

ESCOLA POLITÉCNICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

LEANDRO BENTO POMPERMAIER

**MODELO PRESS: EVOLUINDO A ADOÇÃO DE PRÁTICAS DE  
ENGENHARIA DE SOFTWARE EM STARTUPS DIGITAIS**

Porto Alegre  
2021

PÓS-GRADUAÇÃO - *STRICTO SENSU*



Pontifícia Universidade Católica  
do Rio Grande do Sul

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
ESCOLA POLITÉCNICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**MODELO PRESS: EVOLUINDO  
A ADOÇÃO DE PRÁTICAS DE  
ENGENHARIA DE SOFTWARE  
EM STARTUPS DIGITAIS**

**LEANDRO BENTO POMPERMAIER**

Tese apresentada como requisito parcial  
à obtenção do grau de Doutor em  
Ciência da Computação na Pontifícia  
Universidade Católica do Rio Grande do  
Sul.

Orientador: Prof. Dr. Rafael Prikladnicki

**Porto Alegre  
2021**



## Ficha Catalográfica

P788m Pompermaier, Leandro Bento

Modelo PRESS : evoluindo a adoção de práticas de Engenharia de Software em startups digitais / Leandro Bento Pompermaier. – 2021.

152 p.

Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Rafael Prikladnicki.

1. Startup de software. 2. Startup digital. 3. Processo de desenvolvimento. 4. Minimum Viable Product. 5. Engenharia de software. I. Prikladnicki, Rafael. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS  
com os dados fornecidos pelo(a) autor(a).

Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051



Leandro Bento Pompermaier

**Modelo PRESS: Evoluindo a adoção de práticas de Engenharia de Software em startups digitais**

Tese apresentada como requisito parcial para obtenção do grau de Doutor em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 25 de janeiro de 2021.

**BANCA EXAMINADORA:**

Prof. Dr. Alfredo Goldman vel Lejbman (IME/USP)

Prof. Dr. Ivan do Carmo Machado (IME/UFBA)

Profª. Dra. Sabrina dos Santos Marczak (PPGCC/PUCRS)

Prof. Dr. Rafael Prikładnicki (PPGCC/PUCRS - Orientador)



## **DEDICATÓRIA**

Dedico este trabalho à minha esposa Aline e aos meus filhos Arthur e Cecilia.





“One child, one teacher, one pen, and one book can change the world.”  
(Malala Yousafzai)



## AGRADECIMENTOS

Gostaria de expressar minha gratidão ao meu orientador, Prof. Dr. Rafael Prikladnicki, pelo apoio contínuo nesta minha jornada de pesquisa, por sua paciência, motivação, entusiasmo, imenso conhecimento e principalmente por me escutar em horários alternativos. Sua orientação me ajudou a entender o mundo da pesquisa e a quebrar alguns pré-conceitos.

Além do meu orientador, gostaria de agradecer a professora Dra. Sabrina Marczak e ao prof. Dr. Alfredo Goldman pelas excelentes colocações durante minha defesa de proposta de tese. Tenham certeza de que os comentários perspicazes e as perguntas difíceis ajudaram na construção do meu pensamento científico e contribuíram para a conclusão deste trabalho.

Agradeço aos colegas do Tecnopuc e da Escola Politécnica que sempre me apoiaram, em especial ao meu amigo, e agora doutor, Rafael Chanin. Certamente posso chamá-lo de guardião do doutorado. Muitas vezes mais preocupado do que eu com os prazos e com a estruturação do meu trabalho. Além de contribuir com ricas discussões para a realização dos trabalhos das disciplinas, estava sempre disposto a me escutar e me ajudar.

Por último, mas não menos importante, gostaria de agradecer à minha família: minha esposa Aline, que soube dos momentos em que eu precisava de apoio e motivação e aos meus filhos, Arthur e Cecilia, que me ensinam a cada dia a ser um ser humano melhor.



# MODELO PRESS: EVOLUINDO A ADOÇÃO DE PRÁTICAS DE ENGENHARIA DE SOFTWARE EM STARTUPS DIGITAIS

## RESUMO

Startups de software ou startups digitais vêm apresentando comportamentos que impactam o mercado com seus novos modelos de negócio e sua velocidade de crescimento. De acordo com a flexibilidade pedida pelo mercado e a velocidade que estes empreendimentos empregam no desenvolvimento dos seus produtos e serviços para uma rápida validação de mercado, há uma maior criticidade na escolha das práticas de Engenharia de Software a serem utilizadas. Existem diversos modelos de maturidade de negócio na literatura que são apresentados e discutidos entre empreendedores. Um destes modelos foi definido pelo Instituto Brasileiro de Governança Corporativa (IBGC). Estes modelos analisam o posicionamento das *startups* de software e seus empreendedores do ponto de vista de negócio e nada discutem ou apresentam sobre o ponto de vista da engenharia de software. Normalmente, nestas *startups* digitais, as práticas utilizadas para o desenvolvimento do software que é base para o produto ou serviço do negócio é negligenciado e adotado de forma informal ou até mesmo isolada por alguns profissionais das equipes de desenvolvimento de software. Esta informalidade aumenta os riscos de geração de dívida técnica nas diferentes etapas do desenvolvimento de software, seja no código criado, na arquitetura de software, nos testes e na documentação. Esta pesquisa tem como objetivo apresentar um modelo de práticas de engenharia de software adotadas por *startups* digitais em diferentes estágios de maturidade empresarial, denominado PRESS (Práticas de Engenharia de Software em *Startups*).

Uma pesquisa exploratória foi realizada por meio de métodos de pesquisa qualitativa e quantitativa, os quais forneceram dados para a definição do modelo proposto. Este modelo analisou as seguintes práticas de engenharia de software: equipe de desenvolvimento, requisitos de software, teste de software, arquitetura de software, gerenciamento de

projetos e processo de desenvolvimento de software. O PRESS foi desenvolvido com base nos resultados dos estudos realizados nesta pesquisa. Um estudo de grupo focal também analisou as práticas sugeridas. Os achados apresentam ou indicam evidências de que o modelo pode complementar os modelos de negócios, contribuindo para uma melhor reflexão ou adoção mais eficiente das práticas de Engenharia de Software em *startups*, evitando ou minimizando os riscos anteriormente mencionados.

**Palavras-Chave:** startup de software, startup digital, processo de desenvolvimento, Minimum Viable Product, engenharia de software, modelo de maturidade.

# **PRESS MODEL: EVOLVING THE ADOPTION OF SOFTWARE ENGINEERING PRACTICES IN DIGITAL STARTUPS**

## **ABSTRACT**

Software startups or digital startups have been showing behaviors that impact the market with their new business models and growth speed. According to the flexibility required by the market and the speed that these enterprises employ in developing their products and services for quick market validation, there is crucial importance in choosing the Software Engineering practices to be used. There are several models of business maturity in the literature that are presented and discussed among entrepreneurs. One of these models was defined by the Brazilian Institute of Corporate Governance (IBGC). These models analyze the positioning of software startups and their entrepreneurs from a business point of view. Still, they do not discuss or present anything from the software engineering perspective. Frequently, in these digital startups, the practices used to develop the software which serves as the basis for the business' product or service are neglected, adopted informally, or even isolated by some professionals in the software development teams. This informality increases the risks of generating technical debt in the different software development stages, such as the code created, the software architecture, the tests, or the documentation. This research aims to present a model of software engineering practices adopted by digital startups at different stages of business maturity, named PRESS (Portuguese acronym for Software Engineering Practices in Startups).

Exploratory research was conducted using qualitative and quantitative research methods, which provided data for the proposed model's definition. This model analyzed the following software engineering practices: development team, requirements, software testing, software architecture, project management, and development process. PRESS was developed based on the results of the studies carried out in this research. A focus group study also analyzed the suggested practices. The findings present or indicate evidence that



the model can complement the business models, contributing to a better reflection or more efficient adoption of Software Engineering practices in startups, avoiding or minimizing the risks previously mentioned.

**Keywords:** software startup, digital startup, development process, Minimum Viable Product, software engineering, maturity model.

## LISTA DE FIGURAS

Figura 2.1 – Processo de desenvolvimento do cliente [Bla12] .....	36
Figura 3.1 – Estágios do mapeamento sistemático [PFMM08] .....	54
Figura 3.2 – Desenho da metodologia de pesquisa .....	56
Figura 4.1 – Estágios do processo de seleção .....	63
Figura 4.2 – Criação do esquema de classificação .....	63
Figura 4.3 – Referências por ano de publicação .....	65
Figura 4.4 – Referências por ano .....	66
Figura 4.5 – Referências por ano .....	67
Figura 5.1 – Categorias resultantes das entrevistas (Fonte: Autor) .....	73
Figura 6.1 – Maturidade das Startups .....	79
Figura 6.2 – Engenharia de Software x Fases de Maturidade .....	80
Figura 6.3 – Processo de desenvolvimento das <i>startups</i> .....	81
Figura 6.4 – Papéis x Maturidade .....	82
Figura 6.5 – Elicitação de requisitos pelas <i>startups</i> .....	83
Figura 6.6 – Objetivos de qualidade da arquitetura .....	84
Figura 6.7 – Testes executados pelas startups .....	85
Figura 6.8 – Processo de desenvolvimento das <i>startups</i> .....	85
Figura 6.9 – Habilidades de Engenharia de Software da equipe .....	86
Figura 6.10 – Versão 1 dos níveis de maturidade de adoção de práticas de Engenharia de Software .....	87
Figura 6.11 – Versão 1 do modelo de adoção de práticas de Engenharia de Software por <i>startups</i> .....	87
Figura 7.1 – Categorias mais importantes - <i>Startup</i> K (Fonte: Autor) .....	93
Figura 7.2 – Categorias mais importantes - <i>Startup</i> M (Fonte: Autor) .....	94
Figura 7.3 – Categorias mais importantes - <i>Startup</i> L (Fonte: Autor) .....	94
Figura 7.4 – Categorias mais importantes - <i>Startup</i> N (Fonte: Autor) .....	95
Figura 7.5 – Categorias Citadas pelas <i>Startups</i> (Fonte: Autor) .....	97
Figura 7.6 – Categorias Citadas pelas <i>Startups</i> por Fase de Maturidade (Fonte: Autor) .....	98
Figura 7.7 – Definidores de requisitos - Fase Ideação (Fonte: Autor) .....	99
Figura 7.8 – Definidores de requisitos - Fase Validação (Fonte: Autor) .....	99
Figura 7.9 – Definidores de requisitos - Fase Tração (Fonte: Autor) .....	99

Figura 7.10 – Definidores de requisitos - Fase Escala (Fonte: Autor) . . . . .	100
Figura 7.11 – Equipe de TI - Startup K (Fonte: Autor) . . . . .	101
Figura 7.12 – Equipe de TI - <i>Startup</i> L (Fonte: Autor) . . . . .	102
Figura 7.13 – Equipe de TI - <i>Startup</i> M (Fonte: Autor) . . . . .	103
Figura 7.14 – Equipe de TI - <i>Startup</i> N (Fonte: Autor) . . . . .	104
Figura 7.15 – Equipe de TI - Visão Geral (Fonte: Autor) . . . . .	105
Figura 8.1 – Níveis de maturidade das práticas . . . . .	115
Figura 8.2 – Níveis representados no modelo proposto . . . . .	116
Figura 8.3 – Ideação . . . . .	117
Figura 8.4 – Validação . . . . .	118
Figura 8.5 – Tração . . . . .	120
Figura 8.6 – Escala . . . . .	122
Figura 8.7 – Modelo consolidado . . . . .	122

## LISTA DE TABELAS

Tabela 4.1 – Resultados das pesquisas nas bases de dados .....	61
Tabela 4.2 – Lista dos artigos primários .....	66
Tabela 5.1 – Startups x Critérios .....	69
Tabela 5.2 – Problemas técnicos no desenvolvimento das <i>startups</i> entrevistadas .	75
Tabela 7.1 – Categorias importantes na fase de ideação .....	95
Tabela 7.2 – Categorias importantes na fase de validação .....	96
Tabela 7.3 – Categorias importantes na fase de tração .....	96
Tabela 7.4 – Categorias importantes na fase de escala .....	96
Tabela 7.5 – Relação entre metodologia e software utilizados pelas <i>startups</i> ....	110
Tabela 8.1 – Definição dos níveis de maturidade das práticas .....	115

## LISTA DE SIGLAS

API – Application Programming Interface  
B2B – Business to Business  
B2B2C – Business to Business do Consumer  
B2C – Business to Consumer  
CEO – Chief Executive Office  
CTO – Chief Technology Office  
EP – Experience Paper  
ER – Evaluation Research  
ESSSDM – Early Stage Software Startup Development Model  
IBGC – Instituto Brasileiro de Governança Corporativa  
MVP – Minimum Viable Product  
OP – Opinion Paper  
PO – Product Owner  
PP – Philosophical Paper  
PRESS – Práticas de Engenharia de Software em *Startups*  
PMF – Product Market Fit  
PS – Proposal of Solution  
PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul  
RV – Validation Research  
SA – Sociedade Anônima  
SWEBOK – Software Engineering Body of Knowledge  
TECNO PUC – Parque Científico e Tecnológico da PUCRS  
UX – User eXperience

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>25</b>
1.1	OBJETIVO GERAL	27
1.2	OBJETIVOS ESPECÍFICOS	27
1.3	CONTEXTO DA PESQUISA	28
1.4	ESTRUTURA DA TESE	28
1.5	RESULTADOS ACADÊMICOS E CIENTÍFICOS	28
1.5.1	PUBLICAÇÕES CIENTÍFICAS	29
1.5.2	PROJETOS	31
1.5.3	PARTICIPAÇÃO EM EVENTOS	32
<b>2</b>	<b>BASE TEÓRICA</b>	<b>33</b>
2.1	EMPREENDEDORISMO E INOVAÇÃO	33
2.1.1	<i>STARTUP</i>	33
2.1.2	MODELO DE MATURIDADE DE <i>STARTUPS</i> PROPOSTO PELO IBGC	39
2.1.3	<i>STARTUPS</i> E ENGENHARIA DE SOFTWARE	47
2.2	PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	49
2.2.1	ABORDAGEM PRESCRITIVA	50
2.2.2	ABORDAGEM ADAPTATIVA	51
<b>3</b>	<b>METODOLOGIA DE PESQUISA</b>	<b>53</b>
3.1	MÉTODOS UTILIZADOS	53
3.1.1	REVISÃO SISTEMÁTICA DA LITERATURA	53
3.1.2	ESTUDO DE CAMPO	54
3.1.3	<i>SURVEY</i>	55
3.1.4	GRUPO FOCAL	55
3.2	METODOLOGIA	56
3.3	FASE 0 - EXPLORATÓRIA	56
3.4	FASE I - ESTUDO DE VIABILIDADE	57
3.5	FASE II - ESTUDO DE OBSERVAÇÃO	57
3.6	FASE III - ESTUDO DE AVALIAÇÃO	57
<b>4</b>	<b>FASE 0 - EXPLORATÓRIA - REVISÃO SISTEMÁTICA DA LITERATURA</b>	<b>59</b>

4.1	JUSTIFICATIVA E TRABALHOS RELACIONADOS .....	59
4.2	PROTOCOLO DA REVISÃO SISTEMÁTICA DA LITERATURA .....	60
4.3	ANÁLISE DOS RESULTADOS .....	65
4.3.1	SÍNTESE DO CAPÍTULO .....	67
<b>5</b>	<b>FASE I - ESTUDO DE VIABILIDADE - ESTUDO DE CAMPO COM <i>STARTUPS</i></b>	<b>69</b>
5.1	PROTOCOLO DO ESTUDO .....	69
5.1.1	CARACTERIZAÇÃO DAS <i>STARTUPS</i> .....	70
5.2	RESULTADOS .....	72
5.3	SÍNTESE DO CAPÍTULO .....	76
<b>6</b>	<b>FASE II - ESTUDO DE OBSERVAÇÃO - <i>SURVEY</i></b>	<b>78</b>
6.1	PROTOCOLO DO ESTUDO .....	78
6.2	ANÁLISE DAS RESPOSTAS .....	79
6.3	SÍNTESE DO CAPÍTULO .....	86
<b>7</b>	<b>FASE III - ESTUDO DE AVALIAÇÃO - GRUPO FOCAL</b>	<b>90</b>
7.1	PROTOCOLO DO ESTUDO .....	90
7.2	CARACTERIZAÇÃO DAS <i>STARTUPS</i> .....	91
7.3	CLASSIFICAÇÃO DE CARTÕES .....	92
7.3.1	<b>CARTÃO #1: MOMENTO DA <i>STARTUP</i></b> .....	93
7.3.2	<b>CARTÃO #2: CATEGORIAS MAIS IMPORTANTES POR ETAPA</b> .....	93
7.3.3	<b>CARTÃO #3: DEFINIÇÃO DOS REQUISITOS</b> .....	97
7.3.4	<b>CARTÃO #4: ALTERAÇÕES DOS REQUISITOS POR ETAPA</b> .....	100
7.3.5	<b>CARTÃO #5: PROPORÇÃO DA EQUIPE DE TI POR ETAPA</b> .....	100
7.4	GRUPO FOCAL .....	102
7.5	SÍNTESE DO CAPÍTULO .....	112
<b>8</b>	<b>MODELO PRESS - PRÁTICAS DE ENGENHARIA DE SOFTWARE PARA <i>STARTUP</i></b>	<b>114</b>
8.1	IDEAÇÃO .....	116
8.2	VALIDAÇÃO .....	117
8.3	TRAÇÃO .....	119
8.4	ESCALA .....	121
<b>9</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> .....	<b>124</b>

9.1	DISCUSSÕES .....	124
9.1.1	LIMITAÇÕES DESTA PESQUISA .....	127
9.2	TRABALHOS FUTUROS .....	127
9.3	CONSIDERAÇÕES FINAIS .....	128
	<b>REFERÊNCIAS .....</b>	<b>129</b>
	<b>APÊNDICE A – Roteiro das Entrevistas .....</b>	<b>143</b>
	<b>APÊNDICE B – Roteiro do Survey .....</b>	<b>144</b>
	<b>APÊNDICE C – Cartões - Card Sorting .....</b>	<b>151</b>
	<b>APÊNDICE D – Roteiro para o Grupo Focal .....</b>	<b>154</b>





## 1. INTRODUÇÃO

A evolução tecnológica apresentada nos últimos anos mostra-se um grande desafio para as empresas em geral, pois elas concentram esforços para encontrar novas e mais eficientes formas de conduzir seus negócios [LAFT16]. Há no mercado uma crescente busca por sistemas que operem com baixos custos, tenham alta qualidade bem como um curto ciclo de desenvolvimento. Isto indica que as práticas de Engenharia de Software tem como premissa a necessidade de atingir alta qualidade e produtividade. No desenvolvimento de software, alguns dos principais fatores que afetam a qualidade e a produtividade incluem pessoas, processos técnicos, bem como as tecnologias que estão sendo empregadas para realizar a atividade pretendida [Bos16].

Estatísticas mostram que, nos Estados Unidos, mais de 450 mil *startups* de software começam a operar todo mês [Fai14]. Estas são essenciais para o crescimento da economia, pois são responsáveis por mais de 20% dos empregos criados [Fai14]. Porém, a maioria delas não atinge a “maioridade”, pois a taxa de fracasso entre as *startups* de software é muito alta — mais de 70% das *startups* de software não sobrevivem [GWA14]. Uma das razões desse problema é a questão delas trabalharem em ambientes imprevisíveis e algumas vezes caóticos e, com isso, tenderem a se mover de acordo com as mudanças do seu entorno [GWA14]. Esta flexibilidade e constantes mudanças nos produtos ou serviços oferecidos tendem a aumentar os problemas de dívida técnica entre as *startups*, principalmente nas áreas de codificação, testes, arquitetura e documentação [KUC<sup>+</sup>18].

Atualmente, há um crescente número de empresas nascentes, chamadas de *startups*, que desenvolvem soluções inovadoras. Apesar do atual momento das *startups* e da disseminação do seu conceito, este foi apresentado à comunidade científica na década de 70. Cooper [CB77] definiu as *startups* como novas empresas de alta tecnologia. No seu estudo, analisou *startups* de São Francisco para entender os fatores de sucesso destas. Mais tarde, definiu-se startup como uma instituição humana desenhada para entregar um novo produto ou serviço sobre condições de extrema incerteza [Rie11]. Nesse sentido, Eric Ries [Rie11] define a metodologia Lean Startup como sendo uma metodologia para gestão de empresas em ambientes de grande incerteza. Esta é uma proposta para casos de desenvolvimentos pequenos e/ou empresas nascentes em que os recursos econômicos são limitados [Moo12, Rie11]. Pode-se definir um subconjunto das *startups* que possuem suas soluções baseadas em software como *startups* de software ou *startups* digitais. Estas são consideradas empresas recém-criadas sem histórico operacional e são extremamente rápidas na produção de tecnologias de ponta [PGU<sup>+</sup>14].

Carmel [Car94] em 1994, introduziu o termo startup na Engenharia de Software em seu artigo intitulado "Time-to-completion in *software package startups*". Nesse artigo,

ele analisou as práticas de Engenharia de Software nesse tipo de empresa inovadora e bem-sucedida.

Essas *startups* de software estão cada vez mais obcecadas por entregar um produto de software em um tempo extremamente curto para que este seja validado diretamente pelo usuário final. O uso de metodologia de desenvolvimento de software enxuto e a experimentação de modelos de negócio tornaram-se populares em *startups* de software, principalmente na concepção do produto mínimo viável (em inglês Minimum Viable Product MVP) [NP13]. Muitas *startups* utilizam o termo "protótipo" para definir seus MVPs [DA16], e Ries [Rie11] apresentou um conjunto de tipos de MVPs que são mostrados no Capítulo 2.

O ciclo de vida de uma startup é apresentado pelo Instituto Brasileiro de Governança Corporativa (IBGC) [LAM19]<sup>1</sup> com 4 (quatro) fases bem distintas: ideação, validação, tração e escala. Apesar de simples, este modelo consegue ter uma abrangência nas diferentes etapas da *startups* e discute as principais preocupações que os empreendedores devem ter. Apesar deste modelo apresentar a evolução da startup, o foco está na governança e negócio, o que deixa uma lacuna para o movimento de maturidade em relação às práticas de Engenharia de Software aplicadas pelos empreendedores na construção dos MVPs.

Durante o desenvolvimento dos MVPs pelas *startups* há uma considerável velocidade na construção e na entrega dos produtos de software, porém isto ao custo de abdicar do uso de práticas relacionadas à documentação, estruturas e processos [GPU<sup>+</sup>15]. O gerenciamento do dívida técnica adquirido durante as fases de desenvolvimento de uma startup aumenta a probabilidade de sucesso [KUC<sup>+</sup>19]. Segundo Klotins [KUC<sup>+</sup>19] todas as áreas do conhecimento referentes aos processo de desenvolvimento de software são importantes, porém deve-se averiguar em que momento estas práticas do processo devem ser aplicadas de acordo com o ciclo de vida da startup.

No contexto do desenvolvimento do MVP pelas *startups*, algumas práticas de Engenharia de Software são utilizadas, e este projeto de doutorado buscou identificá-las através de uma metodologia que envolveu uma fase exploratória, através de uma revisão sistemática da literatura. Esta revisão, apresentada no Capítulo 4 levantou os principais resultados de pesquisa da área de *startups* de software e contribuiu para limitar esta pesquisa. Após esta fase, um estudo de campo com startup foi realizado e com isso as principais práticas de Engenharia de Software aplicadas pelas *startups* foram levantados e serviram de base para a elaboração de um *survey* que aplicado junto à comunidade de *startups* trouxe *insights* para a elaboração da primeira versão do modelo de Práticas de Engenharia de Software em *Startups* (PRESS). Este modelo foi construído usando como base o modelo de maturidade apresentado pelo IBGC [LAM19], porém com a visão da adoção das práticas

---

<sup>1</sup>Instituto Brasileiro de Governança Corporativa (IBGC) é uma organização da sociedade civil que tem por objetivo gerar e disseminar conhecimento relacionado a governança corporativa e influenciar as empresas na adoção de melhores práticas

de Engenharia de Software pelas *startups* nos diferentes níveis de maturidade do negócio. Ao final, um grupo focal foi realizado com *startups*. Este grupo focal serviu para avaliação do modelo PRESS. Estes estudos foram realizados focando responder a seguinte questão de pesquisa:

### **Como adotar práticas de Engenharia de Software no desenvolvimento dos MVPs das *startups* digitais durante as diferentes fases de maturidade do negócio?**

Para responder a essa questão de pesquisa, as seguintes questões secundárias foram definidas:

- RQ #1: Como as práticas de Engenharia de Software são utilizadas por uma startup de software durante o processo de desenvolvimento do MVP?
- RQ #2: Quais ferramentas são usadas pelas *startups* de software durante o processo de desenvolvimento do MVP?
- RQ #3: Como as práticas de Engenharia de Software podem ser adotadas pelas *startups* de software nos seus diferentes momentos de maturidade de negócio?

#### **1.1 Objetivo Geral**

O objetivo geral desta pesquisa é propor um modelo de práticas de Engenharia de Software a serem adotadas pelas *startups* digitais nas diferentes fases de maturidade do negócio.

#### **1.2 Objetivos Específicos**

Para alcançar o objetivo geral proposto neste trabalho, identificam-se os seguintes objetivos específicos:

- Aprofundar estudos relacionados ao contexto de Engenharia de Software e *startups* de software;
- Identificar as principais práticas adotadas pelas *startups* de software no desenvolvimento dos MVPs;
- Avaliar o processo de desenvolvimento de software utilizado pelas *startups* de software nas diferentes etapas de desenvolvimento do negócio;
- Documentar e reportar os resultados do estudo, apresentando-os na comunidade científica.

### 1.3 Contexto da Pesquisa

A Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) possui um Parque Científico e Tecnológico (TECNO PUC) com mais de 170 organizações e 7000 pessoas, trabalhando em projetos de criatividade e inovação, em forte colaboração com a PUCRS e a comunidade. Desde a sua fundação (2003), o TECNO PUC contribui para o desenvolvimento empreendedor da sua comunidade acadêmica. Inicialmente através da incubadora RAIAR, e atualmente, através do Tecnopuc *Startups*. No último planejamento estratégico do parque, realizado em 2017, foi definida a intenção estratégica de criação de 1000 negócios inovadores em 10 anos. Dentro deste contexto, o ambiente do Tecnopuc, a conexão com a academia, os projetos de pesquisas envolvendo a área de *startups* de software e Engenharia de Software fazem com que o desenvolvimento desta pesquisa se torne relevante na conjuntura estratégica da universidade.

### 1.4 Estrutura da Tese

O documento está organizado nos seguintes capítulos: o Capítulo 2 expõe toda a fundamentação teórica da pesquisa que está sendo apresentada. Neste capítulo, conceitos de *Startups* de Software, MVP, processo de desenvolvimento de software são apresentados.

No Capítulo 3, a metodologia de pesquisa é apresentada, descrevendo-se as etapas e os métodos utilizados. No Capítulo 4 é apresentado uma revisão da literatura do tema deste trabalho. Os Capítulos 5, 6 e 7 apresentam os resultados do estudo de campo, survey e grupo focal realizados nesta pesquisa. O capítulo 8 apresenta a proposta do modelo de adoção de práticas de Engenharia de Software pelas *startups* e o Capítulo 9 apresenta as considerações finais.

### 1.5 Resultados Acadêmicos e Científicos

Durante o período do doutorado, algumas pesquisas empíricas foram conduzidas e alguns artigos foram trabalhados e publicados com a estratégia de validação de conceitos, para posterior definição do objeto deste trabalho. Além disso, projetos foram criados e submetidos a algumas entidades de fomento à pesquisa, bem como houve a participação em alguns eventos com o objetivo de entender o estado da arte relacionado a esta pesquisa. Esta seção apresenta estes dados.

### 1.5.1 Publicações Científicas

#### Publicações como primeiro autor

1. Leandro Bento Pompermaier, Rafael Chanin, Afonso Sales, Kellen Fraga, and Rafael Prikladnicki. **An empirical study on software engineering and software startups: Findings from cases in an innovation ecosystem.** In SEKE, pages 48–51, 2017. **(QUALIS B1)**

Um estudo empírico, em um contexto de incubadora de empresas, 8 *startups* de software foram pesquisadas e 4 *findings* relatados [PCS<sup>+</sup>17]:

- (a) A adoção de técnicas de gerenciamento de requisitos aumenta de forma considerável o tempo de desenvolvimento, o que é um problema para o lançamento do MVP no mercado.
- (b) Não documentação e / ou gerenciamento de requisitos não interfere na qualidade do MVP desenvolvido.
- (c) Uma definição deficiente ou não definição de uma estrutura e arquitetura de software na fase inicial de uma startup de software aumenta exponencialmente a dívida técnica da empresa.
- (d) A estruturação ou formalização de uma etapa de teste de software aumentará a aceitação pelo mercado dos requisitos desenvolvidos.

2. Leandro Pompermaier, Rafael Chanin, Afonso Sales, and Rafael Prikladnicki. **Mvp development process for software startups.** In International Conference on Software Business, pages 409–412. Springer, 2019. **(QUALIS B1)**

Este artigo apresenta uma proposta de um Processo de Desenvolvimento Mínimo Viável para *Startups* de Software que pode ser usado durante o processo de desenvolvimento MVP [PCSP19].

3. Leandro Pompermaier and Rafael Prikladnicki. **Brazilian startups and the current software engineering challenges: The case of tecnopuc.** In Fundamentals of Software Startups, pages 331–345. Springer, 2020.

Capítulo de livro que apresenta o estudo do case Tecnopuc apresentando oportunidades e desafios que as *startups* encontram quando localizadas em parques tecnológicos [PP20].

**Publicações resultantes da colaboração com colegas - contexto desta pesquisa**

A seguir são apresentados os trabalhos publicados em que o autor deste trabalho teve alguma participação, seja na coleta de dados ou revisão e complementação do relatório final. Foram listados somente os trabalhos em que o assunto principal está relacionado ao tema desta pesquisa.

1. Sohaib Shahid Bajwa, Xiaofeng Wang, Anh Nguyen Duc, Rafael Matone Chanin, Rafael Prikladnicki, Leandro Bento Pompermaier, and Pekka Abrahamsson. **Start-ups must be ready to pivot.** IEEE Software, 34(3):18–22, 2017 (**QUALIS A1**)

Neste artigo [BWD<sup>+</sup>17], há a definição de tipos de pivotagem e delimitação das razões para sua criação. Entre as mudanças relacionadas às pivotagens, as relacionadas ao produto e ao mercado são as mais comuns, sendo o tipo *zoom-in*, que consiste na característica do produto de se transformar no produto como um todo, o mais comum entre as *startups* de software.

2. el Chanin, Leandro Pompermaier, Kellen Fraga, Afonso Sales, and Rafael Prikladnicki. **Applying customer development for software requirements in a startup development program.** In Proceedings of the 1st International Workshop on Software Engineering for Startups, pages 2–5. IEEE Press, 2017 (**QUALIS A1**)

Esse artigo apresenta a experiência de três *startups* de software na utilização do desenvolvimento do cliente [Bla12] na definição dos requisitos do software [CPF<sup>+</sup>17]. Apesar do número restrito da amostra, há indicações de que o processo pode melhorar o entendimento das necessidades do cliente na concepção do produto.

3. Eriks Klotins, Michael Unterkalmsteiner, Panagiota Chatzipetrou, Tony Gorschek, Rafael Prikladnicki, Nirnaya Tripathi, and Leandro Bento Pompermaier. **Exploration of technical debt in start-ups.** In Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP '18, pages 75–84, New York, NY, USA, 2018. ACM (**QUALIS A1**)

Nesse artigo foi realizada uma análise do resultado de uma *survey* envolvendo 86 *startups* de software, com o intuito de identificar aspectos de dívida técnica e informações que caracterizam o contexto de engenharia em *startups* [KUC<sup>+</sup>18]. Foi descoberto que as *startups* acumulam a maior parte da dívida técnica na dimensão de testes, apesar das tentativas de automatizá-las. Além disso, descobrimos que o tamanho e a experiência da equipe de *startup* são um precedente importante ao acúmulo de dívida técnica: equipes maiores enfrentam mais desafios para manter a dívida sob controle.

4. Nirnaya Tripathi, Eriks Klotins, Rafael Prikladnicki, Markku Oivo, Leandro Bento Pompermaier, Arun Sojan Kudakacheril, Michael Unterkalmsteiner, Kari Liukkunen, and Tony Gorschek. **An anatomy of requirements engineering in software startups using multi-vocal literature and case survey.** Journal of Systems and Software, 146:130–151, 2018 (**QUALIS A2**)

Uma revisão bibliográfica multivocal (MLR) foi utilizada para verificar na literatura que elementos da engenharia de requisitos (elicitação, documentação, priorização e validação) são usados em *startups* de software [TKP<sup>+</sup>18]. Além disso, uma pesquisa (*survey*) foi utilizada para reunir dados empíricos de modo a alcançar o objetivo deste estudo. Como resultado, 36 artigos e 80 entrevistas forneceram informações sobre os processos de engenharia de requisitos. Dessa forma, a análise de dados revelou que as técnicas mais utilizadas pelas *startups* de software foram: análises de produtos similares (por exemplo, elicitação), usos de notas informais (por exemplo, para documentação), valores para clientes, produtos e partes interessadas (por exemplo, para priorização) e revisões internas / protótipos (por exemplo, para validação).

5. Eriks Klotins, Michael Unterkalmsteiner, Panagiota Chatzipetrou, Tony Gorschek, Rafael Prikladnicki, Nirnaya Tripathi, and Leandro Pompermaier. **A progression model of software engineering goals, challenges, and practices in start-ups**. IEEE Transactions on Software Engineering, 2019 (**QUALIS A1**)

Nesse artigo, um novo modelo de ciclo de vida para *startups* é proposto, com base na análise de 84 startups de software. Esse ciclo de vida leva em consideração a equipe, engenharia de requisitos, foco em valor, qualidade e testes, arquitetura e projeto e gerenciamento de projetos de software.

6. Edson Oliveira Jr et al. **Surveying the impacts of COVID-19 on the perceived productivity of Brazilian software developers** In: Proceedings of the 34th Brazilian Symposium on Software Engineering, 2020, pp. 586–595 (**QUALIS B2**)

Artigo que apresenta o resultado de uma *survey* relacionada a análise dos impactos do isolamento social devido à COVID-19 na produtividade dos desenvolvedores de software brasileiros. Utilizou-se uma amostra de 413 respondentes indicando que a produtividade dos desenvolvedores aumentou, principalmente por haver menos interrupções durante o horário de trabalho [OJLV<sup>+</sup>20].

### 1.5.2 Projetos

- Projeto "Um Modelo para Desenvolvimento de Software no contexto de *Startups* Digitais- FAPERGS – Fundação de Amparo a Pesquisa do Estado do Rio Grande do Sul - Edital 02/2017 – Pesquisador Gaúcho
- Projeto Institucional de Internacionalização – CAPES-PrInt



### 1.5.3 Participação em eventos

Durante o período deste doutorado o autor da proposta participou dos seguintes eventos relacionados ao tema de empreendedorismo e *startups*:

1. SEKE - the 29th International Conference on Software Engineering & Knowledge Engineering - Pittsburgh - EUA
2. Conferência Anprotec 2017 - Rio de Janeiro - RJ
3. Conferência Anprotec 2018 - Goiânia - GO
4. Conferência Anprotec 2019 - Florianópolis - SC
5. IASP 2017 - World Conference on Science Parks and Areas of Innovation - Istambul - Turquia
6. INBIA 2017 - the 31st International Conference on Business Incubation - Seattle - EUA
7. INBIA 2018 - the 32nd International Conference on Business Incubation - Dallas - EUA
8. CASE 2018 - Conferência anual de startups e empreendedorismo - São Paulo - SP
9. CASE 2019 - Conferência anual de startups e empreendedorismo - São Paulo - SP
10. CASE 2020 - Conferência anual de startups e empreendedorismo - virtual
11. WebSummit - Virtual
12. Gramado Summit 2018 - Gramado - RS

## 2. BASE TEÓRICA

Neste Capítulo, os principais conceitos associados a este trabalho são apresentados.

### 2.1 Empreendedorismo e Inovação

Empreendedorismo pode ser conceituado como a descoberta de oportunidades e subsequente criação de nova atividade econômica, muitas vezes através da criação de uma nova organização [Rey05]. O empreendedorismo está associado à inovação e sua essência está na percepção e no aproveitamento das novas oportunidades de negócios, em criar uma nova forma de uso dos recursos, deslocados de seu emprego tradicional e sujeitos a novas combinações [Whi16]. O empreendedorismo inovador está relacionado à tecnologia, e muitas das soluções que são apresentadas nos dias de hoje foram criadas com base na computação, por exemplo: Uber, AirBnb, Netflix, entre outros. Essa capacidade de criação de novos negócios a partir do uso do conhecimento de computação pode ser explicada pela inserção do ensino de empreendedorismo nos currículos dos cursos de computação nas principais universidades do mundo.

O uso da tecnologia por empresas visa a obtenção de alguma vantagem tecnológica em relação a diferentes *players* no mesmo mercado de atuação, fazendo com que a adoção desse diferencial possa auxiliar no crescimento do negócio e na sua sustentação [Dru14]. Porém, poucas empresas conseguem inovar em tecnologia na velocidade que o mercado exige. Nesse ponto, as *startups* de software exploram e desenvolvem tecnologias inovadoras dedicadas a promover o crescimento pelo uso destas [Kiv18].

Hoje, existem inúmeras tecnologias com ideias inéditas e inovadoras, que, combinadas com outros fatores, podem ajudar a alcançar o sucesso [Gup16]. Esta era digital centrada no empreendedorismo e na inovação, fortemente baseada em computação, influencia o mercado consumidor na adoção cada vez mais rápida dos produtos e serviços criados, acelerando o desenvolvimento da sociedade [McD14].

#### 2.1.1 *Startup*

*Startups* possuem um papel fundamental no processo de transferência da inovação para a sociedade [DFG03] [CP08]. Existem diversos conceitos relacionados a *startups* em diferentes contextos. Uma das primeiras definições foi realizada por Carmel [Car94] in-

dicando que as *startups* seriam empresas jovens. Para Steve Blank [Bla12], *startup* é uma organização temporária projetada para buscar um modelo de negócio repetível e escalável. Neste caso, o foco é o modelo de negócios apresentado pelo empreendimento, conceito que é também apresentado pelo Instituto Brasileiro de Governança Corporativa (IBGC), o qual indica que uma *startup* é uma organização escalável, de alto potencial econômico e inovadora (não necessariamente ligada à tecnologia) [LAM19]. Eric Ries [Rie09] definiu *startup* como sendo uma instituição de pessoas projetada para a criação de produtos ou serviços em condições de extrema incerteza. Por isso, as *startups* devem ter uma estratégia para desenvolver seu protótipo e, em seguida, o produto para garantir uma base de mercado para seus novos produtos e serviços, porém elas têm acesso limitado aos recursos [EWA15].

Nesta linha, o conceito de *startup* de software surge como sendo organizações que têm experiência limitada, que trabalham com limitações de recursos e recebem influências de diferentes fontes, como investidores, mercado e competidores, e usam a tecnologia para dinamizar seus produtos ou serviços [Cro02]. Paternoster [PGU+14] conceitua *startups* de software como aquelas organizações focadas na criação de produtos inovadores e de alta tecnologia, com pouco ou nenhum histórico operacional, com o objetivo de expandir seus negócios em mercados altamente escaláveis de forma agressiva. No contexto deste trabalho, *startup* e *startup* de software serão utilizados como sinônimos.

Para uma *startup*, o mais relevante é definir um modelo de negócio para entender exatamente o seu posicionamento de mercado. Há muitas definições de modelo de negócio, mas a mais difundida e utilizada pelas *startups* é a apresentada por Osterwalder [OP10], que descreve a lógica de como uma organização cria, entrega e captura valor. Na acepção de Osterwalder [OP10], a descrição de modelo de negócio utiliza nove blocos de construção que mostram como uma empresa pretende ganhar dinheiro e entregar valor ao cliente. São eles [OP10]:

1. **Segmentos de Clientes:** Definem diferentes grupos de pessoas ou organizações que uma empresa pretende alcançar e servir.
2. **Proposições de Valor:** Descrevem os pacotes de produtos e serviços que criam valor para um segmento de clientes específico. A proposição de valor é a razão pela qual os clientes escolhem uma empresa em detrimento de outra.
3. **Canais:** Os canais consistem na forma como uma empresa se comunica com seus segmentos de clientes para oferecer uma proposta de valor.
4. **Relacionamento com o Cliente:** O elemento de relacionamento com o cliente descreve os tipos de relações que uma empresa estabelece com segmentos específicos de clientes. A empresa deve esclarecer o tipo de relação que pretende estabelecer com cada segmento de clientes.

