

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
UNIDADE DE PÓS-GRADUAÇÃO, EXTENSÃO E PESQUISA
MESTRADO PROFISSIONAL EM GESTÃO E TECNOLOGIA EM SISTEMAS
PRODUTIVOS

ISABELLA DE ARAUJO CIONINI MENEZES

MELHORIA DE PRÁTICAS EM REQUISITOS DE *SOFTWARE* PARA
DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO: UMA PROPOSTA DE
ROADMAP

São Paulo
Junho/2022

ISABELLA DE ARAUJO CIONINI MENEZES

MELHORIA DE PRÁTICAS EM REQUISITOS DE *SOFTWARE* PARA
DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO: UMA PROPOSTA DE
ROADMAP

Dissertação apresentada como exigência parcial para a obtenção do título de Mestre em Gestão e Tecnologia em Sistemas Produtivos do Centro Estadual de Educação Tecnológica Paula Souza, no Programa de Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos, sob a orientação do Prof. Dr. Marcelo Duduchi Feitosa.

São Paulo

Junho/2022

Menezes, Isabella de Araujo Cionini

M543m Melhoria de práticas em requisitos de software para desenvolvimento de sistemas de informação: uma proposta de Roadmap / Isabella de Araujo Cionini Menezes. – São Paulo: CPS, 2022.

92 f. : il.

Orientador: Prof. Dr. Marcelo Duduchi Feitosa

Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos) – Centro Estadual de Educação Tecnológica Paula Souza, 2022.

1. Sistemas produtivos. 2. Requisitos. 3. Desenvolvimento de sistemas. 4. Sistemas de informação. 5. Roadmap. I. Feitosa, Marcelo Duduchi. II. Centro Estadual de Educação Tecnológica Paula Souza. III. Título.

ISABELLA DE ARAUJO CIONINI MENEZES

MELHORIA DE PRÁTICAS EM REQUISITOS DE *SOFTWARE* PARA
DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO: UMA PROPOSTA DE
ROADMAP

Prof. Dr. Marcelo Duduchi Feitosa
Orientador – CEETEPS

Profa. Dra. Lucia Vilela Leite Filgueiras
Examinador Externo – EPUSP

Profa. Dra. Marília Macorin de Azevedo
Examinador Interno - CEETEPS

São Paulo, 01 de julho de 2022

Aos meus amados pais, por todo carinho,
incentivo e apoio na realização de todos os
meus sonhos.

AGRADECIMENTOS

Ao professor Dr. Marcelo Duduchi Feitosa, pela paciência e por todos os ensinamentos transmitidos durante o mestrado.

A minha querida família por todo o incentivo e apoio e também por entenderem os momentos de ausência.

As minhas queridas amigas Ivânia Ramos dos Santos e as amigas do intercâmbio em Alicante, obrigada por todo o incentivo, vocês são inspirações para mim!

Aos meus amados sobrinhos Leonardo e Lorenzo, obrigada por tornarem os meus dias muito mais alegres!

A todos os professores e funcionários do Centro Paula Souza e de todas as instituições de ensino em que estudei, por tornarem esse momento possível.

“Se podemos sonhar, também podemos tornar
nossos sonhos realidade”
(Walt Disney)

RESUMO

MENEZES, I. A. C. **Melhoria de práticas em requisitos de *software* para desenvolvimento de sistemas de informação: Uma proposta de *roadmap***. 92f. Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos). Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2022.

As empresas, em geral, têm buscado constantemente melhorar a qualidade em seus processos de desenvolvimento de *software*. A área de requisitos é apontada como a responsável por muitos dos problemas que ocorrem no processo de desenvolvimento de *software*. Uma especificação de requisitos de má qualidade pode ainda diminuir a eficiência em todas as etapas restantes do desenvolvimento de um *software* e a compreensão incorreta dos requisitos ou a falta de gestão de mudanças destes ao longo do processo de desenvolvimento, podem acarretar diversos prejuízos aos sistemas de informação desenvolvidos. O presente trabalho, inserido na linha de pesquisa Sistemas de Informação e Tecnologias Digitais dentro do projeto Gestão da Tecnologia da Informação, tem por objetivo identificar as melhores práticas em requisitos de *software* e propor um *roadmap* que auxilie na identificação das práticas atuais em requisitos, identifique os problemas existentes e sugira a adoção de melhores práticas em requisitos de *software* em empresas. Para isto, foi realizada uma análise bibliográfica para identificar as melhores práticas em requisitos de *software* nos diferentes métodos de desenvolvimento e aplicada uma *survey* exploratória para que profissionais do mercado avaliem quais podem ser consideradas como melhores práticas em requisitos de *software*. Foram identificadas na literatura 27 práticas em requisitos que podem ser aplicadas do início ao fim do processo de desenvolvimento. Os resultados da *survey* apontaram um incentivo ao uso dos métodos ágeis por parte das organizações dos respondentes e confirmaram a aplicação de práticas já consagradas, além de identificar o uso inicial de algumas práticas menos conhecidas. O trabalho apresenta também uma avaliação preliminar do *roadmap* proposto que busca conduzir as empresas a melhorar suas práticas em requisitos a partir de uma autoavaliação das práticas atuais e sugestão de melhorias nas práticas de acordo com os principais problemas apontados nas atividades de requisitos.

Palavras-chave: Sistemas Produtivos. Requisitos. Desenvolvimento de Sistemas. Sistemas de Informação. *Roadmap*.

ABSTRACT

MENEZES, I. A. C. **Improvements in software requirements practices for information systems development: A roadmap proposal.** 92f. Dissertation (Professional Master in Management and Technology in Production Systems). Paula Souza State Technological Education Center, São Paulo, 2022.

Companies in general have constantly sought to improve the quality of their *software* development processes. The requirements area is identified as responsible for many of the problems that occur in the software development process. A poor quality requirements specification can also reduce the efficiency in all the remaining stages of software development and the incorrect understanding of the requirements or the lack of management of changes in these throughout the development process, can cause several damages to the systems. of information developed. The present work, inserted in the research line of Information Systems and Digital Technologies within the Information Technology Management project, aims to identify the best practices in software requirements and propose a roadmap that helps in the identification of current practices in requirements, evaluation of existing problems and suggests the adoption of best practices in software requirements in companies. For this, a bibliographic analysis was carried out to identify the best practices in software requirements in the different development methods and an exploratory survey was applied so that market professionals could evaluate which ones can be considered as best practices in software requirements. Twenty-seven requirements practices were identified in the literature that can be applied from the beginning to the end of the development process. The survey results point to an incentive to the use of agile methods by organizations and confirm the application of already established practices, but there is also an opportunity to identify the initial use of some lesser-known practices. This work also presents a preliminary evaluation of the proposed roadmap that seeks to lead companies to improve their practices in requirements from a self-assessment of current practices and suggestion of improvements in practices according to the main problems identified in the requirements activities.

Keywords: Production Systems. Requirements. Systems Development. Information Systems. *Roadmap.*

LISTA DE QUADROS

Quadro 1 - Princípios do Manifesto Ágil	25
Quadro 2 - Comparação entre métodos tradicionais e ágeis	26
Quadro 3 – Desenvolvimento de uma DSR.....	34
Quadro 4 – Publicações selecionadas	41
Quadro 5 – Práticas já utilizadas pela empresa XPTO	68
Quadro 6 – Práticas em requisitos antes e depois do RORSSI.....	70
Quadro 7 - Exemplo de priorização de demandas na empresa XPTO	71

LISTA DE TABELAS

Tabela 1 - Informações CHAOS Report 2020.....	36
---	----

LISTA DE FIGURAS

Figura 1 - Processo dos sistemas de informação	18
Figura 2 - Tipos de requisitos não funcionais.....	22
Figura 3 - Utilização do documento de requisitos	24
Figura 4 - Essência dos métodos ágeis	27
Figura 5 - Arquitetura de um roadmap genérico	30
Figura 6 - Layout Roadmap geral.....	31
Figura 7 - Fases do DSR.....	33
Figura 8 - Procedimentos de pesquisa	35
Figura 9 - Desenvolvimento do produto da pesquisa com DSR.....	39
Figura 10 - Aplicação do PRISMA-P.....	40
Figura 11 - Quantidade de publicações por ano	43
Figura 12 - Localização das empresas	55
Figura 13 - Tipo da organização.....	56
Figura 14 - Porte da organização.....	56
Figura 15 - Método de desenvolvimento de software	57
Figura 16 - Tipo de software desenvolvido	58
Figura 17 - Funções dos participantes da pesquisa na organização	58
Figura 18 - Práticas mais utilizadas pelas empresas para elicitação de requisitos	59
Figura 19 - Práticas mais utilizadas pelas empresas para documentação dos requisitos.....	60
Figura 20 - Práticas mais utilizadas pelas empresas para validação dos requisitos	61
Figura 21 - Práticas mais utilizadas pelas empresas para o gerenciamento dos requisitos	62
Figura 22 - Roadmap proposto	64
Figura 23 - Fase inicial do roadmap RORSSI.....	65
Figura 24 - Fase 2 do roadmap RORSSI.....	65
Figura 25 - Fase 3 do roadmap RORSSI.....	66
Figura 26 - Fase final do roadmap RORSSI.....	66
Figura 27 - Adaptações realizadas no roadmap RORSSI.....	68
Figura 28 - Fase 2 do RORSSI na empresa XPTO	68
Figura 29 - Fase 3 do RORSSI na empresa XPTO	69
Figura 30 - Fase final do RORSSI na empresa XPTO	70

LISTA DE SIGLAS

CEO	<i>Chief Executive Officer</i>
CONEP	Comissão Nacional de Ética em Pesquisa
CNS	Conselho Nacional de Saúde
CRM	<i>Customer Relationship Management</i>
DSR	<i>Design Science Research</i>
JAD	<i>Joint Application Development</i>
LGPD	Lei Geral de Proteção de Dados Pessoais
PRISMA-P	<i>Preferred Reporting Items for Systematic Review and Meta-Analysis Protocols</i>
RORSSI	<i>Roadmap para requisitos de software em sistemas de informação</i>
SI	Sistemas de Informação
TI	Tecnologia da Informação
UF	Unidade de Federação
UML	<i>Unified Modeling Language</i>
XP	<i>Extreme Programming</i>

SUMÁRIO

INTRODUÇÃO.....	15
1 SISTEMAS DE INFORMAÇÃO E O PROCESSO DE <i>SOFTWARE</i>	18
1.1 Sistemas de Informação	18
1.2 Engenharia de <i>Software</i>	19
1.3 Engenharia de Requisitos	20
1.4 Métodos Ágeis de Desenvolvimento de <i>Software</i>	25
2 ROADMAP	29
3 DESIGN SCIENCE RESEARCH	33
4 METODOLOGIA.....	35
4.1 Definição do Tema	36
4.2 Levantamento Bibliográfico	37
4.3 Pesquisa de Campo	37
4.4 Desenvolvimento <i>Roadmap</i>	38
4.5 Avaliação do <i>Roadmap</i>	39
5 RESULTADOS E DISCUSSÃO	40
5.1 Levantamento bibliográfico	40
5.2 Práticas em Requisitos de <i>Software</i> identificadas na literatura	44
5.3 Pesquisa De Campo	54
6 DESENVOLVIMENTO E APRESENTAÇÃO DO <i>ROADMAP</i> PROPOSTO	64
6.1 Aplicação do <i>Roadmap</i> Proposto em uma Empresa de TI	67
6.2 Avaliação do <i>Roadmap</i>	72
CONCLUSÃO.....	76
REFERÊNCIAS	78
APÊNDICES	87
APÊNDICE A – QUESTIONÁRIO PRÁTICAS EM REQUISITOS DE SOFTWARE	88
APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO DO <i>ROADMAP</i> PROPOSTO.....	92

INTRODUÇÃO

As empresas, em geral, têm buscado constantemente melhorar a qualidade em seus processos produtivos. Uma das áreas chave do processo produtivo, alvo dessa constante busca por melhoria, é o desenvolvimento de TI (Tecnologia da Informação), em que entregar um software que atenda às necessidades do cliente, no prazo e custo estipulados é cada vez mais desafiador.

Neste cenário, a área de requisitos é apontada como a responsável por muitos dos problemas que ocorrem no processo de desenvolvimento de software (PRESSMAN e MAXIM, 2016). Atividades como uma compreensão incorreta dos requisitos e falta de gestão de mudanças destes ao longo do desenvolvimento do software são apenas alguns exemplos de atividades que podem acarretar em diversos prejuízos aos projetos de software, podendo levar até mesmo ao seu cancelamento (TORO e PELÁEZ, 2016).

Boyarchuk *et al.* (2020) reforçam ainda que uma especificação de requisitos de má qualidade pode inclusive diminuir a eficiência em todas as etapas restantes do desenvolvimento de um software. Diante da criticidade das atividades que são realizadas por essa área, a engenharia de software torna-se fundamental para que essas atividades sejam realizadas com qualidade, seguindo os princípios, métodos e práticas abordados por ela.

Alguns desses métodos são indicados em ambientes com constantes mudanças e volatilidade em requisitos. É o caso dos métodos ágeis que se tornaram um grande aliado das empresas, pois permitem absorver mais rapidamente e com menos impactos mudanças em requisitos ao longo do desenvolvimento da aplicação. A dinâmica de se propor novas e diferentes soluções em um curto espaço de tempo permitiu um grande crescimento da utilização desses métodos (BORTOLUCI, DUDUCHI E ESTEVAM, 2014).

As abordagens ágeis consideram o projeto (design) e a implementação como atividades centrais no processo de software. Elas incorporam outras tarefas a essas atividades, como a elicitação dos requisitos e os testes, por exemplo. Para lidar com as mudanças constantes em requisitos, esses tipos de métodos não possuem uma atividade de engenharia de requisitos específica ou independente, a elicitação de requisitos é integrada ao desenvolvimento, sendo criados apenas os documentos que realmente irão agregar valor ao negócio do cliente (SOMMERVILLE, 2018).

Dessa forma, para criar essa documentação enxuta abordada pelo ágil, foram adaptadas algumas práticas para os requisitos, como é o caso das histórias de usuário que são dispostas em cartões simples e descrevem resumidamente as necessidades dos usuários e são criados

diretamente dentro do ambiente de desenvolvimento. Essa prática reduz o volume de documentação gerada para o sistema e permite que uma parte funcional do sistema seja entregue ao cliente o mais rápido possível.

Independentemente da metodologia escolhida para o desenvolvimento de uma aplicação, a engenharia de requisitos deve sempre ser observada e seus princípios e práticas aplicados para que o processo de desenvolvimento de software seja organizado e gere bons resultados.

A partir das dificuldades e problemas na especificação de requisitos atual e da importância que as práticas de elicitação e gerência de requisitos tem para o desenvolvimento de *software* e em particular dos sistemas de informação, o presente trabalho considera a seguinte questão de pesquisa:

O que um *roadmap* para a melhoria de requisitos de *software* deve conter?

Com base na presente questão de pesquisa, este trabalho tem por objetivo geral propor um *roadmap* com base nas melhores práticas em requisitos de *software* que contribua para a melhoria do processo de desenvolvimento de sistemas de informação das empresas.

Assim, para alcançar o objetivo geral deste estudo, foram definidos os seguintes objetivos específicos:

- Identificar as práticas adotadas para requisitos de *software* na literatura;
- Identificar como as empresas de desenvolvimento de *software* utilizam práticas de requisitos de *software*;
- Construir um *roadmap* para a avaliação de falhas na atividade de requisitos e adoção de melhores práticas em requisitos de *software*;
- Aplicar e avaliar o *roadmap* em uma empresa de desenvolvimento de *software* para verificar se o *roadmap* é factível, usável e útil.

Este trabalho foi desenvolvido no contexto do projeto de Gestão de tecnologia da informação da linha de pesquisa de Sistemas de Informação e Tecnologias Digitais do Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos em que são estudados tópicos relativos à governança e gestão da TI, eficácia dos projetos da área e sistemas de apoio à decisão.

Esta introdução contextualizou o tema e os objetivos do estudo e como o trabalho foi desenvolvido.

Os três primeiros capítulos apresentam o referencial teórico da presente dissertação.

O primeiro capítulo discute o processo de desenvolvimento de *software*, incluindo os

sistemas de informação, a engenharia de *software* e de requisitos e o tratamento dos requisitos em métodos tradicionais e ágeis de desenvolvimento de *software*.

O segundo capítulo trata do conceito e da construção de *roadmaps*, uma vez que o objetivo deste trabalho é a construção de um *roadmap*. Apresenta e comenta ainda *roadmaps* correlatos a este trabalho.

O terceiro apresenta o Design Science Research, que foi o método tomado como base para a construção do *roadmap* proposto.

O quarto capítulo mostra como a pesquisa foi conduzida e os métodos que foram utilizados para sua realização.

O quinto capítulo detalha os resultados da pesquisa bibliográfica, caracteriza os participantes da *survey* e faz a análise dos dados obtidos descrevendo também as demais práticas apontadas pelos participantes.

O sexto capítulo apresenta o *roadmap* proposto bem como sua construção e avaliação.

Ao final é apresentada a conclusão deste estudo que descreve as contribuições obtidas com a realização deste.

1 SISTEMAS DE INFORMAÇÃO E O PROCESSO DE *SOFTWARE*

Este capítulo aborda os conceitos referentes aos sistemas de informação e a engenharia de *software*, destacando os aspectos relacionados à engenharia de requisitos. Em seguida, discute o desenvolvimento de *software* por métodos ágeis e o tratamento dos requisitos nesta abordagem.

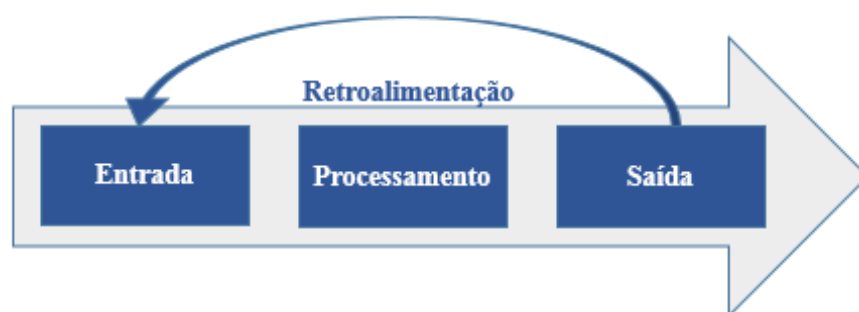
1.1 Sistemas de Informação

Com o constante avanço da TI, as empresas passaram a depender cada vez mais da informação e de sistemas computacionais. Mais do que nunca, informação significa “poder” e seu uso apropriado pode estabelecer o diferencial competitivo e um melhor atendimento a clientes, otimizando a cadeia de serviços, produtos e pesquisas (PEREIRA *et al.*, 2012).

A utilização dos sistemas de informação (SI) permite não somente agilizar os processos administrativos, como também redesenhar as relações com o meio ambiente organizacional, interagindo com outras organizações relacionadas, fornecedores e mais diretamente com seus clientes, obtendo assim valor agregado em seus produtos ou serviços (RAMÍREZ e VEGA, 2015).

Os sistemas de informação coletam, processam e disseminam informações a partir de determinado objetivo. Desse modo, incluem entradas, processamentos e saídas de informações. Fazem parte de um sistema de informação três atividades: entrada, processamento e saída. A entrada tem o papel de capturar os dados brutos internos ou externos à organização, o processamento converte os dados e realiza comparações e a saída faz a transferência das informações processadas para pessoas responsáveis pela tomada de decisão (MARTINS *et al.*, 2012). A figura 1 ilustra essas atividades que compõem os sistemas de informação.

Figura 1 - Processo dos sistemas de informação



Fonte: Adaptado de Laudon e Laudon (2014).

Côrtes (2017) afirma que, além destas atividades, é necessário considerar a existência de mecanismos de retroalimentação (feedback), possibilitando que o sistema seja realimentado com informações anteriormente geradas, permitindo o refinamento dos resultados obtidos ou análise de outras situações e possibilidades. Martins *et. al* (2012) ainda afirmam que estes sistemas são criados utilizando os conceitos da tecnologia da informação e fornecem condições para a tomada correta de decisões.

1.2 Engenharia de *Software*

A engenharia de *software* está diretamente relacionada ao desenvolvimento desses sistemas de informação, pois é um processo composto de atividades de qualidade, processos de desenvolvimento, metodologias e modelagem entre outros, que possibilitam a construção de sistemas com qualidade (CASTRO E SOUZA, 2016).

Assim, para o desenvolvimento do *software*, Pressman e Maxim (2016) consideram que um processo genérico de engenharia de *software* deve ser composto de pelo menos cinco atividades: comunicação, planejamento, modelagem, construção e entrega do *software*.

A comunicação é a etapa em que é realizada a comunicação com clientes e outros envolvidos. A intenção é entender os objetivos dos envolvidos para o projeto e reunir requisitos que ajudem a definir os recursos e as funções do *software*.

O planejamento é a etapa que ajuda a guiar a equipe em sua jornada. Um plano de projeto de *software* descreve as tarefas técnicas a serem conduzidas, os riscos prováveis, os recursos que serão necessários, os produtos resultantes a serem produzidos e um cronograma de trabalho.

A modelagem é a etapa na qual o engenheiro de *software* cria modelos para entender melhor as necessidades do *software* e o projeto que irá atender a essas necessidades.

A construção é a fase de construção do que foi projetado. Essa atividade combina geração de código (manual ou automatizada) e testes necessários para revelar erros de codificação.

Por fim, a entrega é a etapa em que o *software* (como uma entidade completa ou como um incremento parcialmente concluído) é entregue ao cliente, que avalia o produto entregue e fornece feedback, baseado na avaliação.

Ainda segundo Pressman e Maxim (2021), a prática da engenharia de *software* é uma atividade de resolução de problemas que segue um conjunto de princípios básicos, engloba uma série de processos, métodos e ferramentas que possibilitam a construção de sistemas complexos baseados em computador dentro do prazo e com o emprego de qualidade.

1.3 Engenharia de Requisitos

A engenharia de requisitos é uma parte da disciplina de engenharia de *software* que é incumbida de definir e reunir os requisitos que o sistema deverá atender; com essa definição, o desenvolvimento do *software* poderá ser iniciado (SILVA *et al.*, 2012). Sendo assim, o objetivo principal dessa fase é descobrir, analisar, documentar e verificar os requisitos do sistema (FAGUNDES *et al.*, 2020).

Se as necessidades do futuro usuário forem mal interpretadas, o produto será mal desenvolvido, resultando em um *software* de baixa qualidade, o que implica em muitas mudanças e retrabalhos (MENEZES *et al.*, 2015). Sendo assim, o quanto antes forem detectadas eventuais falhas, as correções tornam-se menos custosas.

Pressman e Maxim (2016) mostram que a engenharia de requisitos possui sete tarefas distintas, podendo estas ocorrer em paralelo. As sete tarefas são: concepção, levantamento, elaboração, negociação, especificação, validação e gestão de requisitos.

A concepção ocorre quando uma necessidade de negócio é identificada ou é descoberto um novo serviço ou mercado potencial. Nesta fase, se estabelece um entendimento básico do problema a ser solucionado.

No levantamento, é realizada a mobilização dos envolvidos no negócio do sistema para compartilhar suas metas. Uma vez que essas metas foram capturadas, deve ser estabelecido um mecanismo de atribuição de prioridades, podendo ser criado um raciocínio lógico para a possível arquitetura do projeto (que atenda às metas dos envolvidos).

Na fase de elaboração, as informações obtidas do cliente durante a concepção e o levantamento são expandidas e refinadas. Esta é guiada pela criação e pelo refinamento de cenários que descrevem como o usuário (e outros atores) vão interagir com o sistema. Cada cenário de usuário é analisado para extrair classes de análise e entidades do domínio de negócio visíveis para o usuário.

Na fase de negociação, são conciliados os conflitos que podem surgir. Deve ser solicitado a todos os envolvidos que ordenem seus requisitos e discutam sua prioridade, avaliando também seus custos e riscos, para que então sejam eliminados, combinados e/ou modificados de modo que cada parte atinja certo nível de satisfação.

Na fase de especificação, pode-se gerar um documento por escrito, um conjunto de modelos gráficos, um modelo matemático formal, um conjunto de cenários de uso, um protótipo ou ainda qualquer combinação desses.

Na fase de validação, artefatos produzidos pela engenharia de requisitos têm sua

qualidade avaliada. É validado se os requisitos não apresentam inconsistências, ambiguidade, omissões ou erros.

