



## REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO NO DESENVOLVIMENTO DE SOFTWARE COM INOVAÇÃO

Luciana Maria Azevedo Nascimento

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Guilherme Horta Travassos

Rio de Janeiro

Abril de 2019

REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO NO DESENVOLVIMENTO  
DE SOFTWARE COM INOVAÇÃO

Luciana Maria Azevedo Nascimento

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA  
DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA  
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM  
ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof. Guilherme Horta Travassos, D.Sc

---

Profa. Ana Regina Cavalcanti da Rocha, D.Sc

---

Prof. Toacy Cavalcante de Oliveira, D.Sc.

---

Prof. Marcos do Couto Bezerra Cavalcanti, D.Sc

---

Profa. Andreia Malucelli, Ph.D.

---

Profa. Andréa Magalhães Magdaleno, D.Sc

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2019

Nascimento, Luciana Maria Azevedo

Registro e Compartilhamento de Informação no Desenvolvimento de Software com Inovação/ Luciana Maria Azevedo Nascimento. – Rio de Janeiro: UFRJ/COPPE, 2019.

XII, 146 p.: il.; 29,7 cm.

Orientador: Guilherme Horta Travassos

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2019.

Referências Bibliográficas: p. 122-124.

1. Desenvolvimento de Software. 2. Registro de Informação de Software. 3. Startup de Software. I. Travassos, Guilherme Horta. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. Título.

## AGRADECIMENTOS

Ao meu orientador, Prof. Guilherme Horta Travassos, por todo o conhecimento ensinado de forma generosa, atenciosa, dedicada, enérgica e também muito paciente. Pelo exemplo dado de conduta como pesquisador, como orientador e como professor.

Aos membros do grupo de pesquisa em Engenharia de Software Experimental pelas discussões, visões e contribuições na minha formação e na condução desta tese. E também pelas conversas sobre as dificuldades de se concluir um doutorado.

À minha mãe, Nádia Nascimento e ao meu pai, Reginaldo Nascimento, pelo amor e pelos sacrifícios feitos para que eu pudesse estudar e pelo apoio incondicional à minha escolha de trilhar o caminho da pesquisa científica acadêmica. Também vos agradeço, assim como aos meus irmãos Érika e Luciano e às minhas sobrinhas Eduarda e Isabella, por compreenderem a minha ausência em vários momentos familiares.

Aos professores que me estimularam a ser pesquisadora, tanto do PESC/UFRJ quanto da Universidade Federal do Pará, onde eu tive o meu primeiro contato com a pesquisa acadêmica.

Aos amigos que torceram por mim, independente de compreenderem ou não as minhas decisões. Aos que, com muita empatia, me ensinaram sobre resiliência, reconhecimento e aceitação da transitoriedade das situações da vida e das imperfeições e limitações humanas e assim me ajudaram a cumprir essa missão: Adriana Quintana, Carla Paxiúba, Carlo Miceli, Geisa Lira, Gil Sant'anna e Renata Mesquista, muito obrigada!

Aos amigos de trabalho no CAPGOV, André, Diogo, Thiago, Racicley, Fernando e João Paulo, por contribuírem com os momentos de risos e de boa convivência em dois anos muito difíceis.

Aos funcionários da secretaria do PESC, pelo apoio sempre presente, especialmente ao Gutierrez da Costa.

Aos meus ex-alunos, pelas mensagens de carinho e incentivo.

Novamente, ao meu orientador. Espero um dia ser uma educadora e pesquisadora digna como ele é.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

## REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO NO DESENVOLVIMENTO DE SOFTWARE COM INOVAÇÃO

Luciana Maria Azevedo Nascimento

Abril/2019

Orientador: Guilherme Horta Travassos

Programa: Engenharia de Sistemas e Computação

No desenvolvimento de software para Inovação por Startups, busca-se identificar ou mesmo criar uma demanda de mercado para a qual serão sugeridas soluções. A organização desenvolvedora assume o risco do entendimento do problema e das soluções elaboradas não serem pertinentes ou não serem percebidas como pertinentes pelos potenciais clientes. Isto leva a indústria a executar práticas que busquem validar o produto o mais cedo possível. No caso de software, sua natureza flexível em termos de introdução de modificações possibilita o desenvolvimento de forma incremental e evolutiva, viabilizando frequente experimentação no mercado/contexto pretendido. Existe, nesse contexto, o objetivo de desenvolver evoluções do software no menor *time-to-market* que, em conjunto com a alta incerteza da pertinência do software, faz com que a elaboração e atualização de registros de informação sobre as ideias e características do software sejam consideradas atividades inadequadas. Porém, em um contexto futuro, a falta de conhecimento sobre as ideias e características do software podem trazer prejuízos à produtividade do desenvolvimento, manutenção e evolução do software, ou mesmo impedir o software de continuar a ser uma inovação. Nesta tese, foi elaborado um modelo conceitual de contexto para apoiar Startups de software a tomar decisão sobre suas práticas e tecnologias de apoio ao compartilhamento e registro de informação de software para a equipe de desenvolvimento. Com base em evidências obtidas a partir dos estudos executados, o modelo define elementos que possibilitam a análise de semelhança entre contextos de Startups de software, o que é requisito fundamental para a transferência de tecnologias de registro e compartilhamento de informação de software entre Startups de software.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

REGISTERING AND SHARING OF INFORMATION IN THE DEVELOPMENT OF  
SOFTWARE WITH INNOVATION

Luciana Maria Azevedo Nascimento

April/2019

Advisor: Guilherme Horta Travassos

Department: Computer Science and Systems Engineering

When developing software for Innovation, Startups strive to discover or even create a market opportunity to which they will propose a solution. The Startup takes risks related to the uncertainties on the problem and the proposed solution. It leads Startups to implement practices to validate the proposal as early as possible gradually. The flexible nature of software regarding changes favors the adoption of incremental and evolutionary development methodologies, allowing Startups to probe their ideas in the market frequently. Along with the high uncertainty about the software pertinence, this makes the team consider the elaboration and updating of documentation about software ideas and characteristics as inappropriate activities. However, in a future context after software evolution cycles, the lack of knowledge about the software ideas and characteristics can harm the productivity of the development, maintenance, and evolution of software, or even prevent the development of innovative features. In this thesis, a conceptual context model was developed to support Startups to make decisions about the adoption of practices and technologies to software information sharing and registration for the development team. Based on empirical evidence, the model defines elements to enable software Startups contexts comparison, which is an essential requirement for transferring technology on software information registering and sharing among software Startups.

# SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>IX</b>
<b>LISTA DE QUADROS.....</b>	<b>XI</b>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 CONTEXTO.....	1
1.2 MOTIVAÇÃO DO TRABALHO – O CASO DA EMPRESA BETA .....	4
1.3 PROBLEMA, OBJETIVO E PERGUNTAS DE PESQUISA .....	7
1.4 METODOLOGIA E PERGUNTAS DE PESQUISA .....	9
1.5 ORGANIZAÇÃO DO TEXTO.....	11
<b>2 FUNDAMENTAÇÃO TEÓRICA – STARTUPS E DESENVOLVIMENTO DE SOFTWARE 12</b>	
2.1 STARTUPS: DEFINIÇÃO E CARACTERÍSTICAS.....	12
2.2 O CICLO DE VIDA DE STARTUPS .....	15
2.3 PRÁTICAS DE DESENVOLVIMENTO DE SOFTWARE EM STARTUPS .....	18
2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	19
<b>3 MAPEAMENTO SISTEMÁTICO: PROPOSTAS DE APOIO AO DESENVOLVIMENTO DE SOFTWARE BASEADO EM IDEIAS E DESCOBERTA DE REQUISITOS .....</b>	<b>21</b>
3.1 INTRODUÇÃO .....	21
3.2 PROTOCOLO DO ESTUDO SECUNDÁRIO .....	22
3.2.1 Pergunta de Pesquisa .....	22
3.2.2 Estratégia de Busca.....	22
3.2.3 Busca e Seleção.....	24
3.2.4 Extração de informações dos artigos selecionados .....	26
3.2.5 Procedimento de análise dos artigos selecionados .....	28
3.3 RESULTADOS .....	31
3.3.1 Tópicos e Desafios (Issues and Challenges) .....	32
3.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	39
<b>4 ESTUDO EXPLORATÓRIO: DOCUMENTAÇÃO DE IDEIAS E CARACTERÍSTICAS DE SOFTWARE EM STARTUPS.....</b>	<b>40</b>
4.1 INTRODUÇÃO .....	40
4.2 ELABORAÇÃO DO INSTRUMENTO .....	42
4.2.1 Revisões e execuções-piloto do instrumento.....	42
4.2.2 Estrutura do instrumento .....	44
4.3 EXECUÇÃO .....	47
4.4 RESULTADOS .....	49
4.4.1 Características das organizações e dos respondentes.....	49
4.4.2 Características dos projetos de software para inovação.....	54
4.4.3 Práticas de registro de ideias e características nas organizações do estudo .....	57
4.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	64
<b>5 MAPEAMENTO SISTEMÁTICO: PRÁTICAS E TECNOLOGIAS PARA COMPARTILHAMENTO E REGISTRO DE CONHECIMENTO DE SOFTWARE EM STARTUPS .....</b>	<b>67</b>
5.1 INTRODUÇÃO .....	67
5.2 PROTOCOLO DO ESTUDO SECUNDÁRIO .....	68
5.2.1 Pergunta de Pesquisa .....	68
5.2.2 Estratégia de Busca.....	69
5.2.3 Busca e Seleção.....	71
5.2.4 Procedimento de Análise.....	74
5.3 ANÁLISE DOS ARTIGOS SELECIONADOS .....	74
5.3.1 Mapeamento de tecnologias e práticas de apoio ao desenvolvimento de software 74	

5.3.2	<i>Separação e refinamento de práticas e tecnologias para o registro e compartilhamento de ideias e características de software.....</i>	76
5.3.3	<i>Mapeamento de fatores de influência na adoção de práticas e tecnologias para o registro e compartilhamento de ideias e características de software .....</i>	78
5.3.4	<i>Mapeamento da relação entre as práticas e tecnologias identificadas com os fatores de influência.....</i>	79
5.4	<b>RESULTADOS .....</b>	80
5.4.1	<i>RQ1: O que influencia a tomada de decisão sobre a adoção de práticas e tecnologias para compartilhamento e registro de conhecimento de ideias e características de software em Startups de desenvolvimento de software?.....</i>	80
5.4.2	<i>RQ2: Quais são as tecnologias e práticas adotadas para compartilhamento e registro de conhecimento sobre ideias e características de software em Startups de desenvolvimento de software? .....</i>	87
5.5	<b>CONSIDERAÇÕES FINAIS DO CAPÍTULO .....</b>	90
<b>6</b>	<b>MODELO CONCEITUAL PARA DEFINIÇÃO DE CONTEXTO STARTUP DE ADOÇÃO DE PRÁTICAS E TECNOLOGIAS DE REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO DE SOFTWARE .....</b>	<b>91</b>
6.1	<b>INTRODUÇÃO .....</b>	91
6.2	<b>MODELO DE CONTEXTO DE STARTUP DE SOFTWARE .....</b>	93
6.2.1	<i>Atributos que caracterizam uma Startup de software .....</i>	93
6.2.2	<i>Práticas e tecnologias adotadas.....</i>	96
6.2.3	<i>Escala dos Atributos de Contexto Startup.....</i>	97
6.3	<b>ESTUDO DE AVALIAÇÃO: PERTINÊNCIA DOS ATRIBUTOS INFLUENTES DO MODELO DE CONTEXTO DE STARTUP DE SOFTWARE .....</b>	99
6.3.1	<i>Plano de Estudo .....</i>	100
6.3.2	<i>Resultados.....</i>	106
6.3.3	<i>Discussão dos resultados da avaliação. ....</i>	111
6.4	<b>TRABALHOS RELACIONADOS.....</b>	113
6.5	<b>CONSIDERAÇÕES FINAIS DO CAPÍTULO .....</b>	116
<b>7</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>117</b>
7.1	<b>RESULTADOS E OPORTUNIDADES DE EVOLUÇÃO.....</b>	117
7.2	<b>DISCUSSÃO FINAL .....</b>	119
<b>8</b>	<b>REFERÊNCIAS .....</b>	<b>122</b>
	<b>APÊNDICE 1 - ARTIGOS SELECIONADO PARA O MAPEAMENTO SISTEMÁTICO.....</b>	<b>125</b>
	<b>APÊNDICE 2 – QUESTIONÁRIO DO ESTUDO EXPLORATÓRIO EM STARTUPS .....</b>	<b>128</b>
	<b>APÊNDICE 3 – INSTRUMENTOS DO ESTUDO DE AVALIAÇÃO DA TESE .....</b>	<b>135</b>
	<b>APÊNDICE 4 – DESCRIÇÃO GERAL CONTEXTOS DE STARTUPS MAPEADOS .....</b>	<b>141</b>



# LISTA DE FIGURAS

FIGURA 1. ETAPAS DO DESENVOLVIMENTO DA TESE .....	9
FIGURA 2. PROPORÇÃO DE ARTIGOS ACEITOS POR BASE DE DADOS NA PRIMEIRA EXECUÇÃO DA BUSCA.....	26
FIGURA 3. DISTRIBUIÇÃO DOS TIPOS DE ESTUDO DO CONJUNTO DE ARTIGOS .....	27
FIGURA 4. DISTRIBUIÇÃO DO CONTEXTO DAS PROPOSTAS/ESTUDOS .....	28
FIGURA 5. EXEMPLO DE CÓDIGOS INICIALMENTE GERADOS.....	30
FIGURA 6. ETAPAS DE DESENVOLVIMENTO DO QUESTIONÁRIO .....	42
FIGURA 7. ESTUDO 1 – INFLUÊNCIAS DE OBJETIVOS NA TOMADA DE DECISÃO.....	51
FIGURA 8. MOTIVOS PARA A META DE MENOR <i>TIME-TO-MARKET</i> .....	56
FIGURA 9. ABORDAGENS DE DESENVOLVIMENTO .....	56
FIGURA 10. MOTIVOS PARA MELHORIA DA QUALIDADE/QUANTIDADE DE DOCUMENTAÇÃO .....	66
FIGURA 11. ETAPAS DE SELEÇÃO E ANÁLISE.....	74
FIGURA 12. A) ESTRUTURA DE CÓDIGOS DA PRIMEIRA ETAPA DE ANÁLISE. B) EXEMPLO DE TRECHOS CODIFICADOS.....	76
FIGURA 13. ESTRUTURA DE CÓDIGOS DE PRÁTICAS E TECNOLOGIAS .....	77
FIGURA 14. ESTRUTURA DE CÓDIGOS DE FATORES DE INFLUÊNCIA.....	78
FIGURA 15. MATRIZ DE CODIFICAÇÃO: PRÁTICAS E TECNOLOGIAS, FATORES DE INFLUÊNCIAS E ARTIGOS .....	79
FIGURA 16. EXEMPLO DE CODIFICAÇÃO SOBREPOSTA PARA REPRESENTAR RELAÇÃO DE PRÁTICA/TECNOLOGIA E FATOR. ....	80
FIGURA 17. OCORRÊNCIA DE FATORES DE INFLUÊNCIA EM CONTEXTOS DE STARTUPS MAPEADOS .....	86
FIGURA 18. PRÁTICAS E TECNOLOGIAS REPORTADAS COM FOCO EM DESENVOLVIMENTO DE SOFTWARE.....	88
FIGURA 19. PRÁTICAS E TECNOLOGIAS REPORTADAS COM FOCO EM NEGÓCIO E GESTÃO.....	89
FIGURA 20. PRÁTICAS E TECNOLOGIAS DE COMPARTILHAMENTO E/OU REGISTRO DE CONHECIMENTO .....	89

FIGURA 21. O CONTEXTO DE STARTUP DE SOFTWARE NA ADOÇÃO DE PRÁTICAS E TECNOLOGIAS PARA REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO DE SOFTWARE.....	93
FIGURA 22. ESCALA VAS .....	99
FIGURA 23. INSTRUMENTOS DO ESTUDO .....	100
FIGURA 24. CENÁRIO DO ESTUDO.....	102
FIGURA 25. PERCEPÇÃO DA INFLUÊNCIA DE CARACTERÍSTICAS NA TOMADA DE DECISÃO .....	107
FIGURA 26. PERCEPÇÃO DA SITUAÇÃO DAS CARACTERÍSTICAS INFLUENTES NAS STARTUPS PARTICIPANTES.....	108
FIGURA 27. PERCEPÇÃO DA INFLUÊNCIA DE PROBLEMAS NA TOMADA DE DECISÃO.....	109
FIGURA 28. PERCEPÇÃO DA EXISTÊNCIA DE PROBLEMAS RELACIONADOS AO COMPARTILHAMENTO E REGISTRO DE INFORMAÇÃO .....	110
FIGURA 29. PERCEPÇÃO DA INFLUÊNCIA DE OBJETIVOS DE MELHORIA NA TOMADA DE DECISÃO .....	110
FIGURA 30. PERCEPÇÃO DA PRIORIDADE ATUAL DE OBJETIVOS DE MELHORIA.....	111
FIGURA 31. DISTRIBUIÇÃO DE CONTEXTOS DE STARTUPS MAPEADOS POR PAÍS.....	120

# LISTA DE QUADROS

QUADRO 1. PROBLEMA DE PESQUISA .....	8
QUADRO 2. <i>STRING</i> DE BUSCA.....	24
QUADRO 3. ARTIGOS SELECIONADOS POR ETAPA .....	25
QUADRO 4. ESTRUTURA INICIAL DE CATEGORIAS PARA CODIFICAÇÃO .....	29
QUADRO 5. OBJETIVOS E TÓPICOS RELACIONADOS ( <i>GOALS X ISSUES</i> ) .....	37
QUADRO 6. OBJETIVO DO ESTUDO EXPLORATÓRIO .....	41
QUADRO 7. ESTUDO EXPLORATÓRIO – PROTOCOLO DE ENTREVISTA .....	48
QUADRO 8. ESTUDO EXPLORATÓRIO - PERFIL DAS ORGANIZAÇÕES.....	49
QUADRO 9. DOMÍNIOS DE APLICAÇÃO DAS ORGANIZAÇÕES .....	50
QUADRO 10. ÁREA DE FORMAÇÃO DOS PARTICIPANTES .....	52
QUADRO 11. RESPOSTAS SOBRE EXPERIÊNCIA DOS PARTICIPANTES COM DESENVOLVIMENTO DE SOFTWARE .....	53
QUADRO 12. IMPORTÂNCIA DO PROJETO DE INOVAÇÃO PARA A ORGANIZAÇÃO.....	54
QUADRO 13. ESTÁGIOS DO PROJETO DE SOFTWARE PARA INOVAÇÃO .....	55
QUADRO 14. FATORES QUE INFLUENCIAM A ESCOLHA DAS PRÁTICAS DE REGISTRO DE IDEIAS E CARACTERÍSTICAS DE SOFTWARE.....	63
QUADRO 15. FORMAS DE REGISTRO DE IDEIAS E CARACTERÍSTICAS.....	64
QUADRO 16. EVENTOS E CONSEQUÊNCIAS QUE RESULTARAM EM AUMENTO DE ESFORÇO DEDICADO AO REGISTRO DE CARACTERÍSTICAS DE SOFTWARE .....	64
QUADRO 17. OBJETIVO DO MAPEAMENTO SISTEMÁTICO .....	68
QUADRO 18. TERMOS DA <i>STRING</i> DE BUSCA.....	69
QUADRO 19. <i>STRING</i> DE BUSCA .....	70
QUADRO 20. QUANTIDADE DE ARTIGOS POR ETAPA DE BUSCA E SELEÇÃO.....	71
QUADRO 21. CRITÉRIOS DE INCLUSÃO E EXCLUSÃO. ADAPTADO DE (PATERNOSTER, GIARDINO, <i>ET AL.</i> , 2014) .....	72
QUADRO 22. ARTIGOS SELECIONADOS.....	73
QUADRO 23. CODIFICAÇÃO PROVISÓRIA.....	75
QUADRO 24. CARACTERÍSTICAS INFLUENTES .....	81
QUADRO 25. PROBLEMAS INFLUENTES .....	83

QUADRO 26. OBJETIVOS INFLUENTES.....	84
QUADRO 27. MODERADORES DA TOMADA DE DECISÃO SOBRE ADOÇÃO DE PRÁTICAS E TECNOLOGIAS DE REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO DE SOFTWARE .....	94
QUADRO 28. OBJETIVOS DE MELHORIA INFLUENTES NA TOMADA DE DECISÃO SOBRE ADOÇÃO DE PRÁTICAS E TECNOLOGIAS DE REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO DE SOFTWARE .....	95
QUADRO 29. PROBLEMAS INFLUENTES NA TOMADA DE DECISÃO SOBRE ADOÇÃO DE PRÁTICAS E TECNOLOGIAS DE REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO DE SOFTWARE.....	95
QUADRO 30. PRÁTICAS E TECNOLOGIAS PARA REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO DE SOFTWARE.....	96
QUADRO 31. OBJETIVO DO ESTUDO DE AVALIAÇÃO.....	99
QUADRO 32. DADOS DAS STARTUPS PARTICIPANTES DO ESTUDO DE AVALIAÇÃO	104
QUADRO 33. ESTÁGIO DO PROJETO DE DESENVOLVIMENTO PARA INOVAÇÃO.....	105
QUADRO 34. CARACTERIZAÇÃO DOS RESPONDENTES .....	106
QUADRO 35. REGIÃO DE CONTEXTOS DE STARTUPS REPORTADOS DE FORMA AGRUPADA	120
QUADRO 36. ATRIBUTOS DESCRITIVOS DOS CONTEXTOS DE STARTUPS MAPEADOS	141

# 1 INTRODUÇÃO

Este capítulo apresenta conceitos do contexto no qual esta tese se insere, que é o de desenvolvimento de software por organizações que empreendem no mercado com objetivo de inovar, as Startups de software. Dado o contexto, é apresentada a motivação de pesquisa desta tese, seguida pela descrição do objetivo, da metodologia e perguntas de pesquisa adotada. Por fim, a última sessão descreve a organização dos capítulos do presente texto de tese.

## 1.1 CONTEXTO

Inovação é reconhecida como um dos fatores de importância crucial para o crescimento da Economia, sendo parte de políticas estratégicas de diversos países (OECD, 2005). No Brasil, podemos observar que diferentes ações para promoção da Inovação são tomadas por entidades governamentais, como por exemplo, o Ministério de Ciência, Tecnologia, Inovação e Comunicação (MCTIC), a Financiadora de Estudos e Projetos (FINEP) e as fundações estaduais de fomento como a FAPERJ (Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro), através dos seus editais de incentivos a projetos inovadores com potencial para se tornar empresas nascentes de base tecnológica. Fora do âmbito governamental, vê-se na indústria um interesse pela criação de produtos e/ou serviços que tenham características diferentes das que já existem, com o objetivo de ganhar ou criar novos mercados ao chamar a atenção dos clientes e usuários pretendidos (EDISON, BIN ALI e TORKARA, 2013)

Dentro dessa perspectiva de crescimento econômico, inovação é conceituada em (OECD, 2005) como sendo a implementação de algo novo ou que foi melhorado de forma significativa, podendo ser um produto (bem ou serviço), um processo, novos métodos e estratégias de *marketing*, novas práticas organizacionais, e até mesmo novas estruturas no ambiente de trabalho e formas de lidar com o meio externo (clientes e fornecedores, por exemplo). Além disso, a ocorrência da inovação tem um **grau de novidade**, que pode ser (OECD, 2005), (EDISON, BIN ALI e TORKARA, 2013):

- a) Nova para a empresa – quando a inovação é implementada na empresa (internamente), podendo tal inovação não ser novidade em outras empresas;
- b) Nova para o mercado – quando a organização é a primeira a inserir a inovação no seu mercado, que é definido pela própria organização e seus competidores; e
- c) Nova para o mundo – ocorre se a organização for a primeira a inserir a inovação em todos os mercados e indústrias, incluindo o âmbito internacional.

A indústria de desenvolvimento de software não está indiferente à importância da inovação para obter competitividade, sendo um setor onde o ritmo acelerado das mudanças tecnológicas promove um sentido de urgência em inovar (ROSE e FURNEAUX, 2016). Considera-se que há inovação no setor de desenvolvimento de software quando (LIPPOLDT e STRYSZOWSKI, 2009) há o desenvolvimento de um produto ou processo de software novo, ou desenvolvimento de melhoria em relação às gerações anteriores do produto ou processo de software, ou quando é desenvolvida uma nova característica ou aplicação de produto ou processo de software existente. Além disso, é necessário que ocorra a entrada do produto de software no mercado ou ocorra o uso do processo de software no contexto pretendido. Nesta tese, o escopo de interesse é o desenvolvimento de produto de software com o objetivo de inovar.

Os possíveis graus de novidade de uma inovação, assim como os itens a) e b) supracitados, levam a entender que o desenvolvimento de qualquer software que tenha características diferentes de outros softwares é uma inovação, caso o que foi desenvolvido seja implementado no mercado/contexto pretendido. Nessa perspectiva, os desafios pesquisados e as soluções elaboradas na Engenharia de Software são intrínsecos de inovação, sendo redundante usar o termo “desenvolvimento de software para inovação”. Porém, as diferentes estratégias que a indústria utiliza para desenvolver e inserir no mercado produtos de software criam contextos que trazem à tona desafios para a pesquisa na área, incluindo as metodologias utilizadas na engenharia do software. Dentre as estratégias de desenvolvimento de software para o mercado, (PIKKARAINEN, 2011) afirma que as três descritas a seguir são as comumente utilizadas, tendo elas necessidades e desafios diferentes para alcançar inovação:

1. Desenvolvimento baseado em projeto: os produtos de software são desenvolvidos conforme um contrato estabelecido com o cliente, tendo como resultado o desenvolvimento de uma solução particularizada. As soluções são criadas especificamente para atender as necessidades de clientes e/ou usuários, que devem informar quais são tais necessidades e validar a adequação das soluções. A empresa foca seus esforços em cumprir o projeto, atendendo às requisições tanto em relação às características do produto ou acordos sobre prazos e custos.
2. Desenvolvimento de produto “fora da caixa” (out-of-the-box): ocorre quando a empresa desenvolve produtos de software que não são particularizados para um cliente, mas são oferecidos para um grupo amplo de pessoas. O produto em si passa a ser o foco, ao invés do projeto. Tem-se o desafio de chegar a uma visão do produto, coletar e selecionar ideias de potenciais requisitos funcionais e não-funcionais,

observando diferentes critérios para a tomada de decisão, como por exemplo oportunidade de mercado e retorno de investimento. Essa estratégia de desenvolvimento é a de interesse principal de pesquisa dessa proposta de tese.

3. Desenvolvimento de produto customizado: o produto de software desenvolvido é uma solução semiacabada (um *framework*) a ser adaptado às necessidades específicas dos futuros clientes. Possui características e, portanto, desafios em comum com as duas estratégias anteriores.

O desenvolvimento de software para o mercado a partir de ideias (estratégias 2 e 3 citadas), ao invés de demandas de projeto baseado em contrato com o cliente, não é um modelo de negócio recente – há décadas existe o chamado desenvolvimento de software dirigido ao mercado (*Market-driven development*) ou software empacotado (*packaged software*) (SAWYER, SOMMERVILLE e KOTONYA, 1999), (REGNELL e BRINKKEMPER, 2005). Porém, as possibilidades tecnológicas contemporâneas, como interoperabilidade entre sistemas e plataformas, a negociação de software como um serviço (SaaS), em que o usuário não precisa mais adquirir e manter a infraestrutura de operação do software, aliados à proliferação na sociedade de dispositivos contendo software, aumentaram as possibilidades de inovar pela criação de novas características de produto e serviços, tendo uma postura proativa para criação de novos mercados ao invés de somente reagir às demandas existentes. Além disso, equipes de desenvolvimento de software contam atualmente com plataformas e serviços de apoio que podem ser a infraestrutura suficiente ou boa parte do que é necessário para desenvolver e disponibilizar produtos de software no mercado, como por exemplo: ferramentas *open source*, serviços de hospedagem de sistemas na Internet, serviços de distribuição de software, possibilidade e facilidade de integração e interoperabilidade com diferentes dispositivos para aproveitamento de suas funcionalidades sem o custo de produzi-los.

Comparado com o desenvolvimento de outros produtos manufaturados, o software em si não demanda material de insumo que será transformado durante a fabricação de produtos – muitas vezes, é necessário apenas computadores, ferramentas de apoio à programação e pessoal com habilidades de desenvolvimento de software. Lippoldt e Stryzowski (LIPPOLDT e STRYSZOWSKI, 2009) argumentam também que a distribuição do software pronto, em termos de código-fonte, é facilitada pela possibilidade de copiar o software e disponibilizá-lo, por exemplo, usando a Internet, fazendo com que a entrada do software no mercado ocorra em tempo menor do que no caso de produtos manufaturados. Assim, o desenvolvimento de software para

inovação no mercado é estimulado na atualidade, aumentando a concorrência no setor. Nesse contexto, Startups de software tem chamado a atenção do mercado e de pesquisadores de Engenharia de Software por serem instituições que desenvolvem produtos inovadores que são intensivos em software enquanto lidam com incertezas extremas sobre a aceitabilidade do produto pelos clientes pretendidos e objetivam inovação e desenvolvimento de um modelo de negócio lucrativo que escale no mercado (UNTERKALMSTEINER, 2016).

A presente tese visa apoiar o desenvolvimento de software para inovação por Startups de software empreendedoras sem demandas definidas em contrato. As principais características do contexto de desenvolvimento de software para inovação que interessam para a tese aqui descrita é o alto grau de incerteza, tanto em relação ao domínio do problema quanto em relação à adequação da solução oferecida e do modelo de negócio elaborado. Tais características, juntamente com outras descritas no capítulo 2 desta tese, resultam em necessidade de desenvolver versões do software com menor *time-to-market* possível em um contexto de mudanças rápidas (COLE, 2002), (RIES, 2011), (EDISON, BIN ALI e TORKARA, 2013).

Ao reconhecer a complexidade da interação entre os diversos fatores que afetam o desenvolvimento de software para inovação, essa proposta tem como escopo os desafios relacionados ao registro de informação sobre ideias e características do software desenvolvido para inovação. No escopo desta tese, ideias são propostas de requisitos para o software, enquanto que características são propriedades inerentes de um software desenvolvido, como por exemplo os requisitos funcionais e não funcionais e a arquitetura implementada.

A seguir são apresentadas as motivações do trabalho, o contexto de desenvolvimento de software em Startups, os objetivos e as questões de pesquisa e, por fim, a metodologia de pesquisa. A organização do restante do texto é descrita ao final do capítulo.

## **1.2 MOTIVAÇÃO DO TRABALHO – O CASO DA EMPRESA**

### **BETA**

Os primeiros estímulos para o desenvolvimento da pesquisa aqui proposta ocorreram durante ações de parceria entre o grupo de Engenharia de Software Experimental (ESE) com uma organização desenvolvedora de produtos envolvendo software, além de hardware. Nesse texto a empresa é denominada como empresa Beta.

A empresa Beta desenvolve produtos com software embarcado que coletam dados e os enviam para centrais de dados, de onde podem ser acessados por sistema de software *desktop* e via *web*. A empresa iniciou suas atividades em 2007, buscando



desenvolver um produto inovador que incluiu o desenvolvimento de software. Não houve implementação de práticas de Engenharia de Software como, por exemplo, especificação de requisitos, projeto de sistema, planejamento de testes, entre outros. A empresa tornou-se bem-sucedida no empreendimento, sendo pioneira no Brasil no desenvolvimento de tecnologias de rastreamento de veículos, possuindo filial fora Brasil além de ter criado nova empresa (*spin-off*) para desenvolver um novo tipo de produto para inovação.

Em 2013 a empresa havia crescido no mercado e na quantidade de participantes. Com a obtenção de clientes, o produto inicialmente desenvolvido evoluiu para uma linha de produto, cada produto da linha podendo ter diferentes versões em operação, seja devido à evolução do produto, aos diferentes perfis de clientes ou na customização para clientes específicos. Equipes foram organizadas por linha de produto e tinham como objetivo manter em operação as versões dos clientes correntes, além de evoluir os produtos em novas versões, ou mesmo itens na linha do produto, a partir de demandas ou sugestões de clientes, de parceiros e fornecedores, da alta gerência, da própria equipe de desenvolvimento e também de equipes que lidavam diretamente com atendimento ao cliente, como por exemplo pessoas do setor de vendas. Grande parte dessas demandas e sugestões era repassada aos gerentes de produto através de e-mails ou verbalizadas durante reuniões periódicas de acompanhamento dos projetos. Dessa forma, as demandas e sugestões tinham níveis variados de detalhamento, dependendo da origem (ex.: cliente, desenvolvedor, alta gerência, vendedor). Dadas as demandas e sugestões, estipulavam-se as metas para os próximos lançamentos de versões do produto ou de linha de produto. Uma das características mais importantes de serem mencionadas acerca do contexto na empresa Beta é a cobrança por agilidade, com significado de menor *time-to-market*, sendo o termo mencionado constantemente como meta principal para as equipes.

A parceria entre a empresa Beta e o grupo ESE ocorreu em um momento de redefinição das práticas de trabalho da empresa, percebidas pela gestão da mesma como não mais adequadas para o seu contexto de trabalho. As ações iniciais com o grupo ESE na empresa consistiram em treinamentos, acompanhamento de atividades e desenvolvimento de modelos de artefatos relacionados a requisitos, planejamento e execução de testes do software, gerenciamento de projetos e identificação e modelagem de processos, tanto de desenvolvimento quanto de negócio. Um fato que chamou a atenção foi a falta de cenários de uso explícitos esperados para o produto, independente do formato de descrição. A falta desses cenários prejudicava as atividades das equipes de teste, pois os testadores planejavam e executavam testes a

partir de informações tácitas sobre esses cenários e, com isso, não conseguiam garantir alta cobertura no código.

Todas as ações de melhoria supracitadas competiam pelo esforço e a participação dos profissionais com a cobrança por menor *time-to-market* e o atendimento das requisições dos clientes. Apesar de (durante os treinamentos) os participantes indicarem entender a importância de atividades como descrever requisitos para explicitar as condições esperadas de funcionamento dos sistemas de software e planejar as atividades dos projetos de forma a conseguir fornecer indicadores de andamento para a alta gerência, houveram dificuldades em executar as atividades nos projetos. Estavam envolvidas na iniciativa de melhoria: cinco equipes de desenvolvimento de software embarcado, duas equipes de desenvolvimento de software desktop e web e uma equipe de teste de software. Dessas, duas equipes participaram de forma mais ativa - uma era responsável por um projeto de sistema de software considerado de baixa prioridade na empresa, e a outra equipe era responsável pelos testes em algumas linhas de produto desenvolvidas por outros times de desenvolvimento. Através de observação seguida de questionamentos informais aos gerentes e desenvolvedores das equipes, foram identificadas as seguintes dificuldades em executar as ações de melhoria:

- Em relação à atividade de escrita de requisitos como forma de explicitar os cenários esperados para as características já existentes no produto, os entrevistados citaram ter a sensação de estarem “documentando o passado”, o que não era estimulante e não tinha prioridade em relação às cobranças de desenvolvimento das novas versões e correções de defeitos. Também afirmaram compreender a importância de escrever os requisitos, porém não poderiam diminuir a produtividade de desenvolvimento em prol dessa atividade. Para as novas características, citou-se que a grande maioria eram entendidas e elaboradas ao mesmo tempo que desenvolvidas, sendo utilizados rascunhos com desenhos informais e principalmente os protótipos de produto, envolvendo hardware e software, para observação do que estava sendo elaborado. O código-fonte de versões anteriores era reaproveitado, porém, muitas vezes devido à falta de entendimento do código, havia a prática de reconstrução do mesmo (o que as equipes denominavam como refatoração).
- Sobre as ações relacionadas à gestão de projeto, os gerentes citaram que planejar atividades não era uma atividade trivial devido às demandas de correções de falhas, que tinham alta prioridade em relação à todas outras atividades, e devido às constantes mudanças de metas para

desenvolvimento. As metas estipuladas para os projetos não eram rígidas - se novas demandas/sugestões surgissem e fossem consideradas como tendo maior potencial de apoiar os objetivos estratégicos e de inovação da empresa, manter ou atingir competitividade no mercado, as metas mudavam. No contexto da empresa Beta, ao invés dessa situação ser considerada uma situação que pudesse levar a alguma desorganização gerencial, era uma situação estimulada nas equipes. Ademais, ser criativo e ter perspicácia em perceber novas oportunidades para evoluir os produtos eram habilidades esperadas para os membros das equipes.

A participação da autora desta tese de doutorado nas ações do grupo ESE na empresa Beta permitiu-a observar a mudança de contexto da empresa, que no início de sua operação adotou práticas que lhe ajudaram a ser criativa e inovar no mercado, sendo flexível às mudanças tecnológicas e reativa às demandas e sugestões recebidas. Com o crescimento de clientes, de linhas e versões de produtos e de pessoas envolvidas no desenvolvimento e gestão de atividades, a empresa passou a mudar seus processos. Dentro do escopo de atuação do grupo ESE, observou-se a ocorrência de conflito entre a necessidade de explicitação dos cenários de uso dos produtos (incluindo software embarcados) e a demanda por mais produtividade em desenvolvimento de novas funcionalidades, evolução das existentes e correção de defeitos, dado o objetivo de menor *time-to-market*.

O desafio que empresas Startups de software tem para balancear a necessidade do registro de informações das ideias e características de software com a demanda por menor *time-to-market*, levando em consideração o seu ciclo de vida e também do produto de software desenvolvido com o objetivo de inovação (como será discutido no capítulo 2 deste texto), motivou o desenvolvimento desta tese de doutorado.

### **1.3 PROBLEMA, OBJETIVO E PERGUNTAS DE PESQUISA**

Os resultados do estudo relatado em (GIARDINO, PATERNOSTER, *et al.*, 2015) acerca do contexto de empresas *Startups* de software, indicam que práticas de Engenharia de Software são parcialmente implantadas em Startups de software, porém geralmente apenas em estágios mais avançados do ciclo de vida de Startups. Além disso, sendo o *time-to-market* reduzido, dispende esforço com tarefas de criação e atualização de documentação não é visto como algo apropriado.

Por outro lado, semelhante ao observado na empresa Beta, o mesmo estudo relata que caso o produto de software seja bem-sucedido na obtenção de clientela no mercado, a Startup passará por um estágio de transição no seu ciclo de vida que altera o contexto de desenvolvimento, no qual a falta de conhecimento explícito sobre as

características do software pode dificultar futuras evoluções e manutenções do software, assim como a aquisição de conhecimento por novos participantes da equipe de desenvolvimento, levando à diminuição da produtividade e o risco de descontinuidade do produto. Portanto, o desenvolvimento de software para inovação por Startups, com alta incerteza sobre a pertinência das ideias para as características do software e tendo demanda por menor *time-to-market*, o conflito existente entre produtividade e registro de informações sobre ideias e características de software leva ao problema descrito no Quadro 1:

Quadro 1. Problema de Pesquisa

O problema de	Falta de conhecimento acerca do impacto da falta do registro de informações sobre ideias e características do software desenvolvido para inovação em Startups.
Afeta	A decisão sobre adoção de práticas e tecnologias para compartilhar e registrar informações sobre as ideias e características do software.
Cujo risco é	Aumento do <i>time-to-delivery</i> durante a execução de atividades para desenvolver ideias e para manter ou evoluir características do software.

Dado o problema de pesquisa e ponderando a natureza dinâmica de organizações Startups, o objetivo principal desta tese **é apoiar Startups de software na tomada decisão sobre práticas de compartilhamento e registro de informações das ideias e características do software, considerando o contexto da Startup no momento da tomada de decisão, os estágios do seu ciclo de vida e seus impactos nas necessidades de práticas a adotar.**

Porém, caracterizar o contexto de engenharia de Startups não é uma tarefa trivial, visto que se trata de um contexto intensamente flexível e reativo quanto às práticas usualmente adotadas. Tal flexibilidade e reatividade são necessárias, devido às diversas influências na definição das características do software, como por exemplo: demandas de clientes obtidos, mudanças de mercado e de tecnologias (SUTTON, 2000), que podem redirecionar a visão e objetivos da solução desenvolvida em software; resultados do ciclo de aprendizado validado, ou experimentação contínua, (BOSCH, 2012); suposições e definições do modelo de negócio, que é desenvolvido em paralelo ao produto de software (RIES, 2011), (BLANK, 2013), além do ecossistema em que a Startup está inserida, pela qual ela pode estabelecer parcerias que podem influenciar

tanto no modelo de negócios quanto nas práticas de desenvolvimento (KON, CUKIER, et al., 2014).

Outro desafio para o desenvolvimento de apoio para seleção de práticas de desenvolvimento em Startups de software é a escassez de práticas adequadas para Startups, sendo um desafio de pesquisa para a Engenharia de Software (UNTERKALMSTEINER, 2016). Startups de software geralmente adaptam práticas ágeis, porém estas não se mostram adequadas devido a seus pressupostos. Por exemplo, ter o cliente presente não condiz com a realidade de Startups, se ater a um *backlog* pode restringir a criatividade, necessária para inovação e programação em pares demanda recursos, que geralmente é extremamente limitado em Startups já que elas estão também buscando criar um negócio.

## 1.4 METODOLOGIA E PERGUNTAS DE PESQUISA

Em conformidade com os princípios da Engenharia de Software Experimental (WOHLIN, RUNESON, et al., 2000), a metodologia de pesquisa desta tese é fundamentada na utilização do método científico para execução de pesquisa e assim a organização de um corpo de conhecimento com base em evidência. Para isso, a proposta de Mafra et al (MAFRA, 2006), que enfatiza a necessidade de observação e avaliação baseada em evidência para definição de novas tecnologias de software, serve como base organizacional. A estratégia de Mafra et al (MAFRA, 2006) inicia com a execução de estudos secundários na busca por evidências na literatura técnica para apoiar a elaboração da proposta inicial de tecnologia de software e prossegue com a execução de estudos primários para avaliar e aprimorar a proposta.

Foram realizados estudos visando caracterizar o contexto do problema, identificar propostas na literatura especializada e as práticas usualmente adotadas nas organizações, culminando na elaboração de uma proposição. A Figura 1 mostra as etapas de desenvolvimento desta tese e as atividades de pesquisa relacionadas a cada etapa, descritas a seguir.

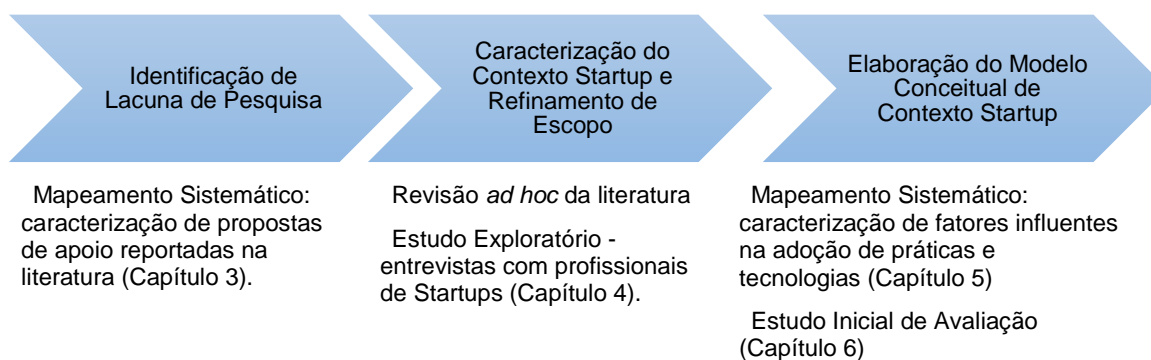


Figura 1. Etapas do Desenvolvimento da Tese

Para obter conhecimento do estado da arte no tema, foi executado um mapeamento sistemático (estudo secundário) da literatura técnica para **caracterização de propostas** de apoio ao desenvolvimento de software baseado em ideias e descoberta de requisitos, ao invés de elicitação de requisitos (ROBERTSON, 2001). A partir do levantamento das propostas de apoio, caracterizou-se os objetivos e desafios observados por pesquisadores da área. A partir dos resultados foram identificadas lacunas de pesquisa, incluindo o tema de interesse desta tese.

