falhas encontradas por versão de *software* de cada grupo.

Tabela 5.7 - Quantidade de falhas encontradas por grupo em cada versão de *software* liberada

<i>Caregiver</i> – Versão	Quantidade Falhas	<i>Dover</i> - Versão	Quantidade Falhas
MVP1	1	1.0.0	18
MVP2	2	1.1.0	10
		1.2.0	5

Para gerenciamento de falhas foi recomendada a utilização de ferramentas existentes no mercado, porém os grupos optaram por fazer o gerenciamento na aba Principal do documento fornecido. As figuras Figura 5.29 e Figura 5.30 ilustram a aba Principal com o mapeamento das Histórias e falhas encontradas pelo grupo *Dover*.

ID	HISTORIA	DESCRICAÇÃO	IMPACTO/DIFICULDADE	PRIORIDADE	MVP	VALIDADO (S/N)
1	Eu [Murilo, estudante universitário] gostaria de [ACESSAR] o cadastro de rotas para que eu possa [REGISTRAR ROTAS E HORÁRIOS] para transportar encomendas	Cadastro de endereços de origem e destino, meio de transporte e horários aos quais o trajeto será percorrido pelo transportador, sera as informações de rotas e horários existentes.	[+/+]		1 sim	sim
2	Eu [Joana, vendedora] quero ter a opção de [ESCOLHER] qual é o melhor [MODELOS DE ENTREGA] de acordo com a minha necessidade	Quem transporta pode cadastrar as formas de entrega que aceita, elas são entrega a local definido, postagem em correio local, agendamento de local-horario com quem vai receber.	[+/-]		9 sim	sim
3	Eu [Joana, vendedora] preciso [PROCURAR] pessoas que tenham disponibilidade para [TRANSPORTAR ENCOMENDAS] toda semana	Formulario que lista as possiveis rotas de envio e quem faz essa rota com em diferentes horarios. Permitindo a quem deseja enviar realizar um agendamento. As rotas tem origem e destino definidos.	[+/+]		2 sim	sim

Figura 5.29 - Aba Principal: Mapeamento das histórias e falhas encontradas pelo grupo *Dover* - Parte I

MODULO	SPRINT	TESTADO (S/N)	PASSOU (S/N)	BUGS
1.0.0	1	sim	sim	- problema ao deletar rota com horario agendado - não foi possivel deletar rota de um usuário - não foi possivel associar uma rota a um usuário - problema para retornar entidades relacionadas a uma rota - erro ao gravar entidades relacionadas à entidade rota
1.1.0	2	sim	sim	- problema em recuperar dados por causa do novo relacionamento - lista de produto vazia ao enviar ao servidor
1.0.0	1	sim	sim	- retornar os agendamentos com os devidos subcampos preenchidos retorna erro - problema com as latitudes e longitudes removidas de modo incorreto - problema na interface interfere ao selecionar os itens da lista - o nome do usuário que envia aparece incorretamente na aba acompanhar do transportador

Figura 5.30 - Aba Principal: Mapeamento das histórias e falhas encontradas pelo grupo *Dover* - Parte II

Ambos os grupos entregaram as versões funcionais de suas soluções. A Figura 5.31 ilustra o aplicativo criado pelo grupo *Dover*, essa figura consiste na tela de “Produto Retirado”, com uma mensagem para o usuário informando que o produto que ele enviou, foi retirado. A Figura 5.32 ilustra a tela “Atribuir Medicamento” criada para a persona que representa um enfermeiro, do grupo *Caregiver*. Esta tela foi elaborada durante o MVP1.



Figura 5.31 - Tela “Produto Retirado” do grupo *Dover*

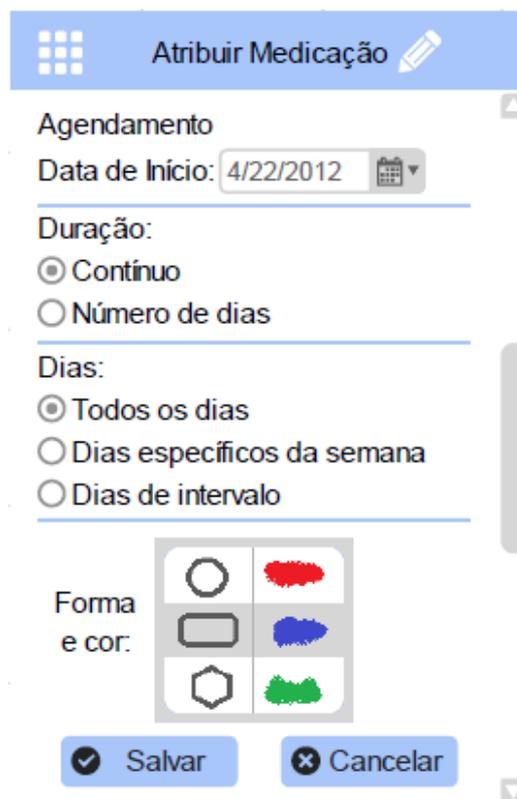


Figura 5.32 - Tela “Atribuir Medicação” do grupo *Caregiver* (MVP1)

5.4.2 Resultado dos Testes Realizados

Para avaliação dos resultados do estudo de caso, ao final da aplicação do estudo, foi solicitado à todas as equipes para realizarem testes do *software* de outra equipe. Para a realização dos testes, foi criada uma planilha que consiste na mesma planilha utilizada na metodologia, porém com foco apenas em testes, descartando a validação. A Figura 5.33 - Planilha de testes de *software* ilustra a planilha de testes criada. Nela, são listados os requisitos ou histórias do *software* desenvolvido, uma breve descrição, se foi testado (sim ou não), passou (sim ou não) e a lista de bugs.

Na planilha, também foram definidas métricas para avaliação dos testes. Essas métricas são: cobertura de testes e índice de qualidade, que são automaticamente calculadas.

#	História	Descrição	Testado (s/n)	Passou (s/n)	Bugs
1	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]		Sim	Sim	- Bug 1 - Bug 2 - Bug 3
2	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]		Não	--	
3	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]		Sim	Não	

Figura 5.33 - Planilha de testes de *software*

Após a última apresentação, com a evolução do MVP, a planilha de testes (Figura 5.33 - Planilha de testes de *software*) foi utilizada por todos os grupos como uma ferramenta para auxiliar na elaboração dos testes. Durante a execução, cada grupo era responsável por realizar testes do *software* do outro grupo, conforme indicado na Tabela 5.2 - Grupos responsáveis pelos testes. Desta forma, o grupo desenvolvedor do *software* foi responsável por listar seus requisitos e/ou histórias para que a outra equipe pudesse testá-las. A Figura 5.34 ilustra a tabela criada pelo grupo Factoring Sys para realização dos testes pelo grupo *Dover*.

#	Histórias ou Requisitos	Descrição
1	Validar e verificar o site entregue	Site desenvolvido para divulgação do sistema como um todo
2	Site:	http://ec2-54-69-185-3.us-west-2.compute.amazonaws.com:8080/factory/
3	Efetuar o Login através do site	No canto esquerdo no menu encontra-se o login
4	Efetuar o login como uma empresa	A empresa poderá lançar documentos para solicitar créditos
5	Usuario:	cliente@test.com
6	Senha:	123456
7	Cadastrar uma nova proposta	Cadastrar uma nova proposta, as factorings cadastradas tem um valor mínimo para receber essa proposta sendo apenas superior a 50000, assim sendo uma proposta menor não irá aparecer para a factoring. Lance quantas propostas forem necessárias para o teste. Ao final do cadastro a proposta pode ser editada.
8	Deslogar do sistema e retornar como uma factoring	

Figura 5.34 - Planilha de testes criada pelo grupo Factoring Sys

Com a lista de requisitos e/ou histórias em mãos, a planilha foi enviada para o grupo responsável por testá-la. Neste momento, não foi dado nenhum direcionamento de como os testes deveriam ser realizados, ficando a critério de cada grupo utilizar métodos e/ou ferramentas de conhecimento próprio.

Para avaliação do resultado dos testes, as seguintes métricas foram levantadas: número de requisitos, cobertura de testes, índice de qualidade e quantidade de bugs. Os resultados obtidos estão listados na Tabela 5.8 - Resultado obtido com os testes realizados de cada grupo.

Tabela 5.8 - Resultado obtido com os testes realizados de cada grupo

	<i>Caregiver</i>	<i>Dover</i>	<i>Factoring Sys</i>	Imp. Distribuída
# Requisitos	5	14	7	7
Cobertura de Testes	60%	100%	100%	100%
Índice de Qualidade	20%	31%	85%	100%
Quantidade de bugs	4	10	2	0

O cálculo das métricas apresentadas na Tabela 5.8 foi feito da seguinte forma:

- Cobertura de testes: quantidade de requisitos testados dividido pela quantidade total de requisitos;
- Índice de Qualidade: quantidade de requisitos que passaram dividido pela quantidade total de requisitos;

Na Figura 5.35 - Resultado dos testes do sistema *Factoring Sys* encontram-se os resultados obtidos pelo grupo *Factoring Sys*, após seu *software* ser testado pelo grupo *Dover*.

Nome do grupo testado	Factoring Sys (Grupo 4)
Versão do Software	???
Nome do Grupo Testador	Dover - Transporte de Mercadoria(Grupo 3)
Quantidade de Bugs	2

Cobertura de Testes 58%

Índice de Qualidade 50%

Figura 5.35 - Resultado dos testes do sistema *Factoring Sys*

5.4.3 Resultado da Avaliação do Usuário

Além dos testes realizados, foi criado um formulário para avaliar o impacto da solução nos clientes e das falhas encontradas por eles. Este formulário foi criado e disponibilizado para as equipes solicitarem a seus clientes que respondessem *online*.

A análise de aceitação é feita através de dois fatores diferentes: facilidade da utilização e utilidade da ferramenta. O primeiro fator é analisado sob a perspectiva da intenção da utilização da ferramenta analisada. Enquanto o segundo fator, utilidade da ferramenta, é analisado a partir do desejo da aplicação da nova ferramenta.

Portanto, visando analisar esses fatores, foi criado um formulário com 5 perguntas para identificação do perfil do usuário, solução utilizada e qual funcionalidade fez com que os clientes a utilizassem. A Figura 5.36 ilustra dois exemplos de perguntas feitas com os usuários das ferramentas criadas, o formulário completo pode encontra-se no Apêndice E.

No seu dia-a-dia, o quão útil você considera a ferramenta?*

1 2 3 4 5 6 7 8 9 10

Nada útil Muito útil, não sei como fiquei sem essa solução até agora

Qual nota você daria em relação a dificuldade de utilizar a ferramenta?*

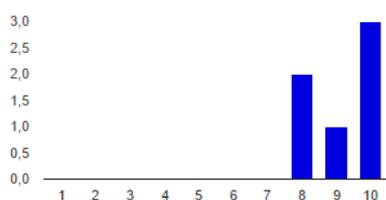
1 2 3 4 5 6 7 8 9 10

Muito Difícil Muito Fácil

Figura 5.36: Exemplo de perguntas realizadas para os usuários

As demais perguntas, num total de 11, foram feitas para analisar e entender a influência da solução oferecida para o usuário. Para isso, foram avaliados alguns pilares: utilidade e dificuldade de utilização da ferramenta, se a solução oferecida atende ao propósito pelo qual fez o usuário a utilizá-la, identificação de falhas e seu impacto, e por fim, avaliação geral da ferramenta através da indicação ou não para outras pessoas. Na Figura 5.37 estão ilustrados os resultados das duas perguntas representadas na Figura 5.36.

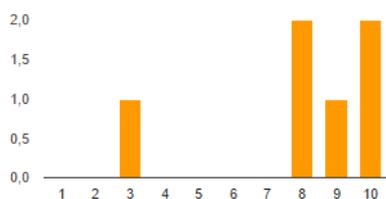
No seu dia-a-dia, o quão útil você considera a ferramenta?



Nada útil: 1	0	0%
2	0	0%
3	0	0%
4	0	0%
5	0	0%
6	0	0%
7	0	0%
8	2	33.3%
9	1	16.7%
Muito útil, não sei como fiquei sem essa solução até agora: 10	3	50%

Muito útil, não sei como fiquei sem essa solução até agora: 10 3 50%

Qual nota você daria em relação a dificuldade de utilizar a ferramenta?



Muito Difícil: 1	0	0%
2	0	0%
3	1	16.7%
4	0	0%
5	0	0%
6	0	0%
7	0	0%
8	2	33.3%
9	1	16.7%
Muito Fácil: 10	2	33.3%

Muito Fácil: 10 2 33.3%

Figura 5.37 - Resultados obtidos com as perguntas realizadas

Esse formulário foi aplicado com os clientes reais, que utilizaram a ferramenta. Ele foi feito *online* através do *Google Forms*⁹, e cada *startup* foi responsável por disponibilizá-lo para seus usuários. Como algumas *startups* não tiveram usuários utilizando a ferramenta, elas não obtiveram nenhum resultado, como é o caso do grupo Impressão Distribuída. Na Tabela 5.9 - Resultado obtido com as notas dadas por usuários são listados os resultados obtidos com o formulário disponível no Apêndice E.

⁹ <https://www.google.com/forms>

Tabela 5.9 - Resultado obtido com as notas dadas por usuários

	<i>Caregiver</i>	<i>Dover</i>	<i>Factoring Sys</i>	Imp. Distrib.
Utilidade da ferramenta	10	8 8	10	--
Dificuldade de utilização	8	10 9	10	--
Atende ao propósito	8	9 10	8	--
Encontrou Problema?	Não	Sim Sim	Não	--
Impacto das falhas	6	2 2	2	--
Recomendação de uso	10	10 9	10	--
Avaliação Geral	9	9 9	9	--
M1	6,5	7,25	7,5	--

A descrição das notas dadas pelos usuários pode ser obtida na seção 5.4.

5.5 ANÁLISE E INTERPRETAÇÃO DOS RESULTADOS

5.5.1 Análise Qualitativa

Analisando os dados entregues pelas equipes que utilizaram o SK, em comparação com as demais equipes que participaram do experimento, na Tabela 5.8 - Resultado obtido com os testes realizados de cada grupo, é possível destacar a quantidade de requisitos testados de cada grupo. Além da quantidade, é possível perceber uma diferença na qualidade de sua descrição.

Enquanto as equipes *Caregiver* e *Dover*, levantaram 5 e 17 requisitos (ou histórias) respectivamente, as equipes Impressão Distribuída e *Factoring Sys*, apresentaram 7 requisitos cada. Em comparação com a quantidade de falhas encontradas, é possível perceber que a forma com a qual as equipes que utilizaram o SK descreveram seus requisitos, influenciou diretamente a quantidade de falhas encontradas. Essa afirmação é possível ser feita, já que foram encontradas 4 falhas no *software* do grupo *Caregiver* e 10 falhas no *Dover*, contra 2 falhas no *Factoring Sys* e nenhuma falha no Impressão Distribuída.

Portanto, ao comparar os requisitos definidos pelas equipes que utilizaram o SK, com as equipes que não utilizaram, é possível perceber que as primeiras equipes demonstraram uma maior riqueza de detalhes. Levando em consideração os critérios de análise definidos na seção 5.2.4 Procedimento de Análise, pode-se destacar que os requisitos definidos pelos grupos que utilizaram a metodologia apresentaram:

- (1) Formato e padronização – Ambos os grupos *Dover* e *Caregiver* seguiram o formato definido pelo *Startup Kaizen*, enquanto os outros grupos não apresentaram nenhum padrão. Por exemplo, o grupo Impressão Distribuída forneceu uma lista de passo-a-passo para o grupo *Factoring Sys* executar, enviando o resultado dos testes;
- (2) Quantidade de informações fornecidas – Tomando como base somente a quantidade de requisitos, os grupos que aplicaram a metodologia forneceram

uma quantidade maior com um total de 21 requisitos contra 14 fornecidos pelos grupos que não utilizaram a metodologia. Além disso, as informações fornecidas possuem mais detalhes em comparação com as equipes que não utilizaram a metodologia, tais como: persona, cenário e ação. Como é possível perceber nas Figuras Figura 5.38 e Figura 5.39, considerando apenas os requisitos, o grupo Impressão Distribuída apenas listou atividades que devem ser realizadas. Nelas, não estão contidas as personas (quem) e os cenários (como). Outro ponto que pode ser destacado na diferença das informações, está relacionado à descrição dos requisitos. Na Figura 5.40 é possível perceber um maior detalhamento da descrição, através da inserção de informações do contexto do uso da funcionalidade, bem como a forma que o sistema irá se comportar durante sua utilização. O mesmo não ocorre nas descrições dos grupos *Factoring Sys* (Figura 5.41) e Impressão Distribuída (Figura 5.38). Estes grupos apenas apresentaram uma breve descrição da ação que deve ser realizada e/ou informações para realiza-las.

- (3) Resultado dos testes realizados – Os grupos que utilizaram a metodologia obtiveram melhores resultados na quantidade de falhas encontradas, devido ao maior detalhamento das informações para os testes. É possível observar esses dados na Tabela 5.8 - **Resultado obtido com os testes realizados de cada grupo**

Portanto, pode-se afirmar que a riqueza de detalhes fornecidos pelos grupos que aplicaram a metodologia, favoreceu a realização dos testes, fazendo com que estes testes obtivessem um resultado maior na quantidade de falhas encontradas.