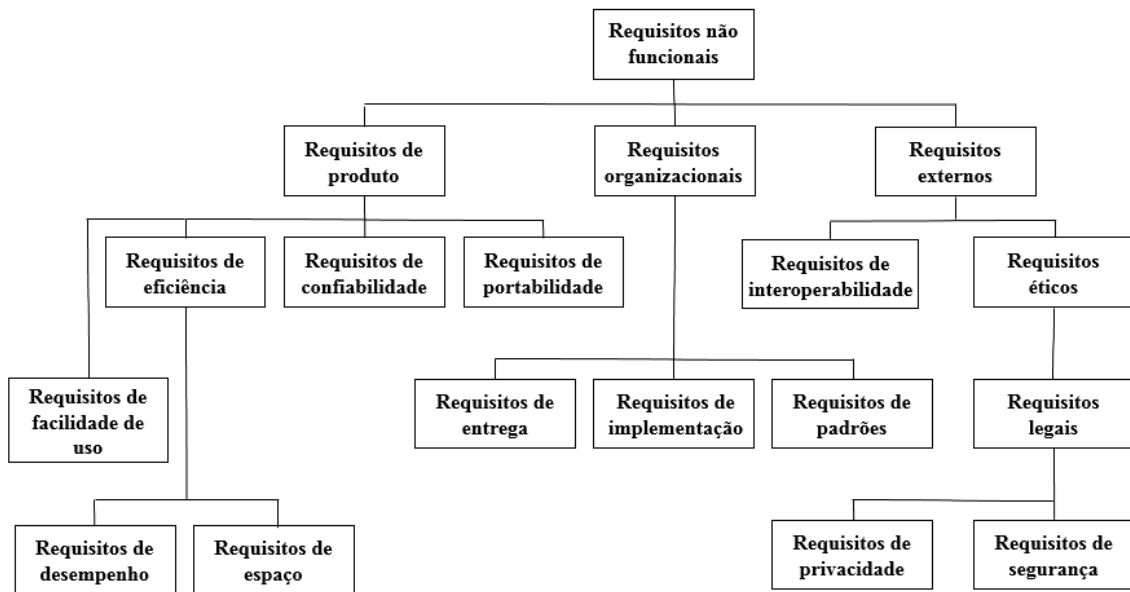
Por fim, a gestão de requisitos caracteriza-se por um conjunto de atividades que auxiliam a identificar, controlar e acompanhar as necessidades e suas mudanças à medida que os requisitos são desenvolvidos.

Assim, os requisitos de *software* definem um atributo, uma capacidade, uma característica ou uma qualidade que um sistema deve exibir para ter valor para os usuários e clientes. São propriedades que os sistemas (ainda em projeto) devem manifestar quando estiverem desenvolvidos (FERNANDES E MACHADO, 2017). Devem apresentar as funcionalidades que o sistema precisa atender (requisitos funcionais), as restrições para essas funcionalidades (requisitos não funcionais) e as regras de negócio (requisitos do domínio).

De acordo com Vazquez e Simões (2016), os requisitos funcionais descrevem o que o *software* faz, considerando uma perspectiva de tarefas e serviços de seus usuários em específico. Indicam ainda como o sistema deve reagir a determinadas entradas e verifica o comportamento do sistema em determinadas situações (GUALHANO, SILVA E VASCONCELOS, 2019).

Machado (2016) reforça ainda que a coleta dos requisitos funcionais irá depender do conhecimento que os usuários passam sobre o negócio. A especificação de um requisito funcional deve determinar o que se espera que o *software* faça, sem a preocupação de como ele faz, ou seja, como o *software* deve atender o negócio.

Os requisitos não funcionais estão relacionados a aspectos de restrições e qualidades específicas que as funcionalidades devem ter, tais como precisão, usabilidade, segurança, rendimento, confiabilidade, performance, entre outras (PEREIRA *et al.*, 2013). Uma especificação de requisitos não funcionais de qualidade também evita que ocorram defeitos que depois que o sistema for implementado, pois se tornam mais custosos de resolver (HERNÁNDEZ, DIHIGO E CINTRA, 2020). A figura 2 mostra os tipos de requisitos não funcionais.

Figura 2 - Tipos de requisitos não funcionais

Fonte: Machado (2016).

Como mostra a figura 2, os requisitos não funcionais podem ser de produto, organizacionais e externos. Dentre os requisitos de produto há os requisitos de facilidade de uso, de eficiência que podem estar relacionados ao desempenho ou ao espaço, requisitos de confiabilidade e portabilidade. Entre os requisitos organizacionais, há aqueles relacionados à entrega do *software*, da implementação do *software* e de padrões organizacionais. Entre os requisitos externos temos os relacionados à interoperabilidade e os requisitos éticos e legais relacionados à privacidade e segurança.

Os requisitos do domínio, também conhecidos como “regras de negócio”, são aqueles requisitos conhecidos do processo, do negócio da empresa, do cliente. Gualhano, Silva e Vasconcelos (2019) definem os requisitos de domínio como sendo requisitos originados do domínio da aplicação do sistema, ou seja, que refletem as características desse domínio, podendo ser funcionais ou não funcionais.

Ainda de acordo com Sommerville (2018), os requisitos de domínio podem ser, em sua essência, novos requisitos funcionais, limitar requisitos funcionais existentes ou estabelecer como determinadas computações devem ser executadas.

Documentar todos estes requisitos é de extrema importância, pois os artefatos gerados guiarão a equipe durante todo o processo de desenvolvimento. Vazquez e Simões (2016) afirmam que a especificação de requisitos é um contrato entre clientes e equipe de desenvolvimento. Ela deve esclarecer aos clientes o que será entregue como produto do trabalho da equipe de desenvolvimento. Esses clientes devem ser capazes de compreender a mensagem

e fornecer *feedback* sobre eventuais falhas na especificação, para que estas sejam corrigidas de imediato, antes que o trabalho errado seja produzido no desenvolvimento. O objetivo é que os clientes aprovelem de forma consciente a especificação de requisitos.

Ainda segundo Vazquez e Simões (2016), a especificação de requisitos não se trata apenas de um único documento. Pode ser a junção de vários tipos de documentos, sendo comum abranger a visão geral, um glossário, os modelos do sistema, a lista de requisitos funcionais e não funcionais e a especificação detalhada de requisitos.

A visão geral cita os objetivos, principais partes interessadas, um escopo preliminar com uma breve descrição das funções que o sistema deverá desempenhar (exemplo: documento de visão).

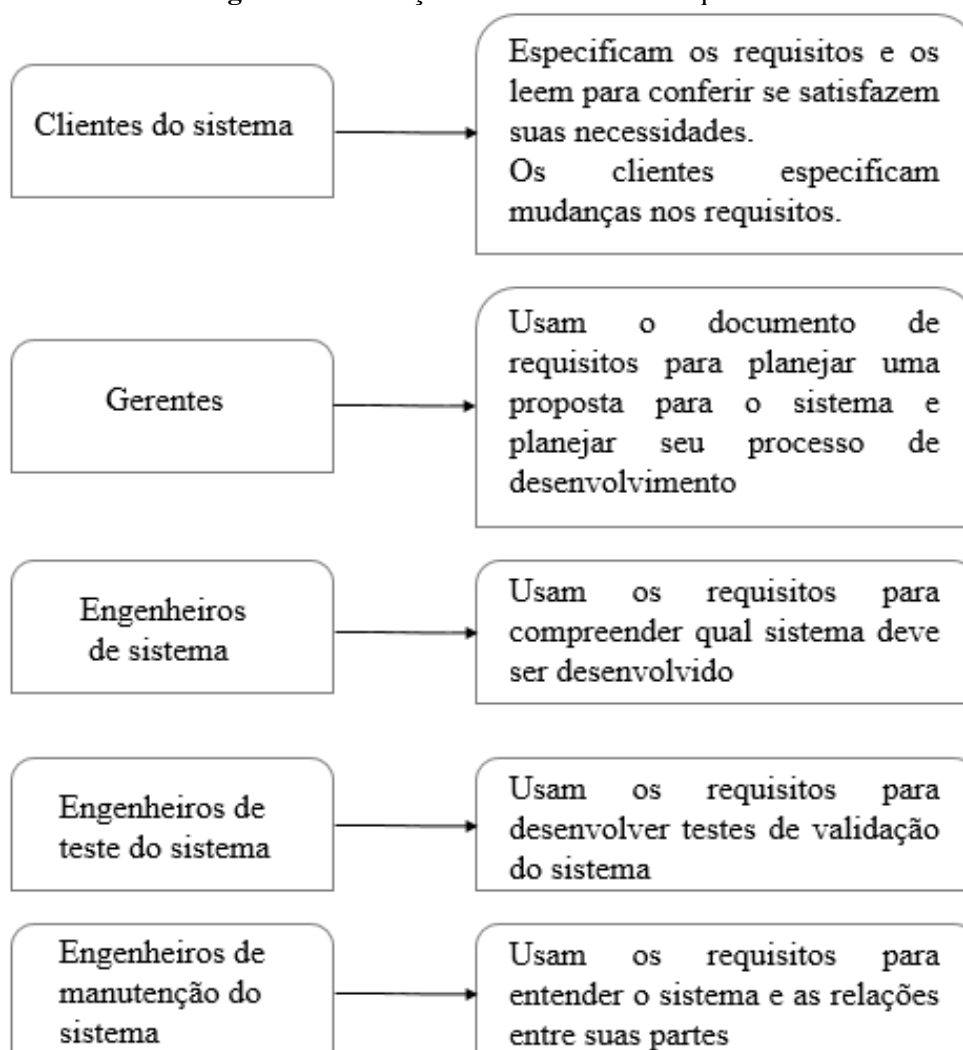
O glossário apresenta a definição dos termos técnicos (de negócio), sinônimos e acrônimos (siglas) usados ao longo do documento.

Os modelos de sistema mostram o relacionamento entre os componentes do sistema e entre o sistema e seu ambiente como diagramas de contexto, diagramas de caso de uso e modelo de processo, por exemplo.

A lista de requisitos funcionais descreve tarefas e serviços que serão fornecidos pelo sistema aos seus usuários como lista de casos de uso e histórias de usuário, por exemplo. Inclui também as interfaces externas do *software*.

A lista de requisitos não funcionais descreve as restrições impostas sobre o *software* e as relaciona aos requisitos funcionais.

Por fim, a especificação detalhada de requisitos detalha os requisitos funcionais como especificações de caso de uso e regras de negócio, por exemplo. É um documento utilizado por diferentes pessoas durante o desenvolvimento. A figura 3 indica como cada função geralmente faz uso do documento de requisitos.

Figura 3 - Utilização do documento de requisitos

Fonte: Adaptado de Sommerville (2018).

Como é possível notar na figura 3, o documento de requisitos não é utilizado apenas na fase de especificação de requisitos. Diferentes pessoas e equipes o utilizam, pois é um norteador para o desenvolvimento, que deve necessariamente ser atualizado conforme o entendimento dos requisitos evolui ou são identificados novos requisitos. Entretanto, a forma de documentar requisitos pode variar dependendo da empresa e metodologia de desenvolvimento adotada por esta.

Laplane (2014) reforça que muitas vezes é necessário escrever e reescrever, repetidamente a documentação de requisitos e revisá-lo com outras partes interessadas pois, conforme o entendimento do requisito avança e as partes interessadas modificam ou adicionam outros requisitos, é fundamental que a documentação detalhe fielmente o requisito final. Na abordagem ágil, apenas são criados artefatos de especificação de *software* que realmente agregam valor ao cliente, tornando a documentação de requisitos mais enxuta.

Ademais, Sommerville (2018) destaca que quando os sistemas têm o seu desenvolvimento terceirizado, ou seja, times diferentes desenvolvem partes do sistema, uma análise detalhada de requisitos é fundamental. Em outras circunstâncias, este documento pode ser feito de forma simplificada.

Para que essa documentação seja produzida corretamente e seja atualizada conforme a necessidade, os autores Khan, Khalid e Haq (2013) afirmam que atividades como rastreamento, priorização e controle de mudanças de requisitos podem ser realizadas para garantir o gerenciamento dos requisitos.

O gerenciamento de requisitos acompanha os requisitos desde a sua concepção até a sua evolução. É essencial para manter o controle dos requisitos durante todo o ciclo de vida do *software* (DIAS e SOUZA, 2018).

A seguir, são abordados os métodos ágeis de desenvolvimento de *software*.

1.4 Métodos Ágeis de Desenvolvimento de *Software*

Segundo Bortoluci, Duduchi e Estevam (2014), os métodos ágeis são uma resposta às novas demandas de mercado que requerem *softwares* que atendam ao levantamento de necessidades de maneira dinâmica em curto espaço de tempo. Esses métodos surgiram no ano de 2001 quando um grupo de dezessete especialistas em metodologias de desenvolvimento se reuniu com o intuito de padronizar seus processos, unificaram princípios comuns dentre os métodos ágeis de desenvolvimento, formando assim o manifesto ágil (SILVA, SOUZA e CAMARGO, 2013). Este, conta com um total de doze princípios que são apresentados no Quadro 1.

Quadro 1 - Princípios do Manifesto Ágil

Nº	Princípio
1	Priorização da satisfação do cliente por meio da entrega contínua e adiantada de <i>software</i> com valor agregado.
2	Mudanças nos requisitos são bem aceitas, mesmo que tardiamente no desenvolvimento.
3	Entrega frequente de <i>software</i> funcionando desenvolvido em poucas semanas ou meses, com preferência à menor escala de tempo possível.
4	Pessoas de negócio e desenvolvedores trabalham diariamente em conjunto por todo o projeto.
5	Equipes são motivadas para a construção de projetos. Fornecimento de ambiente e suporte necessário e a confiança de que os funcionários farão seu trabalho.
6	Considera a conversa face a face o método mais eficiente e eficaz de se transmitir uma informação.
7	<i>Software</i> funcionando é a medida primária de progresso.

Nº	Princípio
8	Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
9	Atenção contínua à excelência técnica e bom design aumentam a agilidade.
10	Simplicidade: a arte de maximizar a quantidade de trabalho não realizado é essencial.
11	As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.
12	Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

Fonte: Adaptado de *The agile manifesto* (2001).

Conforme apresentado no quadro 1, segundo os princípios do manifesto ágil, esses tipos de métodos de desenvolvimento de *software* tem um foco maior na comunicação face a face do que em documentos, prezam pela entrega contínua de partes funcionais do produto e possui uma atuação forte do cliente ao longo do processo de desenvolvimento.

Para Sharma e Hasteer (2016), diferentemente dos métodos tradicionais de desenvolvimento de *software*, que seguem um plano fixo e sequencial, enfatizando a documentação e sendo rígido a qualquer nova mudança, os métodos ágeis se destacam pela valorização das interações entre as equipes e clientes, bem como a rápida resposta e absorção das mudanças identificadas como sendo necessárias nessas constantes interações (SILVA E LOVATO, 2016). Com o objetivo de distinguir um pouco os métodos tradicionais dos métodos ágeis, o Quadro 2 apresenta um comparativo entre esses métodos de desenvolvimento de *software*.

Quadro 2 - Comparação entre métodos tradicionais e ágeis

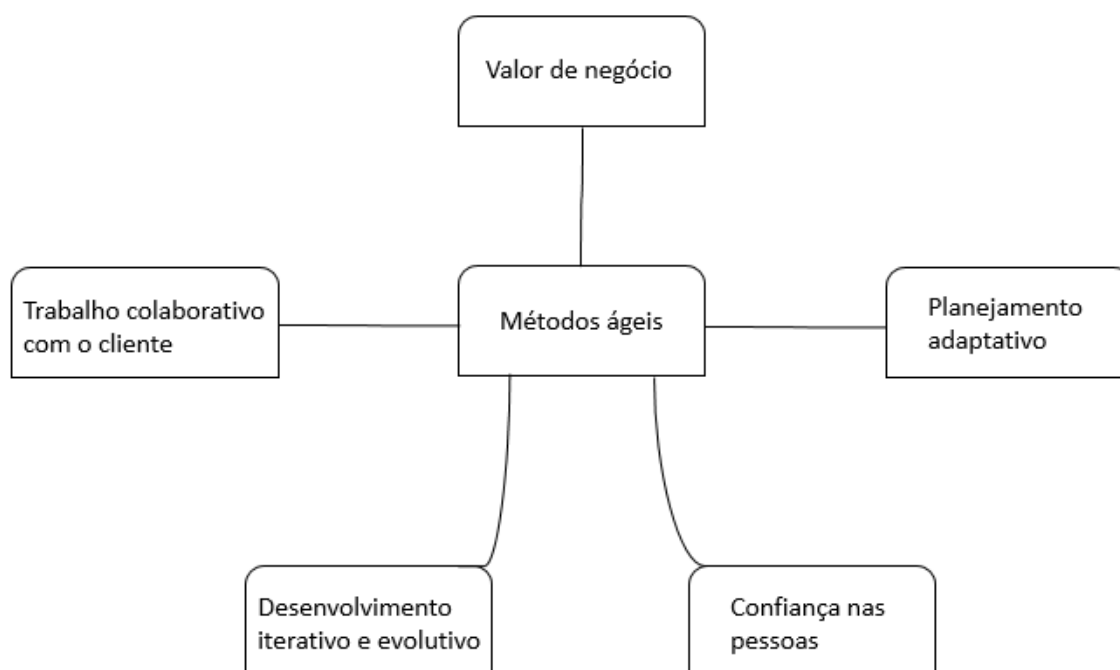
	Tradicional	Métodos ágeis
Pressupostos fundamentais	Sistemas totalmente especificáveis, previsíveis; desenvolvidos a partir de um planejamento extensivo e meticuloso	<i>Software</i> adaptativo e de alta qualidade; pode ser desenvolvido por equipes pequenas utilizando os princípios da melhoria contínua e testes orientados a rápida resposta a mudanças
Controle	Orientado a processos	Orientado a pessoas
Estilo de gerenciamento	Comandar e controlar	Liderar e colaborar
Gestão do conhecimento	Explícito	Tácito
Atribuição de papéis	Individual – favorece a especialização	Times auto-organizáveis – favorece a troca de papéis
Comunicação	Formal	Informal
Ciclo do projeto	Guiado por tarefas ou atividades	Guiado por funcionalidades do produto
Modelo de desenvolvimento	Modelo de ciclo de vida (Cascata, Espiral ou alguma variação)	Modelo iterativo e incremental de entregas

	Tradicional	Métodos ágeis
Forma/estrutura organizacional desejada	Mecânica (burocrática com muita formalização)	Orgânica (flexível e com incentivos a participação e a cooperação social)

Fonte: Adaptado de Prikladnicki, Willi e Milani (2014).

As equipes ágeis possuem autonomia para tomar decisões técnicas necessárias à conclusão do trabalho. O planejamento é mantido em um nível mínimo, e a equipe tem a permissão para escolher sua própria abordagem (por exemplo: método e ferramentas), limitada somente pelos requisitos de negócio e pelos padrões organizacionais (PRESSMAN E MAXIM, 2016). A figura 4 apresenta os princípios dos métodos ágeis.

Figura 4 - Essência dos métodos ágeis



Fonte: Adaptado de Pantaleo e Rinaudo (2015).

Como apresentado na figura 4, os métodos ágeis têm um desenvolvimento com entregas frequentes, o planejamento do projeto é adaptativo e apenas são realizados artefatos que agregam valor ao negócio do cliente. Além disso, o cliente é envolvido durante todo o processo e atua de perto com a equipe.

A rápida absorção de mudanças de requisitos, o que é característico dos métodos ágeis, é possível devido ao seu desenvolvimento iterativo em pequenas partes. A mudança de requisitos nos métodos ágeis não é vista como uma ameaça, mas sim como oportunidades. Beedle *et al.* (2001) afirmam que os processos ágeis tiram vantagem das mudanças em requisitos, visando vantagem competitiva para o cliente.

De acordo com Puri (2009), a experiência da equipe de desenvolvimento desempenha um papel importante nesta situação. A arquitetura do *software* deve ser fácil de modificar e expandir no futuro; sendo assim, uma boa equipe irá garantir que mudanças imprevistas custem o mínimo possível de retrabalho e custos para o cliente.

De acordo com Sommerville (2018), os métodos ágeis não possuem uma atividade de engenharia de requisitos específica ou independente. No geral, a elicitação de requisitos ocorre de maneira integrada ao desenvolvimento. Neste contexto, grande parte dos métodos ágeis utilizam o conceito de “histórias de usuário”, que tem o intuito de formar cenários de uso baseados nas expectativas de um usuário do sistema.

Essas histórias de usuário são organizadas em uma lista de pendências do produto e então são acordadas quais as histórias serão desenvolvidas na próxima iteração em uma sessão de planejamento. Essa escolha é feita com base na prioridade e na velocidade da equipe. Após essa definição, as histórias que foram contidas na iteração são congeladas e as solicitações de mudança vão sendo consideradas em iterações futuras (WIEGERS E BEATTY, 2013).

Puri (2009) salienta que é preciso avaliar o uso dos métodos ágeis, caso o cliente exija a produção de uma documentação (como por exemplo, um documento de requisitos) para a aprovação por uma ou mais partes interessadas do projeto ou caso seja necessário a entrega de modelos e documentação para outra equipe que irá desenvolver o sistema, pois neste tipo de método, a coleta de requisitos é feita de maneira informal, sem a criação de uma documentação para este processo.

Alguns críticos consideram que esta ausência de documentação de requisitos pode acarretar falhas e consequentemente em retrabalhos. Além disso, a abordagem de histórias de usuário pode não oferecer uma base suficiente para a evolução do sistema com o passar do tempo por não apresentar detalhes dos requisitos. Por essa razão, é fundamental observar se a metodologia atende os critérios do desenvolvimento desde o seu início até a finalização (PRESSMAN E MAXIM, 2016).

2 ROADMAP

Além de discutir os aspectos que permeiam o desenvolvimento de *software* é necessário também discutir o conceito e construção de *roadmaps*, uma vez que o objetivo deste trabalho é a construção de um *roadmap*.

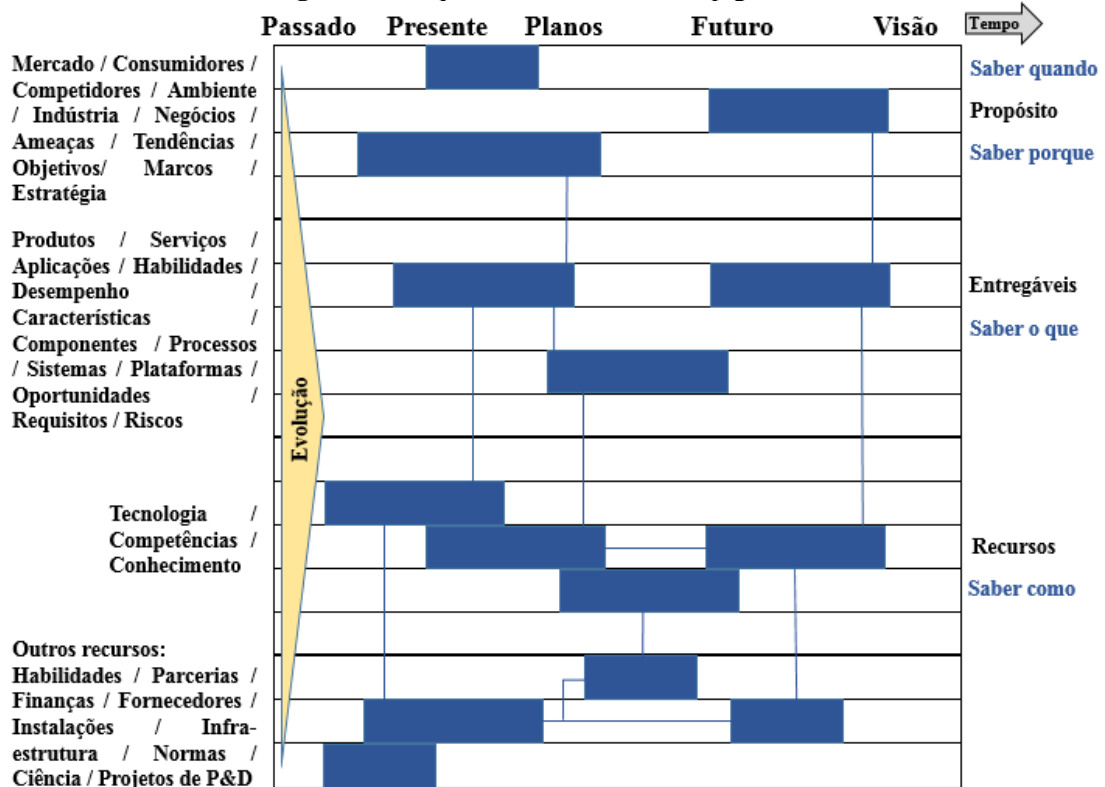
Para Albright (2003), um *roadmap* descreve um ambiente futuro, objetivos a serem alcançados nesse ambiente e planos de como esses objetivos serão alcançados ao longo do tempo. A utilização de *roadmaps* procura responder essas três questões: 1) Para onde estamos indo? 2) Onde estamos agora? e 3) Como chegaremos lá? (PHAAL, FARRUKH, PROBERT, 2005).