Para obter a percepção do estado da prática, foram executadas entrevistas (estudo primário) com profissionais de Startups visando a **caracterização das práticas adotadas e desafios reconhecidos em Startup de software** acerca do registro de informações de ideias e características de software. Considerando os desafios do estado da arte e a percepção do estado da prática, o escopo de tese foi refinado, tendo as seguintes perguntas de pesquisa:

- P1: Que fatores influenciam as organizações Startups na adoção de práticas e tecnologias para compartilhamento e registro de informações sobre as ideias e as características do software que elas desenvolvem?
- P2: Como devem ser organizados os fatores de influência da adoção de práticas e tecnologias para compartilhamento e registro de informação sobre as ideias e as características do software, de forma a apoiar as Startups na tomada de decisão?
- P3: Que práticas e tecnologias as Startups adotam para compartilhamento e registro de informação sobre as ideias e as características de software?

Para adquirir mais evidência científica, foi executado um segundo mapeamento sistemático da literatura (estudo secundário), desta vez tendo como objetivo a **caracterização, a partir de relatos de práticas da indústria Startup, dos fatores influentes na tomada de decisão** acerca de práticas e tecnologias para compartilhamento da informação do software para a equipe de desenvolvimento.

A partir dos resultados do estudo primário e do segundo mapeamento sistemático, seguiu-se a **elaboração de um modelo conceitual**, baseado em evidência, para apoio à tomada de decisão acerca de práticas e tecnologias de compartilhamento e registro de ideias e características de software para a equipe de desenvolvimento de Startups de software.

Por fim, foi executado um estudo de **avaliação da pertinência dos elementos que compõem o modelo conceitual** elaborado, contando com a participação de profissionais de Startups de software.

## **1.5 ORGANIZAÇÃO DO TEXTO**

Além deste capítulo de introdução, esta proposta possui mais 6 capítulos. No capítulo 2 são discutidos os fundamentos teóricos que embasam esta tese sobre Startups e desenvolvimento de software. O capítulo 3 descreve o mapeamento sistemático de caracterização de propostas de apoio reportadas na literatura. O capítulo 4 apresenta o estudo exploratório executado em organizações da indústria de software. No capítulo 5 é descrito o mapeamento sistemático executado acerca dos fatores influentes na decisão de práticas em Startups de software. O modelo conceitual elaborado e o estudo de avaliação são descritos no capítulo 6, enquanto que o capítulo 7 apresenta as considerações finais.

# 2 FUNDAMENTAÇÃO TEÓRICA – STARTUPS E DESENVOLVIMENTO DE SOFTWARE

Neste capítulo, são discutidas as características de empresas Startups que são apresentadas pela literatura técnica e foram brevemente apresentadas no capítulo de introdução desta tese. São descritos também os estágios de evolução de Startups e as práticas para desenvolvimento de software adotadas em Startups de software, assim como os desafios reconhecidos por pesquisadores da área. Por fim, devido à falta de consenso sobre as características de uma Startup, neste capítulo também é apresentada a perspectiva adotada pela presente tese de doutorado para a definição de instituições do tipo Startup.

## 2.1 STARTUPS: DEFINIÇÃO E CARACTERÍSTICAS

Startups são instituições humanas (temporárias) que objetivam desenvolver produtos ou serviços novos ao mesmo tempo que buscam construir um modelo de negócio próspero e escalável, operando em condições de extrema incerteza (RIES, 2011), (BLANK, 2013a). Esta definição engloba três conceitos que moldam Startups e se relacionam: inovação, empreendedorismo e alto grau de incerteza.

Para inovar, Startups procuram um mercado não atendido, completamente ou parcialmente, e investigam suas necessidades, podendo criar tanto inovações disruptivas (produtos ou serviços com características completamente novas e que mudam o mercado), incrementais (produtos com mudanças pequenas baseadas em produtos existentes, oferecendo benefícios relativamente baixos), de mercado (produtos com tecnologia similar a outros existentes, porém com benefícios monetários superiores) (EDISON, BIN ALI e TORKARA, 2013) ou gerar inovação pela disponibilização e/ou inclusão de um produto em um novo local e/ou para um grupo de pessoas não atendidas anteriormente (RIES, 2011).

Devido a seu caráter empreendedor, Startups intenciam a criação de um negócio que se sustente a partir do produto desenvolvido para inovação. Porém, Startups são diferentes de organizações tradicionais, que estabelecem um modelo de negócio contendo elementos como a visão e objetivos da empresa, clientes, parceiros, valor do produto, entre outros, para que tais elementos guiem as atividades da empresa, visando a eficiência no atingimento de seus objetivos. Ao invés de executar um modelo



de negócio definido, uma Startup define hipóteses para seu modelo de negócio e as reavalia constantemente a partir de resultados de ciclos de aprendizado (BLANK, 2013b). Além da meta de obter lucro, Startups objetivam desenvolver um modelo de negócio escalável, pois visam crescimento. Segundo Blank (BLANK, 2013a), essa característica é uma das que diferenciam Startups de pequenas empresas, já que estas não necessariamente visam crescer.

Os objetivos de inovar e desenvolver um modelo de negócio escalável contribuem para formar o contexto de extrema incerteza em que as Startups operam. Como não se trata de um projeto sobre demanda, na qual um cliente pagante da empresa indica uma ou mais pessoas para expressar o conhecimento do domínio do problema e validar a proposta de solução, podem existir incertezas em relação ao domínio do problema, ao perfil e às necessidades das pessoas e/ou organizações que se interessariam pela solução inovadora que a Startup busca desenvolver. Além da incerteza da adequabilidade da solução proposta, existem incertezas quanto ao modelo de negócio quando não se sabe de antemão, por exemplo, o quanto os possíveis usuários estarão dispostos a pagar pela solução, como fazê-los conhecer o produto, por quais meios e para quais plataformas o produto será disponibilizado, entre outras incertezas de negócio.

A ocorrência em conjunto de incertezas relacionadas à aceitação dos clientes pretendidos e à adoção no mercado, fazem parte do contexto de extrema incerteza de Startups que são de interesse da presente tese de doutorado. Tal contexto é diferente de Startups que buscam desenvolver inovação para o mercado, mas não possuem incertezas de aceitação e adoção da solução (por exemplo, uma Startup de biotecnologia que desenvolve a cura para um tipo de câncer) (BLANK, 2013a).

No caso específico de Startups de software, estas têm chamado a atenção pela contribuição na economia pois desenvolverem sistemas intensivos em software que são inovadores e tem potencial de crescimento rápido, ainda que sob condições de extremas incertezas (UNTERKALMSTEINER, 2016), (BERG, BIRKELAND, *et al.*, 2018). Quando a inovação proposta por uma Startup é representada pelas características oferecidas por um produto de software que desenvolve, o tempo de entrada do produto no mercado pode ser bem menor quando comparado com Startups que desenvolvem produtos manufaturados (LIPPOLDT e STRYSZOWSKI, 2009) devido a características inerentes de software, como por exemplo, facilidade de mudanças e evoluções no produto (em termos de alteração de código-fonte), de criação de cópias e versões, de disponibilização com apoio de plataformas na Internet, além do fato de software não ser um produto fabricado, que precisa de insumos físicos a serem obtidos e transformados.

Os relatos de pesquisa na área indicam que não há um consenso sobre as características de Startups de software (PATERNOSTER, GIARDINO, *et al.*, 2014), (UNTERKALMSTEINER, 2016), além das que são inerentes de Startups em geral (desenvolvimento de inovação, busca por um modelo de negócio escalável e extremas condições de incerteza). Outras características contextuais mais frequentemente reportadas são relacionadas à falta de recursos, à alta reatividade e flexibilidade, à intensa pressão por menor tempo de desenvolvimento e ao crescimento rápido da Startup.

Alguns estudos reportados na literatura, como (GIARDINO, WANG e ABRAHAMSSON, 2014), (KON, CUKIER, *et al.*, 2014) e (KLOTINS, UNTERKALMSTEINER e GORSCHKEK, 2019), referenciam o trabalho publicado por Sutton (SUTTON, 2000) que, com base nas características da organização que ele trabalhava na época, descreve Startups de software como empresas que:

- são novas e imaturas, logo com pouco ou nenhum histórico de operação;
- possuem limitações de recursos, sendo os existentes direcionados para colocar o produto no mercado, promovê-lo e estabelecer alianças estratégicas para a empresa;
- podem receber influências de múltiplas fontes, como por exemplo investidores, clientes, parceiros e competidores (sejam eles reais ou potenciais). As influências podem ser divergentes e inconsistentes, fazendo com que a Startup precise constantemente rever seus objetivos e ações; e
- operam em um contexto tecnológico e de mercado dinâmico e, por isso, podem ser sensíveis ao surgimento de novas tecnologias como, por exemplo, novos dispositivos computacionais, novas arquiteturas, novas linguagens de programação, novas tecnologias de comunicação, entre outros.

Sutton argumenta que, apesar de tais características poderem ser encontradas em empresas de software já estabelecidas e também nas que não são orientadas ao mercado, em Startups de software elas ocorrem em um grau extremo. Porém, definir se uma empresa é ou não uma Startup com base em tais características não é trivial, já que o autor não indica como determinar quando as características podem ser consideradas em seu grau extremo.

Ao definir uma Startup de software como uma empresa, Sutton exclui instituições que sejam organizadas como divisões, departamentos ou grupos de trabalho informais, criados com o objetivo de desenvolver um produto inovador e um negócio que se sustente a partir da inovação inserida no mercado, estando dentro de organizações

estabelecidas ou não, que sejam ou não comerciais. Tais instituições são consideradas por outros autores como (RIES, 2011), (BLANK, 2013a) e (NGUYEN-DUC, SHAH e AMBRAHAMSSON, 2016) quando, ao discutirem sobre as características de Startups na perspectiva de processos de desenvolvimento de negócio, afirmam que tempo de atuação e tamanho não definem uma Startup, pois as incertezas de criar inovação enquanto se busca por um modelo de negócio lucrativo e escalável ocorrem tanto para grupos de pessoas que trabalham por conta própria quanto em uma empresa estabelecida no mercado que cria uma divisão visando inovação e empreendedorismo. Sob esta perspectiva, em (NGUYEN-DUC, SHAH e AMBRAHAMSSON, 2016) são dados os seguintes exemplos de Startup de software em estágio inicial: uma empresa com dez anos de atuação, mas ainda em estágio sem fluxo de receita sólido gerado pela proposta de inovação; uma empresa recém-criada, formada por um ou dois fundadores; uma Startup gerada a partir de uma empresa ou de uma iniciativa acadêmica (*spin-offs*). O estágio inicial de uma Startup de software é definido em (NGUYEN-DUC, SHAH e AMBRAHAMSSON, 2016) como o período que inicia com a formação de uma Startup e termina quando a mesma consegue transformar sua ideia de negócio em uma solução viável projetada para realizar as funções principais da inovação de software.

É importante notar que Startups deixam de ser Startups quando atingem seus objetivos de inovação e empreendedorismo, portanto elas são temporárias (BLANK, 2013b). Como brevemente discutido no capítulo 1 desta tese, até chegar à condição de empresa que inovou e escalou no mercado, a Startup passa por mudanças em seu contexto que impactam nas estratégias e práticas adotadas nas atividades da Startup. A subseção a seguir apresenta modelos que discutem a evolução de Startups e os impactos da evolução no seu contexto, especialmente na perspectiva de Startups de software e suas práticas adotadas para desenvolvimento de software.

## **2.2 O CICLO DE VIDA DE STARTUPS**

Crowne (CROWNE, 2002) e Blank (BLANK, 2013a) propõem modelos para representar os estágios de desenvolvimento de Startups, tendo como base a estabilidade das características finais do produto assim como o crescimento da Startup em termos de participação no mercado, significando o quanto a Startup conseguiu escalar.

O modelo *Customer Development* (BLANK, 2013a) é utilizado na abordagem *Lean Startup* (RIES, 2011), (BLANK, 2013b), que tem chamado a atenção da comunidade de profissionais de Startups de software (BOSCH, 2013). O modelo coloca em destaque a necessidade de a Startup lidar com as incertezas acerca do problema e

da solução proposta, tendo quatro estágios: Descoberta do Cliente (*Customer Discovery*), Validação do Cliente (*Customer Validation*), Criação de Cliente (*Customer Creation*) e Criação da Empresa (*Company Building*). Nos dois primeiros estágios a Startup está em busca de um modelo de negócios adequado, enquanto nos dois posteriores ela está em fase de execução do modelo de negócios definido.

No estágio de Descoberta do Cliente, a Startup expressa suas ideias de produto e negócio como hipóteses e executa testes iniciais em relação às suposições sobre as necessidades dos possíveis cliente. Seguindo a abordagem *Lean Startup*, são executados ciclos de aprendizado com as etapas Desenvolver-Medir-Aprender (*Build-Measure-Learn*). A cada ciclo, é desenvolvido um produto mínimo viável (MVP - *Minimum Viable Product*) que possibilite experimentar as suposições a partir da interação com os possíveis clientes (RIES, 2011), (BLANK, 2013b).

Uma vez que a organização aprende com os clientes sobre o domínio do problema e a adequação da solução proposta, ela prossegue para a etapa de Validação do Cliente, quando deve testar outras hipóteses (por exemplo, hipóteses de negócios) e validar o interesse do cliente através das primeiras vendas e/ou observação de uso com possíveis clientes. Os resultados dos ciclos de aprendizado dessa fase são usados pela Startup para decidir se continua o desenvolvimento do produto e investe na sua venda em maior escala ou se muda o direcionamento de suas estratégias e objetivos (*Pivot*), testando novas hipóteses e até mesmo voltando ao estágio anterior.

Quando a Startup tem um produto que está aprimorado o suficiente para ser vendido, começa o estágio de Criação de Cliente, no qual a Startup objetiva o crescimento de demanda e, para tanto, aumenta seus esforços e investimentos em vendas e propaganda. Na última fase, a organização transita de uma empresa Startup para uma empresa que possui departamentos funcionais com processos implementados para executar seu modelo de negócios.

Por sua vez, Crowne (CROWNE, 2002) discute um modelo de evolução de Startups de software que se assemelha ao modelo *Customer Development*, porém o autor enfatiza problemas que afetam as características do software e que podem levar a Startup a falhar. Segundo Crowne, uma Startup de software evolui da sua concepção até a maturidade passando pelos estágios de Inicialização (*Startup*), de Estabilização (*Stabilization*) e de Crescimento (*Growth*), a seguir descritos em suma juntamente com problemas que Crowne considera serem riscos de cada estágio e impactam no software.

O estágio de Inicialização começa quando um ou mais empreendedores tem uma visão de produto e formam um time para desenvolvê-lo, geralmente transformando a visão e as ideias do produto diretamente em código fonte, confiando no comprometimento com o projeto e capacidade de comunicação rápida entre as pessoas

do time ao invés de processos e práticas da Engenharia de Software. Algumas das práticas de trabalho estabelecidas não servirão para quando a Startup escalar, necessitando de melhorias caso a Startup avance de estágio. Pode ser que, devido a limitações de recursos, a Startup contrate pessoas inexperientes para a equipe de desenvolvimento e isso pode contribuir para a criação de um produto não confiável. Outro risco é relacionado à falta de gestão sobre a plataforma do software, visto que decisões sobre componentes utilizados são tomadas diretamente pelos desenvolvedores, levando a uma arquitetura que não é compreendida pelo time e dificulta a evolução do software.

Quando o produto é vendido para o primeiro cliente, começa o estágio de Estabilização do produto. Ainda sem ter um produto confiável e maduro, a Startup entrega o produto para um conjunto de clientes e recebe deles requisições de funcionalidades para o software. É difícil gerenciar os requisitos, o que pode levar a mudanças de práticas previamente escolhidas. Para atender aos clientes, a Startup pode decidir criar versões customizadas do produto e contratar novas pessoas para o time de desenvolvimento, incluindo desenvolvedores mais experientes. Também pode decidir contratar pessoas que lidam com processos de negócio, como vendas e marketing, além de buscar por investidores. A Startup cresce em quantidade de pessoas, logo aumenta a quantidade de interessados e dos que influenciam as decisões tomadas sobre os requisitos do software. Ao invés de concentrar esforços no desenvolvimento e evolução de funcionalidades do software, a equipe de desenvolvimento também passa a fornecer serviços para as outras áreas da Startup, além de se envolver em atividades de vendas e comissionamento devido ao conhecimento especializado sobre o produto desenvolvido. Com o time de desenvolvimento sobrecarregado e recebendo influências de diversas fontes, a produtividade de manutenção e desenvolvimento de novas funcionalidades no software diminui.

O estágio de Crescimento da Startup inicia quando a organização consegue vender o produto de software para novos clientes sem causar sobrecarga para o time de desenvolvimento. A organização passa a despender mais esforço e recursos para estabelecer sua participação e crescimento de mercado, além de implementar processos necessários para apoiar o desenvolvimento e vendas de produtos. Alcançando tais objetivos, a organização chega à maturidade e deixa de ser uma Startup. Caso a Startup não tenha garantido ações de transferência de conhecimento e desenvolvimento de habilidades na equipe, as atividades de desenvolvimento passam a depender de um grupo pequeno de desenvolvedores que continuaram na Startup desde seus estágios iniciais, o que atrasa atividades de desenvolvimento. Além disso,

a Startup recebe demandas para que o software funcione em diferentes plataformas, o que requer mais testes no software e habilidades do time de desenvolvimento.

Os modelos de Crowne e de Blank enfatizam a mudança de contexto durante a evolução da Startup para um negócio que escalou e inovou no mercado. Inicialmente os esforços se concentram em ter um produto e entender o problema e o mercado. O software e suas características são muito instáveis principalmente nos estágios iniciais, que de forma exploratória buscam descobrir quem são os possíveis clientes, entender as suas necessidades e observar a aceitação da solução. Em estágios posteriores, a instituição precisa também investir na execução do modelo de negócio. Assim, cada estágio tem metas diferentes, trazendo desafios e, por consequência, influenciando as práticas implantadas na Startup.

## **2.3 PRÁTICAS DE DESENVOLVIMENTO DE SOFTWARE EM STARTUPS**

Principalmente nos estágios iniciais, a equipe de desenvolvimento de software em Startups tem como um dos principais objetivos a liberação de versões para execução de ciclos de aprendizado, o mais rápido e frequente possível, para apoiar a diminuição de incertezas sobre a aceitação do software e as estratégias de negócio. Assim, as abordagens viáveis de serem utilizadas no contexto são as que permitam o desenvolvimento incremental e evolutivo do software e que também sejam flexíveis às mudanças. Tal objetivo também afeta as práticas adotadas para as atividades específicas de engenharia do software, como evidenciam os resultados de pesquisa reportados em (PATERNOSTER, GIARDINO, *et al.*, 2014) e (SOUZA, MALTA e DE ALMEIDA, 2017), descritos a seguir.

A gestão das atividades em Startups de software é considerada fraca, dada a alta flexibilidade e reatividade necessárias para o contexto de mudanças frequentes. Assim, adota-se o empoderamento da equipe: as pessoas tomam as decisões nas suas tarefas e se comunicam frequentemente para manter os objetivos alinhados.

Quanto aos requisitos, estes são tipicamente *market-driven* (REGNELL e BRINKKEMPER, 2005). Não há clientes pagantes definindo-os e a Startup objetiva atingir uma grande quantidade de clientes dentre os interessados no domínio de problema do software. A documentação de requisitos é inexistente ou deficiente, visto que não há certeza da pertinência dos requisitos e eles podem ser removidos ou alterados após serem validados em um ou mais ciclos de aprendizado.

Em relação ao projeto e arquitetura do software, não há documentação arquitetural ou especificação de projeto, sendo aplicados princípios simples quando há preocupação de ser ter uma arquitetura minimamente escalável. Para o

desenvolvimento, são utilizadas plataformas conhecidas de forma extensiva, além de componentes de código e software de terceiros.

As atividades relacionadas à garantia da qualidade são focadas na experiência do usuário (*UX*), geralmente com observação de aceitabilidade dos primeiros clientes (*early-adopters*) em sessões de Grupos Focais. Se necessário, algumas Startups terceirizam as atividades de teste para que suas equipes foquem no desenvolvimento das funcionalidades do software e estas possam ser validadas o mais cedo possível. Para apoiar a análise dos resultados de validação, são utilizadas métricas ad hoc para medir a aceitabilidade e demanda do produto.

Os resultados dos estudos de pesquisa reportados em (PATERNOSTER, GIARDINO, *et al.*, 2014) e (SOUZA, MALTA e DE ALMEIDA, 2017) contribuem para obtenção de conhecimento sobre o contexto de desenvolvimento de software por Startups, porém não são discutidas as práticas adotadas ao longo do ciclo-de-vida da Startup como reação às mudanças no seu contexto, como por exemplo as citadas na sessão anterior na descrição do modelo proposto por Crowne para evolução de Startups de software.

Em (GIARDINO, PATERNOSTER, *et al.*, 2015) são reportadas evidências de que as práticas adotadas para apoiar a agilidade de desenvolvimento nos estágios iniciais, comprometem a agilidade e a qualidade do software quando a Startup transita para o estágio de crescimento, pois a falta de especificações, padronizações e processos contribuem para o acúmulo de dívida técnica. Quando o time de desenvolvimento cresce e/ou ocorrem mudanças de pessoas na equipe, o acúmulo de conhecimento tácito aumenta a curva de aprendizado dos novatos na equipe e, portanto, o tempo que eles levam para contribuir com o desenvolvimento do software, além de aumentar o risco de inserção de defeitos no software. Quando o produto escala (e assim o número de usuários aumenta), torna-se necessário implantar processos de gestão de negócio e de desenvolvimento e a qualidade do produto em termos de ausência de defeitos passa a ganhar mais importância estratégica, pois a reputação do produto no mercado pode vir a interferir na taxa de crescimento de usuários (GRALHA, DAMIAN, *et al.*, 2018). Porém, assim como aconteceu com a empresa Beta (vide capítulo 1), a falta de conhecimento explícito pode vir a prejudicar ou até mesmo inviabilizar a execução de testes efetivos no software.

## **2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO**

Para elaborar métodos e ferramentas de apoio ao desenvolvimento de software em Startups, é necessário entender as suas características do contexto (UNTERKALMSTEINER, 2016). Porém, ainda há falta de consenso sobre o que são

Startups de software e, portanto, deficiência de conhecimento sobre o contexto de Startups, o que é agravado pelo fato de Startups terem múltiplas influências (tanto internas quanto externas) na tomada de decisão sobre suas práticas.

Levando em consideração a falta de consenso sobre o que caracteriza Startups de software, esta tese de doutorado se apoia na definição de Startups dada por Ries (RIES, 2011) e Blank (BLANK, 2013b) e assim considera como Startup de software instituições humanas que tenham por objetivo criar um modelo de negócio escalável pelo desenvolvimento de um produto ou serviço com o objetivo de inovar no mercado a partir das características do software.



# 3 MAPEAMENTO SISTEMÁTICO: PROPOSTAS DE APOIO AO DESENVOLVIMENTO DE SOFTWARE BASEADO EM IDEIAS E DESCOBERTA DE REQUISITOS

Este capítulo descreve a execução de um mapeamento sistemático da literatura técnica para caracterizar as propostas da pesquisa científica para o desenvolvimento de software a partir de ideias de requisitos e/ou soluções visionadas.

## 3.1 INTRODUÇÃO

Como discutido no capítulo anterior, não existe consenso na literatura técnica acerca das características de Startups, exceto pelas características de desenvolvimento de inovação para um mercado amplo e busca por um modelo de negócio escalável, além do contexto de extremas incertezas. Diante de tal falta de consenso, a estratégia elaborada para a investigação sobre problemas e desafios de registro de informações de software no contexto de Startups foi a de primeiramente aprofundar o conhecimento do estado da arte sobre as seguintes características de desenvolvimento em Startups de software: os requisitos são *market-driven* ( (REGNELL e BRINKKEMPER, 2005)) e, assim como o projeto e arquitetura do software, raramente documentados ( (PATERNOSTER, GIARDINO, *et al.*, 2014)), ( (KLOTINS, UNTERKALMSTEINER e GORSCHKEK, 2019)).

Por serem *market-driven*, os requisitos são inventados ou descobertos sem serem demandados e nem validados por um cliente, sendo a adequação dos requisitos, sob o ponto de vista dos clientes pretendidos, uma incerteza para a organização desenvolvedora. Considerando o tema de registro de informações em um contexto que é *market-driven*, as etapas de investigações iniciais desta tese de doutorado envolveram analisar o estado da arte de propostas para o contexto onde os requisitos são inventados ou descobertos. Através da caracterização das soluções propostas na literatura técnica buscou-se conhecer quais desafios são considerados pela pesquisa científica e quais são as evidências apresentadas por essas soluções propostas para o desenvolvimento de software a partir de ideias de requisitos e/ou soluções envisionadas através da descoberta de requisitos. No contexto dessa pesquisa de doutorado, o termo descoberta de requisitos é utilizado com o significado de perceber uma demanda de mercado não satisfeita, ou necessidade não declarada por um ou mais clientes,

diferente do significado de coletar ou elicitar requisitos específicos de clientes patrocinadores, como ocorre no desenvolvimento de software convencional (sob demanda).

A seção 3.2 a seguir apresenta o protocolo do estudo secundário, que é baseado em (BIOLCHINI, MIAN, *et al.*, 2005), enquanto que os resultados são detalhados na seção 3.3.

## 3.2 PROTOCOLO DO ESTUDO SECUNDÁRIO

### 3.2.1 Pergunta de Pesquisa

Este mapeamento sistemático possui as seguintes perguntas de pesquisa:

Q1. **Questão Principal:** **Que apoio é oferecido pelos métodos, técnicas, práticas e ferramentas propostos para os projetos de desenvolvimento de software com criação de ideias de características e/ou descoberta de requisitos de software?**

**Questões secundárias:**

Q1.1. As propostas apoiam o registro de informações sobre características de software? Quais e como as informações são registradas?

Q1.2. Os resultados das propostas foram observados através de que tipos de estudo?

### 3.2.2 Estratégia de Busca

Antes de elaborar a *string* de busca, uma pesquisa *ad-hoc* na literatura técnica foi realizada usando termos identificados a partir do conhecimento da pesquisadora autora desta de tese. Os termos de busca foram então refinados a partir dos resultados da busca *ad-hoc* e foi definido um conjunto inicial de artigos de controle para o mapeamento sistemático, contendo artigos cujas propostas oferecem apoio para o desenvolvimento de software a partir de ideias ou descoberta de requisitos e cujos resultados foram observados na indústria ou ao menos em algum estudo de observação. Também buscou-se por revisões e mapeamentos sistemáticos relacionados ao tema da pesquisa [ (LEMONS, ALVES, *et al.*, 2012), (SAHA, SELVI, *et al.*, 2012), (WNUK e RUNESON, 2013), (PATERNOSTER, GIARDINO, *et al.*, 2014)], que levaram ao surgimento de novos termos para a busca assim como novos artigos de controle. A seguir tem-se a lista do conjunto final de artigos de controle:

- El-Sharkawy, S., Schmid, K.: A heuristic approach for supporting product innovation in requirements engineering: a controlled experiment. In: Proc. of the 17th REFSQ Conference, Heidelberg, Springer-Verlag, 2011.

- Maiden N., Gizikis A., Robertson S.: Provoking Creativity: Imagine What Your Requirements Could be Like. IEEE Software, vol. 21, issue 5, 2004.
- Maiden, N., Ncube, C., Robertson, S.: Can Requirements be Creative? Experiences with an Enhanced Air Space Management System. In: Proceedings of International Conference on Software Engineering, 2007.
- Sakhnini V., Mich L., Berry D.M.: The effectiveness of an optimized EPMcreate as a creativity enhancement technique for Web site requirements elicitation. Requirements Engineering, 17, 2012.
- Seyff N., Maiden N., Karlsen K., Lockerbie J., Grunbacher P., Graf F., Ncube C.: Exploring how to use scenarios to discover requirements. Requirements Engineering, Volume 14 Issue 2, April 2009.
- Zachos, K., Maiden, N., Tosar, A.: Rich-Media Scenarios for Discovering Requirements. IEEE Software 22, 89-97, 2005.

A *string* de busca foi elaborada seguindo a estrutura PICO (PAI, MCCULLOCH, *et al.*, 2003), que divide a string em partes que indicam a População (*Population*) de interesse da pesquisa, a Intervenção (*Intervention*) que será analisada, Comparação (*Comparison*), caso seja aplicável determinar grupo(s) de comparação, e Resultados (*Outcome*) da intervenção. Por se tratar de um mapeamento sistemático com o objetivo de caracterização, ao invés de síntese de evidência dos efeitos de uma intervenção (como por exemplo, da aplicação de uma tecnologia), não foram definidos termos de Comparação para a *string*.

- **População:**
  - software, application, system, program, software-based product
  - development, engineering, project, design
  - requirements engineering, requirements process, requirements elicitation
- **Intervenção:**
  - method, approach, strategy, tool, mechanism, process, practice, methodology, technique, support
- **Comparação:** Não aplicável
- **Resultado (Outcome):**
  - requirement, idea, solution
  - creation, discovery, generation, innovation, invention, ideation, trigger.
  - new, innovative, early

Inicialmente os termos contendo *requirements* não faziam parte da População na *string*, sendo adicionados após os testes de execução da busca não retornarem alguns artigos de controle. Percebeu-se que esses artigos não utilizavam palavras

relacionadas ao desenvolvimento de software, programa ou sistema nos seus títulos, resumos e nem nas suas palavras-chave, porém usavam termos como, por exemplo “*requirements elicitation*” e “*requirements engineering*”, apresentando suas propostas como aplicáveis às atividades e práticas de Engenharia de Requisitos. O formato da *string* de busca é detalhado a seguir no Quadro 2:

Quadro 2. *String* de Busca

```
P =( "software engineering" OR "software development" OR
"software project*" OR "software design" OR "software-
based product*" OR "software-based system*" OR "software
intensive system*" OR "software system*" OR "requirements
engineering" OR "requirements process" OR "requirements
elicitation" OR "system engineering" OR "system
development" OR "system project*" )
AND
I = ( method OR approach OR strategy OR tool OR
mechanism OR process OR practice OR methodology OR
technique OR support )
AND
O = ("requirement* creation" OR "creation of requirement*"
OR "creati* requirement*" OR "innovative requirement*"
OR "requirement* innovation" OR "requirement* discover*"
OR "discovery of requirement*" OR "discovering
requirement*" OR "requirement* invention" OR "invention
of requirement*" OR "invent* requirements" OR "early
requirements" OR "idea* generation" OR "idea* invention"
OR "invent* idea*" OR "idea* finding" OR "creat* idea*"
OR "idea* creat*" OR "innovative idea*" OR "new idea*"
OR "ideation" OR "idea trigger" OR "innovative solution*"
OR "novel solution*" OR "innovative technical solution*"
) )
```

### 3.2.3 Busca e Seleção

A *string* de busca foi executada em Janeiro/2015 nas bases de dados *Scopus*, *Web of Science (WoS)* e *IeeeXplorer*, retornando respectivamente 874, 250 e 178 artigos (total: 1310 artigos). Após a remoção de artigos duplicados, os artigos restantes foram avaliados conforme critérios de inclusão definidos, aplicados primeiramente pela proponente desse projeto durante a leitura de título e resumo de cada artigo. Seguiu-se então a etapa de leitura completa dos artigos que passaram pelos critérios na etapa anterior e também leitura dos artigos sobre os quais a pesquisadora teve dúvidas para excluí-los. Durante essa etapa, cada artigo foi novamente avaliado em relação aos critérios de inclusão e exclusão e os casos de dúvidas foram levados à discussão com o pesquisador orientador. Os critérios utilizados nas duas etapas foram:

- Critérios de Inclusão:
  - Discutir acerca de desenvolvimento de software E
  - Discutir sobre métodos, práticas, técnicas ou ferramentas para criação de ideias ou descoberta de requisitos de software
- Critérios de Exclusão:
  - Artigo não está escrito em inglês OU
  - Artigo não está disponível OU
  - Artigo não é relacionado com desenvolvimento de software OU
  - Artigo não apresenta métodos, práticas, técnicas ou ferramentas para criação de ideias ou descoberta de requisitos de software OU
  - Artigo não é artigo completo (é revisão, pôster ou *short paper* com menos de quatro páginas)

O Quadro 3 mostra a quantidade de artigos selecionados desde a execução da busca nas bases de dados até a seleção do conjunto final de artigos, composto de 41 artigos, listados no Apêndice 1 - Artigos Selecionado para o Mapeamento Sistemático. Em agosto de 2015 a *string* de busca foi novamente executada apenas na base *Scopus*, resultando na inclusão de um artigo. A decisão de reexecutar a busca apenas na base *Scopus* foi tomada porque na primeira execução constatou-se que os artigos das outras bases constavam em grande parte também na base *Scopus*. A Figura 2 mostra que, no conjunto de artigos selecionados, mais de 95% dos artigos estão indexados na *Scopus*.

Quadro 3. Artigos selecionados por etapa

Etapa da seleção de artigos	Número de artigos aceitos
Execução da <i>string</i> de busca	1310
Remoção de duplicatas com base no DOI	999
Leitura do título, resumo e palavras-chave	126
Leitura completa	41
Atualização da busca	1
<b>Total</b>	<b>42</b>

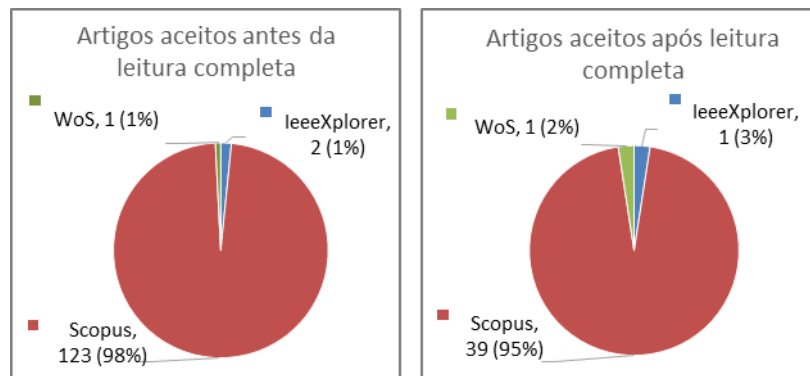


Figura 2. Proporção de artigos aceitos por base de dados na primeira execução da busca

### 3.2.4 Extração de informações dos artigos selecionados

Foram extraídas dos artigos selecionados informações bibliográficas (título, autores, ano, tipo de publicação, resumo, evento ou periódico em que foi publicado) e também o nome da proposta (caso tenha), o tipo de apoio proposto (método/processo, técnica, prática ou ferramenta), o objetivo da proposta, descrição resumida da proposta, quais atividades de engenharia de requisitos apoia, se e como se integra com as atividades do ciclo de desenvolvimento de software, se apoia a explicitação e/ou registro de ideias e características de software e como são registradas, em que domínio de aplicação a proposta foi avaliada e que indicadores os autores utilizaram para observar a efetividade da proposta.

Como a agregação de evidências sobre um determinado fenômeno não foi o objetivo do mapeamento sistemático executado, e também por perceber durante a revisão *ad-hoc* da literatura a falta de rigor científico experimental na avaliação das propostas, mas ainda assim considerá-las como importantes para o mapeamento dos desafios de pesquisa na área, optou-se por não restringir o conjunto de artigos aceitos quanto à força da evidência apresentada. Para fins de análise, os estudos apresentados em cada artigo foram classificados conforme as categorias a seguir, que são baseadas em (DIAS NETO, SUBRAMANYAN, *et al.*, 2008). A Figura 3 mostra a distribuição dos tipos de estudo no conjunto de artigos selecionados no mapeamento.

- **Especação (*Speculation*)**: O artigo apresenta a opinião pessoal de alguém sobre o método, prática, técnica ou ferramenta ser bom ou não, sem apresentar nenhum estudo ou exemplo que indique sua viabilidade em projetos de software.
- **Exemplo (*Example*)**: O artigo descreve o método, prática, técnica ou ferramenta e um exemplo de uso. Porém, não apresenta critérios para comparar o desempenho da proposta.

- Prova de conceito (Proof of Concept): O artigo descreve o uso do método, prática, técnica ou ferramenta em um sistema desenvolvido para demonstrar a viabilidade da proposta (*toy system*) ou em um projeto real sem implicações de pressão comercial. Algumas medidas são coletadas para mostrar que a proposta pode ser aplicada, mas não necessariamente apresenta alguma avaliação de sua efetividade.
- Experiência ou relato da indústria (Experience/Industrial Report): O artigo descreve a utilização da proposta de método, prática, técnica ou ferramenta por uma equipe real de desenvolvimento de software, e inclui algumas medidas ou opiniões que ajudam a entender a utilidade da proposta.
- Experimentação (Experimentation): O artigo avalia a proposta de método, prática, técnica ou ferramenta em algum nível de detalhe através de estudo experimental, como por exemplo, estudo de caso, observação de desenvolvedores através de métodos com rigor científico ou comparação experimental com abordagem de controle. Inclui medidas e análises acerca dos resultados em um contexto específico.

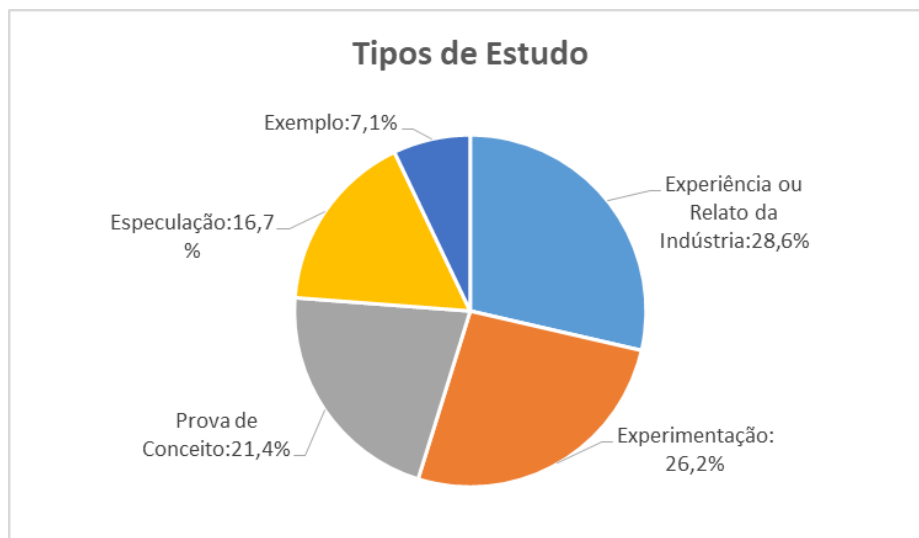


Figura 3. Distribuição dos tipos de estudo do conjunto de artigos

As propostas foram também classificadas quanto ao objetivo de apoiar a inovação (sim/não), contabilizando 28 artigos do total de 42. Por fim, classificou-se o contexto de origem (no caso de apresentar evidência especulativa ou com exemplo) ou de utilização das propostas (no caso dos outros tipos de evidência), de acordo com as categorias: Academia (*Academy* - propostas/estudos avaliados em ambientes de pesquisa acadêmica), Academia-Educação (*Academy-Education* - proposta aplicada em sala de aula, para fins de ensino), Parceria Academia-Indústria (*Academy-Industry*

*Partnership* - proposta aplicada na indústria com intervenção de pesquisadores acadêmicos) e Indústria (*Industry* - propostas e observações da indústria). A Figura 4 mostra a distribuição das categorias identificadas nos artigos.

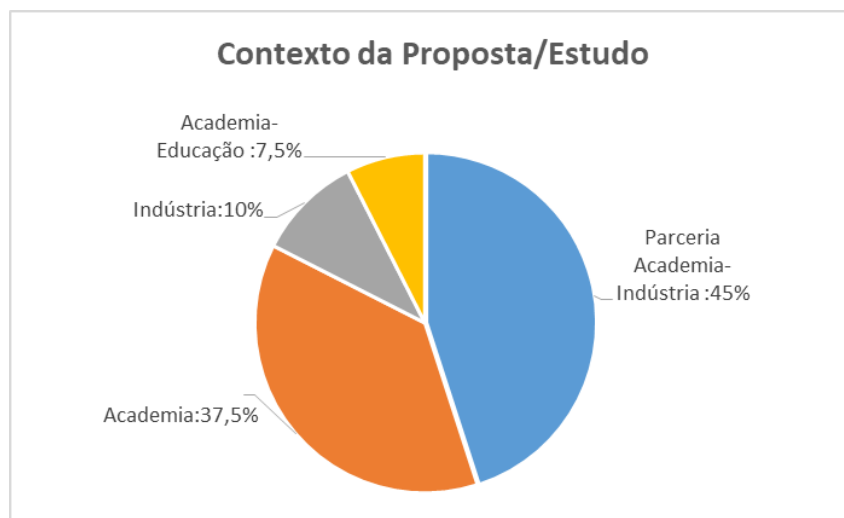


Figura 4. Distribuição do contexto das propostas/estudos

### 3.2.5 Procedimento de análise dos artigos selecionados

Para sistematizar a análise dos artigos selecionados optou-se pela execução de análise temática, que é um método de análise qualitativa para identificação, análise e relato de padrões existentes em um conjunto de dados (BRAUN e CLARKE, 2006), (CRUZES e DYBA, 2012), (SALDAÑA, 2015). As etapas do procedimento de análise, listadas a seguir, foram definidas com base nas etapas descritas para análise temática em (BRAUN e CLARKE, 2012) e síntese temática em (CRUZES e DYBA, 2012), sem executar as atividades de síntese devido a característica abrangente do mapeamento sistemático executado.

1. **Imersão:** Esta etapa envolveu a leitura e releitura dos artigos com objetivo de se “familiarizar” com os dados (BRAUN e CLARKE, 2012). Na primeira leitura completa dos artigos ainda durante a etapa de seleção, foi realizada a extração dos dados e obteve-se uma percepção inicial de quais eram os aspectos considerados pelos diferentes pesquisadores em relação aos problemas envolvendo desenvolvimento de software a partir de ideias de características, ao invés de demanda por contrato. Os dados extraídos foram revisados e comparados quanto ao nível de detalhamento dos dados extraídos, sendo muitas vezes necessário retornar aos artigos originais para avaliar se as diferenças de detalhamento se deram por motivos de omissão de informação nos



artigos ou falha de percepção da pesquisadora durante a leitura, especialmente entre artigos que pareciam tratar de aspectos similares de problemas. Ao final da etapa foi elaborada a estrutura inicial para a etapa de codificação (vide Quadro 4), dando foco para a atividade de codificação conforme as perguntas de pesquisa.

Quadro 4. Estrutura inicial de categorias para codificação

<b>Categoria</b>	<b>Questão de Pesquisa Relacionada</b>
<b><i>Proposal Goal</i></b>	<b>Q1 (Questão Principal)</b> - Que apoio é oferecido pelos métodos, técnicas, práticas e ferramentas propostos para os projetos de desenvolvimento de software com criação de ideias de características e/ou descoberta de requisitos de software?
<b><i>Proposed Support</i></b>	
<b><i>Main Contribution Type</i></b>	
<b><i>Proposal Evidence Type</i></b>	<b>Q1.2</b> Os resultados das propostas foram observados através de que tipos de estudo?
<b><i>Context Information</i></b>	
<b><i>Registered Information</i></b>	<b>Q1.1</b> As propostas apoiam o registro de informações sobre características de software? Quais e como as informações são registradas?
<b><i>Registration Manner</i></b>	
<b><i>Input Information</i></b>	
<b><i>Idea Creator</i></b>	

2. **Codificação:** Nessa etapa, foram identificados nos artigos os segmentos do texto com informações potencialmente relevantes para as perguntas de pesquisa, como por exemplo, conceitos, resultados e observações dos pesquisadores e conforme as categorias definidas na etapa anterior. Os segmentos foram assinalados com códigos de texto definidos para sumarizar a informação, fazendo uso dos termos utilizados em cada artigo para que fossem semanticamente próximos à essência da informação contida no texto, na medida do possível. Com o apoio da ferramenta de análise qualitativa de dados MAXQDA<sup>1</sup>, a codificação de trechos de cada artigo era comparada com a codificação dos artigos previamente codificados e tomava-se a decisão de unir códigos ou dividir códigos, tomando-se cuidado para em ter códigos com nível de abstração

<sup>1</sup> <http://www.maxqda.com/>

adequado para responder as perguntas de pesquisa sem perder informação relevante do texto. A Figura 5 mostra uma parte dos códigos gerados inicialmente para a categoria de objetivos de apoio. Os códigos foram definidos na língua inglesa para manter a consistência com a língua de origem dos artigos que são as fontes de dados do estudo.