#	Histórias ou Requisitos	Descrição
1	Realizar cadastro de clientes e parceiros	Dentro da área específica, efetuar o cadastro
2	No cadastro correspondente, realizar as alterações de dados	Depois de efetuado o login, é possível editar os seus dados
3	Efetuar um pedido	Completar um ciclo completo de compra, desde a realização da compra até o pagamento
4	Consultar o histórico do pedido	Existe a possibilidade de acompanhar o pedido, tanto pelo cliente assim como pelo parceiro

Figura 5.38 - Exemplo de requisitos criados pela equipe Impressão Distribuída

ID	HISTORIA	DESCRICAO
1	Eu [Murilo, estudante universitário] gostaria de [ACESSAR] o cadastro de rotas para que eu possa [REGISTRAR ROTAS E HORÁRIOS] para transportar encomendas	Cadastro de endereços de origem e destino, meio de transporte e horários aos quais o trajeto será percorrido pelo transportador, sera as informações de rotas e horários existentes.
2	Eu [Joana, vendedora] quero ter a opção de [ESCOLHER] qual é o melhor [MODELOS DE ENTREGA] de acordo com a minha necessidade	Quem transporta pode cadastrar as formas de entrega que aceita, elas são entrega a local definido, postagem em correio local, agendamento de local-horario com quem vai receber.
3	Eu [Joana, vendedora] preciso [PROCURAR] pessoas que tenham disponibilidade para [TRANSPORTAR ENCOMENDAS] toda semana	Formulario que lista as possiveis rotas de envio e quem faz essa rota com em diferentes horarios. Permitindo a quem deseja enviar realizar um agendamento. As rotas tem origem e destino definidos.
5	Eu [Carlos, contador] quero [TIRAR] uma foto do produto que desejo enviar para que conste a [FOTO NO STATUS] da minha solicitação de entrega	Foto de marcação tirada no momento em que o produto passa de mãos. Primeiro de quem deseja enviar para o transportador e depois do transportador para quem recebe. As fotos servem de validação para o inicio e finalização da entrega.

Figura 5.39 - Exemplo de requisitos criados pela equipe *Dover*

#	Histórias ou Requisitos	Descrição
1	Como Bianca, técnica de administração, preciso estar de olho no estoque de medicamentos para poder fazer novos pedidos.	Conforme os enfermeiros ministrarem a medicação e apontarem isso no mobile o remédio é decrementado do banco de dados e quando o estoque chegar em determinados níveis alguns avisos são disparados para que Bianca possa fazer novos pedidos.
2	Como Andreia, auxiliar de enfermagem, preciso estar de olho no estoque de medicamentos para garantir que sempre que for ministrar um medicamento em um determinado idoso este medicamento estará disponível.	Conforme os enfermeiros ministrarem a medicação e apontarem isso no mobile o remédio é decrementado do banco de dados e quando o estoque chegar em determinados níveis alguns avisos são disparados. Dessa maneira o administrativo pode fazer novos pedidos e Andreia não vai se preocupar com falta de medicamentos em estoque
3	Como Bianca, técnica administrativa, preciso fazer novos pedidos de medicamentos conforme o estoque for acabando.	Disponibilizar para as enfermeiras as informações de todos os idosos de maneira rápida e fácil pelo aplicativo e pelo sistema web.
4	muitos idosos, e eventualmente posso esquecer informações importantes	Existe a possibilidade de acompanhar o pedido, tanto pelo cliente assim como pelo parceiro
5	Como Andréia auxiliar de enfermagem preciso garantir que ministrei todos os medicamentos dentro e fora do padrão nos horários corretos.	Disparar alarmes programados para avisar do horário de aplicação de um determinado remédio em um determinado idoso para um determinado horário.

Figura 5.40 - Exemplo de requisitos criados pela equipe *Caregiver*

#	Histórias ou Requisitos	Descrição
1	Validar e verificar o site entregue	Site desenvolvido para divulgação do sistema como um todo
2	Site:	http://ec2-54-69-185-3.us-west-2.compute.amazonaws.com:8080/factory/
3	Efetuar o Login através do site	No canto esquerdo no menu encontra-se o login
4	Efetuar o login como uma empresa	A empresa poderá lançar documentos para solicitar créditos
5	Usuario:	cliente@test.com
6	Senha:	123456
7	Cadastrar uma nova proposta	Cadastrar uma nova proposta, as factorings cadastradas tem um valor minimo para receber essa proposta sendo apenas superior a 50000, assim sendo uma proposta menor não irá aparecer para a factoring. Lance quantas propostas forem necessárias para o teste. Ao final do cadastro a proposta pode ser editada.
8	Deslogar do sistema e retornar como uma factoring	

Figura 5.41 - Exemplo de requisitos criados pela equipe *Factoring Sys*

O grupo *Dover* possui o dobro de requisitos listados (14) em relação ao grupo Impressão Distribuída (7), o que auxiliou na execução de testes mais detalhados, facilitando a identificação de falhas. Além disso, apesar da cobertura dos testes em relação a quantidade de requisitos ser 100% em três grupos (*Dover*, *Factoring Sys* e Impressão Distribuída), fazendo uma análise proporcional, é possível constatar que a cobertura foi maior nos grupos que listaram mais requisitos (14 contra 7).

Nas soluções criadas, foram encontradas um total de 10 falhas para o grupo *Dover*, 4 para *Caregiver*, 1 para *Factoring Sys* e nenhuma para Impressão Distribuída. O resultado obtido, mais uma vez mostra que o detalhamento dos requisitos, conforme descrito na metodologia, favorecem os testes e a identificação de falhas.

Além disso, a média de falhas encontradas por requisitos é maior no grupo *Dover* do que nos demais. Enquanto na solução apresentada pelo pessoal do *Dover*, foram encontradas uma média de 75% de requisitos com falhas (número total de falhas encontradas / número total de requisitos), nas equipes Impressão Distribuída e *Factoring Sys*, foram encontradas menos de 30%.

Considerando esses dados, é possível dizer que a quantidade e o detalhamento das informações fornecidas pelas equipes que aplicaram o *Startup Kaizen*, proporcionaram a realização de testes encontrando maior quantidade de falhas. Portanto, considera-se que as falhas encontradas antes da liberação para o cliente, fez com que elas não o impactasse.

Além disso, a métrica de índice de qualidade definida pelo SK, está diretamente relacionado a quantidade de falhas encontradas, já que um requisito que possui uma falha, é

marcado como “Não Passou”. Desta forma, ao final do ciclo de testes, aqueles *softwares* que possuem índices de qualidade baixos, devem passar novamente por um ciclo de testes, até que o índice seja aceitável para a liberação para o usuário. Fazendo com que um limite de tolerância seja estabelecido, evitando prejuízos para a empresa com falhas no ambiente de produção.

Analisando os resultados obtidos, é possível perceber que a realização de mais testes favorece a obtenção de mais falhas, evitando com que elas sejam encontradas pelo o usuário em produção. Por isso, é possível afirmar que os grupos que utilizaram a metodologia, passarão por mais rodadas de testes antes da liberação para o usuário, enquanto o restante liberará o *software* com mais falhas ocultas devido à falta de informações ocasionadas pela falta da utilização de um processo estruturado.

O formato dos requisitos listados pelos grupos que não utilizaram a metodologia, não foi padronizado. Desta forma, o grupo *Factoring Sys* criou uma sequência de passos para a outra equipe executar durante a etapa dos testes. Ao criar uma sequência de etapas para a realização de um teste, os executores se tornam viciados e muitas vezes falhos já que consiste num teste já executado pela equipe desenvolvedora.

Outro ponto que deve ser analisado conforme a evolução dos ciclos de desenvolvimento, é o impacto da qualidade nos clientes. Essa métrica, assim como o índice de qualidade, tende a variar conforme outros ciclos de desenvolvimento ocorrem, fornecendo mais dados sobre o possível impacto no usuário final. Desta forma, com mais ciclos é possível obter mais dados e informações sobre a relação entre o índice de qualidade e o impacto no cliente, já que conforme o índice for evoluindo, as notas dadas pelos clientes também tendem a serem alteradas.

Tomando como base os resultados relatados anteriormente, é possível analisar as perguntas definidas nesse estudo de caso:

Q1: A utilização da metodologia proporcionará um desenvolvimento do software mais estruturado?

Sim. A utilização da metodologia favoreceu os grupos que a utilizaram. O *Startup Kaizen* auxiliou na definição do escopo de desenvolvimento, qualidade das informações obtidas, e estruturação da validação diretamente com seus potenciais clientes. Desta forma, os grupos *Caregiver* e *Dover* conseguiram estruturar um processo de validação com base em requisitos, fazendo uma conexão direta com o desenvolvimento da solução proposta.

Além disso, a utilização de personas e cenários de uso proporcionaram um maior conhecimento sobre seus potenciais clientes, bem como a forma de utilização da ferramenta. As duas equipes formularam critérios de avaliação da validação, e o mesmo não foi possível verificar com as equipes que não utilizaram o SK.

Esses critérios ajudaram na tomada de decisão do que era mais importante, dando mais agilidade ao processo e focando no que era essencial. Durante as apresentações em sala de aula essa diferença ficou clara, dando mais embasamento aos dados apresentados durante a aplicação do estudo de caso.

Q2: A utilização da metodologia proporcionará métricas para tomada de decisão na liberação do software para os usuários?

Sim. A utilização da metodologia auxiliou os membros das equipes na tomada de decisão de quais requisitos (ou histórias) deveriam ou não ser inseridas no *software* em desenvolvimento. Esse tipo de decisão é fundamental dentro de uma *startup*, já que muitas vezes funcionalidades são desenvolvidas e não são utilizadas ou bem aceitas por seus clientes. Essa não utilização causa prejuízo em relação a tempo e dinheiro, durante a criação de uma nova empresa.

Portanto, os dados obtidos com a validação e os testes deixaram o processo de criação de uma *startup* menos empírico e mais baseado em dados. A tomada de decisão

passou então, a ocorrer também com base nos dados gerados, e não apenas na experiência do empreendedor.

Q3: A utilização da metodologia influenciará na qualidade final do *software* desenvolvido?

Não necessariamente. Tomando como base a padronização, inserção de métricas e qualidade das informações fornecidas durante o processo de desenvolvimento, é possível dizer que sim. Porém, tomando como base a métrica definida para análise dessa questão, quantidade de falhas encontradas e índice de qualidade de *software* não é possível confirmar essa pergunta.

Neste caso, para analisar a pergunta para obtenção de melhores resultados, é necessário realizar outros ciclos de desenvolvimento. Durante esses ciclos é possível verificar se conforme os ciclos forem evoluindo, os grupos que aplicaram a metodologia, liberariam menos falhas para seus usuários finais.

Portanto, a aplicação do *Startup Kaizen* auxiliou as *startups* numa melhor estruturação do processo de desenvolvimento durante a fase de validação. Através das inserções de boas práticas de Engenharia de *Software* nas metodologias de desenvolvimento de *startups* teve uma grande influência na geração de dados e resultados quantitativos para a tomada de decisão. Além disso, é pode-se dizer também que auxiliou na redução de custos, já que somente funcionalidades validadas foram inseridas no *software* liberado para o cliente.

5.5.2 Conclusão

O experimento ocorreu com etapas e entregas bem-definidas. Durante essas etapas, foi possível perceber uma evolução nas equipes, além de uma diferenciação na estruturação e conhecimento dos projetos trabalhados.

O experimento proporcionou uma nova forma de organização e estruturação de novos projetos de desenvolvimento, através do conhecimento das necessidades dos usuários com entregas pequenas e constantes. Os resultados obtidos demonstram que é possível construir uma tecnologia através da participação mais próxima dos usuários, de forma rápida e estruturada.

- 6 De forma geral, pode-se dizer que a utilização da metodologia trouxe benefícios no desenvolvimento do software para startups. A sua utilização proporcionou uma melhor estruturação do projeto de desenvolvimento, um contato direto com o usuário para validação de requisitos, realização de testes e identificação de falhas antes da liberação para o usuário.**

6 CONCLUSÃO E TRABALHOS FUTUROS

As técnicas de engenharia de *software* aplicadas a um modelo de desenvolvimento mais ágil com foco no desenvolvimento de inovação podem trazer benefícios para as *startups* e até para as grandes corporações. Com um desenvolvimento mais organizado e estruturado, o desperdício de tempo com correção de problemas e manutenção de código pode ser reduzido facilitando o dia-a-dia num ambiente de mudanças constantes, além de aumentar a qualidade do produto final.

Reduzindo a possibilidade de problemas com a parte técnica fará com que as *startups* foquem suas atenções na descoberta de mercado e clientes. Além disso, o fator qualidade poderá deixar de ser um fator duvidoso quando se trata de medir o sucesso de uma *startup*.

O estudo das metodologias para a criação do *Startup Kaizen*, iniciou-se pelo interesse da inserção de boas práticas de engenharia de *software* visando somente a qualidade final do produto. Conforme a evolução dos estudos e do projeto, foi possível perceber que a realização de um projeto estruturado e bem validado poderia impactar diretamente os resultados finais.

Portanto, além de boas práticas relacionadas a validação, verificação e testes de *software*, foi realizado um apanhado de boas práticas utilizadas tanto pela engenharia de *software* tradicional, quanto das metodologias utilizadas pelas *startups*. Essa união de boas práticas deu-se já que um dos principais pilares das *startups* consiste em não construir um produto que não terá um público alvo bem definido, disposto a pagar por ele.

Desta forma, fez-se necessário inserir noções de modelagem de negócios, através da utilização de ferramentas como o BMC, além das boas práticas para um melhor gerenciamento do projeto e obtenção de um *software* com maior qualidade. Essas boas práticas visam obter um resultado enxuto, sem fazer com que a *startup* perdesse tempo com atividades, artefatos e ferramentas que pudessem atrasar seu desenvolvimento e diminuir sua velocidade.

O *Startup Kaizen* fez a união de boas práticas de Engenharia de *Software* juntamente com as metodologias utilizadas por *startups*, unindo agilidade, negócios e estruturação de um processo de desenvolvimento. As metodologias utilizadas por *startups*, apesar de citarem boas práticas, elas não são detalhadas. Por exemplo, nenhuma das metodologias *Lean Startup*, *Customer Development* e *Design Thinking* apresenta uma fase de Requisitos tão definida e com gerenciamento de requisitos criados.

Além disso, a fase de Validação traz ferramentas para auxiliar na estruturação da validação, através de um planejamento inicial visando a obtenção de dados para a tomada de decisão. As demais fases de Projeto e Implementação, trazem como pontos principais, o planejamento e gerenciamento do projeto através da rastreabilidade de requisitos. Essa rastreabilidade permite um maior conhecimento do *software* desenvolvido, bem como o impacto das alterações realizadas nele. Portanto, com a rastreabilidade é possível realizar um processo de manutenção e estruturação do *software* desenvolvido.

A utilização da metodologia SK com a união de modelagem de negócios e engenharia de *software*, auxiliou na estruturação do desenvolvimento de soluções tecnológicas inovadoras para problemas existentes, mostrou-se vantajosa durante o processo de desenvolvimento. A inserção de boas práticas de engenharia de *software* a metodologias utilizadas por *startups* trouxe vantagens em relação a estruturação e velocidade no desenvolvimento dessas soluções.

A estruturação possibilitou um acompanhamento melhor dos resultados obtidos, além facilitar a medição de indicadores através de métricas bem-definidas. As métricas mapeadas e utilizadas para medição de resultados podem ser utilizadas no dia-a-dia das *startups*, fornecendo informações importantes para a tomada de decisão.

As equipes que utilizaram a metodologia apresentaram dúvidas e dificuldades na utilização das ferramentas fornecidas. Além disso, também foi necessário um acompanhamento de perto para sanar as dúvidas geradas durante o processo. Portanto, torna-se necessária realizar uma melhoria na ferramenta apresentada, tornando-a mais didática para possibilitar sua utilização sem acompanhamento.

Um dos *feedbacks* gerados pela utilização do SK já resultou na criação de uma nova ferramenta de auxílio. Visando facilitar o desenvolvimento da fase de Requisitos, foi criado um *roadmap* com um passo-a-passo com perguntas iniciais para facilitar a estruturação da ideia trabalhada. A Figura 6.1 ilustra o *roadmap* criado, ele foi aplicado com *startups* fora do contexto analisado nesta dissertação.



Figura 6.1 - Roadmap criado para estruturação da ideia na fase de Requisitos

Além da ferramenta, a falta de conhecimento em relação as metodologias utilizadas como base para a criação do *Startup Kaizen*, também dificultaram a sua aplicação por parte dos grupos selecionados. Outro ponto a ser destacado, é o fato das equipes estarem numa sala de aula desenvolvendo um trabalho acadêmico, sem necessidade de gerar uma nova empresa.

Uma análise dos pontos levantados acima deve ser feita visando a obtenção de melhores resultados para a avaliação e validação da metodologia criada. Portanto, faz-se necessário a aplicação da metodologia fora de sala de aula, com *startups* que estejam no mercado e queiram validar sua ideia de forma estruturada e ágil. Essa aplicação visa maior riqueza nos dados para análise dos resultados da utilização da metodologia através da obtenção de uma maior quantidade de respostas de usuários.

Existem também pontos de melhoria dentro da própria metodologia que podem ser realizados. Por exemplo, o SW Canvas não se mostrou muito útil para a evolução do processo de desenvolvimento e validação da *startup*. Portanto, esta ferramenta poderá ser removida e/ou substituída por outra que auxilie durante a validação das hipóteses levantadas.

O *Startup Kaizen* possui uma limitação em relação à equipe de trabalho. Para segui-lo, é necessário que a equipe tenha um conhecimento prévio de desenvolvimento e Engenharia de *Software* para facilita sua aplicação por parte do grupo.