Além disso, eles apresentam e comunicam de forma eficaz a estratégia a ser adotada para atingir um objetivo, permitindo coordenar, alinhar e mobilizar esforços das partes envolvidas para a obtenção deste (TREITEL, 2005). Os *roadmaps* possuem diferentes formas e seu uso pretendido define a que nível de detalhe será desenvolvido.

Coelho, Junior e Tahlm (2012) afirmam que a construção de um *roadmap* se inicia a partir de um conjunto de necessidades. Na primeira fase, é importante identificar qual a necessidade do *roadmap* e se definir o seu escopo. A segunda fase de construção deve ser focada no desenvolvimento efetivo do *roadmap*, ou seja, a identificação do produto que será o foco do estudo, a identificação dos requisitos críticos do sistema e seus alvos, a especificação das principais áreas tecnológicas, identificação de alternativas tecnológicas e prazos para a implementação e a recomendação de tecnologias alternativas que deverão ser buscadas.

Ainda segundo estes mesmos autores, a terceira fase inclui atividades de continuidade que visam as críticas e validação do *roadmap*, indicam uma análise das tecnologias alternativas recomendadas e os objetivos a serem alcançados; o desenvolvimento do plano de implementação; e a revisão e atualização do estudo. É importante garantir que os principais grupos de interesse estejam envolvidos na implementação dos resultados. A Figura 5 ilustra a arquitetura de um *roadmap* genérico.

Figura 5 - Arquitetura de um roadmap genérico



Fonte: Adaptado de Phaal, Farrukh e Probert (2005).

Conforme ilustrado pela Figura 5, na arquitetura de um *roadmap* genérico, no topo deste são tratados os propósitos que as organizações desejam atingir; logo após, na camada intermediária, é abordado de que forma as empresas podem alcançar esses objetivos, e por último, são apresentados os recursos necessários para alcançar o que foi estabelecido nas camadas anteriores.

Foram encontrados três trabalhos relacionados que abordam *roadmaps* voltados a requisitos de *software*. O primeiro artigo identificado, intitulado “*Requirements Engineering: A Roadmap*”, foi citado por mais de 3.000 trabalhos. Neste artigo, os autores Nuseibeh e Easterbrook (2000) relatam a dificuldade de transformar as necessidades dos envolvidos em requisitos de *software*, citam práticas como entrevistas, questionários, prototipação e controle de mudanças de requisitos.

O artigo não apresenta de maneira visual um *roadmap*, apenas utiliza o conceito de apresentar “onde se está no momento” e “onde se deseja chegar”, ou seja, apresenta o cenário de requisitos da época em que o artigo foi produzido e os autores mostram suas perspectivas para um cenário futuro de requisitos.

Sendo uma publicação com mais de vinte anos, o Manifesto Ágil ainda não havia surgido. É possível notar que são abordadas apenas práticas em requisitos mais antigas. Os

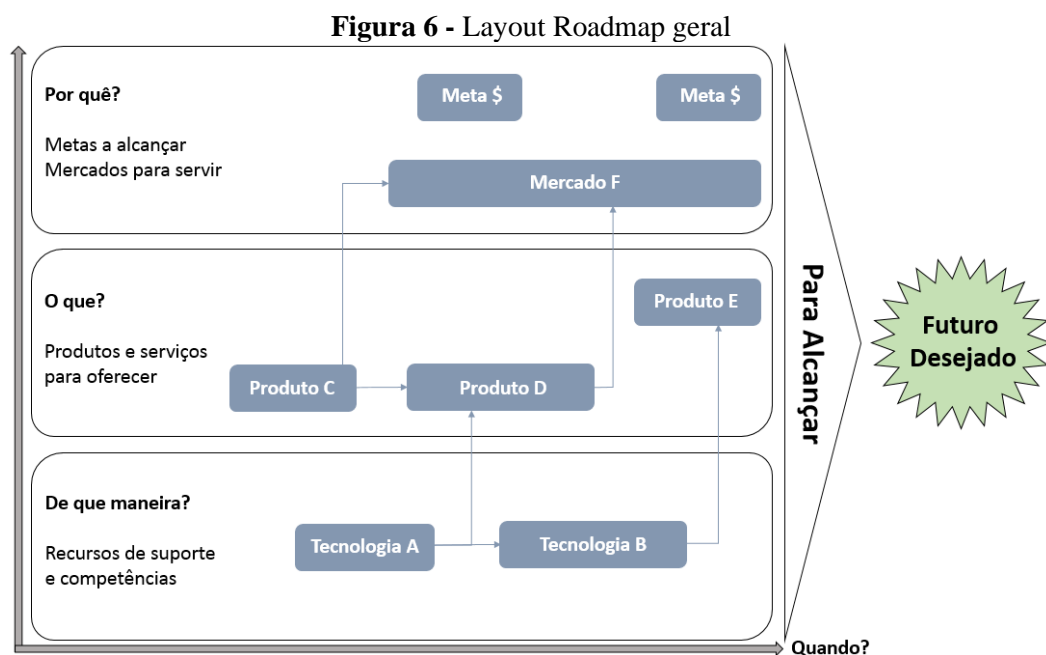
autores inclusive colocam como perspectivas futuras o surgimento de novas práticas para o tratamento efetivo de mudanças de requisitos (o que é visto como um problema para os métodos tradicionais de desenvolvimento de *software*) e sinalizam que a especificação de requisitos sempre será o ponto chave para o sucesso do *software*.

No segundo artigo analisado, também do ano 2000, intitulado “*Software Engineering: a Roadmap*”, os autores Finkelstein e Kramer (2000) afirmam que o desenvolvimento de *software* estava passando por mudanças, sendo que deixaram de ser produzidos sob medida e passaram a ser desenvolvidos para grandes mercados.

Também sinalizaram que a engenharia de requisitos estava sendo radicalmente reformulada para a orientação a objetos, com a aplicação de ambientes de teste e métodos de desenvolvimento com entrega incremental. Segundo os autores, isso permitiria novas oportunidades e desafios para a área de requisitos de *software*. Sendo assim, também é possível notar que é abordada somente uma visão mais antiga das práticas em requisitos de *software*.

No terceiro artigo, intitulado “*Roadmapping: (Missed) opportunities to overcome strategic challenges*”, Siebelink *et al.* (2021) não abordam especificamente *roadmaps* de requisitos, mas trazem uma abordagem atualizada sobre a construção de *roadmaps* no geral.

Os autores definem o *roadmap* como sendo um roteiro que mostra uma ou mais rotas possíveis a serem seguidas para chegar a um destino, ou seja, a sua utilização aponta os passos necessários para se alcançar os objetivos estratégicos, auxiliando as organizações na tomada de decisão. A Figura 6 mostra o layout de um *roadmap* geral.



Fonte: Adaptado de Siebelink *et al.* (2021).

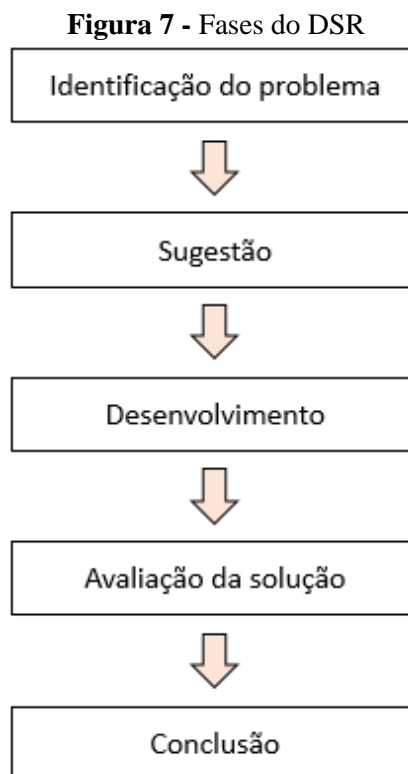
Como já exemplificado acima, na Figura 6 é possível ver que os *roadmaps* em geral devem permitir responder as três perguntas relacionadas as metas, o que se têm no momento e de que forma a meta será atingida e os passos para que cada fase seja atingida de acordo com o tempo estipulado, sendo uma ferramenta norteadora para as empresas na obtenção de resultados.

3 DESIGN SCIENCE RESEARCH

Como o desenvolvimento do artefato proposto nesse trabalho foi baseado no *Design Science Research*, é importante, além de discutir os aspectos que permeiam o processo de desenvolvimento de *software* e a especificação de requisitos e os *roadmaps* discutir também esta metodologia.

Para Dresch, Lacerda e Júnior (2020), o *Design Science Research* (DSR) é um método que fundamenta e operacionaliza a condução da pesquisa quando o objetivo a ser alcançado é um artefato ou uma prescrição. Como método de pesquisa orientado à solução de problemas, a *Design Science Research* busca, a partir do entendimento do problema, construir e avaliar artefatos que permitam transformar situações, alterando suas condições para estados melhores ou desejáveis. É utilizada nas pesquisas como forma de diminuir o distanciamento entre teoria e prática.

O *Design Science Research* é composto de cinco fases, sendo a avaliação do artefato gerado uma atividade chave desta do uso desta metodologia, isto irá fornecer feedback para melhorias no produto gerado e garante o rigor da pesquisa (VENABLE, PRIES-HEJE E BASKERVILLE, 2016). A Figura 7 apresenta as fases de desenvolvimento do DSR.



Fonte: Adaptado de Takeda *et al.* (1990).

Como fase inicial do DSR, é feita a identificação do problema a ser solucionado e o que este envolve (ambiente, atores e entre outros). A fase de sugestão apresenta os requisitos e critérios para a construção do artefato, e como fases seguintes é realizado o desenvolvimento do artefato e avaliação do mesmo até a sua conclusão. O Quadro 3 mostra os pontos a explicitar ao longo das etapas de condução de uma DSR.

Quadro 3 – Desenvolvimento de uma DSR

Etapas de Condução	Saídas da DSR	Pontos a Explicitar
Conscientização	Proposta	<ul style="list-style-type: none"> • Evidenciar a situação problemática. • Explicitar o ambiente externo e seus principais pontos de interação com o artefato. • Explicitar as métricas e os critérios para a aceitação da solução do artefato (quando não for possível a obtenção de uma solução ótima). • Explicitar os atores que se interessam pelo artefato. • Explicitar as classes de problemas, os artefatos existentes e suas limitações.
Sugestão	Tentativa	<ul style="list-style-type: none"> • Explicitar as premissas e requisitos para a construção do artefato. • Registrar todas as tentativas de desenvolvimento do artefato. • Registrar as razões que fundamentaram a exclusão da tentativa de artefato do desenvolvimento. • Verificar possíveis implicações éticas da aplicação do artefato.
Desenvolvimento	Artefato	<ul style="list-style-type: none"> • Justificar a escolha das ferramentas para o desenvolvimento do artefato. • Explicitar os componentes do artefato e as relações causais que geram o efeito desejado para que o artefato realize seus objetivos. • Explicitar as formas pelas quais o artefato pode ser testado.
Avaliação	Medidas de desempenho	<ul style="list-style-type: none"> • Explicitar, em detalhes, os mecanismos de avaliação do artefato. • Evidenciar os resultados do artefato em relação às métricas inicialmente projetadas. • No caso de avaliações qualitativas do artefato, explicitar as partes envolvidas e as limitações de viés. • Evidenciar o que funcionou como o previsto e os ajustes necessários no artefato.
Conclusão	Resultados	<ul style="list-style-type: none"> • Sintetizar as principais aprendizagens em todas as fases do projeto. • Justificar a contribuição do trabalho para a classe de problemas em questão.

Fonte: Adaptado de Lacerda *et al.* (2013).

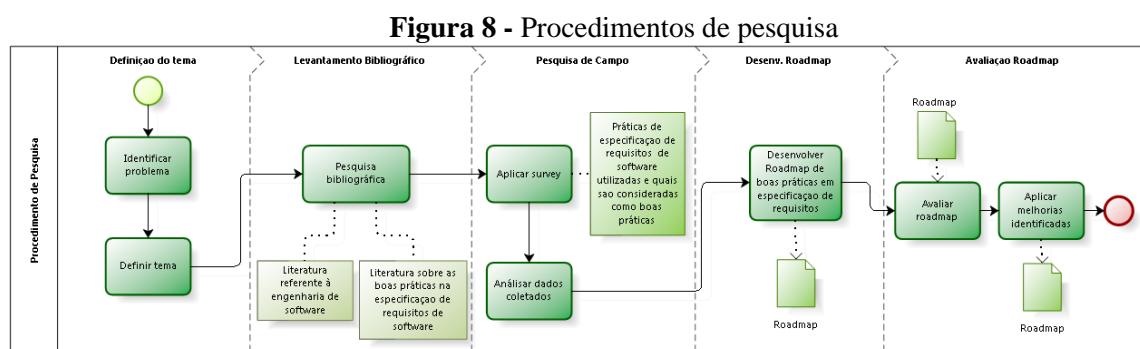
4 METODOLOGIA

A pesquisa que foi desenvolvida neste trabalho se enquadra como uma pesquisa exploratória, pois permitiu a exploração e compreensão do tema, bem como, a observação e descrição de seus fenômenos (VIEIRA, 2002). A pesquisa também se enquadra como aplicada, pois, segundo Cauchick-Miguel (2018), o propósito desta foi o desenvolvimento de um artefato e tecnológica devido a esse artefato produzido precisar satisfazer os critérios de factibilidade, confiabilidade e eficiência (CUPANI, 2006).

Inicialmente, foram empregados procedimentos metodológicos como análise bibliográfica de artigos científicos e livros, a fim de verificar os conceitos que envolvem os requisitos de *software*, como engenharia de requisitos, documentação dos requisitos frente as diferentes metodologias de desenvolvimento de *software*, melhores práticas em requisitos e entre outros. Isso possibilitou identificar as dificuldades e problemas existentes e quais alternativas podem ser utilizadas para que se minimizem as possíveis falhas na execução desta atividade.

Diante das informações obtidas por meio da análise bibliográfica, deu-se então início a elaboração de uma *survey* com o objetivo de validar se as melhores práticas em requisitos de *software* que foram citadas pela literatura realmente são utilizadas pelas empresas de TI e por profissionais da área e também para a descoberta de outras práticas. Com a finalização da etapa de aplicação da *survey*, foi realizada a análise dos dados e por fim, o *roadmap* foi desenvolvido, sendo um produto que visa auxiliar as empresas a empregar melhores práticas em requisitos de *software*, como também, melhorar as práticas já em uso.

O procedimento utilizado para essa pesquisa foi composto por cinco etapas, sendo: a) definição do tema; b) levantamento bibliográfico; c) pesquisa de campo; d) desenvolvimento do *roadmap*; e) avaliação *roadmap*. A figura 8 exemplifica como a pesquisa foi conduzida.



Fonte: A autora (2021).

As etapas definidas para a pesquisa são detalhadas nos itens a seguir.

4.1 Definição do Tema

Ao observar as informações apresentados pelo CHAOS Report de 2020, é possível notar o quão desafiador são os projetos de *software*, pois apenas no ano de 2020, cerca de 50% desses projetos falharam. A tabela 1 mostra os dados que foram disponibilizados no CHAOS Report de 2020.

Tabela 1 - Informações CHAOS Report 2020

Método	De Sucesso	Desafiadores	Falhos
Ágil	42%	47%	11%
Tradicional	13%	59%	28%

Fonte: Adaptado de Vitalitychicago (2020).

De acordo com o apresentado na tabela 1, ao analisar o ano de 2020, infere-se que os *softwares* que foram entregues com sucesso, independentemente do método adotado, não chegaram a 50%. Além disso, percebe-se que em ambos os métodos a construção dos *softwares* é desafiadora e que a porcentagem dos *softwares* que falham, ou seja, que simplesmente não são entregues ainda é alta, sendo essa taxa ainda maior nos *softwares* que são conduzidos pelos métodos tradicionais de desenvolvimento de *software*.

Muitas dessas falhas podem ser apresentadas já nas fases finais do projeto, porém geralmente são oriundas de erros cometidos no início dos projetos, como erros no planejamento, falta de estrutura para atender a demanda, e em sua grande maioria entendimento superficial ou errado dos requisitos de *software* que devem ser atendidos. Neste contexto, a engenharia de *software* disponibiliza técnicas, ferramentas e metodologias que visam apoiar o desenvolvimento de *software* para que este processo seja organizado e gere bons resultados.

Sendo o tema de requisitos de *software* de grande interesse da pesquisadora por sua experiência profissional, o tema foi definido a fim de identificar as melhores práticas em requisitos para times de desenvolvimento de sistemas de informação. Com o propósito de construir um produto que reúna tanto as melhores práticas citadas pela literatura quanto as apontadas pela pesquisa de campo com profissionais da área de TI.

4.2 Levantamento Bibliográfico

Com a definição do tema, buscou-se na literatura o embasamento teórico para a realização deste estudo. Para isto, foi realizada uma pesquisa bibliométrica com o auxílio da ferramenta “*Publish or Perish*” nas bases de dados definidas, sendo Google Acadêmico, Scopus e *Web of Science*.

Para verificar as abordagens mais recentes sobre o tema, o período pesquisado foi compreendido entre os anos de 2015 e 2020 e utilizou-se as seguintes palavras-chave “Boas práticas”, “Especificação de requisitos” e “*Software*”, combinadas também com os mesmos termos em inglês “*Best practices*” e “*Requirements specification*” e também “*Good practices*” e “*Requirements specification*”.

4.3 Pesquisa de Campo

Com a identificação das práticas em requisitos de *software* na literatura, foi desenvolvida uma *survey* com o objetivo de se avaliar quais dessas práticas são mais utilizadas no meio empresarial, como também, a descoberta de demais práticas que podem ser aplicadas para esta atividade.

A investigação realizada na *survey* foi conduzida por meio de coleta de dados e/ou informações com o intuito de avaliar o comportamento das pessoas e/ou dos ambientes em que elas se encontram. A partir da coleta e análise dos dados, o pesquisador pode obter conclusões acerca do fenômeno ou da população em estudo (DRESCH, LACERDA E JÚNIOR, 2020).

Para analisar as práticas em requisitos de *software*, o questionário foi composto por oito questões entre perguntas abertas e fechadas que inicialmente contextualizavam a atuação da organização em que o participante trabalha (como tamanho da empresa, tipo de produto que desenvolve e entre outros) e quais as práticas são utilizadas.

Os questionários foram direcionados a participantes que atuam em empresas de TI e que possuem suas funções relacionadas aos requisitos de *software*. O questionário foi disponibilizado via *Google Forms*, e divulgado por meio das redes sociais (*WhatsApp*, *Instagram*, *LinkedIn*, dentre outros). A aplicação dos questionários iniciou em fevereiro de 2022 e encerrou em abril de 2022.

Como a pesquisa de campo foi aplicada à seres humanos, foram observadas as diretrizes da Comissão Nacional de Ética em Pesquisa (CONEP), que está diretamente ligada ao Conselho Nacional de Saúde (CNS). Essa comissão é a responsável pela avaliação dos aspectos éticos

das pesquisas que envolvem seres humanos no Brasil. Em cumprimento à sua missão, a Comissão elabora e atualiza as diretrizes e normas para a proteção dos participantes de pesquisas (CNS, 2022).

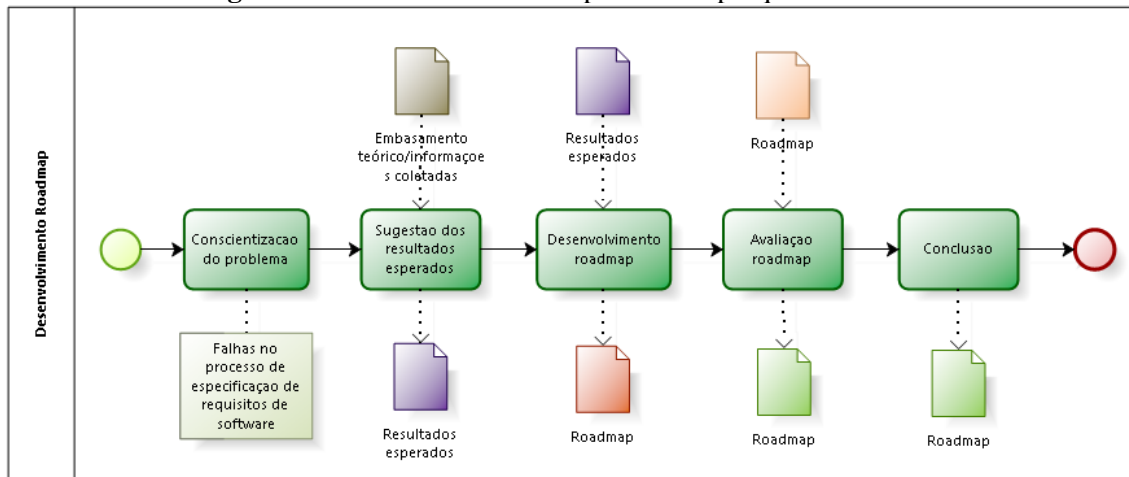
Ainda segundo a CNS (2022), para que a pesquisa seja ética, é necessário respeitar o participante em sua dignidade e autonomia, reconhecendo sua vulnerabilidade, assegurando sua vontade de contribuir e permanecer, ou não na pesquisa, por intermédio da manifestação expressa, livre e esclarecida. Ponderar entre riscos e benefícios, tanto conhecidos como potenciais, individuais ou coletivos comprometendo-se com o máximo de benefícios e o mínimo de danos e riscos. Garantir que danos sejam evitados e ter relevância social, o que garante a igual consideração dos interesses envolvidos, não perdendo o sentido de sua destinação sócio humanitária.

Vale ressaltar que na pesquisa realizada neste trabalho foi incluído no formulário de preenchimento o termo de consentimento livre e esclarecido que possibilitou aos sujeitos da pesquisa o mais amplo esclarecimento sobre a investigação a ser realizada de forma que a participação (ou não) fosse efetivamente livre e consciente, além das diretrizes do CNS, também fossem observadas as orientações da comissão de ética em pesquisa do mestrado do Centro Paula Souza.

4.4 Desenvolvimento *Roadmap*

O resultado desta pesquisa foi um produto tecnológico não patenteável. Sendo assim, foi desenvolvido um *roadmap* que permite conduzir as empresas de desenvolvimento de *software* a adotar melhores práticas nas atividades de requisitos de *software*. A construção do *roadmap* foi orientada pela metodologia *Design Science Research* (DSR). Os itens a seguir detalham como o *roadmap* foi baseado nas etapas do *Design Science Research*.

Para esta pesquisa, optou-se pela utilização do DSR pelo fato de gerar um produto final e ter a necessidade de testá-lo antes da sua disponibilização para uso nas empresas. A utilização deste produto nas empresas pretende solucionar problemas que são atrelados as atividades de requisitos de *software* de maneira simplificada com o emprego de melhores práticas em requisitos que são difundidos pela engenharia de *software*. A Figura 9 ilustra as fases de desenvolvimento do produto da pesquisa orientado pelo método DSR.

Figura 9 - Desenvolvimento do produto da pesquisa com DSR

Fonte: A autora (2021).

Na primeira fase de construção do *roadmap* proposto, foi realizada a conscientização do problema, que são as falhas que ocorrem no processo de especificação de requisitos de *software*. Posteriormente, foram coletadas e analisadas as melhores práticas requisitos de *software* que foram sugeridas pela literatura e também pela aplicação da *survey*, o que possibilitou identificar possíveis soluções para esses problemas. Com base nessas informações, o *roadmap* foi desenvolvido e seu uso foi avaliado por uma empresa de desenvolvimento de *software*.

4.5 Avaliação do Roadmap

O uso do *roadmap* foi avaliado por uma organização que desenvolve soluções web e mobile e está no mercado há mais de dez anos. A avaliação foi realizada de maneira que garantiu que os dados da empresa e de seus colaboradores não fossem expostos.