Code System	Count
Code System	747
Main Contribution Type	68
Requirements Creation/Discovery Support	154
Proposal/Study Goals	0
Promote Stakeholder Participation in Creation	2
Involve Stakeholder in Specification	1
Integrate Customer in Develpm. Process	2
Acquire Scattered Requirements	1
Increase individual Creativity	3
Facilitate Stakeholders Participation in Idea ...	2
Discover Requirements	2
Discover Unknown Requirements	10
Elicit System Sub-goals	3
Create Requirements Ideas for New Technology	8
Innovation	0
Increase Innovative Idea Realization Lik...	2
Identify Opportunity	1
Support Ideation Process	2
Foster Innovative Ideas Creation	4
Create New Market	2

Figura 5. Exemplo de Códigos inicialmente gerados

- Definição de temas:** Os códigos gerados para cada categoria foram então analisados para identificação de temas que expressassem padrões existentes nos dados e que fossem significativos para as perguntas da pesquisa. Assim como descrito em (BRAUN e CLARKE, 2012), essa etapa consistiu tanto em dividir quanto em agrupar códigos que compartilhassem características relacionadas com a categoria que pertenciam. Durante essa etapa, observou-se que o conjunto de códigos da categoria *Proposal Goal* continham objetivos que na verdade eram desafios (*Challenges*) relacionados à tópicos (*Issues*) considerados nas soluções dadas pelas propostas para alcançar um ou mais objetivos principais (*Main Goals*). Alguns desses desafios foram originados das práticas (*Practices*) adotadas, que foram escolhidas segundo o que os autores acreditavam ser importante para o objetivo (mapeados na categoria *Beliefs*). A estrutura de categorias do mapeamento foi atualizada com essas novas categorias, o que tornou necessário rever nos textos de origem os códigos anteriormente mapeados como *Proposal Goal* para entender quais representavam os objetivos principais das

propostas, quais eram relacionados aos desafios tratados pelas propostas e quais eram práticas. Além disso, foram identificadas as motivações para a adoção das práticas (*Beliefs*). Após a revisão de códigos seguiu-se a atividade de análise e definição de temas para as categorias atualizadas e criadas.

4. **Revisão de temas:** Os temas definidos na etapa anterior foram revisados, iniciando-se pela releitura dos códigos dos temas e seus segmentos de texto relacionados para revisão da coerência dos códigos pertencentes ao tema, assim como o quanto o tema expressa um padrão. Em seguida, revisou-se a coerência entre as hierarquias de temas para os casos em que foram criados subtemas, tanto em relação ao nível de abstração dos temas/subtemas quanto a possíveis sobreposições entre temas de mesmo nível. É importante frisar que nessa etapa o foco não é manter os temas, mas sim garantir que os temas selecionados de fato sejam referentes aos padrões dos dados dos artigos e possibilitem analisar o conjunto de artigos em um nível de abstração adequado para os objetivos de pesquisa (BRAUN e CLARKE, 2012). Por isso, durante a revisão alguns temas foram considerados inadequados e revisados, outros descartados, tornando necessário voltar algumas vezes à etapa anterior.

### 3.3 RESULTADOS

Nessa seção são discutidos os resultados do mapeamento sistemático relacionados à pergunta principal de pesquisa, obtidos pela identificação dos objetivos de apoio das propostas e desafios observados. Os temas resultantes da análise temática foram escritos na língua inglesa, a mesma da escrita dos artigos selecionados, mantendo assim a semântica dos temas mais próxima dos artigos de origem.

Os objetivos principais de apoio propostos no conjunto de artigos do mapeamento são prover soluções para a coleta de ideias (*Idea Gathering*) de requisitos e características de software, para auxiliar a geração de ideias (*Idea Generation*), a descoberta de requisitos (*Requirements Discovery*), para aumentar as chances de desenvolvimento de ideias inovadoras (*Likelihood Increase of Innovative Idea Realization*), promover a inovação a longo prazo (*Long-term Innovation*), auxiliar a seleção de aplicações e características para desenvolvimento (*Selection of Applications and Features to be developed*) e apoiar a validação de ideias de requisitos para o software (*Validation of Software Requirements Ideas*).

Para cada objetivo de apoio, diferentes desafios foram observados nas propostas, que apresentam ferramentas, métodos ou práticas para lidar com esses desafios. Portanto, além de observados, são desafios que as propostas lidaram de alguma forma para atingir os objetivos principais.

### 3.3.1 Tópicos e Desafios (*Issues and Challenges*)

Os desafios aos objetivos principais de apoio das propostas foram agrupados em tópicos (*Issues*) elaborados pela execução de análise temática, sendo que alguns possuem ainda subtópicos. Os tópicos e desafios são detalhados a seguir.

#### **Tópico: Colaboração:**

O tópico Colaboração (*Collaboration*) engloba subtópicos relacionados à colaboração de pessoas internas ou externas à organização. Obter colaboração com pessoas externas com perfil representante dos possíveis usuários do produto criado a partir de ideias auxilia na diminuição das incertezas acerca das características do produto. Porém, a equipe técnica e os representantes de usuários frequentemente terão diferentes perspectivas, o que leva à necessidade de atentar para o alinhamento de visão e expectativas das ideias para o produto para melhor aproveitamento da colaboração (P4, P35, P36, P40). Por exemplo, enquanto os representantes de usuários podem estar preocupados com suas necessidades específicas, a equipe técnica pode ter a expectativa de ideias com desafios tecnológicos para inovação (P4).

Um ganho esperado pela colaboração com representantes de usuário é a oportunidade de explorar diferentes perspectivas de soluções, que podem levar às ideias inovadoras (P14, P16). Porém, tem-se o desafio de obter essas ideias sem influenciar os colaboradores com ideias pré-concebidas.

O conhecimento dos representantes de usuário acerca das tecnologias contemporâneas ou sobre as potencialidades da tecnologia também influenciam nas possibilidades de contribuições obtidas pela colaboração (P4, P14, P31, P40), sendo necessário lançar mão de métodos, práticas e/ou técnicas para aumentar as chances de obter ideias para as características do software. O domínio de aplicação de sistemas de apoio para pessoas idosas é um exemplo de domínio com possibilidades para desenvolver sistemas de software inovadores em que grande parte dos potenciais usuários tem pouca experiência com tecnologias contemporâneas (P14).

De acordo com (P6), é importante se preocupar com a efetividade de comunicação com representantes de usuário para o sucesso da colaboração, pois eles podem ter dificuldade de explicitar ideias de forma clara ou dificuldade de organizar conhecimento sob demanda durante as atividades de colaboração. Por fim, a participação ativa dos representantes de usuários é imprescindível para que se tenha

ganhos com a colaboração. Foram citados como desafios (P2, P6, P14, P18, P28, P35): a necessidade de lidar com a diversidade de perfis dos envolvidos e seus diferentes interesses, estimular a participação ativa das pessoas e conseguir a colaboração de potenciais usuários de localidades distribuídas. A falta de métodos que apoiem a participação de múltiplos *stakeholders* em atividades de colaboração também foi citada como fator que dificulta a colaboração.

A colaboração não é referente apenas às pessoas que não são da organização. Captar as ideias e aproveitar o conhecimento das pessoas internas à organização aumenta o potencial da organização desenvolver produtos inovadores (P3, P4, P8, P39), sendo citados dois temas relacionados: a participação dos empregados das diversas áreas/departamentos da organização nas atividades de ideação e o envolvimento da equipe técnica de desenvolvimento de software na criação de ideias de requisitos. Para o primeiro, os desafios são o de estimular as pessoas à contribuírem com a ideação, além de competir com as atividades de desenvolvimento a curto-prazo (como por exemplo, atendimento de demandas, correção de defeitos, atendimento ao cliente, entre outros). Em relação à criação de ideias de requisitos pela equipe técnica, em (P4) tem-se o relato da resistência da equipe técnica em contribuir com requisitos, estando acostumados a receber requisitos e criar as soluções técnicas.

#### **Tópico: Interação em grupo (*Group Interaction*):**

Dois tópicos citados pelos artigos são referentes aos desafios para gerar ou coletar ideias a partir da interação de pessoas trabalhando em grupo: os efeitos sociais (P6, P16, P21) e a eficiência das discussões em grupo (P2, P5, P21, P38, P39).

Embora a interação de pessoas em grupo seja vista como benéfica para criar ideias e coletar opinião, os seguintes efeitos sociais da dinâmica de interação podem prejudicar o alcance do objetivo: a falta de atenção devido à natureza multitarefa da geração de ideias em grupo - cada participante deve analisar o assunto discutido, ouvir e avaliar as ideias apresentadas, combinar ideias ou evoluir as ideias apresentadas; a inércia cognitiva causada pela falta de profundidade de discussão quando ideias são apresentadas de forma sequencial; a dominação da discussão por uma única pessoa ou subgrupo de pessoas; a situação de algumas pessoas confiarem nas ideias de outras do grupo e assim não explorarem suas próprias ideias; a situação em que a opinião de pessoas que falam “mais alto” são repassadas como opinião do grupo; a dificuldade de algumas pessoas expressarem suas ideias na presença de estranhos ou de pessoas com posições hierárquicas mais altas; o receio das pessoas de receber críticas na presença do grupo; a falta de tempo ou oportunidade suficiente para as pessoas

refletirem e integrarem suas ideias; e o esquecimento ou omissão de ideias causados pelo fato de que apenas uma pessoa por vez deve apresentar sua ideia.

Em relação à eficiência da interação em grupo na discussão, os artigos citam os desafios de evitar a geração excessiva de ideias consideradas criativas, porém irrelevantes, evitar discussões sem foco, conseguir gerar ideias que possam ser imediatamente aplicáveis (ao invés de relevantes, mas num tempo futuro) e também aumentar a geração de ideias criativas.

### **Tópico: Ideia (*Idea*):**

O tema Ideia contém os tópicos de problemas com desafios relacionados às características específicas das ideias. Em (P2) é relatada a preocupação com a qualidade da descrição de ideias coletadas, prejudicada, segundo os autores, pela falta de formato padrão para sua descrição. Por sua vez, os artigos (P16) e (P42) descrevem a preocupação com o potencial de inovação das ideias, tendo como desafio ter ideias que não sejam apenas refinamentos ou reclamações sobre requisitos já desenvolvidas. Outro tópico é a sobrevivência de ideias para desenvolvimento, relatado nos artigos que descrevem a importância de não deixar que boas ideias sejam esquecidas ou suprimidas. Nesse sentido, (P26) discute a necessidade de difundir ideias para que alcancem pessoas interessadas, tendo como desafios tornar as ideias compreensíveis de forma a atrair atenção, identificar como compartilhar ideias, motivar as pessoas a compartilhar suas melhores ideias e encontrar colaboradores para refinar e evoluir ideias. Ter tempo apropriado para expressar as ideias também é um desafio relacionado à sobrevivência de ideias (P39).

### **Tópico: Prática de Ideação (*Ideation Practice*):**

O tema Prática de Ideação aborda assuntos relacionados com a execução de práticas para ideação nas organizações. O investimento de recursos na geração de ideias para inovação é abordado em (P3) como algo importante para que organizações sejam inovadoras, tendo como desafio competir por recursos com o que os autores chamam de “desenvolvimento reativo” (desenvolvimento para atender demandas de clientes). O gerenciamento de atividades de ideação é um tópico apontado em (P2), que cita como desafio o rastreamento do progresso de uma ideia sugerida. Já o artigo (P22) relata a preocupação com a diminuição dos custos de reuniões de ideação.

### **Tópico: Criatividade Individual (*Individual Creativity*):**

Considerada fundamental para inovação (P1), descoberta e invenção de requisitos para sistemas baseados em computação, criatividade é definida como a

habilidade de um indivíduo ou grupo de pessoas ter ideias que sejam novas e também úteis (AMABILE, 1998), (STERNBERG, 1999). Dessa forma, algumas propostas do mapeamento chamam a atenção para a importância de apoiar indivíduos na superação de desafios pessoais de criatividade, sendo citados os problemas da perspectiva adequada de pensamento (P6, P17), cujos desafios estão em chegar na perspectiva do problema a ser explorado e também conseguir ter ideias usando a perspectiva de outras pessoas (no caso de ideias para requisitos, as perspectivas dos usuários pretendidos devem ser consideradas). Outros desafios citados como importantes para ter a habilidade de pensamento criativo são (P17, P34, P39): a superação da autocrítica, conseguir evitar temporariamente o pensamento controlado, racional e dirigido à objetivos, e superar o chamado “bloqueio de escritor”, que se refere à dificuldade de ter ideias em um determinado momento.

#### **Tópico: Conhecimento (*Knowledge*):**

O tema Conhecimento engloba preocupações relacionadas ao conhecimento de domínio do problema para o qual se está criando ideias de características de software e também conhecimento acerca das preferências dos usuários pretendidos. Sobre o domínio do problema, os desafios estão em lidar com a falta de conhecimento sobre o contexto de uso futuro do produto sendo criado, em descobrir conhecimento relevante sem saber qual conhecimento precisa saber (SUTCLIFFE e SAWYER, 2013) e obter conhecimento diversificado de diferentes pessoas (P7, P18, P21, P30, P32, P35, P40). Sobre o conhecimento de preferências dos usuários pretendidos, os desafios são identificar que funcionalidades do produto de software eles usariam e por quê usariam, além de lidar com a falta desse conhecimento (P15, P27, P35).

#### **Tópico: Seleção de Ideias (*Selection of Ideas*):**

Para a seleção de ideias, os artigos apontaram como tópico de interesse a filtragem das ideias candidatas, possuindo como desafios a identificação de ideias que apoiem as estratégias de negócio da organização, a identificação de ideias com potencial de render maior lucro e a identificação do que são e quais são as melhores ideias disponíveis (P3, P26, P27, P39).

#### **Tópico: Processo de Desenvolvimento de Software (*Software Development Process*):**

A adequação dos modelos de processos de desenvolvimento de software foi abordada como uma preocupação envolvida no desenvolvimento de software a partir de ideias (P9, P10, P12, P19, P20, P23, P24, P29, P36, P41). Um desafio citado é que as

atividades dos processos de desenvolvimento apoiem a coexploração do domínio de problema e do domínio de solução (P36) de forma iterativa, permitindo que a elaboração de soluções possibilite descobrir áreas do domínio do problema que precisam ser exploradas. O apoio ao desenvolvimento de sistemas de informação inovadores que possam ser modificados rapidamente, assim como reagir de forma rápida e criativa às mudanças, são citados em (P10) como desafios importantes para a definição de processos de desenvolvimento de software. Também foi mencionada a falta de apoio para execução de atividades criativas para ideação de características para o software (P12, P19, P20, P24, P29, P41), assim como geração de requisitos quando não se tem cliente ou quando o cliente não sabe quais são os requisitos (P9, P19, P23, P24).

Outra questão mencionada é quanto a aplicação de técnicas de criatividade em atividades de Engenharia de Requisitos para promover o surgimento de ideias inovadoras ou descobrir requisitos. O artigo (P33) relata como desafio a escolha de qual técnica utilizar, enquanto que o desafio de integrar as técnicas de criatividade e seus resultados com o fluxo do processo de Requisitos é mencionado em (P19, P20, P24, P29).

O Quadro 5 lista os tópicos (primeira coluna, escritos com letra maiúscula) e subtópicos (sublinhados) citados, além de mostrar para quais objetivos de apoio os temas e desafios foram citados. Como já mencionado, os temas foram codificados na língua Inglesa para manter a semântica mais próxima dos artigos analisados no mapeamento sistemático.

Uma análise inicial da relevância dos tópicos foi realizada utilizando os seguintes critérios: o tópico foi citado por mais de um artigo ou o tópico foi observado no contexto real da indústria de software ou o tópico foi relatado em artigo que apresenta algum estudo experimental. Dois tópicos não atenderam a nenhum desses critérios (destacados no Quadro 1): *Integration with Development Process* e *Quality of Idea Suggestions Descriptions*. Por outro lado, os tópicos com desafios relacionados à Colaboração (*Collaboration*) e Conhecimento (*Knowledge*) são os que mais foram observados em diferentes objetivos de apoio, o que é coerente com a característica de alta incerteza do contexto de desenvolvimento a partir de coleta e geração e ideias e descoberta de requisitos.



Quadro 5. Objetivos e tópicos relacionados (Goals x Issues)

	Idea Gathering	Idea Generation	Likelihood of Increase of Innovative Idea Realization	Long-term Innovation	Requirements Discovery	Selection of Applications and Features to be developed	Validation of Software Requirements Ideas
<b>COLLABORATION</b>	X	X			X	X	X
<u>Collaboration with Users Representatives</u>							
Alignment of Views and Expectations		X			X		X
Effectiveness of Communication		X			X		
<b>Integration with Development Process</b>	X					X	X
Potential to Explore Different Perspectives		X					
Technology Knowledge of User Representative		X			X		
User Representative Enthusiasm with Modern Technology		X					
User Representative Participation	X	X				X	X
Employees Participation in Ideation Tasks	X	X			X		
Involvement of Technical Team in Requirements Idea Creation		X					
<b>GROUP INTERACTION</b>	X	X					
Social Effects	X	X					
Effectiveness	X	X					
<b>IDEA</b>	X	X	X				
<u>Idea Survival to Realization</u>							
Diffusion of Ideas			X				
Introduction of Ideas in Meetings		X					
Innovative Potential of Ideas	X						
<b>Quality of Idea Suggestions Descriptions</b>	X						

	Idea Gathering	Idea Generation	Likelihood of Increase of Innovative Idea Realization	Long-term Innovation	Requirements Discovery	Selection of Applications and Features to be developed	Validation of Software Requirements Ideas
<b>IDEATION PRACTICE</b>	X	X		X			
Costs of Meetings		X					
Ideation Management	X						
Resource Investment in Innovative Idea Generation				X			
<b>INDIVIDUAL CREATIVITY</b>		X					
Creative Thinking Ability		X					
Achievement of a Thought Perspective		X					
<b>KNOWLEDGE</b>		X			X	X	X
Domain Knowledge		X			X		X
Knowledge about user preferences		X			X	X	X
<b>SELECTION OF IDEAS</b>			X	X		X	
Appropriate Filtering of Ideas Candidates			X	X		X	
<b>SOFTWARE ENGINEERING PROCESS</b>		X			X		
Suitability of Software Development Process Models		X			X		
Use of Creativity Techniques in Requirements Engineering		X			X		

### 3.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Os desafios citados nos artigos selecionados são indícios de diferentes perspectivas de problemas envolvidos no apoio ao desenvolvimento de software baseados em ideias e descoberta de requisitos ao invés de desenvolvimento convencional (sob demanda). Porém, nenhum dos objetivos e desafios citados nas propostas de apoio se referem ao registro de ideias e características de software, com exceção do tópico Qualidade da Descrição das Sugestões de Ideias (*Quality of Idea Suggestions Descriptions*), considerado irrelevante durante a análise de relevância e pertinência dos tópicos.

A falta de propostas com objetivos para os desafios no tema de pesquisa desta tese indicou a possibilidade de lacuna de pesquisa, porém se fez necessário obter a percepção da prática sobre isto. A pesquisa seguiu, então, com estudo de caracterização em organizações Startups de desenvolvimento de software para inovação, relatado no próximo capítulo.

# 4 ESTUDO EXPLORATÓRIO: DOCUMENTAÇÃO DE IDEIAS E CARACTERÍSTICAS DE SOFTWARE EM STARTUPS.

Neste capítulo é apresentado o estudo executado de caracterização das práticas relacionados ao registro de informações sobre as ideias e características de software no contexto de organizações Startup de software que são de interesse da tese: as que desenvolvem software com objetivo de inovação e de criação de modelo de negócio lucrativo e escalável.

## 4.1 INTRODUÇÃO

A partir da observação empírica na empresa Beta (descrito na introdução deste trabalho) e também dos resultados do estudo de (GIARDINO, PATERNOSTER, *et al.*, 2015) e das características reportadas na literatura técnica, as seguintes suposições foram consideradas para serem analisadas no estudo sobre Startups de software:

- **Suposição 1:** O esforço com documentação para registro de ideias e características de software é reduzido no início do desenvolvimento de software com objetivo de inovação para favorecer a produtividade.
  - Motivos:
    - Alto grau de incerteza sobre a pertinência das ideias e características do software
    - Demanda por menor *time-to-market*
  - **Suposição 2:** A inexistência de registro, ou o registro insuficiente das características de software, é uma fonte de risco à produtividade das atividades de evolução e manutenção do software com objetivo de inovação quando ocorre alguma das causas abaixo:
    - Eventos relacionados ao risco:
      - Aumento da quantidade e complexidade das características do software.
      - Inclusão de novas pessoas na equipe de desenvolvimento.
      - Saída de pessoas da equipe de desenvolvimento, havendo perda de conhecimento sobre as características do software.
    - Impactos:
      - Tempo de aprendizagem dos novos integrantes contribuir com a diminuição do *time-to-market* além do previsto/aceitável.

- Maior tempo de desenvolvimento causado pela falta de informação explícita sobre as características existentes no software.

O objetivo do estudo é detalhado no Quadro 6, seguindo a estrutura GQM (BASILI, CALDIERA e ROMBACH, 2001)). A partir dos resultados pretende-se elaborar um corpo de conhecimento inicial a ser avaliado e evoluído por meio de estudos posteriores, visando fundamentar a elaboração de apoio para de decidir sobre adoção de práticas e tecnologias para documentar características de software.

Quadro 6. Objetivo do Estudo Exploratório

<b>Analisar</b>	Projetos de desenvolvimento de software para inovação
<b>Com o propósito de</b>	Caracterizar
<b>Em relação à</b>	Práticas e desafios de registro de características de software.
<b>Do ponto de vista</b>	Participantes da equipe de desenvolvimento de software, incluindo responsáveis pela decisão sobre práticas aplicadas para o registro de características do software, exercendo papéis de gerência na equipe.
<b>No contexto de</b>	Organizações Startups que desenvolvem software em cujos contextos: <ul style="list-style-type: none"> <li>• <u>Ideias são a origem das características do software</u>, ao invés de usuários pertencentes a um contexto cujas demandas software são afirmadas em contrato. Há, portanto, incerteza da pertinência das características desenvolvidas para os pretendidos usuários;</li> <li>• Existe o <u>objetivo de inovação</u> com desenvolvimento de software que tenha características diferentes ou significativamente melhores em relação ao que existe no mercado/contexto pretendido.</li> <li>• O <u>desenvolvimento é iterativo e incremental</u>, com disponibilização de versões do produto e uso de estratégias de experimentação no mercado para avaliação da pertinência do produto e coleta de novas ideias.</li> </ul>

Foram planejadas duas etapas para o estudo: 1) caracterização inicial do contexto das organizações através de questionário online, observando as suposições de pesquisa; 2) entrevista semiestruturada, com instrumento elaborado à luz do conhecimento obtido pela caracterização inicial das organizações (resultado da etapa anterior). As subseções a seguir detalham estas atividades.

## 4.2 ELABORAÇÃO DO INSTRUMENTO

A elaboração do questionário seguiu as etapas de criação do instrumento inicial, revisão e execução de pilotos, como mostra a Figura 6.

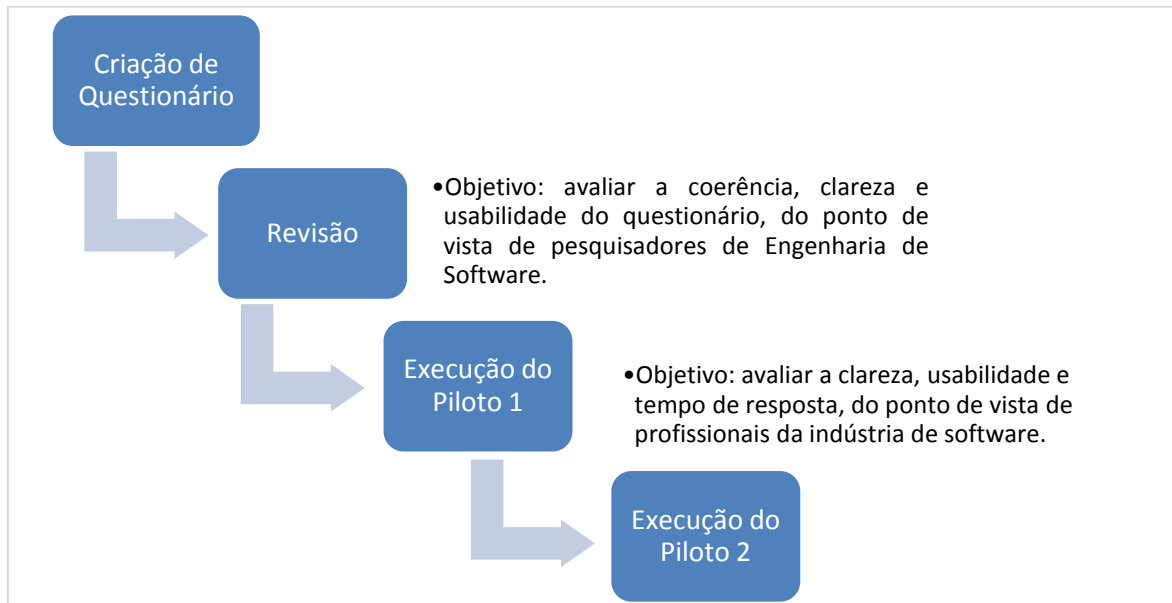


Figura 6. Etapas de Desenvolvimento do Questionário

Após a definição da primeira versão do questionário e ajustes conforme a revisão do pesquisador orientador da tese, o questionário foi transcrito na ferramenta *LimeSurvey*<sup>2</sup>, escolhida para disponibilizar o questionário para a população-alvo devido a sua estabilidade, uso anterior em outras pesquisas do grupo ESE, e sua disponibilidade no servidor LENS-ESE do Laboratório de Engenharia de Software. Seguiu-se então a execução de revisão com acadêmicos de engenharia de software e dois estudos pilotos.

### 4.2.1 Revisões e execuções-piloto do instrumento

A revisão teve a participação de dois estudantes de doutorado e um de mestrado, todos da linha de engenharia de software do programa de Sistemas e Computação da COPPE/UFRJ. O questionário foi enviado online para os participantes em sua versão no *LimeSurvey*, da forma que seria executado com a população-alvo do estudo. As sugestões de correções e melhorias foram enviadas por *e-mail*, em texto-livre contendo indicação de partes do questionário com frases ou termos dúbios, sugestões de inclusão de explicações e exemplos em algumas questões e também sugestões de melhorias de

<sup>2</sup> <https://www.limesurvey.org>

apresentação do questionário, relacionadas à usabilidade provida pela ferramenta utilizada.

O primeiro estudo piloto foi realizado com dois participantes com mais de dez anos de experiência de desenvolvimento de software sob demanda – um como desenvolvedor e gerente de projeto em empresa própria, e o outro participante com experiência como líder de projeto, analista de sistemas e desenvolvedor em empresa pública. O questionário foi enviado de forma online usando o *LimeSurvey*. Os respondentes indicaram terem compreendido claramente as questões, tendo apenas um ponto de melhoria na apresentação das opções de uma questão, que causou confusão nas respostas.

O tempo de resposta foi um dos pontos analisados durante o piloto, pois houve preocupação em diminuir a chance de desistência de contribuição pelos indivíduos contatados. Durante o primeiro piloto, um respondente levou 20 minutos para responder ao questionário, enquanto que o outro precisou de 15 minutos. Após revisão do instrumento, decidiu-se manter os itens de questões existentes para conseguir informações mínimas que apoiassem a análise dos dados frente aos objetivos do estudo. Um cuidado que se teve após comentários dos participantes foi o de disponibilizar opções de resposta para seleção sempre que possível, ao invés de requerer a escrita da resposta.

O segundo estudo piloto foi realizado com dois participantes de equipes de desenvolvimento de software de duas empresas *Startups*, sendo um desenvolvedor estagiário, com formação em Engenharia de Automação, e o outro com formação em Sistemas de Informação que desempenha os papéis de analista de requisitos, desenvolvedor e testador em projeto no qual as pessoas da equipe não tem papel específico definido em relação às atividades de desenvolvimento de software (nas palavras do participante, “*todos fazem tudo*”). Os participantes também foram entrevistados para obtenção de informações da experiência de responder o questionário. O participante estagiário mostrou ter tido dificuldade em entender as perguntas contendo termos de práticas e abordagens de engenharia e de gestão, levando cerca de 35 minutos para concluir o questionário, porém a sua formação e inexperiência foram considerados fatores que contribuíram para isso. Já o participante mais experiente respondeu o questionário em dez minutos e contribuiu para a melhoria no instrumento ao indicar termos e frases de duplo sentido nas questões e opções de respostas, além de evidenciar que algumas questões pensadas para serem obrigatórias não cabiam a todas situações.

#### 4.2.2 Estrutura do instrumento

O instrumento final foi organizado em três seções de perguntas: caracterização da organização, caracterização do respondente, e informações do projeto de inovação que o respondente participa na organização, descritas sucintamente a seguir (o instrumento completo está disponível no Apêndice 2).

A seção de caracterização da organização apresenta perguntas para a identificação do tempo de atividade e porte da organização e domínio de aplicação do software desenvolvido. Nessa seção também é avaliado se a organização do respondente possui em seus projetos as características de interesse para a pesquisa (descritas na seção anterior deste capítulo), observadas pela frequência com que os objetivos listados a seguir são determinantes nas tomadas de decisão dos projetos (opções de resposta - *Likert*: Sempre, Quase Sempre, Raramente, Nunca, Não sei dizer, Não aplicável).

- a. Oferecer produto ou serviço com características inexistentes em outros produtos/serviços e que chamem a atenção dos clientes/usuários pretendidos.
- b. Disponibilizar o produto completo o mais cedo possível no mercado/contexto pretendido.
- c. Disponibilizar versões do produto o mais cedo possível, mesmo que não tenha ainda todas as características planejadas para o produto final.
- d. Coletar ou ter ideias para o produto a partir da sua utilização pelos usuários.

As questões da seção de caracterização do respondente foram elaboradas para capturar a formação dos respondentes e a sua experiência em atividades de gerenciamento e do ciclo de desenvolvimento de software (análise e especificação de requisitos, projeto de sistemas, codificação, teste, gerenciamento de projeto de software), de forma a permitir analisar se o respondente tem conhecimento sobre os conceitos envolvidos nas questões sobre os projetos de software para inovação, além de possíveis influências da formação e experiência da equipe nas práticas aplicadas e na percepção de desafios nos projetos de software. Ainda nessa seção, pede-se para o respondente descrever quais atividades desempenha nos projetos de inovação que participa atualmente.

Por fim, a última seção do questionário solicita informações sobre os projetos atuais de inovação que o respondente participa na organização (ou seja, à época que responder ao questionário), como detalhado a seguir.



A importância dos projetos de inovação para as estratégias de negócio da organização é coletada com o objetivo de analisar qual a prioridade do projeto de inovação para a organização, sendo possível ao respondente escolher uma das opções: i) faz/fazem parte das estratégias de negócio centrais da empresa; ii) é/são importante(s) mas não é/são parte das estratégias de negócio centrais da empresa; iii) tem baixa prioridade em relação aos outros projetos da empresa; e iv) Outros (com espaço para o respondente detalhar).

Para apoiar a análise das suposições 1 e 2 da pesquisa, o instrumento contém item para capturar situações que indicam o estágio do projeto, adaptado de cujas opções representam níveis decrescentes de incerteza quanto às características do produto de software, adaptado de (GIARDINO, PATERNOSTER, *et al.*, 2015):

- Refinando a ideia de negócio, formando a equipe e iniciando o projeto;
- Prototipando ou construindo a primeira versão do produto;
- Avaliando a aceitação do produto após as primeiras vendas/adesões das primeiras versões;
- Adaptando o produto para as necessidades dos clientes que já aderiram ao produto;
- Investindo ativamente no aumento de usuários;
- Adicionando produtos complementares, expandindo o produto ou diversificando o portfólio para entrar em novos mercados/contextos;
- Dando suporte aos usuários atuais e sem tempo ou recursos para crescer ou expandir de forma intensa;
- Devido ao aumento da equipe e/ou da complexidade do produto, estamos melhorando ou adotando práticas de gerenciamento e desenvolvimento, e;
- Outros (o respondente pode detalhar).

Para saber se alta produtividade é requerida nos projetos de desenvolvimento, o que também contribui para a análise das suposições 1 e 2 da pesquisa, o questionário contém um item para a indicação das razões da cobrança por menor *time-to-market* para disponibilizar o produto no mercado, caso exista, com as opções: Incerteza sobre a aceitação do produto no mercado ou contexto de uso pretendido; Limitação de recursos financeiros - necessário obter ganhos para continuar a desenvolver o produto; Pressão do investidor; Risco de ser ultrapassado pela concorrência e Outros.

Considerou-se importante também obter informações sobre práticas de desenvolvimento e gestão para auxiliar a análise das práticas para registro de ideias e

características de software, assim como o entendimento do ciclo de desenvolvimento adotado.

Com o objetivo de confirmar a adequação do projeto de software que o respondente se refere ao critério da pesquisa sobre desenvolvimento de software a partir de ideias, o instrumento contém uma pergunta sobre qual é a origem das ideias para as características do software, sendo possível selecionar uma ou mais das seguintes opções: Análise de outros produtos, Análise de reclamações dos usuários do produto, Cliente patrocinador do projeto, Demanda de usuários do produto, Fundadores da empresa, Especialistas no mercado ou contexto pretendido, Equipe de desenvolvimento, Pesquisa de mercado, Pessoas da empresa mas que não são da equipe de desenvolvimento, Outros.

Para a observação de como as informações sobre ideias e características de software são registradas nos projetos, o questionário apresenta ao respondente um conjunto de opções com base nos resultados do mapeamento sistemático da literatura executado durante a pesquisa dessa tese, podendo indicar se cada opção é usada para registrar ideias, características de software, ambos ou não marcar nenhuma dessas opções. Também foi incluída no questionário uma pergunta sobre motivos para a redução de esforço em documentação no início do projeto, caso tenha ocorrido.

As duas perguntas finais do questionário são específicas da suposição 2, relacionada aos riscos que o registro insuficiente de ideias e características do software traz ao desenvolvimento do software. Com base nas observações empíricas da organização Beta, nos resultados do Mapeamento Sistemático e no resultado relatado em (GIARDINO, PATERNOSTER, *et al.*, 2015) foram elaboradas opções de resposta para seleção, mas é permitido incluir outras respostas.

A penúltima pergunta questiona se o respondente percebeu a ocorrência de aumento de esforço em documentação após o início do projeto e quais são os motivos disso ter ocorrido. São opções de resposta: Aumento da complexidade do software, o que aumentou o risco de inclusão de defeitos; Aumento de funcionalidades oferecidas pelo software; Dificuldade para incluir ou alterar funcionalidades e demais características do produto; Existência de diferentes versões do produto em operação; Inclusão de pessoas na equipe (necessidade de repassar informações); Saída de pessoas da equipe (necessidade de registrar informações); Outros.

Por fim, o respondente informa se considera necessário aumentar a qualidade ou quantidade de documentação do projeto de inovação que participa, indicando as razões para essa melhoria: Para alinhar o entendimento da equipe acerca da visão e objetivos do produto; Para o desenvolvimento de novas funcionalidades no software; Para alterar funcionalidades do software; Para executar testes no software; Para

planejar e/ou acompanhar as atividades de desenvolvimento da equipe; Para explicar as ideias e características do software para outra pessoa da equipe; Para decidir quais características e funcionalidades o software deveria ter ou quais deveriam ser excluídas; Outros.

### 4.3 EXECUÇÃO

A população do estudo foi definida por conveniência, obtida através de pesquisa sobre agências de fomento a empresas de inovação em software e indicações de contatos pessoais da pesquisadora, tanto da indústria quanto da academia. Durante o segundo trimestre de 2016, foram contatadas:

- A coordenação do programa de apoio financeiro e de infraestrutura para organizações *Startups* de inovação em software da agência de fomento FAPERJ (Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro). Não houve retorno ao contato.
- Uma empresa que coordena um espaço de trabalho disponível para diferentes *Startups*, além de prover financiamento e capacitação na área de negócios. O líder de projeto de uma das empresas residentes no espaço de trabalho divulgou o questionário para a sua equipe, porém, apesar de quatro acessos ao questionário, não houveram respostas.
- Seis profissionais de *Startups* de software distintas, dos quais dois responderam o questionário.
- Um grupo de discussão sobre desenvolvimento de software baseado em criatividade, organizado pelo SEBRAE (Serviço de Apoio às Micro e Pequenas Empresas). Na época, o grupo contava com representantes de 23 empresas (nem todos envolvidos com atividades específicas de desenvolvimento de software). Oito pessoas acessaram o questionário, mas não o responderam. Duas pessoas participantes do grupo de discussão foram contatadas através de e-mail e aceitaram cooperar com a pesquisa.
- A coordenação da Incubadora COPPE/UFRJ, com solicitação de apoio para contatar três das suas empresas residentes que foram selecionadas para o estudo após análise da descrição dos produtos e objetivos das empresas, obtidos em material de divulgação. Dois gerentes de duas das empresas selecionadas aceitaram participar do estudo e uma empresa não retornou o contato.

No total, seis profissionais de *Startups* de software participaram do estudo. Foram obtidas apenas duas respostas com envio de questionário online. Durante o contato com as empresas da Incubadora UFRJ, um representante afirmou preferir

explicar pessoalmente as práticas da empresa ao invés de responder o questionário online. Além dele, dois profissionais participantes do grupo de discussão foram contatados via *e-mail* e afirmaram que viram o convite no grupo de discussão, mas não se motivaram a responder o questionário *online*, preferindo serem entrevistados via videoconferência. A partir de então o método de coleta da pesquisa passou a ser o de **entrevista semiestruturada**, tendo o questionário como guia da entrevista, porém havendo flexibilidade para realizar perguntas não planejadas que surgissem devido a novas informações que possam surgir na entrevista ou para esclarecimento das respostas do entrevistado.

As entrevistas foram executadas em três etapas, seguindo o protocolo de entrevista descrito no Quadro 7. Inicialmente, os objetivos da pesquisa foram expostos aos entrevistados, tendo como preocupação esclarecer a perspectiva de interesse sobre as práticas aplicadas nos projetos de inovação (registro de ideias e características). Também se considerou importante enfatizar para os participantes o fato de não existir no contexto da pesquisa uma determinação de quais práticas são corretas ou inadequadas, e que a pesquisa visava descobrir e caracterizar o que funciona para as diferentes equipes de desenvolvimento de inovação. Isto foi enfatizado em vários momentos na entrevista, quando se percebia que os entrevistados respondiam como se o projeto e a equipe estivessem sendo julgados pela pesquisadora. Em seguida, a pesquisadora solicitava aos entrevistados que falassem brevemente sobre o produto de software em desenvolvimento com objetivo de inovação e as práticas aplicadas para registro de ideias e características de software. Por fim, os entrevistados foram solicitados a responder e comentar as perguntas do questionário elaborado.

Quadro 7. Estudo Exploratório – Protocolo de Entrevista

Etapa	Atividades
<b>Apresentação da Pesquisa (cinco a sete minutos)</b>	1 – Apresentar brevemente ao entrevistado o objetivo e as perguntas de pesquisa do estudo. 2 – Esclarecer o escopo de interesse do estudo e fornecer exemplos de características funcionais, arquiteturais e restrições de software. 3 – Fornecer exemplos de diferentes práticas citadas como de uso em Startups para registrar informações de software. Esclarecer ao entrevistado que não há expectativas de julgamento de respostas como certas ou erradas, visto que o objetivo é entender o contexto de Startups de Software. 4 – Comunicar ao entrevistado sobre a garantia de confidencialidade e anonimato da sua participação no estudo.
<b>Apresentação do entrevistado (cinco a dez minutos)</b>	1 – Solicitar ao entrevistado que fale brevemente sobre o projeto de software para inovação que ele participa. Obs.: Tomar nota e fazer perguntas no caso de o entrevistado explicitar informações de interesse ao estudo.

<b>Execução da Entrevista (cerca de trinta minutos)</b>	1 – Solicitar ao entrevistado que responda ao questionário e explique as suas respostas ou dê informações adicionais, caso ache necessário. Obs.: Observar se há inconsistências nas respostas e demais informações fornecidas. No caso de existirem, solicite esclarecimentos ao entrevistado.
---	--

## 4.4 RESULTADOS

Nas subseções a seguir são detalhadas a caracterização das organizações e seus participantes do estudo, assim como a caracterização das práticas que foram relatadas. Como para cada organização houve a participação de um respondente, a identificação usada no texto a seguir é a mesma para organizações.

### 4.4.1 Características das organizações e dos respondentes

O Quadro 8 apresenta as características de porte, ano de início de atuação e quantidade projetos que já foram executados em cada organização com objetivo de inovação. Dentre as seis organizações participantes, duas (D e F) se diferem em relação ao tempo de atuação no mercado e quantidade de projetos para inovação já executados, o que pode indicar maior nível de experiência com projetos de inovação.

Quadro 8. Estudo Exploratório - Perfil das organizações

<b>Id.</b>	<b>Porte da Empresa</b>	<b>Ano de Início de Atuação</b>	<b>Qtd. de Projetos visando Inovação que já executou</b>	<b>Região de Atuação</b>	<b>Método de Coleta da Pesquisa</b>
<b>A</b>	Grande	2014	1	Europa (dez países) e Canadá	Questionário
<b>B</b>	Pequena	2014	3	Rio de Janeiro/RJ	Questionário
<b>C</b>	Micro	2016	1	Rio de Janeiro/RJ	Entrevista
<b>D</b>	Micro	2010	5	Belém/PA	Entrevista
<b>E</b>	Micro	2015	1	Rio de Janeiro/RJ	Entrevista
<b>F</b>	Micro	1992	6	Belém/PA	Entrevista

As empresas A B, e D desenvolvem aplicativos de software de apoio a necessidades consideradas comuns para grande parte da sociedade, como por exemplo, alimentação e educação (vide listagem no Quadro 9). Por sua vez, a empresa C desenvolve aplicativo a ser oferecido como serviço de gestão de limpeza urbana, tendo como alvo específico de mercado órgãos públicos do Brasil, especialmente governos estaduais e prefeituras municipais. A empresa E visa oferecer soluções

tecnológicas de realidade virtual para o setor de Educação, enquanto a empresa F desenvolve aplicativos para o mercado de gestão de eventos e de microempresas.

Dentre as organizações, a empresa A é a única que já escalou no mercado, tendo criado um modelo de negócio que abrange mais de dez países. Assim, ela representa um caso especial para o estudo por ser uma instituição que já passou por diferentes estágios de evolução de Startups. Por sua vez, a empresa F apresenta uma característica especial em relação ao seu tempo de atuação. Ela foi fundada em 1992 como prestadora de serviços em informática, mas atende à definição de Startup nesta tese (discutida no capítulo 2) pois a mesma está desenvolvendo software com objetivo de inovação no mercado e também busca criar um modelo de negócio que sustente a empresa e seja escalável. Como esclarecido pelo entrevistado da empresa F, o objetivo da empresa é redirecionar todas as suas atividades para o projeto de inovação em software assim que o modelo de negócio seja sustentável. Os serviços oferecidos no mercado à época da entrevista são vistos como fonte de renda para financiar o projeto de inovação.

Quadro 9. Domínios de aplicação das organizações

<b>Id</b>	<b>Domínios dos produtos desenvolvidos</b>
<b>A</b>	Venda de comida <i>premium</i> online.
<b>B</b>	Aplicativo mobile para a compra de produtos em praça de alimentação.
<b>C</b>	Gestão de Limpeza Urbana, Conservação Pública, Meio Ambiente, Trânsito Urbano.
<b>D</b>	Educação, Mobilidade Urbana, Plataforma/ <i>Marketplace</i> de Saúde e Atendimento Restaurante.
<b>E</b>	Ambientes de treinamento/educação em realidade virtual.
<b>F</b>	<i>Marketplace</i> de eventos, gestão de pequenas empresas.

Os participantes afirmaram que sempre ou quase sempre o objetivo de oferecer produtos com características inexistentes em outros produtos tem influência nas tomadas de decisão dos projetos (Figura 7), que é uma das características desejadas para a população-alvo do estudo. A resposta “Quase Sempre” foi escolhida por quatro participantes: A, C, D, e F. No caso da organização A, o respondente explicou que a preocupação de oferecer o que a concorrência oferece também é importante para as tomadas de decisão: *“Muitas vezes se olha o concorrente para saber o que estão fazendo para poder também não ficar atrás. Apesar de não concordar muito com essa estratégia, isso acontece”*. Já a empresa C afirmou não ter concorrente para o produto que desenvolve, logo não tem preocupação em ter diferencial no produto como um todo, porém isso é levado em consideração em algumas funcionalidades e na forma de oferecer o serviço (modelo de negócio). No caso da empresa D, além de desenvolver produto diferente dos existentes muitas vezes é observada a possibilidade de reutilizar software que já desenvolveram em projetos anteriores para diminuir o tempo de desenvolvimento do software. A empresa F afirmou ter como meta o pioneirismo, porém muitas vezes as decisões do projeto são tomadas devido a pedidos de usuários ou parceiros, além de influências de *“percepção em mercados externos (visitas em feiras), mudança de mercado, variação de tendências, etc.”*. Considerou-se então que as respostas “Quase sempre” não excluíam a organização da população de organizações que tem objetivo de inovação.