Uma ferramenta do *Startup Kaizen* que se mostrou bastante útil, foi a guia de Planejamento da Validação. Com ela, foi possível realizar uma validação mais estruturada,

através da definição de hipóteses e resultados esperados. Ao planejar a validação, os membros da equipe foram forçados a construir experimentos para obtenção de dados por parte do cliente. E, além da obtenção desses dados, através do resultado esperado fica mais fácil eliminar e/ou confirmar as hipóteses estabelecidas.

Outro ponto de melhoria a ser analisado, é a questão da definição de histórias no primeiro momento, quando a ideia é estruturada. No SK, esse é um dos artefatos de saída da fase de Requisitos, porém, nos primeiros ciclos a definição dessas histórias e/ou requisitos pode não ser necessária, já que existem diversas formas de simular um MVP com ou sem a construção de um *software* de apoio.

Desta forma, ao invés de definir as primeiras histórias e/ou requisitos, é possível apenas fazer uma descrição simples das funcionalidades para análise de impacto *versus* dificuldade. Essa análise tem como objetivo a definição do que será o MVP criado, sem necessariamente realizar o desenvolvimento de um MVP que seja um *software*, podendo ser a simulação da solução através de ferramentas já existentes.

Para trabalhos futuros, recomenda-se o estudo de outras ferramentas utilizadas para validação de mercado, tal como *Validation Board*, verificando como seria possível inseri-las dentro do *Startup Kaizen*. É possível também verificar a inserção da definição inicial de uma *sprint* de desenvolvimento já desde a primeira fase do SK, em Requisitos. Desta forma, as fases de validação com o cliente também seriam realizadas através da definição de um *timebox* (2 semanas, por exemplo), limitando assim um prazo para a realização da validação e análise dos resultados obtidos.

Por fim, a experiência de aplicação da metodologia mostrou-se vantajosa já que muitas vezes, empreendedores não possuem conhecimento de gestão de desenvolvimento de *software*. Essa falta de conhecimento, acabam influenciando diretamente no sucesso de uma *startup*. Assim como ocorre nas grandes empresas, os projetos de *startups* também correm o risco de não serem finalizados, acabarem atrasando, gerando prejuízos irreparáveis para uma empresa iniciante [CROWNE, 2002].

REFERÊNCIAS

ABRANTES, J. F.; TRAVASSOS, G. H. Caracterização de Métodos Ágeis de Desenvolvimento de Software. In: Proceedings of Workshop de Desenvolvimento Rápido de Aplicações do Simpósio Brasileiro de Qualidade de Software (WDRA - SBQS 2007), Porto de Galinhas, 2007. Disponível em: <<http://reuse.cos.ufrj.br/wdra2007/images/artigos/30188.pdf>> Acesso em; 08 set. 16.

BARBOSA, S.; da SILVA, Bruno. Interação Humano-Computador, Campus, 2010. ISBN 85-352-3418-7.

BECK, K; GRENNING, J; MARTIN, R.C.; BEEDLE, M; HIGHSMITH, J; MELLOR, S; VAN BENNEKUM, A; HUNT, A; SCHWABER, K; COCKBURN, A; JEFFRIES, R; SUTHERLAND, J; CUNNINGHAM, W; KERN, J; THOMAS, D; FOWLER, M; MARICK, B. "Manifesto for Agile Software Development". Agile Alliance. 2010.

BJÖRK, J.; LJUNGBLAD, J.; BOSCH, J. Lean Product Development in Early Stage Startups. IW-LCSP, 2013.

BLANK, S; DORF, B. The Startup Owner's Manual - The Step-by-step for Building a Great Company. K and S Ranch Inc., K&S Ranch Publishing Division, 2012.

BLANK, S. Four Steps to the Epiphany. K&S Ranch; 2nd edition, 2013. ISBN 978-0-9892005-2-3.

BODKER, S; CHRISTIANSEN, E; NYVANG, T; ZANDER, P. Personas, people and participation-challenges from the trenches of local government. 12th Participatory Design Conference: Research Papers - Volume 1, pages 91-100, 2012. ISBN: 978-1-4503-0846-5.

BOENTE, A. N. P.; OLIVEIRA, F. S. G.; ALVES, J. C. N. RUP como Metodologia de Desenvolvimento de Software para Obtenção da Qualidade de Software. SEGeT – Simpósio de Excelência em Gestão e Tecnologia, 2008.

BOTH, L. J. R. G; LOPES, A. R. Estudo de Caso Etnográfico: Um exercício de análise metodológica. IX Congresso Nacional de Educação – EDUCERE, 2009.

BROWN, T. Change by Design. HarperCollins e-books, 1 Edição, 276 p., 2009. Kindle Edition. ASIN B002PEP4EG.

CATARINAS. Como ter sucesso com o Lean Startup + Lean UX. 2014. Disponível em: <<http://interface.catarinasdesign.com.br/como-ter-sucesso-lean-startup-lean-ux>> Acesso em; 24 nov. 14.

CHANG, Y; LIM, Y; STOLTERMAN, E. Personas: From Theory to Practices. 5th Nordic conference on Human-computer interaction: building bridges, pages 439-442, 2008. ISBN: 978-1-59593-704-9.

COHN, M. User Stories Applied: For Agile Software Development. 1. ed. [S.l.]: Addison-Wesley Professional, v. 1, 2004.

CORPORATION, R. S. Rational Unified Process. 2001. Disponível em: <<http://www.wthreex.com/rup/portugues/index.htm>>. Acesso em; 24 nov. 2014.

CROWNE, M. Why software product startups fail and what to do about it. IEEE Engineering Management Conference, pages 338 – 343, 2002. ISBN 0-7803-7385-5.

DIREKOVA, N. Design Sprint Methods. 2015. Disponível em: <<https://developers.google.com/design-sprint/downloads/DesignSprintMethods.pdf>>. Acesso em 08 set 2016.

DYCK, S; MAJCHRZAK, T. Identifying Common Characteristics in Fundamental, Integrated, and Agile Software Development Methodologies. IEEE System Science (HICSS), 45th Hawaii International Conference on System Sciences, pages 5299 – 5308, 2012. Maui, HI. ISBN 978-1-4577-1925-7.

FARNSWORTH, C; KUMAR, J, KENNEDY, S. SIG: Design Thinking Beyond Post-it Notes. CHI Conference Extended Abstracts on Human Factors in Computing Systems, 2016. Pages 1115-1118. ISBN: 978-1-4503-4082-3.

GHISI, J. Como fazer sua startup se destacar dos concorrentes ao pensar em UX. 2016. Disponível em: <<http://catarinadesign.com.br/como-fazer-sua-startup-se-destacar-dos-concorrentes/>>. Acesso em: 08 fev. 2017.

GOOGLE VENTURES. The Design Sprint. 2015. Disponível em: <<http://www.gv.com/sprint/>>. Acesso em; 08 set. 2016.

GOOGLE DEVELOPERS. New Form Factors Sprint Playbook. Disponível em: <<https://developers.google.com/design-sprint/new-form-factors/>>. Acesso em; 08 set. 2016.

JINZENJI, K; HOSHINO, T; WILLIAMS, L; TAKAHASHI, K. An Experience Report for Software Quality Evaluation in Highly Iterative Development Methodology Using Traditional Metrics. Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium. ISBN 978-1-4799-2366-3.

KRUCHTEN, P. The Rational Unified Process: An Introduction. 3 ed. Addison Wesley, 2000. 320 p. ISBN 9780321197702.

LIKKANEN, L. A. et al. Lean UX - The Next Generation of User-Centered Agile Development? NordiCHI '14, Helsinki, 26 Outubro 2014.

MAY, B. Applying Lean Startup: An Experience Report – Lean & Lean UX by a UX Veteran: Lessons Learned in Creating & Launching a Complex Consumer App. IEEE Agile Conference (AGILE), pages 141 – 147, 2012. Dallas, TX. ISBN 978-1-4673-2622-3

MOREIRA, D. O que é uma *startup*? Revista Exame, 2010. Disponível em: <<http://exame.abril.com.br/pme/dicas-de-especialista/noticias/o-que-e-uma-startup>>. Acesso em: 08 set. 2016.

MOULIN, R. Aplicando a técnica de card-sorting em projetos de AI. 2011. Disponível em: <<http://www.designinterativo.etc.br/usabilidade/aplicando-a-tecnica-de-card-sorting-em-projetos-de-ai>>. Acesso em: 08 fev, 2017.

MULDER, S; YAAR, Z. The User is Always Right: A Practical Guide to Creating and Using Personas for the Web. New Riders, 2007. 313 pages. ISBN 0-321-43453-6.

MUNDRA, A; MISRA, S. Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software. IEEE Computational Science and Its Applications (ICCSA), pages 119-123, 2013. Ho Chi Minh City.

NOTRH, D. Introducing BDD. Disponível em: <<http://dannorth.net/introducing-bdd>>. Acesso em 08 fev. 2017.

OSTERWALDER, A; PIGNEUR, Y. Business Model Generation. Self Published, 72 p., 2009. ISBN 978-2-8399-0580-0. Disponível em: <http://www.businessmodelgeneration.com/downloads/businessmodelgeneration_preview.pdf>. Acesso em 08 set. 2016.

OW, S. H.; YAACOB, M. H;. A Survey on Software Quality Assurance – A Malaysian Perspective. In: Information Systems Conference of New Zealand University of Malaya, v. 30, n. 31. 1996. pages 154-163.

PAULA FILHO, W. Engenharia de Software – Fundamentos, Métodos e Padrões. Ed. LTC, Rio de Janeiro, RJ. 2003. 602 p.

PNIEWSKA, J; ADRIAN, W.; CZERWONIEC, A. Prototyping - Is it a More Creative Way for Shaping Ideas. International Conference on Multimedia, Interaction, Design and Innovation, 2013. ISBN: 978-1-4503-2303-1.

POPPENDIECK, M; CUSUMANO, M. A. Lean Software Development: A Tutorial. IEEE *Software*, pages 26 – 32, 2012.

PRESSMAN, R. S. Engenharia de Software. 1 ed. Pearson Makron Books, SP. 2006

PROENÇA, D; NADALI, A; BORBINHA, J. Using the Business Model Canvas to Support a Risk Assessment Method for Digital Curation. Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries, pages 261-262, 2015. ISBN: 978-1-4503-3594-2.

RIES, Eric. The Lean Startup. New York: Crown Business, 2011. ISBN 9788581780139.

SCHWABER, K. Agile Project Management with Scrum. 1. ed. [S.l.]: Microsoft Press, 2004.

SCHWABER, K.; SUTHERLAND, J. The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game. [S.l.]: *Scrum.Org* and *ScrumInc*, 2014.

SHANAND, S; OLABARRIAGA, S. Initial Steps in Analyzing Science Gateways Sustainability Through Business Model Canvas. 9th Gateway Computing Environments Workshop, pages 5-8, 2014. IEEE Press Piscataway, NJ, USA. ISBN: 978-1-4799-7030-8.

SOMMERVILLE, I. Software Engineering. Boston: Pearson Education, v. 9 edition, 2011. ISBN-13: 978-0-13-703515-1.

SOUSA, K. D. de; OLIVEIRA, K. M. de; ANQUETIL, N. **Uso de GQM para avaliar implantação de processo de manutenção de software**. In: Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento, Valdivia, Chile, 2003.

VIANNA, M; VIANNA, Y; ADLER, I.; LUCENA, B; RUSSO, B. Design Thinking: Inovação em Negócios. 1 ed MJV Press, 2012. 162p. Rio de Janeiro, RJ. ISBN 978-85-65424-00-4. Disponível em: < http://livrodesignthinking.com.br/livro_dt_MJV.pdf>. Acesso em; 08 set. 2016.

SINGH, M. U-SCRUM: An Agile Methodology for Promoting Usability. IEEE Agile Conference, pages 555-560, 2008. Toronto. ISBN 978-0-7695-3321-6.

WILLIAMS, L.; MAXIMILIEN, E. M. Assessing Test-Driven Development at IBM. In: Proceedings of the 25th International Conference on *Software Engineering*, pages 564-569, 2003. ISBN ~ ISSN:0270-52570-7695-1877-X

WOLFF, S. Scrum Goes Formal: Agile Methods for Safety-Critical Systems. FormSERA '12 Proceedings of the First International Workshop on Formal Methods in *Software Engineering: Rigorous and Agile Approaches*, Zurich, Switzerland, 02 Junho 2012. 23-29.

APÊNDICES

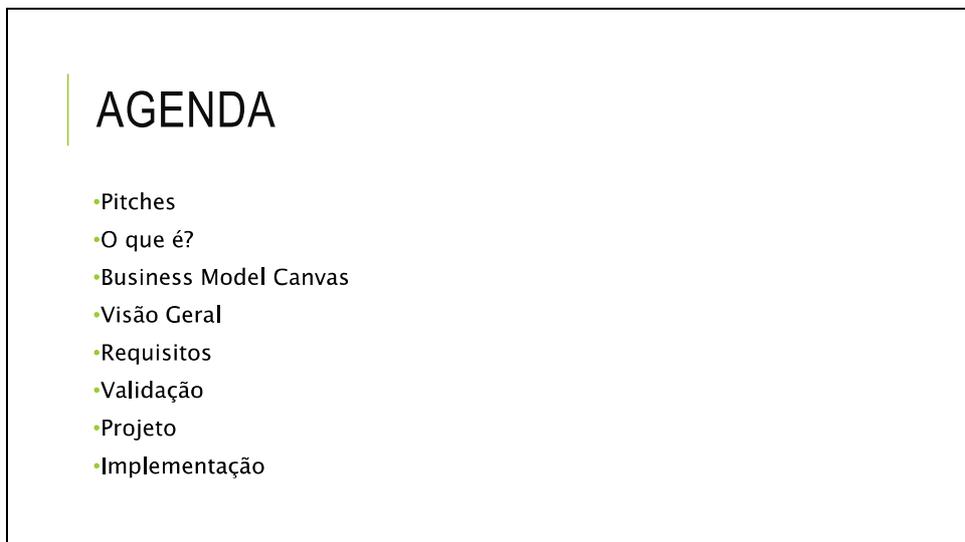
APÊNDICE - A

Apresentação inicial do *Startup Kaizen*.

Slide 1



Slide 2



Slide 3

O QUE É?

Kaizen também é uma palavra japonesa, na qual o Kai significa mudança, e Zen significa para melhor. O sistema kaizen tem como premissa a melhoria contínua e sua filosofia consiste em um importante recurso na busca incessante da melhora de processos produtivos e administrativos, tornando-os mais enxutos e velozes.

Fonte: [Kaizen](#)
Qualidade Brasil – O seu portal brasileiro de Gestão

O diagrama mostra um círculo centralizado com o texto "Startup Kaizen". Setas verdes apontam para este círculo centralizado de sete outros retângulos brancos com bordas verdes, dispostos em um círculo ao redor. Os retângulos contêm os seguintes textos: "Scrum", "Design Thinking", "Qualidade de Software", "Design Sprint", "Business Model Canvas", "Lean Startup" e "Customer Development".

Slide 4

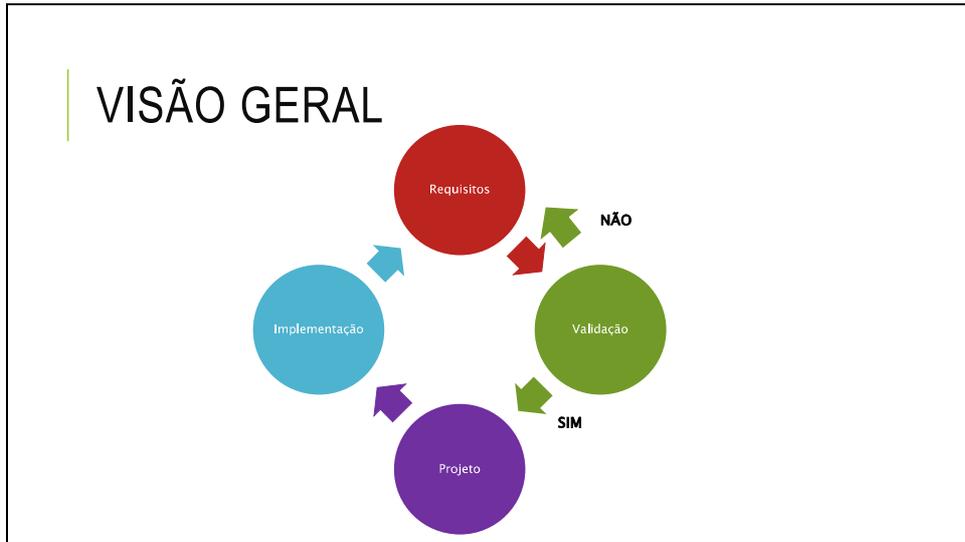
BUSINESS MODEL CANVAS

A tela de modelo de negócios

A tela de modelo de negócios (Business Model Canvas) é uma ferramenta visual para descrever, testar e validar modelos de negócios. Ela é organizada em nove blocos principais:

- Parcerias Chave:** Projeto
- Atividades Chave:** Projeto
- Proposta de valor:** Requisitos
- Relações com clientes:** Validação
- Segmentos de mercado:** Requisitos
- Recursos Chave:** Projeto
- Canais:** Validação
- Estrutura de custos:** Projeto
- Fontes de renda:** Requisitos

Slide 5



Slide 6

REQUISITOS

Toda grande ideia deve ter um pontapé inicial! Entenda qual a real proposta de valor que será entregue para o seu cliente.

Saia com as seguintes perguntas respondidas:

- Qual o MVP da minha ideia?
- Como devo validá-la?
- Qual é a real proposta de valor?
- Quem são meus possíveis clientes?
- Como ganharei dinheiro com a minha ideia?

Um grande círculo vermelho está posicionado à direita do texto, contendo o termo **Requisitos** em letras brancas.