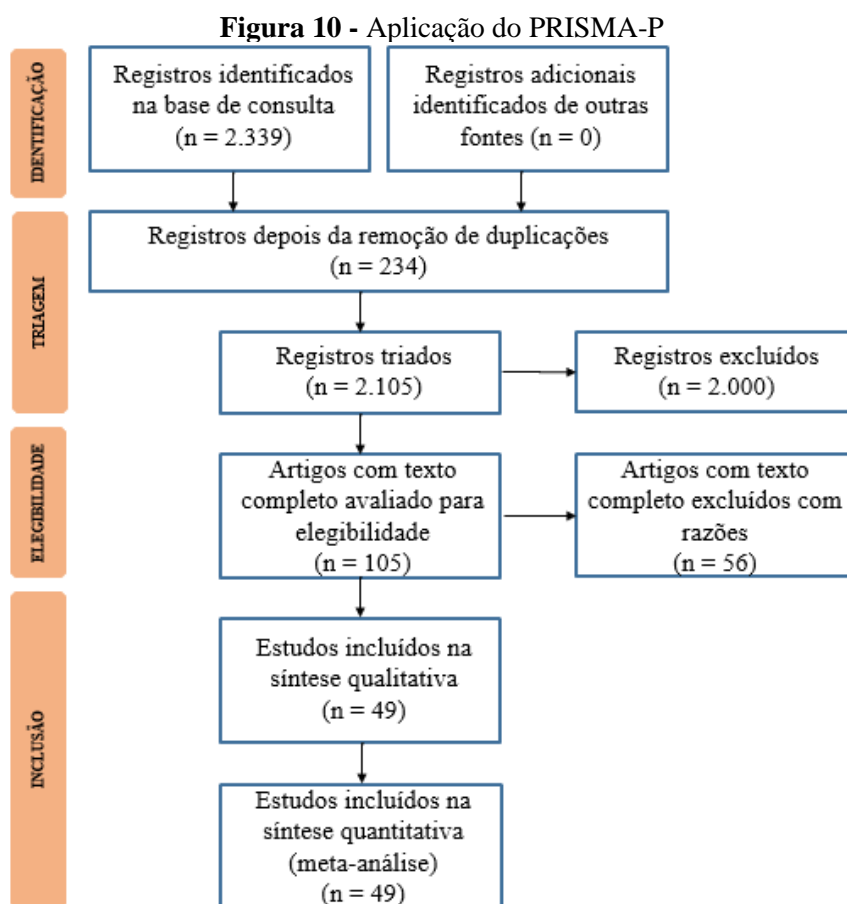
5 RESULTADOS E DISCUSSÃO

Os resultados discussão da pesquisa são apresentadas nos itens seguintes.

5.1 Levantamento bibliográfico

A realização do levantamento bibliográfico permitiu o entendimento das dificuldades que ocorrem nas atividades relacionadas a requisitos de *software*, bem como, a identificação de melhores práticas que podem ser aplicadas para aumento da qualidade e a resolução dos problemas desta área.

O protocolo PRISMA-P (*Preferred Reporting Items for Systematic Review and Meta-Analysis Protocols*) foi utilizado para análise do material retornado na pesquisa. Este protocolo garante que a revisão das publicações seja cuidadosamente planejada, e o que foi planejado, seja documentado antes do início da revisão, resultado em uma investigação íntegra e transparente (MOHER *et al.*, 2016). A figura 10 apresenta como o protocolo PRISMA-P foi aplicado neste estudo.



Fonte: A autora (2021).

Como é possível verificar na Figura 10, foram identificadas um total de 2.339 publicações nas bases de consulta durante a fase de identificação do protocolo PRISMA-P. Em sua fase seguinte, para a triagem, o primeiro critério utilizado foi:

- a) Exclusão de documentos duplicados;

Com a exclusão dos documentos duplicados, restaram 2.105 publicações que foram novamente triadas utilizando-se os seguintes critérios:

- b) Exclusão de materiais como monografias, dissertações, teses, citações e artigos não abertos;
- c) Exclusão de publicações em idiomas diferente de português, espanhol e inglês.

Um total de 2.000 publicações atendiam algum critério de exclusão e por esse motivo não prosseguiram na análise. Das 2.339 publicações identificadas inicialmente, restaram 105 após a fase de triagem.

Na fase de elegibilidade, foi realizada a leitura dos resumos/abstracts dos artigos, a fim de verificar o alinhamento com o tema da pesquisa, sendo excluídas 56 publicações. Já na fase de inclusão, última fase do protocolo PRISMA-P, os 49 registros restantes foram lidos na íntegra. O Quadro 4 apresenta as publicações que foram selecionadas para leitura na íntegra.

Quadro 4 – Publicações selecionadas

	Título do Artigo	Autor(es)
1	Factors affecting requirements engineering in agile <i>software</i> development: A systematic analysis	Abid <i>et al.</i> (2020)
2	Requirement engineering for non-functional requirements	Alashqar, Elfetouh e El-Bakry (2015)
3	Game development <i>software</i> engineering process life cycle: a systematic review	Aleem, Capretz e Ahmed (2016)
4	High-Level representation of time in diagrammatic specification	Al-Fedaghi (2015)
5	Aplicação e análise de processo de desenvolvimento de <i>software</i> : um estudo de caso no GPES-IFPB	Almeida <i>et al.</i> (2018)
6	Evaluating <i>software</i> engineering practices in Palestine	Alnajjar e Naser (2015)
7	Cultural influences on requirements engineering process in the context of Saudi Arabia	Alsanoosy, Spichkova e Harland (2018)
8	Collaborative education: building a skilled <i>software</i> validation and verification community	Archarya, Manohar e Wu (2018)
9	Elaboration of <i>software</i> requirements documents by means of patterns instantiation	Barcelos e Penteado (2017).
10	Proposta de modelagem de processos para apoiar a implementação do processo de gestão de requisitos do MPS.BR	Barreto <i>et al.</i> (2020).
11	Approach to the analysis of <i>software</i> requirements specification on its structure correctness	Boyarchuk <i>et al.</i> (2020).
12	Mapeamento dos processos e artefatos da engenharia de requisitos para o eXtreme programming	Cardoso, Silva e Siqueira (2018)
13	Are the expected benefits of requirements reuse hampered by distance? An experiment	Carrillo de Gea <i>et al.</i> (2016).
14	Engenharia de requisitos: Boas práticas para elicitação de	Carvalho (2018).

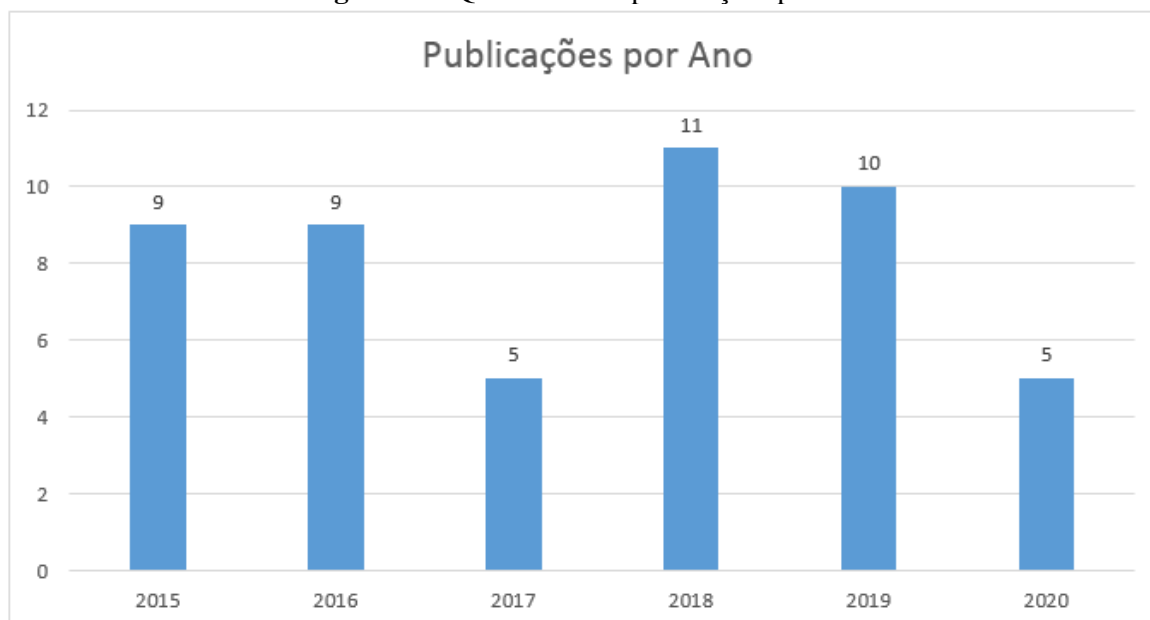
	Título do Artigo	Autor(es)
	requisitos	
15	Utilização da técnica de desenvolvimento orientado por comportamento (BDD) no levantamento de requisitos	Cezerino e Nascimento (2016).
16	Técnicas para o levantamento de requisitos: uma proposta para a obtenção de resultados mais precisos	Dourado e Rodrigues (2015).
17	Survey on requirement elicitation techniques: it effect on <i>software engineering</i>	Elijah <i>et al.</i> (2017).
18	Maintaining requirements using Web Usage Data	Garcia e Paiva (2016).
19	Experiential learning approach for <i>software engineering</i> courses at higher education level	Gonzalez-Huerta <i>et al.</i> (2020).
20	Gestão do escopo para o projeto Biblioteca de reuso de requisitos de <i>software</i> para diferentes domínios	Gualhano, Silva e Vasconcelos (2019).
21	Assessing the state of <i>software</i> in a large enterprise: a twelve year retrospective	Hackbarth <i>et al.</i> (2015).
22	A domain-specific language to manage requirements traceability	Haidrar <i>et al.</i> (2018).
23	Best practices for requirements identification, specification, and validation to guide <i>software</i> implementation and maintenance processes for applications in a electricity supply company	Herrera e Sanz (2015).
24	Experience in applying of ISO 29110 to agile <i>software</i> development	Jirapanthong (2019).
25	Adaptive Concept Map Approach for <i>software</i> requirements validation	Khedr, Ahmed e Kholeif (2019).
26	Caracterização de um ambiente visual para apoiar as cerimônias do SCRUM	Kussunga, Ribeiro e Santos (2019).
27	Project scope management: A strategy oriented to the requirements engineering	Lampa <i>et al.</i> (2017).
28	An MBSE approach to pass from requirements to functional architecture	Lemazurier, Chapurlat e Grossetête (2017).
29	Engenharia de requisitos na resolução de problemas da comunidade: Lições aprendidas	Lima <i>et al.</i> (2019).
30	A utilização de engenharia de domínio para descoberta de padrões de análise em sistemas de controle de mudanças	Lima e Silva (2015).
31	Extending agile methods with requirements engineering	Liu, Jung e Sun (2020).
32	From requirements to automated acceptance tests of interactive apps: an integrated model-based testing approach	Maciel, Paiva e Silva (2019).
33	Engenharia de requisitos em projetos ágeis: uma revisão sistemática da literatura	Medeiros <i>et al.</i> (2015).
34	A new consistency validation approach to enhance the quality of functional security requirements for secure <i>software</i>	Mustafa e Kamalrudin (2018).
35	Empirical study of requirements engineering in cross domain development	Nilsson <i>et al.</i> (2018).
36	Concept based students data retrieval and repository system	Orinda e Ngugi (2018).
37	Karlskrona manifesto: <i>software</i> requirement engineering good practices	Oyedeji e Penzenstadler (2018).
38	Something is Rotten in the state of documenting Simulink models	Pantelic <i>et al.</i> (2019).
39	Approach of agile methodologies in the development of web-based <i>software</i>	Ríos e Pedreira-Souto (2019).
40	A model-based engineering methodology for requirements and formal design of embedded and real-time systems	Ribeiro <i>et al.</i> (2017).
41	Applying role-playing game in <i>software</i> development subjects	Rueda <i>et al.</i> (2016).
42	Levantamento de requisitos no desenvolvimento de jogos	Santos e Gomes (2019).
43	Getting grip on security requirements elicitation by structuring	Schmitt e Liggesmeyer

	Título do Artigo	Autor(es)
	and reusing security requirements sources	(2015).
44	A framework for teaching distributed requirements engineering in Latin American Universities	Sevilla <i>et al.</i> (2018).
45	Uma análise quantitativa da utilização de técnicas de validação de requisitos no mercado de TI em Fortaleza	Souza <i>et al.</i> (2016).
46	Aplicação da etnografia no contexto de fábrica de <i>software</i> na perspectiva da engenharia de requisitos	Souza <i>et al.</i> (2019).
47	Ingeniería de Requisitos: de la especificación de requisitos de <i>software</i> al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de <i>software</i> de la ciudad de Pereira	Toro e Peláez (2016).
48	Applying secure <i>software</i> engineering (SSE) practices to critical space system infrastructure development	Wiemer e Fisher (2016).
49	Considerations for effective requirements analysis in offshore <i>software</i> development projects: lessons from multi-method research	Yadav <i>et al.</i> (2016).

Fonte: A autora (2021).

Ao analisar a quantidade de publicações sobre o tema por ano, é possível observar que o número de publicações teve um pequeno aumento nos anos de 2018 e 2019. A figura 11 ilustra a quantidade de artigos por ano de publicação.

Figura 11 - Quantidade de publicações por ano



Fonte: A autora (2021).

A leitura dos 49 registros alinhados ao tema da pesquisa contribuiu para identificação das práticas em requisitos de *software* que foram mais citadas pelos autores nos últimos anos. Foram identificadas diversas práticas que podem ser empregadas em várias das tarefas abordadas pela engenharia de requisitos, a fim de aumentar a qualidade durante a sua execução.

A seguir, são detalhadas as 27 práticas e os trabalhos relacionados em que são citadas.

5.2 Práticas em Requisitos de *Software* identificadas na literatura

As 27 boas práticas encontradas foram:

- Análise de personas;
- Aplicação de questionários;
- Chuva de ideias (Brainstorming);
- Controle de mudanças/Versionamento de requisitos;
- Criação de casos de teste/Testes de aceitação;
- Criação de casos de uso/Artefatos criados com a UML;
- Documentação de requisitos não funcionais;
- Entrevistas;
- Escrita de requisitos em forma de histórias de usuário;
- Etnografia;
- Grupo focal;
- Histórias de usuário são fornecidas pelo cliente;
- Observação de um sistema existente;
- Priorização dos requisitos;
- Prototipação;
- Rastreabilidade de requisitos;
- Realização de apresentações/workshops;
- Repositório para os artefatos de especificação de requisitos de *software*;
- Reuso de artefatos de especificação de requisitos de *software*;
- Técnica INVEST;
- Técnica JAD;
- Técnica SMART;
- Trawling;
- Uso de *softwares* para preparo da documentação do sistema/gestão dos requisitos;
- Uso de *templates* para a documentação;
- Validação dos requisitos;
- Web Site Design.

Para cada uma das 27 práticas identificadas, são apresentadas a seguir uma breve descrição e as referências da pesquisa bibliográfica que as citam.

a) Análise de personas

A identificação das personas é uma etapa importante, pois elas representam um cliente típico que fará uso do sistema a ser desenvolvido. As personas apresentam as características comuns dos potenciais usuários da aplicação, seus comportamentos e detalhes, desta forma, é possível criar soluções que focam nas necessidades desses usuários.

Trabalhos relacionados: Cardoso, Silva e Siqueira (2018) e Medeiros *et al.* (2015).

b) Aplicação de questionários

Os questionários são compostos de perguntas que devem ser respondidas pelas diferentes partes envolvidas, essa técnica permite envolver um grande número de pessoas com um baixo custo de aplicação. Para os requisitos, essas perguntas devem tentar solucionar possíveis dúvidas, bem como, possibilitar a identificação de novos requisitos.

Trabalhos relacionados: Alnajjar e Naser (2015); Carvalho (2018); Elijah *et al.* (2017); Herrera e Sanz (2015); Lima *et al.* (2019); Medeiros *et al.* (2015); Oyedeji e Penzenstadler (2018); Sevilla *et al.* (2018); Souza *et al.* (2019) e Toro e Peláez (2016).

c) Chuva de ideias (*Brainstorming*)

O termo *brainstorming*, ou chuva de ideias em português, é uma técnica em que os participantes de diferentes grupos de partes interessadas se envolvem em discussões informais para gerar rapidamente o maior número possível de ideias. Aplicada a requisitos, essa técnica permite apresentar as necessidades dos clientes e coletar possíveis requisitos para o sistema que irão atender a essas necessidades.

Trabalhos relacionados: Abid *et al.* (2020); Barreto *et al.* (2020); Elijah *et al.* (2017); Herrera e Sanz (2015); Medeiros *et al.* (2015); Santos e Gomes (2019); Sevilla *et al.* (2018); Souza, *et al.* (2019) e Toro e Peláez (2016).

d) Controle de mudanças/Versionamento de requisitos

É muito comum que, ao iniciar uma aplicação, os requisitos ainda não estejam completos ou gerem dúvidas para as partes envolvidas; por essa razão, é fundamental que os

requisitos tanto funcionais quanto os não funcionais sejam gerenciados desde o momento da sua concepção. Ou seja, devem ser documentados em sua primeira versão e conforme o entendimento do requisitos avança, as alterações neste devem ser versionadas, permitindo que um histórico dos requisitos seja mantido e que seja possível se identificar facilmente as alterações que foram realizadas ao longo do tempo.

Trabalhos relacionados: Alsanoosy, Spichkova e Harland (2018); Garcia e Paiva (2016); Hackbarth *et al.* (2015); Lampa *et al.* (2017); Nilsson *et al.* (2018); Oyedeji e Penzenstadler (2018); Sommerville (2018) e Yadav *et al.* (2016).

e) Criação de casos de teste/Testes de aceitação

Sendo o termo testes de aceitação mais empregado na cultura ágil para a verificação das histórias de usuário, ambas são práticas utilizadas para se verificar se a aplicação atende às expectativas dos usuários finais e se o requisito está apto para entrar em funcionamento. Devem conter o fluxo da funcionalidade, como essa deverá ser executada e quais são os resultados esperados após o seu processamento.

Trabalhos relacionados: Abid *et al.* (2020); Cardoso, Silva e Siqueira (2018); Cezerino e Nascimento (2016); Herrera e Sanz (2015); Jirapanthong (2019); Liu, Jung e Sun (2020); Maciel, Paiva e Silva (2019); Nilsson *et al.* (2018); Souza *et al.* (2016) e Toro e Peláez (2016).

f) Criação de casos de uso/Artefatos criados com a UML

O detalhamento dos casos de uso permite especificar um conjunto de passos que o sistema deverá executar e que irá produzir um resultado que agrega valor aos atores envolvidos, o que permite expressar com simplicidade os requisitos do sistema e a relação entre atores do sistema e as funcionalidades.

Muitos autores reforçam a importância da utilização da *Unified Modeling Language* (UML) para a modelagem das estruturas que irão compor os sistemas orientados a objeto, pois a padronização da especificação fornecida pela UML simplifica o entendimento dos comportamentos do sistema para os mais diversos profissionais e sua manutenção se torna menos custosa.

Trabalhos relacionados: Aleem, Capretz e Ahmed (2016); Al-Fedaghi (2015); Almeida *et al.* (2018); Barreto *et al.* (2020); Carvalho (2018); Dourado e Rodrigues (2015); Herrera e Sanz

(2015); Kussunga, Ribeiro e Santos (2019); Lima *et al.* (2019); Medeiros *et al.* (2015); Ríos e Pedreira-Souto (2019); Rueda *et al.* (2016); Pressman e Maxim (2016); Souza *et al.* (2019); Sommerville (2018) e Toro e Peláez (2016).

g) Documentação de requisitos não funcionais

Enquanto os requisitos funcionais descrevem o que o sistema fará, os requisitos não funcionais informam como o sistema fará, por esse motivo possuem papel fundamental no desenvolvimento do *software*. São requisitos relacionados a atributos de qualidade que o sistema deve ter, restrições, desempenho e entre outros. Deve-se garantir que esses requisitos são sempre coletados e documentados para a aplicação.

Trabalhos relacionados: Pressman e Maxim (2016) e Sommerville (2018).

h) Entrevistas

Prática muito utilizada na fase de levantamento de dados, é simples de aplicar e gera bons resultados. De preferência, é conduzida por um analista experiente que possui um conhecimento genérico sobre o domínio do problema, ele pode iniciar fazendo perguntas e então o domínio do problema passa a ser discutido com as várias partes interessadas com o objetivo de se desenvolver a compreensão dos requisitos que devem ser atendidos pela aplicação que será desenvolvida.

Trabalhos relacionados: Abid *et al.* (2020); Alnajjar e Naser (2015); Alsanoosy, Spichkova e Harland (2018); Barreto *et al.* (2020); Carrillo de Gea *et al.* (2016); Carvalho (2018); Dourado e Rodrigues (2015); Elijah *et al.* (2017); Herrera e Sanz (2015); Lima *et al.* (2019); Medeiros *et al.* (2015); Rueda *et al.* (2016); Sevilla *et al.* (2018); Sommerville (2018); Souza *et al.* (2019); Toro e Peláez (2016) e Yadav *et al.* (2016).

i) Escrita de requisitos em forma de histórias de usuário

Muito utilizada na abordagem ágil, as histórias de usuário são requisitos preliminares e o ponto de partida para que pode ser ampliado para outros requisitos, conforme o entendimento das necessidades dos usuários. No geral, são descritas em cartões e o esforço de cada uma é estimado, o cliente prioriza quais dessas histórias agregam mais valor a seu negócio para que sejam implementadas já na próxima versão que for disponibilizada ao cliente.

Trabalhos relacionados: Abid *et al.* (2020); Alashqar, Elfetouh e El-Bakry (2015); Almeida *et al.* (2018); Cardoso, Silva e Siqueira (2018); Cezerino e Nascimento (2016); Gonzalez-Huerta *et al.* (2020); Herrera e Sanz (2015); Kussunga, Ribeiro e Santos (2019); Medeiros *et al.* (2015); Pressman e Maxim (2016); Sommerville (2018) e Toro e Peláez (2016).

j) Etnografia

A etnografia é uma técnica de observação que analisa os componentes sociais das tarefas desempenhadas em uma organização; sendo assim, um analista participa das atividades habituais dos usuários durante um período de tempo e coleta informações. Essa técnica é eficaz quando a necessidade de um novo sistema é resultado de problemas existentes com processos e procedimentos, também é utilizado na identificação de padrões sociais e relacionamentos complexos entre as partes interessadas e os usuários.

Trabalhos relacionados: Alsanoosy, Spichkova e Harland (2018); Dourado e Rodrigues (2015); Elijah *et al.* (2017); Herrera e Sanz (2015); Lima *et al.* (2019); Oyedeji e Penzenstadler (2018); Sommerville (2018) e Toro e Peláez (2016).

k) Grupo focal

Prática muito aplicada na elicitação de requisitos segundo os autores. O grupo focal possui a função de analisar e levantar *feedbacks* sobre um tópico, dessa forma, é possível identificar as necessidades e percepções dos usuários sobre o *software*. Um moderador lidera as discussões e faz perguntas para que sejam respondidas pelo grupo, sendo essas respostas anotadas e analisadas posteriormente para a obtenção dos requisitos.

Trabalhos relacionados: Alsanoosy, Spichkova e Harland (2018); Medeiros *et al.* (2015) e Toro e Peláez (2016).

l) Histórias de usuário são fornecidas pelo cliente

Os autores defendem que as histórias devem ser simples e permitir que o cliente utilize sua própria linguagem para comunicar o que precisa que a aplicação atenda, de igual forma, é desejável que o cliente escreva também os testes de aceitação que irão validar essas histórias. Caso o time de desenvolvimento perceba que é necessário implementar algum benefício ao sistema ou o cenário não é visível para o cliente, os desenvolvedores podem escrever/complementar essas histórias que foram fornecidas.

Trabalhos relacionados: Cardoso, Silva e Siqueira (2018); Cezerino e Nascimento (2016); Mustafa e Kamalrudin (2018) e Sommerville (2018).

m) Observação de um sistema existente

Com o passar do tempo, muitas aplicações precisam ser melhoradas ou até mesmo substituídas por uma nova. Isto pode ocorrer por diferentes motivos, como por exemplo: a aplicação não suporta mais a demanda atual, o negócio ou a legislação sofreram alterações, a tecnologia precisa ser substituída, entre outros. Neste caso, o sistema anterior é utilizado como base para a coleta dos requisitos que o novo sistema deve ter, realizando as alterações necessárias para o novo sistema.