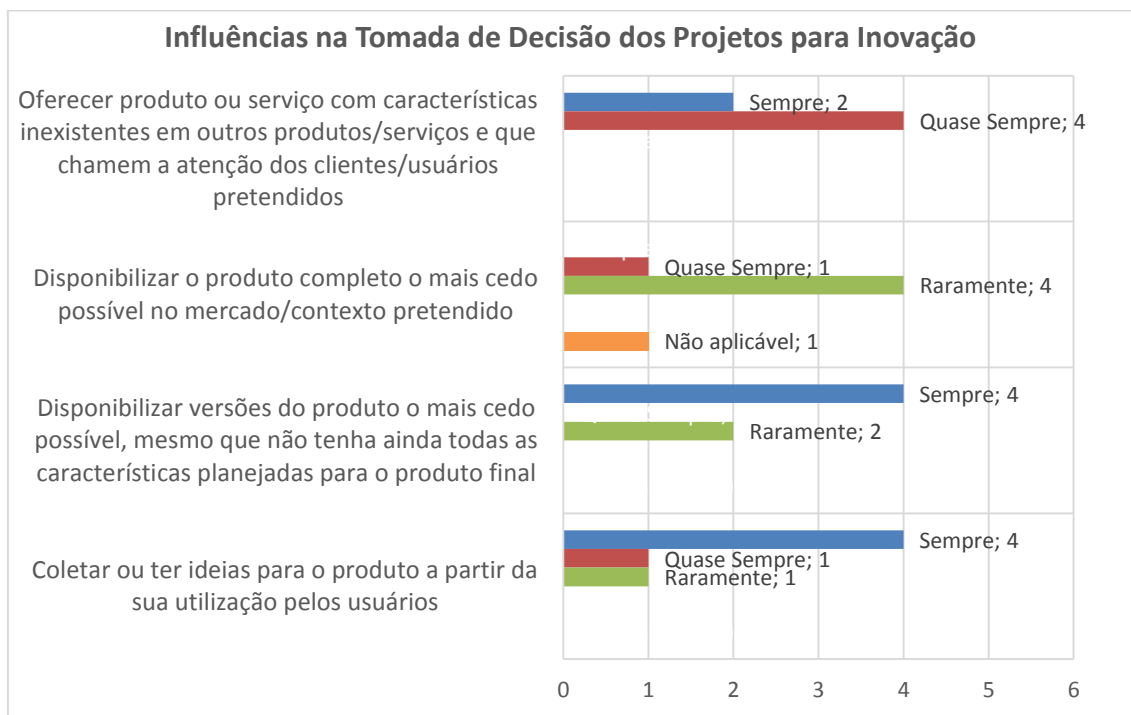


Figura 7. Estudo 1 – Influências de objetivos na Tomada de Decisão

Sobre o objetivo de disponibilizar o produto completo o mais cedo possível no mercado, apenas o participante da empresa A respondeu "Quase Sempre" (vide Figura 7). Em pesquisa por mais informações na empresa, e considerando as respostas sobre o projeto atual do participante, chegou-se à conclusão de que a empresa A tem um produto principal, para o qual são desenvolvidas extensões em projetos, que não são lançados no mercado de forma independente do produto principal. Da perspectiva do produto principal, o mesmo é disponibilizado em versões com evoluções e incrementos, por isso a organização foi mantida como população do estudo.

Os participantes das empresas A e C responderam que raramente consideram nas tomadas de decisão o objetivo de lançar versões do produto o mais cedo possível no mercado, enquanto as outras responderam "Sempre". No caso da organização C, a mesma está no aguardo de um evento específico do mercado pretendido (eleições municipais no Brasil) e declarou que só depois de alguns meses após o evento é que irá comercializar o produto, logo a pergunta não é adequada para ela.

Apenas o respondente da empresa A afirmou que raramente se tem o objetivo de geração de ideias a partir da utilização do produto pelos usuários. Como a resposta foi diferente de "Nunca", e pelo fato do respondente estar trabalhando no desenvolvimento de evolução do produto em projeto que está em estágio inicial (conforme respostas posteriores acerca do projeto), decidiu-se por manter a organização A no estudo.

Em relação à formação e experiência dos respondentes (Quadro 10 e Quadro 11), todos têm formação de graduação em curso de computação ou sistemas de informação, com exceção do participante da empresa E, que possui doutorado em Engenharia Nuclear e tem experiência acadêmica, profissional e como docente em Inteligência Artificial, Computação Gráfica e Realidade Virtual.

Quadro 10. Área de Formação dos Participantes

<b>Id.</b>	<b>Formação</b>	<b>Papéis que desempenha</b>
<b>A</b>	Ciência da Computação	Líder de time, desenvolvimento.
<b>B</b>	Engenharia da Computação	Codificação, testes manuais, avaliação de usabilidade, especificação de requisitos.
<b>C</b>	Ciência da Computação	Coordenação e Negociação de projeto de software
<b>D</b>	Ciência da Computação	Vendas, Gerente de Produto e Arquitetura
<b>E</b>	Engenharia Nuclear	Gerenciamento de projetos.
<b>F</b>	Sistemas de Informação	<i>Product Owner</i> , CEO. Análise de dados reais, validação do produto.



Quadro 11. Respostas sobre Experiência dos Participantes com Desenvolvimento de Software

	<b>A, C, D e F</b>	<b>B</b>	<b>E</b>
Análise e Especificação de Requisitos	Trabalhei em vários projetos na indústria	Trabalhei em vários projetos na indústria	Trabalhei em vários projetos na indústria
Projeto de Sistemas	Trabalhei em vários projetos na indústria	Não tenho experiência	Trabalhei em vários projetos na indústria
Codificação	Trabalhei em vários projetos na indústria	Trabalhei em vários projetos na indústria	Trabalhei em <u>1</u> projeto na indústria
Teste	Trabalhei em vários projetos na indústria	Trabalhei em <u>1</u> projeto na indústria	Trabalhei em <u>1</u> projeto na indústria
Gerenciamento de Projeto de Software	Trabalhei em vários projetos na indústria	Estudei ou li a respeito	Trabalhei em vários projetos na indústria

Os participantes das empresas B e D possuem titulação de mestrado e da empresa C tem doutorado, todos na linha de pesquisa de Engenharia de Software. O participante da empresa D é também docente de Engenharia de Software em curso de graduação. Por sua vez, o participante da empresa F possui dois cursos de pós-graduação na área de gestão e negócios e, assim como os participantes das empresas A e D, possui mais de dez anos de experiência em atividades de desenvolvimento de software. Com base na experiência, formação e papéis desempenhados em projeto de inovação, os participantes foram considerados com perfil adequado para o estudo, de acordo com seu objetivo (descrito no Quadro 6).

É importante observar que o nível de experiência e o perfil de formação dos participantes do estudo não podem ser considerados um padrão nas organizações de desenvolvimento de software para inovação, como por exemplo, as empresas *Startups* que participaram do estudo relatado em (GIARDINO, PATERNOSTER, *et al.*, 2015), as equipes da empresa Beta e os participantes do primeiro piloto executado. Se por um lado pode-se argumentar que o fato diminui a possibilidade de generalização dos resultados, por outro lado a presente pesquisa assume um posicionamento construtivista de pesquisa, reconhecendo que características das organizações, das equipes, do domínio de problema e do mercado/contexto pretendido, das tecnologias de apoio e do próprio software sendo desenvolvido para inovação, dentre outras características e fatores, influenciam nas práticas adotadas no desenvolvimento de software, como discutido no estudo de (KHURUM, FRICKER e GORSCHKEK, 2015).

#### 4.4.2 Características dos projetos de software para inovação

Dos projetos sendo caracterizados no estudo, apenas o da organização A tem baixa prioridade na organização (vide Quadro 12), porém se trata de um projeto para desenvolvimento de produto para expandir o produto principal da empresa, que está em uso no mercado de onze países, diferente dos projetos de inovação das outras organizações participantes. Para as empresas B, C e E o projeto é o único em desenvolvimento e, portanto, é um investimento cujo retorno é indispensável para a sobrevivência da organização. Já no caso de D e F, as empresas desenvolvem outros projetos de software sob demanda para manter a empresa, porém esperam obter crescimento e maior fatia de mercado a partir do produto desenvolvido para inovação.

Quadro 12. Importância do Projeto de Inovação para a Organização

O(s) projeto(s) de software para inovação que trabalho...	A	B	C	D	E	F
... faz/fazem parte das estratégias de negócio centrais da empresa....						
... é/são importante(s) mas não é/são parte das estratégias de negócio centrais da empresa.						
... tem baixa prioridade em relação aos outros projetos da empresa.						

Quanto ao estágio dos projetos (Quadro 13), três do conjunto de seis projetos estão em fase de prototipação da primeira versão (empresas A, B, e E) e ainda não avaliaram o software no mercado/contexto pretendido. No caso da empresa D, ainda que a primeira versão do produto esteja em desenvolvimento, foi estabelecida parceria com possível cliente para avaliação das propostas de funcionalidades. A estratégia de parcerias com possíveis clientes também está sendo usada pelas organizações C e F, porém já existe uma versão operacional do produto e a validação do mesmo ocorre em situações reais. Considerando o estágio de produto, tem-se dois grupos na amostra do estudo: Iniciando (empresas A, B, e E) e Estabilizando (empresas C, D e F).

Quadro 13. Estágios do Projeto de Software para Inovação

Que situações indicam o(s) estágio(s) dos projeto(s) de software para inovação que atualmente trabalha?	A	B	C	D	E	F
Refinando a ideia de negócio, formando a equipe e iniciando o projeto.						
Prototipando ou construindo a primeira versão do produto.						
Avaliando a aceitação do produto após as primeiras vendas/adesões das primeiras versões.						
Adaptando o produto para as necessidades dos clientes que já aderiram ao produto.						
Investindo ativamente no aumento de usuários.						
Adicionando produtos complementares, expandindo o produto ou diversificando o portfólio para entrar em novos mercados/contextos.						
Dando suporte aos usuários atuais e sem tempo ou recursos para crescer ou expandir de forma intensa.						
Devido ao aumento da equipe e/ou da complexidade do produto, estamos melhorando ou adotando práticas de gerenciamento e desenvolvimento.						
Outros:						

Com exceção do participante da empresa C, todas os respondentes indicaram ao menos uma razão para a preocupação com menor *time-to-market* no projeto de software para inovação, sendo a limitação de recursos a mais citada (vide Figura 8). O respondente da empresa F não marcou as opções disponíveis, mas indicou outras razões: a) o risco de mudança de mercado; b) a diminuição do custo de “*feedback*”, que está relacionada com o item sobre incerteza da aceitação do produto, mas na perspectiva do respondente a dificuldade está na quantidade de características para validar, o que aumenta o custo de validação do produto quanto à adequabilidade para o mercado; c) o objetivo de “ser pioneiro”, que poderia estar relacionado com o risco de ser ultrapassado pela concorrência, mas o participante não concordou com essa relação porque, como afirmou, não tem a percepção de concorrência na região que atua: “*não tenho concorrência, é mais pela questão de pioneirismo, ser o primeiro a fazer esse tipo de solução*”.

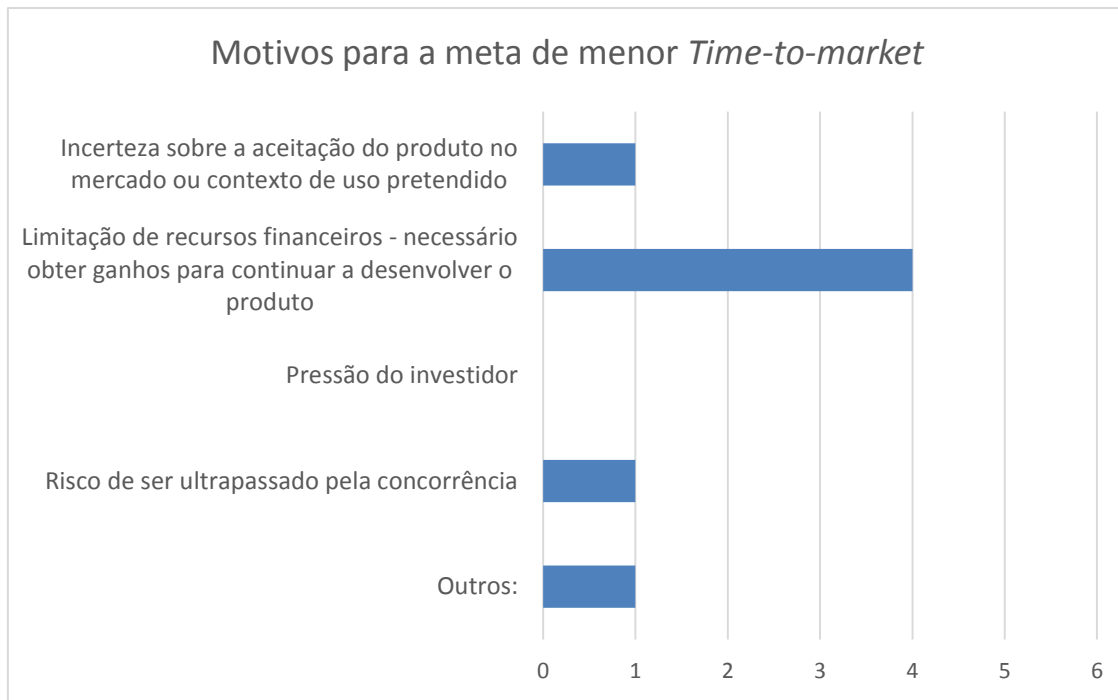


Figura 8. Motivos para a meta de menor *time-to-market*

Como mostra a Figura 9, as abordagens de desenvolvimento mais utilizadas pelos projetos são *Design Thinking*, *Lean Startup*, Processo Ágil (tendo suas práticas aplicadas em parte pelas organizações), além de Processo Iterativo e Incremental. Sobre o *Lean Startup* as organizações citaram a prática de desenvolver MVP's (*Minimum Viable Products*) e experimentar no mercado, usando o termo "validar".

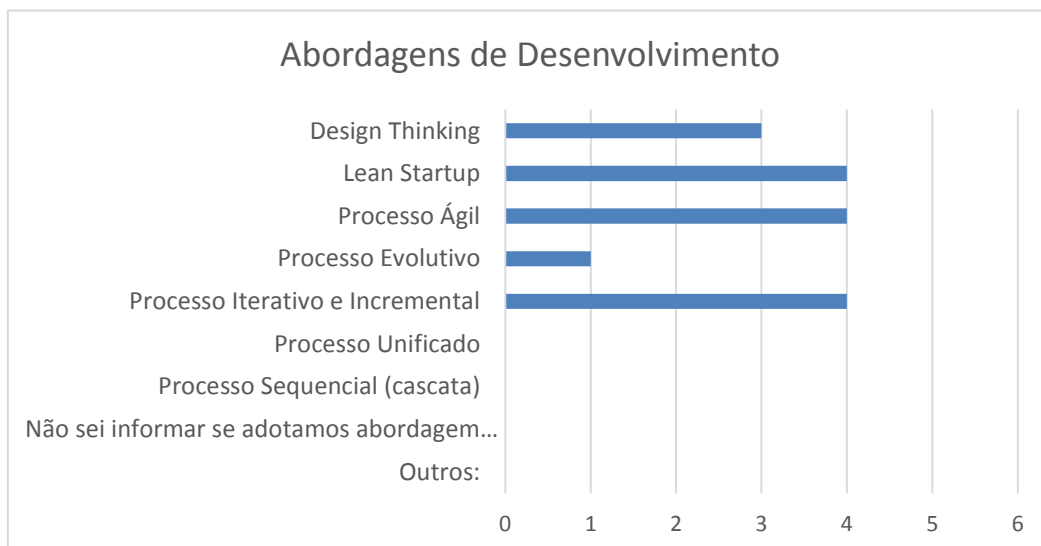


Figura 9. Abordagens de Desenvolvimento

#### 4.4.3 Práticas de registro de ideias e características nas organizações do estudo

Para caracterização das práticas de registro de ideias e características de software das organizações participantes, as informações coletadas com o apoio do questionário foram analisadas em conjunto com as informações obtidas nas entrevistas e pelas respostas dos participantes às questões enviadas durante o período de análise, elaboradas para confirmação de conclusões obtidas dos dados e também para esclarecer dúvidas restantes acerca das respostas.

As informações foram analisadas quanto aos fatores ou eventos que impactaram nas decisões sobre as práticas utilizadas nas organizações para o registro de ideias e características. Também foram analisadas as respostas dos participantes quanto suas perspectivas de necessidade de melhoria nas práticas adotadas nos projetos atuais. A seguir são apresentadas as informações de cada organização e na sequência são listados os fatores e eventos coletados, além dos eventos que podem estar relacionados aos riscos associados à documentação insuficiente com registro de características de software.

##### **Organização A:**

As ideias para as características do software desenvolvido pela organização A são sugeridas pelos próprios desenvolvedores com base em um sistema descrito pelo participante A como “complexo e legado”, para o qual está sendo proposto mudanças de características arquiteturais para futura integração com o sistema principal da empresa, adicionando assim novas funcionalidades ao produto principal. As ideias e as características do software são registradas em texto-livre, imagens com figuras e desenhos, além de protótipos de telas, havendo informalidade tanto na elaboração quanto no registro em relação ao quê e quando registrar. No caso das características do software, o registro é feito informalmente em ferramenta na forma de “*tickets*”<sup>3</sup>.

As práticas de registro, na visão do respondente, foram definidas pela “*natureza*” da empresa e de quem está à frente dela - “*Empresa que foge de processos fixos e foca na entrega mais ágil. Não há geração de nenhum tipo de artefato, a não ser os informalmente criados que são armazenados nos tickets criados de features*”. Por isso, mesmo percebendo a necessidade de melhoria nas práticas de registro das características do software no projeto atual em que trabalha, o respondente não acredita que alguma melhoria possa acontecer. Uma suposição para a organização A é que as

---

<sup>3</sup> Conceito usado em ferramentas de gestão de projetos para registro de pendências a serem resolvidas, podendo ser uma atividade, requisição de característica, reporte de falha, etc.

reuniões constantes da equipe ajudam a equipe a contornar as necessidades de melhoria.

#### **Organização B:**

O produto de software da organização B é o único produto da empresa, descrita pelo participante como empresa *Startup*. As fontes das ideias de características do software são outros produtos analisados, os fundadores da empresa e também pesquisa de mercado realizada. São utilizados protótipos de telas para registrar as ideias e especificação de requisitos para registrar as características de software. O respondente indica ter percebido no projeto a necessidade de melhoria de documentação, mas devido a limitações de recursos financeiros e a meta de desenvolver com menor *time-to-market*, dedicar mais esforço com documentação não é considerado apropriado - *“Não existe uma pressão por diminuir esforço com documentação, mas de fato não disponibilizamos de muito tempo para dedicar à essa tarefa. Por ser uma start-up, boa parte do nosso esforço está concentrado no desenvolvimento do produto para colocá-lo em produção o quanto antes”*.

#### **Organização C:**

A organização C obtém as ideias para as características do software a partir da análise de outros produtos e pesquisa de mercado, mas principalmente das ideias geradas pelos fundadores da empresa, por especialistas no domínio de problema e de pessoas da empresa que não são da equipe de desenvolvimento. São então utilizadas diferentes formas de registrar as ideias e as características, escolhidas informalmente pelas pessoas que explicitam a informação durante as atividades do projeto. Para ideias e características do software, são utilizados documentos com listagens, imagens com desenhos e figuras informais, mapas mentais, personas e protótipos de telas. Especificamente para registrar as ideias são utilizados texto livre e descrições hipotéticas de um dia na vida de um usuário do produto (*One day in life story*), enquanto que para características do software usam-se descrições de cenários, diagramas de projeto UML e protótipos do produto.

Na visão do gerente (respondente da empresa C), a proporção de esforço dedicado ao registro de informações e criação de documentos para uso da equipe não deve ser maior do que 10% do esforço total dispendido no projeto, tendo como justificativa o seguinte: a equipe está unida, as pessoas trabalham no mesmo local e existe baixa expectativa de saída das pessoas da equipe - *“equipe está engajada e confiante de que, o produto dando certo, eles continuarão na empresa. Como não há expectativa de turnover, não teria por quê colocar mais esforço em documentação”*; a equipe se reúne com frequência semanal (às vezes diária) para discussões sobre o projeto, além de realizarem *workshops* quinzenais para discussões sobre tecnologias,

como por exemplo, acerca do uso de determinada linguagem de programação; por conta da necessidade de buscar patrocínio, investiu-se esforço na produção de material de divulgação, com detalhamento de perfis de usuários, objetivos e funcionalidades oferecidas, incluindo exemplos de cenários de uso do produto, o que conta, na perspectiva do respondente, como documentação de apoio para a equipe.

#### **Organização D:**

A organização D desenvolve software sob demanda e também software com objetivo de inovação, tendo a expectativa de conseguir um dia obter sucesso no empreendimento e se dedicar apenas aos projetos de inovação. Para o software atualmente em desenvolvimento para inovação, após ter elaborado as ideias iniciais a organização D estabeleceu parceria com possível cliente visando descobrir requisitos e validar as ideias do software. Além disso, estabeleceu parceria com especialista do domínio de problema.

A equipe de desenvolvimento da organização D trabalha de forma distribuída (segundo o respondente, “totalmente *home office*”), sendo utilizadas ferramentas de apoio à colaboração para discussão acerca do projeto, das ideias e das características do produto, com organização de tópicos que remetem às etapas do ciclo de desenvolvimento e do ciclo de vida do produto, além da identificação do tema discutido. O respondente afirmou que além de apoiar o registro de informações, incluindo sobre tomadas de decisões, a organização das discussões na ferramenta lhe ajuda a observar o avanço das atividades do projeto, visto que ele pode visualizar as discussões ativas por etapa de desenvolvimento.

Sendo uma organização que está desenvolvendo seu quinto projeto de software com o objetivo de inovação, o respondente afirmou que não costuma dispendir esforço considerável no registro de informações no início do projeto devido as incertezas em relação ao que vai ser desenvolvido. Nesse estágio ele elabora o que chamou de “*briefing técnico*” e, por motivos de custo, contrata desenvolvedores inexperientes para desenvolver os protótipos e versões iniciais do produto, para que possam “*amadurecer as ideias*”. Quando sente “*confiança que o software pode dar certo*”, ele contrata pessoal técnico mais experiente, precisando antes disso criar documentação de requisitos e arquitetura do software para repassar aos novos contratados e assim integrá-los à equipe. A partir desse estágio onde a visão do software a ser desenvolvido está mais “*madura*” e a equipe aumenta, o respondente afirmou que precisa gerenciar o produto, o quanto dele está desenvolvido e o que está em desenvolvimento. Por causa disso, o registro das características do software é mais frequente a partir dessa etapa.

Periodicamente, é desenvolvido material de divulgação para apresentar à possíveis parceiros ou patrocinadores, vindo a compor a documentação do software. O

respondente citou ainda que em alguns casos em que tentou obter parceria com outras empresas, teve que elaborar documentação técnica do projeto, incluindo especificações de requisitos e projeto de arquitetura, passando então a usar essas documentações para “*mostrar profissionalismo*” aos potenciais parceiros e patrocinadores.

#### **Organização E:**

Na opinião do respondente da organização E, mesmo quando se trata de um projeto de inovação é importante que haja um direcionamento, no sentido de metas a alcançar. Ao perceber que as metas estão sendo alcançadas, ou quando surgem ideias com potencial de inovação, a equipe pode se desviar do direcionamento inicial para experimentar novas ideias, no sentido de desenvolvê-las para observar a viabilidade e potenciais ganhos a serem obtidos e demais impactos da inclusão delas no produto em desenvolvimento. Segundo o respondente, o papel do gerente é fundamental para controlar o quanto de esforço e tempo do projeto pode ser dedicado pela equipe à “*liberdade criativa*” visando inovação, além de manter ou redirecionar a visão de escopo do projeto.

Pelas razões citadas anteriormente, o respondente da organização E tem como prática a elaboração e a atualização de documentações (texto livre) sobre as ideias e características do software, incluindo um plano de projeto com visão e objetivos do produto, e documento com cronograma de atividades que são definidas de forma a permitir a identificação de qual funcionalidade ou característica será desenvolvida durante a atividade. O nível de detalhamento do plano de projeto descrito depende do tamanho do projeto, mensurado pelo respondente pelo tempo estimado de desenvolvimento (“*até 6 meses eu posso descrever a visão bem simples e então fazer o cronograma. O cronograma sempre tem na planilha Excel*”). Além disso, são elaboradas atas das reuniões de *Brainstorming* em que as ideias são discutidas por uma equipe de criação. Além de documentos escritos informalmente em texto livre, a organização costuma usar Mapas Mentais para explicitar as ideias das características do produto, inclusive características de software.

Uma característica importante do projeto, enfatizada pelo respondente como influente nas decisões de planejamento de projeto e práticas de documentação, é que a organização tem o perfil de empresa de pesquisa e desenvolvimento, significando que não quer inovar apenas atendendo a uma demanda em aberto no mercado, mas quer criar inovação na área de pesquisa que atua (Realidade Virtual), por isso vê como importante se ter objetivos e metas definidos. O respondente citou também que o software desenvolvido interage com dispositivos como, por exemplo, óculos especiais para aplicações de Realidade Aumentada, o que faz ser necessário definir e detalhar com antecedência as características do software a ser desenvolvido devido as possíveis



restrições da tecnologia utilizada. Outras restrições para características de software podem ser oriundas de aplicativos e demais software de apoio utilizados para aplicações de Realidade Virtual. Em suma, as tecnologias utilizadas no projeto podem aumentar o custo de modificação das características do software que com as quais interagem, o que contribui para a prática de registrar as características mesmo que em texto (não são usados modelos arquiteturais, por exemplo).

Em relação à melhoria de práticas para registro de ideias e características de software, o respondente citou que mesmo descrevendo as ideias em atas disponibilizadas aos participantes do projeto, muitas vezes o desenvolvedor entende a ideia a ser desenvolvida de forma diferente do que o gerente pensou quando a descreveu. Nessa situação o gerente volta a discutir com a equipe de criação e às vezes descobre que mesmo a equipe de criação tem expectativas diferentes. Normalmente, nesse caso é realizada outra reunião de *Brainstorming* para rediscutir as ideias e alinhar expectativas.

#### **Organização F:**

A organização F tem experiência de desenvolvimento em seis projetos de software com objetivo de inovação e desenvolve projetos de software sob demanda como forma de sustentar a empresa e seus projetos de inovação enquanto eles não geram retorno de investimento. No projeto de inovação atual, a organização estabeleceu parceria com uma pessoa que trabalha no domínio do problema (gestão de eventos), tendo oportunidade de acompanhar esta pessoa durante suas atividades de trabalho para observação e geração de ideias, descoberta de requisitos e também para validação de versões do produto. Durante o estágio inicial de desenvolvimento do produto ou de características novas em uma versão que já atingiu o mercado, a organização tem a meta de gerar uma versão o mais rápido possível, mas não apenas com o objetivo de disponibilizar no mercado, pois existe também o objetivo de validar o produto com o usuário parceiro do projeto o quanto antes.

O produto é disponibilizado para o mercado de forma incremental inspirado na abordagem *Lean Startup*, pela qual se tem meta de gerar um produto que seja minimamente viável (MVP – *Minimum Viable Product*) a cada iteração de desenvolvimento para permitir sua validação. Assim, antes da sua disponibilização no mercado o produto já passou por ciclos de validação com o cliente parceiro da empresa F, diminuindo dessa forma a incerteza da adequabilidade do produto e também aumentando a confiabilidade do produto. No caso de disponibilizarem versão de produto que já foi adotado por clientes, a equipe define quantos dias serão necessários para validação e durante esse tempo testam o produto nos ambientes de uma parte dos clientes.

O respondente da organização F afirmou que as práticas para registro de ideias e características costumavam ser as mesmas durante as diversas iterações do desenvolvimento do produto: há o registro de ideias em uma ferramenta de apoio que a equipe usa sempre que uma ideia para o software surge. O projeto é organizado em outra ferramenta usando o conceito de “ondas”, semelhante ao conceito de *sprints* da abordagem *Scrum*. No início de cada “onda”, a equipe revisa as ideias registradas e seleciona as que serão desenvolvidas na “onda”. Para ideias que julgam ter pouca incerteza, a equipe cria e detalha os requisitos. No caso de muitas incertezas, conversam com o usuário parceiro ou com os chamados “*early adopters*”, que são pessoas identificadas como possíveis primeiros adeptos do produto. As ideias e características do software são registradas em diferentes formas, conforme o que as pessoas da equipe informalmente julgam como melhor durante a execução das atividades, e são então inseridas nas ferramentas de apoio no formato de documentos com listagens, imagens com figuras e desenhos informais, mapas mentais, descrição de *personas*, descrição de um dia na vida do usuário, protótipos de telas e protótipos funcionais. Também são utilizados quadros de trabalho compartilhados onde as pessoas colam *Post-its*, com detalhes do que está sendo desenvolvido. Baseada na experiência com projetos passados, o respondente afirmou que quando o produto “escala” (alcança uma determinada quantidade de clientes), o custo de validar as novas versões do produto com a estratégia de observar a sua utilização no ambiente de uma amostra dos clientes se torna alto e aumenta o tempo de entrega para os outros clientes. Além disso, a preocupação com confiabilidade do software é maior nesse estágio. Por esses motivos, a empresa está implantando a abordagem de desenvolvimento baseado em teste (TDD – *Test-Driven Development*) (BECK, 2013), que deverá ser utilizada apenas após o produto “escalar”. O gerente espera que, com a especificação de testes para as funcionalidades do software “escalado” sendo utilizada para apoiar o desenvolvimento, o tempo de validação seja reduzido e a confiabilidade do produto seja maior, a ponto de ser válido investir o esforço com planejamento de testes nesse estágio. Para os produtos em estágio inicial, a meta principal continua sendo a de disponibilizar o quanto antes no mercado.

Considerando as informações de todas as organizações participantes do estudo, os fatores listados a seguir foram identificados como influentes na determinação das práticas utilizadas nos seus projetos de desenvolvimento de software para inovação (Quadro 14):

Quadro 14. Fatores que influenciam a escolha das práticas de registro de ideias e características de Software

Elemento	Fator de Influência
<b>Gerente</b>	<ul style="list-style-type: none"> <li>• Opinião sobre práticas e modelos de desenvolvimento de software.</li> <li>• Estratégia que adota para gerenciamento de tarefas e de produto.</li> </ul>
<b>Produto</b>	<ul style="list-style-type: none"> <li>• Incertezas sobre as características do software.</li> <li>• Custo de mudança do software devido a integração com outros dispositivos, hardware e/ou software.</li> </ul>
<b>Equipe</b>	<ul style="list-style-type: none"> <li>• Estabilidade da Equipe.</li> <li>• Frequência e efetividade de comunicação entre as pessoas equipe.</li> <li>• União da equipe.</li> </ul>
<b>Projeto</b>	<ul style="list-style-type: none"> <li>• Limitações de recursos.</li> <li>• Meta de menor <i>Time-to-market</i>.</li> </ul>

Quatro das seis organizações participantes do estudo indicaram utilizar práticas informais em relação ao formato de explicitação para o registro de informações, e, comparando com outros projetos que já participaram, o esforço dedicado para elaboração de modelos ou documentos é menor devido a meta de desenvolver com menor *time-to-market*, a limitações de recursos, a incertezas sobre as características do produto e também devido ao direcionamento dado por quem está à frente do projeto e/ou da organização.

A organização F não considera que dispense menos esforço com registro de informações em comparação aos projetos sob demanda que desenvolve, declarando manter uma “*postura constante*”, com descrição de requisitos e registro de informações em ferramenta de colaboração. Também foi a única organização que indicou todos os itens que são mostrados na Figura 10 como motivos de melhoria da qualidade/ou quantidade da documentação das características e ideias do software no projeto atual de inovação.

Ao comparar o que as organizações usam para registrar ideias e características do software a partir das respostas do questionário (Quadro 15), percebe-se que as organizações com produto em estágio de prototipação (A, B, e E) indicaram poucas formas de registro em comparação às outras organizações (C, D, e F). Na organização C e F as formas de registro são utilizadas conforme a pessoa ou equipe explicitando a informação julga serem adequadas.

Quadro 15. Formas de Registro de Ideias e Características

Forma de Registro de Ideias e Características	A	B	C	D	E	F
Texto livre informal	IC		I	I	CA	IC
Descrição de Casos de Uso, Estórias de Usuário, <i>Storyboards</i>			CA	CA		CA
Diagramas de Projeto da UML (ex.: classes, estados, sequência)			CA	CA		CA
Documento com listagem de ideias ou características			IC	I		IC
Especificação de Requisitos		CA		CA		CA
Imagens com figuras e desenhos informais	IC		IC	I		IC
Mapas Mentais			IC		I	IC
Descrição de Personas			IC	I		IC
Descrição de um dia na vida do usuário ( <i>One Day in Life story</i> )			I			IC
Protótipos de Telas, <i>mockups</i>	IC	I	IC	IC		IC
Protótipos funcionais, mesmo que incompletos			CA			IC
Outros						IC
<b>Legenda - IC: Ideias e Características, I: Ideias, CA: Características</b>						

## 4.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Apesar de nenhum dos participantes do estudo estarem trabalhando em projetos cujos produtos consideram ter atingido uma determinada parcela de mercado (uns estão em fase inicial de prototipação, outros em fase de estabilização das características do produto, tendo parceria com um cliente), foi possível identificar eventos cujas consequências fizeram as organizações mudarem suas práticas iniciais devido ao impacto na produtividade de desenvolvimento, resultando em aumento de esforço com atividades que envolvem registro com mais detalhes de informações sobre características de software (vide Quadro 16).

Quadro 16. Eventos e consequências que resultaram em aumento de esforço dedicado ao registro de características de software

Evento	Consequência
Evolução da visão do produto em termos de objetivos e aumento de confiança na chance de sucesso.	<ul style="list-style-type: none"> <li>Contratação de pessoal qualificado (evento relacionado: Mudança de participantes da equipe).</li> </ul>
Mudança de participantes da equipe (inclusão e saída de pessoas).	<ul style="list-style-type: none"> <li>Necessidade de repassar e registrar informações.</li> </ul>

Evento	Consequência
O número de clientes utilizando o produto atinge quantidade suficiente para a organização considerar que houve aumento de escala.	<ul style="list-style-type: none"> <li>• Aumento do custo e tempo de validação das versões.</li> <li>• Necessidade de aumentar a confiabilidade do produto.</li> </ul>

Quanto ao registro de ideias, as organizações afirmaram não terem percebido perda de oportunidade de inovação por falta de registro de ideias e indicaram que as ferramentas de apoio e de colaboração, assim como o fato das pessoas da equipe se comunicarem frequentemente, ajudam a não perderem ideias. A organização F informou ainda que, periodicamente, no início das iterações de desenvolvimento de novas versões do software revê com a equipe as ideias registradas.

Em relação à percepção da necessidade de melhorar a qualidade e/ou quantidade da documentação relacionada às ideias e características do software para inovação que atualmente trabalham, os participantes selecionaram as motivações mostradas na Figura 10. As respostas mais frequentes são referentes ao alinhamento de entendimento da visão do produto e das ideias e características (quatro de seis selecionaram os itens), o que pode estar relacionado ao fato dos produtos em desenvolvimento estarem em fase inicial de desenvolvimento ou não terem ainda um número significativo de usuários. Os próximos itens mais frequentes (selecionados por três dentre seis) são referentes às tarefas de desenvolvimento das funcionalidades e testes. É possível observar que alguns dos motivos selecionados considerando o estágio atual do projeto podem estar relacionados com as consequências de eventos que provocam mudanças das práticas de registro: Necessidade de explicar as ideias e características do software para outra pessoa da equipe e Necessidade de repassar e registrar informações; Execução de testes no software e Necessidade de aumentar a confiabilidade do produto.

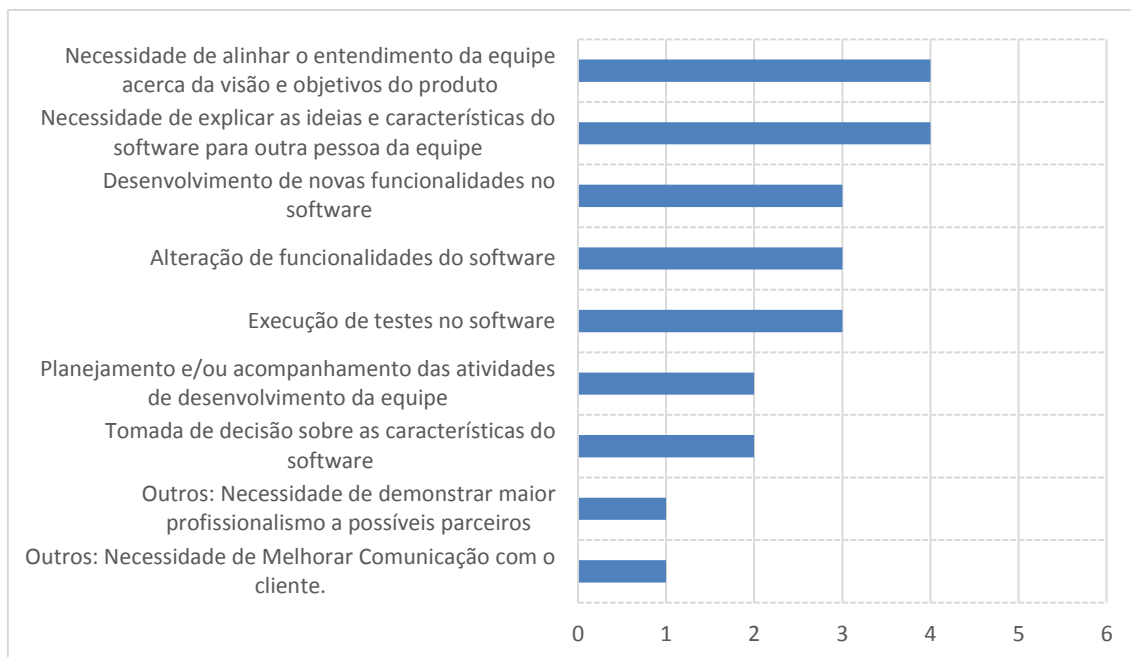


Figura 10. Motivos para Melhoria da Qualidade/Quantidade de Documentação

Não foi possível avaliar a pertinência da suposição 1 do estudo, as organizações participantes não disponibilizaram ainda seus produtos no mercado, tendo no máximo um cliente parceiro do projeto. Ainda assim, elas indicaram que a preocupação com o desenvolvimento com menor *time-to-market* é um fator considerado na decisão sobre as práticas que utilizam para documentação.

Foi possível observar também outros fatores que os gerentes consideraram como importantes para definir as práticas de documentação (Quadro 14), carecendo de mais investigações para analisar a pertinência, relevância e influência desses fatores na adoção de práticas e tecnologias de registro de informações.

O estudo relatado neste capítulo teve participação de seis profissionais de Startups distintas, sendo necessário obter informações de mais Startups para fortalecimento da evidência científica. A aplicação de entrevistas semiestruturadas, método de coleta de dados, possibilitou a identificação de um conjunto de fatores de influência, porém, para fortalecer a evidência científica, seria necessário realizar mais entrevistas envolvendo Startups com diferentes características, como por exemplo, região, porte, dentre outras. Porém, conseguir mais participantes para o estudo descrito nesse capítulo não se mostrou uma tarefa trivial. Por isso, decidiu-se pela execução de um mapeamento sistemático da literatura para, diferente do mapeamento descrito no capítulo 2, analisar relatos de práticas e tecnologias adotadas pela indústria Startup. O capítulo a seguir detalha este mapeamento sistemático.

# 5 MAPEAMENTO SISTEMÁTICO: PRÁTICAS E TECNOLOGIAS PARA COMPARTILHAMENTO E REGISTRO DE CONHECIMENTO DE SOFTWARE EM STARTUPS

Este capítulo descreve a execução e resultados de um mapeamento sistemático sobre práticas e tecnologias de compartilhamento e registro de conhecimento de ideias e características de software adotadas em organizações Startup de desenvolvimento de software.

## 5.1 INTRODUÇÃO

Startups geralmente desenvolvem software de forma iterativa e evolutiva para coletar informações o mais cedo possível sobre a aceitação e adequabilidade das ideias desenvolvidas, além de buscarem não serem ultrapassadas por empresas concorrentes na busca por inserir inovação no mercado. Para adquirir mais produtividade, as equipes em Startups se apoiam em conhecimento tácito ao invés de modelos e documentações pois, além de demandarem esforço de desenvolvimento, eles podem vir a ser descartados devido a possíveis mudanças a partir da análise dos resultados obtidos pela exposição do software aos possíveis clientes.

Diferente do conhecimento explícito, que segundo Takeuchi e Nonaka (TAKEUCHI e NONAKA, 2008) pode ser transferido a indivíduos de maneira formal e sistemática, o conhecimento tácito é pessoal, adquirido por um indivíduo a partir de suas experiências. Do ponto de vista cognitivo, engloba os modelos mentais da pessoa, suas percepções, crenças e valores. Na perspectiva técnica, é formado pelas habilidades informais, ou o chamado *know-how* de uma pessoa. Apesar de benéfico nos estágios iniciais do ciclo de vida de um produto de software desenvolvido por Startups, o acúmulo de conhecimento tácito prejudica a produtividade desejada de desenvolvimento em estágios posteriores (GIARDINO, PATERNOSTER, *et al.*, 2015).

No estudo descrito no capítulo anterior, foram identificados elementos que influenciaram a seleção de tecnologias e práticas para o registro de conhecimento de software e apoiar a equipe de desenvolvimento em Startups. Observou-se também que as organizações adotavam práticas para favorecer o compartilhamento de conhecimento na equipe de maneira informal e com base em comunicação frequente. Com isso, elas esperavam diminuir a necessidade de criação de modelos e documentos,

o máximo e quando possível. O estudo envolveu contato direto com profissionais de Startups, porém seus resultados são limitados visto que se obteve a participação de apenas seis profissionais.

Como forma de aumentar a quantidade de fontes de evidências científicas, decidiu-se pela coleta de dados em relatos de práticas de desenvolvimento de software em Startups, sendo executado um mapeamento sistemático da literatura (estudo secundário) sobre os fatores que contribuem para as mudanças de práticas em registro de informação de software em Startups de desenvolvimento de software. O objetivo geral do estudo é detalhado no Quadro 17:

Quadro 17. Objetivo do Mapeamento Sistemático

Analisar	Projetos de desenvolvimento de software
Com o propósito de	caracterizar
Em relação a	fatores que influenciam a adoção de práticas e tecnologias de compartilhamento e registro de ideias e características de software, para apoiar equipes de desenvolvimento de software
Do ponto de vista	Pesquisadores de Engenharia de Software
No contexto de	Startups de software e descrito na literatura técnica.

As próximas seções deste capítulo apresentam o protocolo do estudo e os resultados, além de discussão acerca do método e dos resultados na última seção.

## 5.2 PROTOCOLO DO ESTUDO SECUNDÁRIO

O protocolo do estudo secundário, descrito nessa seção, é baseado em (BIOLCHINI, MIAN, *et al.*, 2005). Ele apresenta a pergunta de pesquisa, a estratégia de busca definida (termos de pesquisa, *string* de busca e as bases de dados nos quais se buscou os estudos), e os procedimentos de seleção e análise dos artigos aderentes ao contexto definido no objetivo do estudo.

### 5.2.1 Pergunta de Pesquisa

Em acordo com o objetivo geral do estudo, as seguintes perguntas de pesquisa foram definidas para serem investigadas no contexto de Startups de desenvolvimento de software:



- **RQ1:** O que influencia a tomada de decisão sobre a adoção de práticas e tecnologias para o compartilhamento e registro de conhecimento de ideias e características de software em Startups de desenvolvimento de software?
- **RQ2:** Quais são as tecnologias e práticas adotadas para o compartilhamento e registro de conhecimento sobre ideias e características de software em Startups de desenvolvimento de software?