Slide 7

GROOMING - REQUISITOS



Em requisitos, é necessário pensar numa versão inicial e macro dos requisitos do seu software. Eles não precisam de muito detalhamento, preocupe-se mais com a sua funcionalidade, com o resultado que será entregue para o usuário.

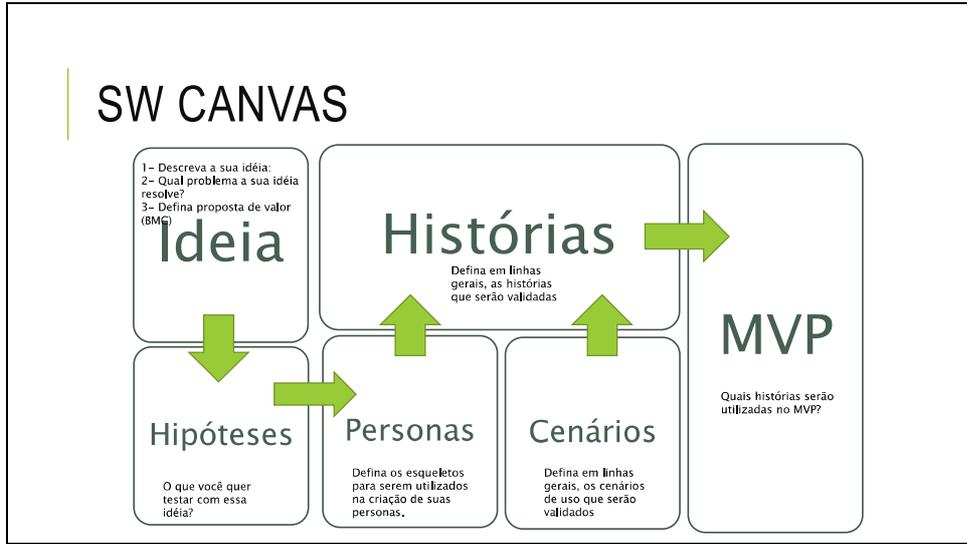
Slide 8

SALTO DE FÉ

Qual o resultado esperado da sua validação?

Defina em forma de números. Ex: monte um formulário para que o usuário diga o quanto ele gostou da funcionalidade avaliada, ou peça que ele coloque em ordem decrescentes das funcionalidades que mais gostou para as que menos gostou.

Slide 9



Slide 10

PERSONAS

Identidade:	Nome, Idade, Sexo, Foto
Status:	primária, secundária, outro stakeholder
Hobbie:	O que essa pessoa gosta de fazer?
Atividades:	Qual é a especialidade da persona? Isso inclui educação, treinamento e competências específicas.
O que a persona gosta de fazer no dia-a-dia?	Em linhas gerais, quais as tarefas básicas ou críticas que a persona gosta de realizar? Qual é a frequência, importância e duração dessas tarefas?
Qual a característica que se destaca nessa persona?	Se fosse resumir essa persona em uma frase, como você
O que você destacaria sobre essa persona?	Qual seu próximo desafio? O que ela mais gosta? O que é um dia perfeito para ela?
Grandes Necessidades	Ela precisa de um jeito de _____ Para obter a experiência de _____ Porque ela valoriza _____

FOTO

Slide 11

PERSONAS

Cynthia | 29 anos

- Estudante de pós graduação em Design de interiores
- Foi para Nova Iorque sozinha fazer intercâmbio e estágio na área
- Gosta de estudar ao ar livre e é adepta de longas caminhadas em parques

Próxima empreitada: Aproveitar a estada nos EUA para fazer um tour.

Grandes necessidades:
 Ela precisa de um jeito de _____
 para obter a experiência de _____
 porque ela valoriza _____



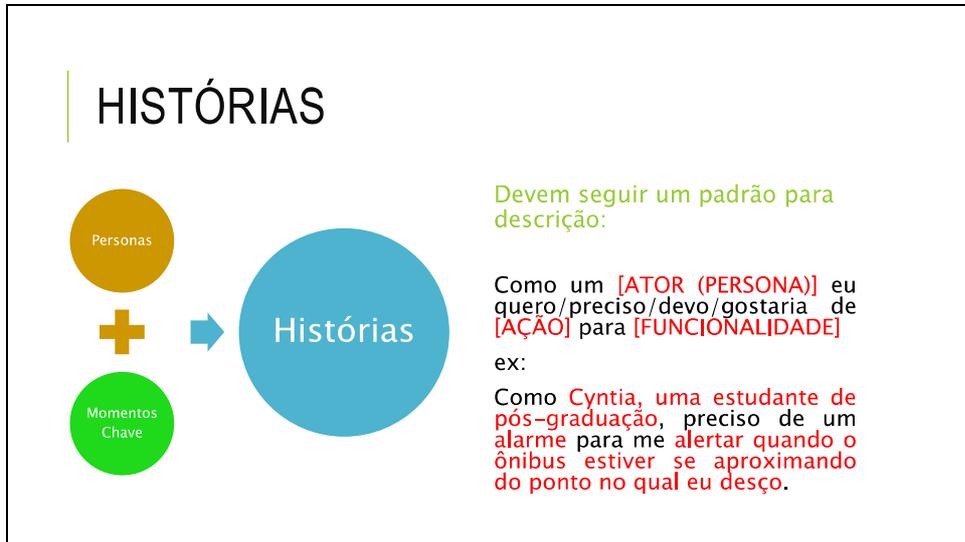
Slide 12

8 MOMENTOS-CHAVE

Defina 8 momentos chaves de uso da sua ideia:

- **Objetivos:** efeitos na situação que motivam as ações realizadas pelos atores;
- **Atores:** pessoas interagindo com o computador ou outros elementos do ambiente; características pessoais relevantes ao cenário;
- **Ambiente** ou contexto: detalhes da situação que motivam ou explicam os objetivos, ações e reações dos atores do cenário;

Slide 13



Slide 14



Slide 15

MVP

QUAIS HISTÓRIAS IRÃO PARA A FASE DE VALIDAÇÃO

Após fazer a classificação das histórias definidas, quais irão para o MVP do seu produto?

Qual é a história principal do seu produto?

Slide 16

VALIDAÇÃO

Depois de entender seus possíveis clientes e funcionalidades, é hora de validar!

Saia com as seguintes perguntas respondidas:

- Quais histórias foram validadas?
- Quais possuem um impacto maior no cliente?
- Quais histórias devem seguir para a etapa de desenvolvimento?



Slide 17

GROOMING - VALIDAÇÃO



Projeto
Validação
Requisitos

Durante a validação, as histórias que serão testadas retornarão com mais detalhes. Esse detalhes devem ser mapeados para seguir para a etapa de projeto.

Slide 18

PLANEJAMENTO DA VALIDAÇÃO

Guia para planejamento da validação

Pré-requisitos:	Quais são os pré-requisitos necessários para que a validação seja executada de forma correta, visando atingir seu objetivo?			
Ambiente:	Qual ambiente será necessário para realizar os testes? Uma sala, laboratório, computador? Qual sistema operacional?			
Ferramentas:	Quais ferramentas serão utilizadas para a validação: protótipo de papel, protótipo de baixa fidelidade, computador?			
Objetivo da Validação:	Qual o objetivo da sua validação? Qual o resultado esperado da sua validação? Defina em forma de números. Ex: monte um formulário para que o usuário diga o quão ele gostou da funcionalidade avaliada, ou peça que ele coloque em ordem decrescentes das funcionalidades que mais gostou para as que menos gostou.			
Salto de fé:				
História	Resultado Esperado	Resultado obtido	Comprova do (s/n)?	Comentários
Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]			---	

Slide 19

APRENDIZADOS

- As histórias, personas e momentos-chave foram confirmados?
- É possível prosseguir para a fase de projeto?
- Quais histórias obtiveram maior interesse do cliente?
- Quais histórias devem ser priorizadas?



Slide 20

PROJETO

Hora de planejar o desenvolvimento e as principais atividade da sua startup!

Saia com as seguintes perguntas respondidas:

- Como as histórias validadas serão desenvolvidas?
- Qual o impacto na inserção de novas histórias?
- Quanto tempo será necessário para desenvolver as histórias selecionadas?
- Quais são as atividades, recursos e parcerias chave da sua startup?



Slide 23

IMPLEMENTAÇÃO

Hora de programar!

Saia com as seguintes perguntas respondidas:

- Quantos bugs foram encontrados por versão de software?
- Quantas histórias foram testadas?
- Quantas versões de software foram liberadas antes da liberação para o cliente?
- Quantos bugs foram corrigidos?



Slide 24

GERENCIAMENTO DE VERSÃO

Controle de versão do software ajuda tanto na qualidade, quanto na opção de retornar versões caso necessário.

Siga o padrão: X.X.X

- X.x.x Major Release – Novas funcionalidades significativas
- x.X.x Minor Release – Melhorias, refatoramentos e evoluções
- x.x.X Revision Release – Correções de bugs

Ferramentas: GIT, Visual Studio, Quality Center, outras

Slide 25

RASTREABILIDADE

Entender ponta-a-ponta o seu software; de requisitos (histórias) até a quantidade de bugs liberadas pelo usuário, e o impacto na aceitação por parte do cliente.



O diagrama mostra um fluxo de três setas horizontais apontando para a direita. A primeira seta é amarela e contém o texto 'Requisitos'. A segunda seta é verde e contém o texto 'Código'. A terceira seta é azul e contém o texto 'Testes'.

Ferramentas: IBM Rational Requisite Pro, Microsoft Visual Studio, HP Quality Center, outras.

Slide 26

BUGTRACKER

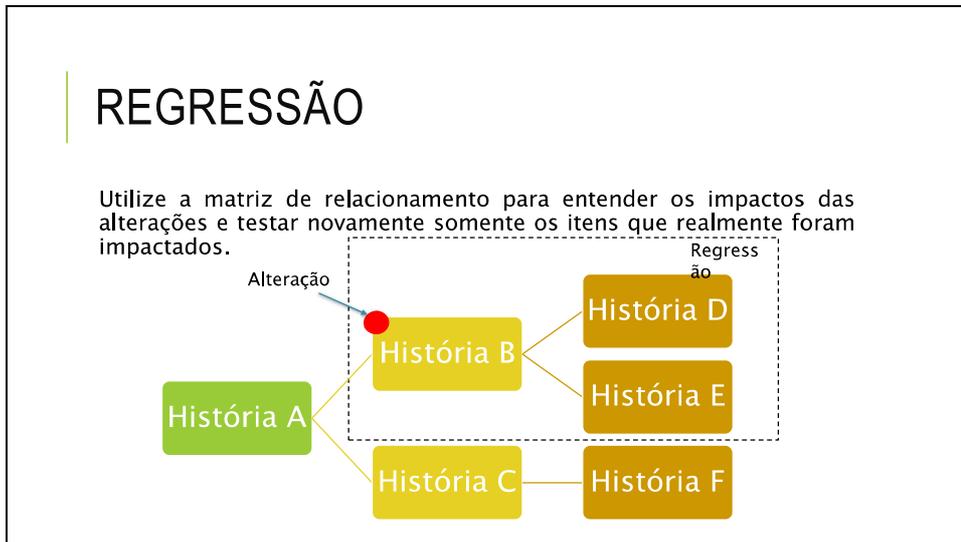


Mapeie e identifique as falhas!

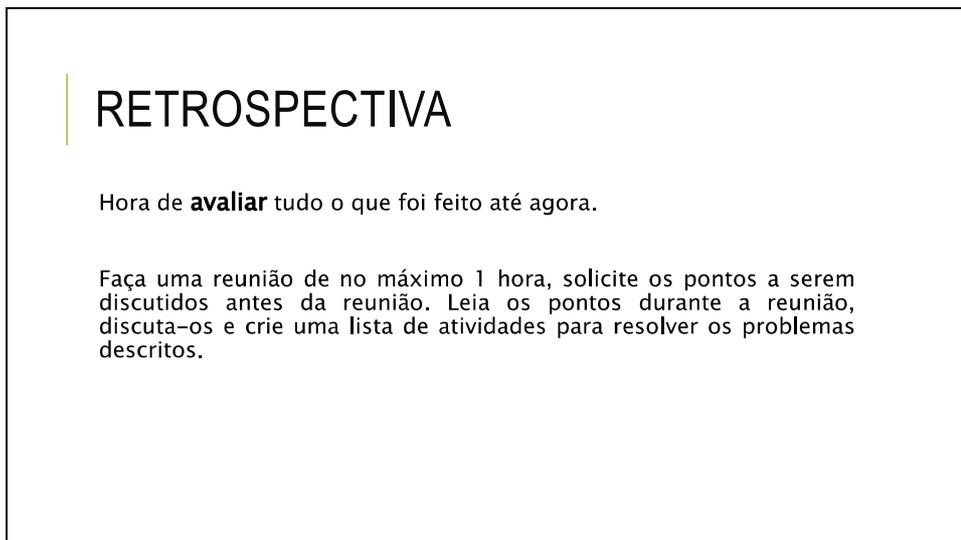
O mapeamento juntamente com a rastreabilidade, controle de versão de software te ajudará a entender o impacto da qualidade gerada no seu possível cliente.

Ferramentas: Mantis, Jira, HP Quality Center, Microsoft Visual Studio, outras.

Slide 27



Slide 28



Slide 29



Startup Sorocaba

Contato:

Nathália Novaes Leonessa

nathalia@startupsorocaba.com

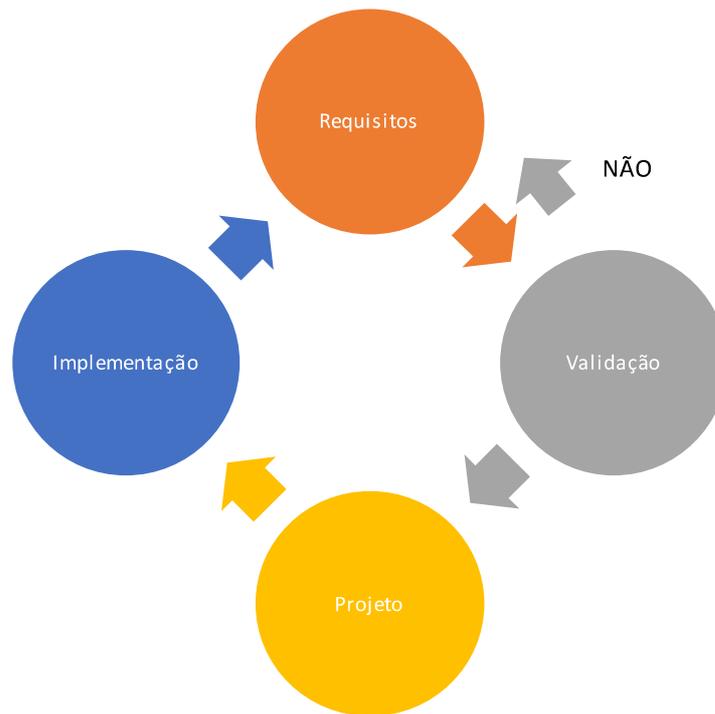
www.startupsorocaba.com

www.facebook.com/startupsorocaba

APÊNDICE – B

Planilha de detalhamento do *Startup Kaizen*.

Aba 1: Processo



Aba 2: Requisitos

Descrição

Nesta etapa, é feita uma análise da ideia com o objetivo de entendê-la melhor, além de verificar sua viabilidade. É uma fase de preparação para a validação, na qual o empreendedor deve entender melhor o seu cliente, as funcionalidades que serão propostas para eles e seu impacto para resolução do problema identificado.

Artefato de Entrada	Atividades	Artefato de Saída	Responsável	Origem	Obrigatório?
Ideia	1- Descrever a ideia:	Ideia do SW Canvas preenchido	Product Owner	Lean Startup	Sim
	2- Identificar problema que a ideia resolve		Product Owner	Lean Startup	Sim
	- Definir proposta de valor	Proposta de Valor do BMC preenchido	Product Owner	BMC	Sim
	3- Entender como os clientes usam/interagem com a ideia	Cenários do SW Canvas preenchido, aba 8 Momentos-chave preenchida	Product Owner	Design Sprint	Sim
	4- Definir os possíveis clientes		Product Owner		Sim
	- Definir e identificar o segmento de clientes	Segmentos de Mercado do BMC preenchido	Product Owner	BMC	Sim
	- Criar Personas	Personas do SW Canvas preenchido, aba Personas preenchida	Product Owner	Design Thinking	Sim
	5- Definir como a empresa ganhará dinheiro		Product Owner	BMC	Sim
	- Definir fontes de Renda	Fontes de Renda do BMC preenchido	Product Owner	BMC	Sim
	6- Classificar o problema de acordo com seu impacto nos possíveis clientes	Problema classificado	Product Owner	RUP	Sim
	7- Descrever os requisitos iniciais da solução em formato de histórias com base nas personas e momentos chave criados	Colunas Histórias e Descrição da aba principal preenchida, Histórias do SW Canvas preenchido	Product Owner	Design Thinking + Scrum	Sim
	8- Classificar as histórias iniciais da solução conforme impacto X complexidade de desenvolvimento	aba Classificação das Histórias preenchida, Coluna Classificação da aba principal preenchida	Product Owner	Design Sprint	Sim
	9- Definir o objetivo da Sprint:		Product Owner	Scrum	Sim
	- O que será testado com essa ideia?	Hipóteses do SW Canvas preenchido	Product Owner	Scrum	Sim
	- Quais histórias serão validadas?	Coluna MVP da aba principal preenchida	Product Owner	Scrum	Sim
	10- Definir as histórias principais que entrarão no MVP	MVP do SW Canvas preenchido	Product Owner	Lean Startup	Sim

Aba
Validação

3:

Descrição:

Esta etapa consiste na validação diretamente com o cliente com o objetivo de obter seus feedbacks sobre os pontos e histórias que foram levantados até a fase de Implementação. Se uma história foi confirmada, ela deverá continuar para a fase de projeto, caso contrário, ela pode ser reformulada ou descartada.