Trabalhos relacionados: Toro e Peláez (2016).

n) Priorização dos requisitos

É necessário definir a prioridade dos requisitos/histórias de usuário de acordo com a complexidade, estabilidade e importância destes, desta forma, a equipe de desenvolvimento pode focar nas funcionalidades principais da aplicação antecipadamente e então gerar valor para o negócio do cliente.

Trabalhos relacionados: Abid *et al.* (2020); Cezerino e Nascimento (2016); Dourado e Rodrigues (2015); Garcia e Paiva (2016); Gonzalez-Huerta *et al.* (2020); Herrera e Sanz (2015) e Nilsson *et al.* (2018).

o) Prototipação

É uma técnica em que um produto é elaborado parcialmente. No geral, são criadas e apresentadas para as partes interessadas as telas de interface com o usuário (ainda sem a aplicação do código) para que facilite o entendimento e validação dos requisitos, sendo criados com base em requisitos preliminares ou em exemplos de sistemas semelhantes. O uso dessa prática incentiva as partes interessadas a desempenhar um papel ativo no desenvolvimento dos requisitos.

Trabalhos relacionados: Abid *et al.* (2020); Aleem, Capretz e Ahmed (2016); Almeida *et al.* (2018); Alnajjar e Naser (2015); Alsanoosy, Spichkova e Harland (2018); Carvalho (2018); Dourado e Rodrigues (2015); Elijah *et al.* (2017); Herrera e Sanz (2015); Kussunga, Ribeiro e

Santos (2019); Liu, Jung e Sun (2020); Medeiros *et al.* (2015); Rueda *et al.* (2016); Souza *et al.* (2016); Souza, *et al.* (2019); Toro e Peláez (2016) e Yadav *et al.* (2016).

p) Rastreabilidade de requisitos

Os requisitos devem ser rastreados, ou seja, devem ter um acompanhamento desde a sua concepção até a sua implementação e conclusão dos testes, desta forma, caso seja necessária uma modificação no requisito, tem-se a capacidade de analisar todos os impactos que poderão ser gerados pela mudança que foi proposta.

Trabalhos relacionados: Barreto *et al.* (2020); Cardoso, Silva e Siqueira (2018); Carrillo de Gea *et al.* (2016); Dourado e Rodrigues (2015); Garcia e Paiva (2016); Haidrar *et al.* (2018); Jirapanthong (2019); Lampa *et al.* (2017); Nilsson *et al.* (2018); Oyedeji e Penzenstadler (2018); Pantelic *et al.* (2019); Pressman e Maxim (2016); Sommerville (2018) e Yadav *et al.* (2016).

q) Realização de apresentações/workshops

Apontada pelos autores como uma prática muito utilizada no levantamento de requisitos, as apresentações ou a realização de *workshops* trazem uma visão do que deve ser abordado. No geral, são conduzidos por um especialista que apresenta uma situação ou problema que se tem hoje e as soluções propostas para que sejam discutidas com as partes interessadas.

Trabalhos relacionados: Medeiros *et al.* (2015); Oyedeji e Penzenstadler (2018) e Souza *et al.* (2019).

r) Repositório para os artefatos de especificação de requisitos de *software*

Os repositórios são locais para o armazenamento dos artefatos de *software* e são muito utilizados no ambiente de desenvolvimento. Sua utilização também para os artefatos que gerados na especificação de requisitos pode trazer diversos benefícios, como melhorar a organização e gerenciamento desses artefatos e ainda possibilitar o fácil reuso desses artefatos.

Trabalhos relacionados: Orinda e Ngugi (2018) e Oyedeji e Penzenstadler (2018).

s) Reuso de artefatos de especificação de requisitos de *software*

Muitas vezes, os requisitos para um novo sistema são maiores, menores ou iguais aos requisitos de um sistema já existente e podem ser então reaproveitados. O reuso de um requisito já especificado e validado anteriormente gera uma economia de tempo, além de poupar recursos do projeto que teriam que ser alocados novamente para uma atividade que já foi realizada no passado.

Trabalhos relacionados: Barcelos e Penteado (2017); Carrillo de Gea *et al.* (2016); Elijah *et al.* (2017); Gualhano, Silva e Vasconcelos (2019); Hackbarth *et al.* (2015); Lima e Silva (2015); Maciel, Paiva e Silva (2019) e Oyedeji e Penzenstadler (2018); Schmitt e Liggesmeyer (2015).

t) Técnica INVEST

A técnica INVEST é utilizada para melhorar a qualidade da escrita das histórias de usuário. Essa técnica possibilita verificar se a história de usuário é independente (deve ser independente de outras histórias), negociável (deve ser concisa e negociável), valiosa (deve gerar valor para o negócio), estimável (possui um tempo definido para ser concluída), é sucinta ou pequena (deve ser possível sua construção durante a sprint) e testável (deve apresentar critérios para o teste da funcionalidade).

Trabalhos relacionados: Medeiros *et al.* (2015).

u) Técnica JAD

O *Joint Application development* (JAD), em português Desenvolvimento conjunto de aplicativos, é uma técnica muito utilizada para a coleta de requisitos, pois promove cooperação, entendimento e trabalho em grupo entre os usuários desenvolvedores, facilita a visão compartilhada do que o produto de *software* deve ser.

Com a sua utilização, os desenvolvedores ajudam os usuários a formular problemas e explorar soluções. É conduzido como um *brainstorming*, em que as partes interessadas e usuários podem discutir sobre o sistema proposto até que todos os requisitos finais sejam reunidos.

Trabalhos relacionados: Elijah *et al.* (2017); Medeiros *et al.* (2015); Rueda *et al.* (2016) e Santos e Gomes (2019).

v) Técnica SMART

A técnica SMART é muito utilizada para o alcance de metas. Em métodos ágeis, essa técnica pode ser empregada para a verificação da escrita das histórias de usuário. Neste último caso, elas deverão obedecer a cinco critérios, sendo: as histórias de usuário devem ser específicas (devem ser facilmente entendidas por qualquer pessoa), é mensurável (deve ser possível avaliar se seu objetivo foi alcançado ou não), atingível (o objetivo deve ser real), relevante (a funcionalidade que está sendo construída deve ser relevante para o usuário) e temporal (precisa de um prazo definido para ser alcançada).

Trabalhos relacionados: Lemazurier, Chapurlat e Grossetête (2017).

w) *Trawling*

O termo *trawling*, em português pesca de arrasto, é derivado de um método de pesca que arrasta uma rede de pesca pela água a fim de capturar diferentes espécies de peixe de uma só vez. Aplicada a requisitos de *software*, é uma técnica muito utilizada para a elicitación de requisitos, sendo coletado pelo analista os requisitos com diferentes tipos de usuários. Nesta atividade, o analista de requisitos e usuários colaboram para que se reúnam todos os requisitos para a aplicação.

Trabalhos relacionados: Medeiros *et al.* (2015).

x) Uso de *softwares* para o preparo da documentação do sistema/gestão dos requisitos

A utilização de um *software* para a gestão dos requisitos pode trazer muitos benefícios para a empresa, além de possibilitar a rastreabilidade, permite a visibilidade na gestão de cada um dos requisitos. Já os *softwares* destinados à elaboração de documentos propiciam uma maior agilidade ao documentar as aplicações e padronizam todos os artefatos gerados.

Trabalhos relacionados: Jirapanthong (2019); Kussunga, Ribeiro e Santos (2019); Lemazurier, Chapurlat e Grossetête (2017); Medeiros *et al.* (2015); Mustafa e Kamalrudin (2018); Nilsson *et al.* (2018); Pantelic *et al.* (2019); Ribeiro *et al.* (2017) e Rueda *et al.* (2016).

y) Uso de *templates* para a documentação

Os *templates* ou modelos para documentação de requisitos de *software* permitem que os documentos sejam elaborados em um tempo menor, uma vez que já possuem uma estrutura

definida das informações que devem ser inseridas. Além disso, organizam e padronizam os documentos produzidos pela organização.

Trabalhos relacionados: Alsanoosy, Spichkova e Harland (2018); Boyarchuk *et al.* (2020); Pantelic *et al.* (2019); Wiemer e Fisher (2016) e Yadav *et al.* (2016).

z) Validação dos requisitos

O processo de validação de requisitos propõe mostrar que os requisitos atendem às necessidades reais dos usuários ou clientes antes da implementação do sistema, garantem que a especificação de requisitos está consistente e que realmente captura as necessidades e restrições dos usuários, dessa forma, se reduz o custo com correções de erros. Algumas técnicas propostas pelos autores são: revisão dos requisitos, criação de casos de teste e prototipação.

Trabalhos relacionados: Abid *et al.* (2020); Archarya, Manohar e Wu (2018); Alsanoosy, Spichkova e Harland (2018); Carvalho (2018); Hackbarth *et al.* (2015); Herrera e Sanz (2015); Khedr, Ahmed e Kholeif (2019); Lima *et al.* (2019); Liu, Jung e Sun (2020); Maciel, Paiva e Silva (2019); Nilsson *et al.* (2018); Pantelic *et al.* (2019); Ríos e Pedreira-Souto (2019); Souza *et al.* (2016) e Souza *et al.* (2019).

aa) Web Site Design

O Web Site Design é uma abordagem utilizada para o desenvolvimento de aplicações web, permite que sites e aplicativos sejam desenvolvidos de forma sistemática. Possui uma abordagem orientada ao público (usuário), pois utiliza os requisitos dos usuários pretendidos para conduzir o processo de design do site. Além disso, o Web Site Design utiliza o conceito “Classe de público” para se referir a cada público-alvo que possui seus próprios requisitos.

As atividades dessa metodologia são iniciadas com a definição do público-alvo pretendido para o produto que se deseja desenvolver, depois, essas pessoas irão auxiliar na determinação dos requisitos funcionais e dos requisitos de informação. Isto permitirá que os desenvolvedores descrevam a estrutura conceitual do site e modelem de que forma as diferentes classes de público poderão navegar pela aplicação, sendo elaborados também nessa fase os requisitos de usuário.

Logo após, são apresentados os aspectos de implementação e também sobre a especificação do design da página, como o posicionamento das informações e links, até a sua finalização, em que a aplicação que foi construída é finalmente apresentada.

Trabalhos relacionados: Alashqar, Elfetouh e El-Bakry (2015).

5.3 Pesquisa De Campo

Ao finalizar a fase de aplicação da pesquisa de campo, foi possível analisar os dados obtidos e entender o cenário de desenvolvimento e requisitos. No total, 47 pessoas participaram da pesquisa, sendo estes participantes atuantes em empresas de TI. A seguir, são apresentados os dados coletados na *survey*.

As pessoas foram contatadas pelas mídias sociais como *WhatsApp*, *Facebook*, *Instagram* e *LinkedIn*, sendo enviado o link da pesquisa após a confirmação de ser um participante que atua em uma empresa de TI brasileira. Assim, obteve-se um total de 47 participantes até o encerramento do período de aplicação da *survey*.

Vale ressaltar que a *survey*, dada a natureza e número restrito de participantes, não teve como objetivo identificar tendências, mas sim, tão somente, verificar entre os participantes se as práticas levantadas na pesquisa bibliográfica são realmente utilizadas por profissionais brasileiros atuantes em empresas de TI.

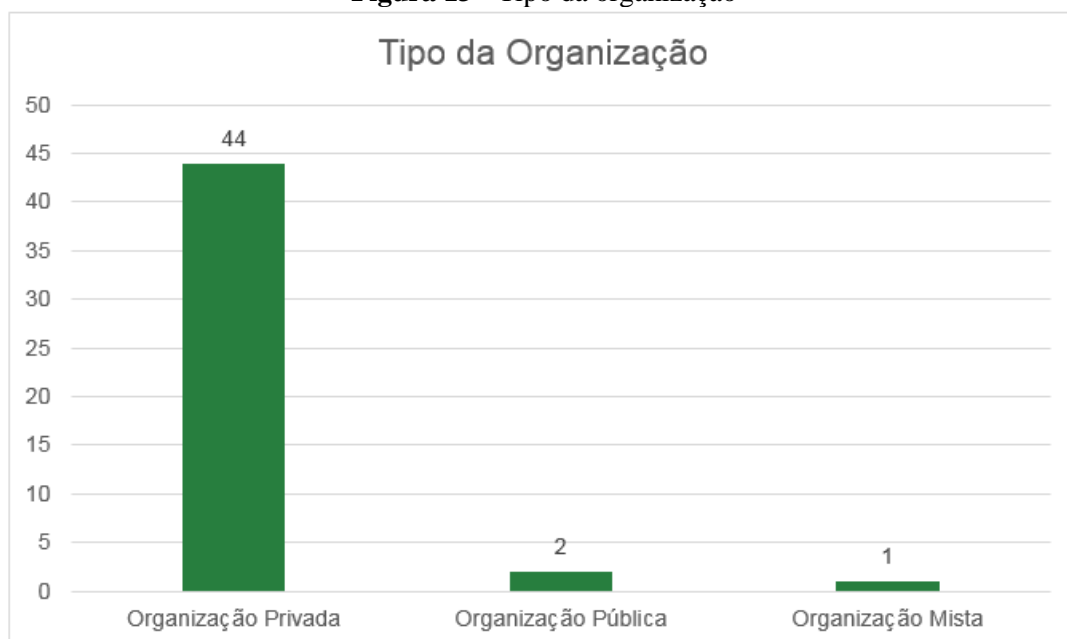
A primeira questão permitiu identificar onde se localizam as empresas em que os participantes da pesquisa atuam, sendo questionado em qual unidade de federação (UF) está localizada a sede da empresa, considerando que no cenário de pandemia da Covid-19 no ano de 2020, muitos profissionais de TI passaram a trabalhar diretamente de suas casas (o que permitiu também a contratação de pessoas que não residem no mesmo local da empresa) e pelo fato de que a empresa pode ter filiais em outros estados do Brasil. A Figura 12 mostra a localização dessas empresas.

Figura 12 - Localização das empresas

Fonte: A autora (2022).

Como é possível notar na Figura 12, a maior parte dos participantes da pesquisa atuam em empresas localizadas na região sudeste e sul do Brasil, sendo que a maior parte dos respondentes atuam no estado de São Paulo e observa-se também uma forte participação do estado do Paraná.

Dentre essas empresas, questionou-se também o tipo das organizações, podendo ser uma organização do tipo privada, pública ou mista, ou seja, é gerenciada por uma parte pública e outra parte privada. A Figura 13 apresenta qual o tipo das organizações em que os participantes atuam.

Figura 13 - Tipo da organização

Fonte: A autora (2022).

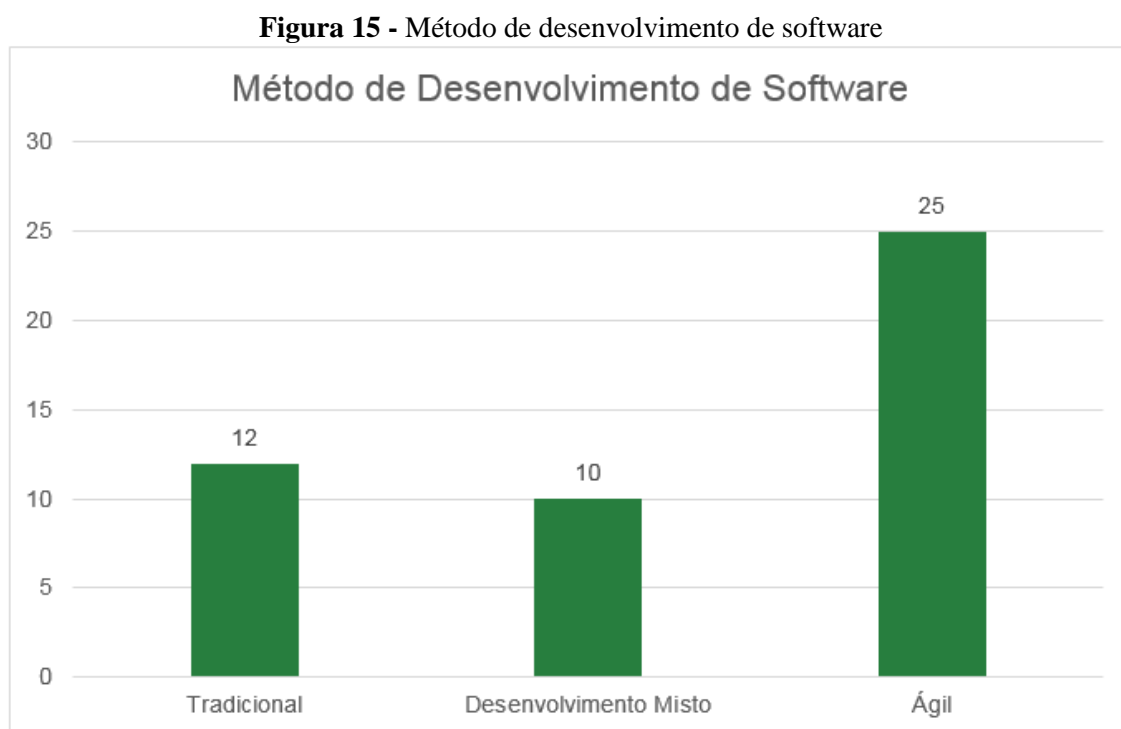
Na Figura 13, nota-se que os participantes em sua maioria atuam em organizações privadas, havendo apenas dois participantes que atuam em organizações públicas e apenas um em uma organização do tipo mista. A questão 3 trata sobre o porte das organizações em que os participantes atuam. A Figura 14 apresenta informações sobre o porte das organizações.

Figura 14 - Porte da organização

Fonte: A autora (2022).

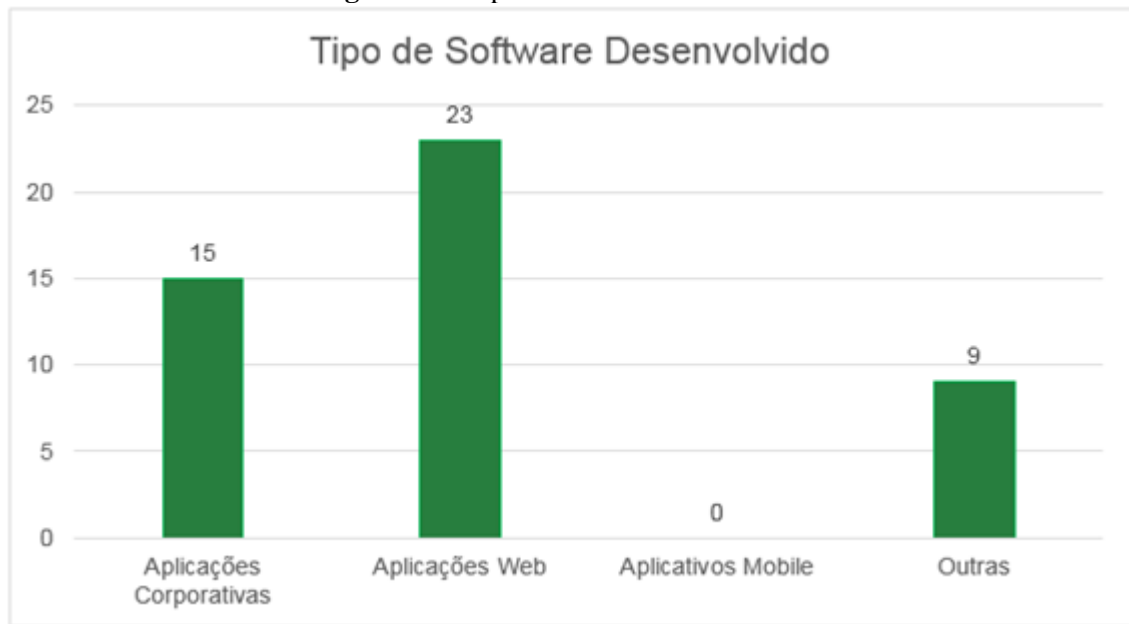
De acordo com a Figura 14, a maioria dos participantes da pesquisa atuam em empresas

de desenvolvimento de *software* de grande porte e empresas de muito grande porte que possuem mais de 500 funcionários (dois tipos de organizações identificadas em todos os estados apontados pelos participantes da pesquisa, outro dado observado é que as empresas de menor porte são apontadas apenas entre as organizações privadas). A Figura 15 mostra quais métodos de desenvolvimento de *software* as empresas utilizam para a criação de seus produtos.



Fonte: A autora (2022).

Na Figura 15, percebe-se que é alto o número de empresas que utilizam os métodos de desenvolvimento tradicionais, porém há um crescimento e interesse na adoção dos métodos ágeis ou mistos, ou seja, que mesclam ambos os métodos (tradicional e ágil). Outros dados observados foram que a maioria das empresas que indicaram a utilização de métodos ágeis é de grande porte (o que pode indicar uma dificuldade das micro e pequenas empresas em implementar o ágil), as organizações públicas utilizam o desenvolvimento misto e os métodos tradicionais não foram apontados pela região Norte e Centro-Oeste do Brasil. A Figura 16 apresenta os tipos de *software* que são desenvolvidos pelas empresas em que os participantes da pesquisa atuam.

Figura 16 - Tipo de software desenvolvido

Fonte: A autora (2022).

É possível verificar, na Figura 16, que os tipos de *software* que são produzidos pelas empresas em que os participantes da pesquisa atuam, são, no geral, aplicações corporativas e web, dentro da amostra coletada, não foi identificado o desenvolvimento mobile. As aplicações corporativas foram identificadas apenas nas organizações privadas, em sua maioria de grande porte, o que também se relaciona a um frequente uso dos métodos ágeis. A Figura 17 mostra as funções dos participantes da pesquisa na organização.

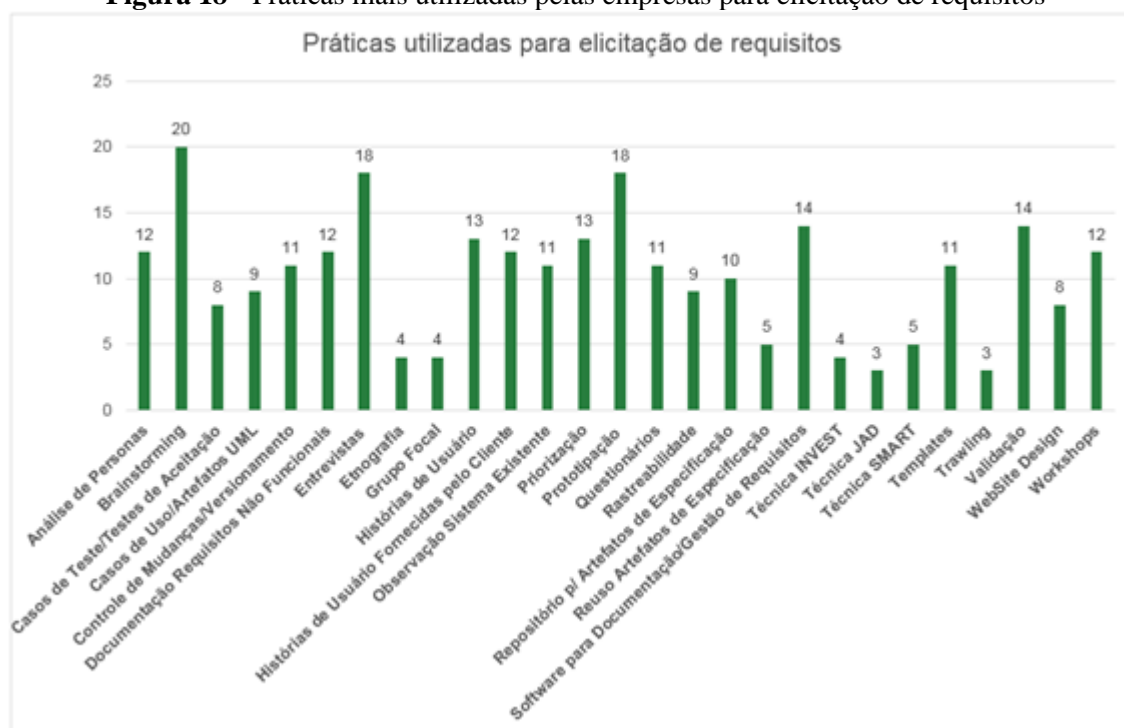
Figura 17 - Funções dos participantes da pesquisa na organização

Fonte: A autora (2022).

Na Figura 17, observa-se que a maioria dos participantes da pesquisa exercem as seguintes funções (questão que permite mais de uma seleção na resposta): analista de sistemas, desenvolvedor e gerente de projetos. Percebeu-se também que, no geral, os analistas de sistemas desempenham mais de uma função nas organizações.