### 5.2.2 Estratégia de Busca

A estratégia de busca foi baseada no trabalho de (PATERNOSTER, GIARDINO, *et al.*, 2014), que pesquisou por práticas de desenvolvimento de software em Startups, visto que a população a ser estudada era a mesma. Apesar da pré-existência de um estudo secundário, os artigos analisados pelo mapeamento sistemático de (PATERNOSTER, GIARDINO, *et al.*, 2014) foram extraídos das bases de dados selecionadas em dezembro de 2013, o que foi considerado temporalmente inconsistente para a presente pesquisa dada a identificação, via pesquisa *ad-hoc* na literatura, de novos estudos sobre práticas de desenvolvimento de software em Startups.

O Quadro 18 mostra os conceitos e termos utilizados como palavras-chave para a estruturação de uma *string* de busca em bases de dados científicas. Seguindo a proposta de (PAI, MCCULLOCH, *et al.*, 2003), os termos estão separados em categorias representando a População de interesse do estudo, a Intervenção a ser observada e a Comparação que será realizada. Após um período de avaliação dos artigos retornados, as palavras-chave da População foram adaptadas logicamente no presente estudo de forma a aumentar a sua abrangência, garantindo o retorno dos estudos anteriormente aceitos em (PATERNOSTER, GIARDINO, *et al.*, 2014). No caso das palavras-chave de Comparação, houve a inclusão do termo *non-functional requirement*.

Quadro 18. Termos da String de Busca

Conceitos Fundamentais	Termos Originais	Termos Utilizados
Software Startups	software startup*; software start-up*; early-stage firm*; early-stage compan*; high-tech venture*;high-tech start-up*; start-up compan*; startup compan*; lean startup*; lean start-up*; software pack-age start-up*; software package startup*; IT start-up*; IT startup*; software product startup*; software start up*; internet start-up*; internet startup*;web startup*; web start-up*; mobile startup*; mobile start-up*;	early-stage firm*; early-stage compan*; high-tech venture*; start-up*; startup* AND software; game; app

Conceitos Fundamentais	Termos Originais	Termos Utilizados
Development	develop*; engineer*; model*; construct*; implement*; cod*; creat*; build*	Os mesmos termos originais
Strategy	product*; service*; process*; methodolog*; tool*; method*; practice*; artifact*; artefact*; qualit*; ilit*; strateg*; software;	product*; service*; process*; methodolog*; tool*; method*; practice*; artifact*; artefact*; qualit*; ilit*; non-functional requirement*; strateg*

Seguindo a estratégia de busca de (PATERNOSTER, GIARDINO, *et al.*, 2014), as bases de dados selecionadas para a busca foram a *Scopus*, a *Inspec/Ei Compendex*, a *Web of Science* e a *ACM Digital Lybrary*. A base *Google Scholar* não foi incluída devido a sua instabilidade de retorno de artigos.

O Quadro 19 apresenta a *string* de busca adaptada para cada base de dados. Devido a limitações de quantidade de termos na base IEEExplorer, as palavras-chave de Comparação foram removidas da *string* executada nesta base, porém foram analisadas quando da leitura de títulos e resumo dos trabalhos. Como o mapeamento sistemático de (PATERNOSTER, GIARDINO, *et al.*, 2014) já havia aplicado os critérios de seleção e exclusão nos artigos retornados das bases em dezembro de 2013, no presente estudo considerou-se apenas artigos publicados a partir do ano 2010, deixando assim um período de dois anos de sobreposição entre os mapeamentos. Dessa forma, espera-se diminuir o risco de o conjunto de artigos dos dois mapeamentos não conter estudos que foram indexados após o ano de 2013, mas foram publicados antes. Exceto pela base *Scopus*, o período da busca foi configurado usando filtros disponibilizados pelas máquinas de busca.

Quadro 19. String de Busca

Fonte	String de busca
Scopus	TITLE-ABS-KEY ( ( ( "early-stage firm" OR "early-stage firms" OR "early-stage company" OR "early-stage companies" OR "high-tech venture" OR "high-tech ventures" OR "start-up" OR "start-ups" OR startup ) AND ( software OR game OR app ) ) AND ( develop* OR engineer* OR model* OR construct* OR implement* OR cod* OR creat* OR build* ) AND ( product* OR service* OR process* OR artifact* OR artefact* OR qualit* OR "non-functional requirement" OR "non-functional requirements" OR ilit* OR methodolog* OR tool* OR method* OR practice* OR strateg* ) ) AND ( PUBYEAR > 2009 )
Inspec/Ei Compendex	( ( ( "early-stage firm" OR "early-stage firms" OR "early-stage company" OR "early-stage companies" OR "high-tech venture" OR "high-tech ventures" OR "start-up" OR "start-ups" OR startup ) AND ( software OR game OR app ) )

Fonte	String de busca
(Engineering Village)	AND ( develop* OR engineer* OR model* OR construct* OR implement* OR cod* OR creat* OR build* ) AND ( software OR product* OR service* OR process* OR artifact* OR artefact* OR qualit* OR "non-functional requirement" OR "non-functional requirements" OR ilit* OR methodolog* OR tool* OR method* OR practice* OR strateg* ) )
Web of Science	TS= ( ( ( "early-stage firm" OR "early-stage firms" OR "early-stage company" OR "early-stage companies" OR "high-tech venture" OR "high-tech ventures" OR "start-up" OR "start-ups" OR startup ) AND ( software OR game OR app ) ) AND ( develop* OR engineer* OR model* OR construct* OR implement* OR cod* OR creat* OR build* ) AND ( software OR product* OR service* OR process* OR artifact* OR artefact* OR qualit* OR "non-functional requirement" OR "non-functional requirements" OR ilit* OR methodolog* OR tool* OR method* OR practice* OR strateg* ) ) )
IEEEExplorer	( ( "early-stage firm" OR "early-stage firms" OR "early-stage company" OR "early-stage companies" OR "high-tech venture" OR "high-tech ventures" OR "start-up" OR "start-ups" OR startup ) AND ( software OR game OR "app" OR "apps" ) )
ACM	recordAbstract:(("early-stage firm" "early-stage firms" "early-stage company" "early-stage companies" "high-tech venture" "high-tech ventures" "start-up" "start-ups" startup*) AND ( software game app )) AND (develop* engineer* model* construct* implement* cod* creat* build*) AND (software product* service* process* artifact* artefact* qualit* "non-functional requirement" "non-functional requirements" ilit* methodolog* tool* method* practice* strateg*)

### 5.2.3 Busca e Seleção

A *string* final de busca foi executada nas bases de dados selecionadas em junho/2018, retornando um total de 3.133 artigos, como pode ser visto no Quadro 20, que também mostra a quantidade de artigos selecionados por fonte de dados após cada etapa de seleção de estudos.

Quadro 20. Quantidade de Artigos por Etapa de Busca e Seleção

Fonte	Busca	Remoção de Duplicatas e Leitura de Título e Resumo	Leitura completa	Aplicação do Novo Critério
Scopus	1025	107	39	11
Inspec/Ei Compendex (Engineering Village)	880	16	0	0
Web of Science	625	17	2	0
IEEEExplorer	521	47	1	1
ACM	82	4	1	1
Artigos aceitos do Mapeamento de (PATERNOSTER, GIARDINO, <i>et al.</i> , 2014) (total: 43)	-	-	-	1
<b>Total</b>	<b>3.133</b>	<b>191</b>	<b>43</b>	<b>15</b>

Após a busca nas bases de dados, seguiu-se a remoção de duplicatas e a leitura de título e resumo dos artigos retornados. Nessa etapa foram aplicados os critérios de inclusão e exclusão definidos no mapeamento sistemático de (PATERNOSTER, GIARDINO, *et al.*, 2014) (vide o Quadro 21, que mostra os critérios escritos em Inglês, conforme o estudo original, para evitar perda ou mudança de significado por motivo de tradução). É importante notar que os critérios foram elaborados visando garantir que os artigos selecionados apresentem resultados baseados em práticas de contextos reais em Startups desenvolvedoras de software, o que exclui, por exemplo, resultados de parceria indústria-academia pela qual uma prática ou tecnologia foi de alguma forma avaliada em uma Startup, mas não adotada como prática de desenvolvimento na organização.

Quadro 21. Critérios de Inclusão e Exclusão. Adaptado de (PATERNOSTER, GIARDINO, *et al.*, 2014)

<b>Critério de Inclusão:</b>	The study presents a contribution to the body of knowledge on software development in startups. A contribution can be in the form of an experience report, applied engineering practices, development models or lessons learned.
<b>Critérios de Exclusão Originais:</b>	<ul style="list-style-type: none"> <li>• Not peer-reviewed (grey literature, books, presentations, blog posts, etc.).</li> <li>• Not written in English.</li> <li>• Clearly obsolete (more than 20 years old).</li> <li>• Full-text non-available.</li> <li>• Related to non-software companies (biotech, manufacturing, electronics, etc.).</li> <li>• Related to established companies (including VSE, SME, and research spin-offs).</li> <li>• Related to technicalities of startups (algorithms, programming languages, etc.</li> <li>• Academic Settings.</li> </ul>
<b>Novo Critério de Inclusão:</b>	The study presents practices or technologies to share or register knowledge about software ideas and characteristics and with the purpose of supporting the development team

O mapeamento sistemático reportado em (PATERNOSTER, GIARDINO, *et al.*, 2014) teve como perspectiva de observação as práticas utilizadas para apoiar as áreas de engenharia de software. O escopo da presente pesquisa é transversal a essas áreas, visto que tecnologias e práticas de compartilhamento e registro de conhecimento podem ser adotadas no âmbito de cada uma das áreas citadas como de engenharia do software, além de poderem existir práticas aplicáveis de forma comum a todas as áreas. Essa diferença de objetivo levou à adição de um critério de inclusão específico para a presente pesquisa sobre o relato apresentar práticas ou tecnologias de apoio ao compartilhamento e registro de conhecimento de ideias e características de software, como mostra o Quadro 21.

Na terceira etapa de seleção, 191 artigos foram integralmente lidos e avaliados quanto aos critérios de inclusão e de exclusão originais. Quarenta e três artigos foram selecionados e, juntamente com os 43 artigos incluídos no estudo de (PATERNOSTER, GIARDINO, et al., 2014), foram lidos e analisados quanto ao critério de inclusão adicionado sobre práticas e tecnologias de compartilhamento e registro de conhecimento acerca de ideias e características de software para apoiar a equipe de desenvolvimento. Por fim, 15 artigos foram selecionados e analisados em busca de resposta à pergunta de pesquisa (vide a listagem no Quadro 22).

Quadro 22. Artigos Selecionados

Id	Referência	Tipo de Pesquisa
[P1]	Almohri, H. M., & Almohri, S. A. (2017). Security evaluation by arrogance: saving time and money. <i>Proceedings of the 1st International Workshop on Software Engineering for Startups</i> (pp. 12-16). IEEE Press.	Experience Paper
[P2]	Coleman, G., & O'Connor, R. (2008). Investigating software process in practice: A grounded theory perspective. <i>Journal of Systems and Software</i> , 81(5), pp. 772-784.	Evaluation Research
[P3]	Duc, A. N., & Abrahamsson, P. (2016). Minimum viable product or multiple facet product? The Role of MVP in software startups. <i>International Conference on Agile Software Development</i> (pp. 118-130). Springer.	Evaluation Research
[P4]	Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., & Abrahamsson, P. (2016). Software development in startup companies: the greenfield startup model. <i>IEEE Transactions on Software Engineering</i> , 42(6), pp. 585-604.	Philosophical Paper
[P5]	Hokkanen, L., & Väänänen-Vainio-Mattila, K. (2015). UX work in startups: current practices and future needs. <i>International Conference on Agile Software Development</i> (pp. 81-92). Springer.	Evaluation Research
[P6]	Holm, J., & Laurila, K. (2015). Towards Action Track 3.0: The Role of Usefulness, Usability, and User Experience in a Startup Company Developing Location-Based Applications. <i>Information Visualisation (iV), 2015 19th International Conference on</i> (pp. 245-254). IEEE.	Experience Paper
[P7]	Kasurinen, J., Strandén, J. P., & Smolander, K. (2013). What do game developers expect from development and design tools? <i>Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering</i> (pp. 36-41). ACM.	Evaluation Research
[P8]	Ko, A. J. (2017). A three-year participant observation of software startup software evolution. <i>Proceedings of the 39th International Conference on Software Engineering: Software Engineering in Practice Track</i> (pp. 3-12). IEEE Press.	Experience Paper
[P9]	May, B. (2012). Applying Lean Startup: An Experience Report. <i>Agile Conference</i> .	Experience Paper
[P10]	Nascimento, L. M., & Travassos, G. H. (2017). Software Knowledge Registration Practices at Software Innovation Startups: Results of an Exploratory Study. <i>Proceedings of the 31st Brazilian Symposium on Software Engineering</i> (pp. 234-243). ACM.	Evaluation Research
[P11]	Nguyen-Duc, A., Wang, X., & Abrahamsson, P. (2017). What influences the speed of prototyping? An empirical investigation of twenty software startups. <i>International Conference on Agile Software Development</i> (pp. 20-36). Springer.	Philosophical Paper
[P12]	Pompermaier, L., Chanin, R., Sales, A., & Prikladnicki, R. (2017). An empirical study on software engineering and software startups: Findings from cases in an innovation ecosystem. <i>The 29th International Conference on Software Engineering and Knowledge Engineering</i> , (pp. 48-51). Pittsburgh, PA, USA.	Evaluation Research
[P13]	Rafiq, U., Bajwa, S. S., Wang, X., & Lunesu, I. (2017). Requirements Elicitation Techniques Applied in Software Startups. <i>Software Engineering and Advanced Applications (SEAA), 2017 43rd Euromicro Conference on</i> , (pp. 141-144).	Evaluation Research
[P14]	Souza, R., Malta, K., & De Almeida, E. S. (2017). Software engineering in startups: a single embedded case study. <i>Software Engineering for Startups (SoftStart), 2017 IEEE/ACM 1st International Workshop on</i> (pp. 17-23). IEEE.	Philosophical Paper
[P15]	Taipale, M. (2010). Taipale, M. (2010). Huitale—a story of a Finnish lean startup. <i>Lean Enterprise Software and Systems</i> (pp. 111-114). Springer, Berlin, Heidelberg.	Experience Paper

## 5.2.4 Procedimento de Análise

Como pode ser visto na Figura 11, desde a terceira etapa de seleção dos artigos, na qual eles passaram a ser lidos integralmente, técnicas de análise qualitativa (SALDAÑA, 2015) foram utilizadas para identificar nos artigos aceitos as informações necessárias para responder às perguntas de pesquisa do estudo. A ferramenta MaxQDA<sup>4</sup> foi utilizada para apoiar a análise qualitativa.

Apesar de possuir alguma semelhança com os procedimentos de análise do mapeamento descrito no capítulo 3 desta tese, o objetivo definido e as perguntas de pesquisa permitiram direcionar e planejar os procedimentos em etapas onde foram executados ciclos de imersão-codificação-refinamento de códigos, típicos de análise qualitativa, na qual diferentes tipos de codificação podem ser utilizados. As próximas subseções detalham as etapas de análise, que são: i) mapeamento de tecnologias e práticas de desenvolvimento de software, ii) separação e refinamento de práticas e tecnologias para registro e compartilhamento de ideias e características de software, iii) mapeamento de fatores de influência na adoção de práticas e tecnologias para registro e compartilhamento de ideias e características de software, e iv) mapeamento da relação entre as práticas e tecnologias identificadas com os fatores de influência.

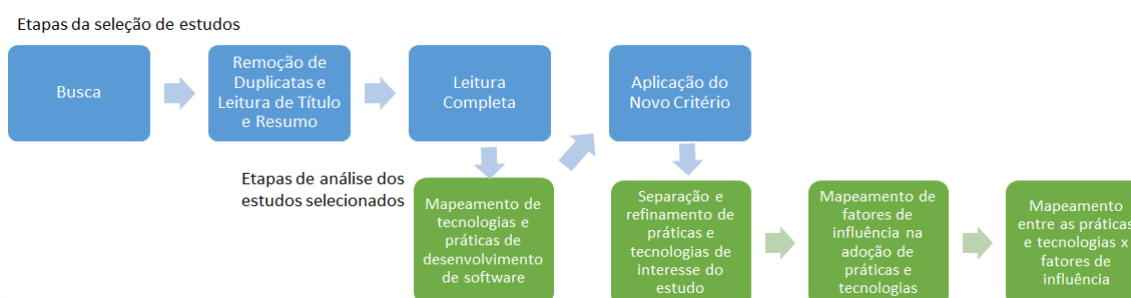


Figura 11. Etapas de Seleção e Análise

## 5.3 ANÁLISE DOS ARTIGOS SELECIONADOS

### 5.3.1 Mapeamento de tecnologias e práticas de apoio ao desenvolvimento de software

Nesta etapa, os artigos que satisfizeram os critérios de inclusão e exclusão foram também codificados. Usou-se um método de codificação exploratória (SALDAÑA, 2015), pela qual códigos são pré-definidos e atribuídos, quando pertinente, a trechos do

<sup>4</sup> <https://www.maxqda.com/>

texto sob investigação. Posteriormente esses códigos podem ser refinados, sendo até possível o uso de outros métodos de codificação.

O método escolhido de codificação exploratória foi o de codificação provisória (*provisionary coding*) (SALDAÑA, 2015), que permite ao pesquisador preparar uma lista inicial de categorias ou tipos de respostas que podem aparecer nos dados em análise. A lista inicial de códigos deve surgir de investigação preparatória do estudo, como por exemplo, de revisões da literatura relacionadas ao estudo, da estrutura conceitual e das perguntas de pesquisa, resultados de pesquisas anteriores, entre outros. No caso da presente pesquisa, na codificação provisória foram utilizadas as categorias de (PATERNOSTER, GIARDINO, *et al.*, 2014) para indicar facetas de foco para atividades de Engenharia de Software, que são mostradas no Quadro 23. A categoria de Desenvolvimento de Software é subdividida ainda nas seguintes subcategorias: a) Engenharia de Requisitos, b) Projeto e Arquitetura, c) Implementação, Manutenção e Implantação, e d) Garantia da Qualidade.

Quadro 23. Codificação Provisória

Facetas de Foco	Propósito da prática ou tecnologia pertencente à Faceta
Software Development	Apoiar a escrita e manutenção de código-fonte.
Process Management	Apoiar o gerenciamento das atividades de desenvolvimento
Tools and Technologies	Criar, depurar, manter e apoiar atividades de desenvolvimento
Managerial/Organizational	Apoiar aspectos do desenvolvimento de software, relacionados ao gerenciamento de recursos e estrutura organizacional.

Para o presente estudo, as facetas foram utilizadas para categorizar o foco de apoio de práticas e tecnologias reportadas nos artigos selecionados. Para o mapeamento das práticas e tecnologias, utilizou-se a codificação descritiva (*descriptive coding*) (SALDAÑA, 2015), que identifica os tópicos discutidos nos trechos analisados, resultando em um inventário dos tópicos. A cada prática ou tecnologia de desenvolvimento de software citada nos artigos, o tópico foi identificado e comparado com os códigos existentes, sendo criado um novo código de tópico caso nenhum pré-existente fosse adequado para o trecho. Após ciclos de leitura de dez artigos, os códigos foram revisados e refinados por um processo de agrupamento ou divisão de códigos. A Figura 12 mostra uma parte do resultado desta etapa, enfatizando como exemplo as práticas e tecnologias reportadas para a faceta de garantia da qualidade.



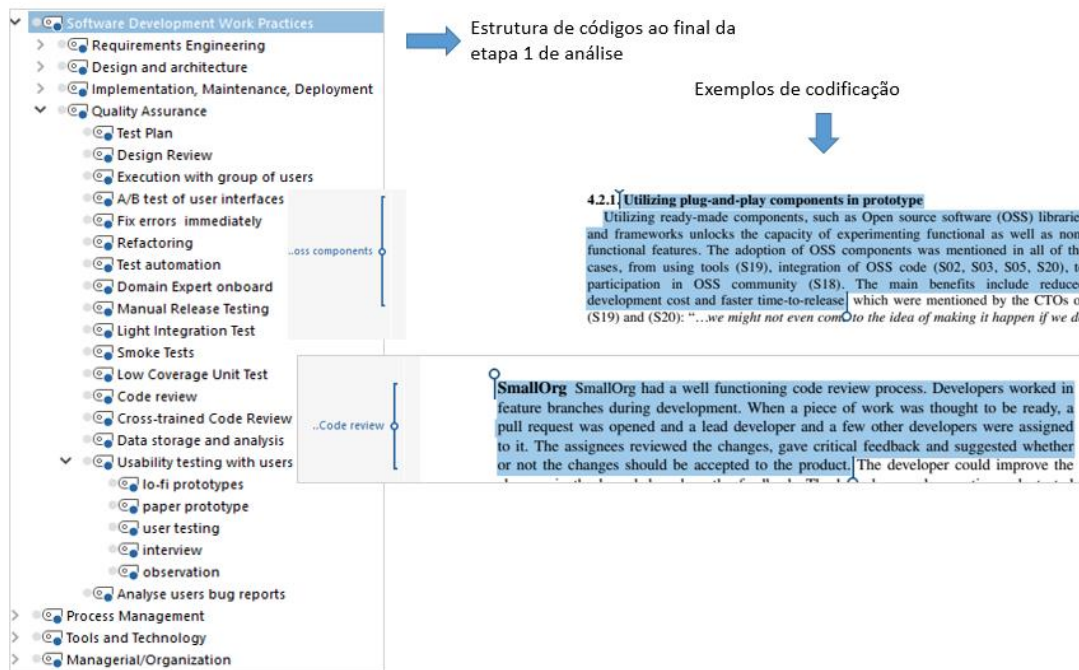


Figura 12. a) Estrutura de Códigos da primeira etapa de análise. b) Exemplo de trechos codificados.

### 5.3.2 Separação e refinamento de práticas e tecnologias para o registro e compartilhamento de ideias e características de software

Nesta etapa de análise, os códigos dos 15 artigos selecionados foram revisados para identificação de quais práticas e tecnologias de desenvolvimento adotadas contribuem para o compartilhamento e/ou registro de ideias e características de software.

Um tipo de prática ou tecnologia de interesse deste estudo é a que tenha como propósito principal o compartilhamento/ou registro de ideias e características de software, como por exemplo, a realização de reuniões frequentes, a criação de artefatos com informações para uso da equipe de desenvolvimento (ex.: estórias de usuário e protótipo de telas), o uso de quadros compartilhados no ambiente para exposição de informações em post-its, dentre outros.

Também se considerou práticas e tecnologias que, apesar de terem outro propósito principal, compartilham e/ou registram conhecimento sobre o software que são úteis para a equipe de desenvolvimento. Exemplo disso são listas de tarefas que descrevem as funcionalidades a serem desenvolvidas ou comportamentos esperados na execução de testes. Também é exemplo o uso de frameworks e padrões conhecidos de codificação, pois propiciam que a equipe forme um modelo mental da arquitetura do código-fonte.

A Figura 13 mostra a codificação final e o número de artigos que citam práticas e/ou tecnologias de cada código. Dois tipos de apoio principais foram identificados: 1)



práticas e tecnologias específicas para compartilhar e/ou registrar ideias e características de software, e 2) práticas e tecnologias de apoio à comunicação em geral, que são institucionalizadas na Startup e afetam a maneira de implantação e execução das práticas e tecnologias identificadas em 1).

▼	📁 Knowledge Sharing/Registration Practices and Technologies	0
▼	📁 Software Development	0
>	📁 Problem Domain Analysis	4
>	📁 Ideas and Characteristics Descriptions	3
>	📁 Design and Architecture	9
>	📁 Implementation and Maintenance	3
>	📁 Product Quality Assurance	2
>	📁 Delivery	1
▼	📁 Business	0
	📁 Marketing Documents	1
▼	📁 Management	0
>	📁 Product	2
>	📁 Tasks	4
>	📁 People Integration	1
>	📁 Tools	3
>	📁 General Sharing/Registering Technologies and Practices	9

Figura 13. Estrutura de Códigos de Práticas e Tecnologias

Com as práticas e tecnologias de interesse para o estudo devidamente mapeadas, os códigos provisionais utilizados como categorias (vide Quadro 23, na seção anterior) foram revisados quanto à adequação para categorizar os dados coletados. Na categoria *Software Development*, a subcategoria *Requirements Engineering* foi removida, sendo adicionada uma subcategoria que engloba o apoio para a equipe de desenvolvimento explorar o domínio do problema para o qual ideias de solução serão geradas e características de software desenvolvidas (*Problem Domain Analysis*), além de uma subcategoria que contém o que é utilizado de apoio para explicitar ideias e características de software. Não foram encontradas práticas e nem tecnologias relacionadas com a implantação do software. No caso de apoio à garantia da qualidade (código *Quality Assurance*), a subcategoria foi especializada porque os dados mapeados são relativos à qualidade do produto, passando o código a ser chamado de *Product Quality Assurance*. Ainda com base nos dados mapeados, foi incluído um código relacionado com a atividade de entrega de software (*Delivery*).

A categoria *Process Management* foi renomeada para *Management* e subdividida para reunir práticas e tecnologias que são de interesse do estudo, mas que primariamente visam apoiar a gestão do produto, de tarefas e de pessoas (códigos *Product*, *Tasks*, e *People*). A categoria de negócios (*Business*) foi criada a partir da

mesma ideia de mapeamento de apoio para outra área de interesse do projeto, mas que também apoia a equipe de desenvolvimento.

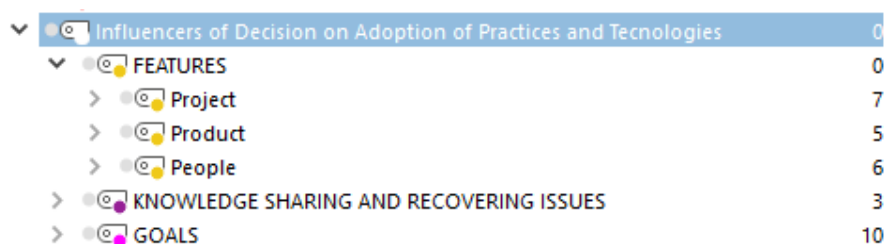
### 5.3.3 Mapeamento de fatores de influência na adoção de práticas e tecnologias para o registro e compartilhamento de ideias e características de software

Assim como no mapeamento das práticas e tecnologias, executou-se análise qualitativa nos 15 artigos selecionados para identificar os elementos de influência na decisão pela adoção das práticas e tecnologias mapeadas na etapa anterior. Os artigos selecionados foram relidos na íntegra nesta etapa, pois a leitura dos trechos de texto codificados na etapa anterior não seria suficiente para o propósito dessa etapa, visto que não necessariamente o texto do artigo apresenta suas motivações de implementação de tecnologia no mesmo trecho que descreve a aplicação da tecnologia.

O método de codificação descritiva também foi utilizado nessa fase, porém sem códigos provisórios pré-definidos. Ao final da leitura dos artigos, obteve-se uma lista de tópicos que surgiram como fatores de influência. Essa lista e seus segmentos codificados foram revisados repetidamente, passando por refinamento e agrupamento em busca de padrões de tópicos. Por fim, três categorias de tópicos foram identificadas para os fatores mapeados:

- I. Características do contexto de desenvolvimento (código: FEATURES).  
Os tópicos da categoria *Features* foram ainda agrupados em subcategorias de características de projeto, de produto ou de pessoas.
- II. Problemas relacionados ao compartilhamento ou recuperação de conhecimento de software (código: KNOWLEDGE SHARING AND RECOVERING ISSUES).
- III. Objetivos de melhoria da Startup quanto às práticas e tecnologias adotadas para compartilhamento e registro de conhecimento de software (código: GOALS).

A Figura 14 mostra a estrutura de códigos criada e a quantidade de artigos que citam fatores de influência em cada categoria.



▼ Influencers of Decision on Adoption of Practices and Technologies	0
▼ FEATURES	0
> Project	7
> Product	5
> People	6
> KNOWLEDGE SHARING AND RECOVERING ISSUES	3
> GOALS	10

Figura 14. Estrutura de Códigos de Fatores de Influência

A releitura dos trechos para refinamento dos códigos também é importante para se chegar numa descrição da influência de cada característica codificada na tomada de decisão sobre práticas e tecnologias. A sessão de resultados desse capítulo apresenta as descrições elaboradas a partir dos trechos extraídos.

### 5.3.4 Mapeamento da relação entre as práticas e tecnologias identificadas com os fatores de influência

Tendo as práticas e tecnologias de interesse do estudo mapeadas, assim como os fatores reportados como influentes na decisão de Startups em desenvolvimento de software pela adoção desses tipos de práticas e tecnologias, seguiu-se para a etapa final da análise, que visa a ligação entre esses dois conjuntos de resultados.

Primeiro, verificou-se a ocorrência dos códigos em cada categoria das tecnologias/práticas com as de *features/goals/issues* em cada artigo selecionado para o estudo. Utilizou-se, para isso, a visualização de matriz de códigos disponibilizada na ferramenta MaxQDA. A Figura 15 mostra uma visão dessa matriz de codificação, detalhando as categorias e subcategorias de tópicos, que serão detalhados na sessão de resultados deste capítulo.

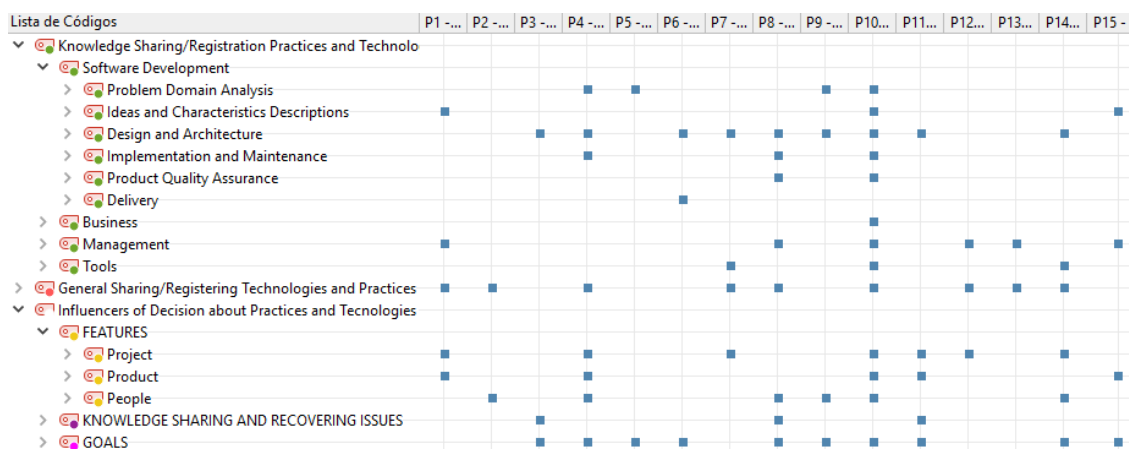


Figura 15. Matriz de Codificação: Práticas e Tecnologias, Fatores de Influências e Artigos

Os trechos dos códigos indicados na matriz de codificação foram relidos e, em caso de existir alguma relação, os trechos codificados com o tópico da prática/tecnologia foram também codificados com o fator de influência pertinente. A Figura 16 mostra um exemplo de ocorrência de codificação de uma prática adotada e o objetivo relacionado.

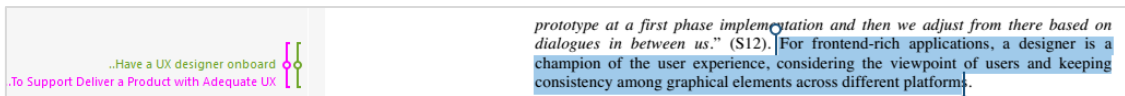


Figura 16. Exemplo de codificação sobreposta para representar relação de Prática/Tecnologia e Fator.

## 5.4 RESULTADOS

Nesta seção, são detalhados os resultados obtidos a partir dos procedimentos de análise, por pergunta de pesquisa.

### 5.4.1 RQ1: O que influencia a tomada de decisão sobre a adoção de práticas e tecnologias para compartilhamento e registro de conhecimento de ideias e características de software em Startups de desenvolvimento de software?

Como já citado na sessão sobre a análise dos dados, no conjunto de casos de desenvolvimento de software por Startups analisados neste estudo secundário, três tipos de elementos foram identificados como influenciadores da adoção das práticas e tecnologias para compartilhamento e registro de conhecimento de ideias e características de software: características do contexto, problemas relacionados com compartilhamento e recuperação de conhecimento e, por fim, objetivos de melhoria da Startup.

O Quadro 24 apresenta as características de contexto que as Startups desenvolvedoras de software estudadas consideraram importantes observar para tomar decisão sobre adoção de práticas e tecnologias no escopo do estudo. Nota-se que do total de doze características citadas, seis delas são observadas quanto ao seu impacto na viabilidade da equipe de desenvolvimento da Startup se apoiar no conhecimento tácito, de forma que as pessoas consigam obter o conhecimento necessário sobre as ideias e características de software para executar suas atividades. Os relatos indicam que as avaliações de tal viabilidade, assim como a situação das características citadas e seus impactos no contexto de desenvolvimento, são realizadas de forma subjetiva a partir da percepção dos profissionais envolvidos na decisão.

Quadro 24. Características Influentes

<b>Categoria: People</b>		
<b>Característica</b>	<b>Descrição da influência</b>	<b>Artigos Codificados</b>
<i>Team co-location</i>	A colocalização dos desenvolvedores favorece a comunicação frequente e / ou presencial para compartilhar conhecimento, substituindo a documentação por discussões informais e confiança em conhecimento tácito. Os benefícios da colocalização da equipe estão diretamente relacionados à capacidade da equipe de se comunicar por meio de reuniões frequentes, à união da equipe e prática de comunicação informal e presencial. Por outro lado, seus benefícios são inversamente proporcionais ao tamanho da equipe e à sua taxa de rotatividade.	[P2, P4,P10]
<i>Expectation of Changes in the Team</i>	Quando há mudanças na equipe, a falta de informações disponíveis dificulta a integração das novas pessoas na equipe e também o início de trabalho delas no projeto, o que diminui a produtividade.	[P4, P8, 10]
<i>Team communication frequency</i>	Quando a equipe de desenvolvimento da Startup tem a percepção de que a comunicação entre as pessoas é eficiente, havendo comunicação constante através de reuniões face a face ou com a ajuda de ferramentas de comunicação, a equipe reduz consideravelmente o registro de informações de software, ou mesmo não consideram valioso o esforço gasto em tal atividade.  A eficiência de comunicação da equipe é inversamente proporcional ao tamanho da equipe e à sua taxa de rotatividade.	[P10]
<i>Team Size</i>	Equipes pequenas de desenvolvimento ajudam as Startups a serem rápidas e flexíveis. Também facilita a comunicação e aumenta a confiança em conhecimento tácito.  Quando a equipe começa a crescer, a comunicação informal e a falta de documentação retardam a velocidade do processo de desenvolvimento.	[P4, P14, 10]
<i>Team Union</i>	A percepção de que a equipe é unida, juntamente com características de colocalização, baixa rotatividade e alta capacidade de comunicação, influenciam as Startups a decidirem por não implementar práticas de registro de conhecimento nos seus projetos de desenvolvimento.	[P10]

<b>Categoria: Product</b>		
<b>Característica</b>	<b>Descrição da influência</b>	<b>Artigos Codificados</b>
<i>Chance of Success in the Market</i>	Uma vez que a chance de aceitação do produto é considerada suficientemente positiva, a Startup pode decidir contratar novos profissionais, geralmente mais experientes. Isto pode levar a mudanças nas práticas para compartilhar e registrar informações sobre o software devido a expectativa de mudanças na equipe.	[P4, P10, P11]
<i>Complexity Increase</i>	Quando aumenta a complexidade de um produto que já foi validado com (possíveis) usuários, as alterações no código podem ser difíceis devido às más práticas de engenharia adotadas para executar iterações rápidas nas etapas anteriores. O medo de causar respostas ruins de usuários devido às alterações no produto validado leva as Startups a substituir as comunicações informais por sistemas rastreáveis.	[P4]
<i>Cost of Quality</i>	Quando o produto alcança um número significativo de usuários, a prioridade de desenvolver versões rapidamente para validá-los diminui e o custo de observar erros oriundos do uso do produto é maior, além do risco de diminuir o número de usuários. Assim, quando o produto escala, o custo da qualidade do produto aumenta e a Startup muda suas práticas para registrar o conhecimento sobre o software, a fim de garantir a qualidade do produto.	[P10]
<i>Integration with Third-Party Solutions</i>	Caso o software seja integrado com soluções de software e/ou hardware de terceiros, a Startup pode decidir pela elaboração de documentação detalhada de projeto do software, principalmente dos pontos de integração com soluções externas, para assim evitar dificuldades de possíveis mudanças futuras.	[P10, P15]
<i>Security Requirement</i>	Ao desenvolver software com requisitos de segurança, a Startup adota práticas para incluir processos e documentos adaptados ao seu contexto de recursos limitados e necessidade de desenvolvimento rápido.	[P1]
<b>Categoria: Project</b>		
<b>Característica</b>	<b>Descrição da influência</b>	<b>Artigos Codificados</b>
<i>Resource Limitation</i>	A limitação de recursos leva as equipes de Startups a desistir implantar práticas ou a adaptar práticas que envolvem documentar informações do software. Recursos de tempo, dinheiro e pessoas envolvidas na adoção de alguma prática e tecnologia são considerados na tomada de decisão.	[P1,P10,P11]
<i>Time to Deliver</i>	Devido ao pouco tempo para entregar o software (juntamente com outros fatores, como limitação de recursos e altas incertezas), as Startups priorizam a adoção de práticas informais de comunicação entre as pessoas da equipe. Por isso, quanto mais os fatores do contexto favorecem a eficiência de comunicação informal na equipe, menos formais serão as práticas e tecnologias adotadas para compartilhamento e registro das ideias e características do software.	[P4, P7, P10, P11, P12, P14]

Em alguns dos relatos analisados, a adoção de práticas e tecnologias para o compartilhamento e registro de conhecimento de software ocorreu devido à identificação da existência de problemas no contexto de desenvolvimento. Os problemas citados como influentes na decisão sobre adoção de práticas de registro e compartilhamento de informações de software estão descritos no Quadro 25. É importante notar que apenas três artigos, do conjunto de 15 selecionados no estudo secundário, reportaram ter adotado práticas/tecnologias devido a percepção de ocorrência de problemas de compartilhamento e recuperação de conhecimento. Não se pode afirmar, entretanto, que esses ou outros problemas do mesmo tipo não ocorram em Startups em contextos distintos do desenvolvimento de software, ou que não são percebidos ou ainda que não são levados em consideração por elas quando decidem por adotar práticas e tecnologias.

Quadro 25. Problemas Influentes

<b>Problema</b>	<b>Descrição</b>	<b>Artigos Codificados</b>
<i>Communication Across Diverged Mindsets Problem</i>	Nas Startups de desenvolvimento de software, as ideias para o produto de software podem vir de indivíduos com experiências e áreas profissionais diferentes, como por exemplo, da área de negócio, projetistas e desenvolvedores. Isto pode causar problemas de comunicação entre as partes interessadas.	[P3, P11]
<i>Comprehension Problem</i>	Dificuldade de compreensão representa a situação em que os modelos mentais dos desenvolvedores sobre a implementação do software são discordantes com seu comportamento real. Isto inclui a situação em que ninguém entende uma ou mais partes da implementação do software.	[P8]
<i>Design rationale Problem</i>	O problema de <i>Design rationale</i> ocorre quando ninguém na equipe se lembra por que algum componente de software se comporta da forma como o faz. A equipe pode entender o código, mas não sabe por que ele foi implementado, quão importante é para os (possíveis) clientes e quais (possíveis perfis de) clientes dependem do componente.	[P8]
<i>Knowledge Boundary Problem</i>	Ocorre quando as equipes de Startups enfrentam problemas em relação à falta de sintaxe comum (limite sintático) ou a ocorrência de diferentes interpretações para termos que são comuns para as pessoas da equipe (limite semântico).	[P3]
<i>Discoverability Problem</i>	O problema de dificuldade de descoberta de informação sobre o software ocorre quando os desenvolvedores negligenciam requisitos e restrições devido à documentação deficiente ou porque é difícil encontrar as especificações, o que os leva a quebrar requisitos funcionais ou outras dependências.	[P8]

Por fim, apesar de que em geral as escolhas de tecnologias e práticas adotadas por Startups de desenvolvimento de software serem fortemente influenciadas pelo objetivo estratégico de acelerar o desenvolvimento e entrega dos produtos, foi possível extrair dos relatos objetivos que as levaram a mudar suas práticas/tecnologias de compartilhamento e registro de conhecimento de software. O Quadro 26 descreve esses objetivos.

Quadro 26. Objetivos influentes

<b>Objetivo</b>	<b>Descrição</b>	<b>Artigos Codificados</b>
<i>To Achieve Common Understanding in the Team</i>	<p>Apesar da falta de documentação, as Startups de desenvolvimento de software se preocupam em criar na equipe uma perspectiva comum sobre as características do software em desenvolvimento/desenvolvido, o que pode significar incluir práticas de compartilhamento e registro de conhecimento mesmo ainda no estágio de ideação e durante a iteração de desenvolvimento.</p> <p>Outra preocupação é manter o entendimento da equipe consistente e atualizada sobre a proposição de valor do produto, para que a equipe expresse tal valor por meio de suas decisões de projeto e mantenha novas decisões consistentes com as existentes.</p>	[P3, P8, P10]
<i>To Increase or Reinforce the Team Communication Ability</i>	<p>Como as equipes de Startups de desenvolvimento de software geralmente contam com conhecimento tácito para diminuir o tempo das iterações de desenvolvimento, a capacidade dos desenvolvedores de expressar ideias e se comunicar de forma eficiente e construtiva é observada pelas Startups. As práticas e tecnologias adotadas para apoiar as Startups a atingir esse objetivo visam facilitar cada desenvolvedor obter todas as informações e conhecimentos que precisa de outros para progredir na sua tarefa.</p>	[P4, P8, P10, P11, P14, P15]
<i>To Support the Development of a Product with Adequate UX</i>	<p>Para algumas Startups, a Experiência do Usuário (UX) é o aspecto de qualidade mais importante do produto desenvolvido, em particular as características de facilidade de uso, atratividade da interface do usuário e fluxo sem interrupções. Nesse caso, a Startup adota práticas que envolvem o registro de informações do software que podem envolver especificações e prototipações de interfaces de usuário, restrições e comportamento esperados que são relacionados com UX, como exemplo, a consistência entre as interfaces de usuário do software.</p>	[P5, P6, P9, P11]



Objetivo	Descrição	Artigos Codificados
<i>To Support Product and Project Management</i>	Como o software desenvolvido tem suas características alteradas e/ou evoluídas a partir do retorno de aprendizado validado, além de muitas vezes suas características serem elaboradas em durante projeto e codificação, o registro das ideias e características do software ajudam a rastrear a evolução do produto em termos das características funcionais e de estrutura de código das suas versões.	[P3, P10]

No conjunto de dados do estudo, tais objetivos não foram necessariamente determinados devido à percepção de ocorrência de problemas no contexto de desenvolvimento, apesar de ser possível notar alguma relação entre eles pela sua descrição. Por exemplo, é factível que uma Startup defina como objetivo de melhoria “*To Achieve Common Understanding in the Team*” devido à percepção da ocorrência de problemas relacionados a “*Comprehension*” ou de “*Design rationale*”. Por outro lado, é possível também que a Startup esteja tomando ações preventivas destes problemas.

Durante a análise dos dados, nem todos os tipos de elementos influentes foram observados em cada relato analisado. Para melhor visualizar a ocorrência desses elementos, cada caso de contexto Startup relatado nos artigos foi identificado, resultando num conjunto de 39 contextos. Trinta e cinco contextos são de Startups específicas, enquanto quatro são dados reportados de um conjunto de Startups observadas conjuntamente. Os elementos influentes foram mapeados em relação aos contextos que se relacionam, como mostra a Figura 17. Dados gerais descritivos dos contextos mapeados estão disponibilizados no Apêndice 4

Em apenas um dos contextos (id. 26) ocorreu a influência tanto de características quanto da ocorrência de problemas e de objetivos de melhoria na tomada de decisão sobre as práticas e tecnologias adotadas. Nos outros contextos foi possível observar a ocorrência de diferentes combinações de tipos de elementos, de apenas um tipo de elemento e até mesmo nenhum tipo de elemento, como é o caso de seis dos 39 contextos mapeados. Isto significa que o artigo relatou práticas e/ou tecnologias utilizadas no contexto, mas não explicitou o que influenciou a adoção das mesmas.