Artefato de Entrada	Atividades	Artefato de Saída	Responsável	Origem	Obrigatório?
Idéia do SW Canvas preenchido	1- Como você vai validar a idéia? - Defina os pré-requisitos para validação: número de pessoas, material necessário, ambiente, etc - Defina o ambiente e ferramentas que serão utilizados durante a validação - Defina o objetivo da validação	Pré-requisitos para validação definidos (Aba Planejamento da Validação)	Product Owner		Sim
Proposta de Valor do BMC preenchido	- Defina o resultado esperado - Defina o Salto de Fé da sua validação com números e resultados esperados	Ambiente e ferramentas para a validação definidos Objetivo da validação definido (Aba Planejamento da Validação)	Product Owner	RUP	Não
Cenários do SW Canvas preenchido, aba 8	- Defina os passos para validação - pode ser de forma macro ou micro	Resultado Esperado da validação definido (Aba Planejamento da Validação)	Product Owner	RUP	Sim
Momentos-chave preenchida	2- Quem são os seus possíveis clientes?	Salto de Fé definido	Product Owner	RUP	Sim
Segmentos de Mercado do BMC preenchido	3- Como você vai chegar até o seu cliente?	Passos da validação definidos (Aba Planejamento da Validação)	Product Owner	Lean Startup	Sim
Personas do SW Canvas preenchido, aba 9	- Definir possíveis canais de comunicação - Definir possíveis formas de relacionamento		Product Owner	RUP	Não
Personas preenchida	4- Faça a validação	Pessoas selecionadas para validação com base nas personas criadas	Product Owner	Lean Startup	Não
Fontes de Renda do BMC preenchido	- Durante validação anote os pontos importantes		Product Owner		
Problema classificado	5- Analise resultados obtidos		Product Owner	Design Thinking	Sim
Colunas Histórias e Descrição da aba principal preenchida, Histórias do SW Canvas preenchido	- Ideia foi validada?	Sim ou Não para as histórias validadas	Product Owner	Lean Startup	Sim
aba Classificação das Histórias preenchida, Coluna Classificação da aba principal preenchida	-- Sim > Siga para Projeto -- Não > Volte para o Requisitos	Coluna Validado da aba Principal preenchida	Product Owner	Lean Startup	Sim
Hipóteses do SW Canvas preenchido			Product Owner	Lean Startup	Sim
Coluna MVP da aba principal preenchida			Product Owner	Lean Startup	Sim
MVP do SW Canvas preenchido			Product Owner	Lean Startup	Sim

Aba 4: Projeto

Descrição:

Nesta etapa é feito o planejamento para as demais etapas através dos resultados obtidos. Nela, as histórias confirmadas serão priorizadas e um planejamento é criado para seu desenvolvimento.

Artefato de Entrada	Atividades	Artefato de Saída	Responsável	Origem	Obrigatório?
Lista de Histórias validadas	1- Classificar e priorizar as histórias validadas em sprints de desenvolvimento		Toda Equipe	Scrum	Sim
Personas confirmadas	- Dividir as histórias em sprints de desenvolvimento, caso necessário	Quantidade de sprints de desenvolvimento que serão realizadas	Toda Equipe	Scrum	Não
Cenários confirmados	2- Verificar o impacto no código com a adição de novas histórias		Toda Equipe	RUP	Sim
Canais de Comunicação confirmados	- Completar ou revisar a matriz de relacionamento (impacta x impactado)	Aba Matriz de Relacionamento preenchida	Desenvolvedor	RUP	Sim
Relações com os Clientes confirmados	3- Definir o tamanho da sprint de desenvolvimento	Quantidade de tempo definida para a sprint de desenvolvimento	Product Owner	Scrum	Não
	4- Definir prazos	Prazo para finalização do desenvolvimento e testes	Toda Equipe	Scrum	Não
	5- Definir Atividades Chave	Atividades Chave do BMC preenchido	Product Owner	BMC	Sim
	6- Definir Recursos Chave	Recursos Chave do BMC preenchido	Product Owner	BMC	Sim
	7- Definir Parcerias Chave	Parcerias Chave do BMC preenchido	Product Owner	BMC	Sim
	8- Calcular Estrutura de Custos	Estrutura de Custos do BMC preenchido	Product Owner	BMC	Sim

Aba 5: Implementação

Descrição:

Nesta etapa é feita a implementação e testes, visando obter um incremento funcional do software e com uma qualidade mínima.

Artefato de Entrada	Atividades	Artefato de Saída	Responsável	Origem	Obrigatório?	Ferramentas
Quantidade de sprints de desenvolvimento que serão realizadas	Desenvolver parte funcional e executável do software	Parte funcional e executável do software	Perfil Técnico	Scrum	Sim	Ferramentas IDE's Ex: Eclipse, Netbeans
Aba Matriz de Relacionamento preenchida	- Gerenciar versão do software - Linkar histórias com linhas de código do software visando rastreabilidade dos requisitos	Versão do software Relação histórias x código (rastreadibilidade)	Perfil Técnico	RUP	Sim	
Quantidade de tempo definida para a sprint de desenvolvimento			Perfil Técnico	RUP	Sim	Depende da linguagem de programação Ex: Junit, PHP Unit, Visual Studio, TestNG, Nunit
Prazo para finalização do desenvolvimento e testes	Realizar testes unitários	Bugs encontrados nos testes unitários	Perfil Técnico	RUP	Não	Ferramentas para integração contínua Ex: Jenkins, etc Plugins de sincronização Ferramentas para automatização de builds Ex: ant, Maven, Nant, etc
Atividades Chave do BMC preenchido	Integração Contínua	Linhas de código desenvolvidas	Perfil Técnico	RUP	Não	Ferramenta de controle de versão centralizado Ex: CSV, GIT
Recursos Chave do BMC preenchido	Gerenciamento de versão	Versão do software	Perfil Técnico	RUP	Sim	Testes de negação, limite, particionamento de equivalência Ferramentas para gerenciamento de testes BugTracker
Parcerias Chave do BMC preenchido	Realizar testes Integração / Funcionais		Perfil Técnico	RUP	Sim	
Estrutura de Custos do BMC preenchido	- Passar história por história verificando se ela foi atendida de forma correta (sem bugs) - Garantir que o cliente consegue executar um ciclo completo passando pelas principais funcionalidades sem bugs	Cobertura de testes	Perfil Técnico	RUP	Sim	
	- Registrar e classificar bugs encontrados	Coluna Testado na aba Principal preenchida	Perfil Técnico	RUP	Sim	
	- Calcular cobertura de testes por histórias	Lista de bugs encontrados	Perfil Técnico	RUP	Não	
	Realizar testes de regressão	Coluna Testado na aba Principal preenchida	Perfil Técnico	RUP	Sim	
	- Verificar se os bugs encontrados foram corrigidos	Bugs verificados e fechados	Perfil Técnico	RUP	Não	Matriz de rastreabilidade
	- Verificar na matriz de relacionamento os impactos sofridos com as alterações	Lista de funcionalidades impactadas com alterações	Perfil Técnico	RUP	Sim	
	- Verificar novamente histórias onde foram encontrados bugs	Histórias onde foram encontrados bugs testadas novamente	Perfil Técnico	RUP	Sim	
	- Verificar novamente histórias que sofreram indiretamente impacto com as correções dos bugs, baseando na análise da matriz de relacionamento		Perfil Técnico	RUP	Sim	
		Histórias impactadas por alterações testadas	Perfil Técnico	RUP	Sim	Ferramentas para automação Ex: Selenium, Test complete
	Automatizar testes	Testes automatizados	Perfil Técnico	RUP	Não	Aba Retrospectiva
	Retrospectiva	Pontos de melhoria e plano de ação	Todos	Scrum	Sim	

Aba 6: Business Model Canvas

A tela de modelo de negócios

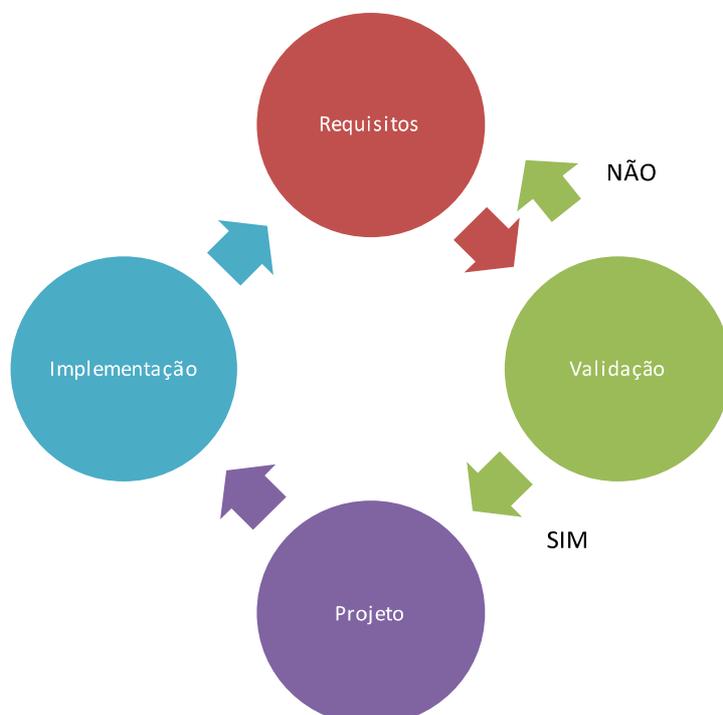
<h3>Parcerias Chave</h3>  <p>Quem são nossos parceiros chave? Por que precisamos deles? Que recursos chave fornecem os nossos parceiros? Que atividades chave realizam nossos parceiros? Como cada integrante em nosso modelo de negócio?</p> <p>EXEMPLOS Parceiros de distribuição Parceiros de produção Parceiros de tecnologia Parceiros de logística</p>	<h3>Atividades Chave</h3>  <p>Que atividades chave requerem nossas propostas de valor? Por que precisamos delas? Quais são as atividades chave mais importantes? Como cada integrante em nosso modelo de negócio?</p> <p>EXEMPLOS Desenvolvimento de produtos Atendimento ao cliente Logística Gestão de operações Gestão de recursos humanos Gestão de tecnologia</p>	<h3>Proposta de valor</h3>  <p>Que valor proporcionamos aos nossos clientes? Por que precisamos disso? Que pontos de produtos ou serviços oferecemos a cada segmento de mercado? Que necessidades dos clientes satisfazemos? Como cada integrante em nosso modelo de negócio?</p> <p>EXEMPLOS Preço mais baixo Qualidade superior Conveniência Personalização Acesso a recursos Acesso a especialistas Acesso a tecnologia Acesso a serviços Acesso a informações</p>	<h3>Relações com clientes</h3>  <p>Que tipo de relação queremos estabelecer com os segmentos de mercado? Que tipo de relações temos estabelecidas? Qual é o custo destas relações? Como cada integrante em nosso modelo de negócio?</p> <p>EXEMPLOS Atendimento ao cliente Autoatendimento Autoatendimento Autoatendimento Autoatendimento Autoatendimento Autoatendimento Autoatendimento Autoatendimento Autoatendimento</p>	<h3>Segmentos de mercado</h3>  <p>Para quem criamos valor? Por que precisamos deles? Que tipo de segmentos de mercado são mais importantes? Como cada integrante em nosso modelo de negócio?</p> <p>EXEMPLOS Mercado de massa Segmentos de nicho Segmentos de nicho</p>
<h3>Recursos Chave</h3>  <p>Que recursos chave requerem nossas propostas de valor? Nossos canais de distribuição? Relações com clientes? Fontes de receita? Tipos de recursos Recursos físicos Recursos humanos Recursos intelectuais Recursos financeiros Recursos tecnológicos</p>	<h3>Canais</h3>  <p>Que canais permitem nossos segmentos de mercado? Como estabelecemos a distribuição e contato com os clientes? Como se conseguem nossos canais? Como cada integrante em nosso modelo de negócio? Quais são os custos dos canais? Como se integram nas atividades diárias dos clientes?</p> <p>EXEMPLOS Loja física Loja física</p>	<h3>Fontes de renda</h3>  <p>Por que valor estão dispostos a pagar nossos clientes? Por que pagam atualmente? Como cada integrante em nosso modelo de negócio? Quanto pagam os diferentes fontes de renda ao total da receita? Como cada integrante em nosso modelo de negócio?</p> <p>EXEMPLOS Preço premium Preço premium</p>	<h3>Fontes de renda</h3>  <p>Por que valor estão dispostos a pagar nossos clientes? Por que pagam atualmente? Como cada integrante em nosso modelo de negócio? Quanto pagam os diferentes fontes de renda ao total da receita? Como cada integrante em nosso modelo de negócio?</p> <p>EXEMPLOS Preço premium Preço premium</p>	<h3>Estrutura de custos</h3>  <p>Quais são os custos mais importantes inerentes ao nosso modelo de negócio? Quais são os recursos chave mais caros? Quais são os maiores custos fixos? Como cada integrante em nosso modelo de negócio? Como cada integrante em nosso modelo de negócio?</p>



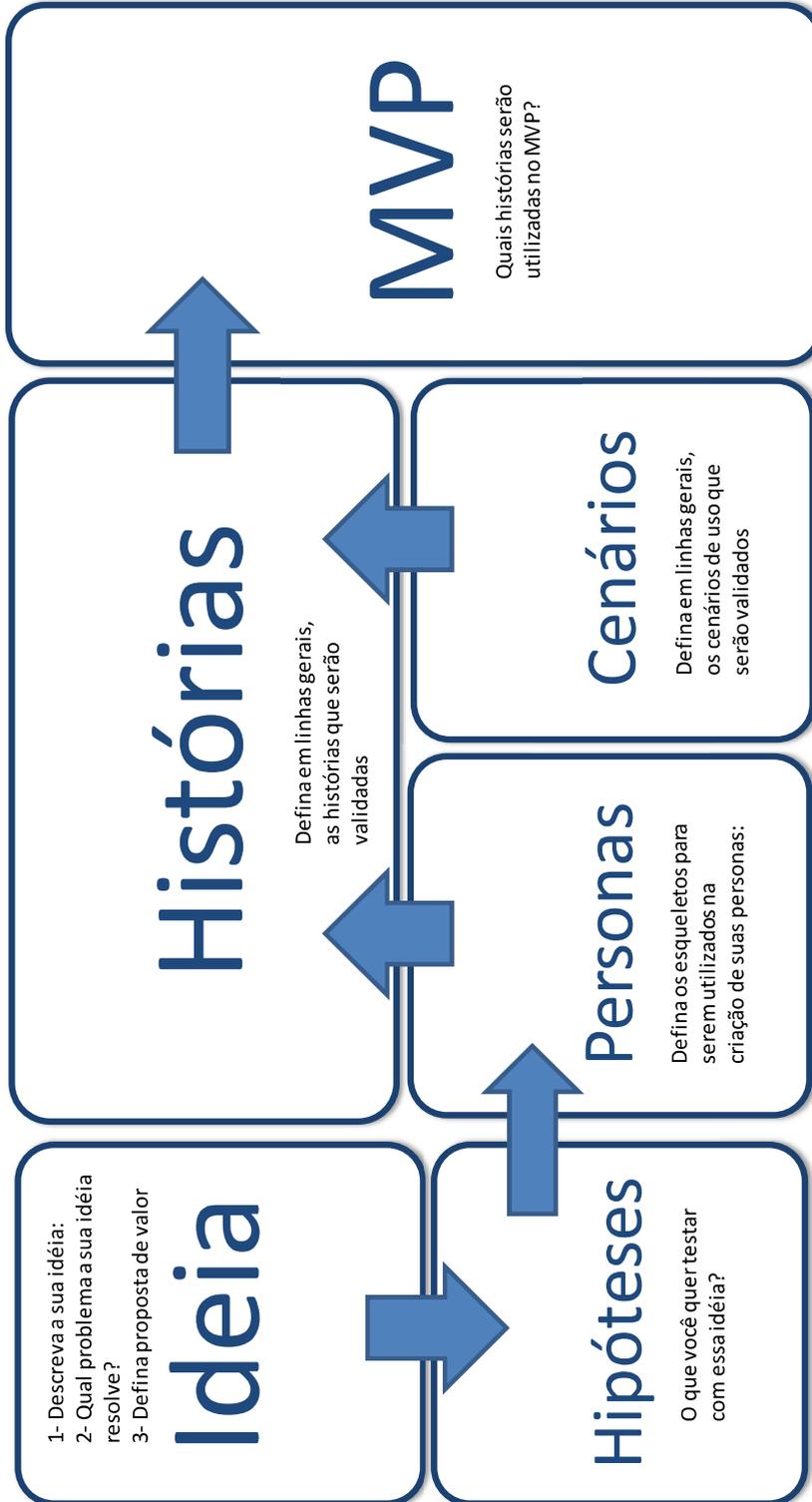
APÊNDICE – C

Planilha de mapeamento dos artefatos gerados pelas equipes com o uso do *Startup Kaizen*.