Após as questões gerais com relação aos dados demográficos da pesquisa, questionou-se sobre as práticas em requisitos de *software* que são utilizadas pelas organizações em que os participantes atuam. A Figura 18 ilustra as práticas em requisitos de *software* que são utilizadas pelas empresas em que os participantes da pesquisa trabalham.

Figura 18 - Práticas mais utilizadas pelas empresas para elicitação de requisitos



Fonte: A autora (2022).

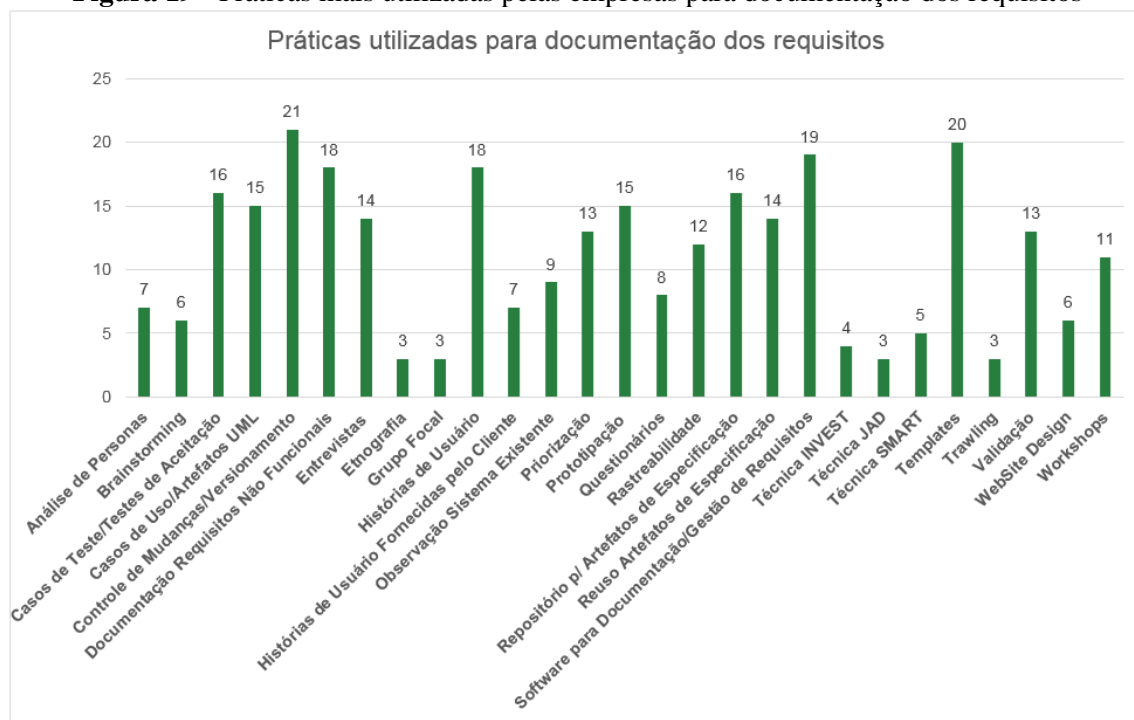
Como é possível visualizar na Figura 18, para a elicitação dos requisitos de *software*, as práticas mais citadas e utilizadas pelas empresas são *brainstorming*, entrevista, utilização de histórias de usuário, priorização, prototipação, uso de *software* para documentação/gestão de requisitos e validação.

Observa-se que, na fase de elicitação é predominante o uso de práticas simples e de fácil domínio para dos analistas. Algumas outras práticas, aplicadas em menor escala pelas empresas dos entrevistados, podem ser muito interessantes quando aplicadas na fase de elicitação como a coleta de requisitos não funcionais, etnografia e questionários.

Demais práticas como casos de teste, criação de casos de uso e priorização podem não ter uma melhor aplicação nesta fase, pois ainda não se tem uma definição mais completa dos

requisitos desejados. A Figura 19 detalha as práticas que são utilizadas para a documentação/especificação destes requisitos.

Figura 19 - Práticas mais utilizadas pelas empresas para documentação dos requisitos



Fonte: A autora (2022).

A Figura 19 ilustra que para documentar os requisitos de *software* as práticas mais utilizadas pelas empresas dos respondentes são a escrita dos testes de aceitação, criação de artefatos UML, versionamento de requisitos, documentação dos requisitos não funcionais, entrevistas, histórias de usuário, prototipação, utilização de repositório para artefatos de especificação, utilização de *softwares* para documentação de requisitos e o uso de *templates*.

Para a fase de documentação de requisitos, também são comumente utilizadas as práticas de entrevista/workshops para documentar requisitos em conjunto com o cliente, documentar uma aplicação com base em um sistema existente e a criação de protótipos. Apesar de pouco praticadas pelas empresas dos entrevistados, as práticas INVEST, JAD e SMART são muito importantes, principalmente em práticas ágeis, pois validam se o requisito/história de usuário é curto, estimável e testável, por exemplo.

É importante destacar a preocupação das empresas em reutilizar artefatos de especificação e consequentemente utilizar repositórios para este fim, isso contribui para que se ganhe agilidade e assertividade nas atividades de requisitos. A Figura 20 mostra as práticas mais aplicadas para a validação dos requisitos de *software* dos respondentes.

Figura 20 - Práticas mais utilizadas pelas empresas para validação dos requisitos

Fonte: A autora (2022).

A Figura 20 apresenta que, para validação dos requisitos de *software*, as práticas mais adotadas pelas organizações dos respondentes são: validação dos casos de teste/testes de aceitação, controle de mudanças dos requisitos, e prototipação. É fundamental que nas validações com o cliente, sejam registradas todas as mudanças em requisitos.

Outras técnicas que podem ser aplicadas para a validação de requisitos são as entrevistas/workshops, questionários e as técnicas INVEST e SMART. Apesar de serem apontadas pelos participantes, as práticas análise de personas, etnografia, grupo focal, JAD e Trawling podem apresentar mais benefícios quando aplicadas em fases anteriores, ou seja, na elicitação e/ou documentação dos requisitos de software.

A Figura 21 ilustra as práticas que são utilizadas para o gerenciamento dos requisitos pelos respondentes.

Figura 21 - Práticas mais utilizadas pelas empresas para o gerenciamento dos requisitos

Fonte: A autora (2022).

Já para o gerenciamento de requisitos, como ilustrado pela Figura 20, as práticas mais utilizadas pelos respondentes são o versionamento e controle de mudanças em requisitos, priorização, uso de *softwares* para a gestão dos requisitos e, também, de repositórios para os artefatos de especificação.

A utilização de softwares para o gerenciamento de requisitos pode facilitar essa atividade, além da possibilidade de gerar dados que podem auxiliar na tomada de decisão na gestão dos requisitos. Os requisitos também devem ser rastreados desde a sua concepção, até a finalização para que seja possível gerenciar toda a solicitação, bem como, as mudanças ao longo de seu desenvolvimento.

No cruzamento dos dados das empresas com as práticas adotadas, destaca-se que apenas entre as empresas de médio porte, em nenhuma de suas fases foram citadas as seguintes práticas etnografia, grupo focal, JAD e *trawling*, o que pode indicar uma dificuldade ou não conhecimento dos profissionais no entendimento desses tipos de práticas em requisitos.

A última questão da *survey* solicita que o participante mencione (caso conheça) em uma questão dissertativa outras melhores práticas em requisitos de *software*, além das citadas na literatura (questão 7). As respostas obtidas são listadas a seguir, seguida de referências posteriormente encontradas na literatura:

a) DevSecOps

Na cultura ágil, o Desenvolvimento e Operações (DevOps) enfatiza a velocidade de desenvolvimento e implantação para o alcance de uma entrega rápida e de valor, porém isso pode implicar em riscos para o processo de desenvolvimento de *software*. Os esforços para integrar a segurança no DevOps resultaram no paradigma DevSecOps que está ganhando interesse significativo da indústria e da academia.

Mao *et al.* (2020) e Rajapakse *et al.* (2022).

b) Técnica Moscow

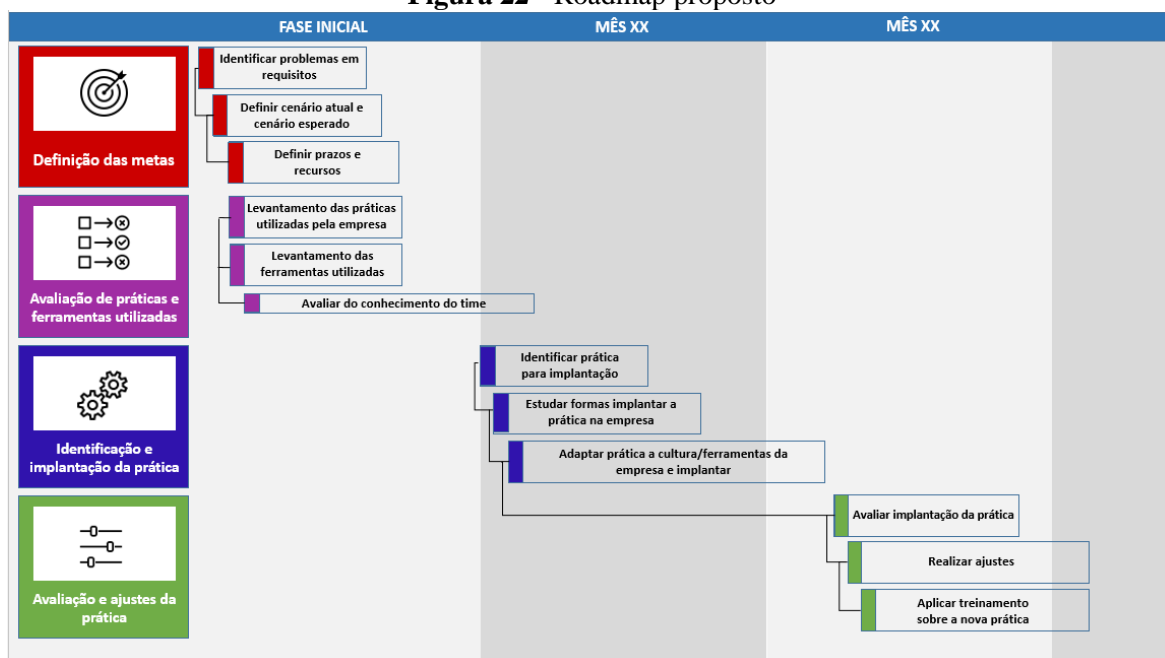
A técnica MoSCoW propõe separar os requisitos em quatro grandes categorias, cada uma com um significado intrínseco (*Must have*, *Should have*, *Could have* e *Won't have*), baseado no julgamento especializado e nos atributos ou propriedades dos requisitos.

Sagrado, Águila e Bosch (2018).

6 DESENVOLVIMENTO E APRESENTAÇÃO DO *ROADMAP* PROPOSTO

Com o propósito de solucionar problemas que ocorrem em empresas de desenvolvimento de *software*, foi desenvolvido um *roadmap* que permite adotar melhores práticas em desenvolvimento de *software* para tratar esses problemas. Muitas vezes, por questões como a cultura da organização, falta de recursos ou conhecimento, implementar novas práticas em determinadas áreas não é uma tarefa simples, principalmente em áreas críticas como é a área de requisitos e o *roadmap* pode auxiliar neste quesito. A Figura 22 apresenta o *roadmap* proposto.

Figura 22 - Roadmap proposto



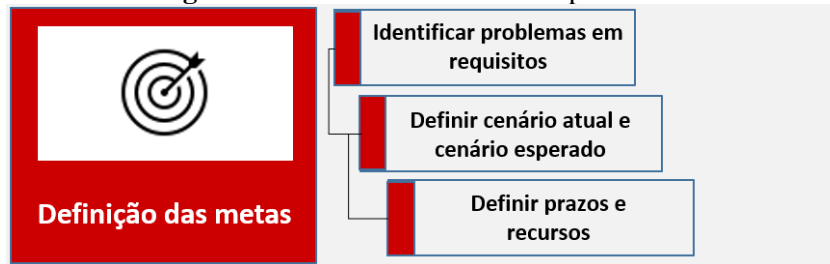
Fonte: A autora (2022).

O *roadmap* proposto, intitulado “RORSSI” (*Roadmap* para Requisitos de *Software* em Sistemas de Informação), um *roadmap* para a melhoria de práticas em requisitos de *software* no desenvolvimento de sistemas de informação, sugere quatro fases que permitem planejar, organizar e deixar mais visível como o problema será solucionado com a implantação das práticas em requisitos. No *roadmap*, as fases, tarefas e tempo em que elas serão executadas podem ser configurados de acordo com as necessidades da organização, além das conexões que indicam dependências entre as tarefas.

Com base no que é exposto por Siebelink *et al.* (2021), a primeira parte do *roadmap* RORSSI visa responder o porquê o *roadmap* está sendo aplicado e quais são seus objetivos. Sendo assim, a fase inicial do *roadmap* busca definir quais metas a organização pretende atingir

ao final da etapa de aplicação do *roadmap*. A Figura 23 apresenta a fase inicial do *roadmap*.

Figura 23 - Fase inicial do roadmap RORSSI



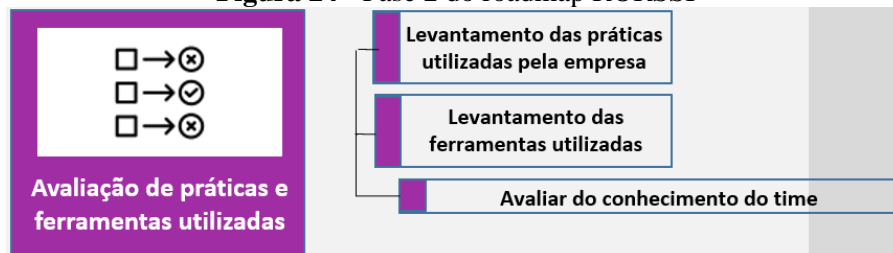
Fonte: A autora (2022).

Como mostra a Figura 23, para a fase de definição de metas foram considerados os passos de “identificação de problemas em requisitos” e “Definição do cenário atual e o cenário esperado” que permitem responder o porquê o *roadmap* está sendo aplicado. O problema relacionado a requisitos de *software* deve ser exposto e a organização deve sinalizar os resultados que espera alcançar após a etapa de aplicação do *roadmap*.

Ainda na fase inicial, a organização deve designar os recursos que serão responsáveis pela aplicação do *roadmap*, como também definir quais são os prazos esperados em cada tarefa indicada no *roadmap*. Estes prazos são indicados na linha azul, localizada no topo do artefato e cada tarefa deve ser disposta de modo que indique qual é o seu tempo definido de aplicação.

A Figura 24 mostra a fase 2 do *roadmap* proposto.

Figura 24 - Fase 2 do roadmap RORSSI



Fonte: A autora (2022).

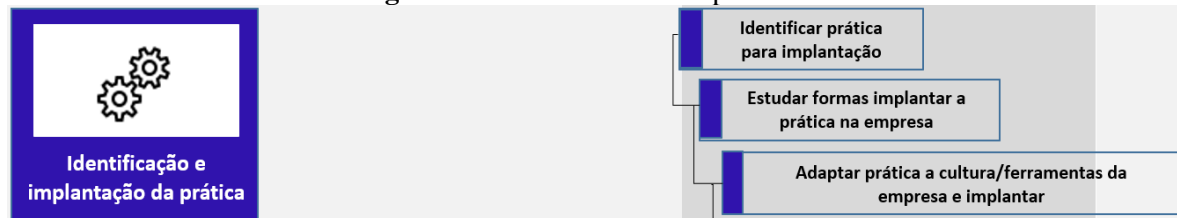
Conforme apresentado pela Figura 24, a fase 2 do RORSSI é baseada na fase de recursos indicada na estrutura de *roadmap* proposta pelos autores Phaal, Farrukh e Probert (2005), essa fase sugere que sejam identificadas quais as práticas em requisitos de *software* e ferramentas já são utilizadas pela empresa, para que a prática a ser adotada/melhorada se adapte ao que a empresa já possui no momento.

Nessa fase também pode ser avaliado, por meio de questionários, entrevistas ou outros recursos, o nível de conhecimento das equipes com relação as práticas e ferramentas que foram

levantadas nesta etapa.

Dessa forma, a Figura 25 ilustra a fase 3 do *roadmap* RORSSI.

Figura 25 - Fase 3 do roadmap RORSSI

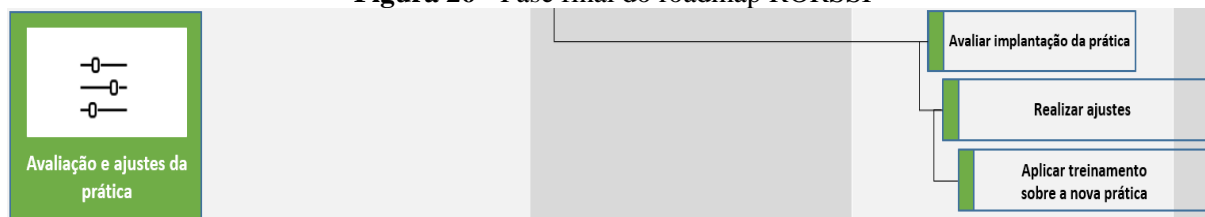


Fonte: A autora (2022).

Ainda com relação à fase de recursos indicada por Phaal, Farrukh e Probert (2005), na Figura 35 é possível observar que a fase 3 do RORSSI está relacionada à incorporação de novas práticas em requisitos de *software*. Sendo assim, os colaboradores responsáveis pela aplicação do *roadmap* devem identificar a prática em requisitos de *software* que pode solucionar o problema indicado (autores indicados no Quadro 4) e estudar a melhor forma de aplicação dessa, de acordo com a cultura da empresa e ferramentas que ela já possui.

A Figura 26 ilustra a fase final de aplicação do *roadmap* RORSSI.

Figura 26 - Fase final do roadmap RORSSI



Fonte: A autora (2022).

Conforme apresentado por Siebelink *et al.* (2021), na Figura 26 é possível observar que a fase final de aplicação do RORSSI visa identificar se a prática em requisitos de *software* que foi aplicada apresentou os resultados esperados como cenário na fase inicial do *roadmap*.

Caso o resultado não tenha sido como o esperado, o *roadmap* sugere outros passos como o de realizar ajustes na prática para que se adeque as necessidades da empresa, e também, a de realizar treinamentos sobre a prática implantada para a visibilidade de toda a organização.

A fim de verificar se o artefato proposto é factível, usável e útil, uma empresa de sistemas de informação realizou a sua aplicação e o avaliou. A factibilidade pode ser testada a partir da constatação de que o processo pôde ser seguido. Para a usabilidade, procura-se identificar a forma com que o processo é conduzido e os problemas ocorridos em cada uma das fases do processo, já a utilidade pode ser testada indiretamente, por meio dos resultados como

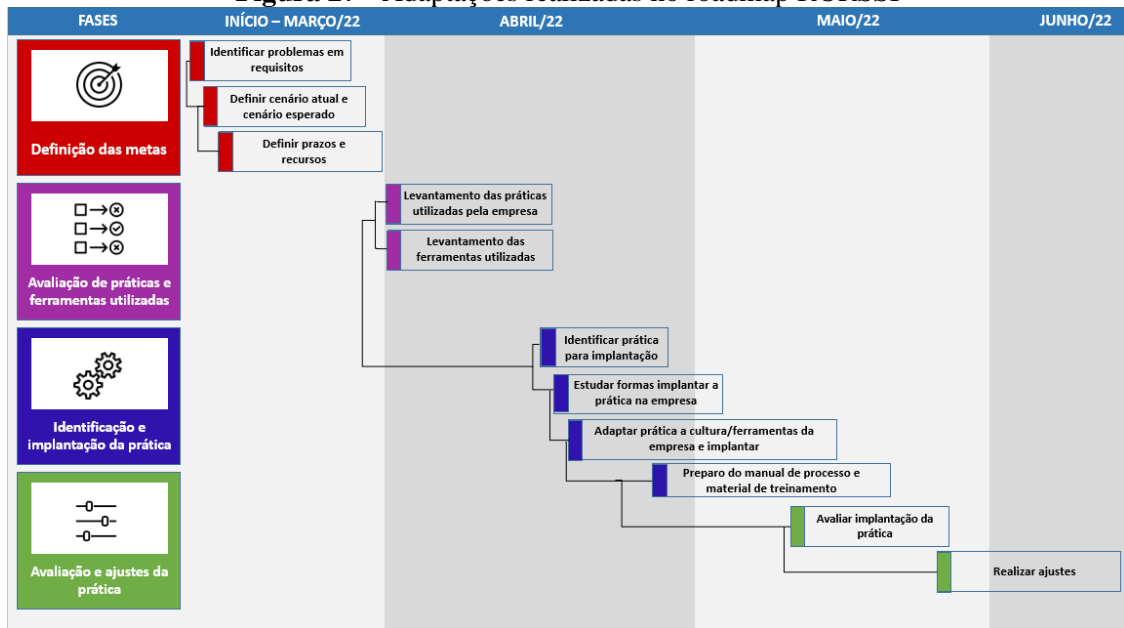
a definição dos objetivos da estratégia de operações, da identificação de problemas e da geração de planos de ação (PLATTS, 1998). Os itens a seguir abordam essa aplicação e avaliação.

6.1 Aplicação do *Roadmap* Proposto em uma Empresa de TI

Baseada na quarta etapa do método *Design Science Research* em que é feita a avaliação da solução que foi proposta, a pesquisadora solicitou a uma empresa de desenvolvimento de *software* (denominada XPTO, para fins dessa pesquisa) que possui cerca de 50 funcionários e atua desde 2006 na área de comercialização de ingressos, localizada na cidade de São Paulo, que aplicasse o *roadmap* para avaliar os resultados de sua utilização. Dessa forma, juntaram-se à pesquisadora dois profissionais que a empresa disponibilizou para a aplicação do *roadmap*, sendo um analista de negócios e um analista de sistemas.

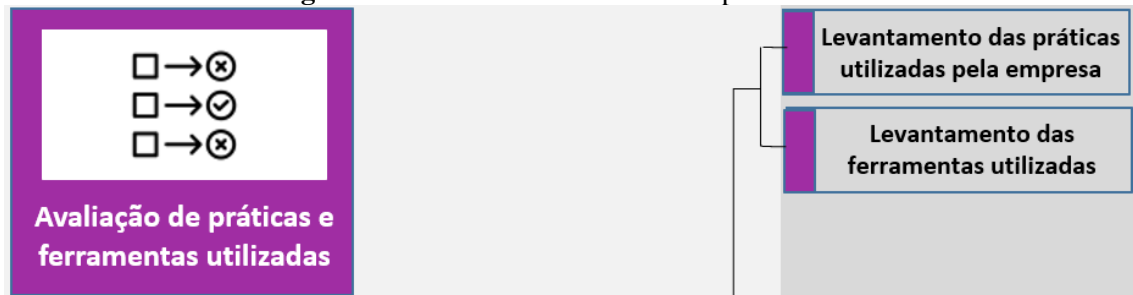
Em reunião com os gestores da empresa, em 28 de março de 2022, foram relatados os problemas como dificuldades na comunicação entre analistas de sistemas, desenvolvedores e analistas de teste, falta de visibilidade das tarefas em execução por cada desenvolvedor, falta de conhecimento das tarefas mais prioritárias e necessidades de planejamento, além das dificuldades de lidar com o gerenciamento de requisitos de *software* e dar início a adoção de práticas ágeis. Em comum acordo, ficou decidido que as ações para sanar os problemas relatados deveriam perdurar até o dia 31 de maio de 2022.

Com a flexibilidade de poder adaptar o *roadmap* segundo as necessidades da organização, no início da utilização do *roadmap* na empresa XPTO, este foi ajustado de acordo com os prazos definidos pela organização e os resultados que ela desejava obter. O ajuste foi realizado nas fases de avaliação de práticas e ferramentas, implantação da prática e na fase de ajustes. A Figura 27 apresenta os ajustes realizados no *roadmap* RORSSI para a aplicação na empresa que avaliou o artefato.

Figura 27 - Adaptações realizadas no roadmap RORSSI

Fonte: A autora (2022).

Na Figura 27, é possível notar que o ajuste efetuado na fase 2 do *roadmap* foi a retirada da etapa de avaliar o conhecimento do time sobre as práticas existentes, pois o levantamento das práticas em requisitos de *software* realizadas pela empresa foi efetuado em conjunto com o analista de sistemas. A Figura 28 ilustra a fase 2 do RORSSI aplicado na empresa XPTO.