	Id. dos Casos Reportados																																										
Fator	1	2	3	4	5	6	7	8	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
<b>Características</b>																																											
<i>Team co-location</i>																	X	X	X					X						X					X		X						
<i>Expectation of Changes in the Team</i>	X																	X	X						X				X														
<i>Team communication frequency</i>	X																X		X					X					X							X							
<i>Team Size</i>																																											
<i>Team Union</i>																				X									X														
<i>Chance of Success in the Market</i>	X											X	X	X	X	X		X	X																								
<i>Complexity Increase</i>																		X																									
<i>Cost of Quality:</i>																								X																			
<i>Integration with Third-Party Solutions</i>																	X																					X					
<i>Security Requirement</i>																												X															
<i>Resource Limitation</i>	X																X		X					X	X			X	X							X							
<i>Time to Deliver</i>	X						X	X	X	X	X	X	X	X	X	X	X	X	X					X						X							X	X					
<b>Problemas</b>																																											
<i>Communication Across Diverged Mindsets</i>													X																										X	X			
<i>Comprehension</i>																											X																
<i>Design rationale</i>																										X																	
<i>Knowledge Boundary</i>																				X	X																						
<i>Discoverability</i>																									X																		
<b>Objetivos</b>																																											
<i>To Achieve Common Understanding in the Team</i>	X																		X	X	X		X	X			X	X								X							
<i>To Increase or Reinforce the Team Communication Ability</i>								X	X	X	X	X					X	X						X														X					
<i>To Support the Development of a Product with Adequate UX</i>					X							X											X					X							X								
<i>To Support Product and Project Management</i>																			X	X	X								X														

Figura 17. Ocorrência de fatores de influência em contextos de Startups mapeados

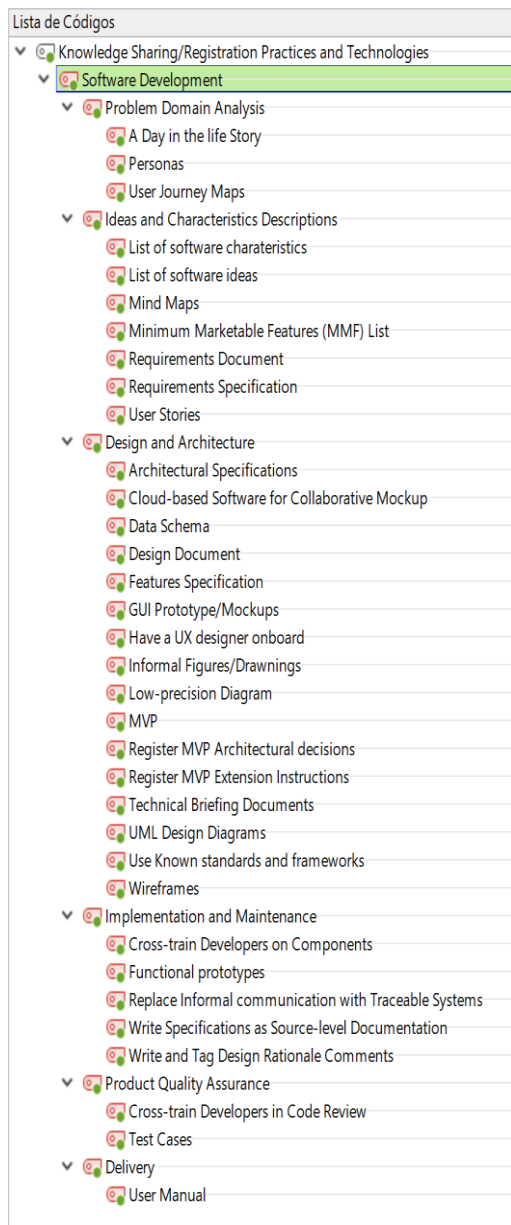
#### **5.4.2 RQ2: Quais são as tecnologias e práticas adotadas para compartilhamento e registro de conhecimento sobre ideias e características de software em Startups de desenvolvimento de software?**

Com esta pergunta de pesquisa, buscou-se observar se existe um padrão de práticas e tecnologias adotadas pelas Startups de desenvolvimento de software. Em um baixo nível de granularidade, não foi encontrado tal padrão nos artefatos, práticas e ferramentas citados nos artigos selecionados para o presente estudo. Porém, foi possível identificar duas categorias principais de práticas e tecnologias: i) as que são adotadas especificamente para apoiar as atividades de desenvolvimento de software, que são parte do interesse principal do estudo, juntamente com os fatores que levam à sua adoção, e ii) as que são parte da cultura de Startups para compartilhamento de conhecimento em geral, ou seja, não somente de ideias e características de software. Por ser parte da cultura, tais práticas podem influenciar como as práticas e tecnologias para desenvolvimento de software são implementadas.

A Figura 18.a) apresenta as práticas e tecnologias adotadas e que são parte da categoria i) supracitada. As tecnologias estão identificadas pelos seus nomes, enquanto que as práticas estão identificadas por uma frase contendo verbo de ação.

A maioria das práticas e tecnologias são informais, ou promovem práticas informais de compartilhamento e registro de conhecimento. Ainda assim, percebe-se o uso de especificações de requisitos e de arquitetura. Além disso, nota-se a adoção de práticas para ajudar na manutenibilidade do produto, a despeito da comum falta de documentação. Exemplo disso são as práticas de treinamento cruzado em componentes do software e revisão de código, o registro de decisões arquiteturais para cada MVP desenvolvido e a inclusão de comentários no código com marcações que as ligam a trechos de código relacionados.

Apesar da variedade de práticas e tecnologias citadas, apenas dez delas foram adotadas em mais de um dos 39 contextos de Startups software mapeados, como pode ser visto na Figura 18.b), que lista essas práticas e indica a quantidade de contextos em que foram mapeadas.



a) Práticas e Tecnologias reportadas

b) Práticas e Tecnologias reportadas em mais de um contexto Startup e a quantidade de contextos mapeados

GUI Prototype/Mockups	18
Design Document	4
Personas	2
Low Precision Diagrams	2
User Journey Maps	2
Have a UX Designer Onboard	2
Requirements Document	2
Mind Map	2
Requirements Specification	2
Features Specification	2

Figura 18. Práticas e Tecnologias Reportadas com foco em Desenvolvimento de Software

Também foram reportadas práticas e tecnologias de apoio para outras áreas envolvidas no projeto de software da Startup, mas que além disso são utilizadas para compartilhar/registrar conhecimento de ideias e características de software para a equipe de desenvolvimento. Por exemplo, em algumas Startups, as descrições das tarefas planejadas são utilizadas para detalhar as características a serem desenvolvidas. Outro exemplo é a equipe utilizar documentos feitos para divulgação do produto, e como tal apresenta as funcionalidades para serem compreendidas por possíveis usuários, como documento para obter a visão geral e objetivos do software que desenvolvem, além das funcionalidades. A Figura 19 apresenta estas práticas e tecnologias, além de outras encontradas.

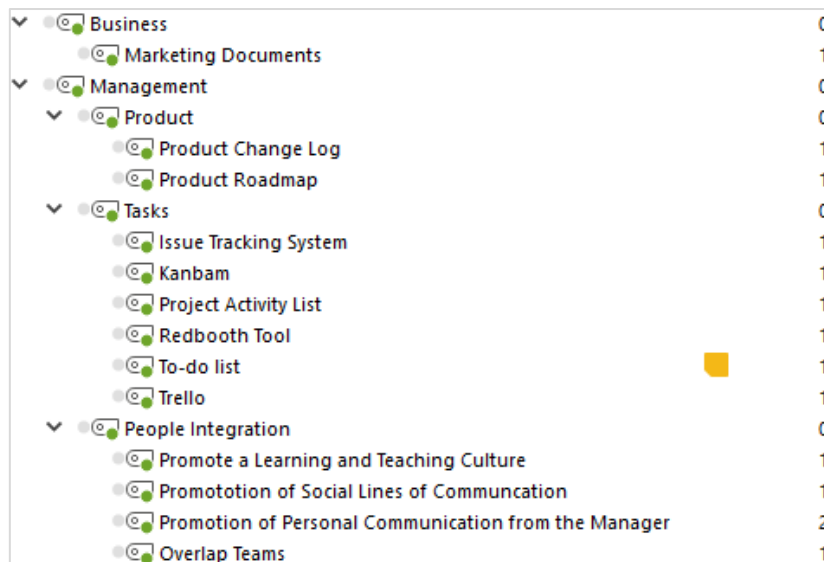


Figura 19. Práticas e Tecnologias Reportadas com foco em Negócio e Gestão

Por sua vez, a Figura 20 apresenta as práticas e tecnologias citadas para compartilhamento e/ou registro de conhecimento em geral nas Startups de desenvolvimento de software. Quando adotadas por uma Startup desta natureza, tais práticas impactam na maneira em que as demais práticas de compartilhamento e registro de conhecimento de software são implementadas. Por exemplo, se a equipe da Startup cria o artefato Especificação de Requisitos, ela o salva utilizando ferramentas integradas de compartilhamento e versionamento de arquivos. Outro exemplo é o caso de a Startup ter como objetivo de melhoria “*To Increase or Reinforce the Team Communication Ability*” e ao mesmo tempo ter como prática geral a preferência por práticas informais de comunicação. Neste caso, a Startup adotou práticas de promoção das ligações sociais entre os desenvolvedores.

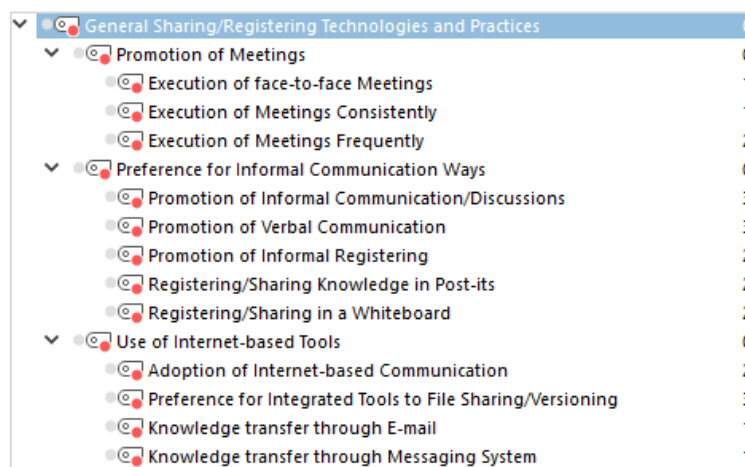


Figura 20. Práticas e Tecnologias de Compartilhamento e/ou Registro de Conhecimento

## 5.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Grande parte das práticas mapeadas apoiam (ou se apoiam na) a comunicação informal, ou são para melhoria das habilidades de comunicação, o que é característico do contexto de Startups. Ainda assim, foram mapeadas práticas de registro de conhecimento de software que, apesar de parte das práticas e tecnologias serem adotadas em poucos contextos, até mesmo somente em um, tais contextos são de Startups reais, ou seja, representam aplicações na indústria.

É importante ponderar que as práticas, tecnologias e fatores de influência reportados são o que as organizações citaram. Representam suas crenças sobre o fenômeno estudado, o que é valioso por serem advindas da experiência prática. Porém, do ponto de vista científico, carecem de estudos que analisem a relação entre os fatores e as práticas e tecnologias. Assim, os resultados podem ajudar a elaborar um conjunto de conjecturas de pesquisa na área. Como contribuição para a indústria, o mapeamento organiza um corpo de conhecimento sobre as tecnologias em uso por Startups de desenvolvimento de software.

# 6 MODELO CONCEITUAL PARA DEFINIÇÃO DE CONTEXTO STARTUP DE ADOÇÃO DE PRÁTICAS E TECNOLOGIAS DE REGISTRO E COMPARTILHAMENTO DE INFORMAÇÃO DE SOFTWARE

Este capítulo descreve o modelo conceitual para definição de contexto Startup visando apoiar a tomada de decisão sobre adoção de práticas e tecnologias de compartilhamento e registro de informação de software para Startups de software, segundo a definição de Startups descrita no capítulo 2 desta tese.

## 6.1 INTRODUÇÃO

Unterkalmsteiner et al. (UNTERKALMSTEINER, 2016) indicam que um dos desafios de elaborar um conjunto de práticas e tecnologias para o contexto de Startups e transferir o conhecimento sobre essas tecnologias é ter um modelo do contexto de engenharia das Startups. A partir da definição do contexto, em granularidade tal que permita comparar Startups, práticas candidatas podem ser identificadas e estudadas quanto à evidência de seus benefícios esperados.

Definir o contexto de engenharia de Startups não é uma tarefa trivial, visto que se trata de um contexto intensamente flexível e reativo quanto às práticas adotadas. Tal flexibilidade e reatividade são necessárias, devido às diversas influências na definição das características do software, como por exemplo: demandas de clientes obtidos, mudanças de mercado e de tecnologias (SUTTON, 2000); suposições e definições do modelo de negócio, que é desenvolvido em paralelo ao produto de software (RIES, 2011), (BLANK, 2013b); resultados do ciclo de aprendizado validado, ou experimentação contínua (BOSCH, 2012), que podem redirecionar a visão e objetivos da solução desenvolvida em software; além do ecossistema em que a Startup está inserida, pela qual ela pode estabelecer parcerias que podem influenciar tanto no modelo de negócios quanto nas práticas de desenvolvimento (KON, CUKIER, *et al.*, 2014).

Khurum, Fricker e Gorschek (KHURUM, FRICKER e GORSCHKEK, 2015) observaram em organizações inovadoras - conjunto esse que as organizações Startup fazem parte - diferentes combinações de fatores, até mesmo ausência de fatores, que

eram considerados na literatura técnica como importantes para gerar inovação. Ao não conseguir encontrar evidências de um padrão de ocorrência desses fatores em organizações que atingiram seus objetivos de inovação, o autor discorre suas conclusões sobre a natureza qualitativa da inovação.

Diante da complexidade para se definir o contexto de engenharia de Startups, nesta pesquisa de doutorado adotou-se a estratégia de abstração do contexto, concentrando na observação de um fenômeno específico: a tomada de decisão sobre práticas e tecnologias de compartilhamento e registro de informação de software para apoiar a equipe de desenvolvimento de software em Startups.

Dado o pouco conhecimento sobre todos os elementos envolvidos no contexto de engenharia de Startups, assim como sobre quais são os seus relacionamentos relevantes para quais fenômenos, esta tese propõe que, para apoiar a transferência de tecnologias e organizar um corpo de conhecimento sobre práticas aplicadas, um **contexto de Startup de software seja definido para um fenômeno/problema de interesse**. No caso desta tese, o problema de interesse é a tomada de decisão sobre práticas e tecnologias a adotar pela equipe de desenvolvimento para compartilhamento e registro de informação de ideias e características de software. Propõe-se também que o **apoio à transferência de tecnologia se dê pela disponibilização de dados de contextos de Startups de software** segundo o modelo, de forma que Startups de software possam obter conhecimento sobre quais práticas e tecnologias adotadas em quais contextos.

O estudo exploratório com entrevistas semiestruturadas em seis Startups de software (capítulo 4) resultou em um conjunto inicial de fatores de influência na tomada de decisão. O conjunto de fatores foi expandido e reorganizado a partir dos resultados de um mapeamento sistemático da literatura (descrito no capítulo 5) acerca de práticas adotadas para desenvolvimento de software em Startups. É importante enfatizar que este mapeamento sistemático da literatura incluiu como fonte de dados os resultados do estudo exploratório por meio da publicação de Nascimento e Travassos (NASCIMENTO e TRAVASSOS, 2017), recuperada pela estratégia de busca do mapeamento sistemático.

A subseção 6.2 a seguir detalha a proposta do modelo conceitual de contexto de Startup de software, incluindo as escalas definidas para os atributos de contexto do modelo. Na subseção 6.3 é apresentado um estudo de avaliação sobre a pertinência dos atributos influentes organizados seguindo o modelo conceitual de contexto de Startup de software definido. Os trabalhos relacionados são discutidos na seção 6.4. Por fim, as considerações finais são apresentadas na seção 6.5.



## 6.2 MODELO DE CONTEXTO DE STARTUP DE SOFTWARE

O modelo de contexto de Startup de software aqui definido representa o estado da Startup de software, em um determinado momento, quanto aos seus elementos de interesse para o fenômeno de adoção de práticas e tecnologias de registro e compartilhamento de informação de software. Com base nos estudos executados e na literatura técnica sobre as características de Startups de software, os elementos do modelo são:

- a) **Atributos que caracterizam** uma Startup de software, dos quais fazem parte as características influentes na tomada de decisão.
- b) Práticas e tecnologias adotadas como **apoio ao registro e compartilhamento de informação** de software e que são relacionadas aos atributos influentes.

Os elementos do modelo estão organizados como ilustrado pela Figura 21 e são detalhados nas subseções a seguir.

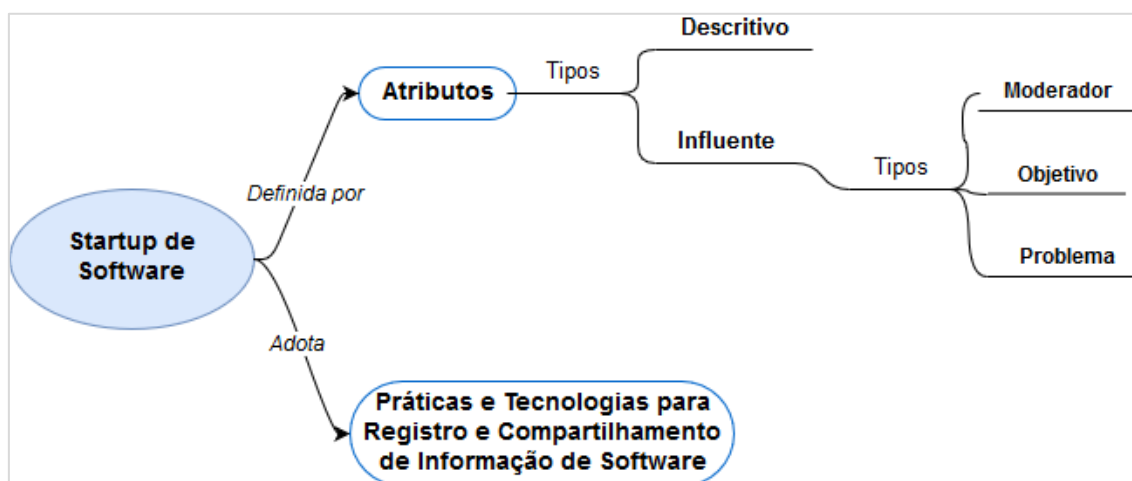


Figura 21. O Contexto de Startup de Software na Adoção de Práticas e Tecnologias para Registro e Compartilhamento de Informação de Software

### 6.2.1 Atributos que caracterizam uma Startup de software

Os estudos executados durante a pesquisa identificaram elementos que influenciaram Startups na seleção de suas práticas e tecnologias de registro e compartilhamento de informação de software. Como resultado do mapeamento sistemático descrito no capítulo 4, esses elementos foram organizados em: i) atributos moderadores da tomada de decisão; ii) objetivos de melhoria relacionados com compartilhamento e recuperação de informação, e iii) problemas relacionados com compartilhamento e recuperação de informação de software e que afetam a equipe de desenvolvimento. Estes elementos formam o conjunto de tipos de **atributos influentes**

na tomada de decisão da Startup, cujos valores atribuídos aos seus itens caracterizam um contexto de Startup de software.

Atributos influentes moderadores são características do contexto da Startup de software cujos valores correntes influenciam na adoção e no grau de informalidade de práticas e/ou tecnologias adotadas pela Startup para registro e compartilhamento de informação de software. O Quadro 27 a seguir, apresenta o conjunto de moderadores extraídos a partir do mapeamento sistemático da literatura sobre desenvolvimento de software em Startups (capítulo 5 desta tese). Os moderadores foram organizados em categorias, segundo o tipo de elemento do processo de desenvolvimento de software que se relacionam: Pessoas, Produto, Projeto.

Quadro 27. Moderadores da Tomada de Decisão sobre Adoção de Práticas e Tecnologias de Registro e Compartilhamento de Informação de Software

<b>PESSOAS</b>
<ul style="list-style-type: none"> <li>• NÍVEL DE CO-LOCALIZAÇÃO DA EQUIPE DE DESENVOLVIMENTO</li> <li>• EXPECTATIVA DE MUDANÇAS NA EQUIPE DE SOFTWARE</li> <li>• FREQUÊNCIA DE COMUNICAÇÃO DA EQUIPE DE SOFTWARE</li> <li>• QUANTIDADE DE PESSOAS NA EQUIPE DE DESENVOLVIMENTO</li> <li>• NÍVEL DE UNIÃO DA EQUIPE</li> </ul>
<b>PRODUTO</b>
<ul style="list-style-type: none"> <li>• NÍVEL DE INCERTEZA SOBRE A ACEITAÇÃO DO PRODUTO NO MERCADO</li> <li>• NÍVEL DE AUMENTO DA COMPLEXIDADE DO SOFTWARE</li> <li>• CUSTO DA GARANTIA DA QUALIDADE DO PRODUTO SEM APOIO DE DOCUMENTOS E/OU MODELOS DO SOFTWARE</li> <li>• INTEGRAÇÃO COM SOLUÇÕES DE HARDWARE E/OU SOFTWARE DE TERCEIROS</li> <li>• NÍVEL DA CRITICIDADE DE REQUISITO DE SEGURANÇA</li> </ul>
<b>PROJETO</b>
<ul style="list-style-type: none"> <li>• NÍVEL DA LIMITAÇÃO DE RECURSOS</li> <li>• NECESSIDADE DE EXECUTAR CICLOS CURTOS DE DESENVOLVIMENTO E ENTREGA DE VERSÕES DO SOFTWARE</li> </ul>

Os atributos influentes do tipo objetivo representam oportunidades de melhorias identificadas para a Startup de software e para as quais deseja-se que práticas e/ou tecnologias de registro e compartilhamento de informação de software apoiem o alcance. Por sua vez, os atributos influentes do tipo problema são situações indesejáveis que ocorrem na Startup, reconhecidas como relacionadas à ineficiência ou falta de registro e/ou compartilhamento de informações de software. O Quadro 28 e o Quadro 29 apresentam, respectivamente, os objetivos de melhoria e problemas extraídos dos relatos de contexto de desenvolvimento em Startups de software que foram fontes de dados do mapeamento sistemático descrito no capítulo 5 desta tese.

Quadro 28. Objetivos de Melhoria influentes na Tomada de Decisão sobre Adoção de Práticas e Tecnologias de Registro e Compartilhamento de Informação de Software

AUMENTAR OU REFORÇAR A HABILIDADE DE COMUNICAÇÃO DA EQUIPE.
ALCANÇAR ENTENDIMENTO COMUM SOBRE O SOFTWARE PELA EQUIPE DE DESENVOLVIMENTO.
APOIAR O DESENVOLVIMENTO DE SOFTWARE COM NÍVEL ADEQUADO DE EXPERIÊNCIA DE USUÁRIO.
APOIAR A GESTÃO DE PROJETO E/OU DO PRODUTO.

Quadro 29. Problemas influentes na Tomada de Decisão sobre Adoção de Práticas e Tecnologias de Registro e Compartilhamento de Informação de Software

DIFICULDADE DE COMUNICAÇÃO ENTRE INTERESSADOS POR SEREM DE DIFERENTES ÁREAS DE TRABALHO (EX.: NEGÓCIO, DESENVOLVIMENTO, DESIGNER DE UX).
DIVERGÊNCIA ENTRE O ENTENDIMENTO DOS DESENVOLVEDORES SOBRE A IMPLEMENTAÇÃO DO SOFTWARE E O REAL FUNCIONAMENTO DE PARTES OU DE TODO O SOFTWARE.
FALTA DE CONHECIMENTO DA EQUIPE SOBRE A IMPORTÂNCIA QUE AS CARACTERÍSTICAS DO SOFTWARE TÊM PARA O NEGÓCIO E PARA OS CLIENTES.
FALTA DE SINTAXE COMUM ENTRE AS PESSOAS DA EQUIPE, OU OCORRÊNCIA DE DIFERENTES INTERPRETAÇÕES PARA TERMOS EM COMUM.
DIFICULDADE DOS DESENVOLVEDORES EM DESCOBRIR INFORMAÇÕES SOBRE DEPENDÊNCIAS DE PROJETO DO SOFTWARE, LEVANDO-OS A IGNORAR REQUISITOS E RESTRIÇÕES.

Além dos atributos influentes supracitados, em pesquisa *ad-hoc* da literatura foram identificados atributos que caracterizam Startups de software, mas que, nos estudos executados, não foram citados como diretamente influentes na tomada de decisão sobre registro e compartilhamento de informação de software. Assim, no escopo de pesquisa desta tese, fazem parte do conjunto de **atributos descritivos da Startup de Software**. São eles: tempo de atuação (SUTTON, 2000), estágio do projeto de inovação (CROWNE, 2002), (BLANK, 2013b), porte da Startup (CROWNE, 2002), (BLANK, 2013b), experiência de validação do produto no mercado (RIES, 2011), (BOSCH, 2012), (BLANK, 2013a), (BOSCH, 2013) e suporte externo e/ou parcerias estabelecidas (KON, CUKIER, *et al.*, 2014). Além destes, para fins de comparação de foram incluídos como atributos descritos a região da Startups de software, a metodologia de desenvolvimento de software e o domínio da aplicação.

## 6.2.2 Práticas e tecnologias adotadas

O modelo de contexto de Startup de software definido para apoiar a tomada de decisão e possível transferência de tecnologia tem como elementos principais as práticas e tecnologias adotadas em Startups para compartilhar ou registrar informação de software de interesse para a equipe de desenvolvimento, que são relacionadas a um ou mais dos elementos influentes da decisão.

Durante o mapeamento sistemático descrito no capítulo anterior, foi extraído de relatos da literatura técnica um conjunto de práticas e tecnologias, como mostra o Quadro 30.

Quadro 30. Práticas e Tecnologias para Registro e Compartilhamento de Informação de Software

PRÁTICAS
<ul style="list-style-type: none"><li>• Escrever especificações documentando em nível-fonte. Exemplo: escrever comentários em tabelas de banco de dados.</li><li>• Escrever comentários com <i>tags</i>, no código-fonte.</li><li>• Registrar as decisões arquiteturais do MVP.</li><li>• Registrar instruções para extensão do MVP.</li><li>• Ter especialista de UX participando na equipe de ideação.</li><li>• Realizar treinamento cruzado de desenvolvedores sobre os componentes do software.</li><li>• Treinamento cruzado de desenvolvedores em revisão de código.</li><li>• Trocar a comunicação informal por sistemas rastreáveis.</li><li>• Utilizar padrões e frameworks conhecidos</li></ul>
TECNOLOGIA - ARTEFATOS
<p>Para Registro de Conhecimento sobre Domínio do Problema e Visão de Solução:</p> <ul style="list-style-type: none"><li>• A Day in Life Story</li><li>• Documento de <i>Technical Briefing</i></li><li>• Mapa de Jornada do Usuário</li><li>• Personas</li></ul> <p>Para Descrever Ideias e Características do Software:</p> <ul style="list-style-type: none"><li>• Documento com propósito de <i>Marketing</i></li><li>• Documento ou Especificação de Requisitos</li><li>• Estórias de Usuário</li><li>• Lista de Características do Software</li><li>• Lista de Características Mínimas para o Mercado (<i>Minimum Marketable Features - MMF</i>)</li><li>• Lista de Ideias para o Software</li><li>• Manual do Usuário</li><li>• Lista de Tarefas (<i>To-do List</i>)</li><li>• Lista de Atividades do Projeto</li></ul> <p>Para Registro de Informação de Projeto e Desenvolvimento do Software:</p> <ul style="list-style-type: none"><li>• Desenhos e Figuras Informais</li><li>• Diagramas de baixa precisão</li><li>• Diagramas de Projeto da UML (ex.: Classes, pacotes)</li></ul>

<ul style="list-style-type: none"> <li>• Documento de Projeto do Software</li> <li>• Especificação Arquitetural</li> <li>• Especificação de Características</li> <li>• Esquema de Dados</li> <li>• Log de Mudanças no Software</li> <li>• Produto Mínimo Viável - MVP</li> <li>• Protótipos de Telas /<i>Mockups</i></li> <li>• Protótipos Funcionais</li> <li>• <i>Roadmap</i> do Produto</li> <li>• <i>Wireframes</i></li> <li>• Casos de Teste</li> <li>• FREQUÊNCIA DE COMUNICAÇÃO DA EQUIPE DE SOFTWARE</li> </ul>
<b>TECNOLOGIA - FERRAMENTAS</b>
<ul style="list-style-type: none"> <li>• Ferramentas de Mockup, colaborativas e na nuvem</li> <li>• <i>Issue Tracking System</i></li> <li>• Kanban</li> <li>• Ferramentas de Gestão de Tarefas. Ex.: Trello, Redbooth.</li> </ul>

### 6.2.3 Escala dos Atributos de Contexto Startup

O modelo de contexto definido foi elaborado para possibilitar que Startups de software comparem seus contextos com os de outras Startups e analisem se as práticas e tecnologias aplicadas nesses outros contextos são candidatas para adoção em seu contexto. Portanto, é requisito do modelo de contexto de Startup de software elaborado que ele possibilite a:

- Caracterização do contexto da Startup que deseja obter informações de práticas e tecnologias para compartilhamento e registro de informação de software.
- Comparação de contextos Startups com outros contextos previamente inseridos em uma base de informação.
- Recuperação de contextos Startups com informação de nível de semelhança com a Startup.

Para atender a esses requisitos, faz-se necessário definir escalas para os valores dos atributos do contexto de Startup de software. Os valores dos atributos descritivos possuem escala nominal, mesmo para o atributo Tempo de Atuação. A escolha da escala justifica-se pela falta de padrão nos relatos obtidos no mapeamento, nos quais o tempo de atuação da Startup foram, quando informados, indicados de diferentes maneiras, como por exemplo: pelo tempo em anos/meses ou pelo ano da fundação da Startup. Além disso, os atributos descritivos foram incluídos no modelo de contexto como informações adicionais das características de uma Startup de software, não sendo utilizados, como são os atributos influentes, para comparação de contexto

relacionado à tomada de decisão sobre práticas e tecnologias de compartilhamento e registro de informação de software.

No caso dos atributos influentes, três tipos de valores possíveis foram identificados:

- Valor numérico inteiro, para a característica Quantidade de Pessoas da Equipe de Desenvolvimento. Portanto, a escala para esta característica é a de razão.
- Valor categórico (Sim / Não) para a característica Integração com Soluções De Hardware e/ou Software de Terceiros, sendo então de escala nominal.
- As demais características influentes são valoradas de forma qualitativa, ou seja, elas capturam a percepção do avaliador quanto ao seu valor no contexto. Para elas, decidiu-se pelo uso da Escala Visual Analógica (VAS - *Visual Analogue Scale*) (WEWERS e LOWE, 1990), detalhada no parágrafo a seguir.

A escala VAS apoia a medição de características ou atitudes que são avaliadas subjetivamente, porém cujas possibilidades de valores estão em um espaço contínuo que é delimitado por dois extremos: o de ausência de valor e o de ocorrência máxima de valor. Um exemplo de uso da escala VAS é encontrado na medicina, quando se possibilita que um paciente indique a intensidade da dor que está sentindo, podendo ser um valor entre “Sem dor” e “o máximo de dor possível” ou um ponto entre os valores 0(zero) e 10(dez), em que dez representa o máximo de dor.

Apresenta-se a escala VAS através de uma linha com seus extremos rotulados. Segundo Wewers e Lowe (WEWERS e LOWE, 1990), a forma mais usual é a utilização de uma linha horizontal de dez cm, na qual o respondente indica o ponto que representa sua percepção da característica que está avaliando. O valor da resposta é calculado pela distância entre o ponto marcado pelo respondente e o ponto extremo de ausência de valor. Por ter esse ponto de ausência de valor e ser possível quantificar as distâncias entre seus pontos, a escala VAS permite a realização de transformações matemáticas em seus valores, como por exemplo, o cálculo de média.

A Figura 22 mostra a escala VAS definida para os atributos influentes do modelo de contexto de Startup de software definido nesta tese, exceto para os atributos Quantidade de Pessoas da Equipe de Desenvolvimento e Integração com Soluções De Hardware e/ou Software de Terceiros. Com o uso da escala, avalia-se no momento referente à caracterização do contexto de Startup de software a percepção do(a):

- Nível dos atributos influentes moderadores (ex.: União da Equipe)
- Intensidade da ocorrência de problemas que são influentes na decisão de adoção das práticas e tecnologias
- Nível de prioridade de objetivos de melhoria.

Figura 22. Escala VAS

### 6.3 ESTUDO DE AVALIAÇÃO: PERTINÊNCIA DOS ATRIBUTOS INFLUENTES DO MODELO DE CONTEXTO DE STARTUP DE SOFTWARE

O modelo de contexto de Startup de software, elaborado nesta tese, está fundamentado em evidências coletadas no estudo executado com seis Startups de software e também no mapeamento sistemático pelo qual, a partir de quinze artigos selecionados, trinta e cinco contextos de Startups específicas foram extraídos e analisados. Porém, em nenhum dos contextos estudados todos os atributos do modelo foram citados, seja por motivo de limitação de observação na execução do estudo e das pessoas participantes ou por o atributo não ser relevante para a Startup. Além disso, os atributos foram nomeados durante o processo de análise temática, que é uma metodologia de análise qualitativa, logo seus resultados são sujeitos à interpretação do pesquisador que realiza a análise. Assim, faz-se necessário observar a pertinência dos atributos definidos pelo modelo de contexto de Startup de software, assim como da adequação das escalas. Esta seção descreve o estudo de avaliação cujo objetivo é descrito no Quadro 31, seguindo a estrutura GQM (BASILI, CALDIERA e ROMBACH, 2001):

Quadro 31. Objetivo do Estudo de Avaliação

Analisar	Os atributos do modelo de contexto Startup
Com o propósito de	Caracterizar
Em relação a	sua eficácia em capturar os fatores de influência na adoção de práticas e tecnologias de compartilhamento e registro de informação de software
Do ponto de vista	Gerentes de projeto ou líderes de equipes de desenvolvimento de software
No contexto de	Startups de software.

### 6.3.1 Plano de Estudo

#### 6.3.1.1 Materiais

Para o estudo, foi elaborado um instrumento para a capturar a percepção dos participantes acerca da pertinência dos atributos influentes, capturados no mapeamento sistemático, na tomada de decisão sobre práticas e tecnologias de apoio a compartilhamento e registro de informação de software. Além disso, o instrumento captura a percepção do respondente quanto ao estado corrente do atributo no contexto da Startup que participa. A Figura 23 a seguir apresenta parte do instrumento, exemplificando seu formato para os atributos influentes, a saber: atributos moderadores (características de produto, projeto e pessoas), problemas relacionados ao compartilhamento e à recuperação de informação de software e, por fim, objetivos de melhorias.

Análise das Características do Contexto da Startup	
Não observada ( )	<b>Equipe</b> Co-localização da Equipe (0 a 10): _____ <i>Inexistente</i> <span style="float: right;"><i>Elevada</i></span>
Não observada ( )	Expectativa de Mudanças na Equipe de Software (0 a 10): _____ <i>Inexistente</i> <span style="float: right;"><i>Elevada</i></span>

Análise da Ocorrência de Dificuldades de Compartilhamento e/ou Registro de Conhecimento sobre Características de Software:	
Não observada ( )	Dificuldade de comunicação entre interessados por serem de diferentes áreas de trabalho (ex.: negócio, desenvolvimento, designer de UX) (0 a 10): _____ <i>Inexistente</i> <span style="float: right;"><i>Elevada</i></span>
Não observada ( )	Divergência entre o entendimento dos desenvolvedores sobre a implementação do software e o real funcionamento de partes ou de todo o software. (0 a 10): _____ <i>Inexistente</i> <span style="float: right;"><i>Elevada</i></span>

Análise de Objetivos Atuais de Melhoria no Compartilhamento e/ou Registro de Conhecimento sobre Características de Software:	
Não observado ( )	Aumentar ou reforçar a habilidade de comunicação da equipe (0 a 10): _____ <i>irrelevante</i> <span style="float: right;"><i>Importantíssimo</i></span>
Não observado ( )	Alcançar entendimento comum sobre o software pela equipe de desenvolvimento (0 a 10): _____ <i>irrelevante</i> <span style="float: right;"><i>Importantíssimo</i></span>

Figura 23. Instrumentos do Estudo



Foi também elaborado um instrumento para capturar os atributos não influentes da Startup e outro para caracterizar o participante. Estes instrumentos estão disponibilizados no Apêndice 3 do presente documento.

### **6.3.1.2 Procedimentos**

O estudo foi planejado para ser executado como entrevistas semi-estruturadas e presenciais, para que permitisse a observação da utilização dos instrumentos pelos participantes. Os procedimentos de execução planejados foram organizados nas seguintes etapas:

1. Apresentação do tema e escopo de pesquisa.
2. Entrega dos formulários de caracterização da empresa e dos participantes (vide Apêndice 3) e solicitação de preenchimento do formulário de forma comentada: o participante lê a pergunta e, se desejar, expressa dúvidas e comentários sobre a mesma, além de responder à pergunta.
3. Apresentação das instruções para o preenchimento do instrumento de coleta de dados acerca dos atributos percebidos como influentes na tomada de decisão sobre práticas e tecnologias de apoio ao compartilhamento e registro de informação de ideias e características de software. Nas instruções, solicita-se ao participante que se imagine num cenário hipotético de tomada de decisão, como ilustrado na Figura 24.
4. Execução de entrevista semi-estruturada com o apoio do instrumento de coleta, questionando o participante se os itens indicados no formulário o influenciariam na tomada de decisão, de acordo com o cenário hipotético. No caso de o item ser considerado como influente pelo participante, ele indica o estado corrente do item na Startup de software que trabalha. Nessa etapa, o pesquisador deve observar se o participante desconhece ou tem dificuldade de entender os termos apresentados, assim como a relação deles com o cenário de tomada de decisão.
5. Questionamento ao participante sobre a existência de outros elementos, ou tipos de elementos, que o influenciariam na tomada de decisão.

**Cenário:**

Percebeu-se, no contexto da sua empresa Startup, a necessidade melhorar ou adotar novas práticas de compartilhamento e/ou registro de conhecimento sobre ideias e características relacionadas ao software desenvolvido na Startup, para apoiar a equipe de desenvolvimento.

A seguir, pensando na sua tomada de decisão:

- Indique se você considera importante observar as características do contexto, problemas e objetivos listados.
- Indique a situação do contexto atual da sua Startup quanto a essas características do contexto, problemas e objetivos listados.
- Indique quais tecnologias você consideraria como candidatas na sua tomada de decisão.

Figura 24. Cenário do Estudo

### 6.3.1.3 Suposições

Usando os elementos do modelo de contexto (os atributos influentes, problemas e objetivos), o conjunto de valores capturados pelos estudos executados e as escalas definidas, os participantes conseguirão informar os elementos que influenciam a tomada de decisão na empresa sobre práticas e tecnologias no tema da pesquisa, além de conseguirem indicar suas percepções sobre o estado corrente desses elementos na Startup.

### 6.3.1.4 Procedimento de Análise

Os dados coletados pelos instrumentos e pela observação do pesquisador serão analisados quanto à eficácia dos itens do modelo de contexto em apoiar o participante a expor os elementos do seu contexto Startup de software que o mesmo percebe como influentes na tomada de decisão.

### 6.3.1.5 Participantes

O Quadro 32 a seguir apresenta uma visão geral das Startups de software participantes, selecionadas para o estudo de avaliação por conveniência, a partir de contatos profissionais da autora desta tese.

As Startups de software participantes do estudo apresentam variedade em algumas características, como os domínios de problema das soluções de software que desenvolvem e o tempo de atuação, tendo a mais antiga sete anos e três meses na época da entrevista, enquanto que a mais nova tem oito meses. Todas já lançaram ao menos dois protótipos ou versões do produto que atualmente desenvolvem com o

objetivo de inovação. Quanto ao porte das empresas, este varia entre micro e pequeno segundo a classificação definida pelo SEBRAE (SEBRAE-DIEESE, 2016).

Quanto às semelhanças, uma delas é a metodologia de desenvolvimento. As Startups de software utilizam práticas ágeis, ou adaptações de práticas ágeis, com exceção da Startup 2, que declarou usar metodologia *ad hoc*. Outra semelhança que pode ser citada é que todas as participantes possuem algum apoio externo, sendo que todas recebem ou já receberam apoio de infraestrutura e mentoria através de incubadoras de empresas e/ou parques tecnológicos.

O Quadro 33 apresenta a percepção dos entrevistados acerca do estágio do projeto de desenvolvimento do software para inovação nas organizações em que trabalham. É importante notar que, na época de execução do estudo, três das cinco organizações participantes estavam em estágio de mudanças de práticas de gerenciamento e de desenvolvimento.

As Startups 1 e 4 apresentam uma situação especial para os objetivos deste estudo pois, nestas organizações, as melhorias implantadas, ou implantação, envolvem tecnologias e práticas de registro das características de software. Por exemplo, na Startup 1 a equipe passou a utilizar um documento híbrido de especificação, contendo a descrição dos requisitos, os atores, diagramas de projeto da UML, especialmente os que representam aspectos temporais (o fluxo, segundo o entrevistado) e protótipos de tela. Por sua vez, a Startup 4 está analisando opções para melhoria das tecnologias e práticas já adotadas.

Quadro 32. Dados das Startups Participantes do Estudo de Avaliação

	Startup 1	Startup 2	Startup 3	Startup 4	Startup 5
<b>Fundação da Startup:</b>	2015	11/2011	10/2017	7/2016	6/2018
<b>Porte</b>	Pequena (10 a 49 funcionários)	Micro (Até 9 funcionários)	Pequena (10 a 49 funcionários)	Micro (Até 9 funcionários)	Micro (Até 9 funcionários)
<b>Quantidade de projetos de software com objetivo de inovação desenvolvidos</b>	Mais de 10	2	3	2	1
<b>Domínio(s) de aplicação dos produtos de software com objetivo de inovação que a empresa atualmente desenvolve</b>	Saúde, Urbanismo, Limpeza Urbana, Restaurantes, Gestão de Obras, Gestão de Equipe	Integração e Análise de dados	Apoio à motoristas e donos de carro em geral	Monitoramento e otimização de processos - Geolocalização interna e Visão computacional Melhorias na segurança.	Gestão do Fluxo de pagamento de contas médicas (cooperativas e prestadoras de serviços médicos como consultas, anestésias, plantões, etc.)
<b>Quantidade de protótipos/versões lançados</b>	Mais de 5	4	21	2	3
<b>Metodologia/prática de desenvolvimento de software</b>	Práticas de processo Ágil - Scrum / Kanban	Ad hoc	Ad hoc, com algumas práticas Ágeis	Scrum / Kanban	SCRUM, no sentido do uso de <i>sprints</i> para organizar as tarefas em algumas situações, pois existem tarefas com maior prioridade que seguem uma linha de FIFO
<b>Apoio externo e/ou parcerias?</b>	Participa de programa em incubadora de empresas. Possui parcerias com órgãos públicos para avaliação da solução.	Participou de programa em incubadora de empresas. Atualmente está em um Parque Tecnológico universitário. Recebeu fomento de organização pública e possui parceria com especialistas.	Participa de programa em incubadora de empresas.	Participa de programa em incubadora de empresas.	Participa de programa em parque Tecnológico universitário.

Quadro 33. Estágio do Projeto de Desenvolvimento para Inovação

	Startup 1	Startup 2	Startup 3	Startup 4	Startup 5
<b>Refinando a ideia de negócio, formando a equipe e iniciando o projeto.</b>					
<b>Prototipando ou construindo a primeira versão do produto.</b>			X	X	
<b>Avaliando a aceitação do produto após as primeiras vendas/adesões das primeiras versões.</b>		X	X	X	X
<b>Adaptando o produto para as necessidades dos clientes que já aderiram ao produto.</b>	X	X			X
<b>Investindo ativamente no aumento de usuários</b>	X	X			
<b>Adicionando produtos complementares, expandindo o produto ou diversificando o portfólio para entrar em novos mercados/contextos</b>	X			X	X
<b>Dando suporte aos usuários atuais e sem tempo ou recursos para crescer ou expandir de forma intensa</b>		X			
<b>Devido ao aumento da equipe e/ou da complexidade do produto, estamos melhorando ou adotando práticas de gerenciamento e desenvolvimento</b>	X			X	X
<b>Outros:</b>					

Quanto aos representantes das Startups de software que foram entrevistados (vide Quadro 34), todos são envolvidos com atividades de gestão da equipe de desenvolvimento, sendo envolvidos nas tomadas de decisão das tecnologias e práticas adotadas. Além disso, eles desempenham atividades de engenharia. É importante notar que em relação à formação acadêmica, todos os entrevistados têm pelo menos mestrado na área de sistemas e computação. Se por um lado isto pode ter facilitado o entendimento dos conceitos e objetivos da pesquisa durante a entrevista semi-estruturada, por outro lado isto pode representar uma limitação do estudo quanto à representatividade dos entrevistados em relação à população de gerentes/líderes de equipe em Startups de software.