Aba 1: Processo



Aba 2: SW Canvas



Aba 3: Business Model Canvas

A tela de modelo de negócios

 <h3>Parcerias Chave</h3> <p>Quem são nossos parceiros chave? Quem são nossos provedores chave? Como podemos alcançar nossos parceiros? Que atividades chave nos ajudam a alcançar nossos parceiros? Que atividades chave nos ajudam a alcançar nossos parceiros?</p> <p>CONSIDERAÇÕES - Identificar parceiros chave - Identificar provedores chave - Identificar canais de distribuição - Identificar atividades chave</p> <p style="text-align: center;">Projeto</p>	 <h3>Atividades Chave</h3> <p>Que atividades chave requerem nossas propostas de valor? Que atividades chave requerem nossas propostas de valor? Que atividades chave requerem nossas propostas de valor? Que atividades chave requerem nossas propostas de valor?</p> <p>CONSIDERAÇÕES - Identificar atividades chave - Identificar provedores chave - Identificar canais de distribuição - Identificar atividades chave</p> <p style="text-align: center;">Projeto</p>	 <h3>Proposta de valor</h3> <p>Que valor proporcionamos aos nossos clientes? Que problemas de nossos clientes ajudamos a solucionar? Que benefícios oferecemos aos nossos clientes? Que necessidades dos clientes satisfazemos?</p> <p>CONSIDERAÇÕES - Identificar o valor proposto - Identificar o problema a ser resolvido - Identificar o benefício a ser oferecido - Identificar a necessidade a ser atendida</p> <p style="text-align: center;">Requisitos</p>	 <h3>Relações com clientes</h3> <p>Que tipo de relação queremos estabelecer com nossos clientes? Que tipo de relação queremos estabelecer com nossos clientes? Que tipo de relação queremos estabelecer com nossos clientes? Que tipo de relação queremos estabelecer com nossos clientes?</p> <p>CONSIDERAÇÕES - Identificar o tipo de relação - Identificar o canal de comunicação - Identificar o momento de interação - Identificar o formato de interação</p> <p style="text-align: center;">Validação</p>	 <h3>Segmentos de mercado</h3> <p>Para quem criamos valor? Quem são nossos clientes ideais? Que tipo de segmentos de mercado queremos atingir? Que tipo de segmentos de mercado queremos atingir?</p> <p>CONSIDERAÇÕES - Identificar o segmento de mercado - Identificar o canal de comunicação - Identificar o momento de interação - Identificar o formato de interação</p> <p style="text-align: center;">Requisitos</p>
 <h3>Recursos Chave</h3> <p>Que recursos chave requerem nossas propostas de valor? Nossos canais de distribuição? Relações com clientes? Atividades chave? Parcerias chave?</p> <p>CONSIDERAÇÕES - Identificar os recursos chave - Identificar os canais de distribuição - Identificar as relações com clientes - Identificar as atividades chave - Identificar as parcerias chave</p> <p style="text-align: center;">Projeto</p>		 <h3>Canais</h3> <p>Que canais preferem nossos segmentos de mercado? Como estabelecermos suficiente contato com os clientes? Que canais preferem nossos segmentos de mercado? Como estabelecermos suficiente contato com os clientes? Que canais preferem nossos segmentos de mercado? Como estabelecermos suficiente contato com os clientes?</p> <p>CONSIDERAÇÕES 1. Identificar os canais preferidos 2. Avaliar a eficácia dos canais 3. Avaliar a eficiência dos canais 4. Avaliar a escalabilidade dos canais 5. Avaliar a sustentabilidade dos canais 6. Avaliar a exclusividade dos canais</p> <p style="text-align: center;">Validação</p>	 <h3>Fontes de renda</h3> <p>Por que valor estão dispostos a pagar nossos clientes? Como podemos cobrar? Como podemos cobrar? Como podemos cobrar?</p> <p>CONSIDERAÇÕES - Identificar as fontes de renda - Identificar o canal de cobrança - Identificar o momento de cobrança - Identificar o formato de cobrança</p> <p style="text-align: center;">Requisitos</p>	 <h3>Estrutura de custos</h3> <p>Quais são os custos mais importantes inerentes ao nosso modelo de negócios? Quais são os custos mais importantes inerentes ao nosso modelo de negócios? Quais são os custos mais importantes inerentes ao nosso modelo de negócios?</p> <p>CONSIDERAÇÕES - Identificar os custos fixos - Identificar os custos variáveis - Identificar os custos indiretos - Identificar os custos diretos</p> <p style="text-align: center;">Projeto</p>

Aba 4: Requisitos

Objetivo: Como identificar e definir MVP

Atividades	Técnicas	Ferramentas
1- Descreva a sua idéia:		SW CANVAS
2- Qual problema a sua idéia resolve? - Defina proposta de valor		Business Model CANVAS
3- Como você classificaria esse problema em relação aos possíveis clientes?	Blocante; Grande; Pequeno; Trivial CANVAS - Definir Grupo de Foco, Criar Personas	Business Model CANVAS, Personas Business Model Canvas Personas
4- Quais são os possíveis clientes? - Defina o segmento de clientes - Crie Personas		8 Momentos Chave
5- Como os seus clientes usam/interagem com a sua ideia?		Business Model Canvas
6- Como você ganhará dinheiro com ela? - Defina fontes de Renda		Principal
7- Descreva os requisitos iniciais da sua solução em formato de histórias com base nas personas e momentos chave criados		Classificação das Histórias Planejamento da Validação
8- Classifique as histórias iniciais da sua solução conforme impacto X complexidade de desenvolvimento		
9- Defina o salto de fé: - O que você quer testar com essa ideia? - Quais histórias você quer validar?		Principal
10- Defina as histórias principais que entrarão no MVP		

Aba 5: Validação

Objetivo: Validar as histórias criadas na fase de Requisitos através da aproximação com o cliente

Atividades	Técnicas	Ferramentas
1- Como você vai validar a idéia? - Defina o objetivo da validação - Defina o resultado esperado - Defina os pré-requisitos para validação: número de pessoas, material necessário, ambiente, etc - Defina os passos para validação - pode ser de forma macro ou micro	Protótipo de papel, entrevistas, pesquisas, observação (investigação de campo)	Planejamento da Validação Formulário Google, Balsamiq Mockup
2- Quem são os seus possíveis clientes?		Personas
3- Como você vai chegar até o seu cliente? - Definir possíveis canais de comunicação - Definir possíveis formas de relacionamento	Com base nas personas criadas, identificar pessoas que possuam perfil semelhante	Business Model Canvas Business Model Canvas Guia de Entrevista
4- Faça a validação - Durante validação anote os pontos importantes		Planejamento da Validação
5- Analise resultados obtidos - Ideia foi validada?		Projeto Requisitos
-- Sim > Siga para Projeto		
-- Não > Volte para o Requisitos		

Objetivo: Estruturar requisitos validados para implementação

Técnico	Técnicas	Ferramenta
1- Classifique e priorize as histórias validadas em sprints de desenvolvimento		GitHub, VisualStudio, Jira,
- Divida as histórias em sprints de desenvolvimento, caso necessário		
2- Verifique o impacto no código com a adição de novas histórias		Matriz Relacionamento
- Complete ou revise a matriz de relacionamento (impacta x impactado)		
- Defina o tamanho da sprint		Principal
3- Defina prazos		Principal

Negócios	Técnicas	Ferramenta
1- Definir Atividades Chave		Business Model Canvas
2- Definir Recursos Chave		Business Model Canvas
3- Definir Parcerias Chave		Business Model Canvas
4- Calcular Estrutura de Custos		Business Model Canvas

Aba 7: Implementação

Objetivo: Desenvolver incremento funcional do software

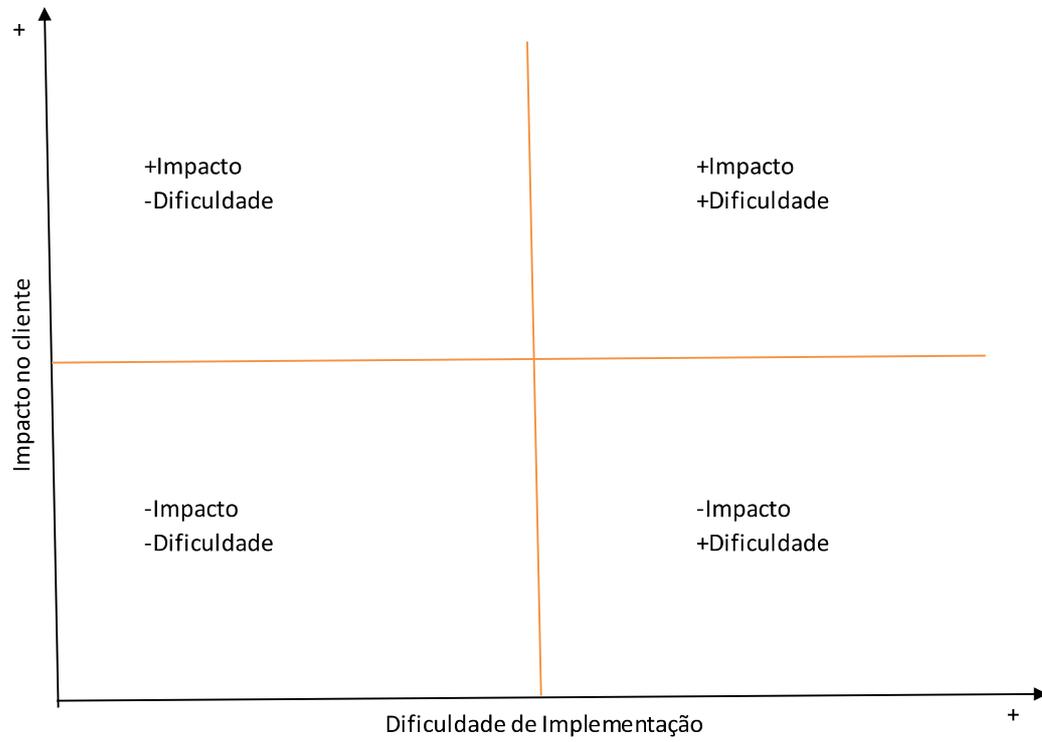
Atividades	Observações	Ferramentas
1- Desenvolver parte funcional e executável do software	Siga planejamento realizado com o backlog de produtos Siga o padrão: X.X.X X.x.x Major Release - Novas funcionalidades significativas x.X.x Minor Release - Melhorias, refatoramentos e evoluções x.x.X Revision Release - Correções de bugs	Utilize uma ferramenta para mapeamento de requisitos e controle de versão de software: GitHub, VisualStudio
- Realize um controle de versão	Garanta a rastreabilidade dos seus requisitos	
- Linkar histórias com linhas de código do software		
2- Realizar testes unitários		
3- Integração Contínua	Validar funcionalidades principais do produto visando garantir que o cliente consiga completar um ciclo em cima dos principais requisitos.	Principal
4- Realizar testes Integração / Funcionais		
- Passar história por história verificando se ela foi atendida de forma correta (sem bugs)		
- Garantir que o cliente consegue executar um ciclo completo passando pelas principais funcionalidades sem bugs		
- Registrar e classificar bugs encontrados		Mantis ou qualquer outra ferramenta de bugtracker
- Calcular cobertura de testes por requisito		Principal
5- Realizar testes de regressão		
- Verificar se os bugs encontrados foram corrigidos		
- Verificar na matriz de relacionamento os impactos sofridos com as alterações		Matriz de Relacionamento
- Verificar novamente histórias onde foram encontrados bugs		
- Verificar novamente histórias que sofreram indiretamente impacto com as correções dos bugs, baseando na análise da matriz de relacionamento		Matriz de Relacionamento
6- Automatizar testes	Automação de testes deve ser feita após a validação de mercado, quando a ideia já estiver consolidada. É primordial começar automatizar processos mais simples e com muita repetição	
7- Retrospectiva	Análise tudo o que ocorreu durante a sprint realizada. O que deu certo? O que deu errado?	Retrospectiva

Aba 8: Principal

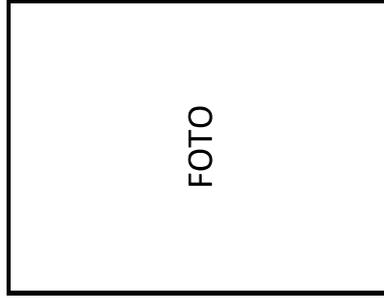
Porcentagem Histórias Validadas 35% Quantidade de Histórias do MVP 0%
 Cobertura de Testes 76% Índice de qualidade 0%

#	História	Descrição	Classificação	Prioridade	MVP	Validado (s/n)?	Sprint	Módulo	Testado (s/n)?	Passou (s/n)?	Bugs
1		Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				--			Sim	--	- Bug 1 - Bug 2 - Bug 3
2		Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				--			Sim	--	
3		Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				--			Sim	--	
4		Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				--			Sim	--	
5		Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				--			---	--	
6		Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				--			---	--	
7		Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				--			---	--	
8		Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]				--			---	--	

Aba 9: Classificação das Histórias



Aba 10:
Personas



Identidade:	Nome, Idade, Sexo, Foto
Status:	primária, secundária, outro stakeholder
Objetivos:	Quais são os objetivos desta persona?
Habilidades:	treinamento e competências específicas.
Tarefas:	Em linhas gerais, quais as tarefas básicas ou críticas que a persona realiza? Qual é a frequência, importância e duração dessas tarefas?
Relacionamentos:	Com quem a persona se relaciona?
Requisitos:	De que a persona precisa?
Expectativas:	Como a persona acredita que o produto funciona? Como ela organiza as informações no seu domínio ou trabalho?
Objetivos pessoais:	
Objetivos Práticos:	

Ator #	Nome	Descrição	Foto
			FOTO
	Ator 1		
	Ator 2		
	Ator 3		
	Ator 4		

Aba 11: 8 Momentos-chave

<p>Defina 8 momentos chaves de uso da sua ideia: ambiente ou contexto: detalhes da situação que motivam ou explicam os objetivos, ações e reações dos atores do cenário; atores: pessoas interagindo com o computador ou outros elementos do ambiente; características pessoais relevantes ao cenário; objetivos: efeitos na situação que motivam as ações realizadas pelos atores;</p>	<p>Cenário 1</p>	<p>Cenário 2</p>	<p>Cenário 3</p>	<p>Cenário 4</p>
<p>Cenário 5</p>	<p>Cenário 6</p>	<p>Cenário 7</p>	<p>Cenário 8</p>	

Guia de Entrevista	
<p>Durante a entrevista, identifique os pontos a seguir:</p> <p>Qual a relação do cliente com tecnologia?</p> <p>Qual o seu conhecimento do domínio do produto?</p> <p>Qual seu conhecimento das tarefas que deverá realizar?</p> <p>Quais suas motivações e valores?</p> <p>Próprio usuário</p> <ul style="list-style-type: none"> ▣ dados demográficos: idade, sexo, status socioeconômico; ▣ educação: grau de instrução, área de formação, cursos realizados, alfabetismo. ▣ O quão bem o usuário lê? ▣ Prefere aprender com outras pessoas? ▣ Prefere aprender fazendo? <p>▣ idiomas e jargões: Que idiomas o usuário conhece e utiliza fluentemente?</p> <p>Possui: ▣ jargão profissional particular</p> <ul style="list-style-type: none"> ▣ vocabulário próprio da empresa, da sua atividade ou de algum grupo social relevante para o seu projeto 	<p>Relação com a tecnologia</p> <ul style="list-style-type: none"> ▣ experiência com computadores: alfabetismo computacional, habilidade com computadores, anos de experiência. <ul style="list-style-type: none"> ▫ Que sistemas computacionais o usuário conhece? ▫ Quais deles costuma utilizar? ▫ Que hardware costuma utilizar? ▣ experiência com um produto específico ou ferramentas semelhantes: <ul style="list-style-type: none"> ▫ experiência com produtos concorrentes e outros produtos específicos do domínio, hábitos de uso, preferências e descontentamentos ▣ tecnologia disponível: <ul style="list-style-type: none"> ▫ hardware (tamanho e resolução do monitor, velocidade do processamento etc.), software e outras ferramentas aos quais tem acesso ▣ é importante investigar: <ul style="list-style-type: none"> ▫ Quem utilizará o sistema? ▫ Quem será afetado por ele? ▫ Quem é responsável por decidir quais objetivos o sistema deve apoiar e quais funcionalidades ele deve ter? ▫ Quem definiu os processos a serem apoiados pelo sistema?

Aba 13: Planejamento da Validação

Histórias confirmadas:

0%

Guia para planejamento da validação				
Pré-requisitos:	Quais são os pré-requisitos necessários para que a validação seja executada de forma correta, visando atingir seu objetivo?			
Ambiente:	Qual ambiente será necessário para realizar os testes? Uma sala, laboratório, computador?			
Ferramentas:	Qual sistema operacional? Quais ferramentas serão utilizadas para a validação: protótipo de papel, protótipo de baixa fidelidade, computador?			
Objetivo da Validação:	Qual o objetivo da sua validação? Qual o resultado esperado da sua validação? Defina em forma de números. Ex: monte um formulário para que o usuário diga o quão ele gostou da funcionalidade avaliada, ou peça que ele coloque em ordem decrescentes das funcionalidades que mais gostou para as que menos gostou.			
Salto de fé:				
<u>História</u>	Resultado Esperado	Resultado obtido	Comprovado (s/n)?	Comentários
Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]			--	
			--	
			--	
			--	

Aba 14: Criação de Histórias

Funcionalidade	Cenário	Ator	História
	Cenário 1	Ator 1	Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]
			Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]
			Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]
			Como um [ATOR (PERSONA)] eu quero/preciso/devo/gostaria de [AÇÃO] para [FUNCIONALIDADE]

APÊNDICE – D

Formulário de avaliação de perfil.

Formulário para Definição de Perfil

Este formulário foi criado para auxiliar na definição de perfil do aluno para medir o impacto nos resultados finais da experimentação da aplicação da metodologia Startup Kaizen.