Figura 28 - Fase 2 do RORSSI na empresa XPTO

Fonte: A autora (2022).

A Figura 28 ilustra a adaptação realizada na fase 2 do RORSSI. Nesta fase, foi realizado um levantamento das práticas em requisitos de *software* e das ferramentas que já eram utilizadas pela organização. O Quadro 5 apresenta essas práticas.

Quadro 5 – Práticas já utilizadas pela empresa XPTO

	Prática em requisitos de <i>software</i>	Fase de aplicação
1	Chuva de ideias (Brainstorming)	Elicitação; Documentação
2	Criação de casos de teste	Documentação, Validação
3	Documentação de requisitos não funcionais	Documentação; Validação
4	Entrevistas	Elicitação

	Prática em requisitos de <i>software</i>	Fase de aplicação
5	Escrita de requisitos em forma de histórias de usuário	Elicitação; Documentação
6	Observação de sistema existente	Elicitação; Documentação
7	Prototipação	Elicitação; Documentação
8	Realização de apresentações	Validação
9	Uso de templates para documentação	Documentação

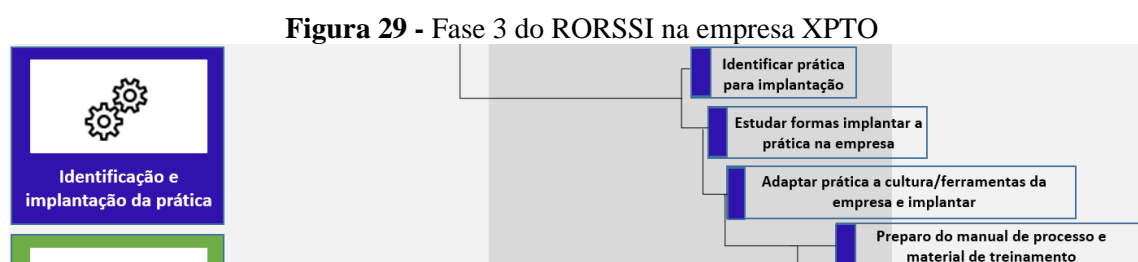
Fonte: A autora (2022).

Como é possível notar no Quadro 5, antes da aplicação do *roadmap* RORSSI, a empresa XPTO já utilizava algumas práticas em requisitos de *software*, sendo possível observar que não eram institucionalizadas práticas para o gerenciamento dos requisitos, sendo este realizado por meio de planilhas de cálculo.

Considerando as dificuldades de comunicação entre analistas de sistemas, desenvolvedores e analistas de teste, além da falta de visibilidade das tarefas em execução por cada desenvolvedor e o gerenciamento de requisitos, decidiu-se em comum acordo com a empresa XPTO institucionalizar o JIRA como ferramenta.

O JIRA é uma ferramenta desenvolvida pela empresa Atlassian, que permite o acompanhamento de tarefas e a gestão de projetos de *software*. Além de permitir a gestão dos requisitos de *software*, no JIRA ainda é possível realizar atividades como criação de casos de teste, priorização, rastreabilidade de requisitos e entre outros.

Sendo o uso de *softwares* para gestão de requisitos considerada uma boa prática em requisitos, na fase 3 de utilização do RORSSI, foi iniciada uma atuação na ferramenta JIRA adquirida e disponibilizada pela empresa. Foi realizado um estudo do que a ferramenta deveria conter para que atendesse as necessidades de gerenciamento de requisitos na empresa. A fase 3 de aplicação do RORSSI na empresa XPTO pode ser verificada na Figura 29.

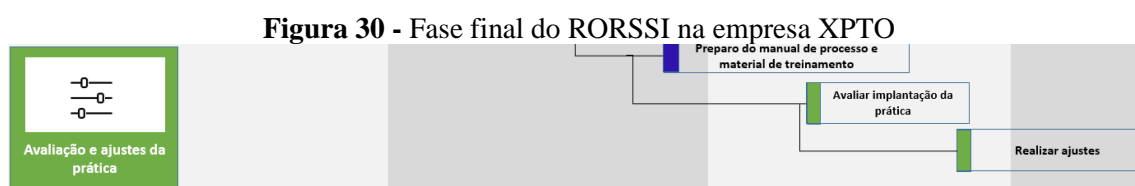


Fonte: A autora (2022).

Como é possível notar na Figura 29, para a empresa XPTO, além dos 3 passos indicados pelo RORSSI na fase 3, foi acrescentado mais um passo relacionado ao preparo de material de treinamento para essas novas práticas. Para isso, foi criado um manual que detalha o novo processo criado no JIRA para atendimento das demandas e vídeos treinamentos também

detalhando esse processo para entendimento das equipes.

No JIRA, o procedimento para atendimento das demandas foi reestruturado, criando campos que permitem um gerenciamento em tempo real do status da demanda com o executor no momento (analista, desenvolvedor, testador), além criação de sprints (duração de 1 semana cada) que permitem realizar o planejamento dessas demandas. A Figura 30 apresenta a fase final de aplicação do RORSSI na empresa XPTO.



Fonte: A autora (2022).

Na figura 30, é possível notar que a fase final de aplicação do RORSSI na empresa XPTO foi composta de apenas dois passos no *roadmap*, que são relacionados a avaliar a implantação da prática por meio da opinião dos recursos que foram alocados para aplicar o *roadmap* e realização de possíveis ajustes citados por eles.

Além da prática de adotar uma ferramenta para a gestão dos requisitos de *software*, com a aplicação do RORSSI também foi possível incorporar outras práticas na empresa. O quadro 6 mostra um comparativo das práticas que eram utilizadas pela empresa XPTO antes da aplicação do RORSSI e após a aplicação deste.

Quadro 6 – Práticas em requisitos antes e depois do RORSSI

Práticas em requisitos de <i>software</i> na empresa XPTO antes da aplicação do RORSSI	Práticas em requisitos de <i>software</i> na empresa XPTO após aplicação do RORSSI
Chuva de ideias (Brainstorming)	Chuva de ideias (Brainstorming)
Criação de casos de teste	Criação de casos de teste
Documentação de requisitos não funcionais	Documentação de requisitos não funcionais
Entrevistas	Entrevistas
Escrita de requisitos em forma de histórias de usuário	Escrita de requisitos em forma de histórias de usuário
Observação de sistema existente	Observação de sistema existente
Prototipação	Prototipação
Realização de apresentações	Realização de apresentações
Uso de templates para documentação	Uso de templates para documentação
-	Controle de mudanças/Versionamento de requisitos
-	Priorização de requisitos
-	Repositório para artefatos de especificação
-	Reuso de artefatos de especificação de requisitos de <i>software</i>
-	Uso de <i>softwares</i> para o preparo da

Práticas em requisitos de <i>software</i> na empresa XPTO antes da aplicação do RORSSI	Práticas em requisitos de <i>software</i> na empresa XPTO após aplicação do RORSSI
	documentação do sistema/gestão dos requisitos

Fonte: A autora (2022).

Como é possível observar no quadro 6, além da implantação do *software* JIRA, foram implantadas outras cinco práticas em requisitos na empresa XPTO. Para solucionar o problema relatado sobre a dificuldade dos desenvolvedores em conhecer as prioridades das demandas, no planejamento de cada *Sprint* no JIRA, foi implantada a prática de indicar a prioridade de execução em cada uma das demandas. O quadro 7 exemplifica a priorização das demandas.

Quadro 7 - Exemplo de priorização de demandas na empresa XPTO

SPRINT 2 (05/06/22 – 11/06/22)
Desenvolvedor 1
N1 – Nome da demanda N2 – Nome da demanda N3 – Nome da demanda

Fonte: A autora (2022).

No Quadro 7, é possível notar que para os planejamentos de *sprints* da empresa XPTO foi criada a prática de indicar o nível de prioridade de atendimento com os indicadores (N1, N2, N3...) juntamente com nome da demanda, o que permitiu maior visibilidade de cada desenvolvedor sobre as suas demandas.

Com o RORSSI também foi possível aplicar na empresa a prática de ter um repositório para os artefatos de especificação. Notada a necessidade de ter um espaço único para manter todos os artefatos produzidos em tempo de especificação de requisitos, foi estudada a possibilidade de criar um repositório para armazenamentos destes artefatos. Foi então feito inicialmente um Google Drive vinculado a um e-mail administrador da empresa, no qual foram criadas pastas de acordo com o assunto de cada demanda, para o armazenamento dos arquivos.

A implantação do repositório permitiu fazer o reuso de alguns artefatos de especificação, como algumas funcionalidades eram as mesmas para os diferentes sistemas produzidos pela empresa XPTO, foi possível reutilizar os artefatos de análise. Também foram estabelecidos *templates* para todas as documentações criadas pela empresa XPTO (documento de especificação, cronograma de projeto, propostas e entre outros), após essa definição, os *templates* foram armazenados no repositório.

O RORSSI também conduziu a implantação da prática de controle de mudanças/versionamento de requisitos no JIRA. Com o histórico de *status* e comentários da

demanda apresentada pela ferramenta, é possível verificar as alterações que o requisito sofreu desde a sua concepção, até o encerramento da demanda.

O item a seguir apresenta a avaliação do *roadmap* que foi realizada pela empresa XPTO.

6.2 Avaliação do Roadmap

Para a avaliação do *roadmap* RORSSI pela empresa XPTO, foi disponibilizado um questionário (Apêndice C), com perguntas abertas relacionadas à aplicação do artefato dentro da empresa. O questionário foi disponibilizado aos dois recursos alocados para a aplicação do *roadmap* via *Google Forms* no dia 08 de junho de 2022 e foram retornados no dia 09 de junho de 2022. A seguir, são apresentadas as respostas de cada um dos recursos.

Analista de sistemas:

1 – O *roadmap* proporcionou a oportunidade de implantação de ferramentas e melhorias nas práticas em requisitos de *software*?

Sim. Melhorou a visão geral das tarefas em aberto, em andamento e finalizadas.

2 – Você considera que a utilização deste *roadmap* pode conduzir as empresas de desenvolvimento de sistemas de informação na utilização de boas práticas em requisitos de *software*? Comente.

Sim. As fases definidas são coerentes com a implementação de boas práticas.

3 – Quais as suas sugestões para a melhoria deste *roadmap*?

Nenhuma. Concordo com as fases definidas para implementação das boas práticas para requisitos. E, na avaliação na prática, manter em aberto possíveis ajustes, deixa claro que melhoria continua.

Analista de negócios:

1 – O *roadmap* proporcionou a oportunidade de implantação de ferramentas e melhorias nas práticas em requisitos de *software*?

Com a implantação do Jira, as demandas ficaram bem-organizadas, facilitando o dia a dia do colaborador. A maior mudança após a implantação do Jira é referente à entrega das demandas dos desenvolvedores, pois antes os desenvolvedores recebiam muitas demandas de todos os lados e acabavam atrasando nas entregas.

2 – Você considera que a utilização deste *roadmap* pode conduzir as empresas de desenvolvimento de sistemas de informação na utilização de boas práticas em requisitos de *software*? Comente.

Sim, considero que o *Roadmap* está bem elaborado e nos ajudou bastante.

3 – Quais as suas sugestões para a melhoria deste *roadmap*?

Considero que está bom dessa forma.

Sendo a pesquisadora também responsável pela aplicação do *roadmap* juntamente com o analista de negócios e o analista de sistemas na empresa XPTO, a seguir são apresentados alguns pontos de destaque sobre a aplicação do *roadmap* e alguns pontos de atenção e melhorias a serem implementadas no RORSSI.

Em primeiro lugar, é importante salientar que o *roadmap* RORSSI apresentou-se útil em sua aplicação. Sendo uma ferramenta gráfica intuitiva, demonstrou os passos necessários para a incorporação de práticas em requisitos de *software* ao longo do tempo que foram definidos em conjunto com a empresa.

Outro ponto positivo de sua aplicação é que o *roadmap* enfatiza a necessidade de adaptar a prática que será adotada, a cultura e ferramentas que a empresa já possui, não deixando o processo rígido ou obrigando que a empresa adquira ferramentas específicas para isso. Sendo assim, a incorporação da prática em requisitos é factível e estruturada de acordo com as necessidades de cada organização.

O RORSSI também demonstrou flexibilidade em adaptar os passos necessários para se atingir o cenário desejado, bem como, realizar ajustes e incluir novas fases, caso o resultado da implantação da prática em requisitos não tenha sido satisfatório. No caso da empresa XPTO, os

resultados foram atingidos conforme o que era esperado, obteve-se melhoria na comunicação das demandas, entendimento das prioridades e melhoria no planejamento por meio da criação das *sprints*.

Na empresa XPTO, o RORSSI foi aplicado na busca de solucionar os problemas relacionados à requisitos de software na organização. A partir do entendimento desses problemas, foi realizado um levantamento das práticas em requisitos que já eram utilizadas na empresa e também de possíveis práticas encontradas na literatura que poderiam solucionar os problemas que foram relatados.

Sendo relatados problemas como falta de visibilidade, priorização de tarefas e falhas de comunicação e sendo identificado na literatura que o uso de um software para gestão dos requisitos poderia auxiliar na rastreabilidade e gestão dos requisitos, decidiu-se em comum acordo com a empresa XPTO institucionalizar a ferramenta JIRA. Foi então, iniciado um trabalho nessa ferramenta para adequá-la as necessidades da organização.

Com a utilização efetiva do JIRA, também foi possível realizar adaptações na ferramenta para a incorporação das práticas de priorização de requisitos e reuso de artefatos de especificação de requisitos de software. Com a visibilidade das tarefas, proporcionada pelo uso do RORSSI e pela organização dos requisitos no JIRA, foi possível criar nomenclaturas que indicam a prioridade de cada tarefa, além de permitir o reaproveitamento de artefatos de especificação em comum, o que segundo a literatura, promove economia de tempo e recursos e permite que as equipes foquem nas principais funcionalidades antecipadamente.

A incorporação da prática do reuso de artefatos de especificação, vislumbrou a implantação de outras práticas por meio do RORSSI. Percebeu-se a necessidade de centralizar o armazenamento desses artefatos criados, dessa forma, foi criado um repositório específico para artefatos de especificação no Google Drive da empresa XPTO. A outra prática aplicada, refere-se ao controle de mudança de requisitos e versionamento, que é controlada tanto pela indicação de versão no repositório de especificação, quanto pelo JIRA, por meio do histórico e dos comentários em cada tarefa.

Com o auxílio do *roadmap*, foram aplicadas melhorias em algumas das práticas em requisitos já utilizadas pela empresa XPTO, como a criação dos casos de teste no documento de requisitos e melhorias nos *templates* utilizados para a documentação em geral.

Após a aplicação do RORSSI, a empresa XPTO, passou a adotar até quinze boas práticas em requisitos de software, e seus efeitos podem ser vistos por toda a organização. A adoção das demais práticas abordadas na literatura também foram avaliadas pela empresa XPTO, porém não foram consideradas como prioritárias para o momento e sua incorporação será reavaliada

posteriormente.

Por fim, é importante destacar a necessidade de melhorias na usabilidade do RORSSI, pois mesmo tornando mais visível como a prática será incorporada ao longo do tempo e os passos necessários para tal, é necessário que o aplicador tenha conhecimentos mais avançados sobre a prática/ferramenta que a empresa deseja utilizar, podendo um aplicador menos experiente apresentar dificuldades na definição dos passos necessários para a incorporação da prática.

CONCLUSÃO

Tendo em vista os objetivos do presente estudo, procurou-se inicialmente identificar na literatura quais as melhores práticas que podem ser aplicadas em requisitos de *software* para o desenvolvimento de sistemas de informação. Essa pesquisa resultou em um total de 27 práticas, que vão desde práticas consagradas e muito utilizadas pelas empresas, como também outras que geralmente são menos conhecidas.

A partir das práticas em requisitos de *software* identificadas na literatura, foi possível estruturar uma *survey* com a listagem dessas práticas em que profissionais de TI puderam confirmar se as práticas listadas realmente são utilizadas pelas empresas de desenvolvimento de sistemas de informação.

A *survey* permitiu também identificar outras práticas que foram citadas pelos participantes da pesquisa de campo, além de identificar dados como o crescimento da utilização dos métodos ágeis por parte das empresas e quais das práticas em requisitos são mais utilizadas em determinadas fases como elicitação, documentação, validação e gerenciamento de requisitos entre os respondentes.

Com o intuito de melhorar as práticas usadas na elicitação e especificação de requisitos de *software* ou até implantar novas práticas, foi desenvolvido um *roadmap* que auxilia as empresas de desenvolvimento de sistemas de informação a aprimorar e implantar melhores práticas de requisitos de *softwares*. O *roadmap* foi construído a partir do estudo das melhores práticas, do conceito de *roadmaps* e de *roadmaps* correlatos.

Assim, com o entendimento dos passos necessários para a criação de um *roadmap* para requisitos de *software*, o *roadmap* foi estruturado procurando-se solucionar o problema em requisitos relatado pela empresa e indicando os passos para a implantação da prática em requisitos de *software*.

O *roadmap* proposto, intitulado “RORSSI”, foi aplicado e avaliado em uma empresa que desenvolve sistemas de informação. A partir dessa aplicação do *roadmap* foi possível verificar sua factibilidade, utilidade e usabilidade. Os resultados da aplicação na empresa XPTO mostraram-se satisfatórios, sendo que os problemas relacionados a requisitos passaram a ser tratados.

A melhoria nas práticas pode ser percebida pela ampliação de seis novas práticas relacionadas às áreas apontadas pela própria empresa nas fases iniciais do *roadmap*.

Com relação à usabilidade do RORSSI, o *roadmap* apresenta sugestão de passos a serem seguidos para a incorporação da prática, porém notou-se que se a aplicação for realizada por

um usuário com menor experiência na execução da prática/ferramenta adotada, pode gerar dúvidas na definição dos passos para se atingir o objetivo.

Como sugestão para trabalhos futuros, poderá ser realizada uma pesquisa detalhada sobre os passos necessários para a implantação de cada prática, ou seja, ao invés de analisar qual prática pode ser adotada por meio do artefato, o *roadmap* já possuirá uma estrutura definida dos passos necessários para a implantação de cada uma das boas práticas identificadas, para que possa ser utilizado também por usuários menos experientes na área de requisitos de *software*. A partir de então, pode ser aplicado em empresas de porte e realidades diferentes quanto ao processo de desenvolvimento para sua validação.

REFERÊNCIAS

- ABID, A. M; DIN, U. Z; KHAN, I. M; NAEEM, T. **Factors affecting requirements engineering in agile software development: A systematic analysis.** TUENGR, 2020. Disponível em: <https://tuengr.com/V11A/11A9AM.pdf>. Acesso em: 25 ago. 2021.
- ALASHQAR, A. M; ELFETOUH, A. A; EL-BAKRY, H. M. **Requirement engineering for non-functional requirements.** JICT, 2015. Disponível em: https://iugspace.iugaza.edu.ps/bitstream/handle/20.500.12358/25105/Alashqar%2c%20Abdelkrareem%20M._5.pdf?sequence=1&isAllowed=y. Acesso em: 20 jul. 2021.
- ALBRIGHT, R. E. **Roadmapping Convergence.** Springer, 2003. Disponível em: https://link.springer.com/chapter/10.1007%2F1-4020-4107-1_3. Acesso em: 21 nov. 2021.
- ALEEM, S; CAPRETZ, L. F; AHMED, F. **Game development software engineering process life cycle: a systematic review.** Springer, 2016. Disponível em: <https://jserd.springeropen.com/articles/10.1186/s40411-016-0032-7>. Acesso em: 25 ago. 2021.
- AL-FEDAGHI, S. **High-Level representation of time in diagrammatic specification.** Elsevier, 2015. Disponível em: <https://www.sciencedirect.com/science/article/pii/S187705091502654X>. Acesso em: 25 ago. 2021.
- ALMEIDA, S. L. F; RODRIGUES, N. N; LIRA, H. B; LIMA, C. D. Q; DUNNING, A. K. M. P; FREITAS, R. M. S. **Aplicação e análise de processo de desenvolvimento de software: um estudo de caso no GPES-IFPB.** Principia, 2018. Disponível em: https://www.researchgate.net/profile/HeremitaLira/publication/330267798_Aplicacao_e_analise_de_processo_de_desenvolvimento_de_software_um_estudo_de_caso_no_GPES-IFPB/links/5d4042e3299bf1995b5739c1/Aplicacao-e-analise-de-processo-de-desenvolvimento-de-software-um-estudo-de-caso-no-GPES-IFPB.pdf. Acesso em: 25 jun. 2021.
- ALNAJJAR, M; NASER, S. S. A. **Evaluating software engineering practices in Palestine.** IJSCMC, 2015. Disponível em: dstores.alazhar.edu.ps/xmlui/bitstream/handle/123456789/114/4115ijscmc04.pdf?sequence=1&isAllowed=y. Acesso em: 21 maio 2022.
- ALSANOOSY, T; SPICHKOVA, M; HARLAND, J. **Cultural influences on requirements engineering process in the context of Saudi Arabia.** Cornell, 2018. Disponível em: <https://arxiv.org/abs/1807.01930>. Acesso em: 14 ago. 2021.
- ARCHARYA, S; MANOHAR, P. A; WU, P. Y. **Collaborative education: building a skilled software validation and verification community.** EDSIG, 2018. Disponível em: <http://proc.iscap.info/2018/pdf/4645.pdf>. Acesso em: 15 maio 2022.
- BARCELOS, L. V; PENTEADO, R. D. **Elaboration of software requirements documents by means of patterns instantiation.** JSERD, 2017. Disponível em: www.jserd.springeropen.com/articles/10.1186/s40411-017-0038-9. Acesso em: 21 maio 2022.

2022.

BARRETO, D. L.; MATOS, M. R.; GARCEZ, L. C. F.; SILVA, S. V.; VASCONCELOS, A. P. V.; MORAIS, A. S. C. **Proposta de modelagem de processos para apoiar a implementação do processo de gestão de requisitos do MPS.BR.** SG, 2020. Disponível em: <https://www.revistasg.uff.br/sg/article/view/1637/1343>. Acesso em: 02 ago. 2021.

BEEDLE, M.; BENNEKUM, A. V.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. **The agile manifesto.** Estados Unidos, 2001.

BORTOLUCI, R.; DUDUCHI, M.; ESTEVAM, A. O. **Análise de aspectos do processo de desenvolvimento de *software* em métodos ágeis.** CPS, 2014. Disponível em: www.pos.cps.sp.gov.br/files/artigo/file/481/c4952f4daee20d551b0863622cf05b93.pdf. Acesso em: 26 jan. 2022.

BOYARCHUK, A.; PAVLOVA, O.; BODNAR, M.; LOPATTO, I. **Approach to the analysis of *software* requirements specification on its structure correctness.** TSIS, 2020. Disponível em: ceur-ws.org/Vol-2623/paper9.pdf. Acesso em: 02 set. 2021.

CARDOSO, R. D.; SILVA, R. B.; SIQUEIRA, F. L. **Mapeamento dos processos e artefatos da engenharia de requisitos para o eXtreme programming.** ClbSE, 2018. Disponível em: https://www.researchgate.net/profile/Fabio-Levy-Siqueira/publication/332427603_Mapeamento_dos_processos_e_artefatos_da_Engenharia_de_Requisitos_para_o_eXtreme_Programming/links/5cb4d8c6a6fdcc1d49979053/Mapeamento-dos-processos-e-artefatos-da-Engenharia-de-Requisitos-para-o-eXtreme-Programming.pdf. Acesso em: 20 jun. 2021.

CARRILLO DE GEA, J. M.; NICOLÁS, J.; FERNÁNDEZ-ALEMÁN, J. L.; TROVAL, A.; IDRI, A. **Are the expected benefits of requirements reuse hampered by distance? An experiment.** Springer, 2016. Disponível em: <https://link.springer.com/article/10.1186/s40064-016-3782-0>. Acesso em: 21 jul. 2021.

CARVALHO, B. D. S. **Engenharia de requisitos:** Boas práticas para elicitação de requisitos. Ada Lovelace, 2018. Disponível em: <http://anais.unievangelica.edu.br/index.php/adalovelace/article/view/4646>. Acesso em: 02 ago. 2021.

CASTRO, R. M.; SOUZA, G. S. **O uso de recursos lúdicos para o ensino de processos em engenharia de *software*.** SBC, 2016. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/9670>. Acesso em: 25 jun. 2021.

CAUCHICK-MIGUEL, P. A. **Metodologia da pesquisa em engenharia de produção e gestão de operações.** Rio de Janeiro: Elsevier, 2018.