5. **Fontes de receita:** A renda que uma empresa gera com cada segmento de clientes é representada no quadro "Fontes de receita do modelo de negócios".
6. **Atividades principais:** Cada empresa precisa realizar uma série de atividades para cumprir com êxito as necessidades do cliente. As atividades principais são aquelas atividades que são de extrema importância para a empresa e permitem que ela funcione com êxito.
7. **Recursos-chave:** Cada modelo de negócio requer ativos ou recursos-chave para permitir que uma empresa possa criar e oferecer uma proposição de valor, alcançar os mercados, manter relações com segmentos de clientes e arrecadar receitas.
8. **Parcerias-chave:** As parcerias-chave constituem a rede de fornecedores que fazem o trabalho do modelo de negócio. Geralmente existem três razões para a criação de parcerias: otimização e economia de escala, redução de risco e incerteza e aquisição de recursos e atividades particulares.
9. **Estrutura de Custos:** O elemento de estrutura de custo descreve todos os custos incorridos para operar um modelo de negócio.

Uma *startup* não pode ser confundida com uma pequena empresa, pois, embora uma empresa estabelecida (mesmo pequena) tenha um modelo de negócio validado, as *startups* estão em busca de um modelo de negócio. Nesse sentido, elas apresentam alguns aspectos recorrentes [PGU<sup>+</sup>14]:

- Pouca experiência;
- Limitação de recursos;
- Várias influências;
- Tecnologia e mercados dinâmicos.

O desenvolvimento de uma *startup* de software tem um ciclo de vida diferente de uma empresa de software tradicional, isto é, do ponto de vista do processo, as *startups* estão mais preocupadas em tentar sobreviver ao mercado do que definir procedimentos [CO08a]. Nesse sentido, Steve Blank [Bla12] elaborou um processo de quatro etapas para o desenvolvimento inicial de uma *startup*: *Customer Discovery*, *Customer Validation*, *Customer Creation* e *Company Building*. Nas fases de descoberta e validação do cliente, o Produto Mínimo Viável (MVP) é construído e usado para validar as hipóteses das necessidades do cliente [DA16]. Além disso, de acordo com Bosch [Bos16], as *startups* de software passam por vários estágios antes de atingirem os níveis de maturidade, ou seja, precisam passar por várias etapas até poderem criar um modelo de negócio que funcione e seja viável.

De forma resumida, os passos podem ser descritos como segue (Figura 2.1):

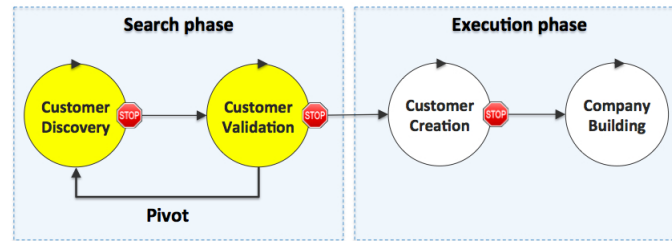


Figura 2.1 – Processo de desenvolvimento do cliente [Bla12]

- **Descoberta do cliente:** É a fase de testar hipóteses de mercado e entender o problema real dos consumidores. Basicamente, crie um plano para testar as reações do cliente a essas hipóteses e transforme-as em fatos [Bla12].
- **Validação do cliente:** Consiste no desenvolvimento de um modelo de negócio replicável e escalável, ou seja, em testar se o modelo de negócio resultante é repetível e escalável [Bla12].
- **Criação do Cliente:** Início da execução. Aqui, cria-se a demanda do usuário final, que é direcionada para o canal de vendas para que o negócio escale [Bla12].
- **Criação da Empresa:** Transição de *startup* para uma empresa [Bla12].

Normalmente, as decisões relacionadas ao desenvolvimento do produto de uma *startup*, como o uso de práticas de Engenharia de Software ou não, fazem parte da equipe executiva da *startup* [Som10]. No entanto, o ganho inicial obtido em termos de flexibilidade e velocidade é contrabalançado pela necessidade de reestruturar o produto, fase em que o negócio começa a crescer [GWA14].

#### 2.1.1.1 Produto Mínimo Viável - MVP

A definição de Produto Mínimo Viável (MVP) é um conceito fundamental da metodologia *lean startup* [Rie11]. A definição de MVP é um processo iterativo baseado no feedback contínuo obtido dos primeiros usuários. O termo MVP foi definido por Frank Robinson [Rob01] em 2001 e depois disseminado por Eric Ries [Rie11] e Blank [Bla14]. Este é um conceito em contínua evolução, definido por Eric Ries [Rie11] como “uma versão de um novo produto, que permite a uma equipe coletar o máximo de aprendizado validado sobre os clientes com o mínimo de esforço”.

Introduzir um novo produto ou um serviço no mercado é um exercício complicado, com *feedback* desconhecido dos clientes, pois requer determinação, visão e recursos para poder inserir esse produto no mercado [Mis14]. Esses elementos ajudarão o novo produto a competir com outro produto existente ou a torná-lo conhecido para ser utilizado pelos usuários finais.

Para empresas já estabelecidas é exigido pouco esforço para divulgar o novo produto ou serviço, visto que elas já atendem diferentes clientes em diferentes mercados. Portanto, um novo negócio estabelecido identifica um problema que afeta o mercado e acredita que o novo produto ou serviço será adotado pelo mercado-alvo, uma vez que restringe esse problema. Assim como empresas estabelecidas, as empresas iniciantes planejam cuidadosamente uma estratégia, desenvolvendo um produto completo, conforme exigido no mercado, com base na visão proposta, antes de liberá-lo para o mercado-alvo [Mis14]. Este ato de liberar um produto totalmente desenvolvido para o mercado antes de testá-lo é uma proposição arriscada, devido à alta incerteza na participação de mercado [Rie11]. Isso repercute mais nos negócios em desenvolvimento, pois eles operam em ambientes incertos, ao contrário dos negócios estabelecidos, que operam com clientes conhecidos e servem-se deles na determinação de mercados [Bla12].

Diante disso, um MVP deve ser um produto com recursos suficientes para satisfazer os primeiros clientes, o que fornecerá *feedback* para a decisão final necessária ao desenvolvimento completo do produto em um futuro próximo, conforme declarado por Ries [Rie11]. Essa técnica é usada para testar o mercado e é benéfica para a empresa, pois ajuda a melhorar a nova ideia para atender às necessidades do cliente. Nesse sentido, a maioria das empresas enfrenta os desafios de gerenciar a complexidade do produto e as necessidades de medir e avaliar o impacto do teste.

Segundo Ries [Rie11], existem três maneiras de considerar a avaliação desses impactos. A primeira maneira, que ele chamou de "*the sticky engine*", em que a empresa se funde com outra empresa estabelecida que está produzindo quase o mesmo produto e desenvolve um produto final, liberando-o para o mercado já existente, determinado pela empresa estabelecida. Com o uso dessa estratégia, as empresas medem o produto usando o *feedback* do mercado e também acompanham a quantidade de vendas feitas pela empresa principal. Menos vendas da empresa significam que os clientes estão deixando de usar o produto ou há pedidos para a melhoria deste. Quando ocorrem mais vendas, isso indica que o produto final é bem-aceito pelo mercado. Por exemplo, um negócio de inicialização de microprocessador pode se fundir com uma empresa de *smartphone* e projetar um produto instalando o primeiro no *smartphone*.

Na segunda maneira, que Ries [Rie11] chamou de "*viral engine*", em que o crescimento de uma nova ideia depende do produto existente no mercado da mesma empresa, a empresa divulga um produto existente com detalhes que descrevem o novo produto. Em grande parte das vezes, as empresas distribuem esses novos produtos ou serviços de forma gratuita.

Uma terceira via é indicada por Ries [Rie11] como "*paid engine*", na qual a empresa relaciona a receita vitalícia do cliente com os custos de aquisição deste. Quanto maior a diferença, maior a taxa de crescimento da empresa.

Apesar dos conceitos de MVP apresentados por [Rie11] e [Bla12], este conceito possui diversas definições que foram apresentadas em diferentes publicações. O estudo apresentado por [LT16] mostra a evolução desse conceito. Uma primeira definição de MVP, apresentada por [Rob01] possui uma conotação econômica:

*MVP is not a minimal product, it is a strategy and process directed toward making and selling a product to customers.*

Porém, o conceito mais conhecido e mais referenciado [LT16] é o apresentado por Ries [Rie09].

*MVP is a version of a new product, which allows a team to collect the maximum amount of validated learning about customers with the least effort. It is an iterative process of idea generation, prototyping, presentation, data collection, analysis and learning.*

Após a definição de MVP por Eric Ries, outras definições foram publicadas. Steve Blank [Bla14] publicou no seu blog a indicação de que um MVP possui apenas aqueles recursos (e não mais) que permitem que o produto seja implantado. Kniberg [K112] coloca que MVP significa liberar cedo e frequentemente, e aprendizado validado significa usar métricas e testes A / B para descobrir o que realmente funciona e o que não funciona. Poole [Poo12] coloca o MVP como uma ferramenta comercial, mostrando que o MVP é uma oferta que gera receita para a empresa e motiva os clientes a fornecer *feedback* e recomendá-la a outros clientes em potencial.

Ries [Rie11] apresenta uma classificação dos tipos de MVPs que são discutidos entre os empreendedores:

- *Explainer video*: uma animação curta que explica o que seu produto faz e por que os usuários devem comprá-lo.
- *Landing Page*: uma página da web na qual os visitantes “aterrissam” após clicar em um link de um e-mail ou outro tipo de campanha. Serve para comunicar rapidamente as propostas da *startup*.
- *Wizard of Oz*: uma interface de usuário que se parece com um produto real em funcionamento, mas na qual o processo de negócios real é realizado manualmente. O objetivo deste MVP é demonstrar o trabalho completo feito pelo produto.
- *Concierge MVP*: um serviço manual que consiste exatamente nas mesmas etapas que os usuários seguiriam com o produto.
- *Piecemeal MVP*: semelhante ao *Wizard of Oz*, com a diferença de que a execução das tarefas é feita usando ferramentas existentes.

- *Mockup MVP*: como protótipo de papel e *wireframe*, representava a interface de usuário do produto sem qualquer funcionalidade.
- *Public project proposal*: uso de plataformas de *crowdsourcing* e/ou *crowdfunding* para validar a aceitação do mercado.
- *Single feature MVP*: um protótipo que implementa a função mais importante do produto.
- *Rip off MVP*: um produto de sucesso para se obter *feedback* e, em seguida, girar em uma direção diferente com outro produto.

O termo "protótipo" também é frequentemente usado no contexto das *startups* como um termo intercambiável com MVP [DA16]. Segundo Duc et. al. [DA16], existem diferentes tipos de protótipos de software frequentemente usados nas fases iniciais do desenvolvimento de software, como descartáveis ou protótipos rápidos, que consomem muito poucos esforços com análise de requisitos mínimos para construir um protótipo. Um ponto importante e comumente utilizado pelas *startups* é o protótipo evolutivo [Flo84], que é a construção de uma funcionalidade para uso real e, a partir disto, o protótipo evolui para o sistema. Outro tipo de protótipo usado pelas *startups* é o chamado protótipo incremental [Flo84], que baseia-se na construção de vários protótipos funcionais dos vários subsistemas para uma futura integração e definição do sistema completo.

Para esta pesquisa, serão considerados somente os MVPs ou protótipos que utilizem desenvolvimento de software para sua construção.

Em resumo, *startups* são empresas que desenvolvem uma nova ideia sob condições de maior incerteza [Rie11]. Uma *startup* opera em torno de uma visão de que seu produto resolverá unicamente os problemas prementes dos clientes em seu mercado alvo. Seus fundadores geralmente esperam que o produto entregue um retorno sem precedentes ao investimento. Essa visão inclui duas suposições importantes: a suposição de fornecimento de valor (isto é, a hipótese de valor) e a suposição de crescimento no mercado (ou seja, a hipótese de crescimento). Dessa forma, o MVP oferece à maioria das empresas a capacidade de identificar um problema e também fornecer o melhor produto ou serviço aos clientes, moldando-o de acordo com a preferência do cliente.

### 2.1.2 Modelo de Maturidade de *Startups* Proposto pelo IBGC

O Instituto Brasileiro de Governança Corporativa (IBGC) desenvolveu um guia para apoiar *startups* e *scale-ups* na adoção de práticas de governança e gestão sob uma perspectiva da sustentabilidade e sucesso do negócio [LAM19]. O objetivo deste guia é o de



gerar conhecimento em governança corporativa e influenciar na adoção das melhores práticas.

Este modelo foi utilizado como base para esta pesquisa, principalmente para entender os diferentes momentos de maturidade de uma *startup*, do ponto de vista de negócio, e verificar dentro de cada uma destas fases que práticas de Engenharia de Software são adotadas pelos empreendedores na construção dos seus produtos de software.

Como já definido anteriormente (seção 2.1.1), uma *startup* é uma organização escalável, de alto potencial econômico e inovadora e uma *scale-up* é o passo seguinte de uma *startup* que superou os primeiros obstáculos do negócio, comprovando que o modelo de negócios é escalável, inovador e de crescimento acelerado.

A definição de governança corporativa dada pelo IBGC [LAM19]:

"Governança corporativa é o sistema pelo qual as empresas e demais organizações são dirigidas, monitoradas e incentivadas, envolvendo os relacionamentos entre sócios, conselho de administração, diretoria, órgãos de fiscalização e controle e demais partes interessadas"

A governança corporativa assenta-se em quatro princípios básicos: transparência, equidade, prestação de contas (*accountability*) e responsabilidade corporativa.

No início, o negócio é pouco mais que uma ideia. Depois, ele já está operacional, testando o mercado e o modelo de negócios. Se estrutura melhor e começa a conquistar mais clientes, até que, em certo momento, já está bem estabelecido, torna-se uma *scale-up*, e o seu desafio é crescer aceleradamente escalando suas atividades sem que os custos cresçam no mesmo ritmo. Este modelo de maturidade de *startup* do IBGC sugere as práticas no campo da governança, de forma a orientar a atuação dos empreendedores. As recomendações podem variar conforme o desenvolvimento específico de cada *startup*. Esta edição do IBGC Segmentos [LAM19] se propõe a abordar as particularidades da governança nas quatro fases de crescimento do negócio:

- Ideação
- Validação (MVP)
- Tração (Product Market Fit)
- Escala

As empresas bem-sucedidas que chegam à maturidade costumam passar com êxito por estas quatro fases em seu trajeto de evolução. A governança requerida varia de acordo com o momento e a fase de desenvolvimento da empresa [LAM19]. Em cada uma dessas fases a governança leva em conta quatro pilares:

- estratégia & sociedade;
- pessoas & recursos;
- tecnologia & propriedade intelectual;
- processos & *accountability*.

### 2.1.2.1 Ideação

Na fase de ideação (hipótese), a *startup* encontra-se entre o desenvolvimento da ideia e o entendimento do problema que se propõe a resolver ou a lacuna do mercado que se propõe a ocupar. É a fase de concepção e de união entre pessoas que se ligam com um propósito definido e começam a desenvolver o que poderá se tornar uma empresa no futuro. O negócio ainda não tem, necessariamente atividade operacional nem existência formalizada do ponto de vista jurídico. Enfoque da governança é o de estruturar os papéis e as responsabilidades dos sócios, especificar as formas de contribuição e a intensidade de dedicação, a remuneração e futura participação, bem como opções de saída e descontinuidade. Garantir a titularidade da propriedade intelectual da sociedade e o alinhamento entre os sócios, o processo de tomada de decisão e a construção de consenso.

No momento da concepção do negócio, os fundadores de uma *startup* podem contribuir de formas distintas: alguns aportam capital, outros dedicam tempo ou conhecimento ao desenvolvimento da futura empresa. Após o início das atividades operacionais da empresa, é desejável que todos os fundadores passem a colaborar em tempo integral e de modo exclusivo com o negócio. Cada sócio tem que estabelecer como irá agregar à sociedade: com capital financeiro ou intelectual, especificando aportes financeiros e volume de tempo disponível. As ideias não têm valor financeiro ou econômico. Elas podem ser a base de um negócio inovador e de alto potencial, mas só adquirem valor por meio da sua efetiva implementação. Enquanto a empresa ainda não tem faturamento e não pode pagar salários, os sócios dependem de recursos próprios para seu sustento pessoal. Estabelecer a própria capacidade financeira pessoal e individual para o período pré-operacional e calcular quanto tempo esses recursos vão durar. Estipular previamente quais são as condições para que cada fundador receba a sua participação societária é uma boa prática de governança nas *startups*.

É desejável que os fundadores assumam um compromisso de permanecer nela pelo tempo necessário para desenvolvê-la. Essa condição é também conhecida como *re-vesting*. Ela visa evitar a atribuição de uma determinada participação societária a pessoas que dedicaram pouco tempo e descumpriram o compromisso de colaboração por um prazo minimamente aceitável. Numa *startup*, o ideal é que os fundadores definam as regras de saída e/ou entrada de novos sócios para evitar problemas no futuro. A fase de ideação é um momento perfeitamente adequado para esse tipo de combinação, para estabelecer regras

simples de ingresso e de saída de sócios. Isso pode ser feito por meio de um acordo de fundadores (*founders agreement*), que traz as formas de contribuição de cada sócio.

Na fase de ideação, na maioria dos casos, os talentos e recursos da futura empresa se concentram nos fundadores. O pilar de pessoas & recursos tem menor prioridade no campo da governança corporativa. É recomendável avaliar a conveniência de se empreender de forma individual, como um único sócio.

É necessário abordar questões relacionadas à propriedade intelectual já na fase de ideação. A proteção da propriedade intelectual poderá ser tratada dentro do próprio *founders agreement* mencionado anteriormente ou em contrato particular entre os fundadores.

Sobre o aspecto estritamente financeiro, é recomendável um controle de caixa com registro de entradas e saídas, revisado periodicamente e projetado para curtos períodos de tempo. Dessa forma os processos servem para:

- Alinhar as expectativas dos fundadores em relação à *startup*.
- Discutir sobre a capacidade financeira e a perda suportável pelos fundadores.
- Estabelecer como cada sócio irá contribuir para a sociedade.
- Dimensionar a colaboração de cada um.
- Estipular condições para o recebimento de participações societárias (*revesting*).
- Estabelecer as condições de ingresso e saída de sócios.
- Proteger a propriedade intelectual gerada e os segredos de negócios.
- Redigir, assinar e manter vias físicas do acordo de fundadores (*founders agreement*).

#### 2.1.2.2 Validação

Na fase de validação, as proposições e suposições levantadas na primeira fase serão realmente testadas.

Constituir a empresa e organizar regramentos quanto a direitos e deveres dos sócios, incluindo as primeiras reflexões sobre o propósito da organização. Organizar práticas referentes a potenciais empregados-chave e a relação com clientes e parceiros estratégicos. Nessa fase, a empresa já está formalizada, de fato, mas também do ponto de vista jurídico. O acordo de fundadores (*founders agreement*) firmado na fase de ideação e que estabelecia as bases da sociedade pode ser revisado e desdobrado em dois novos documentos: o estatuto ou contrato social e o acordo de sócios. O estatuto ou contrato social é um documento obrigatório de constituição de uma sociedade e deve ser levado para registro público na junta comercial do estado onde se estabelece o endereço da sede da empresa.

O estatuto regula aspectos obrigatórios para o funcionamento da sociedade: especifica o seu nome (razão social), capital social, objeto social, localização, quadro societário e quem a representa, entre outros itens. A legislação brasileira prevê a existência de diferentes estruturas societárias. Num primeiro estágio, geralmente se opta por uma sociedade limitada. Em estágios mais avançados, costuma-se estabelecer uma sociedade anônima (SA) de capital fechado, que oferece mais flexibilidade para o ingresso de novos sócios e para exigências adicionais.

O único documento obrigatório para fundar uma sociedade empresária é o ato de constituição. No caso de uma sociedade limitada, esse ato chama-se contrato social. No caso de uma sociedade anônima de capital fechado, chama-se estatuto social. O acordo de sócios é um documento privado, arquivado na sede da empresa e que não precisa ser registrado na junta comercial. O acordo de sócios pode prever que tudo o que for produzido em termos de propriedade intelectual (softwares, marcas, etc.) seja patrimônio da sociedade e não da pessoa física que o desenvolveu. Os temas do acordo de fundadores (*founders agreement*) são, em geral, formalizados em outros documentos:

- Informações obrigatórias sobre a sociedade (natureza pública) : contrato ou estatuto social.
- Informações facultativas sobre o relacionamento dos sócios (natureza privada): acordo de sócios.

Os acordos de sócios tratam basicamente de dois tipos de temas: direitos patrimoniais (aqueles que impactam no valor das participações dos sócios) e direitos políticos (aqueles que impactam na gestão e controle do negócio). Os principais temas contemplados são: exercício do poder de controle; A partir dessa fase, é importante compreender a diferença entre o patrimônio pessoal de cada sócio e o patrimônio da empresa. Da mesma forma, é interessante que se estabeleçam as regras de saída da sociedade, bem como as condições para que isso aconteça. O acordo de sócios pode especificar o método de apuração do valor das suas quotas (ou ações), a forma de pagamento dos respectivos haveres sociais ou regras de compra e venda da participação societária do sócio que deixa o negócio. Chegamos agora às cláusulas da resolução de conflitos. É necessário estabelecer como a empresa irá resolver os conflitos societários: por mediação, por arbitragem ou no poder judiciário. Caso os sócios optem por resolver seus problemas por mediação e/ou arbitragem, devem prever isso de forma expressa no contrato social e no acordo de sócios.

Caso o contrato social e o acordo de sócios omitam o tema, automaticamente se estabelece o poder judiciário (do estado onde fica a sede/matriz da sociedade) como o ambiente para a resolução dos conflitos. Cada uma das opções tem suas vantagens e desvantagens. A mediação pode servir como uma primeira tentativa de resolver a questão, mas, quando essa tentativa não é bem-sucedida, faz-se necessário recorrer à arbitragem

ou ao judiciário. A arbitragem se destaca pela maior rapidez, pelo sigilo e pela possibilidade de ter especialistas, mas sua principal desvantagem é o maior gasto (custas, honorários, etc.). O poder judiciário possibilita menores gastos (custas, honorários, etc.), mas o prazo para a decisão final é mais longo, pois esta se sujeita a variadas instâncias e recursos. Na fase de validação, é comum que a *startup* comece a captar recursos de investidores (por exemplo, por meio de programas de aceleração ou de investidores anjo).

Esses recursos são captados por meio de investimento na empresa mediante aquisição de participação societária ou por meio de empréstimo de dinheiro conversível em quotas ou ações (também conhecido como mútuo conversível). É muito importante informar-se sobre os efeitos de um *down round* nas captações de recursos por meio de mútuos conversíveis. O *down round* ocorre quando existem operações sucessivas de captação, mas a operação subsequente é realizada com base em um *valuation* menor do que o *valuation* da rodada de captação anterior. Nesses casos, pode ocorrer a diminuição de participação no capital social e, simultaneamente, a perda de valor patrimonial.

Também é necessário considerar se a aquisição de determinada participação por um sócio restringirá as possibilidades de ingresso de novos sócios ou investidores no futuro. Ao aportar capital na sociedades, o novo investidor costuma exigir alguns direitos. A negociação deve ser conduzida com ponderação: os fundadores precisam ter em conta que essa situação se repete a cada rodada de investimento.

Na fase de validação não são apenas os novos investidores que passam a fazer parte da *startup*. Assim, torna-se necessário alinhar expectativas e contrapartidas de relações estabelecidas com mentores, *advisors*, conselheiros consultivos e, no futuro, conselheiros de administração que passem a executar tarefas e atividades em favor da empresa. Para atrair e reter talentos no negócio, é comum começar a avaliar a possibilidade de constituição de *option pools* e contratos de opções de compra de ações (*stock options*). O *stock option pool* é a parcela do capital social da empresa destinada aos funcionários por meio de opções de compra de quotas ou ações a serem adquiridas, na maioria das vezes, por um preço simbólico. Para tornar operacional a sua utilização adequada, é necessário realizar o detalhamento em duas frentes: o primeiro passo é estabelecer a fatia de capital e as regras gerais para sua utilização, incluindo, por exemplo, precificação da participação, posições-chave a serem contempladas, e as regras de manutenção, recompra ou devolução da participação em situações de encerramento da relação de trabalho. O segundo passo é aprovar as condições de negociação no plano individual para cada pessoa-chave contemplada pelo programa, definindo se as condições particulares para receber a participação prometida e as alçadas de decisão internas para que se ofereça a alguém essa participação. Ambas as situações, são em geral, deliberadas e aprovadas previamente em reuniões de sócios e/ou investidores. Essas regras, que estipulam os prazos para a obtenção de participação societária, são conhecidas como cláusulas de *vesting*. A cláusula de *vesting* determina o período total de tempo que o profissional deve trabalhar na empresa

para alcançar o percentual máximo de participação societária. Durante o período de tempo de *vesting* são estabelecidos *cliffs*. Os *cliffs* são datas ou marcos específicos atingidos pelo profissional e, uma vez alcançados, asseguram o acesso à parcela da participação societária total combinada. Sobre propriedade intelectual, é importante a *startup* realizar o registro em várias jurisdições, pois muitas vezes a proteção assegurada pelo registro está limitada ao território de cada país em que esse registro é realizado. Na fase de validação, a *startup* apresenta uma atividade operacional inicial e, por consequência, eventuais receitas e despesas.

Esse é o momento de assegurar a existência de controles internos organizados de apuração de resultados, que permitam, inclusive, a prestação de contas a eventuais investidores. É importante que os sócios entendam o conceito e a relevância das demonstrações financeiras e do gerenciamento de caixa. Do ponto de vista tributário, é o momento de buscar apoio para entender os regimes de enquadramento fiscal da *startup*. É importante lembrar que o tratamento e o pagamento inadequado de tributos são tratados pela lei brasileira como crime.

Nessa fase da *startup* também é hora de avaliar a formalização de contratos com clientes e o desenvolvimento de termos de uso de softwares. Os contratos com os clientes finais ajudam a preservar o valor da empresa.

### 2.1.2.3 Tração

Nessa fase, denominada tração, o serviço/produto oferecido pela *startup* está validado e os desafios principais consistem em conquistar clientes e aumentar o faturamento sem abrir mão dos princípios e valores da organização. Depois de passarem pelo período conhecido como "Vale da Morte" das *startups*, logo após a ideação e a validação do modelo de negócio, os empreendedores chegam a um ponto de inflexão importante: entram nas fases de tração e de escala, que caracterizam as *scale-ups*. Na primeira, a prioridade do negócio está na construção de uma base sólida que o levará a escalar em alta velocidade.

A *startup* já é uma *scale-up*. Todos os recursos devem ser utilizados no melhor interesse da empresa. Na fase de tração, a *scale-up* dá os primeiros sinais de solidez: formou um mercado consumidor e criou um modelo de negócios validado. A implantação de um conselho (consultivo ou de administração) é cada vez mais necessária para dar apoio estratégico aos sócios na jornada de crescimento da organização. O conselho de administração é definido pelo IBGC como "o órgão colegiado encarregado do processo de decisão de uma organização em relação ao seu direcionamento estratégico". Nessa fase, é uma boa prática a formalização e disseminação da cultura e do direcionamento estratégico da organização.

Recomenda-se a existência de um planejamento formal do negócio que inclua recursos, metas e indicadores que permitam a gestão estruturada do desempenho e a

prestação de contas para os públicos interno e externo. É recomendável que o empreendedor entenda bem a relação com os investidores de acordo com as regras do país em que o recurso foi captado, que não necessariamente são iguais às normas do mercado local. Nesse sentido, o ingresso de sócios estrangeiros pode trazer benefícios, mas também pode agregar mais complexidade, novos riscos e necessidade de conhecimento de diferentes legislações.

#### 2.1.2.4 Escala

Nessa última fase de escala (crescimento), a empresa já está estabelecida. Seu desafio é crescer em um ritmo acelerado, garantido a exploração das oportunidades e a expansão do negócio em termos geográficos, de mercado ou produtos, conforme a pertinência da estratégia da organização. A *scale-up* já conta, por exemplo, com conselheiros consultivos ou de administração, que colaboram para o direcionamento estratégico da organização. Os fundadores passam a ter mais um papel executivo. Para tanto, é necessário que amadureçam a sua visão de empresa e negócio em uma função executiva, dedicando-se a disseminar a cultura, definir as linhas de atuação, criar condições para atrair talentos e estruturar um time capacitado para conduzir o empreendimento a longo prazo.

Deve-se considerar a possibilidade de incluir conselheiros externos e independentes em sua composição. O código de conduta expressa os princípios éticos e valores, dá diretrizes e orienta a atuação de todos os envolvidos com o negócio, principalmente em relação a processos críticos. Para a tomada de decisões baseadas na ética, a organização pode ter como base quatro aspectos: a sua identidade, o código de conduta, as diversas políticas que regulam aspectos críticos e de conformidade e, finalmente, a estratégia da companhia e o seu sistema de gerenciamento de riscos corporativos. Decisões éticas baseiam-se em:

- identidade da empresa;
- código de conduta;
- política da organização;
- estratégias; e
- gerenciamento de riscos corporativos.

Recomenda-se promover um aumento progressivo de proteção, com especial atenção para temas como confidencialidade, não competição, proteção aos segredos de negócios e revisão sistemática do investimento no portfólio de propriedades intelectuais. Como mecanismos de defesa de seus interesses, a empresa deve estar atenta à atuação de

ex-sócios, conselheiros ou executivos que estejam sujeitos a obrigações e deveres de confidencialidade ou não competição.

Para lidar com a crescente complexidade e em busca de uma estrutura de governança mais robusta e madura, nessa fase podem ser implantados órgãos de fiscalização e controle complementares.

É importante que a empresa padronize a forma de divulgação de informações entre seus sócios e investidores, de forma a otimizar os esforços de relacionamento com eles. Ao término da fase de escala, é esperado que a empresa tenha alcançado o estágio avançado nos quatro pilares do modelo de governança em negócios com alto potencial de crescimento. No primeiro pilar, ela adotou um processo formal de revisão e aprovação da estratégia e implementou um processo de gestão de riscos e *compliance*, dedicou-se a promover a postura ética e criou um código de conduta, implementou mecanismos de atração e retenção de talentos e estipulou um plano de sucessão, avançou na proteção da propriedade intelectual e caminhou para uma estrutura mais robusta de processos e prestação de contas, por meio da implementação de mecanismos de controles (como a auditoria independente).

### 2.1.3 *Startups* e Engenharia de Software

O desenvolvimento de software em uma *startup* é algo visto como um desafio, principalmente quando o tema está relacionado ao uso de metodologias de desenvolvimento e o uso de ferramentas para estruturar e acompanhar as atividades. As *startups* possuem estruturas organizacionais enxutas e altamente flexíveis para conseguir compreender as necessidades do mercado e a entrega rápida de um produto ou serviço que atenda a uma parcela dos clientes, que muitas vezes não estão bem definidos na concepção do negócio [Bla12].

Alguns estudos apresentaram modelos para conduzir atividades de desenvolvimento de software em *startups*, porém não apresentaram resultados significativos. Coleman [CO08b] fez um estudo de como o processo de software é utilizado por *startups*, concluindo que nas fases iniciais as equipes de desenvolvimento destes negócios nascentes utilizam qualquer processo que atenda os objetivos do negócio. Indica ainda que as *startups* dependem muito da experiência dos profissionais que integram a equipe de desenvolvimento.

Um framework proposto por Bosch et. al [BHOBL13] chamado de *Early Stage Software Startup Development Model* (ESSSDM) oferece suporte ao processo de tomada de decisões para *startups* nas fases iniciais. O modelo ESSSDM auxilia a *startup* na validação das ideias iniciais do negócio para que este processo de encontrar a ideia chave para a *startup* seja rápido e preciso [BHOBL13]. O modelo possui 3 partes: (1) geração de ideias,



(2) um *backlog* de ideias priorizadas e (3) um funil por meio do qual as ideias são validadas sistematicamente usando o *loop Build-Measure-Learn* [BHOB13] [Rie09].

Giardino et. Al [GPU+15] realizou um estudo com 13 startups para entender o seu comportamento na adoção de atividades de Engenharia de Software. Como resultado, os autores descobriram que as *startups* são conduzidas com algumas características que definem as decisões adotadas na construção do negócio. Estas características são a incerteza, a falta de recursos e a pressão do tempo. Isto faz com que a startup, ao lançar o produto no mercado, foque na identificação dos ajustes necessários a serem feitos na solução proposta de acordo com o *feedback* inicial dos primeiros clientes ou usuários e das métricas coletadas [GPU+15]. No modelo apresentado por Giardino et. Al [GPU+15], chamado de *Greenfield Startup Model*, 4 objetivos relacionados ao desenvolvimento de software foram apresentados:

- Integração de soluções escaláveis com iterações rápidas e conjunto mínimo de funcionalidades;
- Empoderamento dos membros da equipe conferindo-lhes responsabilidade e autonomia para se envolverem em todas as atividades da fase de desenvolvimento;
- Melhoria dos padrões de fluxo de trabalho através do início de um gerenciamento de projeto mínimo; e
- Adaptação das práticas de desenvolvimento *Lean* e *Ágil* após a fase inicial caótica da *startup*.

Um dos pontos que são comumente mostrados em pesquisas que envolvem o desenvolvimento de software em *startup* é a adoção de práticas ágeis [GPU+15] [MCSP20] [WEB+16] [PMK+17] [MHASH19] [SRSM19] [YM13].

Yau [YM13] apresentou um estudo relacionado à adoção rigorosa de metodologia ágil pelas *startups*. A principal contribuição deste estudo foi que este uso rigoroso e puro de uma metodologia ágil pode não fornecer benefícios para compensar o custo e o tempo investidos pela equipe de desenvolvimento. Em outro estudo ministrado por Mkpojiogu [MHASH19] um *survey* realizado entre *startups* da Arábia Saudita indicou que 72.4% dos respondentes indicaram utilizar ágil em alguma das práticas de desenvolvimento indicando ainda que as 5 principais metodologias ágeis adotadas são:

- Metodologia Scrum;
- *Dynamic Software Development Methodology*;
- Metodologia *Lean*;
- *eXtreme Programming*; e

- *Feature Driven Development.*

Souza et. Al [SRSM19] realizou uma coleta de dados em 14 *startups* com o objetivo de investigar as práticas ágeis em uso pelas *startups*. Um dos primeiros resultados indica que o uso de tais práticas ajudam a integrar, otimizar e automatizar o desenvolvimento de software [SRSM19].

Baseado nas respostas de 1526 *startups* de software, Pantiuchina et. al [PMK<sup>+</sup>17] observou que a maioria das *startups* não utiliza as práticas ágeis para melhorar a qualidade do seus produtos ou serviços, ou seja, questões relacionada a refatoração regular e testes tem baixa prioridade e, conseqüentemente, há um aumento da dívida técnica. Por outro lado, a adoção com o viés da velocidade no desenvolvimento é o grande foco das *startups* para que possam atender aos lançamentos frequentes das versões dos seus produtos e o planejamento ágil a curto prazo.

## 2.2 Processo de desenvolvimento de software

O desenvolvimento de um software, desde os estágios iniciais até a manutenção do sistema, envolve especificar, desenvolver, gerenciar e evoluir sistemas de software [Som10]. Isto é feito através da adoção de práticas de Engenharia de Software (ES). O termo ES surgiu em 1968 com o objetivo de dar suporte ao desenvolvimento utilizando processos, métodos, técnicas e ferramentas [Som10]. Nas startups a realidade não é diferente.

O uso sistemático usado na Engenharia de Software é às vezes chamado de processo de software [Som10]. Segundo Sommerville [Som10] um processo de software é uma seqüência de atividades que leva à produção de um produto de software utilizando quatro atividades fundamentais:

1. Especificação de software, onde clientes e engenheiros definem o software a ser produzido e as restrições à sua operação.
2. Desenvolvimento de software, onde o software é projetado e programado.
3. Validação de software, onde o software é verificado para garantir que é o que o cliente exige.
4. Evolução do software, em que o software é modificado para refletir as mudanças nos requisitos do cliente e do mercado.

Diferentes modelos de processo foram propostos e são utilizados por organizações de diferentes tamanhos [KS13]. Estes modelos são aplicados por estas organizações para se ter um maior controle dos custos, tempo, qualidade e mudanças dos requisitos dos

clientes [KS13]. De acordo com Prikladnicki et al. [PNAE03], a qualidade do software é fortemente dependente da qualidade do processo utilizado na sua preparação. Da mesma forma, uma reflexão feita por Fuggeta [Fug00], esclarece que pesquisadores e praticantes têm percebido que o desenvolvimento de software é um esforço coletivo, complexo e criativo. Desse modo, a qualidade do produto de software depende bastante das pessoas, organizações e procedimentos utilizados para criá-lo e disponibilizá-lo [Fug00].

As próximas seções apresentam uma visão geral das principais abordagens de desenvolvimento de software.

### 2.2.1 Abordagem Prescritiva

A abordagem prescritiva é o processo mais tradicional de desenvolvimento de software que envolve um processo de projeto sequencial [Pre15]. Um dos modelos prescritivos mais conhecidos é o modelo cascata e as fases que estão envolvidas neste são: concepção, projeto do sistema, implementação, integração e teste, implantação do sistema e manutenção.

Alguns aspectos importantes do modelo cascata que devem ser levados em consideração na sua escolha são [BM12]:

- O requisito deve estar claro e bem definido antes de iniciar a fase de projeto.
- O teste só será realizado quando o código estiver totalmente desenvolvido.
- Cada fase de desenvolvimento prossegue em ordem sem qualquer sobreposição.
- A documentação e os testes acontecem no final de cada fase, o que ajuda a manter a qualidade do projeto.
- O testador estará envolvido apenas na fase de testes.

A concepção é a primeira fase da abordagem em cascata que envolve a compreensão do que é necessário para ser projetado. Essa fase também compreende o entendimento da função e finalidade do software. Na fase de concepção, as especificações da entrada e a saída do produto final são marcadas e estudadas de maneira adequada [Pre15].

No estágio de projeto, as especificações do primeiro estágio são estudadas adequadamente e o projeto do software é desenhado. Há benefícios para o projeto do sistema no modelo em cascata para desenvolvimento de software. O primeiro benefício é a especificação do hardware e do requisito do sistema. Outro benefício é a definição da arquitetura geral do sistema [Pre15].

Na fase de implementação, o sistema é desenvolvido em unidades que são integradas na próxima fase do desenvolvimento [Pre15]. Nessa fase, a preocupação é que

cada unidade esteja funcionalmente criada e testada, para posterior integração e testes de sistema.

Todas as unidades do estágio de implementação são integradas ao sistema. Essas unidades devem ser devidamente testadas antes da integração. O objetivo desses vários testes do software é descobrir se há algum erro, e outra finalidade dele é garantir que o cliente não enfrente qualquer dificuldade na instalação do software [Pre15].

## 2.2.2 Abordagem Adaptativa

Jim Highsmith [Hig13] apresenta uma abordagem adaptativa para o desenvolvimento de software para a construção de software e sistemas altamente complexos. Esse modelo, baseado na colaboração e auto-organização das equipes, define um ciclo de vida para o modelo baseando-se em três fases: especulação, colaboração e aprendizagem [Hig13]. Os métodos ágeis são exemplos desta abordagem adaptativa.

Metodologias ágeis são um grupo de métodos de desenvolvimento de software que são baseados em um processo iterativo e incremental [SSG12]. As quatro principais características fundamentais para todas as metodologias ágeis são: planejamento, desenvolvimento iterativo e evolutivo, rapidez e resposta flexível para mudar e promover a comunicação [Mah09].

O Manifesto Ágil define a abordagem do desenvolvimento de software ágil [All01]. Nesse manifesto, há a indicação dos 12 princípios por trás da abordagem ágil [All01]:

- Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas disso.
- Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
- Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
- Construir projetos ao redor de indivíduos motivados, dando-lhes o ambiente e suporte necessário, e confiar que farão seu trabalho.
- O método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
- Software funcional é a medida primária de progresso.

- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter incessantemente, passos constantes.
- Contínua atenção à excelência técnica e ao bom design, aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
- As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
- Em intervalos regulares, o time reflete em como ficar mais efetivo, e então, se ajustam e otimizam seu comportamento de acordo.