*Obrigatório

Nome Completo *

Qual o seu grupo? *

Idade *

- Menos que 21 anos
- Entre 20 e 25 anos
- Entre 26 e 30 anos
- Acima de 30 anos

Você atua ou já atuou no mercado de TI? *

- Sim
- Não

Em qual das áreas abaixo você atua ou já atuou no mercado de TI? *

- Nenhum, nunca atuei no mercado de TI
- Desenvolvimento
- Negócios
- Qualidade
- Infraestrutura
- Processos
- Sistemas
- Outro:

Quanto tempo de atuação você possui? *

- Nunca atuei
- Menos do que 1 ano
- Entre 1 e 3 anos
- Entre 3 e 5 anos
- Mais do que 5 anos

Qual das opções abaixo melhor descreve a sua atuação no momento? *

- Estudante de pós-graduação/mestrado/doutorado apenas
- Estudante de pós-graduação/mestrado/doutorado e profissional de T.I.
- Profissional de T.I. apenas
- Outro:

Aponte o nível de conhecimento nos tópicos abaixo: *

Faça a avaliação com base no que você sabia antes de iniciar as disciplinas de Eng. de SW e IHC

	Não conhecia	Possuía conhecimento teórico apenas	Possuía conhecimento acadêmico prático apenas	Possuía conhecimento teórico e prático	Já havia aplicado o conhecimento em projetos no mercado de trabalho
Lean Startup	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Customer Development	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Business Model Canvas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Minimum Viable Product (MVP)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Design Thinking	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Personas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Histórias	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Salto de fé	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Behavior Driven Development (BDD)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Matriz de Relacionamento	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenários de utilização (8 Momentos Chave)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testes de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Qualidade de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rastreabilidade de Requisitos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Metodologias ágeis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

APENDICE – E

Formulário para avaliação dos usuários.

Pesquisa de Satisfação dos Usuários

Este formulário foi criado para medir o índice de satisfação dos usuários durante a utilização do software desenvolvido.

Esta é uma pesquisa acadêmica, os dados serão utilizados no estudo de caso para o mestrado em Ciência da Computação na UFSCar - Sorocaba.

O formulário é composto por 15 perguntas de multipla escolha, divididas em duas etapas:

- Identificação do cliente e software avaliado;
- Avaliação do software através de notas de 1 a 10 de acordo com sua experiência.

***Obrigatório**

Nome *

Sexo *

Idade *

- 15 - 20 anos
- 21 - 25 anos
- 25 - 30 anos
- Mais do que 30 anos

Software utilizado *

Qual funcionalidade essa ferramenta entrega que fez com que você a utilizasse? *

Continuar »

Pesquisa de Satisfação dos Usuários

*Obrigatório

Avaliação do Software

Nesta etapa, avalie o software selecionando na escala de 1 a 10 a nota que melhor reflete a sua satisfação com o software.

No seu dia-a-dia, o quão útil você considera a ferramenta? *

1 2 3 4 5 6 7 8 9 10

Nada
útil



Muito útil, não sei como fiquei sem essa solução até agora

Qual nota você daria em relação a dificuldade de utilizar a ferramenta? *

1 2 3 4 5 6 7 8 9 10

Muito Difícil



Muito Fácil

O quanto você acha que a ferramenta atende ao propósito pelo qual fez com que você a utilizasse? *

1 2 3 4 5 6 7 8 9 10

Nada



Atende 100%

Você encontrou algum problema ao utilizar a ferramenta? *

- Sim
 Não

Qual o impacto do problema encontrado na utilização da ferramenta? *

1 2 3 4 5 6 7 8 9 10

Nenhum Alto - eu não consegui completar a atividade que pretendia

Qual o grau de dificuldade para realização da funcionalidade principal que te atraiu para a utilização da ferramenta? *

1 2 3 4 5 6 7 8 9 10

Muito Difícil Muito Fácil

O quanto você recomendaria a utilização dessa ferramenta para outras pessoas? *

1 2 3 4 5 6 7 8 9 10

Não recomendaria Com certeza recomendaria

De forma geral, como você avaliaria a ferramenta? *

1 2 3 4 5 6 7 8 9 10

Ruim - não atende minha
necessidade e está cheia de
problemas

Excelente - pode até
apresentar alguns
problemas, mas atende
minha necessidade

Deixe seu comentário geral sobre a ferramenta

Esse item não é obrigatório. Deixe um comentário geral sobre a sua experiência na utilização da ferramenta.

Deixe seu email de contato

Esse item é opcional, porém pedimos por favor que deixe seu contato para um eventual contato para esclarecimento de dúvidas em relação as respostas dadas.

« Voltar

Enviar

ANEXOS

ANEXO - A

Aba Principal gerada pela equipe *Dover*

ID	HISTÓRIA	DESCRIÇÃO	IMPACTO/DIFICULDADE	PRIORIDADE	MVP	VALIDADO (S/N)	MÓDULO	SPRINT	TESTADO (S/N)	PASSOU (S/N)	BUGS
1	Eu (Munlo, estudante universitário) gostaria de [ACESSAR] o cadastro 1 (de rotas para que eu possa [REGISTRAR ROTAS E HORÁRIOS] para transportar encomendas.	Cadastro de endereços de origem e destino, meio de transporte e horários, aos quais o trajeto será percorrido pelo transportador.	[+]	1 sim	sim	1,0,0	1 sim	1 sim	sim	- problema ao deletar rota com horário agendado - não foi possível deletar rota de um usuário - não foi possível associar uma rota a um usuário - problema para retornar entidades relacionadas a uma rota - erro ao gravar entidades relacionadas à entidade rota	
2	Eu (Lobana, vendedora) quero ter a opção de [ESCOLHER] qual é o melhor [MODELO DE ENTREGA] de acordo com a minha necessidade.	Quem transporta pode cadastrar as formas de entrega que aceita, elas são entrega a local definido, postagem em correio local, agendamento de local/horário com quem vai receber.	[+]	9 sim	sim	1,1,0	2 sim	2 sim	sim	- problema em recuperar dados por causa do novo relacionamento - lista de produto vazia ao enviar ao sender	
3	Eu (Lobana, vendedora) preciso [PROCURAR] pessoas que tenham disponibilidade para [TRANSPORTAR ENCOMENDAS] toda semana.	Formulário que lista as possíveis rotas de envio e quem faz essa rota com em diferentes horários. Permitindo a quem deseja enviar realizar um agendamento. As rotas tem origem e destino definidos.	[+]	2 sim	sim	1,0,0	1 sim	1 sim	sim	- retornar os agendamentos com os devidos subcampos preenchidos retoma erro - problema com as latitudes e longitudes removidas de modo incorreto - problema na interface interfere ao selecionar os itens da lista - o nome do usuário que envia aparece incorretamente na aba acompanhar do transportador	
4	Eu (Raíael, estudante) gostaria de [REALIZAR] o check-in de minha localização de maneira que o [RASTREAMENTO] da encomenda esteja no curso da entrega.	Registro de posição momentânea do transportador. Posição que só disponível aos envolvidos em algum transporte de entrega.	[+]	não	não						
5	Eu (Carlos, contador) quero [TRAR] uma foto do produto que desejo enviar para que conste a [FOTO NO STATUS] da minha solicitação de entrega.	Foto de marcação tirada no momento em que o produto passa de mãos. Primeiro de quem deseja enviar para o transportador e depois do transportador para quem recebe. As fotos servem de validação para o início e finalização da entrega.	[+]	8 sim	sim	1,1,0	2 sim	2 sim	sim	- atualizar componentes gráficos na thread da camera - tamanho da foto - string de conexão com o serviço de blob - problema como os caminhos das fotos no blob - cancelamento da foto trava aplicação - email com a foto chega vazio	
6	Eu (Carlos, contador) quero ter a opção de [REGISTRAR] o meu grau de satisfação com a entrega contribuindo para o [RANKING DE USUÁRIOS] da aplicação.	Define notas e comentários sobre os envolvidos, como transportes bem sucedidos, pontualidade, velocidade, contato. Ou do lado de quem envia como definir corretamente a encomenda, agenda corretamente com quem recebe. Permite que os participantes de um transporte definam pontos ao serviço em etapas da execução.	[+]	3 SIM	sim	1,0,0	1 sim	1 sim	sim	- problemas com os componentes de interface, os botões para atualizar status estão confirmando próximo status - os dados do usuário não estão sendo gravados corretamente no banco de dados	
7	Eu (Mara, auxiliar de escritório) preciso [DISPONIBILIZAR] rotas para o [ENVIO DE MERCADORIAS LIVRES] quando surgir alguém interessado nesta rota.	Cadastrar por encomendas livres que podem ser entregues dentro de um período e ficam disponíveis para quem se referenciar para transportar na rota desejada.	[+]	não	não						
8	Eu (Munlo, estudante universitário) quero [REALIZAR] uma busca por encomendas livres para a rota registrada para envio. O transportador pode se cadastrar para levar.	Buscar por encomendas livres para a rota registrada. Encomendas que foram registradas para envio. O transportador pode se cadastrar para levar.	[+]	não	não						
9	Eu (Lobana, vendedora) quero ter a opção de [PAGAR] as encomendas que envio através de uma [CORTA PAGAMENTO] centralizada.	Permitir realizar depósitos em forma de crédito. Que é repassado a quem transporta quando um envio é aceitado.	[+]	não	não						
10	Eu (Frandea, estudante universitário) gostaria de [REALIZAR] o [REGISTRO DE ENTREGA] de uma encomenda que eu transporto.	Transportador realiza o registro da entrega, data e hora e para quem entregou.	[+]	4 sim	sim	1,0,0	1 sim	1 sim	sim	- problemas ao gravar entidades relacionadas - data e hora apresentam informações globais como fuso horário - o controle de fluxo do estado atual da aplicação inesperadamente	
11	Eu (Mara, auxiliar de escritório) gostaria de [ENVIAR] convites com a intenção de [CONVIDAR PESSOAS] para aumentar o nível de confiança da aplicação.	Envio de convites para pessoas terem acesso à aplicação e que podem realizar o envio ou transporte de encomendas.	[+]	5 sim	sim	1,0,0	1 sim	1 sim	sim	- geração de código de convite com caracteres inválidos - interface não valida dados enviados ao servidor e o retorno quebrando a aplicação	
12	Eu (Raíael, estudante) quero [EFETUAR] o [REGISTRO NA APLICAÇÃO] após receber o convite.	Usuário da aplicação, tanto quem deseja enviar quanto quem deseja transportar, utiliza o convite enviado para se registrar na aplicação.	[+]	6 sim	sim	1,0,0	1 sim	1 sim	sim	- senha do usuário encriptada acrescenta espaços em branco desnecessariamente - após digitar o código e email do usuário a interface permite continuar sem validar nome e senha	
13	Eu (Lobana, vendedora) gostaria de ter como opção [ACESSAR] a aplicação através da minha [CONTA NO FACEBOOK] para que eu possa ter uma identificação mais precisa dos usuários com os quais eu posso falar.	O usuário tanto que quer transportar como aquele que deseja enviar encomendas através da aplicação podem se conectar através de suas respectivas contas no facebook para aumentar o nível de confiança em seus usuários e seus respectivos perfis.	[+]	7 sim	sim	1,1,0	2 sim	2 sim	sim	- logout event do facebook - tela de progresso no convite trava	
14	Eu (Raíael, estudante) gostaria de além de realizar o login usando minha conta do facebook, a possibilidade de [CONVIDAR PELO FACEBOOK] para usar o aplicativo.	O usuário que transporta e que envia deseja convidar outras pessoas para utilizar o aplicativo através do facebook.	[+]	8 sim	sim	1,2,0	3 sim	3 sim	sim	- Instabilidade no servidor	
15	Eu (Mara, auxiliar de escritório) quero [RECEBER NOTIFICAÇÃO] sobre o [STATUS DA ENCOMENDA] através de email.	O usuário que envia o produto deseja receber notificação sobre o andamento da entrega de sua encomenda e a foto do produto retirado e a foto do produto na entrega.	[+]	9 sim	sim	1,2,0	3 sim	3 sim	sim	- O usuário não está sendo notificado da mudança de status - O email não está sendo enviado com a foto anexada	
16	Eu (Carlos, contador) quero ter a opção de [ESPECIFICAR] um raio em km para aumentar ou diminuir a [ABRANGÊNCIA DA BUSCA] por rotas.	O usuário deseja ter um campo adicional na tela de busca para que possa delimitar o raio de busca por um endereço de destino, para que possa verificar a disponibilidade de quem irá para a região buscada.	[+]	10 sim	sim	1,2,0	3 sim	3 sim	sim	- a busca por pontos próximos não está retornando dados válidos	
17	Eu (Munlo, estudante universitário) gostaria que o usuário tivesse a opção de [CADASTRAR] as [CARACTERÍSTICAS DO PRODUTO] para posterior consulta.	O transportador deseja que o usuário que envia um produto cadastre as características como peso, altura, largura e comprimento para posterior consulta de disponibilidade de espaço para envio do mesmo.	[+]	11 sim	sim	1,2,0	3 sim	3 sim	sim	- o produto cadastrado apresenta inconsistências dos dados no banco de dados	

ANEXO - B

Aba Principal gerada pela equipe Caregiver

#	História	Descrição	Classificação	Prioridade	MVP	Validado (s/n)?	Sprint	Módulo	Testado (s/n)?	Passou (s/n)?	Bugs
1	Como Blanca, técnica de administração, preciso estar de olho no estoque de medicamentos para poder fazer novos pedidos. Como Andreia, auxiliar de enfermagem, preciso estar de olho no estoque de medicamentos para garantir que sempre que for ministrado um medicamento em um determinado idoso este medicamento estará disponível.	Compra de estoque mínimo e possibilidade de compra		Alta	2	Sim	2	2	2	Não	-Avisa o estoque mínimo e possível comprar mas não tem um link direto do aviso para tela de compra
2	Como Blanca, técnica administrativa, preciso fazer novos pedidos de medicamentos conforme o estoque for acabando.	Aviso de estoque mínimo e aviso a adm		Alta	2	Sim	2	2	2	Não	-Avisa o estoque mínimo e possível comprar mas não tem um link direto do aviso para reforçar informação ao administrativo
3	Como Andréia auxiliar de enfermagem preciso ter fácil acesso as informações dos idosos pelos quais eu sou responsável, pois sou responsável por muitos idosos, e eventualmente posso esquecer informações importantes	Compra de medicamento		Alta	2	Sim	2	2	2	Sim	Não há BUGS
4	Como Andréia auxiliar de enfermagem preciso garantir que ministrei todos os medicamentos dentro e fora do padrão nos horários corretos.	Visualizar informação dos idosos		Média	1	Sim	1	1	1	Sim	Não há BUGS
5	Como Andréia auxiliar de enfermagem preciso atribuir a medicação aos idosos	Aviso de horário de medicamento		Alta	2	Sim	2	2	2	Sim	Não há BUGS
6	Como Blanca, técnica administrativa, preciso cadastrar e atualizar informações de novos idosos e medicamentos.	Atribuição de medicação		Alta	1	Sim	1	1	1	Não	A atribuição é feita a um idosos por vez e não a vários idosos ao mesmo tempo
7		CRUD		Baixa	1	Não	1	1	1	--	Não foi implementado a pedido do professor

ANEXO – C

Guia de Validação gerada pela equipe *Caregiver*

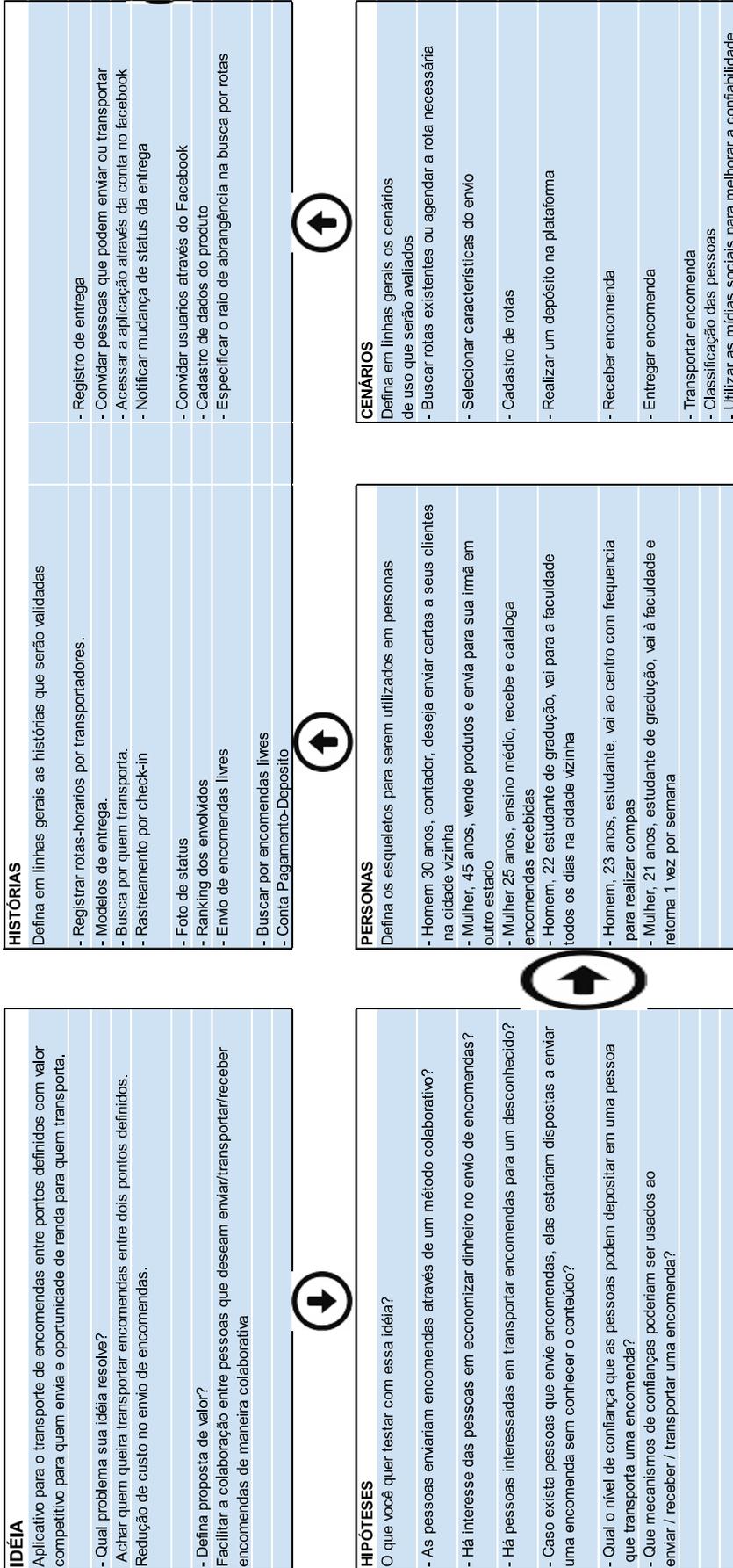
Guia para planejamento da validação

Pré-requisitos:	Será necessário a disponibilidade dos enfermeiros e administradores, permissão da instituição e os protótipos prontos para serem validados.
Ambiente:	Os testes serão realizados dentro da ILPI no ambiente de trabalho dos enfermeiros de administradores da instituição. Para fazer os teste será utilizado o notebook do avaliador e o mesmo levará um celular android para testar a aplicação mobile.
Ferramentas:	Foi feita uma validação suja utilizando um protótipo de baixa fidelidade (com mockups e POP) e uma segunda validação para entrega 2 utilizando um protótipo de alta fidelidade que englobou 40% do sistemas, apresentando as funcionalidades principais.
Objetivo da Validação:	Verificar se o que esta sendo implementado esta atendendo a expectativa do usuário e se será necessário alterar alguma coisa nas funcionalidades principais para entrega final da sprint 2.
Salto de fé:	https://drive.google.com/drive/folders/0B1Ym_QFjotrW1zduXQWGZlUTA