Quadro 34. Caracterização dos Respondentes

	Startup 1	Startup 2	Startup 3	Startup 4	Startup 5
<b>Nº de projetos que participou na Startup</b>	Mais de 10	2	1	6	3
<b>Maior grau de formação</b>	Mestrado em Engenharia de Sistemas e Computação	Mestrado em Engenharia de Sistemas e Computação	Doutorado em Engenharia de Sistemas e Computação	Mestrado em Engenharia de Sistemas e Computação	Mestrado em Ciência da Computação
<b>Atividades que desempenha no projeto de software para inovação</b>	Responsável pela arquitetura do software, processo de desenvolvimento e melhoria do desempenho da equipe técnica.	<p>Pesquisa de soluções existentes, disponíveis e estáveis.</p> <p>Pesquisa de artigos científicos na área.</p> <p>Especificação dos requisitos e Codificação.</p> <p>Distribuição de tarefas e elaboração de relatório de atividades realizadas pela equipe.</p>	<p>Gestão e contato com cliente.</p> <p>Analista, desenvolvedor e testador.</p>	Desenvolvimento, Administração, Gerenciamento de Equipes, Empreendedor.	<p>Coleta Inicial de Requisitos e Prototipação.</p> <p>Arquitetura e Programação. Gerência de Projetos</p>

### 6.3.2 Resultados

Seguindo os procedimentos planejados para a execução das entrevistas semiestruturadas, os respondentes indicaram se cada um dos itens identificados por esta pesquisa de tese como influentes na tomada de decisão sobre o tema de pesquisa, organizados conforme o modelo definido para contexto de Startup de software – Moderadores, Problemas e Objetivos de Melhoria – lhes influenciariam na tomada de decisão sobre a adoção de tecnologias e práticas para compartilhamento e registro de informação acerca de ideias e características do software. Em seguida, os respondentes caracterizaram, quanto aos mesmos elementos, a situação corrente das Startups de software que trabalham. As próximas subseções detalham os resultados.

Em relação aos materiais, primeiramente houve evolução no questionário elaborado quanto a escrita de cada item, para que ficasse mais claro ao entrevistado como deveria ser usada a escala de cada elemento para indicar o contexto atual da Startup de software. Por exemplo, o item “União da Equipe” passou a ser “Nível de União

da Equipe” e, conforme a escala definida, pode ter um valor entre zero (0-Inexistente) e dez (10 – Elevado). Após a execução da terceira entrevista, a ferramenta *Limesurvey*<sup>5</sup> foi utilizada para armazenar, disponibilizar e coletar as respostas do questionário, que precisou ser adaptado para a ferramenta.

### 6.3.2.1 Percepção sobre as Características Influentes

A Figura 25 mostra a frequência com que os respondentes citaram se as Características Influentes os influenciariam, sendo possível a eles responder como Sim, Não, ou Não sei dizer.

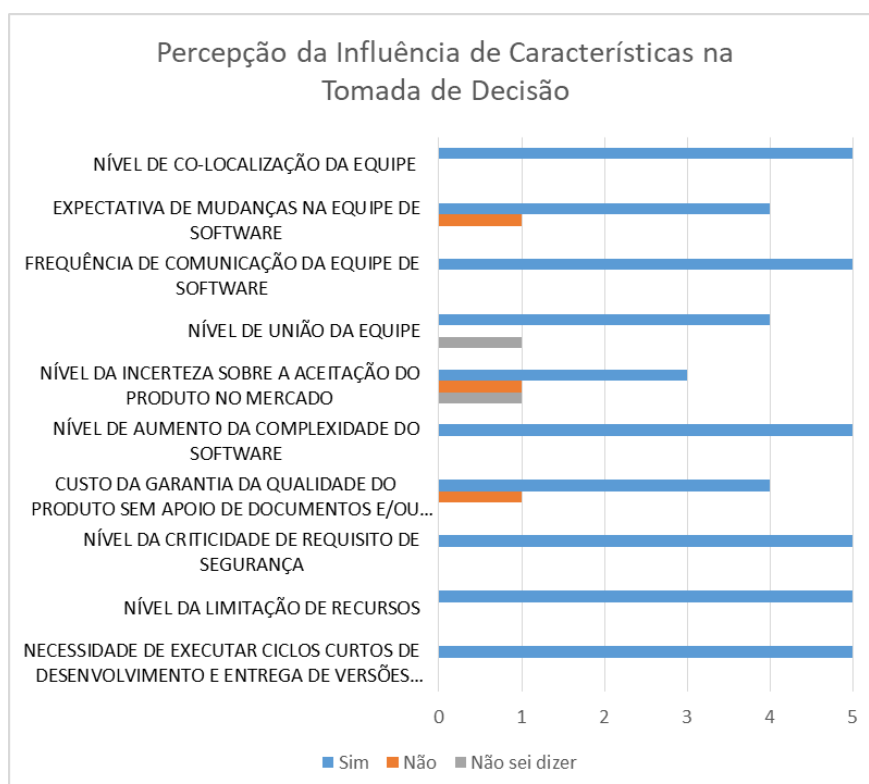


Figura 25. Percepção da Influência de Características na Tomada de Decisão

O respondente da Startup 5 não considerou que a característica “Expectativa de Mudança de Software” influenciaria sua decisão. Além disso, ele afirmou não saber dizer se o ‘Nível de União da Equipe’ teria influência na tomada de decisão. É possível que esta percepção ocorra devido às outras características da Startup de software, como por exemplo o próprio tempo de atuação da Startup 5, o fato de não ter passado por mudanças significativas na equipe e por não ter uma expectativa alta de mudanças na equipe (vide Figura 26).

<sup>5</sup> <https://www.limesurvey.org>

A característica “Nível de Incerteza sobre a aceitação do produto no mercado” foi considerada como não influente pelo respondente da Startup 2, enquanto que o respondente da Startup 5 afirmou não saber dizer se a característica lhe influenciaria. No caso da Startup 2, o domínio de aplicação da solução que desenvolve (Integração de Análise de Dados) é considerado promissor e, como mostra a Figura 26, a mesma tem a percepção de que a incerteza de aceitação no mercado não é alta.

Por fim, a característica “Custo da Garantia da Qualidade do Produto sem Apoio de Documentos e/ou Modelos do Software” também não foi considerada como influente pelo respondente da Startup 2, podendo esta percepção estar também relacionada ao domínio de problema que a organização trabalha.

A Figura 26 mostra a percepção dos respondentes quanto aos Atributos Influentes. Nenhum respondente relatou nem demonstrou dificuldades em utilizar a escala definida para indicar qual a situação corrente dos elementos definidos no modelo na Startup de software em que trabalham.

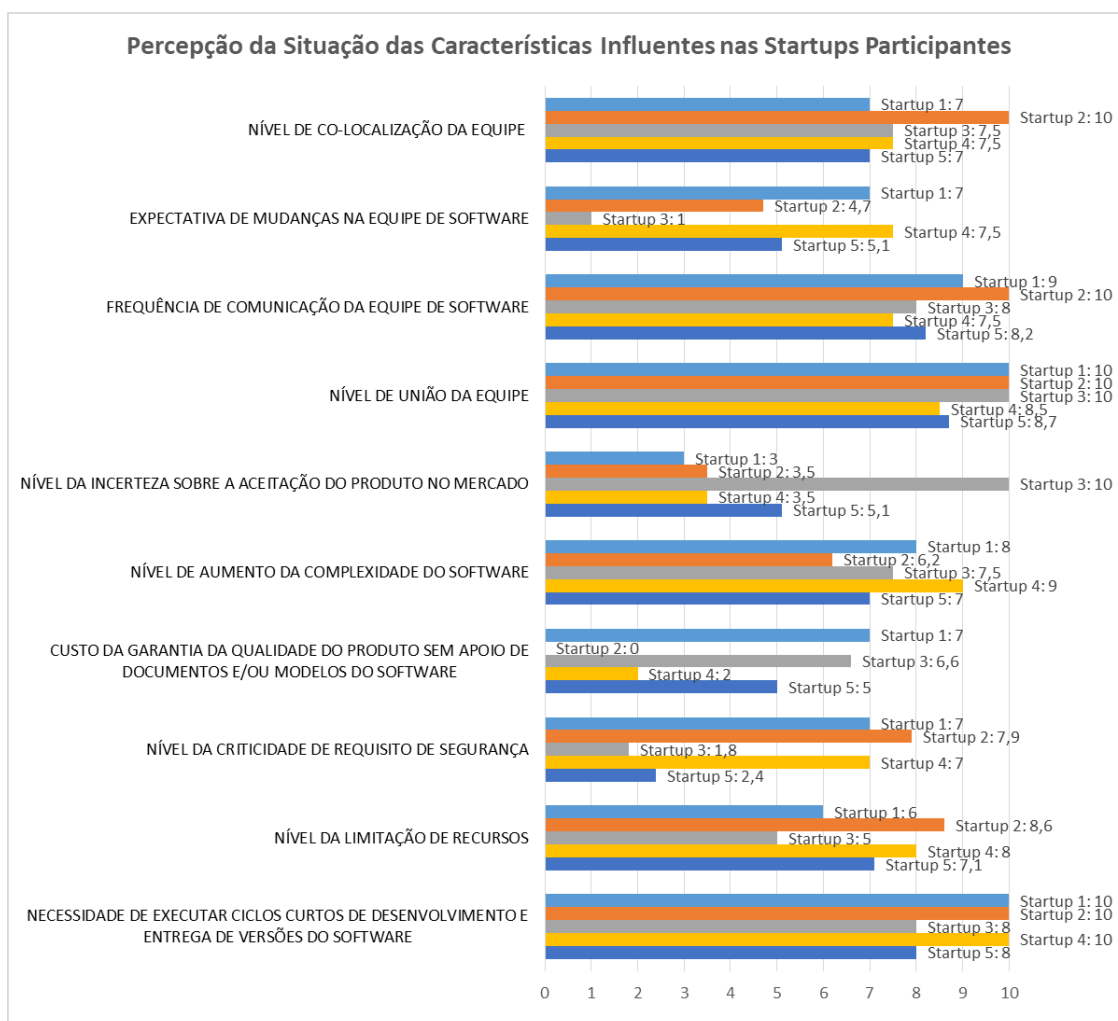


Figura 26. Percepção da Situação das Características Influentes nas Startups Participantes



### 6.3.2.2 Percepção sobre os Problemas Relacionados com Compartilhamento e Recuperação de Informação de Software

Como mostra a Figura 27, ao indicarem se os problemas extraídos a partir do mapeamento sistemático da literatura influenciariam a tomada de decisão nas organizações que trabalham, apenas o problema de “Falta de Sintaxe Comum entre as Pessoas da Equipe ou Ocorrência de Diferentes Interpretações para Termos em Comum” foi considerado como não influente por um respondente, no caso, o representante da Startup 2. É possível supor que esta percepção esteja atrelada ao fato de que na equipe de desenvolvimento da Startup não se percebe a ocorrência deste problema, como mostra a Figura 28.

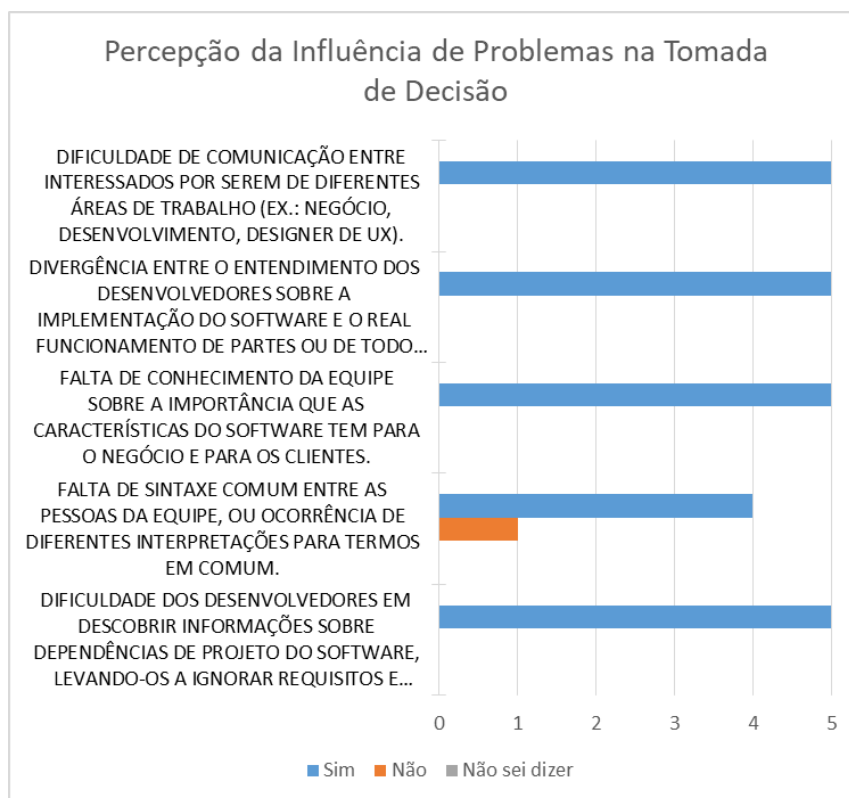


Figura 27. Percepção da Influência de Problemas na Tomada de Decisão

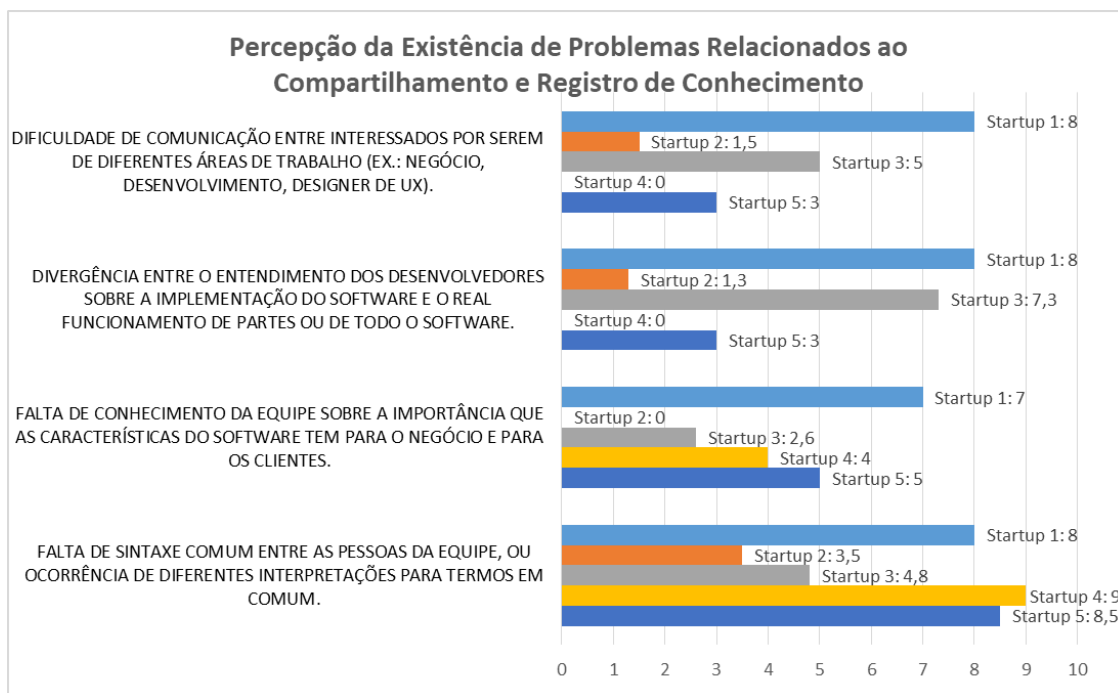


Figura 28. Percepção da Existência de Problemas Relacionados ao Compartilhamento e Registro de Informação

### 6.3.2.3 Percepção sobre os Objetivos de Melhoria

Dentre os objetivos de melhoria apresentados, apenas o de “Alcançar Entendimento Comum sobre o Software pela Equipe de Desenvolvimento” não foi considerado como influente (vide Figura 29), sendo essa a resposta do representante da Startup 4. Ao caracterizar a prioridade corrente dos objetivos de melhoria na organização, o mesmo respondente indicou que para tal objetivo a prioridade é inexistente (0-zero), como pode ser visto na Figura 30.

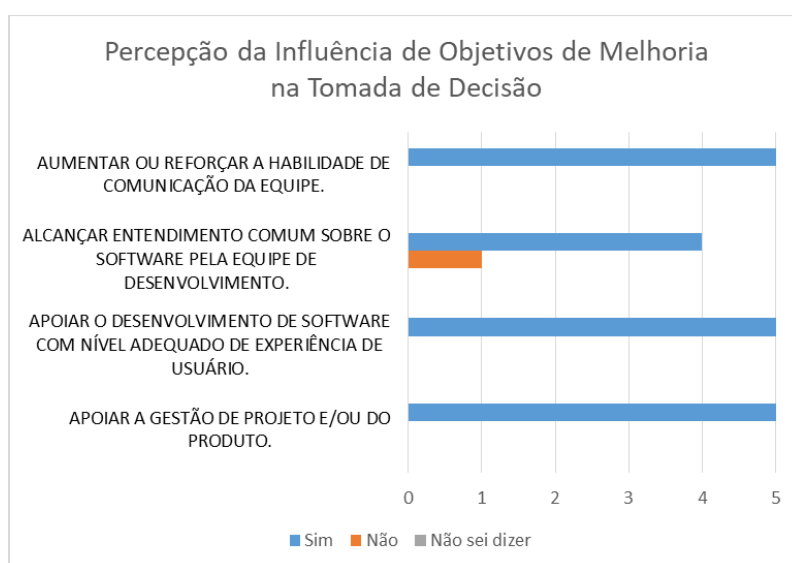


Figura 29. Percepção da Influência de Objetivos de Melhoria na Tomada de Decisão

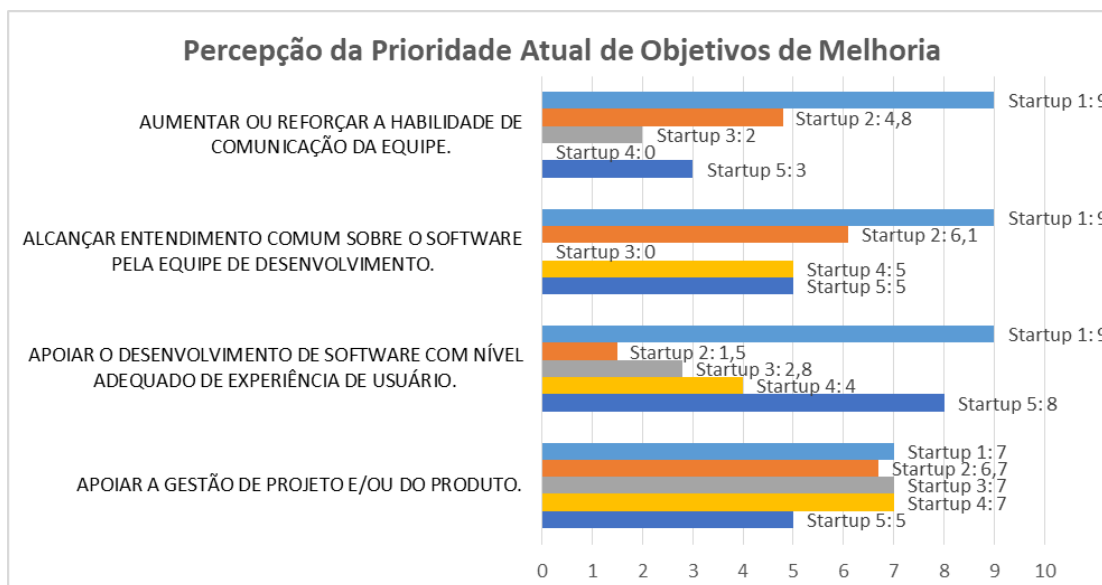


Figura 30. Percepção da Prioridade Atual de Objetivos de Melhoria

É importante frisar que os valores atribuídos pelos respondentes para a prioridade corrente dos objetivos de melhoria não expressam o quanto eles consideram os objetivos como relevantes. Por exemplo, o respondente da Startup 2 mencionou que a limitação de recursos impede que alguns dos objetivos citados tenham maior prioridade na Startup, apesar de serem desejáveis de serem alcançados na organização.

### 6.3.3 Discussão dos resultados da avaliação.

Durante a execução de entrevista semi-estruturada, os respondentes indicaram suas percepções quanto à influência de cada elemento apresentado na tomada de decisão no escopo de problema do estudo. Em seguida, os respondentes caracterizam a situação corrente de suas Startups quanto aos mesmos elementos, utilizando a escala definida.

Mesmo o estudo sendo executado com poucos representantes de organizações Startups, considera-se que se obteve indícios da eficácia do modelo conceitual elaborado, seus elementos e escalas definidas. Durante esta pesquisa, adotou-se uma perspectiva de reconhecimento do contexto das Startups para a tomada de decisão, ao contrário de uma visão consensual. Assim, os elementos que não foram reconhecidos por um participante do estudo continuam sendo parte do modelo, visto que foram considerados influentes por outros participantes.

Além dos resultados reportados neste documento, os participantes fizeram comentários que também representam indícios da utilidade prática do modelo conceitual definido. Quando ao final da entrevista a pesquisadora perguntou se os respondentes

poderiam citar outros elementos que influenciariam a tomada de decisão, considerando o cenário proposto no estudo, apenas o respondente da Startup 2 indicou novos elementos. O representante da Startup 1, respondeu que não teria o que acrescentar e pediu que fosse informado sobre onde o conjunto de contextos e práticas estaria disponível para consulta.

Já o respondente da Startup 3 afirmou que ele não tinha o que acrescentar e que ele tinha aprendido durante a entrevista. Segundo o mesmo, durante a entrevista em alguns momentos ele não reconheceu algumas características, problemas ou objetivos de melhoria como influentes na tomada de decisão, porém, por saber que as informações foram extraídas de outros contextos de organizações Startup, o mesmo se perguntava qual seria a situação que fez o elemento ser influente para a Startup, o que lhe fez lembrar de situações passadas na sua organização.

O respondente da Startup 4 afirmou que as mudanças de práticas na Startup que gerencia são, na maioria das vezes, decorrentes de problemas percebidos e que a participação no estudo o ajudou a refletir sobre qual é a situação corrente da empresa. Além disso, o participante sugeriu que o modelo conceitual proposto evolua para apoiar Startups a observarem diferenças nos seus contextos a partir de análise temporal.

Por sua vez, o respondente da Startup 5 sugeriu que a pesquisadora crie um produto a partir da proposta de tese e busque parcerias para formar um corpo de conhecimento a partir de informações de contextos adicionais de Startups de software e utilizar inteligência computacional para comparação de contextos.

Os comentários e sugestões dos participantes do estudo podem ser considerados oportunidades de pesquisa e de desenvolvimento, a partir da solução desenvolvida na tese, a serem consideradas em trabalhos futuros.

Durante o preenchimento do instrumento, o participante da Startup 1 pediu explicação sobre o significado dos seguintes itens: Nível de Co-localização da Equipe, Nível de Aumento da Complexidade do Software e Nível da Criticidade de Requisitos de Segurança. O participante da Startup 2 também pediu explicação sobre o significado do item Custo da Garantia da Qualidade do Produto sem Apoio de Documentos e/ou Modelos do Software. As dúvidas foram esclarecidas pela pesquisadora no momento da entrevista, porém percebeu-se a necessidade de disponibilizar, aos usuários do modelo conceitual que desejem caracterizar uma Startup, a descrição de cada atributo do modelo e também exemplificar o procedimento de análise dos elementos.

O estudo possui as seguintes ameaças à validade:

- As Startups participantes foram selecionados por conveniência, a partir de contatos da pesquisadora com profissionais da Incubadora de Empresas e do Parque Tecnológico da UFRJ. Dados o ecossistema de

inovação acadêmico que participam e a formação acadêmica dos indivíduos participantes, é possível que eles representem um grupo específico de Startups cujas características podem influenciar tanto no entendimento dos conceitos do estudo, quanto no reconhecimento da pertinência dos elementos apresentados como influentes na tomada de decisão. Como trabalho futuro, novos estudos deverão ser conduzidos para diminuir essa ameaça de validade.

- A quantidade de participantes por Startup é pequena. Porém, dado o papel de liderança que eles executam na empresa e considerando o objetivo do estudo, o fato dos elementos apresentados terem sido reconhecidos pelos participantes como influentes na tomada de decisão no contexto das Startups que os participantes fazem parte contribui para indícios iniciais da validade da proposta.
- É possível que os respondentes dos questionários tenham sido induzidos em suas repostas por causa do instrumento e da forma que os itens e perguntas são apresentados. Para diminuir essa ameaça, o instrumento foi respondido pelos participantes enquanto eram entrevistados e forneciam explicações para suas escolhas de repostas.

## 6.4 TRABALHOS RELACIONADOS.

Durante a execução do mapeamento sistemático executado em 2016 (capítulo 3), não foram encontradas propostas relacionadas ao registro e compartilhamento de informação de software com observação das características de Startups de software, incluindo os seus estágios de evolução. Durante o decorrer do desenvolvimento desta tese, alguns novos trabalhos surgiram, grande parte visando entender o contexto de Startups, como por exemplo, como a dívida técnica ocorre em Startups (KLOTINS, UNTERKALMSTEINER, *et al.*, 2018).

Práticas para Engenharia de Requisitos adotadas em Startups de software, incluindo para especificação de requisitos, foram analisadas por Tripathi et al. (TRIPATHI, 2018), tendo resultados semelhantes aos encontrados no estudo exploratório descrito no capítulo 4 desta tese e em (NASCIMENTO e TRAVASSOS, 2017): uso de quadro-branco (físicos ou digitais), *post-its*, ferramentas do tipo *ticket-based*, uso de listas e desenhos informais para registro e compartilhamento de informações sobre os requisitos, com falta de processos pré-definidos para especificação de requisitos. Os autores reportam como resultado a observação de diferentes níveis de documentação de requisitos, variando entre os níveis de produto (objetivo), de características, de função e de componentes, com detalhamento de

solução. Porém, diferente desta tese, os fatores de contexto que influenciam a seleção das práticas e os estágios da Startup não são levados em consideração na pesquisa.

Nesta tese, tomou-se como requisito de elaboração do modelo de contexto de Startups de software considerar as características de Startups, incluindo a de mudanças de objetivos, desafios e práticas devido a evolução da Startup de sua formação até ser um negócio inovador que escalou no mercado. Além disso, um conjunto de fatores e diversidades de características e possíveis influências da Startup de software levam a essa mudança de práticas. Nesse sentido, a proposta de Gralha e Damian et al. (GRALHA, DAMIAN, *et al.*, 2018) e a de Klotins, Unterkalmsteiner e Gorschek (KLOTINS, UNTERKALMSTEINER e GORSCHKEK, 2018) possuem uma perspectiva de solução mais próxima da seguida por esta tese, sendo detalhadas a seguir.

A partir da observação da evolução de dezesseis Startups de software, Gralha e Damian et al. (GRALHA, DAMIAN, *et al.*, 2018) propõem em uma teoria acerca da evolução de práticas para requisitos em Startups de software, que define estágios de evolução de práticas de requisitos para as seguintes áreas: artefatos, gestão do conhecimento, dívida técnica, definição de papéis na equipe, planejamento, sendo as três primeiras áreas relacionadas com o registro de informação de software, tema desta tese de doutorado.

De acordo com a teoria, os artefatos de requisitos de software são inicialmente orientados à implementação. O aumento na quantidade de clientes, no número empregados da Startup e no número de empregados trabalhando remotamente impulsionam a evolução das práticas relacionadas à construção de artefatos para o estágio orientado ao usuário. A evolução para o próximo estágio, no qual os artefatos de software passam a conter mais informações e descrições rastreáveis entre artefatos, é motivada pelo aumento da quantidade de demandas de clientes e da quantidade de funcionalidades do software.

De forma semelhante, os resultados de Gralha e Damian et al. (GRALHA, DAMIAN, *et al.*, 2018) indicam que a evolução das práticas de gestão de conhecimento em Startups de software do estágio Informal e Não-estruturado para Informal e semiestruturado ocorre devido ao aumento no número empregados da Startup e no número de empregados trabalhando remotamente. O estágio seguinte de evolução é o Estruturado, desejado pela Startup quando a quantidade de demandas de clientes e a quantidade de funcionalidades do software aumentam.

Em relação à dívida técnica de requisitos, o aumento no número empregados e na quantidade de demandas de clientes levam a Startup a avançar do estágio de dívida técnica Conhecida e Aceita para Rastreada e Registrada. O aumento da taxa de

retenção de clientes e também de retorno de percepção negativa estimulam a evolução para o estágio Gerenciado e Controlado de dívida técnica de requisitos.

Apesar de que a proposta de Gralha e Damian et al. (GRALHA, DAMIAN, *et al.*, 2018) foca em requisitos, enquanto que a tese aqui descrita foca em características e ideias de software (que inclui requisitos e demais características, como por exemplo a arquitetura do software), os chamados pontos de flexão (*turning points*) supracitados são semelhantes às características influentes de colocalização da equipe, expectativa de mudanças na equipe, tamanho da equipe, complexidade do produto e custo de qualidade (vide Quadro). A semelhança é compreensível dado o período de execução e contexto do problema de pesquisa investigado serem semelhantes, sendo a publicação (NASCIMENTO e TRAVASSOS, 2017), sobre os resultados do estudo exploratório desta tese, referenciada em (GRALHA, DAMIAN, *et al.*, 2018).

Por outro lado, enquanto a proposta de Gralha e Damian et al. propõe uma teoria de evolução de práticas específicas para requisitos de software com base em seus pontos de flexão, e assim apoiar mudanças de práticas em Startups de software, nesta tese propõe-se que se observe o contexto de Startup (para o fenômeno de seleção de práticas de registro de informação) considerando problemas e objetivos de melhoria, além de atributos descritivos, de forma a apoiar a comparação de contextos entre Startups de software com objetivo de transferência de tecnologia. Apesar da diferença de propostas de apoio, é possível que as duas propostas se complementem para apoiar a recomendação de práticas e tecnologias para Startups, sendo necessária a execução de estudos observar a complementaridade.

Outro trabalho relacionado é proposto por Klotins, Unterkalmsteiner e Gorschek (KLOTINS, UNTERKALMSTEINER e GORSCHKEK, 2018), que elaboraram um Mapa de Contexto de Startup para descrever os fatores que influenciam a engenharia de produtos intensivos em software em Startups de software. Os fatores estão organizados em uma estrutura que inicia com as categorias Práticas, Objetivos e Ambiente. Cada categoria é refinada em níveis até chegar nos fatores, cuja apresentação contém o nome, descrição, contendo a definição do fator, informação de que medidas podem ser usadas para avaliar o fator, com quais fatores do modelo se relacionam, referências da literatura e descrições de experiências em Startups que citam o fator. A proposta é descrita como um trabalho em andamento, tendo como um dos seus objetivos apoiar a transferência de tecnologia entre Startups pela comparação de contextos. Diferente desta tese, a proposta organiza um corpo de conhecimento sobre fatores que influenciam diversas áreas de engenharia de software, porém, até a data de finalização da presente tese, não apresenta mecanismos que apoiem uma Startup a comparar os fatores do seu contexto, devendo ler e analisar a descrição de cada fator quanto à pertinência ao

escopo do problema de tomada de decisão que deseja. Além disso, a utilização da proposta de Klotins, Unterkalmsteiner e Gorschek (KLOTINS, UNTERKALMSTEINER e GORSCHKEK, 2018) segue uma abordagem binária (o fator está ou não presente no contexto), o que não condiz com os resultados de pesquisa desta tese no tocante à avaliação de fatores moderadores da tomada de decisão acerca de práticas e tecnologias de registro e compartilhamento de informações de software em Startups de software.

## **6.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO**

O presente capítulo apresentou o modelo conceitual de contexto de Startup de software definido para representar os elementos influentes na tomada de decisão acerca da adoção de práticas e tecnologias de registro e compartilhamento de informação de software para a equipe de desenvolvimento de software.

Analisando os trabalhos relacionados (seção 6.4), é possível notar que um diferencial da tese elaborada é o direcionamento de captura da perspectiva da Startup de software quanto aos seus atributos e quanto aos elementos que ela leva em consideração para a tomada de decisão, no escopo do tema de pesquisa. A estruturação de tais atributos em um modelo de contexto específico para o fenômeno pesquisado auxilia não somente na organização dos elementos envolvidos, mas também no apoio à comparação de contextos entre Startups de software.

O objetivo principal do estudo aqui descrito foi a observação da efetividade dos elementos definidos no modelo conceitual desenvolvido na tese de doutorado quanto à representação dos elementos influentes na tomada de decisão acerca de tecnologias e práticas de compartilhamento e registro de informação de ideias e características de software, no contexto de Startups. O estudo contou com a participação de cinco representantes de Startups, as quais possuem diferenças principalmente em relação ao tempo de atuação e porte, mas tem em comum o fato de obterem apoio de incubadoras e parques tecnológicos universitários.

Apesar das suas limitações (discutidas na seção 6.3.3), os resultados do estudo indicam que os elementos são pertinentes, sendo a estruturação em atributos influentes moderadores, problemas e objetivos de melhoria, entendida pelos participantes.

Nas considerações finais desta tese (capítulo 7 a seguir), são apresentadas oportunidades de evolução identificadas e demais trabalhos futuros.



# 7 CONSIDERAÇÕES FINAIS

A partir de um problema observado de forma empírica em um caso real da indústria, nesta tese foram investigados os desafios relacionados ao registro de ideias e características de software no contexto de desenvolvimento de software por Startups. Como discutido no capítulo 2 desta tese, principalmente nos estágios iniciais da Startup de software, a necessidade de diminuir as incertezas tem grande influência nas atividades e nas práticas de desenvolvimento de software, além dos demais objetivos da organização. Tais objetivos e o próprio contexto de desenvolvimento mudam durante a evolução da Startup, muito pelo fato do desenvolvimento do software estar atrelado ao desenvolvimento do modelo de negócio da Startup (BLANK, 2013b). A partir desse entendimento do contexto de Startups de software, a perspectiva de pesquisa desta tese passou a ser sobre os desafios de decidir que práticas e tecnologias adotar para compartilhar e/ou registrar informação sobre as ideias e as características do software desenvolvido para inovação em Startups de software.

Na próxima seção, para cada pergunta de pesquisa da tese são discutidos os resultados finais da tese e oportunidades de evolução.

## 7.1 RESULTADOS E OPORTUNIDADES DE EVOLUÇÃO

**P1: Que fatores influenciam as organizações Startups de software na adoção de práticas e tecnologias para compartilhamento e registro de informações sobre as ideias e as características do software que elas desenvolvem?**

Foram identificados um conjunto de elementos influentes na tomada de decisão sobre o tema da pergunta de pesquisa. O estudo primário executado (capítulo 4) para caracterizar Startups de software quanto às suas práticas de registro contribuiu com resultados que delinearão a proposição do modelo de contexto de Startup de software, pois permitiu observar fatores que influenciam as organizações participantes na decisão das práticas adotadas para registro de ideias e características de software. O estudo secundário executado (capítulo 5), com foco no mapeamento de fatores influentes na tomada de decisão a partir de relatos na literatura sobre as práticas de empresas Startups, contribuiu para a identificação e organização de tipos de atributos de contexto Startup de software que são de interesse para o problema de pesquisa dessa tese.

Foram também identificadas oportunidades de pesquisa futuras sobre os fatores de influência. Investigar quais relações existem entre fatores, se existem prioridades diferentes e como tais relações e prioridades impactam na tomada de decisão pode ser uma oportunidade de aprimorar o modelo de contexto definido. Por exemplo, é possível

que haja correlação entre os moderadores Nível de Co-localização da Equipe e o Nível de União.

Além disso, a definição de cada fator é influenciada pelas limitações inerentes aos estudos, como por exemplo, a percepção dos entrevistados e as decisões subjetivas do pesquisador ao executar análise temática de relatos da literatura técnica. Um impacto disso pode ser a granularidade ou agrupamento das propriedades de contexto representadas por cada fator. Por exemplo, o fator moderador que diz respeito à complexidade do software desenvolvido pela Startup pode estar representando um conjunto de características do código-fonte, da arquitetura, dos requisitos, dentre outros elementos do produto de software ou que impactam nas decisões sobre ele. Investigar quais os níveis de granularidade relevantes para o modelo de contexto definido para representar os fatores é uma oportunidade de evolução.

**P2: Como devem ser organizados os fatores de influência da adoção de práticas e tecnologias para compartilhamento e registro de informação sobre as ideias e as características do software, de forma a apoiar as Startups de software na tomada de decisão?**

Os fatores de influência identificados foram organizados como atributos de contexto de Startups de software, categorizados em moderadores, problemas e objetivos de melhoria. Com base na observação do estudo primário (capítulo 4) e nos relatos da literatura técnica, foram definidas escalas de medição para os atributos de contexto de forma a permitir capturar a percepção de valor de cada atributo em um determinado contexto de Startup de software.

Para os atributos influentes, o julgamento de valor se dá de forma subjetiva pela análise da percepção da intensidade de sua ocorrência, que pode variar entre inexistente/irrelevante até elevado(a), que indica o valor máximo de ocorrência do atributo. Uma possível oportunidade de pesquisa é a investigação de quais atributos influentes podem ser avaliados de forma objetiva e, assim, evoluir o modelo de contexto definido com escalas que capturem tal julgamento com menos subjetividade.

**P3: Que práticas e tecnologias as Startups de software adotam para compartilhamento e registro de informação sobre as ideias e as características de software?**

Um conjunto de práticas e tecnologias foi mapeado (capítulo 6) para formar um corpo de conhecimento sobre o que as Startups de software adotam para apoiar o desenvolvimento de software, no escopo de pesquisa desta tese.

No corpo de conhecimento organizado há limitações quanto a informações sobre a implantação e execução das práticas e tecnologias, sendo oportunidade de evolução para a estruturação deste corpo de conhecimento. Por exemplo, ainda não se tem informação de quem executa, quando executa, usando qual nível de formalidade e detalhamento de informação da característica do software que é registrada e/ou compartilhada com o uso da prática ou tecnologia.

## **7.2 DISCUSSÃO FINAL**

Como já discutido no capítulo 6 desta tese, mesmo sendo baseado em evidências científicas o modelo conceitual de contexto de Startups de software elaborado nesta tese apresenta ameaças à validade causadas tanto pelas limitações dos estudos e métodos de pesquisa utilizados, do conteúdo dos dados coletados, quanto da capacidade de observação da pesquisadora, principalmente nos estudos secundários, nos quais foram utilizados métodos qualitativos de análise. Por outro lado, a execução de estudo de avaliação com representantes de cinco empresas Startups de software reforçam os indícios de pertinência dos elementos envolvidos no contexto de Startups de software para o problema e perguntas de pesquisa da tese, assim como das escalas de avaliação destes elementos.

Outra limitação vem do fato de que as Startups de software investigadas no estudo exploratório (capítulo 4) e as que participaram do estudo de avaliação (capítulo 6) são brasileiras, exceto por uma Startup participante do estudo exploratório. Por outro lado, o modelo conceitual de contexto de Startup definido nesta tese também é fundamentado pelos resultados de um mapeamento sistemático da literatura (capítulo 5), no qual foi possível identificar trinta e cinco contextos de Startups pertencentes a diferentes países (vide Figura 31) e mais quatro resultados de análise agrupando observações de contextos de Startups, como mostra o Quadro 35.

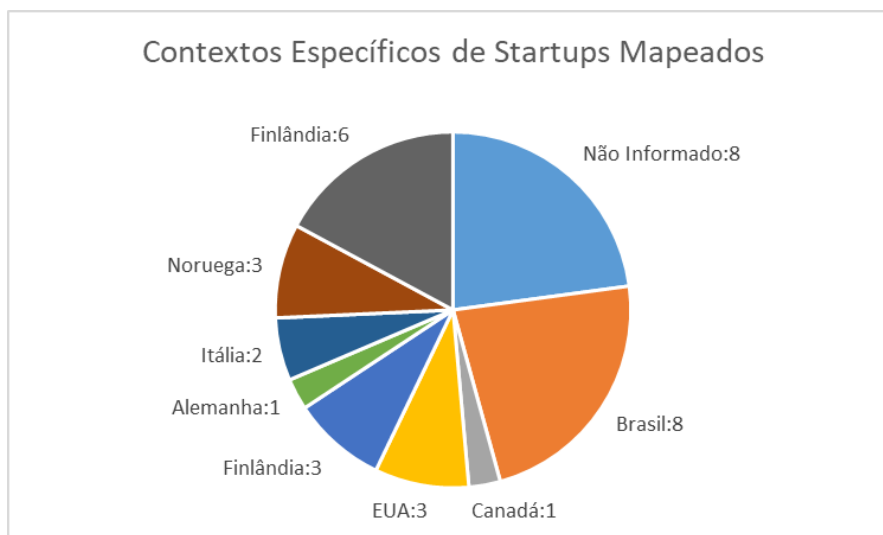


Figura 31. Distribuição de Contextos de Startups Mapeados por País

Quadro 35. Região de Contextos de Startups Reportados de Forma Agrupada

Região	Número de Startups analisadas
Alemanha (2), EUA (4), Itália (4), Suécia (1), Nova Zelândia (1), Reino Unido (1)	13
Brasil (5), Alemanha (1)	6
Brasil	8
Não informada	6

Vislumbra-se que a contribuição desta tese possibilite a elaboração de tecnologias de apoio à Startups de software no problema de decidir que práticas e tecnologias adotar para compartilhamento e registro de informação de software. Como exemplo de trabalhos futuros podem ser citados os desenvolvimentos de tecnologias para:

- Disponibilização de um corpo de conhecimento contendo os contextos mapeados conforme o modelo conceitual.
- Elaboração de metodologia e mecanismo de apoio para Startups de software caracterizarem seus contextos e buscarem por práticas e tecnologias adotadas em contextos semelhantes.
- Mecanismo e metodologia de retroalimentação do corpo de conhecimento.

Apesar do modelo de contexto de Startup de software ter sido elaborado a partir de dados obtidos de relatos da prática em organizações Startups de software (tanto nos

estudos primários quanto nos secundários), é importante notar que estes podem estar fundamentados tanto na experiência quanto em crenças dos envolvidos na tomada de decisão sobre as práticas adotadas. Assim, na perspectiva de contribuição acadêmica, espera-se que as contribuições desta tese auxiliem na elaboração de suposições de pesquisa para estudos futuros. Com a evolução do conhecimento científico sobre as práticas e tecnologias, o modelo conceitual de contexto Startup pode ser evoluído para apresentar evidências das práticas e tecnologias em relação aos seus elementos – os moderadores, os problemas e os objetivos de melhoria.