### 3. METODOLOGIA DE PESQUISA

Para a condução da pesquisa proposta neste documento, um estudo empírico do tipo exploratório envolvendo métodos qualitativos e quantitativos de coleta e análise de dados, vinculado a experimentação e análise em campo, se faz necessário. De uma forma geral, a metodologia foi inspirada nos estudos de [MBT06] e [SCT01], e é dividida em quatro fases. Esta metodologia é detalhada neste Capítulo.

#### 3.1 Métodos utilizados

##### 3.1.1 Revisão sistemática da literatura

Mapeamento da literatura fornece *insights* sobre as maneiras pelas quais o pesquisador pode limitar o escopo a uma área necessária de investigação [CC17]. Keele [K<sup>+</sup>07] define uma revisão sistemática da literatura como "Uma ampla revisão de estudos primários em uma área de tópico específica que visa identificar quais evidências estão disponíveis sobre o tópico."

Segundo Keele [K<sup>+</sup>07], existem três motivos principais para a realização de tal estudo:

- Identificação de grupos de evidências.
- Direcionar o foco de futuras revisões sistemáticas.
- Identificação de áreas para mais estudos primários a serem realizados.

No contexto desta pesquisa, foi utilizada uma revisão sistemática da literatura do tipo mapeamento sistemático. O tema deste mapeamento foi o processo de desenvolvimento de software utilizado por *startups* digitais na construção de seus negócios (envolvendo técnicas, práticas, metodologias, etc). Para realizar este estudo de mapeamento, seguiu-se o protocolo proposto por Petersen et al. [PFMM08], cujos principais estágios são mostrados na Figura 3.1.

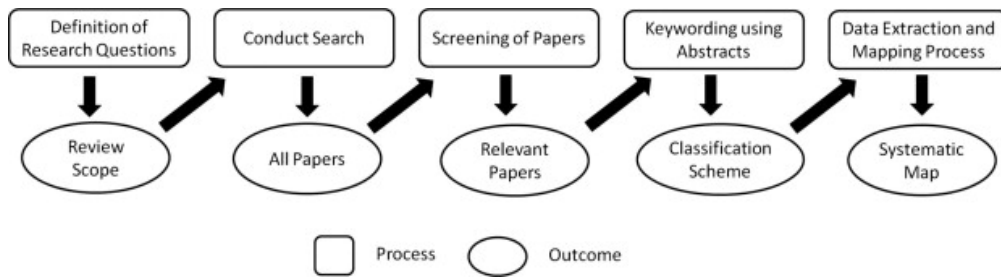


Figura 3.1 – Estágios do mapeamento sistemático [PFMM08]

Este mapeamento foi realizado no início do processo desta pesquisa e complementada ao final do período do doutorado, com o intuito de:

- Entender o estado da arte referente ao uso de práticas de Engenharia de Software pelas *startups*.
- Identificar lacunas nas pesquisas existentes que suportassem as decisões para futuras investigações.
- Fornecer um embasamento teórico sobre o tema para o pesquisador.

### 3.1.2 Estudo de campo

Um estudo de campo é um método de investigação com o foco em uma comunidade de trabalho, de estudo, ou voltada para qualquer outra atividade humana [Car02]. No estudo de campo a pesquisa é realizada por observação direta das atividades e/ou por entrevistas com participantes da comunidade, com o intuito de captar informações e interpretações sobre o grupo estudado.

Kvale [Kva83] define o estudo de campo como uma pesquisa qualitativa como algo que tem o objetivo de reunir descrições do mundo da vida do entrevistado no que diz respeito à interpretação do significado dos fenômenos descritos. Nesta pesquisa utilizou-se a entrevista com *startups* para a coleta de dados. Entrevista é um método básico usado na pesquisa qualitativa [Olt16].

Nas entrevistas face-to-face não há atraso significativo entre a pergunta e a resposta [Opd06]: o entrevistador e o entrevistado podem reagir diretamente sobre o que o outro diz ou faz. Uma vantagem dessa comunicação síncrona é que a resposta do entrevistado é mais espontânea, sem reflexão prolongada [Opd06].

Nesta pesquisa, optou-se por uma entrevista semiestruturada pois oferece uma amplitude maior do que as demais, dada sua natureza qualitativa. A entrevista semiestruturada é um encontro em que o entrevistador não segue à risca uma lista formalizada de perguntas [BW94]. O protocolo criado serve como guia para a condução dos questionamentos, porém permite uma discussão maior em determinados tópicos.

Como resultado, nesta pesquisa, as entrevistas realizadas serviram para direcionar o pensamento sobre como é o comportamento das *startups* no que diz respeito ao uso de Engenharia de Software nas diferentes etapas de desenvolvimento do negócio e com isso possibilitar a criação de uma proposta de modelo para futuras avaliações e validações.

### 3.1.3 Survey

Um *survey* é um método estruturado de coleta de informações de uma parcela de uma população devidamente definida [Sch04]. Este tipo de técnica pode ser definida como a coleta de informações de uma amostra de indivíduos por meio de suas respostas às perguntas [CS11]. Este tipo de pesquisa permite uma variedade de métodos para recrutar participantes, coletar dados e utilizar vários métodos de instrumentação. Um *survey* pode usar estratégias de pesquisa quantitativas (por exemplo, usando questionários com itens classificados numericamente), estratégias de pesquisa qualitativa (por exemplo, usando perguntas abertas) ou ambas as estratégias (ou seja, métodos mistos) [Sch04].

Baseado nas questões de pesquisa deste trabalho, o *survey* foi utilizado para mapear com maior detalhamento e abrangência as práticas de Engenharia de Software utilizado pelas *startups* e o questionário foi criado baseado nos eixos levantados na análise das entrevistas realizada anteriormente. Com isso, o objetivo da aplicação do *survey* foi o de validar tais eixos e ajustar o modelo proposto desta pesquisa.

### 3.1.4 Grupo Focal

Um grupo focal é uma discussão informal entre indivíduos selecionados sobre tópicos específicos [BTS86]. Apesar de existirem variações sobre o método básico [Kit94] há uma ou mais discussões, nas quais os participantes se concentram em tópicos selecionados pelo pesquisador que apresenta estes em formatos de perguntas. Segundo Carey [CS94], recomenda-se grupos entre 6 e 8 participantes e agrupamentos homogêneos, particularmente em relação ao "status" ou "maturidade".

Kontio et al. [KLB04] fornecem diretrizes sobre as etapas para a realização de estudos de grupo de foco em um contexto de Engenharia de Software:

- **Passo 1 - Definir o problema de pesquisa:** O objetivo é discutir a utilização de práticas de Engenharia de Software pelas *startups* nos diferentes estágios de maturidade.
- **Passo 2 - Selecionar os participantes:** Foram escolhidas *startups* que estivessem em níveis mais avançados de maturidade e que possuíssem software como ferramenta chave para o negócio.



- **Passo 3 - Conduzir a sessão do grupo focal:** Assim que os participantes foram identificados e convidados, foi conduzida a sessão do grupo focal. Ela ocorreu de forma online utilizando a ferramenta zoom ([www.zoom.us](http://www.zoom.us))

Assim, o grupo focal realizado foi uma pesquisa investigativa, buscando preencher lacunas da etapa anterior. Executado de forma online síncrono requerendo que todos os participantes estivessem conectados ao mesmo tempo [Fli13]. As vantagens do grupo focal online são [Fli13]: maior grau de anonimato para participantes e que a dinâmica de grupo desempenha um papel menor em comparação com os grupos focais no mundo real, sendo que é menos provável que os participantes dominem o grupo ou inibam outra contribuição dos participantes.

### 3.2 Metodologia

A Figura 3.2 apresenta o desenho de pesquisa que sumariza as estratégias de pesquisa e a linha do tempo para a execução deste estudo. O desenho foi inspirado no trabalho de Shull et al [SCT01]. Cada fase é descrita em detalhes a seguir.

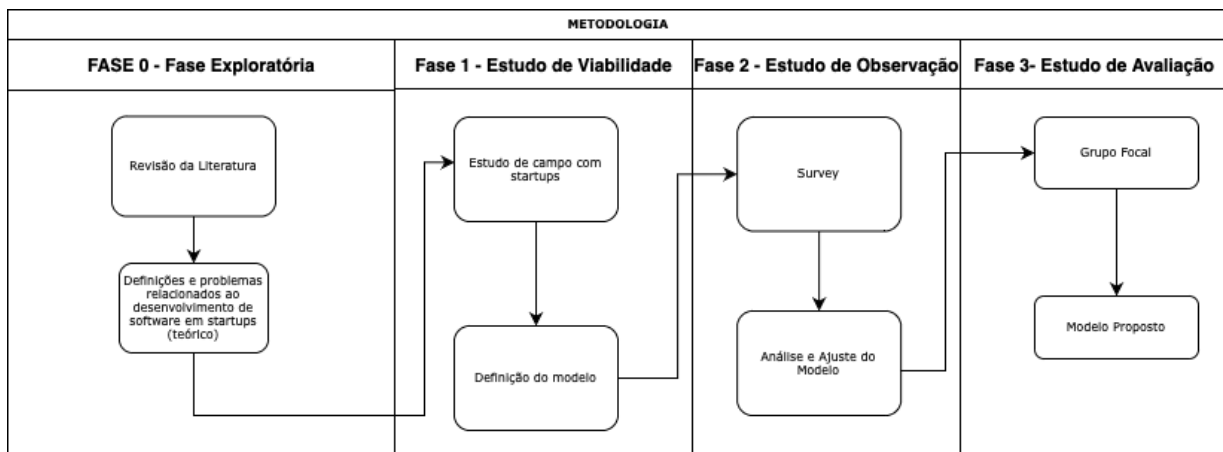


Figura 3.2 – Desenho da metodologia de pesquisa

### 3.3 Fase 0 - Exploratória

Esta fase consiste no entendimento do estado da arte e entender como as *startups* estão trabalhando e empregando a Engenharia de Software na construção dos seus produtos. Uma revisão da literatura foi realizada com o objetivo de analisar o que está sendo estudado e pesquisado nesta área.

### 3.4 Fase I - Estudo de Viabilidade

Estudos de viabilidade são peças de pesquisa feitas antes de um estudo principal [ACCL10]. Nesse contexto, ter o entendimento de como as *startups* de software desenvolvem seus MVPs (da concepção da ideia até as primeiras versões do produto) e que práticas e/ou ferramentas elas utilizam é crucial para esta pesquisa. Para isto, um estudo de campo foi realizado com dez empreendedores de *startups* de software. Foram escolhidas *startups* de software que já possuíssem produtos com clientes e/ou usuários, caracterizando-se como *startups* com o MVP validado pelo mercado.

Para a análise das entrevistas 6 passos foram seguidos:

1. Definição das perguntas e perfil dos entrevistados.
2. Entrevistas.
3. Transcrição das entrevistas.
4. Análise das transcrições e codificação das principais declarações.
5. Categorização da codificação realizada.
6. Análise com base nas categorias definidas.

O roteiro utilizado neste estudo de campo encontra-se no Apêndice A.

### 3.5 Fase II - Estudo de Observação

A terceira fase da metodologia é a execução de um estudo de observação, que tem por objetivo coletar dados sobre o tema e obter um entendimento minucioso dos dados coletados [SCT01]. Nesta etapa, um *survey online* com perguntas fechadas foi aplicado com *startups* de diferentes regiões do país. A análise deste *survey* encontra-se no Capítulo 4 e o instrumento de coleta do *survey* encontra-se no Apêndice B.

### 3.6 Fase III - Estudo de Avaliação

Execução de um grupo focal para entender de forma mais minuciosa o comportamento histórico das *startups* durante todos os estágios de maturidade. A escolha das *startups* com níveis de maturidade avançado é essencial para o sucesso desta etapa, visto

que, as discussões sobre as práticas escolhidas pelas equipes deste do primeiro dia da *startup* até o dia do grupo focal são importantes para a análise e avaliação do modelo. Foram utilizadas duas técnicas para ajudar nesta avaliação: classificação de cartões (os cartões encontra-se no Apêndice C) e Grupo Focal (o roteiro utilizado encontra-se no Apêndice D). O detalhamento e análise da aplicação destes métodos encontra-se no Capítulo 5.

## 4. FASE 0 - EXPLORATÓRIA - REVISÃO SISTEMÁTICA DA LITERATURA

A revisão da literatura teve como objetivo a identificação de estudos existentes e já publicados no tema desta pesquisa. Um dos principais métodos utilizados para apoiar na identificação destes estudos é a revisão sistemática da literatura. Ela é caracterizada como um estudo secundário usado para buscar na literatura existente uma gama de diferentes estudos a partir de questões de pesquisa específicas [Web03]. Essas revisões empregam protocolos cuidadosamente definidos para determinar quais estudos devem ser incluídos, bem como para analisar sua contribuição da forma mais imparcial possível.

### 4.1 Justificativa e Trabalhos Relacionados

Ao analisar o processo de desenvolvimento de software, encontram-se várias atividades ou práticas consideradas essenciais [FS17]. Estas práticas permitem que os desenvolvedores consigam atender as necessidades dos *stakeholders*, mantendo assim um fluxo contínuo de trabalho [dTMP19]. A utilização de práticas de desenvolvimento de software em *startups* é algo desafiador para os membros da equipe [CO08a]. Segundo Fuggetta [Fug00], o gerenciamento do desenvolvimento de software é alcançado por meio do uso de processos de software, sendo que isto é definido pelo conjunto de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos necessários para conceber, desenvolver, implantar e manter um produto de software. Neste contexto, utilizar práticas de Engenharia de Software em *startups* é algo complexo e multifacetado, mesmo as *startups* sendo criativas e flexíveis por natureza [Sut00].

Outras revisões da literatura foram realizadas sobre este mesmo tema. Paternoster et al. [PGU<sup>+</sup>14] desenvolveu uma revisão com o objetivo principal de entender o estado da arte da pesquisa em *startups* de software. Como conclusão, o estudo mostrou que em 2013 somente 4 artigos tinham fortes contribuições relacionadas ao uso de Engenharia de Software no contexto das *startups* [PGU<sup>+</sup>14].

Em 2015, Klotins et al. [KUG15] realizou um mapeamento da literatura, classificando os estudos primários em 11 das 15 áreas do conhecimento do SWEBOK, concluindo que a revisão não fornecia um suporte confiável para a Engenharia de Software em qualquer fase do ciclo de vida de uma *startup*.

## 4.2 Protocolo da Revisão Sistemática da Literatura

Para a execução desta revisão sistemática da literatura, as seguintes questões de pesquisa foram utilizadas como guia:

- RQ #1: Como as práticas de Engenharia de Software são utilizadas por uma *startup* de software durante o processo de desenvolvimento do MVP?
- RQ #2: Quais ferramentas são usadas pelas *startups* de software durante o processo de desenvolvimento do MVP?

As bases de dados científicas selecionadas nas quais foram realizadas as buscas são apresentadas a seguir. Estas bases de dados foram escolhidas considerando a sua cobertura e uso no domínio da Engenharia de Software e *startups*, além da capacidade de executar a consulta referente a este estudo. As seguintes bases de dados foram utilizadas:

- ACM Digital Library
- IEEExplore
- Scopus
- Science Direct

Para a construção da *string* de busca, as seguintes palavras-chave foram consideradas: Engenharia de Software, desenvolvimento de software, *startup* de software, *startup*, processo de desenvolvimento de software, prática de desenvolvimento de software, metodologia de Engenharia de Software, técnica, método, modelo, estratégia, abordagem, metodologia, *framework* e ferramenta. E com isto, a *string* de busca ficou da seguinte forma:

Population: P:= (Software startup <or> Startup)

Intervention: I := (( Software Engineering <or> "Software Development") <or> "Software development process«and» (MVP <or> "minimum viable product«or» prototype) )

Outcome: O := (Technique <or> Method <or> Model <or> Strategy <or> Approach <or> Methodology <or> Framework <or> Tool )

E com o intuito de entender melhor a área de *startups* de software e o uso de práticas de Engenharia de Software por estas empresas nascentes e considerando uma revisão da literatura sobre o desenvolvimento de software por *startups* realizado por Paternoster et. al. [PGU<sup>+</sup>14], que analisou estudos publicados até 2013 e do estudo apresentado por Klotins [KUG15] referente a revisão da literatura com o foco nas práticas de Engenharia de Software em v, publicado no ano de 2015. Inicialmente a revisão da literatura utilizou o

período de 2015 à 2018 (período inicial desta pesquisa) e refeita utilizando o período escolhido para este estudo foi o de 2015 a 2020 para melhor representar o estado da arte do tema utilizado neste trabalho. Além disso, para cada uma das bases de dados utilizadas, ajustes na *string* de busca foram necessárias, devido às particularidades de cada *engine* de busca. Na sequência são apresentadas como ficaram cada uma das strings de buscas de acordo com a base de dados:

- **Scopus** : ( TITLE-ABS-KEY ( ( "Software startup"OR startup ) ) AND TITLE-ABS-KEY ( ( ( "Software Engineering"OR "Software Development") OR "Software Development Process"AND ( mvp OR "Minimum Viable Product"OR prototype ) ) ) AND TITLE-ABS-KEY ( ( technique OR method OR model OR strategy OR approach OR methodology OR framework OR tool ) ) ) AND PUBYEAR > 2014 AND PUBYEAR < 2021
- **Science Direct** : ( ( "Software startup"OR startup ) ) AND ( ( "Software Engineering"OR "Software Development") OR "Software Development Process"AND ( mvp OR "Minimum Viable Product"OR prototype ) ) )
- **IEEEXplore** : ("Software startup"OR Startup) AND ((( "Software Engineering"OR "Software Development") OR "Software development process"AND (MVP OR "minimum viable product"OR prototype) )
- **ACM** : "query": AllField:(Startup) AND AllField:(( "Software Engineering"OR "Software Development") OR "Software Development Process"AND ( mvp OR "Minimum Viable Product"OR prototype ) ) ) "filter": Publication Date: (01/01/2015 TO 12/31/2020), NOT VirtualContent: true

O processo de seleção dos estudos iniciou com o resultado da aplicação destas *string* de buscas, conforme apresentado na Tabela 4.1. A pesquisa nas bases de dados Scopus, Direct Science, IEEE e ACM usando a *string* de busca resultou em um total de 1.175 referências, sendo 737 estudos não duplicados. As buscas foram direcionadas aos seguintes tipos de documentos: capítulos de livros, artigos de periódicos e artigos de conferências.

Tabela 4.1 – Resultados das pesquisas nas bases de dados

Base de dados	#Artigos
Scopus	26
Science Direct	44
IEEE	119
ACM	986
<b>Total</b>	<b>1175</b>

Os critérios de seleção e exclusão foram definidos e utilizados para refinar a escolha dos estudos desta revisão e estes são apresentados a seguir:

- Não serão selecionadas publicações em que descrevam e/ou apresentam ‘keynote speeches’, cursos, workshops e similares;
- Somente serão selecionados trabalhos escritos em inglês;
- Somente serão selecionados trabalhos publicados entre 2015 e 2020;
- Não serão selecionados trabalhos que não passaram por um processo de revisão por pares;
- Serão excluídos trabalhos referentes a áreas do conhecimento que não seja Engenharia de Software;

O primeiro passo foi verificar trabalhos duplicados entre os 1.175 resultados das bases de busca utilizadas. Na etapa seguinte, foi realizada uma análise nos títulos e resumos, buscando as palavras-chave "Engenharia de Software" e "Startup". A partir do resultado desse filtro, todos os trabalhos resultantes foram lidos para identificar a relevância desses estudos, e assim os 37 artigos considerados no mapeamento foram filtrados. O processo e o número de referências de cada etapa do processo são apresentados na Figura 4.1.

Este processo de análise dos trabalhos resultantes das pesquisas realizadas nas bases de dados foi realizada utilizando uma ferramenta chamada Rayyan (<https://rayyan.qcri.org/>) que permite, de forma ágil, importar as referências dos trabalhos e realizar uma triagem dos mesmos, com o objetivo de facilitar o trabalho do revisor [OHFE16]. Além desta ferramenta, uma planilha em excel foi utilizada para finalizar a análise dos resultados.

Na fase de definição das palavras-chaves ("*keywording*"), o objetivo é criar um esquema de classificação de forma eficiente, garantindo que todos os documentos relevantes sejam levados em consideração [PFMM08]. O processo utilizado é apresentado na Figura 4.2.

Iniciando pela leitura dos resumos dos estudos primários e atribuindo-lhes um conjunto de palavras-chave para identificar a principal área de contribuição do artigo. A seguir, é realizada uma combinação das palavras-chave, categorizando em alto nível, refinando o esquema. Para esta atividade uma planilha excel foi utilizada para facilitar a inserção das palavras-chave, bem como informações adicionais relativas a extração de dados e mapeamento.

Para a extração de dados e mapeamento, algumas informações adicionais foram adicionadas à ferramenta citada anteriormente. São elas:

- Focus Facet: as categorias criadas durante o processo do esquema de classificação;

1. **Equipe de desenvolvimento**

2. **Gerenciamento de projetos**

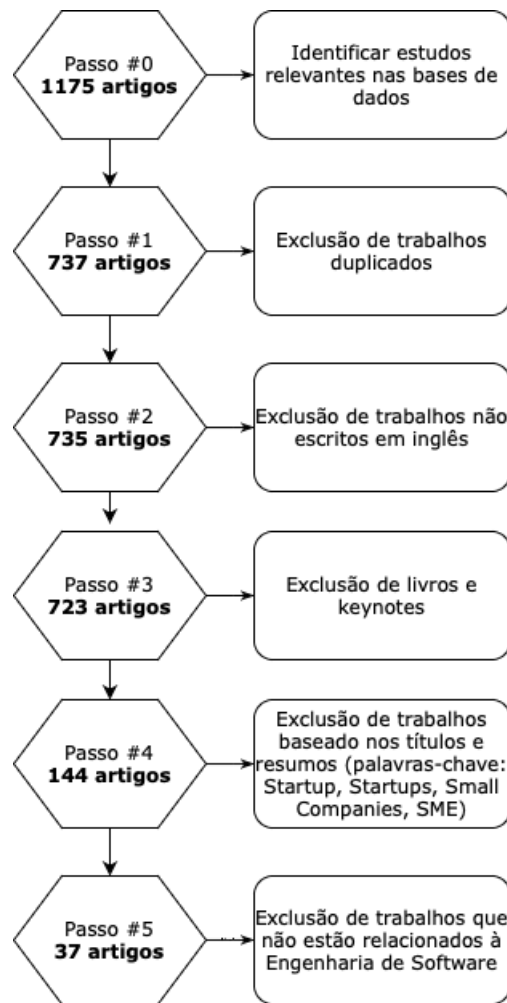


Figura 4.1 – Estágios do processo de seleção

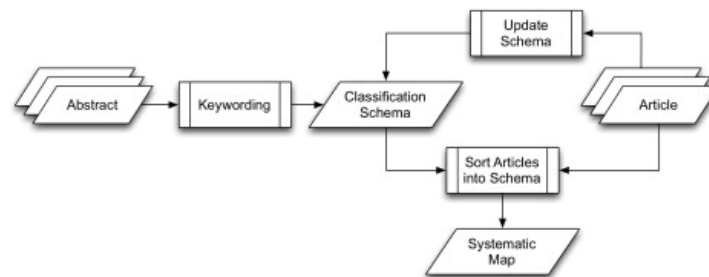


Figura 4.2 – Criação do esquema de classificação

### 3. Processo de desenvolvimento

### 4. Métodos ágeis

### 5. Requisitos

### 6. Dívida Técnica

- Tipo de contribuição: Baseado em um trabalho de Shaw [Sha03];

1. **Conselhos/recomendações:** implicação com base em opiniões pessoais.

2. **Framework/Método:** Framework/método usado no desenvolvimento de software.



3. **Orientações:** Orientações com base nos resultados da pesquisa.
  4. **Lições aprendidas:** Conselhos acionáveis derivados dos resultados de pesquisa obtidos.
  5. **Modelo:** Representação do modelo de um determinado contexto com base em um processo conceituado.
  6. **Ferramentas:** Ferramentas utilizadas durante o desenvolvimento de software.
- Método de pesquisa: método utilizado na pesquisa;
    1. **Estudo de caso:** Um relatório sobre uma situação específica que foi estudada;
    2. **Estudo empírico:** Estudo baseado em evidências empíricas;
    3. **Estudo experimental:** Um estudo em que uma intervenção é introduzida para observar seus efeitos;
    4. **Survey:** Processo para coletar dados, analisar as informações e relatar os resultados
  - Tipo de pesquisa: tipo de pesquisa (adaptado de Wieringa et al. [WMMR06]): A classificação utilizada por Wieringa [WMMR06] foi utilizada como base para definir o tipo de pesquisa dos artigos analisados. Esta classificação é definida como:
    1. **Evaluation Research (ER):** técnicas ou soluções são implementadas e avaliadas na prática, e as consequências investigadas;
    2. **Validation Research (VR):** técnicas que foram propostas, mas ainda não foram executadas na prática (é um estudo típico de laboratório);
    3. **Proposal of Solution (PS):** A solução para um problema é proposta e seus benefícios são discutidos.
    4. **Philosophical Paper (PP):** estrutura a área na forma de uma taxonomia ou quadro conceitual;
    5. **Experience Paper (PE):** inclui a experiência pessoal do autor na percepção de como aconteceu na prática;
    6. **Opinion Paper (OP):** a opinião pessoal do autor sobre um problema sem trabalho relacionado e métodos de pesquisa.
  - Paper Rank: nota de 0 a 10, sendo 0 baseada no trabalho de Dyba et al. [DD08]
    1. **Referências:** O estudo está bem referenciado? (1 ponto)
    2. **Objetivos:** Os objetivos dos estudos estão claros? (1 ponto)
    3. **Amostra:** A coleta de dados foi realizada adequadamente? (1 ponto)
    4. **Método:** A metodologia de análise foi bem aplicada? (1 ponto)

5. **Contexto:** O contexto do estudo está claramente descrito? (1 ponto)
  6. **Descobertas:** As descobertas são confiáveis? (1 ponto)
  7. **RQ #1:** Responde a RQ #1? (2 pontos)
  8. **RQ #2:** Responde a RQ #2? (2 pontos)
- Contribuição: conclusões importantes do estudo.

### 4.3 Análise dos Resultados

Esta seção apresenta os dados mais relevantes extraídos dos 37 estudos primários, após a análise e a classificação. A Tabela 4.3 apresenta a lista destes 37 artigos.

Na Figura 4.3, pode-se observar a distribuição das publicações de acordo com os anos de publicação. Este gráfico foi gerado somente com os dados recuperados dos anos definidos para esta revisão.

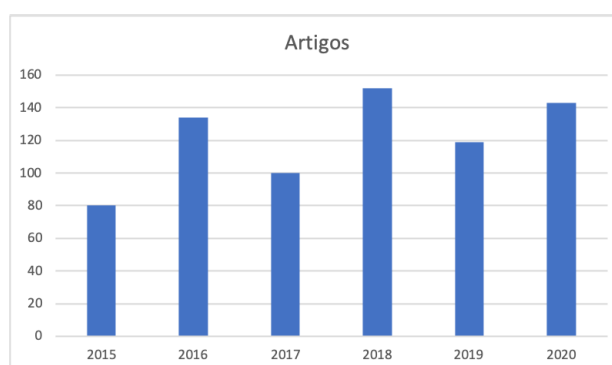


Figura 4.3 – Referências por ano de publicação

Se comparado com a primeira versão desta revisão da literatura (Figura 4.4), pode-se constatar que há um aumento das publicações referentes ao assunto deste trabalho, mostrando um crescimento significativo até 2016 e posteriormente um volume não muito diferente de um ano para o outro (como pode-se ver na Figura 4.3).

A fim de ilustrar lacunas potenciais no campo de pesquisa, são apresentados alguns gráficos para melhor entender o posicionamento dos grupos de pesquisa de acordo com a temática deste trabalho. Os trabalhos primários selecionados além de estarem relacionados à Engenharia de Software, possuem focos bem definidos e estes foram mapeados nesta revisão. Os focos são: Dívida Técnica, Equipe de Desenvolvimento, Metodologias Ágeis, Processo de Desenvolvimento e Requisitos. Na Figura 4.5, estes focos são cruzados com os tipos de pesquisa e os métodos de pesquisa. Fica claro que a maioria dos trabalhos são do tipo *Validation Research*, indicando resultados ainda iniciais e não validados. Além disso, boa parte destes trabalhos são estudos empíricos ou *surveys*, indicando que há ainda

Tabela 4.2 – Lista dos artigos primários

Primeiro autor	Focus Facet	Tipo de Pesquisa	Método de Pesquisa	Tipo de Contribuição	Paper Rank
Cavalcante, B H [CLBdFJ18]	Processo de desenvolvimento	VR	Survey	Orientações	9,6
Klotins, E [Klo17]	Processo de desenvolvimento	VR	Estudo de Caso	Framework/Método	9,6
Giardino, Carmine [GPU+15]	Processo de desenvolvimento	VR	Estudo Empírico	Orientações	9,6
Melegati, J [MCS+20]	Processo de desenvolvimento	VR	Estudo Empírico	Conselhos/Recomendações	9,4
Pantiuchina, J [PMK+17]	Métodos ágeis	VR	Survey	Orientações	9,4
Nguyen-Duc, Anh [NDSA15]	Processo de desenvolvimento	PS	Estudo Empírico	Modelo	9,3
Berg, V [BBND+18]	Processo de desenvolvimento	VR	Estudo Empírico	Conselhos/Recomendações	9,1
Souza, Renata [SMS+19]	Processo de desenvolvimento	VR	Survey	Modelo	9
Tripathi, N [TKP+18]	Requisitos	VR	Estudo Empírico	Conselhos/Recomendações	9
Duc, Anh Nguyen [DA17]	Equipe de desenvolvimento	VR	Estudo de Caso	Conselhos/Recomendações	8,9
Seyam, M [SM15]	Métodos ágeis	VR	Survey	Conselhos/Recomendações	8,8
Thongsukh, S [TA+17]	Métodos ágeis	ER	Estudo Empírico	Framework/Método	8,6
Melegati, Jorge [MGKW19]	Requisitos	ER	Estudo Empírico	Modelo	8,5
Njima, Mercy [Nji19]	Dívida Técnica	VR	Estudo Empírico	Orientações	8,4
Rafiq, U [RBWL17]	Requisitos	VR	Estudo de Caso	Modelo	8,3
Nguyen-Duc, A [NDDSA17]	Processo de desenvolvimento	VR	Estudo Empírico	Orientações	8,2
Nguyen-Duc, A [NDSA16]	Processo de desenvolvimento	ER	Survey	Modelo	8,2
Khanna, D [Kha18]	Equipe de desenvolvimento	ER	Estudo Empírico	Framework/Método	8,1
Gralha, C[GDW+18]	Requisitos	VR	Estudo de Caso	Orientações	8
Albuga, S [AO18]	Requisitos	ER	Estudo Empírico	Framework/Método	7,8
Yin, Huishi [YP17]	Requisitos	VR	Survey	Conselhos/Recomendações	7,7
Cico, Orges [CDJ20]	Processo de desenvolvimento	VR	Survey	Conselhos/Recomendações	7,5
Melegati, Jorge [Mel19]	Requisitos	VR	Estudo Empírico	Orientações	7,5
Chicote, M [Chi17]	Dívida Técnica	VR	Estudo Empírico	Conselhos/Recomendações	7,4
Seppanen, P [SOL16]	Equipe de desenvolvimento	VR	Survey	Orientações	7,3
Alves, C [ACA20]	Requisitos	ER	Estudo Empírico	Lições aprendidas	6,9
Melegati, Jorge [Mel18]	Processo de desenvolvimento	PS	Estudo Empírico	Framework/Método	6,8
Aldaej, Abdullah [Ald19]	Dívida Técnica	VR	Estudo de Caso	Conselhos/Recomendações	6,8
Njima, Mercy [ND19]	Dívida Técnica	VR	Estudo Empírico	Conselhos/Recomendações	6,8
Seyam, M	Métodos ágeis	VR	Estudo Empírico	Conselhos/Recomendações	6,7
Ibba, Simona [IBPT18]	Processo de desenvolvimento	VR	Survey	Conselhos/Recomendações	6,5
Lim, S L [LBI20]	Requisitos	ER	Estudo de Caso	Framework/Método	6,5
Besker, T [BML+18]	Dívida Técnica	VR	Survey	Conselhos/Recomendações	6,5
Mangiza, Phillip [MB20]	Equipe de desenvolvimento	VR	Estudo Empírico	Orientações	6,4
Klotins, Eriks [KUC+18]	Dívida Técnica	VR	Survey	Conselhos/Recomendações	6,3
Ko, A J [Ko17]	Processo de desenvolvimento	OP	Estudo de Caso	Lições aprendidas	5,8
Kemell, K -K. [KES+20]	Equipe de desenvolvimento	VR	Estudo de Caso	Conselhos/Recomendações	4,7

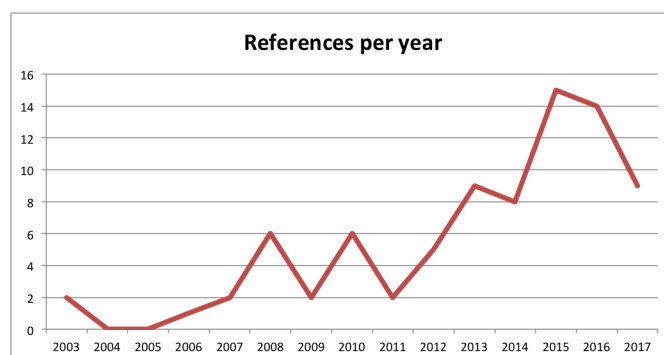


Figura 4.4 – Referências por ano

trabalho a ser realizado na pesquisa. Analisando os focos dos trabalhos, observa-se que

algumas práticas de Engenharia de Software ainda não figuram nos objetos de pesquisa, como por exemplo, verificação e validação.

Focus Facet	ER			ER Total	OP			OP Total	PS			PS Total	VR			VR Total	Total
	Estudo de Caso	Estudo Empírico	Survey		Estudo de Caso	Estudo Empírico	Survey		Estudo de Caso	Estudo Empírico	Survey						
Débito Técnico												1	3	2	6	6	
Equipe de desenvolvimento			1	1								2	1	1	4	5	
Métodos ágeis			1	1									1	2	3	4	
Processo de desenvolvimento			1	1	1	1	1	2	2			1	4	4	9	13	
Requisitos	1	3		4								2	2	1	5	9	
<b>Total</b>	<b>1</b>	<b>5</b>	<b>1</b>	<b>7</b>	<b>1</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>6</b>	<b>11</b>	<b>10</b>	<b>27</b>	<b>37</b>				

Figura 4.5 – Referências por ano

#### 4.3.1 Síntese do Capítulo

No estudo apresentado por Giardino et. al [GPU+15] um modelo definido com 7 categorias que direciona como as estratégias de desenvolvimento são planejadas e as práticas são utilizadas nas *startups*. Por outro lado, Nguyen-Duc [NDSA15] apresenta um estudo preliminar de uma visão alternativa da evolução das *startups* utilizando práticas como, *Customer Discovery* e prototipação. Entretanto, este modelo possui somente uma validação preliminar com 3 *startups*.

Alguns trabalhos relacionam o uso de práticas ágeis durante o ciclo de vida de uma *startup*. Seyam [SM15] explora a combinação de práticas ágeis (no caso *pair programming*) com UX Design, com o intuito de integrar na captura e desenvolvimento de requisitos, levando em consideração a experiência do usuário. Pantiuchina et.al. [PMK+17] apresenta um estudo que tem como objetivo entender o uso de práticas ágeis pelas *startups*. Estas práticas estão relacionadas a qualidade, velocidade e comunicação. Neste estudo, há a indicação que práticas relacionadas com velocidade (lançamento frequente e planejamento ágil) são mais utilizadas se comparadas com as demais.

Estudos relacionados a engenharia de requisitos aparecem com maior frequência nesta revisão. Lim et. al. [LBI20] apresenta um método para *startup* conseguirem acessar *stakeholders* de grandes corporações, com o intuito de entendimento de um determinado problema, e com isso mapear os requisitos da solução. Alves [ACA20] conclui que existe uma limitação de estudos empíricos sobre como as v de software executam a engenharia de requisitos. E Melegati et al. [MGKW19] apresenta um modelo de engenharia de requisitos para *startups* de software.

Outro aspecto apresentado nos estudos está relacionado ao dívida técnica. Klotins et. al. [KUC+18] realizou uma pesquisa com 86 v ao redor do mundo e verificou que as *startups* acumulam grande parte da dívida técnica na fase de testes e que o tamanho e a experiência da equipe implicam diretamente no controle ou não da dívida técnica. Destaca ainda que monitorar os níveis de dívida técnica e adoção de práticas preventivas dimi-

nui o risco do impacto negativo da dívida técnica no desenvolvimento da *startup*. Besker [BML<sup>+</sup>18] indica que a pressão que as *startups* sofrem para chegarem ao mercado o mais rápido possível e a incerteza relacionada as suas hipóteses são fatores importantes de geração de dívida técnica. No estudo realizado com *startups*, ele buscou entender o acúmulo intencional de dívida técnica, concluindo que nas fases iniciais a busca pelo "bom" orienta os empreendedores na quantidade de dívida que eles podem conter.

Pode-se analisar que a temática de processo de desenvolvimento, além de ter o maior número de trabalhos selecionados, apresentou uma maior importância para este trabalho. Isto está relacionado a forma mais ampla abordada pelos pesquisadores em relação às práticas de Engenharia de Software e ferramentas que auxiliam as *startups* de software a desenvolverem seus MVPs.

Nenhum artigo selecionado para análise tem como objetivo o estudo de ferramentas que podem ser utilizadas pelas *startups*. O foco de tais artigos se concentra em relatos de casos e pesquisas que envolvem o uso de práticas de Engenharia de Software pelas *startups*.

Neste Capítulo, aplicou-se a técnica de revisão da literatura com o intuito de entender o estado da arte de pesquisa na área de Engenharia de Software e *startups* de software. Nesta revisão 37 estudos primários foram selecionados e analisados sob a ótica das duas questões de pesquisa que guiaram este trabalho. Isso tudo permitiu que o direcionamento do trabalho deste doutorado fosse definido no sentido de entender e mapear as práticas de Engenharia de Software que as *startups* utilizam durante a criação dos seus MVPs e como isto evolui com o tempo.

## 5. FASE I - ESTUDO DE VIABILIDADE - ESTUDO DE CAMPO COM STARTUPS

Neste capítulo são apresentados os resultados do estudo de campo realizado com um conjunto de 10 startups. Foram selecionadas startups que já possuem um produto de software desenvolvido, ou seja, as *startups* escolhidas estão posicionadas nas fases de validação, tração ou escala, de acordo com o modelo de maturidade de *startups* definido pelo IBGC [LAM19] apresentado no Capítulo 2. O instrumento de coleta utilizado neste estudo de campo encontra-se no Apêndice A. A coleta de dados foi realizada usando entrevistas semiestruturadas cobrindo as 3 questões de pesquisa principais desta tese, apresentadas no Capítulo 1.

### 5.1 Protocolo do estudo

Para a realização das entrevistas semiestruturadas, uma amostra que pudesse representar diferentes tipos de *startups*, diferentes tamanhos de equipes e diferentes estágios de maturidade foi definida. Assim posto, 10 (dez) *startups* foram escolhidas para participarem desta amostra, sendo analisados os critérios da fase de maturidade segundo o IBGC [LAM19], se recebeu algum investimento, se há funcionários na *startup* além dos sócios e se reside em algum ambiente de inovação (parque tecnológico ou incubadora de empresas). A Tabela 5.1 apresenta estes critérios.

Tabela 5.1 – Startups x Critérios

Startup	Fase de maturidade (modelo do IBGC)	Investimento	Funcionários	Reside em ambiente de inovação
A	Tração	Não	Não	Sim
B	Validação	Sim	Não	Não
C	Validação	Não	Não	Não
D	Validação	Sim	Sim	Não
E	Tração	Não	Sim	Não
F	Validação	Não	Não	Sim
G	Escala	Sim	Sim	Não
H	Tração	Não	Não	Não
I	Escala	Sim	Sim	Sim
J	Tração	Não	Sim	Não

Para a coleta dos dados, o primeiro passo foi o de entrar em contato com um dos fundadores de cada *startup* que já possuíssem clientes e/ou usuários e fazer o convite para

participar das entrevistas. Para as *startups* residentes do Tecnopuc (três das dez) foi recebido um aceite de imediato, porém para evitar que houvesse um viés negativo na pesquisa, por ter na amostra somente *startups* conhecidas pelo autor desta pesquisa, *startups* de outros estados e outros ambientes foram convidadas. Os convites foram realizados por e-mail e à medida que os aceites eram recebidos, foram definidas a data e a hora da entrevista, utilizando o Whatsapp. Para a entrevista, a ferramenta *Zoom* foi utilizada, pois 7 (sete) *startups* entrevistadas eram estabelecidas em outras cidades.