<u>História</u>	Resultado Esperado	Resultado obtido	Comprovado (s/n)?	Comentários
Como Bianca, técnica de administração, preciso estar de olho no estoque de medicamentos para poder fazer novos pedidos.	Conforme os enfermeiros ministrarem a medicação e apontarem isso no mobile o remédio é decrementado do banco de dados e quando o estoque chegar em determinados níveis alguns avisos são disparados para que Bianca possa fazer novos pedidos.	Apesar de Bianca poder receber os avisos de estoque baixo sobre um determinado medicamento, a auxiliar de enfermagem Andreia consegue somente visualizar na aplicação mobile uma tela com os próximos avisos de medicações a serem ministradas. Consequentemente, Andreia ainda não recebe o aviso efetivo de quando deve medicar um idoso, e com isso, o remédio ainda não é decrementado do banco de dados.	Sim	Sem comentários.
Como Andreia, auxiliar de enfermagem, preciso estar de olho no estoque de medicamentos para garantir que sempre que for ministrar um medicamento em um determinado idoso este medicamento estará disponível.	Conforme os enfermeiros ministrarem a medicação e apontarem isso no mobile o remédio é decrementado do banco de dados e quando o estoque chegar em determinados níveis alguns avisos são disparados. Dessa maneira o administrativo pode fazer novos pedidos e Andreia não vai se preocupar com falta de medicamentos em estoque	Bianca já recebe os avisos no sistema web sobre os medicamentos que estão em baixa no estoque, e pode visualizar a quantidade atual e quantidade mínima que deve ter em estoque, de cada medicamento. Todavia, Andreia ainda não consegue indicar que uma medicação foi aplicada a um idoso, pois ainda não esta recebendo os avisos de aplicação de medicação, e consequentemente, não há decréscimo da quantidade de um medicamento no banco de dados.	Sim	Sem comentários.
Como Bianca, técnica administrativa, preciso fazer novos pedidos de medicamentos conforme o estoque for acabando.	Bianca pode fazer os pedidos de forma rápida e prática pelo sistema sem precisar ir comprar os medicamentos pessoalmente.	Bianca ainda não tem acesso a pesquisa de preços de medicamento, assim como a compra do mesmo. Ela só pode, por enquanto, atualizar no sistema uma nova quantidade comprada de um medicamento, a partir da funcionalidade de "Medicação em Estoque".	Sim	O botão "Comprar" presente nessa tela não condiz com a operação real, que é a de atualização de um medicamento. Como sugestão, a usuária indicou a mudança do nome do botão para "Atualizar", ou algo semelhante.
Como Andréia auxiliar de enfermagem preciso ter fácil acesso as informações dos idosos pelos quais eu sou responsável, pois sou responsável por muitos idosos, e eventualmente posso esquecer informações importantes	Disponibilizar para as enfermeiras as informações de todos os idosos de maneira rápida e fácil pelo aplicativo e pelo sistema web.	São exibidas para Andreia, na aplicação web e mobile, por enquanto, apenas as informações mais essenciais sobre o idoso, como próximos medicamentos a serem tomados, nome, idade e complexidade.	Sim	Na tela de "Atribuição de Medicamento ao Idoso", após a enfermeira selecionar uma data para início da aplicação da medicação e o período de dias em que a mesma será ministrada (por exemplo, 10 dias), seria interessante o sistema informar a data final, ou seja, até quando o remédio deve ser aplicado. Houve também a não compreensão do campo com a informação "complexidade", exibido sobre cada idoso, na tabela informativa para atribuição de medicação para um idoso. Como sugestão, a classificação da complexidade do idoso deve ser alterada para "A", "B" ou "C", e não "Alta", "Média" ou "Baixa", como esta atualmente no sistema.
Como Andréia auxiliar de enfermagem preciso garantir que ministrei todos os medicamentos dentro e fora do padrão nos horários corretos.	Disparar alarmes programados para avisar do horário de aplicação de um determinado remédio em um determinado idoso para um determinado horário.	Atualmente, é exibido para Andreia na aplicação mobile apenas a consulta a tabela de "Próximos Avisos" para verificar as próximas medicações que devem ser aplicadas. Andreia ainda não recebe os devidos alarmes programados de acordo com a atribuição de um medicamento feita a um idoso.	Sim	Sem comentários.

ANEXO – D

SW Canvas gerado pela equipe *Dover*



ANEXO – E
 Planilha de testes – *Caregiver* testado
 Impressão Distribuída

por

#	Histórias ou Requisitos	Descrição	Testado (s/n)?	Passou (s/n)?	Bugs
1	Como Bianca, técnica de administração, preciso estar de olho no estoque de medicamentos para poder fazer novos pedidos.	Conforme os enfermeiros ministrarem a medicação e apontarem isso no mobile o remédio é decrementado do banco de dados e quando o estoque chegar em determinados níveis alguns avisos são disparados para que Bianca possa fazer novos pedidos.	Sim	Não	O botão de atribuir medicação não funcionou
2	Como Andreia, auxiliar de enfermagem, preciso estar de olho no estoque de medicamentos para garantir que sempre que for ministrar um medicamento em um determinado idoso este medicamento estará disponível.	Conforme os enfermeiros ministrarem a medicação e apontarem isso no mobile o remédio é decrementado do banco de dados e quando o estoque chegar em determinados níveis alguns avisos são disparados. Dessa maneira o administrativo pode fazer novos pedidos e Andreia não vai se preocupar com falta de medicamentos em estoque	Sim	Não	O botão de atribuir medicação não funcionou
3	Como Bianca, técnica administrativa, preciso fazer novos pedidos de medicamentos conforme o estoque for acabando.	Disponibilizar para as enfermeiras as informações de todos os idosos de maneira rápida e fácil pelo aplicativo e pelo sistema web.	Não	--	Durante o teste não foi encontrada essa função
4	muitos idosos, e eventualmente posso esquecer informações importantes	Existe a possibilidade de acompanhar o pedido, tanto pelo cliente assim como pelo parceiro	Sim	Sim	
5	Como Andréia auxiliar de enfermagem preciso garantir que ministrei todos os medicamentos dentro e fora do padrão nos horários corretos.	Disparar alarmes programados para avisar do horário de aplicação de um determinado remédio em um determinado idoso para um determinado horário.	Não	--	Durante o teste não foi encontrada essa função

ANEXO – F
Planilha de testes – *Dover* testado por *Caregiver*

ID	HISTORIA	DESCRICAÇÃO	TESTADO(S/N)	PASSOU(S/N)	Grupo CAREGIVER preenchendo
1	Eu [Murilo, estudante universitário] gostaria de [ACESSAR] o cadastro de rotas para que eu possa [REGISTRAR ROTAS E HORÁRIOS] para transportar encomendas	Cadastro de endereços de origem e destino, meio de transporte e horários aos quais o trajeto será percorrido pelo transportador, sera as informações de rotas e horários existentes.	sim	Não	Não possui botão de voltar, não é possível deletar rota com horário já agendado e não é possível editar uma rota de transporte
2	Eu [Joana, vendedora] quero ter a opção de [ESCOLHER] qual é o melhor [MODELOS DE ENTREGA] de acordo com a minha necessidade	Quem transporta pode cadastrar as formas de entrega que aceita, elas são entrega a local definido, postagem em correio local, agendamento de local-horário com quem vai receber.	sim	Não	Eu devo inserir a distância mas como eu devo saber isso?
3	Eu [Joana, vendedora] preciso [PROCURAR] pessoas que tenham disponibilidade para [TRANSPORTAR ENCOMENDAS] toda semana	Formulário que lista as possíveis rotas de envio e quem faz essa rota com em diferentes horários. Permitindo a quem deseja enviar realizar um agendamento. As rotas tem origem e destino definidos.	sim	Não	Não consegui encontrar na aplicação essa funcionalidade, seleionei quero enviar inseri origem destino e km e só apareceu a rota mas não a frequencia.
5	Eu [Carlos, contador] quero [TIRAR] uma foto do produto que desejo enviar para que conste a [FOTO NO STATUS] da minha solicitação de entrega	Foto de marcação tirada no momento em que o produto passa de mãos. Primeiro de quem deseja enviar para o transportador e depois do transportador para quem recebe. As fotos servem de validação para o inicio e finalização da entrega.	sim	sim	Para salvar a foto esta demorando bastante.
6	Eu [Carlos, contador] quero ter a opção de [REGISTRAR] o meu grau de satisfação com a entrega contribuindo para o [RANKING DE USUÁRIOS] da aplicação	Define notas e comentarios sobre os envolvidos, como transportes bem sucedidos, pontualidade, velocidade, contato. Ou do lado de quem envia como definir corretamente a encomenda, agenda corretamente com quem recebe. Permite que os participantes de um transporte definam pontos ao serviço	sim	sim	Sem bugs.
10	Eu [Femanda, estudante universitária] gostaria de [REALIZAR] o [REGISTRO DE ENTREGA] de uma encomenda que eu transporte	Transportador realiza o registro da entrega, data e hora e para quem entregou.	sim	sim	Sem bugs.
11	Eu [Mara, auxiliar de escritório] gostaria de [ENVIAR] convites com a intenção de [CONVIDAR PESSOAS] para aumentar o nível de confiança da aplicação	Envio de convites para pessoas terem acesso à aplicação e que podem realizar o envio ou transporte de encomendas.	sim	Não	Por e-mail o convite foi enviado, pelo facebook não funcionou. Clica-se em convidar pelo facebook e não abre a tela de permissão do facebook e sequer o convite é feito.
12	Eu [Rafael, estudante] quero [EFETUAR] o [REGISTRO NA APLICAÇÃO] após receber o convite	Usuário da aplicação, tanto quem deseja enviar quanto quem deseja transportar utiliza o convite enviado para se registrar na aplicação	sim	Não	Recebi o convite e o código com sucesso por email me registrei e funcionou. Não consegui registrar com facebook, uma tela de erro é retornada.
13	Eu [Joana, vendedora] gostaria de ter como opção [ACESSAR] a aplicação através da minha [CONTA NO FACEBOOK] para que eu possa ter uma identificação mais precisa dos usuários com os quais eu posso lidar	O usuário tanto que quer transportar como aquele que deseja enviar encomendas através da aplicação podem se conectar através de suas respectivas contas no facebook para aumentar o nível de confiança em seus usuários e seus respectivos perfis	sim	Não	Não foi possível acessar via facebook. Tela de erro aparece.
14	Eu [Rafael, estudante] gostaria de além de realizar o login usando minha conta do facebook, a possibilidade de [CONVIDAR PELO FACEBOOK] para usar o aplicativo	O usuário que transporta e que envia deseja convidar outras pessoas para utilizar o aplicativo através do facebook	sim	Não	Não foi possível, ameaça carregar mas nenhuma página abre e não consegui fazer o convite.
15	Eu [Mara, auxiliar de escritório] quero [RECEBER NOTIFICAÇÃO] sobre o [STATUS DA ENCOMENDA] através de email	o usuário que envia o produto deseja receber notificação sobre o andamento da entrega de sua encomenda e a foto do produto retirado e a foto do produto na entrega	sim	Sim	Sem bugs. Funcionando corretamente.
16	Eu [Carlos, contador] quer ter a opção de [ESPECIFICAR] um raio em km para aumentar ou diminuir a [ABRANGENCIA DA BUSCA] por rotas	O usuário deseja ter um campo adicional na tela de busca para que possa delimitar o raio de busca por um endereço de destino, para que possa verificar a disponibilidade de quem irá para a região buscada	sim	Não	Conseguo delimitar a distancia. Porém fiz uma pesquisa de rota que funcionaria pois já sabia que estava previamente cadastrada, e então mudei o destino (por um que eu sabia que não estava cadastrado) e a rota anterior não sumiu da tela dando a ideia de que argentina-sao paulo (distancia 1km) pudesse ser sorocaba- sao paulo
17	Eu [Murilo, estudante universitário] gostaria que o usuário tivesse a opção de [CADASTRAR] as [CARACTERÍSTICAS DO PRODUTO] para posterior consulta.	O transportador deseja que o usuário que envia um produto cadastre as características como peso, altura, largura e comprimento para posterior consulta da disponibilidade de espaço para envio do mesmo	sim	Não	Conseguo cadastrar as informações mas não consigo fazer a consulta delas posteriormente.

ANEXO – G
 Planilha de testes – *Factoring*
 Sys testado por *Dover*

#	Histórias ou Requisitos	Descrição	Testado (s/n)	Passou (s/n)	Bugs
1	Validar e verificar o site entregue	Site desenvolvido para divulgação do sistema como um todo http://ec2-54-69-185-3.us-west-2.compute.amazonaws.com:8080/factory/	Sim	Não	Ícones da página principal estão quebrados E-mail de contato se comporta de modo estranho
2	Site:				
3	Efetuar o Login através do site	No canto esquerdo no menu encontra-se o login	Sim	Sim	
4	Efetuar o login como uma empresa	A empresa poderá lançar documentos para solicitar créditos	Sim	Sim	
5	Usuário:	cliente@test.com	--	--	
6	Senha:	123456	--	--	
7	Cadastrar uma nova proposta	Cadastrar uma nova proposta, as factorings cadastradas tem um valor mínimo para receber essa proposta sendo apenas superior a 50000, assim sendo uma proposta menor não irá aparecer para a factoring. Lance quantas propostas forem necessárias para o teste. Ao final do cadastro a proposta pode ser editada.	Sim	Sim	Não encontre bugs relacionados. Sugestões: As telas podem ser tipo wizards para ficar mais fácil entender
8	Deslogar do sistema e retornar como uma factoring		Sim	Sim	
9	Usuário:	factory@test.com	--	--	
#	Senha:	123456	--	--	
#	Propostas	Caso tenha sido feito algum registro de proposta o mesmo deve aparecer na dashboard caso seja superior a 50 mil.	Sim	Sim	Não encontre bugs relacionados. Sugestões: As telas podem ser tipo wizards para ficar mais fácil entender
#	Visualizar e aprovar uma proposta	A proposta recebida pode ser aceita ou reprovada pela factoring, caso ela seja aceita a factoring recebe uma qualificação e a	Sim	Sim	Obs: Recebi o e-mail de validação

ANEXO – H

Planilha de testes – Impressão Distribuída
testado por *Factoring Sys*

#	Histórias ou Requisitos	Descrição	Testado (s/n)?	Passou (s/n)?	Bugs
1	Realizar cadastro de clientes e parceiros	Dentro da área específica, efetuar o cadastro	Sim	Sim	
2	No cadastro correspondente, realizar as alterações de dados	Depois de efetuado o login, é possível editar os seus dados	Sim	Sim	
3	Efetuar um pedido	Completar um ciclo completo de compra, desde a realização da compra até o pagamento	Sim	Sim	
4	Consultar o histórico do pedido	Existe a possibilidade de acompanhar o pedido, tanto pelo cliente assim como pelo parceiro	Sim	Sim	
5	Cadastrar serviços	Na área administrativa é possível cadastrar os serviços que ficarão disponíveis aos clientes	Sim	Sim	
6	Consultar clientes e parceiros	Também na área administrativa é possível consultar clientes e parceiros cadastrados no sistema	Sim	Sim	
7	Acompanhar o status e pontuar o parceiro	Após efetuar o pedido, o mesmo poderá ser acompanhado pelo cliente, após efetivação o parceiro poderá receber pontuação conforme satisfação do cliente	Sim	Sim	