## 8 REFERÊNCIAS

- AMABILE, T. A model of creativity and innovation in organizations.. **Research in organizational behavior**, v. 10, n. 1, 1998.
- BASILIO, V. R.; CALDIERA, G.; ROMBACH, H. D. Goal question metric paradigm. **Encyclopedia of Software Engineering**, New York, 2001. 528-532.
- BECK, K. **Test-driven development: by example**. [S.l.]: Addison-Wesley Professional., 2013.
- BERG, V. et al. Software startup engineering: A systematic mapping study. **Journal of Systems and Software**, 2018. 255-274.
- BIOLCHINI, J. et al. **Systematic review in software engineering**. System Engineering and Computer Science Department COPPE/UFRJ. Rio de Janeiro, p. 45. 2005. (Technical Report ES 679.05).
- BLANK, S. **The four steps to the epiphany: successful strategies for products that win**. [S.l.]: BookBaby, 2013a.
- BLANK, S. Why the lean start-up changes everything. **Harvard Business Review**, 91, n. 5, 2013b. 63-72.
- BOSCH, J. . O. H. . B. J. . L. J. **The early stage software startup development model: A framework for operationalizing lean principles in software startups**. Lean Enterprise Software and Systems. [S.l.]: [s.n.]. 2013. p. 1-15.
- BOSCH, J. **Building products as innovation experiment systems**. International Conference of Software Business. Berlin, Heidelberg: [s.n.]. 2012. p. 27-39.
- BRAUN, V.; CLARKE, V. Using thematic analysis in psychology. **Qualitative Research in Psychology**, v. 3, n. 2, p. 77-101, 2006.
- BRAUN, V.; CLARKE, V. Thematic Analysis. In: COOPER, H. **APA Handbook of Research Methods in Psychology: Vol. 2. Research Designs**. [S.l.]: [s.n.], v. 2, 2012.
- CROWNE, M. **Why software product startups fail and what to do about it. Evolution of software product development in startup companies**. Engineering Management Conference. [S.l.]: [s.n.]. 2002.
- CRUZES, D.; DYBA, T. **Recommended steps for thematic synthesis in software engineering**. International Symposium on Empirical Software Engineering and Measurement. [S.l.]: [s.n.]. 2012. p. 275-284.
- DIAS NETO, A. et al. Improving Evidence about Software Technologies: A Look at Model-Based Testing. **IEEE Software**, v. 25, n. 3, 2008.
- EDISON, H.; BIN ALI, N.; TORKARA, R. Towards innovation measurement in the software industry. **Journal of Systems and Software**, v. 86, n. 5, p. 1390– 1407, May 2013.
- GIARDINO, C. et al. **Key challenges in early-stage software startups**. International Conference on Agile Software Development. Springer, Cham.: [s.n.]. 2015. p. 52-63.
- GIARDINO, C. et al. Software Development in Startup Companies: The Greenfield Startup Model. **IEEE Transactions on Software Engineering**, v. 42, n. 6, 2015.

- GIARDINO, C.; WANG, X.; ABRAHAMSSON, P. **Why early-stage software startups fail: a behavioral framework.** International Conference of Software Business. [S.I.]: Springer, Cham. 2014. p. 27-41.
- GRALHA, C. et al. **The evolution of requirements practices in software startups.** Proceedings of the 40th International Conference on Software Engineering. [S.I.]: ACM. 2018. p. 823-833.
- ISO. **ISO Guide 73:2009, Risk management — Vocabulary.** International Organization for Standardization. [S.I.]. 2009.
- KHURUM, M.; FRICKER, S.; GORSCHKE, T. The contextual nature of innovation – An empirical investigation. **Information and Software Technology**, v. 57, 2015.
- KLOTINS, E. et al. **Exploration of technical debt in start-ups.** 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP). [S.I.]: IEEE. 2018. p. 75-84.
- KLOTINS, E.; UNTERKALMSTEINER, M.; GORSCHKE, T. Software-intensive product engineering in start-ups: a taxonomy. **IEEE Software**, v. 35, n. 4, p. 44-52, 2018.
- KLOTINS, E.; UNTERKALMSTEINER, M.; GORSCHKE, T. Software engineering in start-up companies: An analysis of 88 experience reports. **Empirical Software Engineering**, v. 24, n. 1, p. 68-102, 2019.
- KON, F. et al. **A panorama of the israeli software startup ecosystem..** [S.I.]. 2014. (Available at SSRN 2441157).
- LEMOS, J. et al. **A systematic mapping study on creativity in requirements engineering.** Proceedings of the 27th Annual ACM Symposium on Applied Computing. [S.I.]: [s.n.]. 2012. p. 1083-1088.
- LIPPOLDT, D.; STRYSZOWSKI, P. **Innovation in the Software Sector.** Paris: OECD Publishing, 2009.
- MAFRA, S. N. . B. R. F. . T. G. H. **Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software.** XX Simpósio Brasileiro de Engenharia de Software (SBES). Florianópolis: [s.n.]. 2006.
- NASCIMENTO, L. M. A.; TRAVASSOS, G. H. **Software Knowledge Registration Practices at Software Innovation Startups: Results of an Exploratory Study.** Proceedings of the 31st Brazilian Symposium on Software Engineering. [S.I.]: ACM. 2017. p. 234-243.
- NGUYEN-DUC, A.; SHAH, S.; AMBRAHAMSSON, P. **Towards an early stage software startups evolution model.** 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). [S.I.]: IEEE. 2016. p. 120-127.
- OECD. **Oslo Manual: Guidelines for Collecting and Interpreting Innovation data.** 3. ed. [S.I.]: [s.n.], 2005.
- PAI, M. et al. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. **The National medical journal of India**, v. 17, n. 2, 2003.
- PATERNOSTER, N. et al. Software development in startup companies: A systematic mapping study. **Information and Software Technology**, v. 56, n. 10, p. 1200-1218, 2014.
- PIKKARAINEN, M. . C. W. . B. N. . H. A. A. **The Art of Software Innovation: Eight Practice Areas to Inspire your Business.** [S.I.]: Springer Publishing Company, 2011.
- REGNELL, B.; BRINKKEMPER, S. Market-Driven Requirements Engineering for Software Products. In: \_\_\_\_\_ **Engineering and Managing Software Requirements.** [S.I.]: [s.n.], 2005.

- RIES, E. **The Lean Startup**: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. [S.l.]: Crown Business, 2011.
- ROBERTSON, S. Requirements trawling: techniques for discovering requirements. **International Journal of Human-Computer Studies**, v. 55, n. 4, 2001.
- ROSE, J.; FURNEAUX, B. Innovation Drivers and Outputs for Software Firms: Literature Review and Concept Development. **Advances in Software Engineering**, 2016.
- SAHA, S. K. et al. **A systematic review on creativity techniques for requirements engineering**. International Conference on Informatics, Electronics & Vision. [S.l.]: [s.n.]. 2012.
- SALDAÑA, J. **The coding manual for qualitative researchers**. [S.l.]: Sage, 2015.
- SAWYER, P.; SOMMERVILLE, I.; KOTONYA, G. **Improving Market-Driven RE Processes**. International Conference on Product-Focused Software Process Improvement (Profes '99). [S.l.]: [s.n.]. 1999.
- SEBRAE-DIEESE. Anuário do trabalho na micro e pequena empresa, 2016. Disponível em:  
<<https://m.sebrae.com.br/Sebrae/Portal%20Sebrae/Anexos/Anu%C3%A1rio%20do%20Trabalho%20nos%20Pequenos%20Neg%C3%B3cios%202016%20VF.pdf>>. Acesso em: 30 Junho 2019.
- SOUZA, R.; MALTA, K.; DE ALMEIDA, E. **Software engineering in startups**: a single embedded case study. 2017 IEEE/ACM 1st International Workshop on Software Engineering for Startups (SoftStart). [S.l.]: IEEE. 2017. p. 17-23.
- STERNBERG, R. **Handbook of creativity**. [S.l.]: Cambridge University Press, 1999.
- SUTCLIFFE, A.; SAWYER, P. **Requirements elicitation**: Towards the unknown unknowns. IEEE International Requirements Engineering Conference. [S.l.]: [s.n.]. 2013.
- SUTTON, S. M. The role of process in software start-up. **IEEE Software**, 17, n. 4, 2000. 33-39.
- TAKEUCHI, H.; NONAKA, I. **Gestão do conhecimento**. Porto Alegre: Bookman, 2008.
- TRIPATHI, N. E. A. An anatomy of requirements engineering in software startups using multi-vocal literature and case survey. **Journal of Systems and Software**, v. 146, p. 130-151, 2018.
- UNTERKALMSTEINER, M. . E. A. Software Startups—A Research Agenda. **e-Informatica Software Engineering Journal**, 10, n. 1, 2016.
- WEWERS, M. E.; LOWE, N. K. A critical review of visual analogue scales in the measurement of clinical phenomena., v. 13, n. 4, p. 227-236, 1990.
- WNUK, K.; RUNESON, P. **Engineering open innovation—towards a framework for fostering open innovation**. International Conference of Software Business. [S.l.]: [s.n.]. 2013. p. 48-59.



# APÊNDICE 1 - ARTIGOS SELECIONADO PARA O MAPEAMENTO SISTEMÁTICO.

Id.	Artigo
[P1]	El-Sharkawy, S., & Schmid, K. (2011, March). A heuristic approach for supporting product innovation in requirements engineering: a controlled experiment. In <i>International Working Conference on Requirements Engineering: Foundation for Software Quality</i> (pp. 78-93). Springer Berlin Heidelberg.
[P2]	Lai, H., Peng, R., Sun, D., & Liu, J. (2012). A lightweight forum-based distributed requirement elicitation process for open source community. <i>arXiv preprint arXiv:1210.3110</i>
[P3]	Gorschek, T., Fricker, S., Palm, K., & Kunsman, S. (2010). A lightweight innovation process for software-intensive product development. <i>IEEE software</i> , 27(1), 37
[P4]	Yang-Turner, F., & Lau, L. (2011, August). A pragmatic strategy for creative requirements elicitation: from current work practice to future work practice. In <i>Requirements Engineering for Systems, Services and Systems-of-Systems (RESS), 2011 Workshop on</i> (pp. 28-31). IEEE
[P5]	Niknafs, A., & Berry, D. M. (2013, July). An industrial case study of the impact of domain ignorance on the effectiveness of requirements idea generation during requirements elicitation. In <i>2013 21st IEEE International Requirements Engineering Conference (RE)</i> (pp. 279-283). IEEE.
[P6]	Triantafyllakos, G., Palaigeorgiou, G., & Tsoukalas, I. A. (2011). Designing educational software with students through collaborative design games: The We! Design&Play framework. <i>Computers &amp; Education</i> , 56(1), 227-242.
[P7]	Sutcliffe, A., Rayson, P., Bull, C. N., & Sawyer, P. (2014, August). Discovering affect-laden requirements to achieve system acceptance. In <i>2014 IEEE 22nd International Requirements Engineering Conference (RE)</i> (pp. 173-182). IEEE.
[P8]	Jayakanthan, R., & Sundararajan, D. (2011, September). Enterprise crowdsourcing solutions for software development and ideation. In <i>Proceedings of the 2nd international workshop on Ubiquitous crowdsourcing</i> (pp. 25-28). ACM.
[P9]	Goto, T., Tsuchida, K., & Nishino, T. (2014, August). EPISODE: An Extreme Programming Method for Innovative Software Based on Systems Design. In <i>Advanced Applied Informatics (IIAIAI), 2014 IIAI 3rd International Conference on</i> (pp. 780-784). IEEE.
[P10]	G., & Fitzgerald, G. (2004). Evolving Self-Organizing Activities - Addressing Innovation and Unpredictable Environments. <i>IFIP TC8/WG8.6 Seventh Working Conference on IT Innovation for Adaptability and Competitiveness</i> , 141, pp. 3-19.
[P11]	Seyff, N., Maiden, N., Karlsen, K., Lockerbie, J., Grünbacher, P., Graf, F., & Ncube, C. (2009). Exploring how to use scenarios to discover requirements. <i>Requirements Engineering</i> , 14(2), 91-111.
[P12]	Hollis, B., & Maiden, N. (2013). Extending agile processes with creativity techniques. <i>IEEE software</i> , 30(5), 78-84.
[P13]	Yang-Turner, F., & Lau, L. (2011, August). Extending use case diagrams to support requirements discovery. In <i>Requirements Engineering for Systems, Services and Systems-of-Systems (RESS), 2011 Workshop on</i> (pp. 32-35). IEEE.
[P14]	Rice, M., & Carmichael, A. (2013). Factors facilitating or impeding older adults' creative contributions in the collaborative design of a novel DTV-based application. <i>Universal access in the information society</i> , 12(1), 5-19.

[P15]	Rodrigues, L. F., Costa, C. J., & Oliveira, A. (2014, May). Gamification: the using of user discussion groups in the software development in e-banking. In <i>Proceedings of the International Conference on Information Systems and Design of Communication</i> (pp. 27-34). ACM.
[P16]	Bragge, J. O. H. A. N. N. A., Merisalo-Rantanen, H. I. L. K. K. A., & Hallikainen, P. E. T. R. I. (2005). Gathering innovative end-user feedback for continuous development of information systems: a repeatable and transferable e-collaboration process. <i>IEEE Transactions on Professional Communication</i> , 48(1), 55-67. p
[P17]	Oswald, D. (2011). Ideation and Design of Novel iPad Apps: A Design Education Case Study. In <i>DS 69: Proceedings of E&amp;PDE 2011, the 13th International Conference on Engineering and Product Design Education, London, UK, 08.-09.09. 2011.</i>
[P18]	Ohshiro, K., Watahiki, K., & Saeki, M. (2005, December). Integrating an idea generation method into a goal-oriented analysis method for requirements elicitation. In <i>12th Asia-Pacific Software Engineering Conference (APSEC'05)</i> (pp. 9-pp). IEEE.
[P19]	Maiden, N., & Robertson, S. (2005, August). Integrating creativity into requirements processes: Experiences with an air traffic management system. In <i>13th IEEE International Conference on Requirements Engineering (RE'05)</i> (pp. 105-114). IEEE.
[P20]	Maiden, N., Manning, S., Robertson, S., & Greenwood, J. (2004, August). Integrating creativity workshops into structured requirements processes. In <i>Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques</i> (pp. 113-122). ACM.
[P21]	Yasuoka, M., Kadoya, K., & Niwa, T. (2014, January). Introducing a game approach toward
[P22]	Zachos, K., & Maiden, N. (2008, September). Inventing requirements from software: an empirical investigation with web services. In <i>2008 16th IEEE International Requirements Engineering Conference</i> (pp. 145-154). IEEE.
[P23]	Karlsen, I. K., Maiden, N., & Kerne, A. (2009, June). Inventing requirements with creativity support tools. In <i>International Working Conference on Requirements Engineering: Foundation for Software Quality</i> (pp. 162-174). Springer Berlin Heidelberg.
[P24]	Maiden, N., Ncube, C., & Lockerbie, J. (2008, June). Inventing requirements: experiences with an airport operations system. In <i>International Working Conference on Requirements Engineering: Foundation for Software Quality</i> (pp. 58-72). Springer Berlin Heidelberg.
[P25]	Bhowmik, T., Niu, N., Savolainen, J., & Mahmoud, A. (2015). Leveraging topic modeling and part-of-speech tagging to support combinational creativity in requirements engineering. <i>Requirements Engineering</i> , 20(3), 253-280.
[P26]	Singer, L., Seyff, N., & Fricker, S. A. (2011, September). Online social networks as a catalyst for software and IT innovation. In <i>Proceedings of the 4th international workshop on Social software engineering</i> (pp. 1-5). ACM.
[P27]	Peffer, K., & Tuunanen, T. (2005). Planning for IS applications: a practical, information theoretical method and case study in mobile financial services. <i>Information &amp; Management</i> , 42(3), 483-501.
[P28]	Näkki, P., Koskela, K., & Pikkarainen, M. (2011, June). Practical model for user-driven innovation in agile software development. In <i>Concurrent Enterprising (ICE), 2011 17th International Conference on</i> (pp. 1-8). IEEE.
[P29]	Maiden, N., Gizikis, A., & Robertson, S. (2004). Provoking creativity: Imagine what your requirements could be like. <i>IEEE software</i> , 21(5), 68-75

[P30]	Sutcliffe, A., & Sawyer, P. (2013, July). Requirements elicitation: Towards the unknown unknowns. In <i>2013 21st IEEE International Requirements Engineering Conference (RE)</i> (pp. 92-104). IEEE.
[P31]	Robertson, S. (2001). Requirements trawling: techniques for discovering requirements. <i>International Journal of Human-Computer Studies</i> , 55(4), 405-421.
[P32]	Zachos, K., Maiden, N., & Tosar, A. (2005). Rich-media scenarios for discovering requirements. <i>IEEE software</i> , 22(5), 89-97.
[P33]	Grube, P. P., & Schmid, K. (2008, September). Selecting creativity techniques for innovative requirements engineering. In <i>2008 Third International Workshop on Multimedia and Enjoyable Requirements Engineering-Beyond Mere Descriptions and with More Fun and Games</i> (pp. 32-36). IEEE.
[P34]	Obrenovic, Ž., Gašević, D., & Eliens, A. (2008). Stimulating creativity through opportunistic software development. <i>IEEE software</i> , 25(6), 64-70.
[P35]	Seland, G. (2006, October). System designer assessments of role play as a design method: a qualitative study. In <i>Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles</i> (pp. 222-231). ACM.
[P36]	Pohl, K., & Sikora, E. (2007). The co-development of system requirements and functional architecture. In <i>Conceptual Modelling in Information Systems Engineering</i> (pp. 229-246). Springer Berlin Heidelberg.
[P37]	Sakhnini, V., Mich, L., & Berry, D. M. (2012). The effectiveness of an optimized EPMcreate as a creativity enhancement technique for Web site requirements elicitation. <i>Requirements Engineering</i> , 17(3), 171-186.
[P38]	Niknafs, A., & Berry, D. M. (2012, September). The impact of domain knowledge on the effectiveness of requirements idea generation during requirements elicitation. In <i>2012 20th IEEE International Requirements Engineering Conference (RE)</i> (pp. 181-190). IEEE.
[P39]	Kultima, A. (2010, May). The organic nature of game ideation: game ideas arise from solitude and mature by bouncing. In <i>Proceedings of the International Academic Conference on the Future of Game Design and Technology</i> (pp. 33-39). ACM.
[P40]	Hoffmann, H., Leimeister, J. M., & Krcmar, H. (2010). Tool support for the participatory design of end user oriented applications in the automobile.
[P41]	Jones, S., Lynch, P., Maiden, N., & Lindstaedt, S. (2008, September). Use and influence of creative ideas and requirements for a work-integrated learning system. In <i>2008 16th IEEE International Requirements Engineering Conference</i> (pp. 289-294). IEEE.
[P42]	Kim, S., In, H. P., Baik, J., Kazman, R., & Han, K. (2008). VIRE: Sailing a blue ocean with value-innovative requirements. <i>IEEE software</i> , 25(1), 80-87.

# APÊNDICE 2 – QUESTIONÁRIO DO ESTUDO EXPLORATÓRIO EM STARTUPS

## Práticas e Desafios no Registro de Ideias e Características de Software em Projetos de Inovação

### Caracterização da Empresa

---

Nome da empresa:

Ano de início de atuação da empresa:

Porte da Empresa<sup>6</sup>:

Favor escolher apenas uma das opções a seguir\*:

- Micro - até 9 funcionários
- Pequena - de 10 a 49 funcionários
- Média - de 50 a 99 funcionários
- Grande - mais de 100 funcionários

\* Considere também pessoas que não trabalham com desenvolvimento de software.

**Com quantos projetos de software com objetivo de inovação (incluindo os que estão em desenvolvimento) a empresa já se envolveu?**

Considere produtos como, por exemplo, aplicativos mobile, sistemas web, sistemas desktop, sistemas embarcados, software como serviço, etc.

**Quais o(s) domínio(s) de aplicação dos produtos de software com objetivo de inovação que a empresa desenvolve?**

Ex.: Administração Jurídica, Administração Escolar, E-commerce, Educação à Distância, Games para Entretenimento, Gestão de Documentos, Ferramentas ou ambientes para Desenvolvimento de Software, etc.

**Indique com que frequência a empresa considera importante os objetivos abaixo (a ponto de influenciar o planejamento de ações e tomadas de decisão nos projetos de software para inovação da empresa ).**

---

<sup>6</sup> Classificação SEBRAE (SEBRAE-DIEESE, 2016)

Por favor, escolha a resposta adequada para cada item:

	Sempre	Quase Sempre	Raramente	Nunca	Não sei dizer	Não aplicável
Oferecer <u>produto ou serviço</u> com <u>características inexistentes</u> em outros produtos/serviços e que chamem a atenção dos clientes/usuários pretendidos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Disponibilizar o <u>produto completo</u> o mais cedo possível no mercado/contexto pretendido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Disponibilizar <u>versões do produto</u> o mais cedo possível, mesmo que não tenha ainda todas as características planejadas para o produto final	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Coletar ou ter <u>ideias</u> para o produto a partir da sua <u>utilização pelos usuários</u>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Cite outros objetivos que influenciam o planejamento de ações e tomada de decisão nos projetos de software para inovação da empresa.**

#### **CARACTERIZAÇÃO DO RESPONDENTE**

**E-mail para contato\*:**

\* Seus dados não serão divulgados.

**Em quantos projetos de desenvolvimento de software para inovação você já participou na empresa?**

**Trabalha atualmente em algum projeto de software para inovação na empresa?**

Sim  Não

**Qual a sua área de formação?**

- Ciência da Computação
- Engenharia da Computação
- Sistemas de Informação
- Outros

**Indique seu nível de experiência em:**

Por favor, escolha a resposta adequada para cada item:

	Não tenho experiência	Estudei ou li a respeito	Pratiquei em 1 projeto exemplo anterior	Trabalhei em 1 projeto na indústria	Trabalhei em vários projetos na indústria
Análise e Especificação de Requisitos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projeto de Sistemas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Codificação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Teste	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gerenciamento de Projeto de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Que atividades você desempenha no(s) projeto(s) de desenvolvimento de software para inovação que participa atualmente?**

**Obs.: Pergunta é feita ao respondente sob a seguinte condição:**

A resposta foi 'Sim' na questão "Trabalha atualmente em algum projeto de software para inovação na empresa? "

#### INFORMAÇÕES DOS PROJETOS DE INOVAÇÃO

**Obs.: As perguntas desta sessão só aparecem aos respondentes sob a seguinte condição:**

A resposta foi 'Sim' na questão "Trabalha atualmente em algum projeto de software para inovação na empresa? "

**Responda os itens a seguir de acordo com a sua percepção sobre os projetos de software para inovação que ATUALMENTE participa na empresa.**

**O(s) projeto(s) de software para inovação que trabalho...**

- ... faz/fazem parte das estratégias de negócio centrais da empresa.
- ... é/são importante(s) mas não é/são parte das estratégias de negócio centrais da empresa.
- ... tem baixa prioridade em relação aos outros projetos da empresa.
- Outros

**Que situações indicam o(s) estágio(s) dos projeto(s) de software para inovação que atualmente trabalha? \***

- Refinando a ideia de negócio, formando a equipe e iniciando o projeto
- Prototipando ou construindo a primeira versão do produto
- Avaliando a aceitação do produto após as primeiras vendas/adesões das primeiras versões
- Adaptando o produto para as necessidades dos clientes que já aderiram ao produto
- Investindo ativamente no aumento de usuários
- Adicionando produtos complementares, expandindo o produto ou diversificando o portfólio para entrar em novos mercados/contextos

- Dando suporte aos usuários atuais e sem tempo ou recursos para crescer ou expandir de forma intensa
- Devido ao aumento da equipe e/ou da complexidade do produto, estamos melhorando ou adotando práticas de gerenciamento e desenvolvimento
- Outros:

**Caso haja a meta de disponibilizar software para o mercado/contexto pretendido no menor tempo possível (*time-to-market*), indique o(s) motivo(s):**

- Incerteza sobre a aceitação do produto no mercado ou contexto de uso pretendido
- Limitação de recursos financeiros - necessário obter ganhos para continuar a desenvolver o produto
- Pressão do investidor
- Risco de ser ultrapassado pela concorrência
- Outros:

**Que abordagens de desenvolvimento de software são utilizadas nos projetos de software para inovação?**

- Design Thinking*
- Lean Startup*
- Processo Ágil
- Processo Evolutivo
- Processo Iterativo e Incremental
- Processo Unificado
- Processo Sequencial (cascata)
- Não sei informar se adotamos abordagem de desenvolvimento de software
- Outros:

**Que abordagens ou práticas de gestão de projetos são utilizadas nos projetos de software para inovação?**

- Liberdade de escolha de tarefas pelas pessoas e autogerenciamento
- Elaboração e acompanhamento de um Plano de Projeto, com definição de cronograma e estimativa de esforço
- Reuniões diárias de acompanhamento de tarefas
- Abordagem Scrum
- Não sei informar que práticas e abordagens de gestão usamos
- Outros:

**De onde vem as IDEIAS para o software desenvolvido com objetivo de inovação?**

- Análise de outros produtos
- Análise de reclamações dos usuários do produto
- Cliente patrocinador do projeto
- Demanda de usuários do produto
- Fundadores da empresa
- Especialistas no mercado ou contexto pretendido
- Equipe de desenvolvimento
- Pesquisa de mercado
- Pessoas da empresa, mas que não são da equipe de desenvolvimento
- Outros:

**De que formas as ideias e características de software nos projetos para inovação são registradas?**

	Para registrar IDEIAS do Software	Para registrar CARACTERÍSTICAS do Software	Para registrar AMBOS
Texto livre informal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Descrição de Casos de Uso, Estórias de Usuário, <i>Storyboards</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Diagramas de Projeto da UML (ex.: classes, estados, sequência)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Documento com listagem de ideias ou características	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Especificação de Requisitos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Imagens com figuras e desenhos informais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mapas Mentais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Descrição de Personas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Descrição de um dia na vida do usuário (One Day in Life story)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Protótipos de Telas, <i>mockups</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Protótipos funcionais, mesmo que incompletos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Outros	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



Quais outras formas de registro de ideias e/ou características de software são utilizadas nos projetos de inovação que participa?

Se o **esforço** dedicado ao registro de ideias e características do software foi **reduzido no estágio inicial** do projeto para inovação, qual a sua opinião sobre os motivos para isso ter ocorrido?

**Obs.: Pergunta só é feita ao respondente sob as seguintes condições:** Na questão “Que situações indicam o(s) estágio(s) dos projeto(s) de software para inovação que atualmente trabalha?” A resposta foi 'Adicionando produtos complementares, expandindo o produto ou diversificando o portfólio para entrar em novos mercados/contextos' ou 'Dando suporte aos usuários atuais e sem tempo ou recursos para crescer ou expandir de forma intensa' ou 'Devido ao aumento da equipe e/ou da complexidade do produto, estamos melhorando ou adotando práticas de gerenciamento e desenvolvimento' ou 'Adaptando o produto para as necessidades dos clientes que já aderiram ao produto' ou 'Investindo ativamente no aumento de usuários' ou 'Avaliando a aceitação do produto após as primeiras vendas/adesões das primeiras versões'

- Alta probabilidade de mudanças nas características do software, dada a incerteza das ideias
- Falta de opções de modelos de documentação adequados para o tipo de projeto
- Falta de pessoas disponíveis para a tarefa
- Falta de pessoas com habilidades para a tarefa
- Limite de recursos financeiros
- Pouco tempo para desenvolver o produto
- Preferência da equipe
- Outros:

Se o **esforço** dedicado ao registro de ideias e características do software **aumentou após o estágio inicial** do projeto para inovação, qual a sua opinião sobre os motivos para isso ter ocorrido?

**Obs.: Pergunta só é feita ao respondente sob as seguintes condições:** Na questão “Que situações indicam o(s) estágio(s) dos projeto(s) de software para inovação que atualmente trabalha?” A resposta foi 'Adicionando produtos complementares, expandindo o produto ou diversificando o portfólio para entrar em novos mercados/contextos' ou 'Dando suporte aos usuários atuais e sem tempo ou recursos para crescer ou expandir de forma intensa' ou 'Devido ao aumento da equipe e/ou da complexidade do produto, estamos melhorando ou adotando práticas de gerenciamento e desenvolvimento' ou 'Adaptando o produto para as necessidades dos clientes que já aderiram ao produto' ou 'Investindo ativamente no aumento de usuários' ou 'Avaliando a aceitação do produto após as primeiras vendas/adesões das primeiras versões'

- Aumento da complexidade do software, o que aumentou o risco de inclusão de defeitos
- Aumento de funcionalidades oferecidas pelo software
- Dificuldade para incluir ou alterar funcionalidades e demais características do produto
- Existência de diferentes versões do produto em operação
- Inclusão de pessoas na equipe (necessidade de repassar informações)
- Saída de pessoas da equipe (necessidade de registrar informações)
- Outros:

Se você considera **necessário melhorar a qualidade e/ou aumentar a quantidade de documentação das ideias e características do software no projeto que participa**, indique o(s) motivo(s):

- Para alinhar o entendimento da equipe acerca da visão e objetivos do produto
- Para o desenvolvimento de novas funcionalidades no software

- Para alterar funcionalidades do software
- Para executar testes no software
- Para planejar e/ou acompanhar as atividades de desenvolvimento da equipe
- Para explicar as ideias e características do software para outra pessoa da equipe
- Para decidir quais características e funcionalidades o software deveria ter ou quais deveriam ser excluídas
- Outros:

# APÊNDICE 3 – INSTRUMENTOS DO ESTUDO DE AVALIAÇÃO DA TESE

## Caracterização da Empresa

---

Nome da empresa:

Ano de início de atuação da empresa:

Porte da Empresa<sup>7</sup>:

- Micro - até 9 funcionários
- Pequena - de 10 a 49 funcionários
- Média - de 50 a 99 funcionários
- Grande - mais de 100 funcionários

\* Considere também pessoas que não trabalham com desenvolvimento de software.

**Com quantos projetos de software com objetivo de inovação (incluindo os que estão em desenvolvimento) a empresa já se envolveu?**

Considere produtos como, por exemplo, aplicativos mobile, sistemas web, sistemas desktop, sistemas embarcados, software como serviço, etc.

**Que situações indicam o(s) estágio(s) do(s) projeto(s) de software para inovação que atualmente trabalha?**

- Refinando a ideia de negócio, formando a equipe e iniciando o projeto
- Prototipando ou construindo a primeira versão do produto
- Avaliando a aceitação do produto após as primeiras vendas/adesões das primeiras versões
- Adaptando o produto para as necessidades dos clientes que já aderiram ao produto
- Investindo ativamente no aumento de usuários
- Adicionando produtos complementares, expandindo o produto ou diversificando o portfólio para entrar em novos mercados/contextos
- Dando suporte aos usuários atuais e sem tempo ou recursos para crescer ou expandir de forma intensa
- Devido ao aumento da equipe e/ou da complexidade do produto, estamos melhorando ou adotando práticas de gerenciamento e desenvolvimento
- Outros:

---

<sup>7</sup> Classificação SEBRAE (SEBRAE-DIEESE, 2016)

**Quais o(s) domínio(s) de aplicação dos produtos de software com objetivo de inovação que a empresa desenvolve?**

Ex.: Administração Jurídica, Administração Escolar, E-commerce, Educação à Distância, Games para Entretenimento, Gestão de Documentos, Ferramentas ou ambientes para Desenvolvimento de Software, etc.

**Seque alguma metodologia de desenvolvimento? Qual?**

**Quantos protótipos/versões do software desenvolvido foram lançados?**

**A empresa tem apoio externo e/ou parcerias?**

Ex.: Participa de programa de incubadora, recebeu investimento de alguma fonte, entre outros.

**Cenário:**

---

Percebeu-se no contexto da sua empresa Startup a necessidade melhorar ou adotar novas práticas de compartilhamento e/ou registro de conhecimento sobre ideias e características relacionadas ao software desenvolvido na Startup.

A seguir, pensando na sua tomada de decisão:

- Indique se você considera importante observar características do contexto, problemas e objetivos listados.
- Indique a situação do contexto atual da sua Startup quanto a essas características do contexto, problemas e objetivos listados.
- Indique quais tecnologias você consideraria como candidatas na sua tomada de decisão.

## Instrumentos de Avaliação da Pertinência e Percepção de Atributos do Contexto Startup

### Análise das Características do Contexto da Startup

Equipe		
<b>NÍVEL DE CO-LOCALIZAÇÃO DA EQUIPE:</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
_____		
0: <i>Inexistente</i>	10: <i>Elevada</i>	Não observei ( )
<b>EXPECTATIVA DE MUDANÇAS NA EQUIPE DE SOFTWARE:</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
_____		
0: <i>Inexistente</i>	10: <i>Elevada</i>	Não observei ( )
<b>FREQUÊNCIA DE COMUNICAÇÃO DA EQUIPE DE SOFTWARE:</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
_____		
0: <i>Inexistente</i>	10: <i>Elevada</i>	Não observei ( )
<b>QUANTIDADE DE PESSOAS NA EQUIPE DE DESENVOLVIMENTO:</b>		
<b>NÍVEL DE UNIÃO DA EQUIPE:</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
_____		
0: <i>Inexistente</i>	10: <i>Elevada</i>	Não observei ( )
<b>Produto</b>		
<b>NÍVEL DA INCERTEZA SOBRE A ACEITAÇÃO DO PRODUTO NO MERCADO:</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
_____		
0: <i>Inexistente</i>	10: <i>Elevada</i>	Não observei ( )

<b>NÍVEL DE AUMENTO DA COMPLEXIDADE DO SOFTWARE:</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
<hr/>		
<i>0: Inexistente</i>	<i>10: Elevada</i>	Não observei ( )
<b>CUSTO DA GARANTIA DA QUALIDADE DO PRODUTO SEM APOIO DE DOCUMENTOS E/OU MODELOS DO SOFTWARE:</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
<hr/>		
<i>0: Inexistente</i>	<i>10: Elevada</i>	Não observei ( )
<b>A SOLUÇÃO DE SOFTWARE TEM INTEGRAÇÃO COM SOLUÇÕES DE HARDWARE E/OU SOFTWARE DE TERCEIROS?</b>		
Sim ( ) Não ( )		
<b>NÍVEL DA CRITICIDADE DE REQUISITO DE SEGURANÇA:</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
<hr/>		
<i>0: Inexistente</i>	<i>10: Elevada</i>	Não observei ( )
<b>Projeto</b>		
<b>NÍVEL DA LIMITAÇÃO DE RECURSOS (0 A 10):</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
<hr/>		
<i>0: Inexistente</i>	<i>10: Elevada</i>	Não observei ( )
<b>NECESSIDADE DE EXECUTAR CICLOS CURTOS DE DESENVOLVIMENTO E ENTREGA DE VERSÕES DO SOFTWARE:</b>		
Influencia a tomada de decisão? SIM ( ) NÃO ( )		
Situação atual (marque um valor entre 0 e 10):		
<hr/>		
<i>0: Inexistente</i>	<i>10: Elevada</i>	Não observei ( )

**Análise da Ocorrência de Dificuldades de Compartilhamento e/ou Registro de Conhecimento sobre Características de Software:**

**DIFICULDADE DE COMUNICAÇÃO ENTRE INTERESSADOS POR SEREM DE DIFERENTES ÁREAS DE TRABALHO (EX.: NEGÓCIO, DESENVOLVIMENTO, DESIGNER DE UX).**

Influencia a tomada de decisão? SIM ( ) NÃO ( )

Situação atual (marque um valor entre 0 e 10):

0: *Inexistente*

10: *Elevada*

Não observei ( )

**DIVERGÊNCIA ENTRE O ENTENDIMENTO DOS DESENVOLVEDORES SOBRE A IMPLEMENTAÇÃO DO SOFTWARE E O REAL FUNCIONAMENTO DE PARTES OU DE TODO O SOFTWARE.**

Influencia a tomada de decisão? SIM ( ) NÃO ( )

Situação atual (marque um valor entre 0 e 10):

0: *Inexistente*

10: *Elevada*

Não observei ( )

**FALTA DE CONHECIMENTO DA EQUIPE SOBRE A IMPORTÂNCIA QUE AS CARACTERÍSTICAS DO SOFTWARE TEM PARA O NEGÓCIO E PARA OS CLIENTES.**

Influencia a tomada de decisão? SIM ( ) NÃO ( )

Situação atual (marque um valor entre 0 e 10):

0: *Inexistente*

10: *Elevada*

Não observei ( )

**FALTA DE SINTAXE COMUM ENTRE AS PESSOAS DA EQUIPE, OU OCORRÊNCIA DE DIFERENTES INTERPRETAÇÕES PARA TERMOS EM COMUM.**

Influencia a tomada de decisão? SIM ( ) NÃO ( )

Situação atual (marque um valor entre 0 e 10):

0: *Inexistente*

10: *Elevada*

Não observei ( )

**DIFICULDADE DOS DESENVOLVEDORES EM DESCOBRIR INFORMAÇÕES SOBRE DEPENDÊNCIAS DE PROJETO DO SOFTWARE, LEVANDO-OS A IGNORAR REQUISITOS E RESTRIÇÕES.**

Influencia a tomada de decisão? SIM ( ) NÃO ( )

Situação atual (marque um valor entre 0 e 10):

0: *Inexistente*

10: *Elevada*

Não observei ( )

**Análise de Objetivos Atuais de Melhoria no Compartilhamento e/ou Registro de Conhecimento sobre Características de Software:**

<p><b>AUMENTAR OU REFORÇAR A HABILIDADE DE COMUNICAÇÃO DA EQUIPE:</b></p> <p>Influencia a tomada de decisão? SIM ( ) NÃO ( )</p> <p>Prioridade atual (marque um valor entre 0 e 10):</p> <hr/> <p><i>0: Inexistente</i> <i>10: Elevada</i> Não observei ( )</p>
<p><b>ALCANÇAR ENTENDIMENTO COMUM SOBRE O SOFTWARE PELA EQUIPE DE DESENVOLVIMENTO:</b></p> <p>Influencia a tomada de decisão? SIM ( ) NÃO ( )</p> <p>Prioridade atual (marque um valor entre 0 e 10):</p> <hr/> <p><i>0: Inexistente</i> <i>10: Elevada</i> Não observei ( )</p>
<p><b>APOIAR O DESENVOLVIMENTO DE SOFTWARE COM NÍVEL ADEQUADO DE EXPERIÊNCIA DE USUÁRIO:</b></p> <p>Influencia a tomada de decisão? SIM ( ) NÃO ( )</p> <p>Prioridade atual (marque um valor entre 0 e 10):</p> <hr/> <p><i>0: Inexistente</i> <i>10: Elevada</i> Não observei ( )</p>
<p><b>APOIAR A GESTÃO DE PROJETO E/OU DO PRODUTO:</b></p> <p>Influencia a tomada de decisão? SIM ( ) NÃO ( )</p> <p>Prioridade atual (marque um valor entre 0 e 10):</p> <hr/> <p><i>0: Inexistente</i> <i>10: Elevada</i> Não observei ( )</p>



# APÊNDICE 4 – DESCRIÇÃO GERAL CONTEXTOS DE STARTUPS MAPEADOS

Quadro 36. Atributos Descritivos dos Contextos de Startups Mapeados

Id.	Age	Stage	Size	Experience	External Support	Region	Methodology	Application Domain
1	6 years	Prototyping or building a first version of the product. Validating the product acceptance after the first sales/adoption of the first versions. Adapting the products to customer needs.	Micro (up to 9 employees)	5 software innovation projects		Brazil	Iterative and incremental development process	
2			Small	Recent ((less than 5 released products)		South-Eastern Finland		Gaming in mobile platforms
3			Small	Recent ((less than 5 released products)		South-Eastern Finland		Gaming in mobile platforms
4			Medium	Startup, developing first product.		South-Eastern Finland		Gaming (PC platforms
5			Small	Startup, developing first product.		South-Eastern Finland		Browser games
6	3 years	Adapting the product to customer needs Expanding the product	Small (2 people)	Startup, 1 product		Tampere, Finland	Incremental	Location-based applications
7			Medium	Startup, developing first product		South-Eastern Finland		Games to Mobile platforms and PC
8	3 years		12 employees		Academic Incubator	Brazil	Evolutionary approach	Website such as marketplace

Id.	Age	Stage	Size	Experience	External Support	Region	Methodology	Application Domain
10	1 year		10 employees		Academic Incubator	Brazil	Evolutionary approach	Website such as marketplace
11	1 year		8 employees		Academic Incubator	Brazil	Evolutionary approach	Website such as marketplace
12	4 months		10 employees		Academic Incubator	Brazil	Evolutionary approach	Website such as marketplace
13	More than 6 months	At least an initial customer set, i.e. first customer payments or a group of users		4 prototypes			Agile	News generator
14	At least 6 months	At least an initial customer set, i.e. first customer payments or a group of users		5 prototypes			Distributed Agile	Construction management
15	At least 6 months	At least an initial customer set, i.e. first customer payments or a group of users		5 prototypes			From adhoc to Distributed Agile	Educational quiz system
16	At least 6 months	At least an initial customer set, i.e. first customer payments or a group of users		2 prototypes			Ad hoc	Ticket platform
17	At least 6 months	At least an initial customer set, i.e. first customer payments or a group of users		3 prototypes				Shipping services
18	From 6 months to 20 years	Set of 6 startups, each one in one or more of the stages: - Refining the business idea, building the team, starting up: 1 startup - Prototyping or building a first version of the product: 4 startups - Validating the product acceptance after the firsts sales / adoption of the first versions: 3 startups - Adapting the products to customer			2 startups participate in Incubator program	5 from Brazil, 1 from Germany	Incremental and Iterative	

Id.	Age	Stage	Size	Experience	External Support	Region	Methodology	Application Domain
		needs. - Aggressively driving user growth: 2 startups - Adding complimentary products/ expanding the product/ Diversifying product portfolio in order to enter new markets: 2 startups						
19	From 5 to 43 months		From 3 to 20 employees			(United States (4), Italy (4), Germany (2), Sweden(1), United Kingdom (1), New Zealand (1)	Evolutionary Approach	pure web (8), web-and mobile (4), and weband desktop applications (1)
20	1 year	Prototyping or building a first version of the product.	Micro (up to 9 employees)		Incubator	Brazil	Iterative and incremental	
21	At least 3 years	At most Early Growth Stage	6 people	At least it has a first paying customer		Italy	Lean startup,Tailor Agile	Online Photo Marketplace
22	At least 6 months of operation	At most Early Growth Stage	3 people			Norway	Ad hoc, iterative and incremental	Marketplace of food hub
23	At least 3 years	At most Early Growth Stage	4 people			Norway	Distributed Scrum	Collaboration Platform for Construction
24	15 months	Validating the product acceptance after the first sales / adoption of the first versions.		version beta launched		NY,EUA	Agile, Lean Startup, Lean UX. (implemented near the project ending)	dating application

Id.	Age	Stage	Size	Experience	External Support	Region	Methodology	Application Domain
25	24 years of foundation as organization	Validating the product acceptance after the first sales / adoption of the first versions; Adapting the products to customer needs; Aggressively driving user growth; Adding complimentary products/ Expanding the product/ Diversifying product portfolio in order to enter new markets.	Micro (up to 9 employees)	6 software innovation projects		Brazil	Iterative and Incremental	
26	3 years		16 people		\$2.54 million in venture capital	Seattle, Washington, USA		Q&A retrieval (software help)
27	about 4 years		11 team members (4 engineers)		seed fund of \$300,000	EUA	inspired by Extreme Programming	mobile payments
28	about 2 years		5 people			Finland		SaaS application
29	At least 3 years	At most Early Growth Stage	18 people			Norway	Tailor Agile	Sale Visualization
30	At least 3 years	Implementation	3 people			Finland	Ad hoc	Under water camera product
31	Less than 1 year	Validating the product acceptance after the first sales / adoption of the first versions; Aggressively driving user growth; Adding complimentary products/ Expanding the product/ Diversifying product portfolio in order to enter new markets.	Micro (up to 9 employees)	6 software innovation projects		Brazil	Iterative and Incremental	
32	More than 6 months	At least an initial customer set, i.e. first customer payments or a group of users		3 prototypes			Agile	E-learning

Id.	Age	Stage	Size	Experience	External Support	Region	Methodology	Application Domain
33	Launched in 2014. Report is from 2017	Conceptual	2 founders, 4 employees, 1 developer			Canada		Health, Fitness (Mobile App)
34	Launched in 2014. Report is from 2017	Functional	2 founders, 6 employees, 1 developer			Italy		E-commerce (Web and Mobile app).
35	about 2.5 years		8 startups were interviewed		Participation in an incubator program or located in a scientific and technological park.	Rio Grande do Sul, Brazil		
36	2 years	Evolving their main product: Refining the business idea, building the team, starting up Prototyping or building a first version of the product.	Large (more than 99 employees)	A is worldwide known as a case of Startup success since its software product is implemented in the market of eleven countries (at the time of the study).		Germany	Iterative and incremental	
37	At least 3 years	Adapting the products to customer needs Adding complimentary products/ Expanding the product/ Diversifying product portfolio in order to enter new markets				Finland	Lean product development, agile software development, Kanban, Scrum, eXtreme programming	House holding services

Id.	Age	Stage	Size	Experience	External Support	Region	Methodology	Application Domain
38			6 startups were interviewed. They have from 3 to 15 people in software development				Tailored Version of XP	Telecommunications, Medical devices, Public Sector
39	More than 6 months	At least an initial customer set, i.e. first customer payments or a group of users		2 prototypes			Agile	Photo marketplace
40	More than 6 months	At least an initial customer set, i.e. first customer payments or a group of users		7 prototypes			Informal Agile	Underwater camera