Desta forma, para esta pesquisa, uma amostra não probabilística intencional foi utilizada. Isto é, são amostragens em que há uma escolha deliberada por parte do pesquisador, por conveniência, de acordo com seus critérios e julgamentos [MR14]. Este tipo de amostra define que a escolha seja realizada por especialistas [MR14], e neste caso as indicações das *startups* se deu por intermédio dos gestores dos ambientes de inovação ao qual as *startups* estão vinculadas.

#### 5.1.1 Caracterização das *startups*

##### **Startup A**

A *startup A* é uma *startup* da área do agronegócio e possui 3 (três) sócios, sendo que estes são os únicos que trabalham na operação. Possuem, cada um deles, papéis bem definidos, sendo que um é responsável pelo entendimento do negócio, relação com o mercado e vendas. Os outros dois sócios são da área de tecnologia da informação, sendo um deles programador *Web* e o outro um programador responsável pela arquitetura do software. A *startup A* possui 13 clientes ativos provenientes de diversos estados do Brasil. A solução é um SaaS (*Software as a Service*) para a gestão de animais. A entrevista foi realizada com o fundador CEO responsável pelo negócio e pela tecnologia escolhida para a solução.

##### **Startup B**

A *startup B* é uma *startup* da área financeira que faz a gestão de cripto ativos. Possui dois sócios, com perfil técnico (tecnologia da informação). O negócio está em fase de validação e possui 3 (três) clientes e um investidor-anjo. Somente um sócio está atuando em tempo integral na empresa, enquanto o outro sócio exerce atividade profissional em uma empresa de tecnologia. A entrevista foi realizada com fundador que possui o papel de CTO.

### **Startup C**

A *startup C* desenvolveu uma solução para melhorar a jornada do cliente em uma óptica. Possui um modelo B2B (*Business to Business*) [IDAB+19]. Possui 3 sócios, sendo que um deles é um sócio corporativo, isto é, uma empresa do ramo de ópticas possui participação societária na empresa e, muitas vezes, induz o que deve ou não ser desenvolvido pelos outros dois sócios. Um sócio possui o perfil técnico e o outro possui um perfil gerencial. A solução está em utilização por 5 clientes, sendo que ainda não há geração de receita. A entrevista foi realizada com o fundador desenvolvedor da solução.

### **Startup D**

A *startup D* é uma *startup* do ramo imobiliário, que disponibiliza uma ferramenta para as incorporadoras se relacionarem com os corretores de imóvel, promovendo maior detalhamento do processo de venda e reduzindo os custos da operação. A equipe é composta por 4 (quatro) profissionais, todos da área de tecnologia. Possui 2 (dois) clientes em fase inicial de uso. A entrevista foi realizada com o CEO, que foi o responsável pela definição inicial da tecnologia utilizada na solução da *startup*.

### **Startup E**

A *startup E* atua com tecnologias voltadas à transformação digital de processos. Possui uma plataforma que auxilia na gestão de processos digitais, sendo integrável a sistemas de empresas. Trabalha em um modelo B2B com uma equipe de 6 (seis) pessoas, sendo 2 (dois) sócios e 4 (quatro) funcionários da *startup*, sendo estes a equipe técnica de desenvolvimento. O CTO foi o fundador entrevistado nesta *startup*.

### **Startup F**

A *startup F* é uma empresa inovadora focada em saúde preventiva, que conta com uma solução completa para a estratégia da saúde da família. A entrevista foi realizada com o fundador e responsável pelo desenvolvimento da solução.

### **Startup G**

A *startup G* possui uma solução que facilita as integrações entre diferentes sistemas. Possui 6 (seis) profissionais na equipe e mais de 20 (vinte) clientes ativos. Trabalha no modelo B2B [IDAB+19], focando nas equipes de tecnologia da informação das empresas. Entrevista realizada com o CTO da *startup*.



### **Startup H**

A *startup* H desenvolveu um sistema para organizadores de competições esportivas. Modelo SaaS que permite ao organizador ter um site para divulgar e administrar suas competições. Possui 3 sócios, sendo dois da área de tecnologia da informação e um da área de gerenciamento. A entrevista foi realizada com o fundador responsável pela prospecção do negócio.

### **Startup I**

A *startup* I é uma *startup* da área financeira de antecipação de recebíveis para empresas que precisam de capital. Possui uma equipe de 13 (treze) profissionais, distribuídos em diversas áreas. Possui operação em 3 (três) estados brasileiros e trabalha com parte da equipe de forma distribuída. A entrevista foi conduzida com CEO da empresa.

### **Startup J**

A *startup* J é uma *startup* da área da saúde, focada na capacitação dos médicos através de uma plataforma de ensino à distância. Possui um modelo de assinatura B2C (*Business to Consumer*) [IDAB<sup>+</sup>19] e está desenvolvendo uma solução B2B2C (*Business to Business to Consumer*) [IDAB<sup>+</sup>19] para atender um novo mercado. A equipe possui 7 profissionais, sendo um sócio, um atuando na área de vendas e marketing e 5 (cinco) atuando na equipe de desenvolvimento. A entrevista foi realizada com o fundador que possui a função de CTO na *startup*.

## **5.2 Resultados**

Para a realização das entrevistas, os seguintes passos foram realizados:

- Elaboração das perguntas (Apêndice A).
- Revisão por pares : as perguntas foram revisadas por um colega, aluno de doutorado, e por uma doutora em Engenharia de Software, especialista em Engenharia de Software empírica.
- Pré-teste: foi realizada uma entrevista no modo pré-teste com uma *startup* residente do Tecnopuc.
- Entrevistas

Com as entrevistas realizadas, um processo de análise e codificação foi feito. Em uma primeira análise de cada uma das entrevistas, foram identificados em média 24 (vinte e quatro) códigos por entrevista, totalizando 247 códigos. Após esta primeira análise, um refinamento foi realizado, chegando a um total de 19 códigos que, classificados e agrupados, transformaram-se em um total de 6 categorias (Figura 5.1), mapeando assim o que mais se destacou nas falas dos empreendedores.

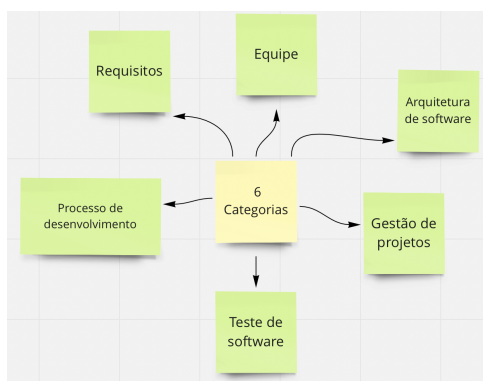


Figura 5.1 – Categorias resultantes das entrevistas (Fonte: Autor)

São elas: equipe, gerenciamento de projetos, arquitetura de software, teste de software, requisitos e processo de desenvolvimento. E no contexto de cada uma destas categorias, alguns pontos foram mais destacados.

Na categoria **Equipe**, há considerações relativas à competência em inovação, negócios e tecnologia. Ao criar uma nova *startup*, um conjunto completo de habilidades é necessário para iniciar este novo empreendimento e uma vasta quantidade de conhecimento relacionado a software é necessária nas fases de maturidade da *startup* [Kha18]. Além disso, a heterogeneidade da equipe muda à medida que evolui pelas diferentes etapas do processo [VCW<sup>+</sup>06] e isto ajuda a *startup* no enfrentamento dos problemas que ela encontra durante o seu desenvolvimento, principalmente pela combinação das experiências e conhecimentos que cada integrante da equipe possui. Conforme dito pelo empreendedor da *Startup A*:

*...cada sócio tem uma habilidade específica que nos ajuda como negócio...*

Planejar e desenvolver um novo produto é uma atividade densa e, normalmente, necessita ser executada por uma equipe preparada para executar atividades específicas em diferentes áreas do conhecimento. A maioria das equipes de *startups* é homogênea, o que significa que os membros da equipe compartilham atributos, habilidades e características semelhantes [Bra19]. Esta é uma realidade nas *startups* entrevistadas. Das 10 (dez), 9 (nove) iniciaram suas operações com um sócio-fundador da área da tecnologia da informação e na média eram equipes compostas por 3 (três) fundadores. A complementaridade dos times foi um dos destaques nas entrevistas. A divisão das responsabilidades fica clara

em vários momentos dos depoimentos. Por exemplo, o empreendedor da *Startup H* cita a composição e as responsabilidades dos fundadores:

*O <sócio 1> é responsável pela parte de tecnologia, o <sócio 2> é responsável por conexões com o mercado e com a parte comercial e eu sou o responsável pela parte de produto entendimento do produto.*

Quando o assunto direcionava para a parte técnica, a **Arquitetura do Software** foi comentada como essencial nas fases mais avançadas de maturidade da *startup*. E indicavam que nas fases iniciais, principalmente na fase de ideação, um modelo de reuso de tecnologias e modelos já existentes fora a solução utilizada para agilizar a entrada no mercado. Segundo o empreendedor da *Startup I*, a criação de uma ferramenta temporária para validar o interesse do público foi desenvolvida com uma estrutura simples e um processo manual para atender a demanda:

*Não tinha nem site. Então a gente começou, a gente fez um sitezinho lá, coloco no ar ... e eu e o <sócio 1> criávamos o boleto manualmente...*

O empreendedor da *Startup I* indica um caminho semelhante no início das atividades da *startup*:

*.. fiz a planilha no excel para poder fazer os cálculos...*

Crowne [Cro02] argumenta que na fase inicial de um novo empreendimento, os empreendedores costumam ser tecnicamente capazes e o produto se desenvolve de forma rápida e direta. Porém, problemas críticos de desenvolvimento de produtos podem levar ao fracasso da empresa e são citados por Crowne [Cro02]:

- **P1 - Desenvolvedores inexperientes:** Os desenvolvedores trabalham heroicamente e os cronogramas de entrega do produto raramente são cumpridos. Quando o produto é entregue, ele não atende aos requisitos do cliente. Não é confiável e falha frequentemente no ambiente de produção. A retificação de defeitos leva mais tempo do que o esperado e geralmente cria mais defeitos.
- **P2 - O produto não é realmente um produto:** O produto, normalmente, é customizado para um determinado cliente, ou vários clientes, gerando um conjunto grande de versões que devem ser cuidadosamente mantidas.
- **P3 - O produto não tem dono:** Ninguém sabe quem tem autoridade para decidir quais recursos vão para o produto, ou a autoridade fica com um comitê que é disfuncional.

- **P4 - Não há planejamento para o desenvolvimento do produto:** O plano de negócios da empresa não estabelece objetivos e metas para o desenvolvimento de produtos. O *product owner* (PO) toma decisões importantes em uma base ad-hoc, sem ser capaz de conectá-las a um plano estratégico maior.
- **P5 - Plataforma do produto não reconhecida:** A importância das tecnologias e componentes que fazem parte do produto não é compreendida, discutida e gerenciada. A seleção desses componentes é deixada inteiramente para os desenvolvedores de produtos.

Analisando as *startups* entrevistadas, várias delas passaram ou passam por problemas desta natureza, conforme Tabela 5.2.

Tabela 5.2 – Problemas técnicos no desenvolvimento das *startups* entrevistadas

Startup	P1	P2	P3	P4	P5
Startup A			1	1	1
Startup B		1	1	1	1
Startup C	1			1	1
Startup D					
Startup E				1	
Startup F					
Startup G		1	1	1	
Startup H	1	1		1	1
Startup I					
Startup J	1	1			

As hipóteses relacionadas ao cliente ou ao mercado de atuação das *startups* são a base real e inicial para a captura dos **requisitos**. De acordo com as entrevistas, boa parte dos requisitos iniciais (principalmente nas fases de ideação e validação) são baseadas nas crenças do fundador. Relatos da dependência do fundador na elicitação dos requisitos, com foco fortemente em requisitos funcionais. Nenhum dos entrevistados apresentou preocupação em definir requisitos não-funcionais, porém apresentaram preocupações quando da evolução da *startup* e a revisão da tecnologia que fora escolhida inicialmente. Para a gestão dos requisitos, métodos ágeis e ferramentas visuais são as mais comentadas e utilizadas por essas empresas nascentes.

O **gerenciamento de projetos** em *startups* é realizado informalmente. Para pequenas empresas e *startups* é muito difícil gerenciar e alocar recursos entre os membros de equipes que trabalham em projetos diferentes ao mesmo tempo. Nas entrevistas, esta informalidade na gestão dos projetos ficou clara. Durante as fases de ideação e validação os empreendedores apresentaram pouco interesse em práticas de gestão de projetos, conforme o empreendedor da *Startup B*:

*...fizemos as pesquisas com os clientes no google forms e buscamos ali os itens que devem ser implementados... da planilha que é gerada pelo forms...*

e, também apresentado pelo empreendedor da *Startup J*:

*Hoje é tudo manual, e como a equipe também é pequena, não tem a necessidade de grandes protocolos para gestão dos projetos...*

Por outro lado, nas fases de maturidade mais avançadas, preocupações com planejamento, monitoramento e uso de ferramentas para organizar os projetos foram apresentadas. A *Startup C* indicou que *...agora usamos o JIRA para controlar o projeto e todos saberem o que deve ser feito*". As *startups D, E e H* indicaram o uso do Trello como a organizadora do *backlog*, enquanto que as *startups A e J* indicaram que o planejamento dos novos projetos é uma prática atual nessas empresas.

Em relação ao **teste de software**, todas as *startups* indicaram que iniciam as atividades executando testes ad hoc, normalmente pelo programador. A *startup I* indicou que algumas vezes o próprio cliente fazia o papel de testador.

*...em alguns momentos colocávamos a solução para o cliente sem termos testado internamente e esperávamos a posição dele...*

No atual momento, todas as *startups* implementaram algum tipo de automação de testes, sejam testes de unidade ou funcionais. E nenhuma possui um profissional dedicado a esta atividade dentro da equipe de desenvolvimento dos produtos.

Na categoria **processo de desenvolvimento de software** ou ciclo de desenvolvimento, somente há preocupação na formalização de algum processo após a fase de validação. A implementação de controles e estruturas para gerenciar adequadamente as atividades de desenvolvimento de software é um grande desafio para as *startups* de software. No contexto das *startups* entrevistadas, os fundadores (ou parte dos fundadores) possui conhecimento de desenvolvimento de software, porém, conforme relataram há preocupação na sobrevivência excluir qualquer ímpeto de formalizar o processo. Como disse o fundador da *startup B* "...estamos sempre correndo para conseguir virar o mês, não tenho tempo para implantar um processo. Quando crescermos pensaremos nisso...".

### 5.3 Síntese do capítulo

Este capítulo apresentou o resultado de um estudo de campo realizado com 10 *startups* em diferentes níveis de maturidade, com o intuito de buscar um maior entendimento sobre o uso de práticas de Engenharia de Software (questão de pesquisa RQ #1) e descobrir quais ferramentas elas utilizam durante a construção do MVP (questão de pesquisa

RQ #2). Além disso, alguns questionamentos foram direcionados para tentar descobrir em que momento a *startup* busca um maior formalismo no uso destas práticas e ferramentas de Engenharia de Software no seu ciclo de vida (questão de pesquisa RQ #3).

Na etapa de codificação das entrevistas, 6 categorias foram mapeadas, indicando as principais áreas da Engenharia de Software que são abordadas pela *startup*, independente do seu nível de maturidade. São elas: equipe, requisitos de software, arquitetura de software, gestão de projetos, teste de software e processo de desenvolvimento.

Algumas descobertas destas entrevistas podem ser resumidas nos seguintes itens:

- Equipe: (a) Competências generalistas são importantes nas fases iniciais de uma *startup*, bem como competências em inovação, negócios e tecnologia; (b) Heterogeneidade da equipe é importante para o seu crescimento, sendo mais necessárias à medida que novas necessidades surgem no contexto da *startup*.
- Arquitetura de software: nas fases iniciais da *startup*, os empreendedores optam por reusar soluções pré-existentes no mercado e somente após a fase de validação que inicia uma preocupação em escolher a melhor tecnologia para o problema que está sendo solucionado.
- Requisitos de software: Fases iniciais com foco em requisitos funcionais e dependência do conhecimento do fundador.
- Gestão de projetos: Informal e visual em um primeiro momento (fases iniciais) e uso de ferramentas e processos mais definidos nas fases de tração e escala.
- Teste de software: Pouco realizado no início da vida de uma *startup* (focado nos testes do programador e na resposta do uso pelos clientes ou usuários). Com o crescimento do negócio, automação foi um item citado e implementado por algumas *startups*.
- Processo de desenvolvimento: informal nas fases iniciais e alguma formalização após a fase de validação.

Com isso, foi possível capturar as práticas de Engenharia de Software utilizadas nas *startups* em diferentes estágios de maturidade, bem como as ferramentas que foram utilizadas. Assim, o primeiro esboço do modelo proposto envolveu a identificação destas seis categorias, que serviram de insumo para as etapas seguintes desta pesquisa.

## 6. FASE II - ESTUDO DE OBSERVAÇÃO - SURVEY

*Startups* possuem um modo peculiar para desenvolver um novo produto. A flexibilidade e reatividade das *startups* permitem um rápido desenvolvimento e lançamento de produtos inovadores [KUG19]. Apesar de existirem alguns estudos relacionados ao desenvolvimento de software nas *startups* ([KUC+19], [LPE+19], [GPU+15], [MCSP20]) nenhum explora o momento em que a *startup* adota ou deixa de adotar alguma prática de Engenharia de Software.

Assim, a partir das práticas classificadas nas 6 categorias identificadas no estudo de campo apresentado no Capítulo 5, foi planejado um estudo exploratório realizado com 41 *startups* visando entender não somente as práticas de Engenharia de Software utilizadas por estas empresas nestas 6 categorias, mas qual a relação da adoção das práticas de acordo com a etapa de maturidade de cada *startup*. Como base foi utilizado o modelo de maturidade apresentado pelo IBGC [LAM19] descrito no Capítulo 2. Este capítulo apresenta os resultados deste estudo.

### 6.1 Protocolo do estudo

Algumas informações importantes podem ser extraídas do resultado das respostas deste *survey*, muito em razão da diversidade de *startups*, suas áreas de atuação, o momento de cada uma delas e as diferentes localizações no Brasil. A aplicação deste *survey* obteve a participação de startups originárias de 7 (sete) estados brasileiros: Rio Grande do Sul (48.8%), Santa Catarina (17.1%), Paraná (9.8%), Rio de Janeiro (9.8%), Pernambuco (7.3%), São Paulo (4.9%) e Mato Grosso do Sul (2.4%).

O *survey* foi concebido com a preocupação de ter uma abrangência nos eixos levantados na fase anterior (estudo de campo) e que indicam áreas da Engenharia de Software que uma *startup* considera durante suas etapas de crescimento. Foram abordadas nas questões os seguintes eixos:

- Equipe de desenvolvimento de software;
- Requisitos de software
- Arquitetura de software
- Teste de software
- Gerenciamento de projetos
- Processo de desenvolvimento de software

O *survey* foi criado utilizando a ferramenta *Google Forms* e disponibilizada online para que fosse possível uma maior abrangência dos resultados. Para divulgação, as redes sociais e grupos do *Whatsapp* de *startups* e gestores de ambientes de inovação foram utilizados.

A população-alvo deste estudo foram empreendedores de *startups* nas diferentes fases de maturidade. A utilização da ferramenta para coleta dos dados deste *survey* (*Google Forms*) evitou que a pesquisa fosse respondida mais de uma vez pelo mesmo respondente. A participação foi voluntária e o preenchimento do formulário levou entre 7 e 10 minutos.

Para validar o *survey* criado (Apêndice B), ele foi aplicado, por conveniência, em duas *startups* residentes do TECNOPUC e analisado por um especialista em startups. Após análise da percepção das startups e do especialista, o *survey* foi ajustado e as respostas descartadas.

Durante um período de 2 semanas foram coletadas informações de 41 diferentes startups. Após o período de coleta, as informações foram analisadas e as descobertas são apresentadas neste capítulo.

## 6.2 Análise das respostas

Analisando o perfil dos 41 respondentes, pode-se observar que não foram identificadas *startups* na fase de ideação, conforme Figura 6.1. Porém, as startups respondentes analisaram a adoção das práticas de Engenharia de Software de acordo com o histórico de cada uma. Na Figura 6.1, a análise foi feita de acordo com a história de cada *startups* nas fases em que se encontravam no momento da pesquisa e nas fases que elas já tinham passado.

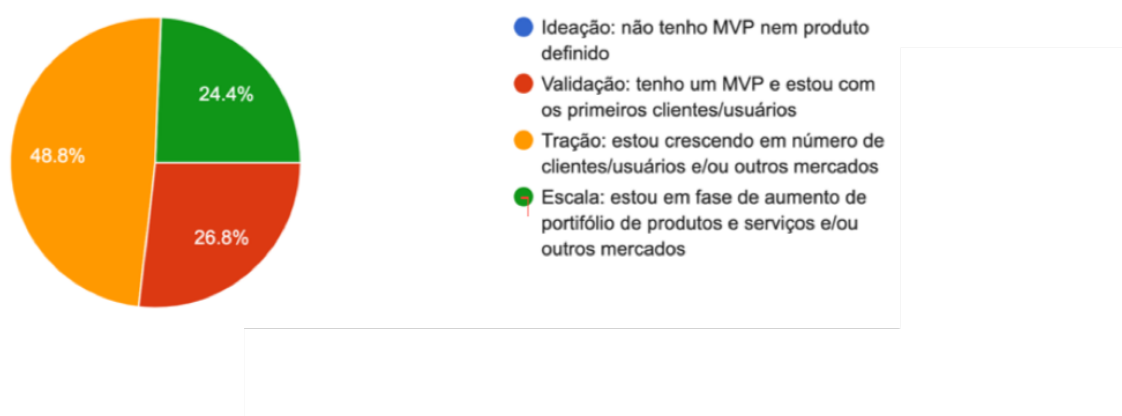


Figura 6.1 – Maturidade das Startups



Isto pode ter ocorrido pelo fato de que *startups* na fase inicial (ideação) ainda não possuem experiência e resultados de desenvolvimento de software. Elas focam muito mais na identificação do que deve ser apresentado ao mercado pela startup para solucionar o problema do cliente ou no nicho de clientes escolhido.

Como observa-se na Figura 6.2 na fase de Ideação as *startups* tinham uma preocupação maior em desenvolver suas equipes de desenvolvimento e identificar os requisitos necessários para a criação do MVP. Este gráfico (Figura 6.2) apresenta o resultado da reflexão dos empreendedores referente a quais práticas de Engenharia de Software (requisitos de software, teste de software, gerenciamento de projetos, arquitetura de software, equipe e processo de desenvolvimento) foram importantes nas fases do negócio (ideação, validação, tração e escala).

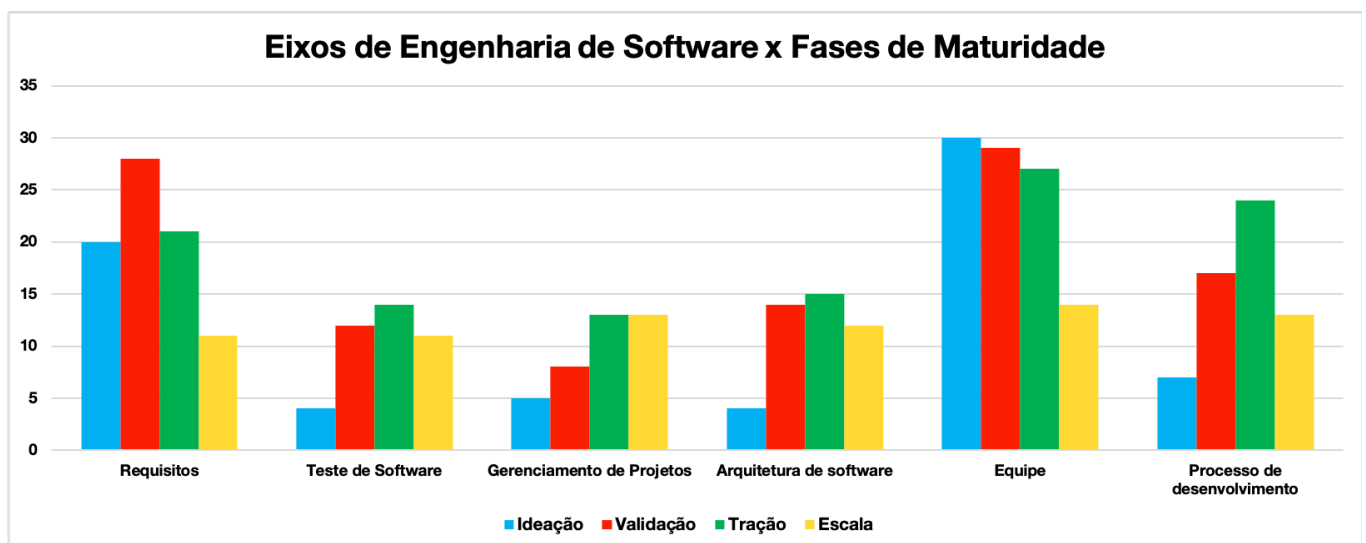


Figura 6.2 – Engenharia de Software x Fases de Maturidade

Klotins [KUC<sup>+</sup>19] apresentou que em fases iniciais, chamadas de *"inception"* e *"stabilization"*, construção da primeira versão do produto e preparação do produto para escalar, respectivamente, as startups definem atividades prioritárias relacionadas a: (1) equipe (ter um time efetivo, trabalhando com engajamento, liderança e complementar); (2) requisitos (validação das ideias, processo de incorporar as ideias dos clientes no produto, definição do escopo do MVP e gestão de mudanças); (3) teste de software (funcional, definição do nível de qualidade *"good-enough"*, definição dos requisitos de qualidade e execução manual de testes); (4) projeto e arquitetura de software (mitigação de riscos de tecnologia e utilização de modelos já existentes); (5) gerenciamento de projetos (foco no desempenho do projeto); e (6) proposta de valor (externo, a partir da visão do cliente, e interno, a partir da análise de potencial de mercado).

Observa-se que a equipe de desenvolvimento é uma das grandes preocupações por parte das *startups*. De acordo com Giardino et. al. [GPU<sup>+</sup>15], os primeiros integrantes da equipe de desenvolvimento são os fatores de sucesso e de grande produção das *star-*

*tups*, com entregas rápidas, com comprometimento e assumindo múltiplos papéis de forma auto-organizada para realizar as entregas que a *startup* necessita. Fica evidente que este foco na equipe pode ocasionar efeitos colaterais no produto que está sendo desenvolvido, por exemplo, o acúmulo de dívida técnica pela não estruturação ou uso de práticas de Engenharia de Software nas fases iniciais do negócio [GPU+15].

Analisando as respostas deste *survey*, observa-se que 92.7% das *startups* tiveram problemas com erros no produto desenvolvido, 58.5% tiveram que desenvolver novamente alguma funcionalidade por erro na compreensão do problema/requisito e 68.3% tiveram que resolver problemas de usabilidade das suas soluções. Estes problemas podem indicar que uma melhor definição do processo de desenvolvimento auxiliaria na diminuição destes problemas. A Figura 6.3 mostra que somente 17.1% possuem um processo formalizado, em que há a definição do processo e este é seguido por toda a equipe de desenvolvimento.

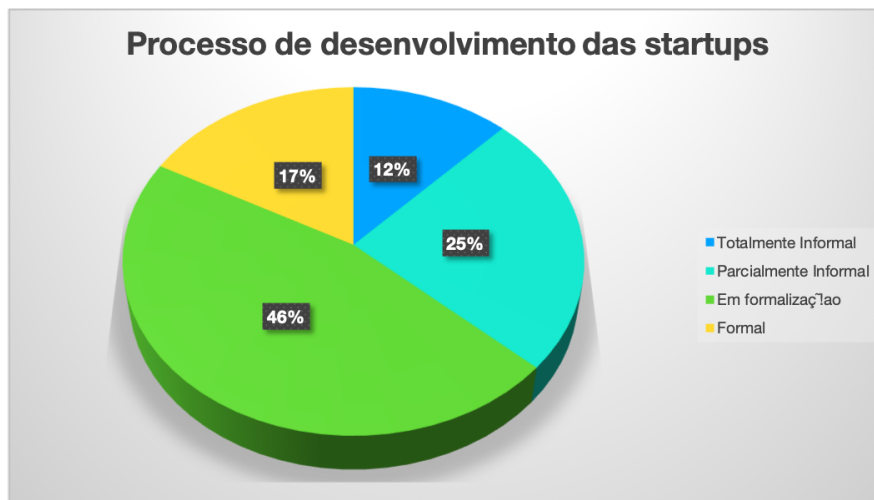


Figura 6.3 – Processo de desenvolvimento das *startups*

Analisando os papéis que as startups possuem ou possuíam nas diferentes fases de maturidade do negócio, percebe-se uma diversidade de responsabilidades da equipe (Figura 6.4).

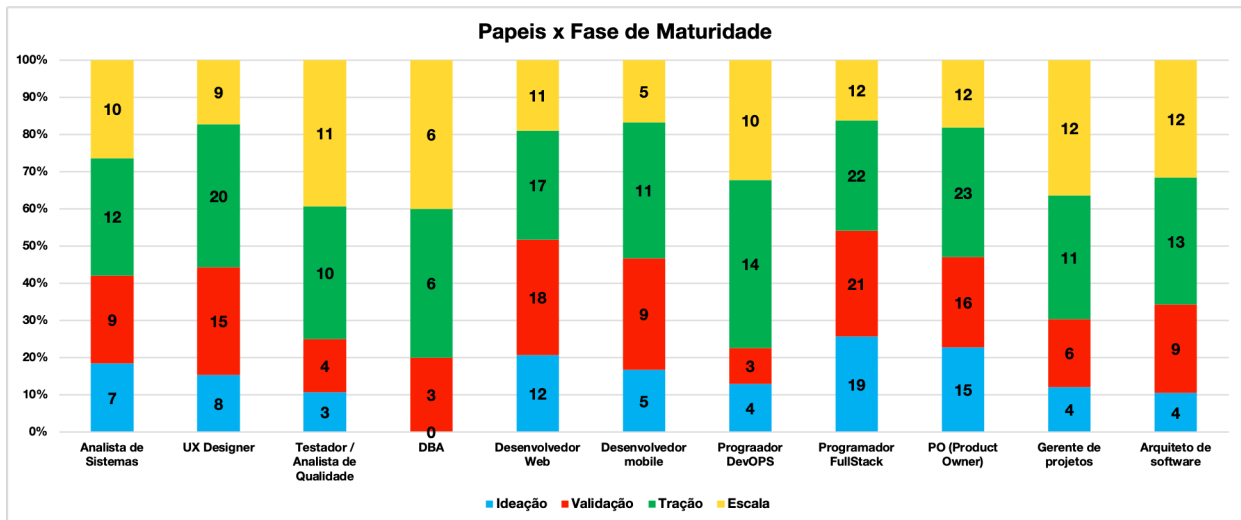


Figura 6.4 – Papeis x Maturidade

Pontos importantes demonstrados a partir das respostas obtidas:

- Para o desenvolvimento de um MVP baseado em hipóteses não validadas, isto é, sem o conhecimento do que o cliente deseja ou da certeza da solução proposta pela *startup*, papéis mais genéricos são mais presentes. Programador *FullStack*, por exemplo, foi o papel que possuía uma presença significativa nas diferentes fases de maturidade, principalmente as iniciais (ideação, validação e tração).
- Desenvolvedor Web, desenvolvedor *FullStack* e *Product Owner* estão presentes de forma significativa na fase de ideação. Pode-se afirmar que estes papéis demonstram importância na concepção da primeira versão do produto, seja pela análise do que o mercado está desejando (através do PO), seja através da codificação de uma solução (seja web ou genérica).
- Apesar da importância de entendimento do mercado em todas as fases, o PO tem uma relevância maior na fase de tração. Pode-se afirmar que isto ocorre, pois nesta fase as hipóteses de cliente, mercado e solução já foram validadas nas fases anteriores e o momento da *startup* é o de crescimento no mercado escolhido.
- A gestão de projetos e a arquitetura de software são apresentadas de forma mais informal nas fases iniciais (ideação e validação) pelos fatores de incertezas da *startup* e possuem um crescimento na importância nas fases de tração e escala, muito provavelmente pelo fato de aumento de equipe, funcionalidades e diversificação de produtos e serviços.
- A inexistência de uso de técnicas de elicitação de requisitos na fase de ideação (Figura 6.5) pode ser analisada pelo fato dos requisitos desta fase serem oriundos do conhecimento ou experiência dos fundadores. Por outro lado, a fase de tração mostra uma grande relevância para esta prática, principalmente através de *brainstorming*, observação e pesquisas com clientes ou usuários.

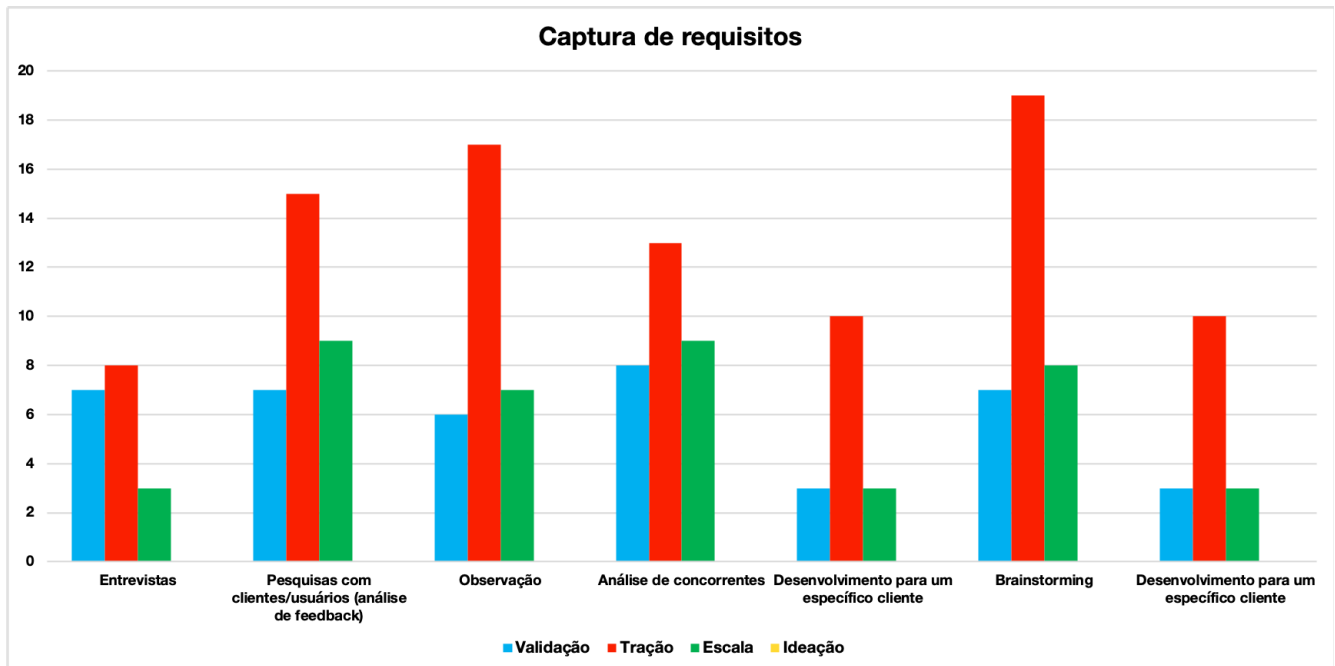


Figura 6.5 – Elicitação de requisitos pelas *startups*

Sobre a arquitetura do software pode-se afirmar que as opções de projeto adotadas afetam não apenas as funções da solução, mas também as considerações relacionadas ao fornecimento, como custo, tempo e escalabilidade. Portanto, é extremamente importante considerar métricas adequadas para calcular as vantagens e desvantagens de várias opções de projeto para otimizar o ganho de vida útil e reduzir o custo total de propriedade do produto de software [NOKGR12].

A qualidade destas definições fica clara para o empreendedor à medida que o projeto da solução se consolida e as hipóteses são validadas. A Figura 6.2 mostra a relevância da arquitetura de software, principalmente nas fases de tração e escala. Mas para os empreendedores os grandes objetivos de se ter qualidade na arquitetura do software está relacionado a:

- Tempo de lançamento no mercado;
- Funcionalidade (precisão, interoperabilidade com outros sistemas, segurança, conformidade de funcionalidade);
- Confiabilidade (maturidade, tolerância a falhas, capacidade de recuperação de erros, conformidade de fiabilidade);
- Eficiência (tempo e utilização de recursos, conformidade com a eficiência);
- Manutenibilidade (analísabilidade, mutabilidade, estabilidade, capacidade de teste, conformidade de manutenção);
- Portabilidade (adaptabilidade a diferentes ambientes, instabilidade, coexistência com outros sistemas, substituíbilidade, conformidade com a portabilidade).

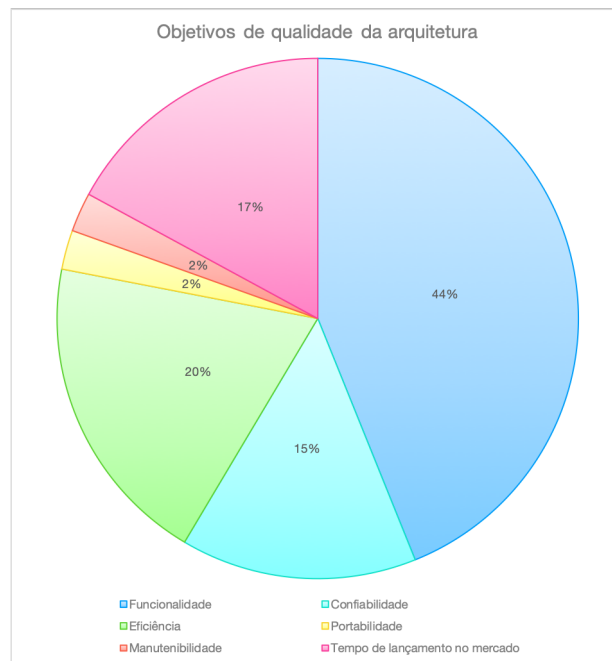


Figura 6.6 – Objetivos de qualidade da arquitetura

O projeto da arquitetura é essencial para a qualidade da solução proposta pelos empreendedores. De acordo com a Figura 6.6 as funcionalidades definidas para o MVP são pontos-chave para uma boa definição de arquitetura, sem deixar de lado a eficiência (20%) e o tempo de lançamento no mercado (17%). Portabilidade e manutenção não foram apresentadas como críticas para estas definições. Pode-se inferir que em momentos de definição do que deve ser a solução da *startup* e como ela deve ser apresentada ao cliente, os empreendedores se preocupam com entregar algo que estes clientes usarão (funcionalidade), no tempo adequado (tempo de lançamento) realizando exatamente o que fora definido (eficiência) de forma confiável (confiabilidade).

Quando questionadas sobre práticas de testes de software, os respondentes apresentaram algo que já fora apresentado em outras pesquisas relacionadas ao tema. Leal et. al. [LPE<sup>+</sup>19] indica que testes de usabilidade são as práticas de testes mais comum entre as *startups*. Klotins [KUC<sup>+</sup>19] apresenta reflexões de *startups* que indicam o uso de testes de usabilidade para validar hipóteses e melhorar as soluções apresentadas pela *startup*. Giardino et. al. [GUP<sup>+</sup>14] indica que a aplicação de práticas de qualidade nas funcionalidades de maior valor para os clientes é algo crítico. Na aplicação do *survey* os resultados estabeleceram que a usabilidade e interação humano-computador é algo essencial nas fases de validação, tração e escala (vide Figura 6.7). Além disso, testes de aceitação são importantes, tendo um destaque na fase de tração. Isto pode ocorrer pelo aumento de clientes ou usuários e um maior *feedback* para mudanças nas funcionalidades dispostas, e consequentemente, uma preocupação por parte da *startup* em entregar algo sem a presença de erros.

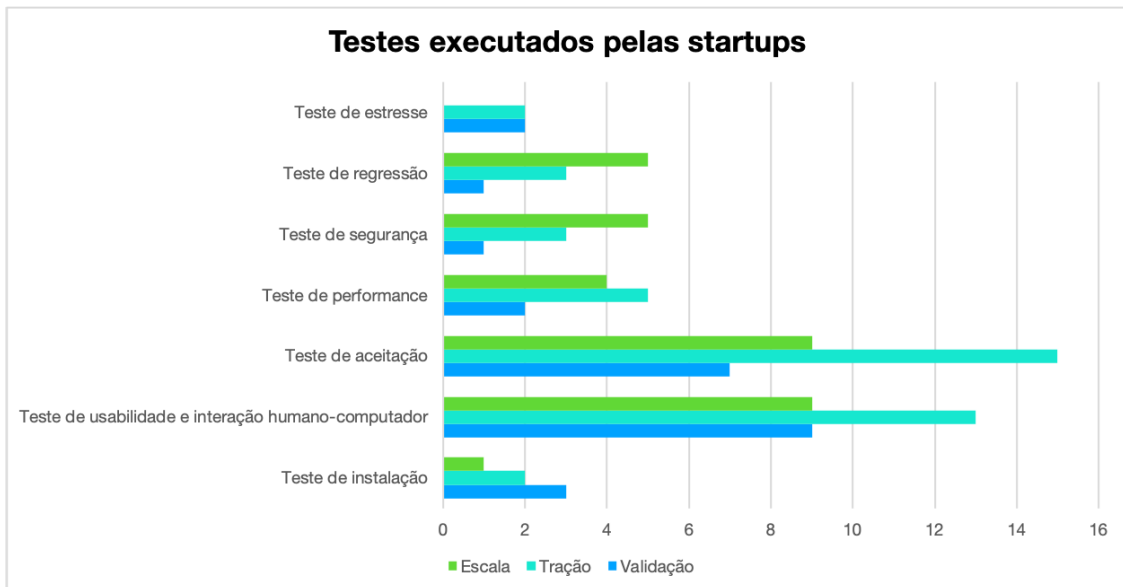


Figura 6.7 – Testes executados pelas startups

No que tange o processo de desenvolvimento de software, o conjunto de *startups* respondentes deste *survey* utilizam na grande maioria um processo com práticas ágeis (Figura 6.8). Um total de 82.9% utilizam métodos ágeis o que está de acordo com pesquisas similares realizadas. Por exemplo, em [LPE<sup>+</sup>19] 86% das *startups* indicaram adotar práticas oriundas de métodos ágeis, enquanto que o estudo apresentando em [Gat12] apresenta que mais de 70% utilizam métodos ágeis.

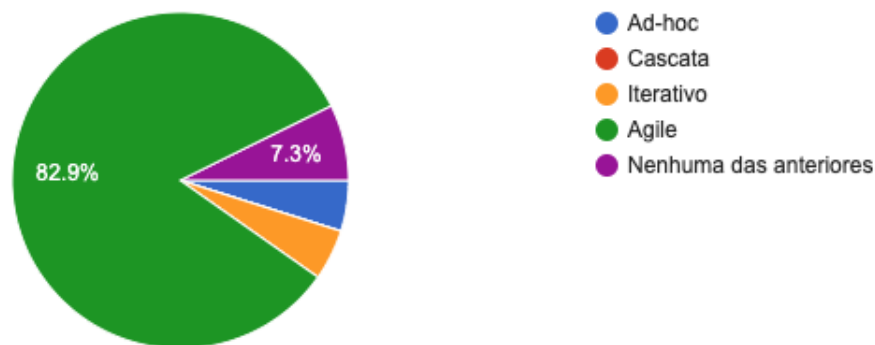


Figura 6.8 – Processo de desenvolvimento das *startups*

Giardino [GPU<sup>+</sup>15] apresenta o conceito de "*speed up development*" que representa uma abordagem de desenvolvimento de software baseada no foco da equipe em implementar um conjunto mínimo de funcionalidades que atendam ao momento de validação da *startup*. Isto está alinhado aos princípios ágeis, que são comumente usados pelos empreendedores para conseguir atender ao mundo de incertezas e mudanças que se encontram.

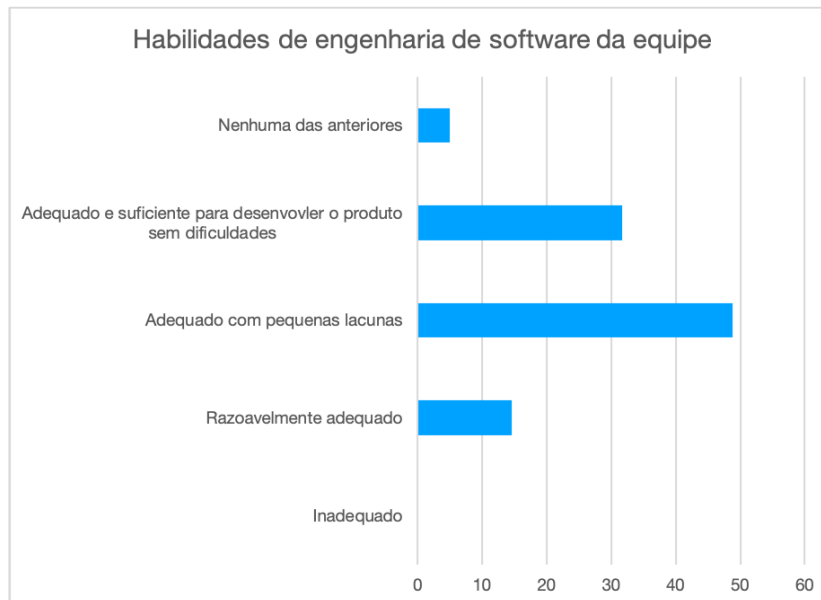


Figura 6.9 – Habilidades de Engenharia de Software da equipe

Desta forma a equipe de desenvolvimento deve ser madura o suficiente para entender as responsabilidades individuais e, de forma alinhada e integrada, realizar as entregas necessárias para que a startup atenda seus objetivos. A Figura 6.9 mostra que boa parte das *startups* pesquisadas possui equipes com habilidades adequadas aos objetivos de desenvolvimento da empresa, porém com algumas lacunas. Isto pode ser entendido como a falta de alguns perfis necessários ou a falta de conhecimento profundo em determinadas tecnologias.

### 6.3 Síntese do capítulo

Neste capítulo foram apresentadas as informações obtidas de um *survey* planejado e respondido por 41 *startups* de software. Estas *startups*, conforme apresentado, encontram-se nas fases de validação, tração ou escala, o que significa que já possuem algum MVP no mercado com algumas validações. O *survey* foi abrangente o suficiente para entender como as *startups* se posicionam ou se posicionaram de acordo com seis áreas (equipe de desenvolvimento, requisitos, teste de software, arquitetura de software, gerenciamento de projetos e ciclo de desenvolvimento de software).

Algumas respostas seguiram uma tendência apresentada em outras pesquisas similares, como por exemplo, a adoção de métodos ágeis pelas *startups*. Porém alguns pontos foram apresentados e se transformam em itens de atenção para esta pesquisa:

- Na fase de ideação, papéis mais genéricos são mais importantes do que os mais específicos;

- O projeto de software, organizado como se conhece em grandes empresas, começa a ser uma preocupação em fases mais avançadas da *startup* (tração ou escala). Antes disso, o número de incertezas quase que obriga que as *startups* tenham equipes dinâmicas, generalistas e com capacidade de adaptação rápida;
- A preocupação com as escolhas tecnológicas, a robustez da solução ou a arquitetura da solução é algo que interfere na *startup* somente após as validações iniciais de mercado;
- Os testes de software são realizados praticamente pelos clientes ou usuários, principalmente enquanto há escassez de recursos por parte da *startup* (fases de ideação e validação).

Para melhor representar os níveis de maturidade de adoção das práticas de Engenharia de Software foi criada uma representação gráfica (Figura 6.10). OS níveis de maturidade de adoção foram baseados no modelo apresentado pelo IBGC [LAM19] e adaptado de acordo com as análises realizadas nesta pesquisa.



Figura 6.10 – Versão 1 dos níveis de maturidade de adoção de práticas de Engenharia de Software

A partir da análise dos dados coletados neste *survey*, foi possível evoluir para uma versão do modelo (Figura 6.11) apresentando as práticas de Engenharia de Software e a sua adoção de acordo com o momento de maturidade do negócio da *startup*.

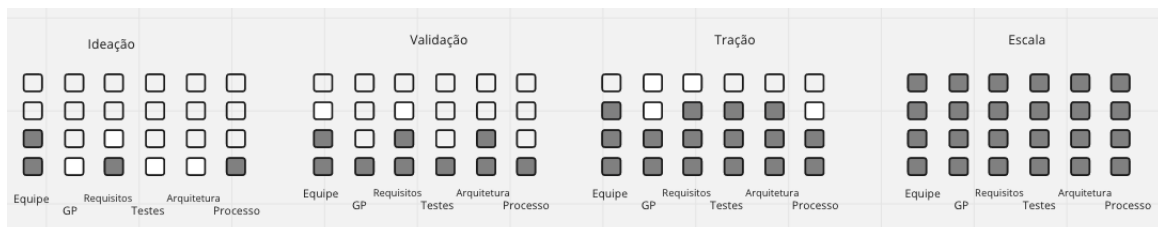


Figura 6.11 – Versão 1 do modelo de adoção de práticas de Engenharia de Software por *startups*

Para cada eixo apresentado no modelo, a maturidade do uso da prática é indicado com o preenchimento dos quadrados indicados no modelo, mostrando assim, dentro da maturidade do negócio (ideação, validação, tração ou escala) qual o nível de adoção das



práticas (equipe, gerenciamento de projetos, requisitos de software, testes de software, arquitetura de software e processo de desenvolvimento de software).

Para as equipes de desenvolvimento, estar em um nível inicial significa possuir uma equipe auto-gerenciada, evoluindo em um nível intermediário para ter uma orientação focada no negócio, ou seja, o desenvolvimento do software necessário é baseado no entendimento das necessidades do negócio que a equipe adquire durante o desenvolvimento da *startup*. Estar em um nível avançado, significa adicionar uma preocupação com os perfis técnicos da equipe, para chegar em um nível maduro de equipe que contemple uma complementariedade destes perfis.

No item requisitos de software, a *startup* inicia com o foco total no entendimento do negócio e do mercado que deseja atuar (inicial), passando no nível intermediário pelo entendimento e mapeamento das dores do cliente em requisitos de software. Nos níveis avançado e maduro, a *startup* se preocupa pela definição e uso de ferramentas e processos, respectivamente.

Para testes de software, um nível inicial os testes são realizados de forma ad hoc pela equipe ou, em alguns casos, pelo cliente (neste caso, a *startup* lança uma versão do MVP para *early adopters* e monitora o uso para identificação de eventuais problemas. No nível intermediário, a equipe inicia a implementação de testes de unidade. No nível avançado, o PO executa testes funcionais em busca de problemas e melhorias nas funcionalidades disponibilizadas no MVP. Já no nível maduro, a *startup* possui uma preocupação em estabelecer um processo de testes de software com um profissional ou equipe dedicada a estas atividades.

O nível inicial da arquitetura de software, a *startup* cria seu MVP com soluções tecnológicas pré-existentes, chamado de "modelo lego de arquitetura". Com um melhor entendimento do mercado e das dores do cliente (nível intermediário), os fundadores com formação técnica criam a solução tecnológica baseada em experiências anteriores. No nível avançado, a *startup* passa a se preocupar com o posicionamento de mercado e com isto realizada uma análise de soluções similares existente no mercado e se baseia nesta comparação para a definição da arquitetura de software. No nível maduro, a *startup* cria a solução tecnológica baseada nas necessidades do mercado atuante.

Para o gerenciamento de projetos, a *startup* inicia (nível inicial) com o PO executando o papel de gerente de projetos, basicamente verificando junto ao mercado o que deve ou não deve ser desenvolvido e repassando estas informações para a equipe de desenvolvimento. No nível intermediário, a *startup* começa a usar ferramentas para um melhor planejamento das atividades da equipe e passa, no nível avançado, a usar ferramentas para monitorar e melhorar o processo de desenvolvimento. No nível maduro, a *startup* começa a ter novos projetos em paralelo, fazendo com que o uso de ferramentas seja necessário para otimizar o uso da equipe de desenvolvimento.

No que tange o ciclo de desenvolvimento, a *startup*, no nível inicial, o processo está totalmente informal, passando a ter um processo minimamente definido, focado na programação (nível intermediário). No nível avançado, a *startup* utiliza ferramentas de gerenciamento de configuração e possui um processo de desenvolvimento de software melhor definido. No nível maduro, a *startup* utiliza ferramentas em todas as etapas do desenvolvimento de software.

Estas definições serviram para um melhor entendimento do que as *startups* focam e aplicam nos diferentes níveis de maturidade do negócio.

Na fase de ideação, foi observado que as *startups* priorizam os componentes de suas equipes e suas expertises técnicas com capacidade de se auto-gerenciarem e com foco no desenvolvimento do negócio e nos primeiros requisitos de software. Outro fator importante nesta fase é a orientação da equipe na construção do MVP, baseado nos requisitos de software levantados pela equipe sem uma preocupação com o formalismo de um processo de desenvolvimento.

Na fase de validação, a preocupação dos empreendedores continua sendo com a equipe altamente capacitada tecnicamente e auto-gerenciada, porém com a entrada no mercado e o retorno deste em relação ao uso do MVP lançado, requisitos técnicos começam a fazer parte da rotina da *startup*, bem como a realização de testes ad hoc ou até mesmo pelos próprios clientes e a correção de *bugs*. A arquitetura do software é criada com base no conhecimento e experiência dos fundadores e partem da premissa do uso de componentes já existentes e de conhecimento da equipe. O gerenciamento de projetos aparece somente na atuação de um dos fundadores, normalmente atuando na figura de PO e estabelecendo a relação com o mercado e clientes.

Na fase de tração, a *startup* possui preocupações com a produtividade da equipe e o tempo de resposta para as necessidades do mercado. Práticas de gestão de projetos começam a ser inseridos no contexto do desenvolvimento, há um maior uso de ferramentas e controles no processo de desenvolvimento, há uma busca por automação de testes e uma preocupação em desenvolver uma arquitetura de software robusta o suficiente para suportar o crescimento do negócio.

Na fase de escala, a *startup* já é considerada uma *scale-up* e com características próprias de uma empresa. Há um rigor maior no desenvolvimento, na constituição da equipe, muitas vezes distribuída. Com isso, a *startup* começa a ter um processo mais formal de desenvolvimento, com o uso de ferramentas para auxiliar na comunicação da equipe e na gestão e priorização dos requisitos de software. As decisões relativas a arquitetura de software são baseadas nas necessidades do negócio e novos projetos são iniciados pela *startups*, requerendo um maior rigor na gestão de projetos.

## 7. FASE III - ESTUDO DE AVALIAÇÃO - GRUPO FOCAL

Este capítulo apresenta o resultado do estudo realizado com um grupo focal realizado com 4 *startups*. Estas *startups* têm seu perfil descrito a seguir e foram escolhidas pelo nível avançado de maturidade (mínimo no nível de tração no modelo do IBGC) e por estarem localizados em diferentes cidades.

### 7.1 Protocolo do estudo

Um grupo focal é um método de pesquisa qualitativa que funciona selecionando um grupo de pessoas, colocando-as em um ambiente interativo e investigando suas percepções, opiniões, crenças e atitudes sobre um tópico específico [Mor96]. Este estudo realizou um grupo focal com a participação de 4 *startups* e teve como principal objetivo de avaliar o modelo proposto e descrito no capítulo anterior. As *startups* participantes se encontravam nas fases de tração ou escala e foi executado para melhor entender o uso das práticas de Engenharia de Software durante as fases de construção dos seus produtos ou serviços e identificar as práticas comumente utilizadas pelas *startup* nas fases de ideação, validação, tração e escala.

Como mecanismo de alinhamento inicial com estas startups, uma técnica chamada de classificação de cartões foi utilizada na fase inicial do grupo focal, com o intuito de "quebra-gelo" e para buscar informações pontuais sobre cada uma das *startups*.

Após a definição dos cartões e dos questionamentos a serem utilizados no grupo focal, uma *startup* residente do Tecnopuc foi convidada para fazer a validação da aplicação das técnicas. Esta validação teve como objetivos a verificação da clareza dos questionamentos propostos e o tempo necessário para as respostas dos cartões. Após a análise destes itens, os questionamentos e os cartões foram ajustados e, então, utilizados na execução do grupo focal.

Em um dia e hora marcado as startup se reuniram via zoom ([www.zoom.us](http://www.zoom.us)) e a dinâmica da classificação dos cartões e do grupo focal foi realizada. a sessão foi gravada e posteriormente transcrita para uma análise mais profunda sobre as respostas e discussões apresentadas pelas *startups*.

O protocolo do grupo focal encontra-se no Apêndice D e os cartões utilizados estão no Apêndice C.

## 7.2 Caracterização das *startups*

Nesta seção são apresentados os perfis das *startups* participantes do grupo focal.

### **Startup K**

*Startup* que atua no segmento de seguros e possui uma plataforma para facilitar as negociações com seguros empresariais. Foca em facilitar a experiência do cliente com seguros, comparando todas opções para encontrar a cobertura correta, realizar a compra e ainda gerir suas apólices de forma online e segura. Utiliza tecnologias de inteligência artificial e aprendizado de máquina para melhorar a assertividade junto aos seus clientes. Conta com mais de 1.200 (mil e duzentos) clientes ativos e apresenta um crescimento na base de 10% ao mês. Sua base operacional está localizada na cidade de Canoas, no Rio Grande do Sul. A *startup* foi representada pelo sócio-fundador que possui o papel de CTO.

### **Startup L**

*Startup* do segmento da saúde, com uma solução que auxilia as equipes hospitalares na criação e gestão das escalas de trabalho dos profissionais. Utiliza inteligência artificial para a geração das escalas e está presente em grandes hospitais do país. Sua sede se localiza na cidade de Porto Alegre. A *startup* foi representada pelo sócio-fundador que possui o papel de CEO.

### **Startup M**

*Startup* que possui uma solução direcionada ao relacionamento entre equipes de trabalho, trabalhando com processos colaborativos para dimensionar a comunicação de uma marca através da gestão de ativos digitais. Sua sede está localizada na cidade de São Leopoldo, no Rio Grande do Sul. A *startup* foi representada pelo sócio-fundador responsável pelo desenvolvimento da solução.

### **Startup N**

*Startup* que conta com uma solução que simplifica o processo entre os consumidores e produtos vendidos em plataformas de *e-commerce*. Possui clientes espalhados pelo Brasil e a sede está localizada na cidade de Campo Grande do Mato Grosso do Sul. A *startup* foi representada pelo sócio-fundador que concebeu o negócio e criou as primeiras versões da solução.

### 7.3 Classificação de Cartões

O objetivo principal da metodologia de classificação de cartões ou *Card Sorting* é ajudar as pessoas a realizar uma categorização de um conjunto de elementos [MMRAGCHV19]. Esta metodologia foi escolhida por ajudar os participantes na compreensão inicial das discussões que foram realizadas no grupo focal. A proposta, para esta pesquisa, foi obter um retrato da situação das *startups* em determinados momentos de sua jornada. Aliado ao grupo focal, ele trouxe uma visão bastante objetiva de cada *startup*, o que serviu como um suporte ao entendimento das questões trazidas por cada participante.

Basicamente os participantes de um grupo recebem um conjunto de cartas que devem classificar juntos em vários grupos diferentes deixando-os livres para uma tomada de decisões sem influências. Os métodos de classificação de cartões [RK71] podem ter dois propósitos diferentes:

- Estudar como as pessoas classificam um determinado conjunto de objetos: Pela forma como uma determinada quantidade de pessoas classifica determinados objetos em diferentes grupos é possível determinar como pensam ou como se sentem sobre um determinado assunto.
- Ajudando os usuários a realizar classificações complexas: O método de classificação de cartões é uma maneira conveniente para os especialistas realizarem processos de classificação, usando sua afinidade, semelhança ou outros critérios.

Este estudo foi realizado de forma virtual, disponibilizando os cartões aos participantes por meio da ferramenta Jamboard (<https://jamboard.google.com/>) com as 4 (quatro) *startups*. A classificação de cartões foi realizada antes do grupo focal, com tempo estimado de 10 minutos. Cada participante recebeu um link contendo 5 conjuntos de cartões (Apêndice C), com objetivo de coletar informações sobre o momento da *startup*.

A análise foi realizada de forma individual, em um primeiro momento, e coletiva num segundo momento, onde se buscou identificar padrões e contrastes entre os diferentes momentos de maturidade das *startups*. Foram coletados dados sobre o momento da *startup*, categorias mais importantes em cada fase, responsável por definir os requisitos em cada fase, alterações e suas proporções em cada fase e equipe de TI em cada fase.

A seguir, são apresentados os resultados da aplicação da metodologia para cada um dos cartões.

### 7.3.1 Cartão #1: Momento da *Startup*

As quatro *startups* estavam em momentos de maturidade muito parecidos entre tração e escala. A *Startup K* estava em uma fase de tração, enquanto as outras já se encontravam em um momento de transição da fase de tração para a fase de escala.

### 7.3.2 Cartão #2: Categorias mais importantes por etapa

*Startup K*: Teve grande variação de quantidades e tipos de categorias. Na etapa da ideação, os requisitos foram a categoria mais importante, não sendo mencionado nenhum outro. Na validação, foi a equipe e a arquitetura de software. Na tração, arquitetura de software se manteve, incluindo também gerenciamento de projeto, ciclo de desenvolvimento, teste de software e equipe. A Figura 7.1 apresenta o que a *startup K* considerou mais importante em cada uma das etapas de maturidade.

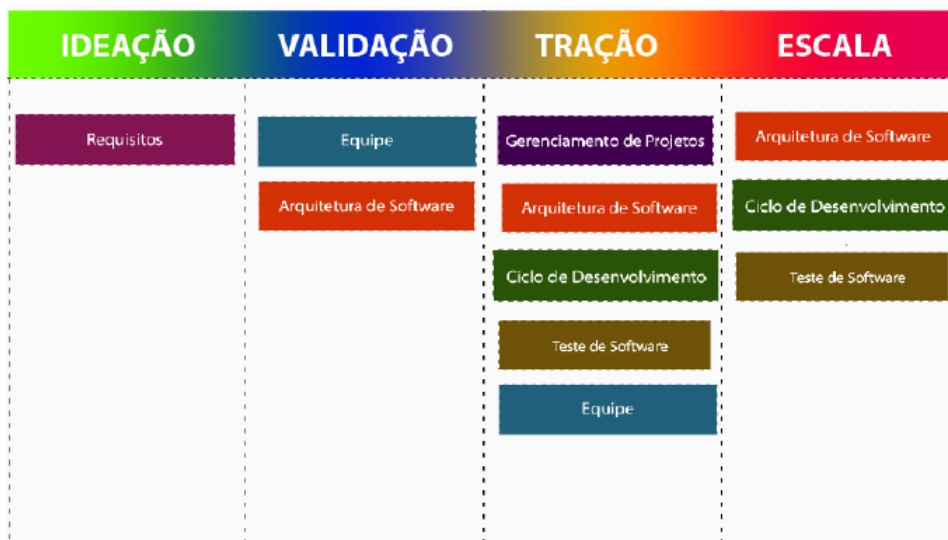


Figura 7.1 – Categorias mais importantes - *Startup K* (Fonte: Autor)

*Startup N*: Teve alguma variação de categorias, tendo elencado uma maior proporção delas nas primeiras etapas. Na ideação as categorias requisitos, gerenciamento de projetos, equipe e arquitetura de software foram mais importantes. Na validação, estas mesmas se mantiveram, sendo acrescentadas ciclo de desenvolvimento e teste de software. Na tração, mantiveram-se o gerenciamento de projetos e o ciclo de desenvolvimento. Na escala não foram mencionadas categorias. A Figura 7.2 apresenta as escolhas da *startup*.

*Startup L*: Houve alguma variação de categorias. Na ideação, apenas os requisitos e a equipe foram mencionados como mais importantes. Na validação, foram acrescentadas gerenciamento de projetos, arquitetura de software e ciclo de desenvolvimento. Na tração e



Figura 7.2 – Categorias mais importantes - *Startup M* (Fonte: Autor)

na escala, mantiveram-se apenas arquitetura de software e teste de software. A Figura 7.3 mostra o posicionamento da *startup* referente a sua jornada de maturidade.

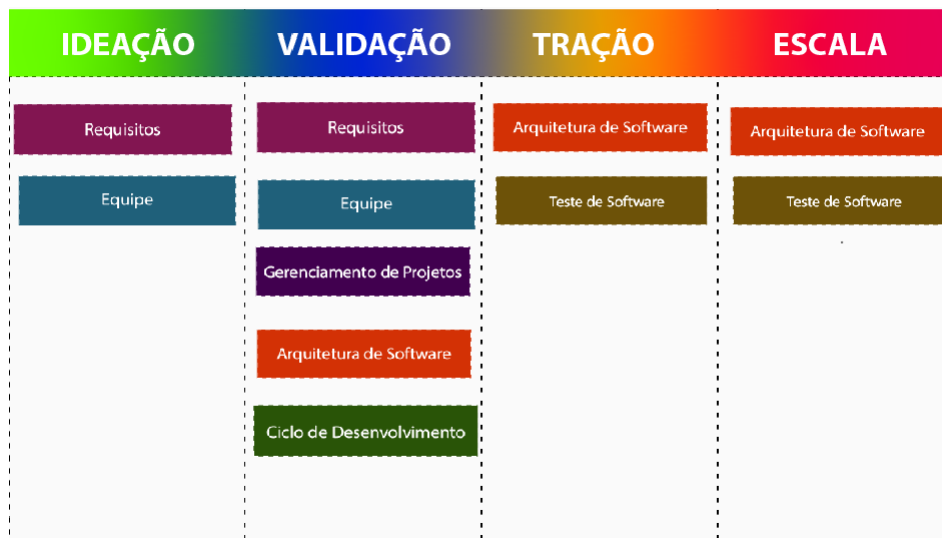


Figura 7.3 – Categorias mais importantes - *Startup L* (Fonte: Autor)

*Startup M*: Houve alguma variação de categorias. Na Ideação, equipe e requisitos foram citados como mais importantes. Na Validação, equipe, ciclo de desenvolvimento e gerenciamento de projetos. Na tração, foi acrescentado a arquitetura de software. Na escala, foram mantidas equipe, ciclo de desenvolvimento, arquitetura de software e foi acrescentado o teste de software. A Figura 7.4 apresenta as escolhas da *startup* referente a este cartão.

### Análise das categorias por etapa

#### Ideação



Figura 7.4 – Categorias mais importantes - *Startup N* (Fonte: Autor)

Na ideiação, requisitos foi a categoria que se manteve padrão para todas as 4 *startups*. Equipe se manteve importante para 3 *startups* nesta etapa. Apenas uma *startup* mencionou gerenciamento de projetos e arquitetura de software como importante na ideiação (Tabela 7.1).

Tabela 7.1 – Categorias importantes na fase de ideiação

	Startup K	Startup L	Startup M	Startup N
Gerenciamento de projetos				
Arquitetura de software				
Ciclo de desenvolvimento				
Teste de software				
Equipe				
Requisitos de software				

### Validação

Na validação, a equipe foi a categoria padrão para as 4 *startups*. Gerenciamento de projetos, arquitetura de software e ciclo de desenvolvimento foram citadas por 3 *startups*. Requisitos por 2 e teste de software por uma (Tabela 7.2).

### Tração

Na tração, o teste de software foi a categoria padrão para as 4 *startups*. Arquitetura de software e ciclo de desenvolvimento foram mencionadas por 3. Gerenciamento de projetos e Equipe por duas. A categoria requisitos não foi mencionada nesta etapa (Tabela 7.3).

### Escala

Na escala, apenas 3 *startups* elencaram as categorias. Teste de software e Arquitetura de Software foram padrão para estas. Ciclo de desenvolvimento foi mencionada por 2



Tabela 7.2 – Categorias importantes na fase de validação

	Startup K	Startup L	Startup M	Startup N
Gerenciamento de projetos				
Arquitetura de software				
Ciclo de desenvolvimento				
Teste de software				
Equipe				
Requisitos de software				

Tabela 7.3 – Categorias importantes na fase de tração

	Startup K	Startup L	Startup M	Startup N
Gerenciamento de projetos				
Arquitetura de software				
Ciclo de desenvolvimento				
Teste de software				
Equipe				
Requisitos de software				

e Equipe por uma *startup*. Gerenciamento de projetos e Requisitos não foram mencionadas nesta etapa (Tabela 7.4).

Tabela 7.4 – Categorias importantes na fase de escala

	Startup K	Startup L	Startup M	Startup N
Gerenciamento de projetos				
Arquitetura de software				
Ciclo de desenvolvimento				
Teste de software				
Equipe				
Requisitos de software				

### **Compilação e análise das categorias por *startup***

Todas as *startups* citaram todo o conjunto de categorias, porém elas foram bastante diversificadas na distribuição das etapas. Isto demonstra diferentes necessidades por cada *startup*.

Cada *startup* teve algum tipo de categoria que foi mais mencionada no processo de amadurecimento da *startup*, conforme a seguir (Figura 7.5).

- Startup K: Arquitetura de software
- Startup L: Arquitetura de software
- Startup M: Equipe

- Startup N: Gerenciamento de projetos

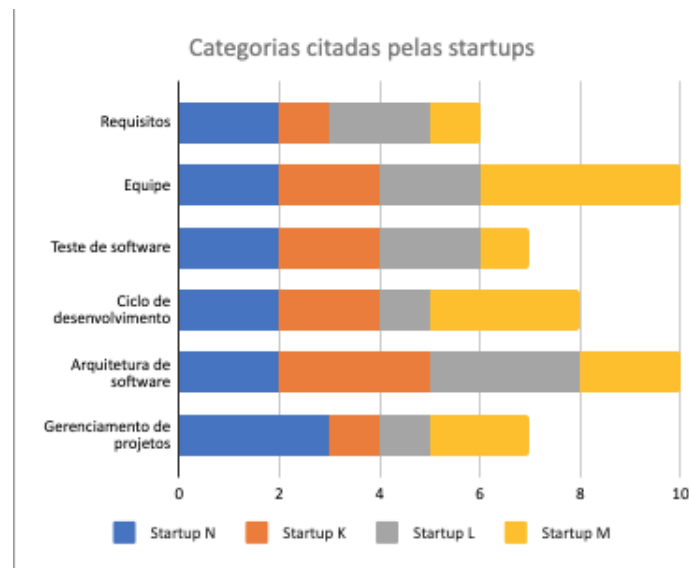


Figura 7.5 – Categorias Citadas pelas *Startups* (Fonte: Autor)

Do total, as categorias mais citadas nas etapas pelos participantes foram arquitetura de software e equipe, ambas com 10 menções. Em seguida, o ciclo de desenvolvimento, com 8 menções. Gerenciamento de projetos e teste de software, com 7 menções e requisitos 6. A Figura 7.6 apresenta as categorias citadas pelos participantes de acordo com o nível de maturidade.

Alguns pontos interessantes são apresentados nesta figura (Figura 7.6). Enquanto "Requisitos" é um assunto importante na fase de ideação, de acordo com os participantes, este assunto perde sua importância à medida que a *startup* evolui no processo de maturidade. Por outro lado, teste de software, arquitetura de software e ciclo de desenvolvimento ganham importância à medida que a *startup* ganha maturidade.

### 7.3.3 Cartão #3: Definição dos requisitos

- *Startup K*: A definição dos requisitos variou em todas as etapas. Na ideação, foram responsáveis o fundador (não técnico) e o UX designer. Na validação, o Cliente, o fundador (técnico) e o fundador (não técnico). Na tração, o Engenheiro de software, o Cliente e o UX designer.
- *Startup L*: Teve transições de definidores dos requisitos. Na ideação, apenas o fundador (não técnico) foi o responsável. Na validação, além do fundador (não técnico), o PO e o Cliente foram definidores. Já na escala, mantiveram-se o PO, o Cliente e entrou o Engenheiro de software como definidor.

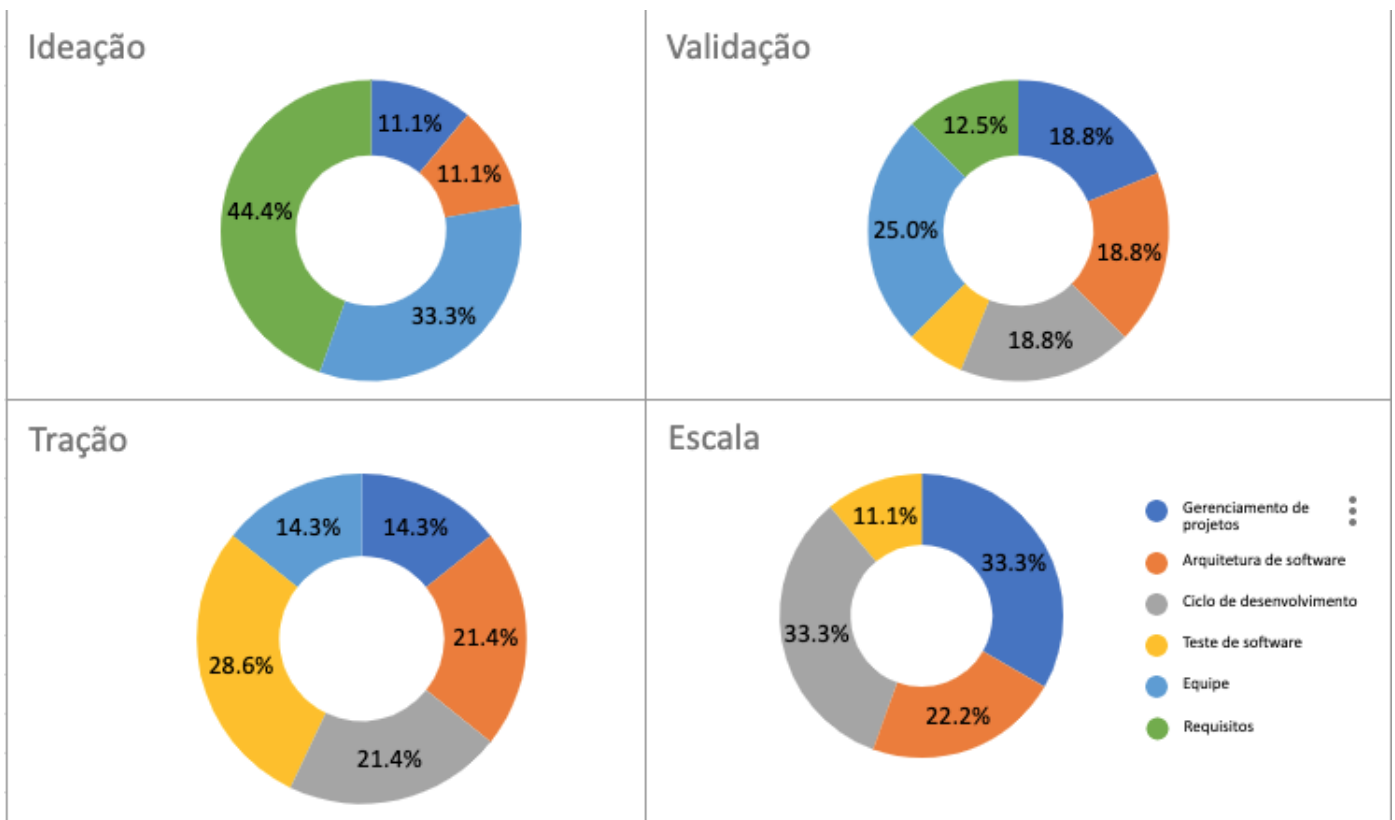


Figura 7.6 – Categorias Citadas pelas *Startups* por Fase de Maturidade (Fonte: Autor)

- *Startup M*: A definição dos requisitos teve maior variação nas fases iniciais. Na ideiação, o fundador (técnico) e o fundador (não técnico) foram os definidores. Na validação, o Cliente e o fundador (não técnico). Na tração e na escala, o Cliente e o PO tiveram este papel de definição.
- *Startup N*: a definição dos requisitos se manteve uniforme, desde a ideiação até a tração, contando com fundador (técnico), fundador (não técnico) e Cliente.

### Compilação e análise dos definidores de requisitos por etapa de maturidade

#### Ideação

O fundador técnico e o fundador não técnico fizeram parte dos definidores de requisitos para 75% das *startups*. O Cliente e o UX designer fizeram parte dos definidores de requisitos para 25% das *startups*. Engenheiro de software e PO não foram definidores de requisitos para as *startups* nesta etapa (Figura 7.7).

#### Validação

O Cliente foi definidor para 100% das *startups* na etapa de validação. Tanto o fundador técnico como o fundador não técnico e o *UX Designer* foram definidores de requisitos para 75% das *startups* nesta etapa. O PO foi definidor para 25% das *startups* na Validação (Figura 7.8).

#### Tração

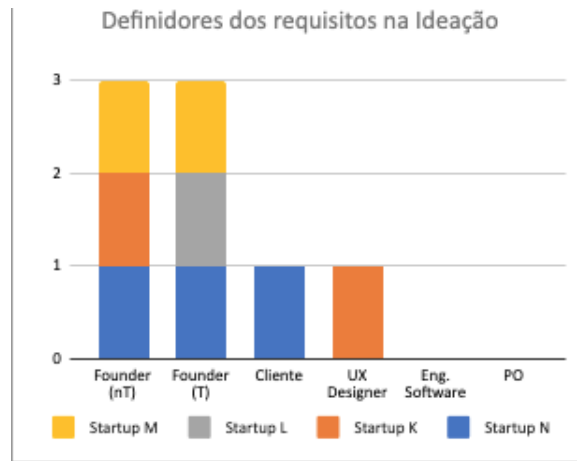


Figura 7.7 – Definidores de requisitos - Fase Ideação (Fonte: Autor)

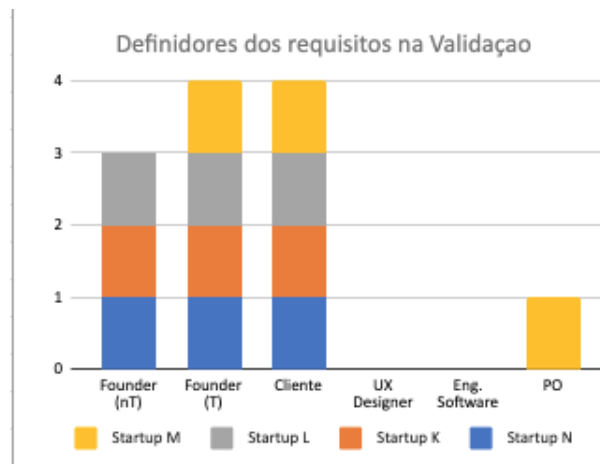


Figura 7.8 – Definidores de requisitos - Fase Validação (Fonte: Autor)

Na tração, o cliente foi definidor de requisitos para 100% das *startups*. PO foi definidor para 50% e o fundador técnico para 75%. *UX Designer*, engenheiro de software e fundador não técnico, para 25% (Figura 7.9).

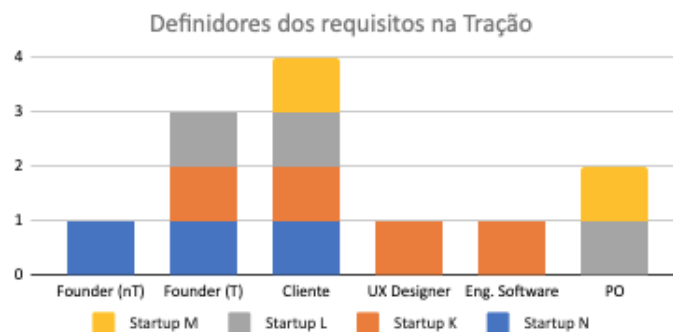


Figura 7.9 – Definidores de requisitos - Fase Tração (Fonte: Autor)

### Escala

Na escala, o cliente e o PO foram definidores de requisitos para 50% das *startups*. O engenheiro de software, para 25% (Figura 7.10).

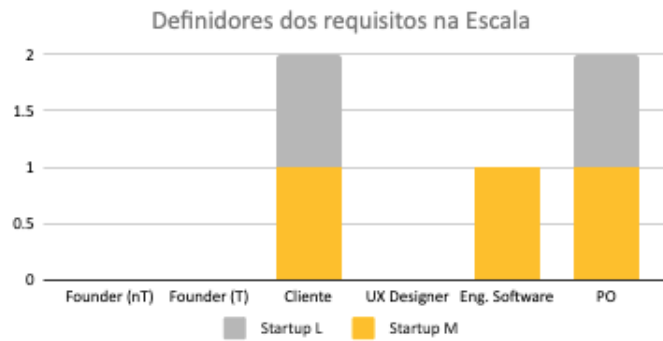


Figura 7.10 – Definidores de requisitos - Fase Escala (Fonte: Autor)

#### 7.3.4 Cartão #4: Alterações dos requisitos por etapa

*Startup K*: teve sua alteração mais significativa entre a validação e a tração. Tendo alterações pequenas na ideação e na validação.

*Startup L*: Teve suas principais alterações na validação e na tração. Na ideação, 3 pequenas alterações.

*Startup M*: Teve sua alteração mais significativa entre a validação e a tração. Na ideação e na validação, uma alteração pequena em cada etapa.

*Startup N*: Teve alterações mais significativas na ideação e na validação. Na validação também teve uma alteração média e na tração uma alteração pequena.

#### **Análise das alterações por etapa**

Visualização da soma das alterações das 4 *startups* por etapa. As maiores alterações ocorreram entre a ideação e a tração. Sendo a maior intensidade nas fases de validação para tração. Alterações menores ocorreram mais na ideação. Uma startup relatou alteração média na validação e uma relatou alteração pequena na tração.

#### 7.3.5 Cartão #5: Proporção da equipe de TI por etapa

*Startup K*: Na ideação não teve equipe técnica. Na validação teve 25% e na tração 50%. Não menciona o cenário para a fase de escala (Figura 7.11).

*Startup L*: começou com 50% de equipe técnica na ideação. Na validação e na tração teve um aumento para 75%. Na escala voltou para 50% (Figura 7.12).

*Startup M*: Começou com 75% de equipe técnica na ideação e assim seguiu na validação. Na tração aumentou para 50% e na escala para 25% (Figura 7.13).

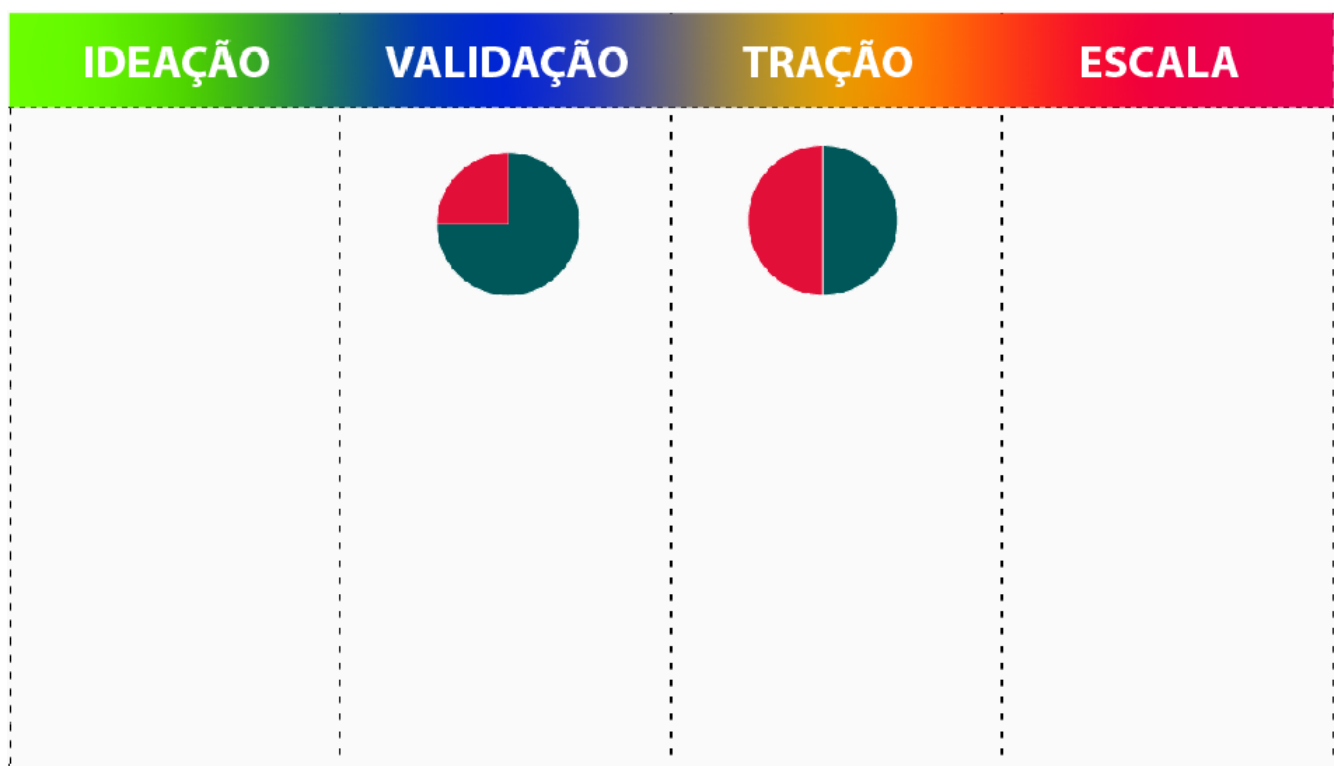


Figura 7.11 – Equipe de TI - Startup K (Fonte: Autor)

*Startup N*: Começou com 25% de equipe técnica na ideação. Na validação e na tração aumentou para 75%. Não menciona o cenário para a fase de escala (Figura 7.14).

### **Análise da proporção da equipe de TI por etapa**

A visualização de dados, com base no *card sorting*, mostra a evolução da equipe de TI durante as etapas das *startups* (Figura 7.15). Cada linha da Figura 7.15 indica a evolução da equipe de TI em termos de quantidade de profissionais de cada uma das *startups* participantes do grupo focal.

Importante analisar que das 4 *startups*, somente uma não tinha profissionais de TI na fase de ideação. Por outro lado, a fase de validação e tração mostrou-se como as fases de maior importância e investimentos na equipe de tecnologia das *startups*. Se compararmos com a visão dos empreendedores nas diversas etapas de maturidade, mostrado na Figura 7.6, pode-se perceber que este incremento na equipe de TI está associado a uma preocupação maior em relação a atividades de definição da arquitetura de software e testes de software.

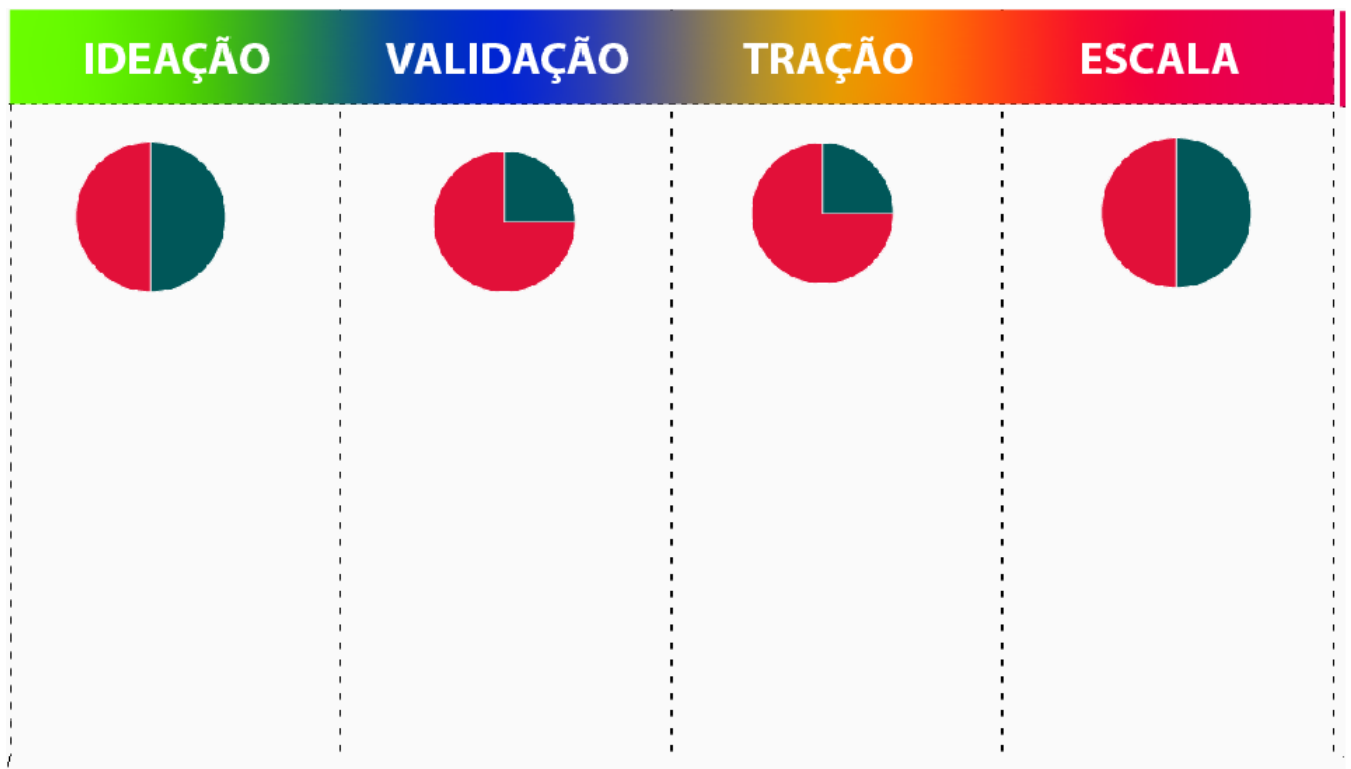


Figura 7.12 – Equipe de TI - *Startup L* (Fonte: Autor)

#### 7.4 Grupo Focal

Após a conclusão da classificação dos cartões, fez-se o uso de um grupo focal, que é uma metodologia estabelecida comumente utilizada nas ciências sociais [Mor96]. Na Engenharia de Software, Kontio et al. [KLB04] referem-se aos grupos de foco como um método rápido e de baixo custo para obter informações qualitativas. Para esta pesquisa, o grupo focal foi escolhido por ser exploratório por natureza [THB10] e com isso não somente o que as pessoas têm a dizer, mas fornecer percepções sobre as fontes de comportamentos e motivações complexos [MK<sup>+</sup>93].

As respostas foram transcritas e transformadas em tópicos, contendo os principais pontos mencionados pelos participantes. Estes tópicos foram categorizados, identificando as 6 (seis) categorias identificadas a priori, além de 5 (cinco) categorias identificadas posteriormente, durante a análise dos dados :

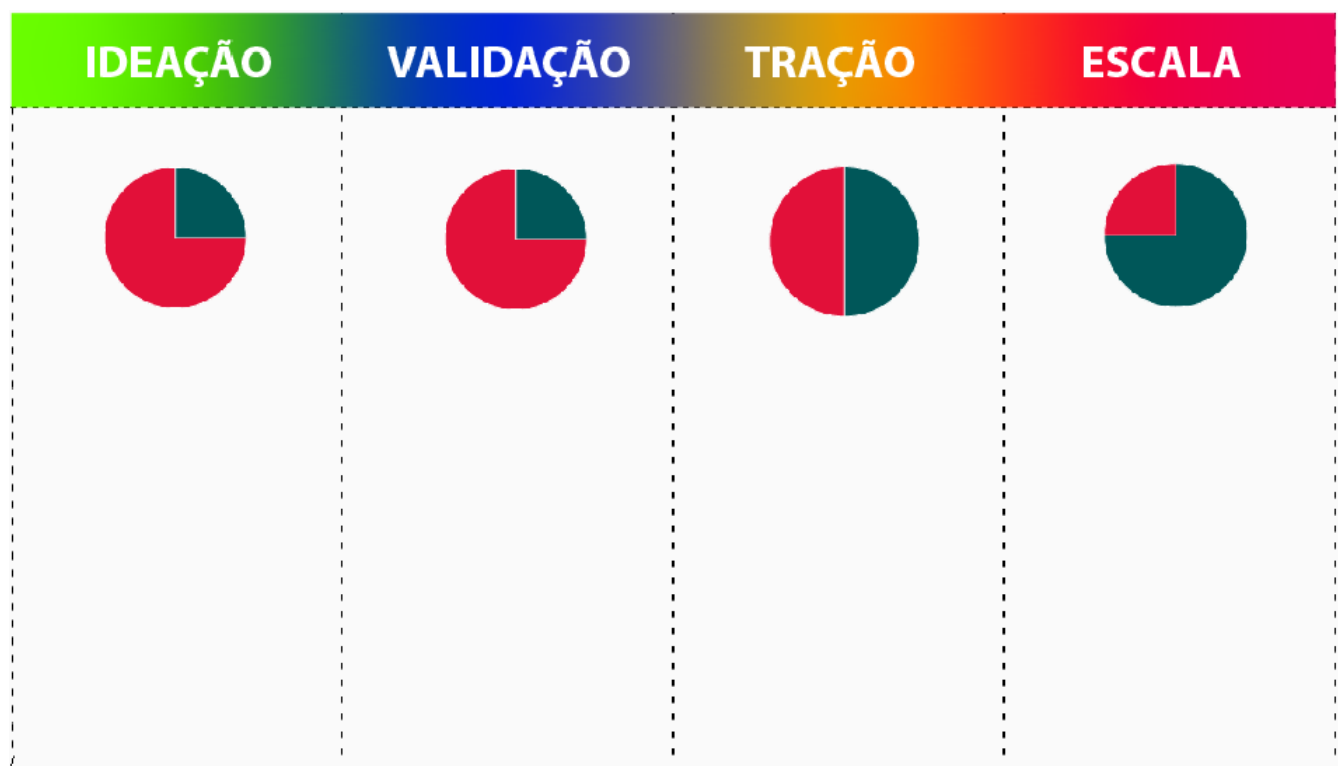


Figura 7.13 – Equipe de TI - *Startup M* (Fonte: Autor)

- Teste de software
- Equipe
- Gerenciamento de projetos
- Arquitetura de software
- Ciclo de desenvolvimento
- Requisitos
- Softwares utilizados (novo)
- Experiências anteriores (novo)
- Negócio (novo)
- Aprendizado (novo)



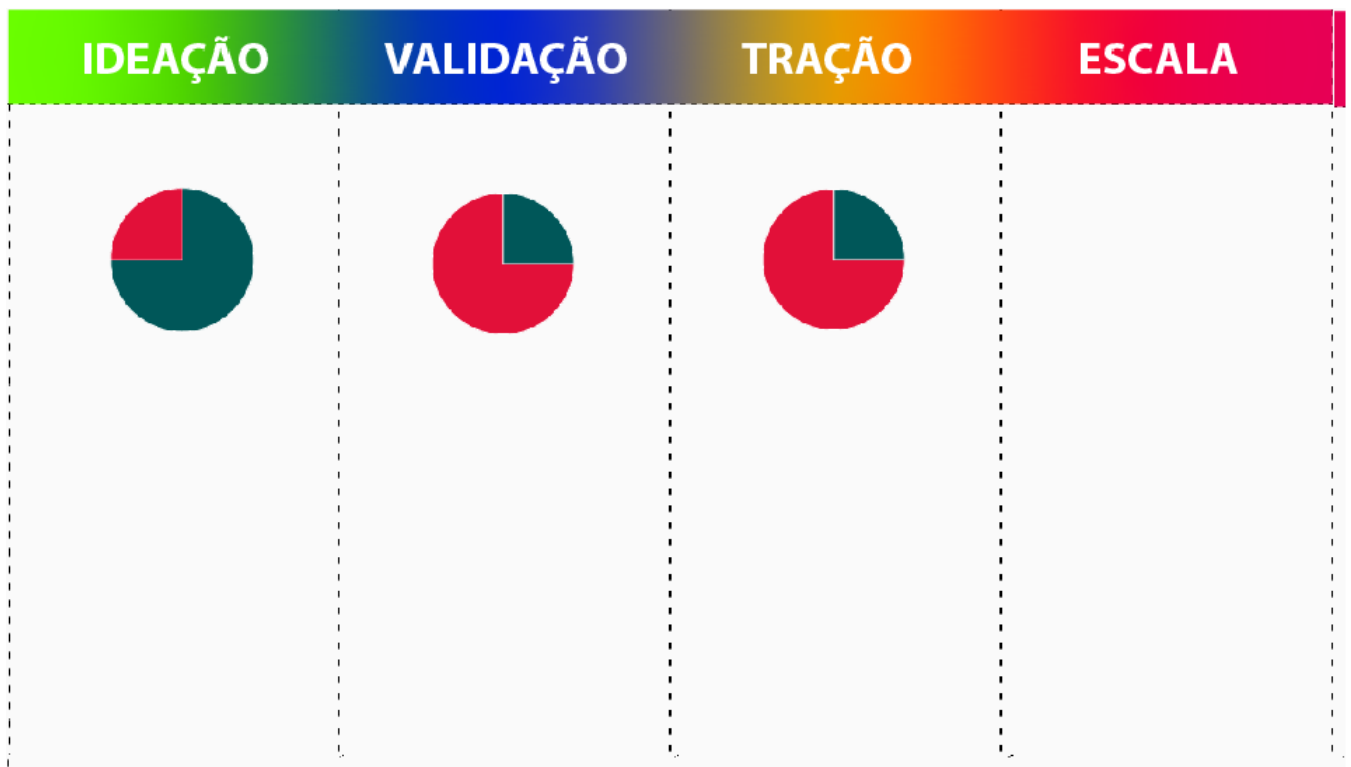


Figura 7.14 – Equipe de TI - *Startup N* (Fonte: Autor)

- Pandemia (novo)

A análise busca compreender as trocas entre os participantes, indo de consensos e dissensos, a crenças e perspectivas.

A análise de conteúdo é um procedimento clássico para analisar materiais de texto de qualquer origem, de produtos da mídia a dados de entrevistas [Fli12]. É um método baseado no uso de categorias derivadas de modelos teóricos. Normalmente, aplicam-se estas categorias aos textos, em vez de desenvolvê-las a partir do próprio material - embora evidentemente se possa examinar as categorias à luz dos textos em análise. A análise de conteúdo tem por objetivo classificar o conteúdo dos textos alocando as declarações, sentenças ou palavras a um sistema de categorias [Fli12].

A seguir é realizada uma análise do grupo focal de acordo com o protocolo inicial utilizado

**Questão 1: Como foi a jornada inicial da *startup* na definição do que deveria ser e o que não deveria ser desenvolvido?**

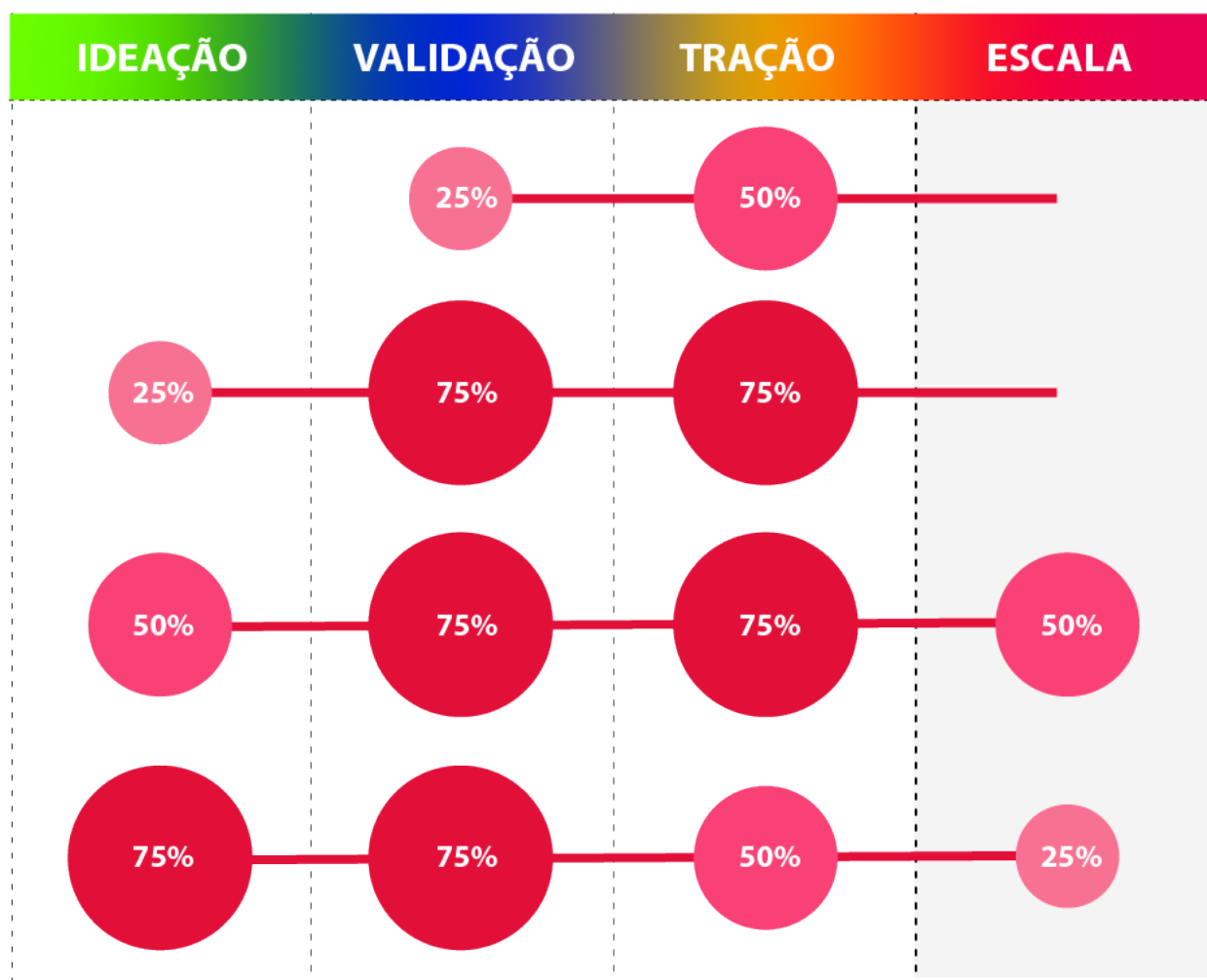


Figura 7.15 – Equipe de TI - Visão Geral (Fonte: Autor)

### **Startup K**

Os fundadores têm perfil de negócios. Na ideação, precisou reforço de marketing para validar teses (hipóteses) da área de seguros. Criou o MVP para facilmente endereçar as dores dos clientes e facilitar a interação com o sistema. Usou basicamente serviços oferecidos gratuitamente e teve custos somente com o projeto do website e uma licença de um produto de software. Passaram quase 1 ano com o MVP, com tentativas e erros, e isto mostrou que o produto tinha aderência no mercado. No momento de tomar as decisões do que implementar, já com sócio da área de tecnologia, ficou mais fácil.

A empresa precisava ser rentável desde o primeiro dia, para que os fundadores se dedicassem a ela. Não caíram na armadilha de criar um produto Pegaram um produto existente, para entregar de forma diferente. Para monetizar e testar as validações era mais fácil. Precisam tomar decisões agora do que vai dar mais escala em cima do mesmo produto. A proposta é conceber novos produtos a partir de um posicionamento seguro. Utilizaram esta estratégia para monetizar e testar as hipóteses de negócio mais rapidamente.

### **Startup L**

Os empreendedores possuíam uma *startup* anterior que lhes deu uma experiência para indicar o que não deveria ser feito. O propósito do novo negócio surgiu pelo relacionamento que eles possuíam com um profissional da área da saúde e esta demanda (a de gerenciar escalas) foi foco de uma pesquisa mais profunda até descobrirem que a maioria dos setores que realizam este tipo de serviço, o realizam de forma manual, sem uma solução computacional. O primeiro passo foi entender onde estava o problema e onde estava consumindo tempo e "roubando" produtividade dos gestores e desenvolver algo neste ponto. Criaram várias hipóteses e iniciaram os testes. Com o primeiro MVP, entraram em contato com um hospital que disponibilizou as informações para que eles pudessem executar o MVP e testar algumas hipóteses. O MVP era um *script* que gerava séries de dados em excel. Gerava escalas que um gestor levava 15 dias fazendo, em alguns minutos. Na *startup* anterior eles desenvolviam antes e depois validavam e então aprenderam que o método não funciona. Resolveram fazer o oposto. Hoje tem um processo muito mais definido com uma equipe de 20 pessoas, sendo a maioria da área de tecnologia da informação. Utilizam uma metodologia ágil e conseguem entregar funcionalidades muito fácil para o cliente. A cada 15 dias ele recebe uma *news* com o que tem de novo. Sempre trabalham com conceito de MVP. Cada funcionalidade nova é planejada para conseguir se resolver da forma mais rápida com o emprego do mínimo de esforços. Há um contato muito próximo do cliente para a parte de criação do produto. Há um ciclo rápido de validação, fazendo "*inceptions*" e dinâmicas de levantamento de requisitos - questionários, etc., para medir o impacto do que pode causar de efeito.

### **Startup M**

Decisão inicial partiu dos fundadores e a partir disto começaram a validar. Com os primeiros indícios de validação, utilizaram os primeiros clientes como base para os primeiros requisitos. O *feedback* dos clientes foi fundamental para novas funcionalidades, ajustes e caminhos a seguir. A análise do que os clientes solicitam e a observação do mercado indicam onde a *startup* emprega seus esforços para o desenvolvimento de novas funcionalidades. A equipe de tecnologia de informação é a mais representativa e com isso apostam no desenvolvimento inteiro das funcionalidades e dos produtos para então verificar se o mercado mostra aceitação.

### **Startup N**

Tiveram uma conversa inicial entre pessoas técnicas e não técnicas para discutir tudo que queriam fazer. Em um momento posterior, entrou um sócio desenvolvedor e a partir disto elaboraram o MVP. Este primeiro MVP demorou 1 (um) ano para ser desenvolvido. Na época, não tinham nenhuma orientação, como o uso de um modelo criado em um *Business Model Canvas*, optaram por algo bem simples de desenvolver desde os primeiros meses do negócio.

### **Análise da questão 1**

**REQUISITOS:** A proposta inicial surgiu dos fundadores (técnicos e não técnicos). Todos começaram com um MVP para validar o produto. Apenas uma *startup* não tinha fundador técnico, mas contou com apoio comercial para a validação.

Atualmente, para 3 (três) das *startups*, as definições dos requisitos são feitas em conjunto, integrando fundadores, membros como POs e clientes, que ganharam destaque durante a validação. Para uma delas, as definições partem somente do fundador não técnico.

**CICLO DE DESENVOLVIMENTO:** Apenas uma *startup* partiu para o desenvolvimento do produto, as demais elaboraram um MVP para validação junto a seus clientes / consumidores antes. Algumas mencionam que isto foi um aprendizado de experiências anteriores.

**EQUIPE:** Inicialmente, as equipes das 4 (quatro) *startups* foram bastante diversificadas. 3 (três) tinham fundadores técnicos, mas em proporções diferentes em relação ao restante da equipe.

**GERENCIAMENTO DE PROJETOS:** Uma das *startups* conta que evoluiu para um processo mais estruturado, de metodologia ágil, sempre se comunicando com o cliente.

**APRENDIZADO:** Para as *startups* que elaboraram o MVP antes do produto, foi uma questão de aprendizado anterior.

**Questão 2: Qual foi a evolução do time da *startup*, do *day ONE* até o dia de hoje, em termos de quantidade, perfil, responsabilidades? Pergunta complementar:**

**Todas as definições de requisitos são dos sócios no momento atual? Pergunta complementar:**

**Como as equipes se organizam em relação a isso? O que é prioridade para desenvolverem ou não? Pergunta complementar: Qual software utilizam?**

**Pergunta complementar: Que ferramentas usam para substituir o *board*?**

### ***Startup K***

No início da operação, focaram muito na equipe comercial. A solução é uma máquina de vendas, então foram adicionados elementos que dessem escala para a equipe. Conforme conseguiram dar escala para a equipe, começaram na época com a entrada de um sócio da área de tecnologia. Com a experiência deste novo sócio, a experiência com tecnologia e negócios foi possível evoluir no produto e atrair novos talentos para a equipe de desenvolvimento. Atualmente a equipe possui dois profissionais mais experientes e os demais juniores. A atuação do PO é responsabilidade do sócio não técnico. Atualmente, estão mais preocupados em ter alguém que traduza o que pode ser oportunidade de negócio vindo do mercado do que entrar no detalhe micro do produto. Todas as definições são

feitas exclusivamente pelos sócios. Usam a ferramenta *Club House* para fazer o controle dos projetos, tanto dos *stories* como dos *epics*. Para a equipe de tecnologia, fazem *daily* e *plannings* quinzenais para aquilo que é estabelecido pelos sócios. Ainda não tem a figura plena do PO. É a visão dos sócios e se prioriza ao longo do tempo. A pandemia não alterou muito as entregas. Por ter um sócio técnico focado em processos o uso de ferramentas como o Trello facilitou a adaptação da empresa no momento da pandemia.

### **Startup L**

O negócio anterior dos fundadores desta *startup* serviu para mostrar o que não fazer. Começou com o fundador técnico que reuniu alguns amigos/conhecidos não técnicos. No início acreditavam na ideia do que poderia dar certo, não tinha dinheiro mas sim pessoas que acreditaram nessa ideia. Trocaram um percentual da empresa para que novos sócios pudessem ajudar no desenvolvimento do produto. Com a equipe multidisciplinar de fundadores, começaram a criação do produto. Tiveram facilidade para conseguir o investimento. Segundo um dos fundadores: "No início cada dólar é mais caro, dá muito mais trabalho ganhar dinheiro. Com a validação do negócio é mais fácil, o dinheiro se torna barato."

No início só contratavam serviços terceirizados, trabalhos esporádicos de design e outros, mas com o crescimento do produto, obrigaram-se a internalizar estes serviços. Começaram a contratar pessoas da parte de produto, desenvolvedores e designers, foco apenas sêniores. Como estavam criando algo novo, que não tinha onde se basear, precisavam de experiência. Este ainda é o foco e pagam um salário acima de mercado. Neste momento estão mais focados na parte comercial, suporte e *customer success*.

Fazem isso mais ou menos em parceria. Realizam muitas pesquisas com cliente para entender onde existe a "dor do cliente" e tentam compilar a informação identificando o melhor caminho. A PO faz o trabalho de acompanhamento direto com os clientes.

Trabalham com metodologia ágil, com reuniões quinzenais em que fazem o planejamento. É passado para toda equipe as funcionalidades pensadas. Nas cerimônias da metodologia tem *review* para ver o que não funcionou ou deu errado na *sprint* anterior. Na *planning* tentam passar as prioridades, se tem algo afetando algum cliente ou se vão trabalhar em outro mercado. Durante o período da *sprint*, fazem reuniões, as *daily meetings*, para o acompanhamento. Utilizam muito software de gerência de projeto para esta atividade.

Utilizam ferramentas como o ClickUp e o GitHub. Isto auxilia o trabalho remoto, necessário durante o período da pandemia.

### **Startup M**

Sempre investiram muito no time técnico e no suporte. Desde os primeiros clientes adotam metodologia de *customer success*. Por trabalharem com clientes de porte médio e grande, conseguiram dar atenção a cada conta. De acordo com o fundador: "... não precisa

trabalhar com suporte em escala, conseguimos trabalhar bem próximos aos clientes. E isso é importante."

No time de desenvolvimento, sempre trabalharam com uma *squad* mínima. Começaram com desenvolvedor júnior e estagiário, e isso foi um erro, segundo o fundador. Deveriam ter começado com pleno e sênior e ir introduzindo estagiários ao longo do tempo. Hoje o time está mais pleno para sênior e este é o foco de contratação. O papel de PO é bastante focado nos fundadores, que analisavam os requisitos junto aos clientes. Sendo um fundador da área de design, a parte de *User eXperience* (UX) sempre foi algo existente na *startup*. Atualmente tem a parte de gestão do produto, responsável pela parte de refinamento, que é capturar a ideia básica e detalhar em requisitos para desenvolvimento. Desta forma consideram que a *squad* esteja no formato ideal, composto pelo UX designer e gestor de produto.

Buscam juntar vários tipos de demandas. Algumas são requisitos de clientes que geralmente são melhorias incrementais.

Tem requisitos estratégicas, de visão do produto e onde se quer chegar. E tem alguns requisitos de estratégia da *startup*, focado no mercado que atuam ou nos concorrentes. Isto é responsabilidade do gestor de produto. A ideia é que os fundadores foquem mais nos objetivos do negócio e o gestor de produtos na tradução dos requisitos para serem integrados ao produto.

O gestor de produtos faz a priorização do que entra no Kanban e o que fica no *backlog*, além de acompanhar o ciclo de desenvolvimento. Com o *Kanban*, acompanham a capacidade atual do time e a quantidade de atividades em desenvolvimento. Utilizam a ferramenta Jira.

### **Startup N**

A equipe iniciou com 3 (três) sócios, sendo que apenas 1 (um) desenvolvedor. Após 3 (três) anos, contrataram 2 (dois) programadores: 1 (um) sênior e 1 (um) júnior. O design sempre foi terceirizado e com a automação de alguns processos diminuíram a necessidade de contratar mais pessoas. No momento estão em busca de pessoas para aumentar a equipe e estão com dificuldade para encontrar programadores, principalmente no estado de origem da *startup*.

Possuem um planejamento do que pretendem fazer a curto, médio e longo prazo. Porém, várias demandas são oriundas dos atuais clientes e, com isso, ocorrem várias alterações no planejamento da *startup*.

Na pandemia, por exemplo, pararam tudo e mudaram o que estavam fazendo por uma demanda específica de um cliente. Com isso, tiveram que se adaptar às novas demandas e ao interesse do consumidor que mudou. Existem projetos que querem fazer e que darão resultado a longo prazo, mas estão muito conectados ao que está acontecendo no momento.

A equipe é enxuta e utilizam o Trello para o controle das atividades e planejamento. Acreditam que inserir uma metodologia de desenvolvimento no atual momento da *startup* tende a atrapalhar o desempenho da equipe.

### **Análise da questão 2**

**EQUIPE:** As equipes começaram com formações variadas. A questão da senioridade da equipe surge durante a conversa. 3 (três) *startups* comentam como a senioridade é importante para atingir melhores resultados e evitar retrabalho. Porém, no início optaram por profissionais juniores e foram exigindo senioridade ao longo do tempo. Questões técnicas também foram mencionadas como impacto nas equipes, incluindo a escolha da tecnologia. Uma startup tem dificuldades em encontrar profissionais pela sua localização. Foco no suporte ao cliente é uma questão mencionada por 2 (duas) das *startups*. A função comercial na equipe foi mencionado, mas para a maioria surge após a *startup* começar a crescer.

**REQUISITOS:** Os requisitos no geral são definidos pelos sócios. 3 (três) das *startups* dividem esta definição com clientes. Uma delas cita que a pandemia do covid-19 impulsionou novos comportamentos em seus usuários/clientes, sendo necessário novas definições com base neste fator.

**GERENCIAMENTO DE PROJETOS:** Quanto à organização para atender as demandas e prioridades, as metodologias são variadas, assim como os softwares e ferramentas, conforme mostra a Tabela 7.5:

Tabela 7.5 – Relação entre metodologia e software utilizados pelas *startups*

<b>Metodologia</b>	<b>Software</b>
Startup L trabalha com metodologia ágil, incluindo cerimônias com plannings, reviews, retrôs das sprints, além de dailys de acompanhamento.	Click up (principal) GitLab
Startup M utiliza o board de kanban (JIRA), gerenciado pelo gestor de produtos, fazendo o planning, retros e reviews, como se fosse uma stand up.	Jira Product board (anterior, apenas testaram) Diamond D (anterior, apenas testaram)
Startup K faz controle dos projetos, stories e epics quinzenais e dailys para a tecnologia.	Trello (anterior) Club house (atual)
Startup N acredita que uma metodologia poderia atrapalhar no momento.	Trello

**Questão 3: Sobre as escolhas de tecnologia para a *startup*, o que deu certo, o que deu errado e o que ainda precisa ser feito?**

### **Startup K**

Optaram por linguagens mais consolidadas, fáceis para contratar pessoas, principalmente na região onde estão inseridos. Se veem como conservadores. Fizeram iniciativas com linguagens de programação novas. Testaram uma estratégia de um aplicativo com nova tecnologia, mas serviu somente para aprendizado. Consideram que por serem conservadores no desenvolvimento do produto, a manutenção é mais trabalhosa, gerando atrasos em algumas entregas.

### **Startup L**

O foco é escalar. Fizeram pesquisa baseada em tecnologias que o fundador técnico dominava. Com a experiência do fundador técnico, foi construída uma *application programming interface* (API) para testar a ideia no mercado. Pensaram inicialmente em utilizar o que consideravam a tecnologia ideal, mas pelo fato da dificuldade em encontrar profissionais resolveram focar em tecnologias mais consolidadas e que tinham mais facilidade em encontrar profissionais. Para novos integrantes possuem um filtro e entrevistam entre 10 (dez) a 15 (quinze) pessoas antes de fazer a escolha. O foco está na senioridade. Em relação às atividades da *startup*, estão focando nos testes de software. Sempre que possível, agregam novas tecnologias no produto e estão preocupados com segurança da informação.

### **Startup M**

No início não teve muita discussão sobre a tecnologia. Foi muito da experiência que já tinham e fazia sentido para determinado cenário.

Foram conservadores na escolha da tecnologia com o foco em desenvolver rapidamente o produto. Atualmente a tecnologia utilizada não é muito diferente da que foi utilizada na concepção do produto, porém algumas novas tecnologias foram agregadas, evoluindo a arquitetura da solução. Fazem um trabalho constante de avaliação geral para entender quais são os gargalos que tem de infraestrutura e escala.

### **Startup N**

Acham comum, no começo, o responsável pela tecnologia utilizar seus conhecimentos e experiências prévias na concepção do novo produto. Começaram com uma linguagem de programação específica e depois tiveram vários problemas. Devido a isso, mudaram o produto inteiro 3 (três) vezes. Utilizam alguns robôs que capturam os preços em tempo real e, neste ponto, tiveram problemas por não possuírem o conhecimento técnico internalizado.

## **Análise da questão 3**

### **ARQUITETURA DE SOFTWARE:**

A experiência dos fundadores foi mencionada por 3 (três) *startups* com relação ao uso de tecnologias. Na visão das *startups*, o que funcionou em relação a escolha das tecno-



logias foi, para 2 (duas) *startups*, escolher tecnologias mais consolidadas, que facilitassem trazer profissionais qualificados.

A construção prévia de uma API foi mencionada por 2 (duas) *startups*. De forma geral, as tecnologias são reaproveitadas, e o código não é reescrito.

## 7.5 Síntese do capítulo

Este capítulo apresentou o resultado obtido da realização de um grupo focal com 4 *startups*, precedido de uma classificação de cartões. O objetivo principal era validar premissas do modelo de adoção das práticas de Engenharia de Software por startups apresentado no capítulo anterior.

Com base nas respostas das *startups* aos cartões e as discussões realizadas durante o grupo focal conseguiu-se entender com maior clareza alguns pontos importantes do modelo e realizar ajustes importantes no mesmo. Estas análises podem ser resumidas da seguinte forma:

- **Equipe de desenvolvimento** : Verificou-se um incremento na importância da equipe de TI nas fases de validação e tração. A base da equipe nas fases iniciais, principalmente na fase de ideação, era baseada no grupo de fundadores. Apostam na senioridade da equipe como diferencial para uma melhor entrega final. Uma definição clara foi a evolução da equipe de acordo com a evolução do negócio da *startup*.
- **Requisitos de software** : Requisitos de software se mostraram importantes na fase de ideação e, mesmo não sendo citada pelos empreendedores na classificação de cartões, a importância do requisito de software foi apresentado em discussões sobre o tema durante o grupo focal. A evolução da necessidade de um melhor gerenciamento dos requisitos está associado a própria evolução do negócio da *startup*.
- **Teste de software** : Práticas de teste de software são apresentadas de forma totalmente informal e ad hoc na fase de validação e evolui no formalismo e na adoção de técnicas e ferramentas nas fases de tração escala.
- **Arquitetura de software** : Decisões tecnológicas são assumidas pelos fundadores, principalmente por aqueles que possuem uma formação técnica. Na fase de validação a arquitetura da solução da *startup* é criada de forma informal, normalmente baseada na experiência do fundador e com componentes pré-existentes. Com o negócio crescendo e tendo suas hipóteses validadas, a arquitetura do software normalmente é recriada e baseada em empresas semelhantes. E na fase de escala a preocupação das decisões tecnológicas ficam focadas nas necessidades do negócio.

- **Gerenciamento de projetos** : Apesar do gerenciamento do projeto estar na preocupação dos empreendedores durante todas as fases, conforme foi apresentado na dinâmica da classificação dos cartões, ele é executado de forma totalmente informal e baseado na confiança na equipe na fase de ideação. Na fase de validação começam a ser definidos papéis para este fim, principalmente com a presença do fundador que faz o papel de PO. Somente nas fases de tração e escala a gestão de projetos possui um maior formalismo para o planejamento e monitoramento dos projetos e, também, inicia-se o uso de ferramentas de gestão de projetos.
- **Processo de desenvolvimento** : Durante a fase de ideação as *startups* não possuem nenhum processo de desenvolvimento. Na fase de validação, devido a velocidade de entregas e mudanças dos requisitos para a validação das hipóteses, o processo de desenvolvimento é informal e muitas vezes desorganizado. Somente nas fases de tração e escala o uso de ferramentas e um formalismo nas etapas de desenvolvimento são empregados pela *startup*.

Estas descobertas ajudaram na avaliação do modelo empírico da adoção de práticas de Engenharia de Software de acordo com o crescimento das *startups* de software, que será apresentado no próximo capítulo.

## 8. MODELO PRESS - PRÁTICAS DE ENGENHARIA DE SOFTWARE PARA *STARTUP*

Nos capítulos anteriores foram destacados conceitos importantes e adoção de práticas pelas *startups*. Sabendo que uma *startup* é um grupo de pessoas que trabalha com incertezas, seja de mercado ou tecnológica e que tem alto potencial de escala, principalmente pelo uso intensivo de tecnologia, ter fases de desenvolvimento do negócio bem definidas é algo importante para a longevidade do negócio. O IBGC propôs um modelo de governança corporativa para *startups*, com fases de maturidade e elementos que precisam estar presentes, tais como estratégia e sociedade, pessoas e recursos, tecnologia e propriedade intelectual e processos e *accountability* [LAM19]. Estes elementos são essenciais para que a *startup* possa crescer e se transformar de projeto para um negócio e uma organização de fato.

Por outro lado, as *startups* de software, que têm na tecnologia um dos principais diferenciais, têm como maior dificuldade a necessidade de identificar e utilizar adequadamente práticas de Engenharia de Software, de acordo com o nível de maturidade da *startup*. Se na ideação, validação, tração e escala existem preocupações relacionadas ao negócio, como por exemplo, definição do negócio, o acordo de sócios, a titularidade da propriedade intelectual, deve-se igualmente ter preocupações pelo lado do desenvolvimento de software. Se uma *startup* possui recursos escassos para validar suas hipóteses [Rie11], estes recursos devem ser bem aplicados também no desenvolvimento do produto.

Por este motivo, nesta tese propõe-se um modelo de adoção de práticas de Engenharia de Software de acordo com a maturidade da *startup*. Este modelo foi desenvolvido a partir de estudos empíricos de base qualitativa e quantitativa que contribuíram para chegar na versão apresentada neste capítulo.

Primeiramente, o modelo contempla a visão de maturidade a partir das seis categorias identificadas no estudo de campo apresentado no Capítulo 4. São elas:

- Equipe
- Requisitos de software
- Testes de software
- Arquitetura de software
- Gerenciamento de projetos
- Ciclo de desenvolvimento

Para cada um desses itens, foi analisado e definido um conjunto de critérios que indicam o nível de maturidade de aplicação das práticas pelas *startups* em cada uma destas 6 categorias (Figura 8.1).

A Tabela 8.1 apresentou a relação dos eixos de Engenharia de Software e o que é considerado em cada nível de maturidade definido. Ela foi construída como resultado do *survey* realizado, juntamente com os aprendizados obtidos no estudo de campo com *startups*. Estas definições foram avaliadas no grupo focal realizado.

	Inicial	Intermediário	Avançado	Maduro/Completo
Equipe	Equipes auto-gerenciadas	Equipes auto-gerenciadas Orientação ao negócio	Equipes auto-gerenciadas Orientação ao negócio Preocupação com as competências técnicas	Equipes auto-gerenciadas Orientação ao negócio Preocupação com as competências técnicas Preocupação com a complementariedade dos perfis
Requisitos	Foco no entendimento do negócio	Foco no mapeamento das dores do cliente/mercado em requisitos de software	Foco no mapeamento das dores do cliente/mercado em requisitos de software Definição de ferramentas e processo	Consolidação do processo e ferramentas
Teste de software	Testes ad hoc ou feitos pelo cliente	Testes de unidade	Testes funcionais executados pelo PO	Processo de testes de software Profissional ou equipe dedicada
Arquitetura de software	Modelo lego	Informal/desenvolvida pelo fundador	Arquitetura baseada no mercado	Arquitetura baseada no negócio
Gerenciamento de projetos	PO como gerente de projetos	Preocupação com planejamento das atividades	Preocupação com monitoramento e melhoria do processo	Uso de ferramentas Projetos em paralelo
Ciclo de desenvolvimento	Totalmente informal e não organizado	Processo de desenvolvimento minimamente definido (foco na programação) Uso de ferramentas de apoio a programação	Gerenciamento de configuração Processo de desenvolvimento definido	Uso de ferramentas em todas as etapas do processo de desenvolvimento

Tabela 8.1 – Definição dos níveis de maturidade das práticas

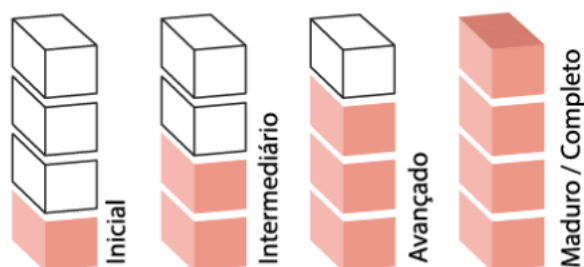


Figura 8.1 – Níveis de maturidade das práticas

A partir disso, foi feita a relação das práticas de Engenharia de Software com o grau de maturidade da *startup*, indicando os níveis a serem alcançados em cada um dos eixos.

Estes níveis foram escolhidos para manter uma paridade com o estudo do IBGC [LAM19] e com isso ter uma paridade entre o modelo proposto pelo IBGC e o modelo proposto por esta pesquisa, visando inclusive futura proposta de extensão do próprio modelo do IBGC para *startups* de software. A Figura 8.2 ilustra a forma como estes níveis são representados no modelo proposto.

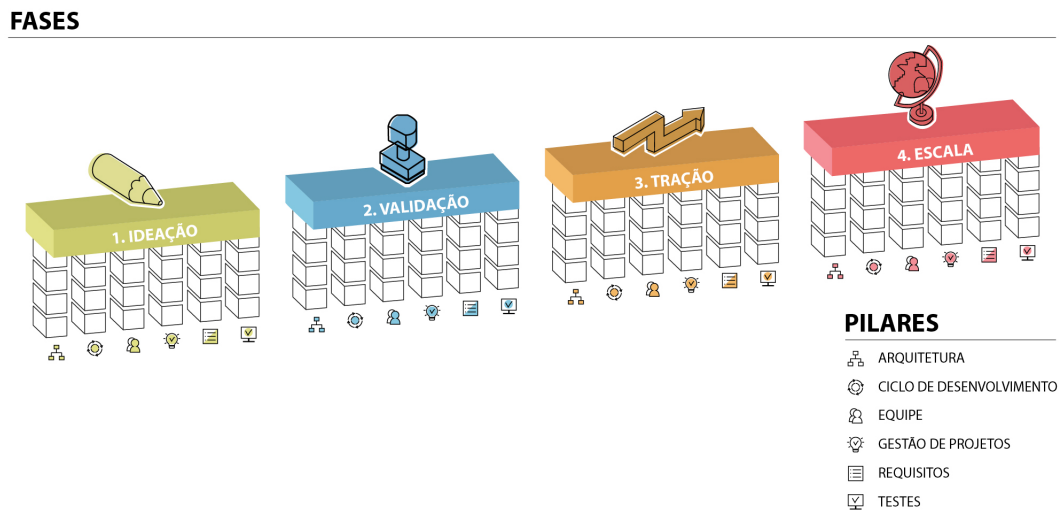


Figura 8.2 – Níveis representados no modelo proposto

## 8.1 Ideação

Na fase de ideação a *startup* ainda procura respostas do mercado para validar suas hipóteses. Está entre o desenvolvimento da ideia e o entendimento do problema que se propõe a resolver [LAM19]. No contexto de negócio, o empreendedor está em fase de alinhamento entre os fundadores e qual o propósito a *startup* terá. No contexto de Engenharia de Software, uma preocupação em ter uma equipe ou parte da equipe técnica para conseguir desenvolver algo que ajude na validação das hipóteses é parte constante dos primeiros passos do empreendimento.

De acordo com o analisado durante esta pesquisa, a equipe e os requisitos são pontos de atenção para estes novos empreendimentos. Na parte da equipe, por se tratar basicamente dos fundadores da *startup*, a preocupação é que mesmo que todos estejam iniciando as atividades empreendedoras, o alinhamento interno e a autogestão para que todas as atividades técnicas sejam executadas é essencial. Neste item, como a *startup* ainda está em uma fase embrionária, as discussões sobre definições do que será o projeto e o que será necessário, em termos técnicos, deve ser o foco da equipe.

Na questão dos requisitos, os empreendedores estão focados no que será o produto ou serviço. Nesta fase, os empreendedores analisam os requisitos de negócio, ainda sem muita preocupação com requisitos técnicos. O entendimento do mercado, das dores do cliente e de alternativas de como construir o MVP de forma mais rápida e mais barata é o objetivo desta fase. A Figura 8.3 apresenta a evolução destes pilares nesta fase de ideação.

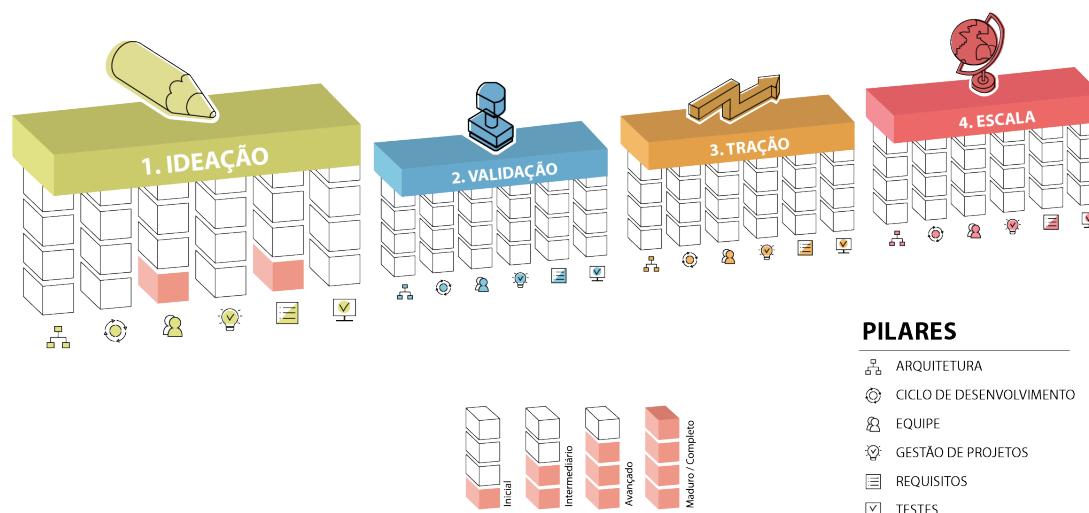


Figura 8.3 – Ideação

Uma análise do que deve ser o checklist do empreendedor nesta fase, tem-se:

- Definir responsabilidades entre os integrantes da equipe
- Discutir o objeto a ser trabalhado na construção do MVP
- Estabelecer mecanismos claros de comunicação entre os integrantes da equipe
- Definir como a análise das dores do cliente ou análise do mercado serão registradas

## 8.2 Validação

Na fase de validação o empreendedor possui o desafio de validar as hipóteses levantadas na fase anterior (ideação) [LAM19]. Segundo o IBGC [LAM19] nesta fase a *startup* já está formalizada e apta a receber aportes de recursos de terceiros, com o objetivo de construir o MVP que validarão as incertezas relacionadas ao negócio que está sendo criado.

Analisando como a Engenharia de Software atua nesta fase, práticas relacionadas às equipes de desenvolvimento, requisitos, testes de software, arquitetura de software,

gerenciamento de projetos e atividades do ciclo de desenvolvimento começam a serem implementadas pela *startup* (Figura 8.4).

Em relação à equipe de desenvolvimento, os integrantes continuam na abordagem de ser uma equipe auto-gerenciada, com forte atuação dos sócio fundadores. É nesta fase que novos integrantes começam a participar do desenvolvimento, como pode-se observar nos resultados apresentados do grupo focal. A orientação das ações está centrada no negócio e nas hipóteses levantadas na fase de ideação e o objetivo é entender como esta análise do negócio, que se transformou em uma hipótese a ser testada, se transforma em um requisito a ser utilizado na concepção do MVP ou dos MVPs construídos e utilizados pela *startup* para validar estas incertezas.

Isto está diretamente relacionado com a preocupação da *startup* em controlar estas descobertas sobre o que será o produto ou serviço que está sendo construído pela equipe. Nesta fase, o MVP que normalmente é trabalhado pelas *startups* tem uma construção de software na sua base e, com isso, a gestão destes requisitos começa a ser uma preocupação para a equipe de desenvolvimento. As primeiras ferramentas começam a ser utilizadas para melhor entender, priorizar e desenvolver os requisitos. Mas ainda não é algo incorporado na cultura da *startup*, ou seja, é o uso da experiência anterior dos profissionais que trabalham na *startup* que inserem as práticas de gestão de requisitos.

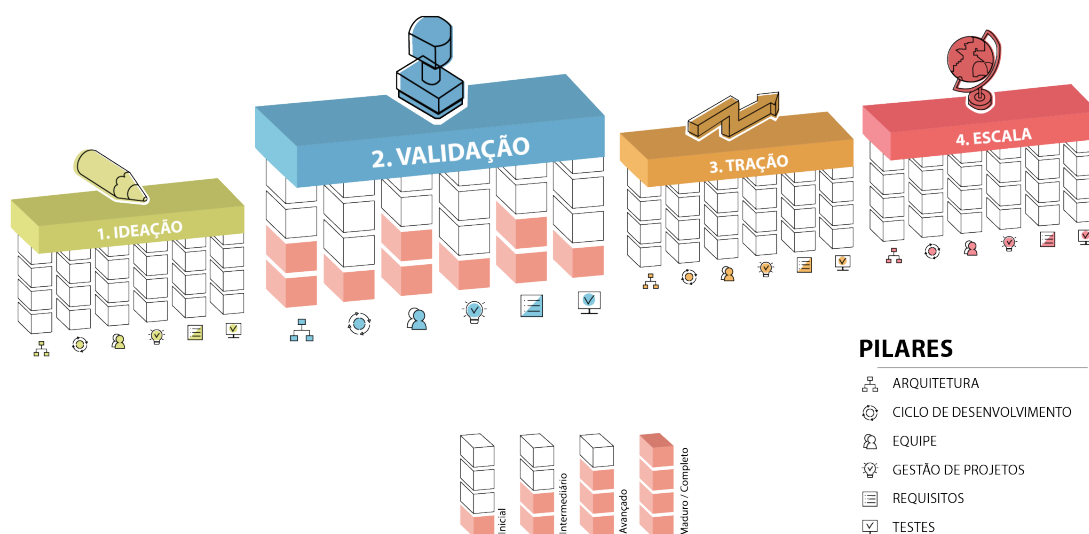


Figura 8.4 – Validação

As escolhas de tecnologia são iniciadas nesta fase, com o entendimento do negócio e a visão de quais características o MVP terá, a equipe de desenvolvimento estabelece as primeiras versões da arquitetura de software. Esta criação é realizada ou com elementos pré-construídos e a disposição no mercado, ou criado de forma rápida e informal por integrantes da equipe de desenvolvimento usando experiências e conhecimentos adquiridos em projetos executados no histórico profissional dos mesmos. Como a maioria das *startups* possui recursos escassos e o tempo para validar alguma hipótese no mercado é

escasso também, os empreendedores tomam a decisão de definir a arquitetura, não se preocupando com a robustez ou a certeza da tecnologia escolhida.

Estas questões relacionadas ao tempo, tem impacto também nos testes de software que são executados nesta fase. De forma informal, os testes são executados pela equipe que desenvolveu o MVP, de forma ad hoc e, em muitas situações, nenhum teste é realizado durante o desenvolvimento do MVP e sim por *early adopters*, que sinalizam melhorias a serem realizadas ou problemas encontrados no produto ou serviço disponibilizado.

Outra situação importante nesta fase, é a figura do gerente de projetos ser compartilhada com o papel do PO do projeto. E este busca informações da aceitação ou não do MVP e suas necessidades de mudanças ou melhorias. Esta centralização não afeta o desempenho do desenvolvimento, pois a *startup* ainda se encontra nos seus primeiros passos. E, nesta fase, o ciclo de desenvolvimento é totalmente informal e não organizado, muitas vezes centrado na experiência da equipe de desenvolvimento da *startup*.

Para o empreendedor é importante entender que práticas podem auxiliá-lo nesta fase de validação e, para isto ele deve se preocupar com:

- Existência de algum membro da equipe que tenha o conhecimento do negócio;
- Como se dará a transição da dor do cliente ou do mercado em requisito de software;
- Escolha de tecnologia que seja de fácil uso por parte da equipe e de fácil adaptação às mudanças necessárias para a validação do negócio;
- Como ter o PO gerenciando as atividades do projeto e definindo as prioridades de desenvolvimento.

### 8.3 Tração

Na fase de tração, a *startup* já possui o seu produto ou serviço validado pelo mercado e procura nesta fase a conquista de um maior número de clientes ou usuários e um aumento do faturamento [LAM19]. Com isto, algumas decisões realizadas pelos empreendedores precisam ser revistas, como por exemplo, a tecnologia escolhida ou a forma como o produto ou serviço foi criado, pois com o aumento de clientes e usuários e, conseqüentemente, a quantidade de informações uma solução mais robusta se fará necessária.

Nesta fase há grandes avanços relacionados à Engenharia de Software, visto que a *startup* validou o cliente, mercado e produto nas fases anteriores e com isso deve consolidar sua solução criando mecanismos para atrair e reter clientes, conforme é apresentado na Figura 8.5. Há um amadurecimento relacionado à equipe de desenvolvimento, requisitos, testes de software, arquitetura de software e no ciclo de desenvolvimento.



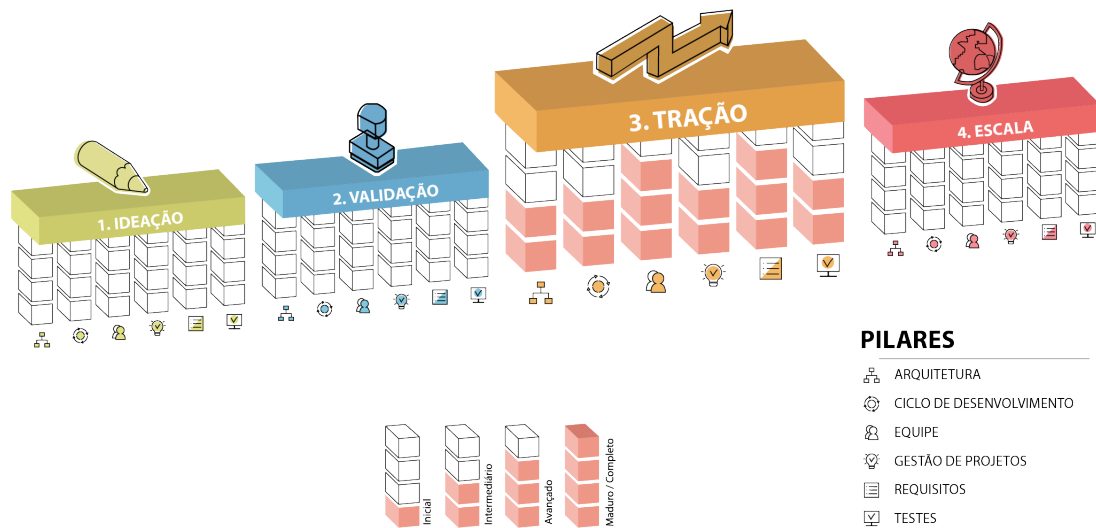


Figura 8.5 – Tração

A equipe de desenvolvimento deve, além de continuar a ser auto-gerenciada e possuir uma forte orientação ao negócio (todos os integrantes da equipe devem conhecer as nuances do mercado no qual a *startup* atua), começar a remodelar a equipe de acordo com as competências técnicas necessárias para o crescimento. Se antes a equipe podia escolher soluções pré-prontas que não atendessem 100% das necessidades do produto, simplesmente para validar alguma hipótese, neste momento as soluções apresentadas ao mercado devem estar 100% aderentes ao que se propõe.

Os requisitos de software devem ser melhor gerenciados por ter um aumento dos requisitos em razão da *startup* querer entregar uma solução mais completa do que nas fases anteriores. Isto faz com que a *startup* defina um processo de gestão de requisitos, bem como, a adesão de ferramentas para facilitar o trabalho da equipe e a gestão do projeto.

Em relação a qualidade, testes de unidade são implementados pelos programadores e os testes funcionais pelo PO. Desta forma, a *startup* inicia uma preocupação em ter no seu processo de desenvolvimento ações relativas a verificação e validação. Ainda não há um rigor na execução das atividades de testes de software, mas há um indicativo da necessidade, principalmente para que o produto ou serviço disponibilizado ao cliente ou usuário não apresente problemas na execução.

Um dos pontos que mais amadurecem em uma *startup* nesta fase é a arquitetura de software. Nas fases anteriores, a preocupação dos empreendedores era colocar algo para o cliente ou usuário com o intuito de testar as hipóteses de negócio. Nesta fase, a preocupação dos empreendedores é fazer com que a solução apresentada suporte o crescimento esperado, seja pela aquisição de novos clientes, seja pelo tráfego de uma quantidade maior de informações. Assim, o desenvolvimento de uma arquitetura de software baseada em soluções similares de mercado é o caminho escolhido pelas *startups*.

Na parte do ciclo de desenvolvimento, a formalização do processo e o uso de ferramentas se tornam essenciais para acompanhar o crescimento da equipe, o crescimento dos requisitos de software e o crescimento do negócio. Assim, uma boa gestão de configuração integrada a um processo de gestão e uso de ferramentas de desenvolvimento profissionaliza as atividades de desenvolvimento de software da *startup*.

Como *checklist* desta fase o empreendedor tem:

- A equipe de desenvolvimento deve possuir papéis técnicos que abranjam a solução proposta;
- Um processo de gestão de requisitos, bem como a adoção de ferramentas é algo importante nesta fase de crescimento;
- Adoção de testes de software, seja testes de unidade ou funcional, é extremamente importante para a entrega de valor ao cliente;
- A arquitetura de software deve ser robusta o suficiente para garantir o crescimento esperado pela *startup*;
- A gestão de projetos deve ser baseada em dados e para isto a adoção de processo e ferramentas é crucial;
- O ciclo de desenvolvimento deve ser formalizado para o melhor entendimento das responsabilidades de cada integrante da equipe e o uso de ferramentas, como ferramenta de gestão de configuração, se torna essencial para o bom andamento das atividades.

## 8.4 Escala

Na fase de escala, a *startup* está em transformação. Ou seja, é o momento em que a *startup* se transforma em uma empresa estabelecida [LAM19]. O crescimento escalável acelerado é esperado para este tipo de negócio e para isto os empreendedores devem explorar oportunidades de expansão do negócio em termos geográficos, de mercado ou produto [LAM19]. Desta forma, novos desafios são apresentados, tanto para o negócio da *startup* quanto para a equipe de desenvolvimento.

Todos os pilares relacionados à Engenharia de Software tem algum impacto e amadurecimento, conforme apresentado na Figura 8.6. A equipe de desenvolvimento tem que garantir o seu crescimento em número de pessoas, nas competências técnicas e na complementariedade dos papéis. Fazendo com que os desafios do crescimento sejam bem absorvidos.

A consolidação de processo, seja de gestão de requisitos, de qualidade de software e gestão de projetos são essenciais para esta transformação. O uso de ferramentas

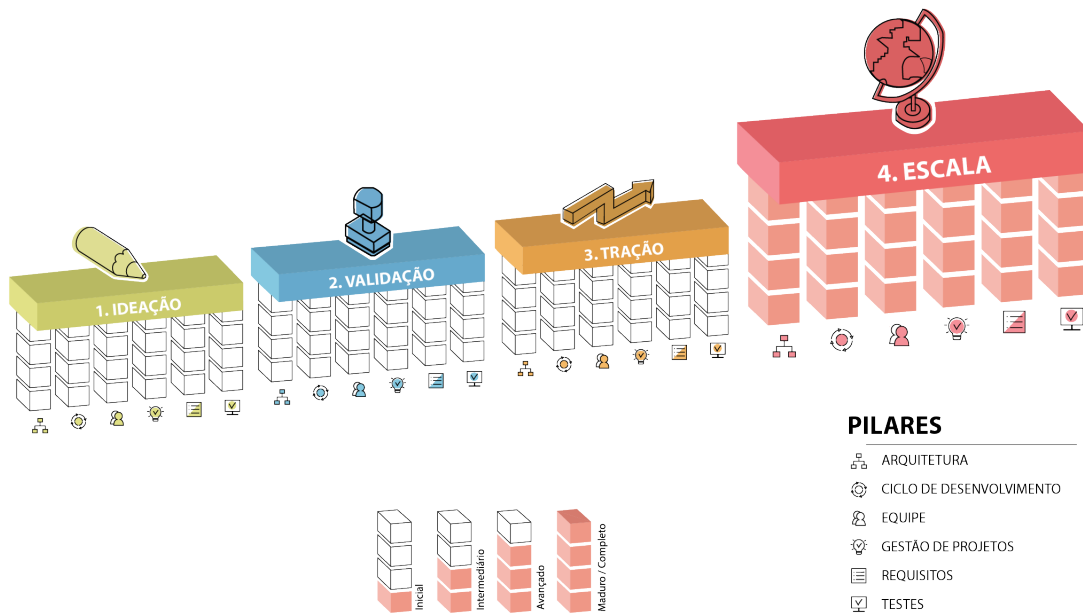


Figura 8.6 – Escala

propiciam ao negócio a escalabilidade no desenvolvimento de software. Além disso, na parte de teste de software, um profissional dedicado ou uma equipe designada para estas atividades é um dos principais avanços nesta área.

Em relação a arquitetura de software, o pensamento é de ter algo baseado nas necessidades do negócio e nos desafios de crescimento, seja pelo crescimento territorial, com novas filiais, seja pelo crescimento de mercado, com atuação em mercados nacionais e internacionais. Este crescimento requer uma estratégia de arquitetura de software baseada nestes encaminhamentos, no que o negócio indicará.

## FASES

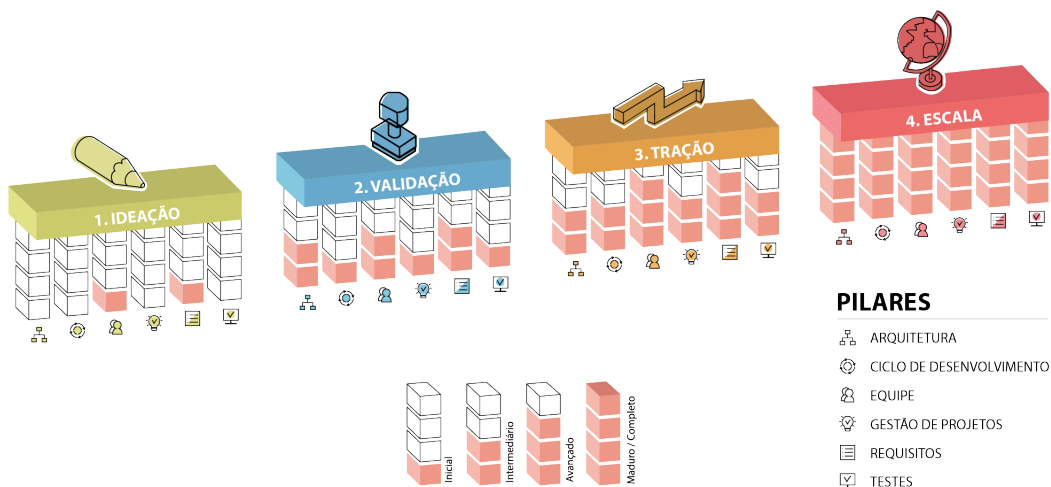


Figura 8.7 – Modelo consolidado

Como *checklist* desta fase o empreendedor tem:

- A equipe de desenvolvimento deve ter papéis complementares;
- Os requisitos são gerenciados adequadamente, com processos bem definidos e ferramentas;
- A tecnologia escolhida para a arquitetura de software suporta o crescimento esperado pela *startup*;
- A *startup* possui uma cultura de gestão de projetos
- Na área de testes de software há um profissional ou equipe dedicada;
- Todo o ciclo de desenvolvimento é bem definido e com o uso de ferramentas adequadas.

Nesta fase, os objetivos dos empreendedores foram alcançados, ou seja, a transformação da ideia em um negócio se concretiza e o amadurecimento das práticas de negócio e também das práticas de Engenharia de Software acontecem e são fundamentais para a perpetuidade do negócio. A Figura 8.7 apresenta esta evolução do ponto de vista da Engenharia de Software e também o modelo proposto por esta pesquisa.

## 9. CONCLUSÕES E TRABALHOS FUTUROS

Nos últimos anos, o termo “*startup*” tornou-se cada vez mais popular em termos de organizações que causaram impacto no mercado, principalmente pelo uso de tecnologias que permitem seu crescimento rápido e tornam seus modelos de negócio escaláveis. Nesse sentido, há uma identificação de criação de produtos de software de forma diferenciada que diverge do que normalmente era característico das empresas tradicionais. Questões como a identificação de requisitos sem ter o cliente ou mercado definido, a criação de soluções baseadas no uso de componentes existentes para acelerar a entrega do produto ou, até mesmo, a transformação rápida de uma característica de um produto no próprio produto são exemplos de como, para esse tipo de organização, o desenvolvimento de software mudou.

Neste capítulo final será apresentada a discussão sobre as descobertas desta pesquisa e os trabalhos futuros necessários para complementar os resultados apresentados.

### 9.1 Discussões

De acordo com as questões de pesquisa, apresentadas na introdução deste documento, algumas discussões e descobertas são apresentadas nesta seção. Lembrando que a principal questão de pesquisa é a seguinte:

Como adotar práticas de Engenharia de Software no desenvolvimento dos MVPs das *startups* digitais durante as diferentes fases de maturidade do negócio?

A abordagem utilizada nesta pesquisa direcionou os primeiros esforços para entender como as *startups* de software utilizavam as práticas de Engenharia de Software no desenvolvimento dos seus produtos e serviços. Na fase exploratória, uma revisão sistemática foi realizada com o objetivo principal de verificar a existência pesquisas sobre o tema m, junto às comunidades científicas.

Mesmo sendo um tema pouco explorado nas pesquisas científicas, alguns pontos importantes foram encontrados e que contribuíram para o prosseguimento desta pesquisa:

- À época, discussões mais genéricas e de mapeamento sobre processos de desenvolvimento de software foram mais presentes do que discussões sobre práticas nas diferentes etapas do desenvolvimento de um produto;
- O tema requisitos de software foi um dos pontos apresentados no resultado da revisão sistemática, bem como a adoção de métodos ágeis pelas *startups*;
- Não foi encontrada nenhuma publicação relacionada diretamente ao uso de ferramentas por *startups*;

- Nenhuma citação sobre os estágios de maturidade da *startup* e a adoção de práticas de Engenharia de Software.

Em uma segunda etapa, um estudo de campo com os empreendedores ajudou a definir com o que realmente as *startups* se preocupavam durante a criação dos seus negócios, sob o ponto de vista da Engenharia de Software. Foram definidas perguntas abertas para que uma maior quantidade de informações e realidades pudessem ser analisadas. Além disso, a escolha das *startups* foi fundamental para que elas pudessem relatar suas experiências e os avanços na adoção ou não das práticas de Engenharia de Software.

Desta forma, após analisar as entrevistas e codificá-las, foram identificados os principais eixos da Engenharia de Software que estão presentes nas decisões de desenvolvimento dos MVPs das *startups* de software. São eles:

- **Equipe:** Neste ponto, fica clara a importância que as *startups* têm na definição da equipe de desenvolvimento. Por necessitarem, muitas vezes, de mudanças rápidas e adoção de novas tecnologias, os membros da equipe se tornam peças fundamentais para o sucesso do empreendimento.
- **Requisitos de software:** Em um cenário onde as *startups* de software trabalham na definição dos requisitos de suas soluções, os problemas com a definição de requisitos crescem exponencialmente devido ao fato de que eles ainda não têm a certeza de quem será o cliente ou o problema não está explicitamente definido [PCSP19]. Esses dois pontos mostram que práticas de requisitos de software são essenciais na construção do negócio da *startup*.
- **Arquitetura de software:** As definições de quais tecnologias a *startup* deverá utilizar, desde da ideação à fase de escala, amadurecem a cada etapa do desenvolvimento. Iniciam com a utilização do conhecimento e experiência de um dos fundadores ou através da junção de tecnologias pré-definidas existentes no mercado. Após as validações de mercado e com a certeza de que o caminho escolhido pela *startup* é o correto, mudanças na arquitetura são necessárias ou até mesmo uma reengenharia é aplicada, visto que nas fases de tração e escala, a solução apresentada precisa ser mais robusta para suportar um maior número de clientes ou usuários, sem falar em questões de segurança.
- **Teste de software:** Uma das atividades com alto custo e que pouco é adotada pelas *startups* é o teste de software [ZMMW01]. Por muitas vezes os empreendedores possuem o desafio de satisfazer as necessidades dos clientes com um produto de alta qualidade em um tempo extremamente reduzido. Às vezes, também é observado que as *startups* não sabem como e o que testar. A evolução da adoção de práticas de teste de software acontece de forma estruturada somente nas fases mais avançadas

(tração e escala). Antes disso, a *startup* adota somente testes ad hoc ou até mesmo delega os testes para os clientes ou usuários da solução.

- **Gerenciamento de projetos:** Muitos empreendedores não valorizam o gerenciamento de projetos por acharem que burocratiza o processo da *startup*. A confiança em equipes auto gerenciadas e ciclos rápidos de desenvolvimento e entrega fazem com que a adoção de práticas relacionadas à gestão de projetos seja feita somente quando há um aumento significativo da equipe, do projeto ou do portfólio de produtos e serviços da *startup*.
- **Ciclo de desenvolvimento:** Quando o tema é ciclo de desenvolvimento, o entendimento é a abordagem utilizada nas diferentes etapas de maturidade da *startup*. Uma informação que foi consolidada é a adoção de metodologias ágeis pelas *startups*. Porém a forma como isto ocorre durante as quatro fases de maturidade de uma *startup* é de fato um real aprendizado para os empreendedores e para a equipe de desenvolvimento. Iniciando com algo totalmente creditado na capacidade de entrega dos fundadores e dos primeiros colaboradores, passando por processos focados em programação e adoção de ferramentas para aumentar a produtividade da equipe e chegando em uma fase com preocupações de gerenciamento de configuração, métricas e adoção e integração de outras atividades de desenvolvimento de software.

Em uma terceira etapa, um *survey* foi realizado para entender melhor o comportamento das *startups* em relação à adoção ou não das práticas de Engenharia de Software e buscar validar as premissas levantadas na fase do estudo de campo, principalmente relacionada à importância dos eixos (equipe, requisitos de software, arquitetura de software, teste de software, gerenciamento de projetos e ciclo de desenvolvimento). Foram 41 respondentes que de alguma forma validaram os eixos e indicaram um maior detalhamento do que é utilizado pelas *startups* e em que fase de maturidade. Nesta fase, a primeira versão do modelo PRESS (Práticas de Engenharia de Software em *Startups*) foi criado.

Já na fase final, a fase de avaliação, um grupo focal com *startups* em fases avançadas de desenvolvimento (fase de tração ou escala) foi executado e com isso consolidando o modelo empírico de adoção de práticas de engenharia de software por *startup*, foco desta pesquisa.

Com base nos resultados das pesquisas quantitativas e qualitativas realizadas no decorrer deste doutorado e apresentadas neste documento, acredita-se que este modelo empírico possa melhorar o entendimento das necessidades reais dessas empresas nascentes e do momento da adoção de determinadas práticas de Engenharia de Software.

Do ponto de vista acadêmico, este trabalho traz uma contribuição em analisar as práticas de Engenharia de Software não somente sob a perspectiva da equipe técnica, mas sob o olhar das necessidades do negócio e o real valor adicionado à uma empresa de tecnologia.

### 9.1.1 Limitações desta pesquisa

Nessa pesquisa foi apresentado o modelo PRESS, visando a adoção de práticas de Engenharia de Software pelas *startups* digitais nas diferentes etapas de desenvolvimento do negócio. Como em qualquer estudo empírico, algumas limitações foram identificadas.

Na fase exploratória, no contexto do desenvolvimento da revisão sistemática da literatura, pode-se inferir que, pelo fato do tema ser relativamente novo e envolver áreas do conhecimento diferentes, mas correlatas (Engenharia de Software e Governança Corporativa), uma abrangência maior das bases de dados pudesse contribuir com uma análise mais apurada das pesquisas realizadas sobre tema. Entretanto, isto aumentaria o ruído com estudos não relacionados, e por isso optou-se somente por estudos na área de Engenharia de Software.

Durante o estudo de campo (estudo de viabilidade) uma das limitações encontradas está relacionada ao aceite em participar de uma entrevista. Foram convidadas mais de 30 *startups* e somente 10 aceitaram disponibilizar um tempo para o estudo de campo. Além disso, o fato de questionar os empreendedores sobre fatos passados, principalmente relacionadas às primeiras decisões, pode gerar uma informação menos precisa do que observar as *startups* em campo desde o seu início. Entretanto, por restrições de tempo para o desenvolvimento desta pesquisa, isto não foi possível.

Sobre o *survey*, a principal limitação está relacionada à coleta de dados. Apesar de ter obtido 41 respondentes para o *survey*, a maioria se concentra na região sul do país e não obteve-se respondentes de *startups* que estivessem na fase de ideação.

Durante o estudo de avaliação e a execução do grupo focal a limitação está relacionada à possível falta de informações aprofundadas sobre o histórico de desenvolvimento de software apresentado pelos empreendedores.

Finalmente, o modelo proposto como resultado desta pesquisa foi desenvolvido utilizando um rigoroso processo científico, orientado por protocolos de pesquisa e revisão por outros pesquisadores quando possível. Porém, todo estudo empírico possui ameaças e limitações, e por isso, os resultados apresentados nesta pesquisa não podem ser generalizados.

## 9.2 Trabalhos futuros

Existem algumas lacunas que podem ser desenvolvidas em trabalhos futuros. Algumas alternativas para melhoria desta pesquisa são:



- (a) fazer uma análise mais detalhada em cada um dos eixos apresentados e em cada uma das etapas de maturidade da *startup*;
- (b) criar submodelos para cada eixo, indicando boas práticas a serem utilizadas pelas *startups* em seus diferentes estágios;
- (c) realizar uma análise crítica em relação aos modelos de maturidade existentes, como MPS.br e CMMi;
- (d) replicar a pesquisa em outros países e verificar ajustes no modelo.

### 9.3 Considerações finais

Os resultados apresentados nesta tese apresentam interessantes descobertas e ensinamentos para a comunidade em geral. Por um lado, um estudo preliminar relacionado à adoção de práticas de Engenharia de Software por *startups*, em um mundo relativamente novo e focado em resultados de negócio, mostra a informalidade no uso de práticas de Engenharia de Software de acordo com as incertezas deparadas pelos empreendedores. Por outro lado, a evolução dos negócios e o crescimento da empresa movimenta a equipe de desenvolvimento a buscar maior formalismo e boas práticas de mercado.

O que foi apresentado aqui é conclusivo, no sentido de definir um comportamento linear para estas empresas nascentes. Porém, este estudo indica um comportamento dos empreendedores e da adoção de práticas de Engenharia de Software que deve ser mais aprofundado e pesquisado. Estar no limiar entre Engenharia de Software e negócios, principalmente no campo das *startups*, pode ser uma oportunidade de descobrir a melhor forma como o desenvolvimento de software pode adicionar valor para o negócio, para o cliente e para a equipe de desenvolvimento.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [ACA20] Alves, C.; Cunha, J.; Araújo, J. “On the pragmatics of requirements engineering practices in a startup ecosystem”. In: Proceedings of the IEEE 28th International Requirements Engineering Conference (RE), 2020, pp. 311–321.
- [ACCL10] Arain, M.; Campbell, M.; Cooper, C.; Lancaster, G. “What is a pilot or feasibility study? a review of current practice and editorial policy”, *BMC medical research methodology*, vol. 10–1, Jul 2010, pp. 67.
- [Ald19] Aldaej, A. “Towards effective technical debt decision making in software startups”, *ACM SIGSOFT Software Engineering Notes*, vol. 44–3, Nov 2019, pp. 22–22.
- [All01] Alliance, A. “Agile manifesto”. URL <http://www.agilemanifesto.org>, Acessado em: Junho 2018.
- [AO18] Albuga, S.; Odeh, Y. “Towards prioritizing software business requirements in startups”. In: Proceedings of the 8th International Conference on Computer Science and Information Technology (CSIT), 2018, pp. 257–265.
- [BBND<sup>+</sup>18] Berg, V.; Birkeland, J.; Nguyen-Duc, A.; Pappas, I. O.; Jaccheri, L. “Software startup engineering: A systematic mapping study”, *Journal of Systems and Software*, vol. 144, Oct 2018, pp. 255–274.
- [BHOBL13] Bosch, J.; Holmström Olsson, H.; Björk, J.; Ljungblad, J. “The early stage software startup development model: A framework for operationalizing lean principles in software startups”. In: *Lean Enterprise Software and Systems*, Fitzgerald, B.; Conboy, K.; Power, K.; Valerdi, R.; Morgan, L.; Stol, K.-J. (Editores), 2013, pp. 1–15.
- [Bla12] Blank, S. “The startup owner’s manual: The step-by-step guide for building a great company”. BookBaby, 2012, 608p.
- [Bla14] Blank, S. “Perfection by subtraction—the minimum feature set”. URL <http://steveblank.com/2010/03/04/perfection-by-subtraction-the-minimum-feature-set/>, Acessado em: Outubro 2018.
- [BM12] Balaji, S.; Murugaiyan, S. “Waterfall vs. v-model vs. agile: A comparative study on sdlc”, *International Journal of Information Technology and Business Management*, vol. 2–1, Jun 2012, pp. 26–30.

- [BML<sup>+</sup>18] Besker, T.; Martini, A.; Lokuge, R. E.; Blincoe, K.; Bosch, J. “Embracing technical debt, from a startup company perspective”. In: Proceedings of the IEEE International Conference on Software Maintenance and Evolution (ICSME), 2018, pp. 415–425.
- [Bos16] Bosch, J. “Speed, data, and ecosystems: The future of software engineering”, *IEEE Software*, vol. 33–1, Jan-Feb 2016, pp. 82–88.
- [Bra19] Brattström, A. “Working with startups? these are the three things you ought to know about startup teams”, *Technology Innovation Management Review*, vol. 9–11, Nov 2019, pp. 5–14.
- [BTS86] Beck, L. C.; Trombetta, W. L.; Share, S. “Using focus group sessions before decisions are made.”, *North Carolina Medical Journal*, vol. 47, Feb 1986, pp. 73–74.
- [BW94] Barriball, L.; While, A. “Collecting data using a semi-structured interview: a discussion paper”, *Journal of Advanced Nursing-Institutional Subscription*, vol. 19–2, Feb 1994, pp. 328–335.
- [BWD<sup>+</sup>17] Bajwa, S. S.; Wang, X.; Duc, A. N.; Chanin, R. M.; Prikladnicki, R.; Pompermaier, L. B.; Abrahamsson, P. “Start-ups must be ready to pivot”, *IEEE Software*, vol. 34–3, May-Jun 2017, pp. 18–22.
- [Car94] Carmel, E. “Time-to-completion in software package startups”. In: Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, 1994, pp. 1–1.
- [Car02] Carlos, G. A. “Como elaborar projetos de pesquisa”. Atlas, 2002, vol. 4, 175p.
- [CB77] Cooper, A.; Bruno, A. “Success among high-technology firms”, *Business horizons*, vol. 20–2, Nov 1977, pp. 16–22.
- [CC17] Creswell, J.; Creswell, D. “Research design: Qualitative, quantitative, and mixed methods approaches”. Sage publications, 2017, 273p.
- [CDJ20] Cico, O.; Duc, A. N.; Jaccheri, L. “An empirical investigation on software practices in growth phase startups”. In: Proceedings of the Evaluation and Assessment in Software Engineering, 2020, pp. 282–287.
- [Chi17] Chicote, M. “Startups and technical debt: managing technical debt with visual thinking”. In: IEEE/ACM 1st International Workshop on Software Engineering for Startups (SoftStart), 2017, pp. 10–11.

- [CLBdFJ18] Cavalcante, B. H.; Leal, G. C. L.; Balancieri, R.; de Farias Junior, I. "Technical aspects of software development in startups: A systematic mapping". In: Proceedings of the XLIV Latin American Computer Conference (CLEI), 2018, pp. 100–109.
- [CO08a] Coleman, G.; O'Connor, R. "An investigation into software development process formation in software start-ups", *Journal of Enterprise Information Management*, vol. 21–6, Oct 2008, pp. 633–648.
- [CO08b] Coleman, G.; O'Connor, R. "Investigating software process in practice: A grounded theory perspective", *Journal of Systems and Software*, vol. 81–5, May 2008, pp. 772–784.
- [CP08] Colombo, M.; Piva, E. "Strengths and weaknesses of academic startups: a conceptual model", *IEEE Transactions on Engineering Management*, vol. 55–1, Jan 2008, pp. 37–49.
- [CPF+17] Chanin, R.; Pompermaier, L. B.; Fraga, K.; Sales, A.; Prikładnicki, R. "Applying customer development for software requirements in a startup development program". In: Proceedings of the 1st International Workshop on Software Engineering for Startups, 2017, pp. 2–5.
- [Cro02] Crowne, M. "Why software product startups fail and what to do about it. evolution of software product development in startup companies". In: Proceedings of the IEEE International Engineering Management Conference, 2002, pp. 338–343.
- [CS94] Carey, M. A.; Smith, M. "Capturing the group effect in focus groups: A special concern in analysis", *Qualitative health research*, vol. 4–1, Feb 1994, pp. 123–127.
- [CS11] Check, J.; Schutt, R. "Research methods in education". Sage Publications, 2011, 907p.
- [DA16] Duc, A. N.; Abrahamsson, P. "Minimum viable product or multiple facet product? the role of mvp in software startups". In: Proceedings of the International Conference on Agile Software Development, 2016, pp. 118–130.
- [DA17] Duc, A. N.; Abrahamsson, P. "Exploring the outsourcing relationship in software startups: A multiple case study". In: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, 2017, pp. 134–143.

- [DD08] Dybå, T.; Dingsøyr, T. “Strength of evidence in systematic reviews in software engineering”. In: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, 2008, pp. 178–187.
- [DFG03] Davila, A.; Foster, G.; Gupta, M. “Venture capital financing and the growth of startup firms”, *Journal of business venturing*, vol. 18–6, Nov 2003, pp. 689–708.
- [Dru14] Drucker, P. “Innovation and entrepreneurship”. Routledge, 2014, 288p.
- [dTMP19] de Toledo, S. S.; Martini, A.; Przybyszewska, A.; Sjøberg, D. I. “Architectural technical debt in microservices: a case study in a large company”. In: Proceedings of the IEEE/ACM International Conference on Technical Debt (TechDebt), 2019, pp. 78–87.
- [EWA15] Edison, H.; Wang, X.; Abrahamsson, P. “Lean startup: why large software companies should care”. In: Scientific Workshop Proceedings of the XP, 2015, pp. 2.
- [Fai14] Fairlie, R. “State of entrepreneurship address”. URL <https://www.clustermapping.us/sites/default/files/files/resource/2014stateofEntrepreneurshipAddress.pdf>, Acessado em: Fevereiro 2019.
- [Fli12] Flick, U. “Introdução à metodologia de pesquisa: um guia para iniciantes”. Penso Editora, 2012, 256p.
- [Fli13] Flick, U. “The SAGE handbook of qualitative data analysis”. Sage, 2013, 640p.
- [Flo84] Floyd, C. “A systematic look at prototyping”. In: *Approaches to prototyping*, Springer, 1984, pp. 1–18.
- [FS17] Fitzgerald, B.; Stol, K.-J. “Continuous software engineering: A roadmap and agenda”, *Journal of Systems and Software*, vol. 123, Jan 2017, pp. 176–189.
- [Fug00] Fuggetta, A. “Software process: A roadmap”. In: Proceedings of the Conference on The Future of Software Engineering, 2000, pp. 25–34.
- [Gat12] Gat, I. “Technical debt as a meaningful metaphor for code quality”, *IEEE software*, vol. 29–6, Nov-Dec 2012, pp. 52–55.

- [GDW<sup>+</sup>18] Gralha, C.; Damian, D.; Wasserman, A.; Goulão, M.; Araújo, J. “The evolution of requirements practices in software startups”. In: Proceedings of the IEEE/ACM 40th International Conference on Software Engineering (ICSE), 2018, pp. 823–833.
- [GPU<sup>+</sup>15] Giardino, C.; Paternoster, N.; Unterkalmsteiner, M.; Gorschek, T.; Abrahamsson, P. “Software development in startup companies: the greenfield startup model”, *IEEE Transactions on Software Engineering*, vol. 42–6, Jun 2015, pp. 585–604.
- [GUP<sup>+</sup>14] Giardino, C.; Unterkalmsteiner, M.; Paternoster, N.; Gorschek, T.; Abrahamsson, P. “What do we know about software development in startups?”, *IEEE software*, vol. 31–5, Sep-Oct 2014, pp. 28–32.
- [Gup16] Gupta, P. “How technology along with innovation is helping entrepreneurs build successful startups”. URL <https://www.entrepreneur.com/article/273841>, Acessado em: Outubro 2018.
- [GWA14] Giardino, C.; Wang, X.; Abrahamsson, P. “Why early-stage software startups fail: a behavioral framework”. In: Proceedings of the International Conference of Software Business, 2014, pp. 27–41.
- [Hig13] Highsmith, J. “Adaptive software development: a collaborative approach to managing complex systems”. Addison-Wesley, 2013, 358p.
- [IBPT18] Ibba, S.; Baralla, G.; Pinna, A.; Tonelli, R. “Survey: how much the academic startups know and use agile software and lean startup methodologies?” In: Proceedings of the 19th International Conference on Agile Software Development: Companion, 2018, pp. 1–3.
- [IDAB<sup>+</sup>19] Iankova, S.; Davies, I.; Archer-Brown, C.; Marder, B.; Yau, A. “A comparison of social media marketing between b2b, b2c and mixed business models”, *Industrial Marketing Management*, vol. 81, Aug 2019, pp. 169–179.
- [K<sup>+</sup>07] Keele, S.; et al.. “Guidelines for performing systematic literature reviews in software engineering”, Relatório Técnico, Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007, 57p.
- [KES<sup>+</sup>20] Kemell, K.-K.; Elonen, A.; Suoranta, M.; Nguyen-Duc, A.; Garbajosa, J.; Chanin, R.; Melegati, J.; Rafiq, U.; Aldaej, A.; Assyne, N.; et al.. “Business model canvas should pay more attention to the software startup team”. In: Proceedings of the 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2020, pp. 342–345.

- [Kha18] Khanna, D. “Experiential team learning in software startups”. In: Proceedings of the 19th International Conference on Agile Software Development: Companion, 2018, pp. 1–3.
- [KI12] Kniberg, H.; Ivarsson, A. “Scaling agile@ spotify with tribes, squads, chapters & guilds”. URL <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScalin.pdf>, Acessado em: Dezembro 2018.
- [Kit94] Kitzinger, J. “The methodology of focus groups: the importance of interaction between research participants”, *Sociology of health & illness*, vol. 16–1, Jan 1994, pp. 103–121.
- [Kiv18] Kivo, D. “How to find success as an entrepreneur in the digital age”. URL <https://www.forbes.com/sites/theyec/2018/08/31/how-to-find-success-as-an-entrepreneur-in-the-digital-age/>, Acessado em: Outubro 2018.
- [KLB04] Kontio, J.; Lehtola, L.; Bragge, J. “Using the focus group method in software engineering: obtaining practitioner and user experiences”. In: Proceedings. International Symposium on Empirical Software Engineering. ISESE’04., 2004, pp. 271–280.
- [Klo17] Klotins, E. “Using the case survey method to explore engineering practices in software start-ups”. In: IEEE/ACM 1st International Workshop on Software Engineering for Startups (SoftStart), 2017, pp. 24–26.
- [Ko17] Ko, A. “A three-year participant observation of software startup software evolution”. In: Proceedings of the IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), 2017, pp. 3–12.
- [KS13] Kaur, R.; Sengupta, J. “Software process models and analysis on failure of software development projects”, *CoRR*, vol. abs/1306.1068, Feb 2013, pp. 1–4, 1306.1068.
- [KUC+18] Klotins, E.; Unterkalmsteiner, M.; Chatzipetrou, P.; Gorschek, T.; Prikladnicki, R.; Tripathi, N.; Pompermaier, L. B. “Exploration of technical debt in start-ups”. In: Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice, 2018, pp. 75–84.
- [KUC+19] Klotins, E.; Unterkalmsteiner, M.; Chatzipetrou, P.; Gorschek, T.; Prikladnicki, R.; Tripathi, N.; Pompermaier, L. “A progression model of

software engineering goals, challenges, and practices in start-ups”, *IEEE Transactions on Software Engineering*, Feb 2019, pp. 1–1.

- [KUG15] Klotins, E.; Unterkalmsteiner, M.; Gorschek, T. “Software engineering knowledge areas in startup companies: a mapping study”. In: Proceedings of the International Conference of Software Business, 2015, pp. 245–257.
- [KUG19] Klotins, E.; Unterkalmsteiner, M.; Gorschek, T. “Software engineering in start-up companies: An analysis of 88 experience reports”, *Empirical Software Engineering*, vol. 24–1, May 2019, pp. 68–102.
- [Kva83] Kvale, S. “The qualitative research interview”, *Journal of phenomenological psychology*, vol. 14, Jan 1983, pp. 171–196.
- [LAFT16] Lenberg, P.; Alégroth, E.; Feldt, R.; Tengberg, L. G. W. “An initial analysis of differences in software engineers’ attitudes towards organizational change”. In: Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering, 2016, pp. 1–7.
- [LAM19] LAMY FILHO, Alfredo; PEDREIRA, J. L. B. “Governança Corporativa para Startups & Scale-ups.” IBGC Segmentos, 2019, 49p.
- [LBI20] Lim, S. L.; Bentley, P.; Ishikawa, F. “Reaching the unreachable: A method for early stage software startups to reach inaccessible stakeholders within large corporation”. In: Proceedings of the IEEE 28th International Requirements Engineering Conference (RE), 2020, pp. 376–381.
- [LPE+19] Leal, G. C. L.; Prikladnicki, R.; Ebert, C.; Balancieri, R.; Pompermaier, L. B. “Practices and tools for software start-ups”, *IEEE Software*, vol. 37–1, Jan-Feb 2019, pp. 72–77.
- [LT16] Lenarduzzi, V.; Taibi, D. “Mvp explained: A systematic mapping study on the definitions of minimal viable product”. In: Proceedings of the Software Engineering and Advanced Applications (SEAA), 42th Euromicro Conference on, 2016, pp. 112–119.
- [Mah09] Maher, P. “Weaving agile software development techniques into a traditional computer science curriculum”. In: Proceedings of the Sixth International Conference on Information Technology: New Generations, 2009, pp. 1687–1688.



- [MB20] Mangiza, P.; Brown, I. "Requisite skills profile of software development professionals for startups". In: Conference of the South African Institute of Computer Scientists and Information Technologists, 2020, pp. 102–109.
- [MBT06] Mafra, S. N.; Barcelos, R. F.; Travassos, G. H. "Aplicando uma metodologia baseada em evidência na definição de novas tecnologias de software". In: Proceedings of the 20th Brazilian Symposium on Software Engineering, 2006, pp. 239–254.
- [McD14] McDaniel, B. "Entrepreneurship and Innovation: An Economic Approach". Routledge, 2014, 288p.
- [MCS+20] Melegati, J.; Chanin, R.; Sales, A.; Prikladnicki, R.; Wang, X. "Mvp and experimentation in software startups: a qualitative survey". In: Proceedings of the 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2020, pp. 322–325.
- [MCSP20] Melegati, J.; Chanin, R.; Sales, A.; Prikladnicki, R. "Towards specific software engineering practices for early-stage startups". In: Proceedings of the International Conference on Agile Software Development, 2020, pp. 18–22.
- [Mel18] Melegati, J. "What influences software startups to use lean startup?". In: Proceedings of the 19th International Conference on Agile Software Development: Companion, 2018, pp. 1–3.
- [Mel19] Melegati, J. "Improving requirements engineering practices to support experimentation in software startups". In: Proceedings of the 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2019, pp. 1172–1175.
- [MGKW19] Melegati, J.; Goldman, A.; Kon, F.; Wang, X. "A model of requirements engineering in software startups", *Information and software technology*, vol. 109, May 2019, pp. 92–107.
- [MHASH19] Mkpojiogu, E.; Hashim, N. L.; Al-Sakkaf, A.; Hussain, A. "Software startups: Motivations for agile adoption", *Int. J. Innovative Technol. Exploring Eng.*, vol. 8–8S, Jun 2019, pp. 454–459.
- [Mis14] Miski, A. "Development of a mobile application using the lean startup methodology", *International Journal of Scientific & Engineering Research*, vol. 5–1, Jan 2014, pp. 1743–1748.

- [MK<sup>+</sup>93] Morgan, D.; Krueger, R.; et al.. “When to use focus groups and why”, *Successful focus groups: Advancing the state of the art*, vol. 1, Dec 1993, pp. 3–19.
- [MMRAGCHV19] Morente-Molinera, J. A.; Ríos-Aguilar, S.; González-Crespo, R.; Herrera-Viedma, E. “Dealing with group decision-making environments that have a high amount of alternatives using card-sorting techniques”, *Expert Systems with Applications*, vol. 127, Aug 2019, pp. 187–198.
- [Moo12] Moogk, D. R. “Minimum viable product and the importance of experimentation in technology startups”, *Technology Innovation Management Review*, vol. 2–3, Mar 2012, pp. 23.
- [Mor96] Morgan, D. “Focus groups as qualitative research”. Sage publications, 1996, vol. 16, 85p.
- [MR14] Marshall, C.; Rossman, G. B. “Designing qualitative research”. Sage publications, 2014, 352p.
- [ND19] Njima, M.; Demeyer, S. “Value-based technical debt management: an exploratory case study in start-ups and scale-ups”. In: Proceedings of the 2nd ACM SIGSOFT International Workshop on Software-Intensive Business: Start-ups, Platforms, and Ecosystems, 2019, pp. 54–59.
- [NDDSA17] Nguven-Duc, A.; Dahle, Y.; Steinert, M.; Abrahamsson, P. “Towards understanding startup product development as effectual entrepreneurial behaviors”. In: Proceedings of the International Conference on Product-Focused Software Process Improvement, 2017, pp. 265–279.
- [NDSA15] Nguyen-Duc, A.; Seppänen, P.; Abrahamsson, P. “Hunter-gatherer cycle: a conceptual model of the evolution of software startups”. In: Proceedings of the International Conference on Software and System Process, 2015, pp. 199–203.
- [NDSA16] Nguyen-Duc, A.; Shah, S. M. A.; Amrahamsson, P. “Towards an early stage software startups evolution model”. In: Proceedings of the 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2016, pp. 120–127.
- [Nji19] Njima, M. “Architecting for scale: the case for systematic software reuse in managing technical debt in start-ups”. In: Proceedings of the 13th European Conference on Software Architecture-Volume 2, 2019, pp. 55–58.

- [NOKGR12] Nord, R.; Ozkaya, I.; Kruchten, P.; Gonzalez-Rojas, M. "In search of a metric for managing architectural technical debt". In: Proceedings of the Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, 2012, pp. 91–100.
- [NP13] Nilsson, H.; Petersson, L. "How to manage technical debt in a lean startup", Dissertação de Mestrado, Department of Computer Science and Engineering (Chalmers), Chalmers University of Technology, Göteborg, Sweden, 2013, 54p.
- [OHFE16] Ouzzani, M.; Hammady, H.; Fedorowicz, Z.; Elmagarmid, A. "Rayyan—a web and mobile app for systematic reviews", *Systematic reviews*, vol. 5–1, Dec 2016, pp. 210.
- [OJLV+20] Oliveira Jr, E.; Leal, G.; Valente, M. T.; Morandini, M.; Prikladnicki, R.; Pompermaier, L.; Chanin, R.; Caldeira, C.; Machado, L.; de Souza, C. "Surveying the impacts of covid-19 on the perceived productivity of brazilian software developers". In: Proceedings of the 34th Brazilian Symposium on Software Engineering, 2020, pp. 586–595.
- [Olt16] Oltmann, S. "Qualitative interviews: A methodological discussion of the interviewer and respondent contexts". In: Forum Qualitative Sozialforschung/Forum: Qualitative Social Research, 2016, pp. 16.
- [OP10] Osterwalder, A.; Pigneur, Y. "Business model generation: a handbook for visionaries, game changers, and challengers". John Wiley & Sons, 2010, 278p.
- [Opd06] Opendakker, R. "Advantages and disadvantages of four interview techniques in qualitative research". In: Forum qualitative sozialforschung/forum: Qualitative social research, 2006, pp. 13.
- [PCS+17] Pompermaier, L. B.; Chanin, R.; Sales, A.; Fraga, K.; Prikladnicki, R. "An empirical study on software engineering and software startups: Findings from cases in an innovation ecosystem". In: SEKE, 2017, pp. 48–51.
- [PCSP19] Pompermaier, L.; Chanin, R.; Sales, A.; Prikladnicki, R. "Mvp development process for software startups". In: Proceedings of the International Conference on Software Business, 2019, pp. 409–412.
- [PFMM08] Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. "Systematic mapping studies in software engineering." In: EASE, 2008, pp. 68–77.

- [PGU<sup>+</sup>14] Paternoster, N.; Giardino, C.; Unterkalmsteiner, M.; Gorschek, T.; Abrahamsson, P. "Software development in startup companies: A systematic mapping study", *Information and Software Technology*, vol. 56–10, Oct 2014, pp. 1200–1218.
- [PMK<sup>+</sup>17] Pantiuchina, J.; Mondini, M.; Khanna, D.; Wang, X.; Abrahamsson, P. "Are software startups applying agile practices? the state of the practice from a large survey". In: *Proceedings of the International Conference on Agile Software Development*, 2017, pp. 167–183.
- [PNAE03] Prikladnicki, R.; Nicolas Audy, J. L.; Evaristo, R. "Global software development in practice lessons learned", *Software Process: Improvement and Practice*, vol. 8–4, Sep 2003, pp. 267–281.
- [Poo12] Poole, R. "Global mindset: an entrepreneur's perspective on the born-global approach", *Technology innovation management review*, vol. 2–10, Oct 2012, pp. 27–31.
- [PP20] Pompermaier, L.; Prikladnicki, R. "Brazilian startups and the current software engineering challenges: The case of tecnopuc". In: *Fundamentals of Software Startups*, Springer, 2020, pp. 331–345.
- [Pre15] Pressman, R. "Software engineering: a practitioner's approach". Palgrave Macmillan, 2015, 860p.
- [RBWL17] Rafiq, U.; Bajwa, S. S.; Wang, X.; Lunesu, I. "Requirements elicitation techniques applied in software startups". In: *Proceedings of the 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2017, pp. 141–144.
- [Rey05] Reynolds, P. "Understanding business creation: Serendipity and scope in two decades of business creation studies", *Small Business Economics*, vol. 24–4, May 2005, pp. 359–364.
- [Rie09] Ries, E. "What is the minimum viable product?" URL <http://venturehacks.com/articles/minimum-viable-product.>, Acessado em: Abril 2019.
- [Rie11] Ries, E. "The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses". Crown Books, 2011, 320p.
- [RK71] Rao, V.; Katz, R. "Alternative multidimensional scaling methods for large stimulus sets", *Journal of Marketing Research*, vol. 8–4, Nov 1971, pp. 488–494.

- [Rob01] Robinson, F. "A proven methodology to maximize return on risk". URL <https://www.syncdev.com/minimum-viable-product/>, Acessado em: Dezembro 2018.
- [Sch04] Scheuren, F. "What is a survey?" In: What is a survey?, 2004, pp. 9–14.
- [SCT01] Shull, F.; Carver, J.; Travassos, G. "An empirical methodology for introducing software processes". In: Proceedings of the 8th European Software Engineering Conference Held Jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2001, pp. 288–296.
- [Sha03] Shaw, M. "Writing good software engineering research papers". In: Proceedings of the 25th International Conference on Software Engineering., 2003, pp. 726–736.
- [SM15] Seyam, M.; McCrickard, S. "Collaborating on mobile app design through pair programming: A practice-oriented approach overview and expert review". In: Proceedings of the International Conference on Collaboration Technologies and Systems (CTS), 2015, pp. 124–131.
- [SMS+19] Souza, R.; Malta, K.; Silva, R.; Masiero, P.; Almeida, E.; Machado, I. "A case study about startups' software development practices: A preliminary result". In: Proceedings of the XVIII Brazilian Symposium on Software Quality, 2019, pp. 198–203.
- [SOL16] Seppanen, P.; Oivo, M.; Liukkunen, K. "The initial team of a software startup narrow-shouldered innovation and broad-shouldered implementation". In: Proceedings of the International Conference on Engineering, Technology and Innovation/IEEE International Technology Management Conference (ICE/ITMC), 2016, pp. 1–9.
- [Som10] Sommerville, I. "Software Engineering". USA: Addison-Wesley Publishing Company, 2010, 9th ed., 816p.
- [SRSM19] Souza, R.; Rocha, L.; Silva, F.; Machado, I. "Investigating agile practices in software startups". In: Proceedings of the XXXIII Brazilian Symposium on Software Engineering, 2019, pp. 317–321.
- [SSG12] Sharma, S.; Sarkar, D.; Gupta, D. "Agile processes and methodologies: A conceptual study", *International journal on computer science and Engineering*, vol. 4–5, May 2012, pp. 892.
- [Sut00] Sutton, S. "The role of process in software start-up", *IEEE Software*, vol. 17–4, Jul-Aug 2000, pp. 33–39.

- [TA+17] Thongsukh, S.; Ayuthaya, S. D. N.; et al.. “Startup framework based on scrum framework”. In: Proceedings of the International Conference on Digital Arts, Media and Technology (ICDAMT), 2017, pp. 458–463.
- [THB10] Tremblay, M. C.; Hevner, A.; Berndt, D. “The use of focus groups in design science research”. In: *Design Research in Information Systems*, Springer, 2010, pp. 121–143.
- [TKP+18] Tripathi, N.; Klotins, E.; Prikladnicki, R.; Oivo, M.; Pompermaier, L. B.; Kudakacheril, A. S.; Unterkalmsteiner, M.; Liukkunen, K.; Gorschek, T. “An anatomy of requirements engineering in software startups using multi-vocal literature and case survey”, *Journal of Systems and Software*, vol. 146, Dec 2018, pp. 130–151.
- [VCW+06] Vanaelst, I.; Clarysse, B.; Wright, M.; Lockett, A.; Moray, N.; S’Jegers, R. “Entrepreneurial team development in academic spinouts: An examination of team heterogeneity”, *Entrepreneurship Theory and Practice*, vol. 30–2, Mar 2006, pp. 249–271.
- [Web03] Webster, N. “Anaesthesia and Critical Care. An Exam Revision Companion”, *BJA: British Journal of Anaesthesia*, vol. 91–5, 11 2003, pp. 767–767, <https://academic.oup.com/bja/article-pdf/91/5/767/707764/aeg643.pdf>.
- [WEB+16] Wang, X.; Edison, H.; Bajwa, S. S.; Giardino, C.; Abrahamsson, P. “Key challenges in software startups across life cycle stages”. In: Proceedings of the International Conference on Agile Software Development, 2016, pp. 169–182.
- [Whi16] Whitney, D. “New media innovation and digital entrepreneurship”. URL <https://www.businessmagazinegainesville.com/new-media-innovation-and-digital-entrepreneurship/>, Acessado em: Outubro 2018.
- [WMMR06] Wieringa, R.; Maiden, N.; Mead, N.; Rolland, C. “Requirements engineering paper classification and evaluation criteria: a proposal and a discussion”, *Requirements Engineering*, vol. 11–1, Nov 2006, pp. 102–107.
- [YM13] Yau, A.; Murphy, C. “Is a rigorous agile methodology the best development strategy for small scale tech startups?”, Relatório Técnico, University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-13-01., 2013, 10p.

- [YP17] Yin, H.; Pfahl, D. "A preliminary study on the suitability of stack overflow for open innovation in requirements engineering". In: Proceedings of the 3rd International Conference on Communication and Information Processing, 2017, pp. 45–49.
- [ZMMW01] Zettel, J.; Maurer, F.; Münch, J.; Wong, L. "Lipe: a lightweight process for e-business startup companies based on extreme programming". In: Proceedings of the International Conference on Product Focused Software Process Improvement, 2001, pp. 255–270.

## APÊNDICE A – ROTEIRO DAS ENTREVISTAS

	<b>Informações Básicas</b>	QP#1	QP#2	QP#3
Qual o nome da startup?	x			
Qual é a sua função na startup?	x			
Conte-me um pouco sobre o momento atual da startup, do produto e de seus clientes.	x			
Como você descreve o negócio da startup?	x			
Qual é o tamanho do time da startup?	x			
Quantos estão envolvidos com o desenvolvimento de software?	x		x	
Me explica como a equipe de desenvolvimento está organizada			x	
[Para startups em fase de tração ou escala] Se você pudesse voltar no tempo, você mudaria a organização da equipe de desenvolvimento de software da startup ? Se sim, o que você faria de diferente?			x	
Como os requisitos do MVP foram definidos?		x		
Quais técnicas foram utilizadas para a captura dos requisitos? (entrevista, pesquisa com clientes, observação, análise da concorrência, brainstorming, desenvolvimento para um cliente específico, etc)?		x		
E como eles foram registrados?				x
[Para startups em fase de tração ou escala] Como são tratados as mudanças de requisitos?		x		
Como foi realizada a escolha da tecnologia utilizada no desenvolvimento do MVP?		x		x
[Para startups em fase de tração ou escala] Houve alguma alteração na arquitetura do software ? Se sim, por que?		x		
Como o MVP é validado?		x		
Como os problemas levantados na fase de validação do MVP são registrados?				x
Quem identifica as atividades de desenvolvimento? E como é feita a priorização?		x		
Como as atividades são direcionadas para a equipe de desenvolvimento?		x		x
Me fala um pouco sobre o processo de desenvolvimento. Vocês trabalham com modelo tradicional ou ágil ?		x		
[Para startups em fase de tração ou escala] Da forma como foi iniciada a startup e o momento atual, houve alguma mudança no processo de desenvolvimento de software? Se sim, qual? E por que?		x		x



## APÊNDICE B – ROTEIRO DO SURVEY

### Pesquisa sobre práticas de engenharia de software nos diferentes níveis de maturidade de governança de uma Startups de Software

Meu nome é Leandro Pompermaier, sou o Líder do Tecnopuc Startups e Líder da Anjos do Brasil no Rio Grande do Sul. Estou realizando este survey como parte do meu doutoramento (sendo realizado no Programa de Pós-Graduação em Ciência da Computação da PUCRS) e tem como objetivo investigar as práticas de engenharia de software das startups que são ou foram utilizadas nas diferentes fases de maturidade.

O survey é pequeno e pode ser respondido em cerca de **\*\* 10 minutos \*\***.

Todas as informações coletadas nesta pesquisa serão mantidas em sigilo, somente a consolidação dessas informações será utilizada para futuras publicações. Além disso, o relatório derivado desta pesquisa será disponível para todos os participantes que desejam recebê-lo.

Agradeço antecipadamente pelo seu apoio!

Leandro Pompermaier  
<https://www.linkedin.com/in/pompermaier/>

Qual o nome da startup? \*

Short answer text

Qual o estado de origem da sua startup? \*

Choose

Qual a sua principal relação com a startup? \*

- Sou um dos fundadores da startup
- Sou um funcionário da startup
- Sou um investidor da startup
- Other: \_\_\_\_\_

Qual o estágio atual da startup? \*

- Ideação: não tenho MVP nem produto definido
- Validação: tenho um MVP e estou com os primeiros clientes/usuários
- Tração: estou crescendo em número de clientes/usuários e/ou outros mercados
- Escala: estou em fase de aumento de portfólio de produtos e serviços e/ou outros mercados

Descreva de forma sucinta a solução proposta pela startup \*

Your answer \_\_\_\_\_

Quais papéis da equipe de desenvolvimento sua startup possui/possuiu em cada uma das fases de maturidade?

	Fase de Ideação	Fase de Validação	Fase de Tração	Fase de Escala	Não se aplica
Analista de sistema	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UX Designer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Testador/Analista de Qualidade	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DBA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Programador Web	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Programador Mobile	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Programador DEVOps	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Programador FullStack	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PO (Product Owner)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gerente de Projetos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquiteto de Software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Quais métodos de captura de requisitos foram utilizados pela startup? \*

- Entrevistas
- Pesquisas com clientes/usuários (análise de feedback)
- Observação
- Análise de concorrentes
- Brainstorming
- Desenvolvimento para um específico cliente
- Nenhuma das anteriores
- Other: \_\_\_\_\_

Qual destas categorias foi mais importante para o desenvolvimento do software da startup em cada etapa de maturidade?

	Fase de Ideação	Fase de Validação	Fase de Tração	Fase de Escala	Não se aplica
Requisitos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Verificação e Validação (teste de software)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gerenciamento de Projetos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitetura de Software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Equipe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Processo de desenvolvimento	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Quem foi o responsável pela definição dos requisitos em cada uma das fases de maturidade?

	Fase de Ideação	Fase de Validação	Fase de Tração	Fase de Escala	Não se aplica
Founder (PERFIL NÃO TÉCNICO)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Founder (PERFIL TÉCNICO)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Engenheiro de software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cliente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UX Designer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Qual destes problemas já ocorreram com a solução da startup? \*

- Não teve problemas
- Teve que ser desenvolvida novamente porque o entendimento (análise) do problema estava errado;
- Teve que ser desenvolvida novamente porque o código não estava expressando adequadamente a solução do problema analisado;
- deixou de ser usada;
- Problemas de desenvolvimento do algoritmo
- Problemas com a tecnologia escolhida
- Problemas de usabilidade
- Apresentou erros/bugs
- Other: \_\_\_\_\_

Qual o nível de formalidade do processo de desenvolvimento utilizado pela Startup? \*

- Totalmente informal: não existe uma definição de processo de desenvolvimento/práticas
- Parcialmente informal: existe uma definição de processo de desenvolvimento/práticas, porém não há uma cobrança na sua utilização
- Em formalização : as etapas/práticas são definidas a medida em que as necessidades aparecem no projeto e após isso são seguidas pela equipe
- Formal: há uma definição de processo de desenvolvimento que é seguido pela equipe

Que nível de habilidades de engenharia de software a equipe possui? \*

- Inadequado para o tamanho e complexidade do produto / serviço
- Razoavelmente adequado, mas com muitas lacunas
- Adequado com pequenas lacunas
- Adequado e suficiente para desenvolver o produto / serviço sem dificuldades
- Nenhuma das anteriores

Quais são as fontes de ideias (requisitos) para recursos de produtos / serviços? \*

- Fontes internas (por exemplo, invenção, brainstorming)
- Análise de produtos similares
- Normas / leis / regulamentos
- Objetivos de negócios
- Tendências de mercado
- Clientes potenciais e existentes
- Nenhuma das anteriores

Qual é o principal objetivo de qualidade da arquitetura de produto / serviço? \*

- Tempo de lançamento no mercado
- Funcionalidade (precisão, interoperabilidade com outros sistemas, segurança, conformidade de funcionalidade)
- Confiabilidade (maturidade, tolerância a falhas, capacidade de recuperação de erros, conformidade de fiabilidade)
- Eficiência (tempo e utilização de recursos, conformidade com a eficiência)
- Manutenibilidade (analisabilidade, mutabilidade, estabilidade, capacidade de teste, conformidade de manutenção)
- Portabilidade (adaptabilidade a diferentes ambientes, instabilidade, coexistência com outros sistemas, substituíbilidade, conformidade com a portabilidade)
- Nenhuma das anteriores

Que tipo de processo de desenvolvimento melhor caracteriza aquele utilizado na startup? \*

- Ad-hoc
- Cascata
- Iterativo
- Agile
- Nenhuma das anteriores

Em que nível o produto / serviço é testado? \*

- Teste de unidade
- Teste de integração
- Teste de sistema
- Nenhuma das anteriores

Qual (is) o (s) objetivo (s) para testar o produto / serviço? \*

- Teste de aceitação
- Teste de instalação
- Teste alfa / beta
- Teste de regressão
- Teste de performance
- Teste de segurança
- Teste de estresse
- Teste de usabilidade e interação humano-computador
- Nenhuma das anteriores

Gostaria de deixar o seu e-mail para receber o relatório derivado desta pesquisa?

Your answer \_\_\_\_\_

## APÊNDICE C – CARTÕES - CARD SORTING

1/5

Em qual destes momentos a startup se encontra hoje?

(Posicione o logo conforme a situação)

IDEAÇÃO

VALIDAÇÃO

TRAÇÃO

ESCALA



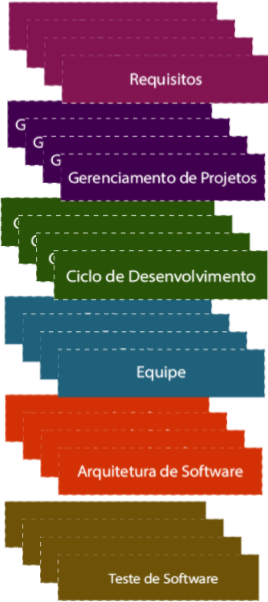


2/5

Identifique a(s) categoria(s) mais importante(s) de cada etapa:

(Pode ser mais de 1 opção por etapa)

Categorias:



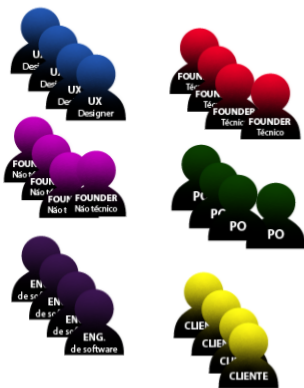
IDEAÇÃO	VALIDAÇÃO	TRAÇÃO	ESCALA

3/5

Identifique quem definiu / ou foi responsável pelos requisitos em cada fase:

(Pode ser mais de 1 opção por etapa)

Definidores/  
responsáveis:



IDEAÇÃO	VALIDAÇÃO	TRAÇÃO	ESCALA

4/5

Os principais Requisitos (funcionais ou não funcionais) de sua solução sofreram alterações em quais momentos e em qual proporção?

**Alterações:**

Grandes alterações



Alterações médias



Alterações pequenas



5/5

Qual gráfico define a proporção de equipe de TI (●) x Outras funções (●) em cada etapa da startup?

(1 opção de gráfico por etapa)

**Gráficos:**

100% TI



75% TI



50% TI



25% TI



## **APÊNDICE D – ROTEIRO PARA O GRUPO FOCAL**

**Pergunta #1** - Como foi a jornada inicial da Startup na definição do que deveria ser desenvolvido e o que não deveria ser desenvolvido?

**Pergunta #2** - Qual foi a evolução do time da startup, do day ONE até o dia de hoje, em termos de quantidade, perfil, responsabilidades?

**Pergunta #3** - Sobre as escolhas de tecnologia para a startup, o que deu certo, o que deu errado e o que ainda precisa ser feito?



Pontifícia Universidade Católica do Rio Grande do Sul  
Pró-Reitoria de Graduação  
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar  
Porto Alegre - RS - Brasil  
Fone: (51) 3320-3500 - Fax: (51) 3339-1564  
E-mail: [prograd@pucrs.br](mailto:prograd@pucrs.br)  
Site: [www.pucrs.br](http://www.pucrs.br)