CEZERINO, A.; NASCIMENTO, F. P. **Utilização da técnica de desenvolvimento orientado por comportamento (BDD) no levantamento de requisitos.** RICA, 2016. Disponível em: <https://rica.unibes.com.br/rica/article/view/728>. Acesso em: 08 jun. 2021.

COELHO, J. A. F.; JUNIOR, S. B.; TAHLM, E. F. **Roadmap tecnológico: um estudo preliminar**. RECADM, 2012. Disponível em: <http://www.periodicosibepes.org.br/index.php/recadm/article/view/1043>. Acesso em: 03 dez. 2021.

COMISSÃO NACIONAL DE ÉTICA EM PESQUISA. **CNS**. Brasília. Disponível em: <https://tecnoblog.net/responde/referencia-site-abnt-artigos/>. Acesso em 20 jun. 2022.

CÔRTEZ, P. L. **Administração de sistemas de informação**. São Paulo: Saraiva, 2017.

CUPANI, A. **La peculiaridad del conocimiento tecnológico**. STUDIA, 2006. Disponível em: <https://www.scielo.br/j/ss/a/STXgdYmmHXL4Qjcb5xYqrDm/abstract/?lang=es>. Acesso em: 02 jun. 2022.

DIAS, C. R.; SOUZA, A. C. S. **T-AADSP Requirements** – Gerenciamento de requisitos com base na abordagem AADSP. IFBA, 2018. Disponível em: <https://labrasoft.ifba.edu.br/publicacoes/tcc/t-aadsp-requirements-gerenciamento.pdf>. Acesso em: 20 maio 2022.

DOURADO, A. C.; RODRIGUES, J. C. **Técnicas para o levantamento de requisitos: uma proposta para a obtenção de resultados mais precisos**. CORE, 2015. Disponível em: <https://core.ac.uk/download/pdf/211928398.pdf>. Acesso em: 15 jul. 2021.

DRESCH, A.; LACERDA, D. P.; JÚNIOR, J. A.V. A. **Design Science Research: Método de pesquisa para avanço da ciência e tecnologia**. Porto Alegre: Bookman, 2020.

ELIJAH, J.; MISHRA, A.; UDO, M. C.; ABDULGANIYU, A.; AIBINU, A. M. **Survey on requirement elicitation techniques: it effect on software engineering**. IJIRCE, 2017. Disponível em: https://www.researchgate.net/profile/Elijah-Joseph/publication/317606075_Survey_on_Requirement_Elicitation_Techniques_It's_Effect_on_Software_Engineering/links/5942f0f7e9b6910eae62d/Survey-on-Requirement-Elicitation-Techniques-Its-Effect-on-Software-Engineering.pdf. Acesso em: 03 jul. 2021.

FAGUNDES, P. B.; FREUND, G. P.; VITAL, L. P.; BARROS, C. M.; MACEDO, D. D. J. **Taxonomias, ontologias e tesauros: possibilidades de contribuição para o processo de engenharia de requisitos**. SEER, 2020. Disponível em: <https://www.seer.ufrgs.br/EmQuestao/article/view/90347>. Acesso em: 06 set. 2021.

FERNANDES, J. M.; MACHADO, R. J. **Requisitos em projetos de software e de sistemas de informação**. São Paulo: Novatec, 2017.

FINKELSTEIN, A.; KRAMER, J. **Software engineering: a roadmap**. ACM, 2000. Disponível em: <https://dl.acm.org/doi/abs/10.1145/336512.336519>. Acesso em: 07 mar. 2022.

GARCIA, J. E.; PAIVA, A. C. R. **Maintaining requirements using Web Usage Data**. Elsevier, 2016. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050916323729>. Acesso em: 10 jul.

2021.

GONZALEZ-HUERTA, J; MOLLÉRI, J. S; SÁBLIS, A; ZABARDAST, E. **Experiential learning approach for software engineering courses at higher education level**. Cornell, 2020. Disponível em: <https://arxiv.org/abs/2012.14178>. Acesso em: 10 ago. 2021.

GUALHANO, M. A; SILVA, S. V; VASCONCELOS, A. P. V. **Gestão do escopo para o projeto Biblioteca de reuso de requisitos de software para diferentes domínios**. NAVUS, 2019. Disponível em: <https://pdfs.semanticscholar.org/50d3/09e8cd6baf2ceb4211bad54bbf7d43682645.pdf>. Acesso em: 01 jul. 2021.

HACKBARTH, R; MOCKUS, A; PALFRAMAN, J; WEISS, D. **Assessing the state of software in a large enterprise: a twelve year retrospective**. The Art and Science, 2015. Disponível em: <http://mockus.us/papers/assess2.pdf>. Acesso em: 10 ago. 2021.

HAIDRAR, S; ANWAR, A; BRUEL, J.M; ROUDIES, O. **A domain-specific language to manage requirements traceability**. JS, 2018. Disponível em: <http://www.jsoftware.us/vol13/348-JSW15339.pdf>. Acesso em: 15 maio 2022.

HERRERA, G. Y. L; SANZ, J. C. J. **Best practices for requirements identification, specification, and validation to guide software implementation and maintenance processes for applications in a electricity supply company**. S&T, 2015. Disponível em: <https://www.redalyc.org/pdf/4115/411543658005.pdf>. Acesso em: 15 set. 2021.

HERNÁNDEZ, Y. M; DIHIGO, A. G; CINTRA, A. V. **Estrategia de desarrollo de requisitos no funcionales en aplicaciones para la salud**. 2020. Disponível em: <http://www.revinformatica.sld.cu/index.php/rcim/article/view/373>. Acesso em: 30 ago. 2021.

JIRAPANTHONG, W. **Experience in applying of ISO 29110 to agile software development**. JIST, 2019. Disponível em: <https://ph02.tci-thaijo.org/index.php/JIST/article/view/187845>. Acesso em: 23 ago. 2021.

KHAN, M. N. A; KHALID, M; HAQ, S. **Review of requirements management issues in software development**. MECS, 2013. Disponível em: https://d1wqtxts1xzle7.cloudfront.net/49258419/Review_of_Requirements_Management_Issues20160930-21565-u7888h-with-cover-page-v2. Acesso em: 22 maio 2022.

KHEDR, A; AHMED, A. A; KHOLEIF, S. **Adaptive concept map approach for software requirements validation**. FUTURE, 2019. Disponível em: <https://digitalcommons.aaru.edu.jo/fcij/vol4/iss1/4/>. Acesso em: 15 maio 2022.

KUSSUNGA, F; RIBEIRO, P; SANTOS, N. **Caracterização de um ambiente visual para apoiar as cerimônias do SCRUM**. AISEL, 2019. Disponível em: <https://repositorium.sdum.uminho.pt/handle/1822/71460>. Acesso em: 15 jul. 2021.

LACERDA, D. P; DRESCH, A; PROENÇA, A; JÚNIOR, J. A. V. A. **Design Science Research: método de pesquisa para a engenharia de produção**. G&P, 2013. Disponível em: <https://www.scielo.br/j/gp/a/3CZmL4JJxLmxCv6b3pnQ8pq/?lang=pt&format=pd>. Acesso em: 04 ago. 2021.

LAMPA, I. L.; CONTESSOTO, A. G.; AMORIM, A. R.; ZAFALON, G. F. D.; VALÊNCIO, C. R.; SOUZA, R. C. G. **Project scope management: A strategy oriented to the requirements engineering**. SCITEPRESS, 2017. Disponível em: <https://www.scitepress.org/Papers/2017/63186/63186.pdf>. Acesso em: 23 jul. 2021.

LAPLANTE, P. A. **Requirements Engineering for software and systems**. Boca Raton: CRC Press, 2014.

LAUDON, C. K.; LAUDON, P. J. **Sistemas de informação gerenciais**. São Paulo: Pearson, 2014.

LEMAZURIER, L.; CHAPURLAT, V.; GROSSETÊTE, A. **An MBSE approach to pass from requirements to functional architecture**. IFAC, 2017. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405896317319183>. Acesso em: 15 ago. 2021.

LIMA, I. S. C.; SILVA, E. O. **A utilização de engenharia de domínio para descoberta de padrões de análise em sistemas de controle de mudanças**. SEER, 2015. Disponível em: <https://seer.uniacademia.edu.br/index.php/cesi/article/view/298>. Acesso em: 10 jul. 2021.

LIMA, H. H.; SILVA, G. S. S.; VASCOCELOS, M. K. C.; BERNARDINO, M.; BORDIN, A. S. **Engenharia de requisitos na resolução de problemas da comunidade: Lições aprendidas**. SOL, 2019. Disponível em: <https://sol.sbc.org.br/index.php/eres/article/view/8502/8403>. Acesso em: 15 set. 2021.

LIU, H.; JUNG, J.; SUN, Q. **Extending agile methods with requirements engineering**. IJCER, 2020. Disponível em: https://www.researchgate.net/profile/Juergen-Jung-3/publication/343262716_Extending_Agile_Methods_with_Requirements_Engineering/links/5f2021f8299bf1720d6ad234/Extending-Agile-Methods-with-Requirements-Engineering.pdf. Acesso em: 21 maio 2022.

MACHADO, F. N. R. **Análise e gestão de requisitos de software: onde nascem os sistemas**. São Paulo: Érica, 2016.

MACIEL, D.; PAIVA, A. C. R.; SILVA, A. R. **From requirements to automated acceptance tests of interactive apps: an integrated model-based testing approach**. ENASE, 2019. Disponível em: <https://pdfs.semanticscholar.org/742d/1c5490570078a2a8ded929f4a8d84d2e324e.pdf>. Acesso em: 14 jul. 2021.

MAO, R.; ZHANG, H.; DAI, Q.; HUANG, H.; RONG, G.; SHEN, H.; CHEN, L.; LU, K. **Preliminary findings about DevSecOps from grey literature**. IEEE, 2020. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9282798>. Acesso em: 20 abr. 2022.

MARTINS, P. L.; MELO, B. M.; QUEIROZ, D. L.; SOUZA, M. S.; BORGES, R. O. **Tecnologia e sistemas de informação e suas influências na gestão e contabilidade**. SEGeT, 2012. Disponível em: <https://www.aedb.br/seget/arquivos/artigos12/28816533.pdf>. Acesso em: 20. out. 2021.

MEDEIROS, J. D. R. V; ALVES, D. C. P; VASCONCELOS, A. M. L; SCHUENEMANN, C. T. L. L.S; WANDERLEY, E. **Engenharia de requisitos em projetos ágeis: uma revisão sistemática da literatura**. Principia, 2015. Disponível em: <https://periodicos.ifpb.edu.br/index.php/principia/article/view/459>. Acesso em: 02 ago. 2021.

MENEZES, P. M; COUTO, T; CORREIA, T. A; SANTANA, L. **A engenharia de requisitos: um caso de implementação de um sistema para engenharia de requisitos**. SET, 2015. Disponível em: <https://periodicos.set.edu.br/exatas/article/view/2502>. Acesso em: 23 out. 2021.

MERSINO, A. Why agile is better than waterfall (Based on Standish Group Chaos Report 2020). **Vitality Chicago**, novembro, 2021. Disponível em: <https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/>. Acesso em: 21 jun. 2022.

MOHER, D; SHAMSEER, L; CLARKE, M; GHERSI, D; LIBERATI, A; PETTICREW, M; SHEKELLE, P; STEWART, L. A; PRISMA-P GROUP. **Ítems de referencia para publicar protocolos de revisions sistemáticas y metaanálisis: declaración PRISMA-P 2015**. SCIELO, 2016. Disponível em: https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S217451452016000200010. Acesso em: 14 jul. 2021.

MUSTAFA, N; KAMALRUDIN, M. **A new consistency validation approach to enhance the quality of functional security requirements for secure software**. JTECE, 2018. Disponível em: <https://core.ac.uk/download/pdf/235222002.pdf>. Acesso em: 20 jun. 2021.

NILSSON, S; BUFFONI, L; SANDAHL, K; JOHANSSON, H; TAHIR SHEIK, B. **Empirical study of requirements engineering in cross domain development**. Diva, 2018. Disponível em: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1239853&dswid=-3899>. Acesso em: 15 ago. 2021.

NUSEIBEH, B; EASTERBROOK, S. **Requirements Engineering: a roadmap**. ACM, 2000. Disponível em: <https://dl.acm.org/doi/abs/10.1145/336512.336523>. Acesso em: 07 mar. 2022.

ORINDA, E. O; NGUGI, H. N. **Concept based students data retrieval and repository system**. IJSES, 2018. Disponível em: <http://ijses.com/wp-content/uploads/2018/07/176-IJSES-V2N6.pdf>. Acesso em: 15 mai. 2022.

OYEDEJI, S; PENZENSTADLER, B. **Karlskrona manifesto: software requirement engineering good practices**. CEUR, 2018. Disponível em: <http://ceur-ws.org/Vol-2223/paper3.pdf>. Acesso em: 20 set. 2021.

PHAAL, R; FARRUKH, C. J. P; PROBERT, D. R. **Developing a technology roadmapping system**. PSU, 2005. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.455.2248&rep=rep1&type=pdf>. Acesso em: 10 ago. 2021.

PANTALEO, G; RINAUDO, L. **Ingeniería de software**. Buenos Aires: Alfaomega, 2015.

PANTELIC, V; SCHAAP, A; WASSYNG, A; BANDUR, V; LAWFORD, M. **Something is Rotten in the state of documenting Simulink models**. SCITEPRESS, 2019. Disponível em: <https://www.scitepress.org/Papers/2019/75860/75860.pdf>. Acesso em: 04 ago. 2021.

PEREIRA, S, R; PAIVA, P. B; SOUZA, P. R. S; SIQUEIRA, G; PEREIRA, A. R. **Sistema de informação para gestão hospitalar**. JHI, 2012. Disponível em: <http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/206>. Acesso em: 18 jun. 2021.

PEREIRA, T. C; ALENCAR, F. M. R, SILVA, J. R. F, CASTRO, J. F. B. **Requisitos não-funcionais em modelos de processos de negócio: Uma revisão sistemática**. SBC, 2013. Disponível em: <https://sol.sbc.org.br/index.php/sbsi/article/view/5674/5571>. Acesso em: 15 set. 2021.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional**. Porto Alegre: AMGH, 2016.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional**. Porto Alegre: AMGH, 2021.

PRIKLADNICKI, R; WILLI, R; MILANI, F. **Métodos ágeis para desenvolvimento de software**. Porto Alegre: Bookman, 2014.

PURI, C. P. **Agile management: Feature Driven Development**. Nova Delhi. Global India Publications, 2009.

RAJAPAKSE, R. N; ZAHEDI, M; ALI BABAR, M; SHEN, H. **Challenges and solutions when adopting DevSecOps: A systematic review**. Elsevier, 2022. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950584921001543>. Acesso em: 20 abr. 2022.

RAMÍREZ, J. L; VEGA, O. **Sistemas de información gerencial e innovación para el desarrollo de las organizaciones**. Telematiquejournal, 2015. Disponível em: ojs.urbe.edu/index.php/telematique/article/view/2715. Acesso em: 12 jul. 2021.

RIBEIRO, F. G. C; RETTBERG, A; PEREIRA, C. E; SOARES, M. S. **A model-based engineering methodology for requirements and formal design of embedded and real-time systems**. AISEL, 2017. Disponível em: <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1748&context=hicss-50>. Acesso em: 08 jun. 2021.

RÍOS, J. M; PEDREIRA-SOUTO, N. **Approach of agile methodologies in the development of web-based software**. MDPI, 2019. Disponível em: <https://www.mdpi.com/2078-2489/10/10/314>. Acesso em: 09 set. 2021.

RUEDA, S; PANACH, J. I. I; CABOTÀ, J. B; VALVERDE, F. **Applying role-playing game in software development subjects**. Researchgate, 2016. Disponível em: https://www.researchgate.net/profile/SilviaRueda2/publication/311363255_APPLYING_ROLEPLAYING_GAME_IN_SOFTWARE_DEVELOPMENT_SUBJECTS/links/5b78251ea6fdc5f8b51c997/APPLYING-ROLE-PLAYING-GAME-IN-SOFTWARE-DEVELOPMENT-SUBJECTS.pdf. Acesso em: 13 jul. 2021.

SAGRADO, J; ÁGUILA, I, M; BOSCH, A. **Expansión cuantitativa del método MoSCoW para la priorización de requisitos**. UAL, 2018. Disponível em: <http://repositorio.ual.es/handle/10835/6110>. Acesso em: 25 abr. 2022.

SANTOS, A. W. L; GOMES, F. J. L. **Levantamento de requisitos no desenvolvimento de jogos**. BJD, 2019. Disponível em: <https://www.brazilianjournals.com/index.php/BRJD/article/view/3210>. Acesso em: 30 ago. 2021.

SCHMITT, C; LIGGESMEYER, P. **Getting grip on security requirements elicitation by structuring and reusing security requirements sources**. CSIMQ, 2015. Disponível em: <https://csimq-journals.rtu.lv/article/view/csimq.2015-3.02/712>. Acesso em: 21 mai. 2022.

SEVILLA, G; ZAPATA, S; GIRALDO, F; TORRES, E; COLLAZOS, C. **A framework for teaching distributed requirements engineering in Latin American Universities**. World Academy, 2018. Disponível em: <https://publications.waset.org/10009860/a-framework-for-teaching-distributed-requirements-engineering-in-latin-american-universities>. Acesso em: 25 set. 2021.

SHARMA, S; HASTEER, N. **A comprehensive study on state of scrum development**. IEEE, 2016. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7813837>. Acesso em: 05 nov. 2021.

SIEBELINK, R; HOFMAN, E; HALMAN, J. I. M; NEE, I. **Roadmapping: (Missed) opportunities to overcome strategic challenges**. Elsevier, 2021. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0007681321000161>. Acesso em: 07 mar. 2022.

SILVA, D. E. S; SOUZA, I. T; CAMARGO, T. **Metodologias ágeis para o desenvolvimento de software: aplicação e o uso da metodologia Scrum em contraste ao modelo tradicional de gerenciamento de projetos**. UNG, 2013. Disponível em: <http://revistas.ung.br/index.php/computacaoaplicada/article/view/1408>. Acesso em: 20 out. 2021.

SILVA, E. C; LOVATO, L. A. **Framework Scrum: eficiência em projetos de software**. GEP, 2016. Disponível em: <https://periodicos.uninove.br/gep/article/view/9640>. Acesso em: 10 out. 2021.

SILVA, J. C; SOUSA, S. P. A; KULESZA, R; BRITO, A.V. **Uma avaliação do emprego do jogo Modelando como apoio ao ensino de engenharia de requisitos**. SBC, 2012. Disponível em: http://www2.sbc.org.br/csbc2012/anais_csbc/eventos/wei/artigos/Uma%20avaliacao%20do%20emprego%20do%20jogo%20Modelando%20como%20apoio%20ao%20ensino%20de%20Engenharia%20de%20Requisitos.pdf. Acesso em: 19 out. 2021.

SOMMERVILLE, I. **Engenharia de software**. São Paulo: Pearson, 2018.

SOUZA, L; MIRANDA, E; LUCENA, M; GOMES, A. **Aplicação da etnografia no contexto de fábrica de software na perspectiva da engenharia de requisitos**. WER, 2019.

Disponível em: http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER19/WER_2019_paper_33.pdf. Acesso em: 10 jul. 2021.

SOUZA, M; MAIA, P. H. M; ROBERTO, F. M; MAIA, C. L. B; BRASIL, M. M. A. **Uma análise quantitativa da utilização de técnicas de validação de requisitos no mercado de TI em Fortaleza**. RSC, 2016. Disponível em: <https://revistas.unifacs.br/index.php/rsc/article/view/3870>. Acesso em: 30 ago. 2021.

TAKEDA, H; VEERKAMP, P; TOMIYAMA, T; YOSHIKAWA, H. **Modeling design processes**. AI, 1990. Disponível em: <https://ojs.aaai.org/index.php/aimagazine/article/view/855>. Acesso em: 08 dez. 2021.

TREITEL, R. **Roadmap et Roadmapping**: tout ce que vous voulez savoir sur les *roadmaps* et vous n'avez jamais osé demander, 2005. Disponível em: <http://igart.free.fr/>. Acesso em: 14 dez. 2021.

TORO, A; PELÁEZ, L. E. **Ingeniería de Requisitos**: de la especificación de requisitos de *software* al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de *software* de la ciudad de Pereira. Scielo, 2016. Disponível em: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S190983672016000200016. Acesso em: 15 out. 2021.

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: Software** orientado a negócio. Rio de Janeiro: Brasport, 2016.

VENABLE, J; PRIES-HEJE, J; BASKERVILLE, R. **FEDS**: a framework for evaluation in Design Science Research. Springer, 2016. Disponível em: <https://link.springer.com/article/10.1057/ejis.2014.36>. Acesso em: 01 out. 2021.

VIEIRA, V. A. **As tipologias, variações e características da pesquisa de marketing**. FAE, 2002. Disponível em: <https://revistafae.fae.edu/revistafae/article/view/449>. Acesso em: 15 set. 2021.

WIEGERS, K; BEATTY, J. **Software Requirements**. Estados Unidos: Microsoft Press, 2013.

WIEMER, D; FISHER, D. **Applying secure software engineering (SSE) practices to critical space system infrastructure development**. SPACEOPS, 2016. Disponível em: <https://arc.aiaa.org/doi/pdf/10.2514/6.2016-2392>. Acesso em: 07 set. 2021.

YADAV, V; ADYA, M; NATH, D; SRIDHAR, V. **Considerations for effective requirements analysis in offshore software development projects**: lessons from multi-method research. AIS, 2016. Disponível em: <https://aisel.aisnet.org/cais/vol39/iss1/11/>. Acesso em: 10 out. 2021.

APÊNDICES

Apêndice A – Questionário práticas em requisitos de *software*

Apêndice B - Questionário de avaliação do *roadmap* proposto

APÊNDICE A – QUESTIONÁRIO PRÁTICAS EM REQUISITOS DE SOFTWARE

PESQUISA DE DISSERTAÇÃO DE MESTRADO - PRÁTICAS EM REQUISITOS DE *SOFTWARE* (PARTICIPANTES QUE ATUAM NO BRASIL)

Prezado (a) Participante,

Você está sendo convidado a participar da pesquisa “Melhores práticas em requisitos de *software*”.

Sua contribuição muito enriquecerá nosso trabalho, pois participando desta pesquisa, você nos trará uma visão específica pautada na sua experiência sobre o assunto.

Esclarecemos, contudo, que sua participação não é obrigatória. Sua recusa não trará nenhum prejuízo em sua relação com a pesquisadora ou com a instituição proponente.

Os objetivos deste estudo são avaliar quais das práticas de requisitos de *software* que foram identificadas na literatura são mais adotadas pelos profissionais da área, bem como, possibilitar a descoberta de demais práticas.

As informações obtidas por meio desta pesquisa serão confidenciais e asseguramos o sigilo sobre sua participação. Os dados serão divulgados de forma a não possibilitar sua identificação, protegendo e assegurando sua privacidade.

A qualquer momento você poderá tirar suas dúvidas sobre o projeto e sua participação.

Ao final desta pesquisa, o trabalho completo será disponibilizado no site do Programa de Mestrado.

Pesquisadora: Isabella de Araujo Cionini Menezes

E-mail: isabella.menezes@cspso.sp.gov.br

Orientador: Prof. Dr. Marcelo Duduchi Feitosa

E-mail: marcelo.feitosa@cspso.sp.gov.br

Declaro que entendi os objetivos de minha participação na pesquisa e concordo em participar. Registro também que concordo com o tratamento de meus dados pessoais para finalidade específica desta pesquisa, em conformidade com a Lei nº 13.709 – Lei Geral de Proteção de Dados Pessoais (LGPD).

☐ Concordo

☐ Não concordo

QUESTIONÁRIO

Dados demográficos da pesquisa:

1 – Selecione a UF em que se situa a empresa que você trabalha atualmente? (Sede da empresa/escritório no Brasil)

2 - Em que tipo de organização você atua?

- a) Organização privada
- b) Organização pública
- c) Organização mista

3 – Qual é o porte da organização?

- a) Micro – até 9 funcionários
- b) Pequena – de 10 a 49 funcionários
- c) Média – de 50 a 99 funcionários
- d) Grande – de 100 a 499 funcionários
- e) 500 ou mais funcionários

4 – Qual método de desenvolvimento de *software* é utilizado para a criação dos produtos?

- a) Método tradicional (Cascata, Processo Unificado e entre outros)
- b) Método ágil (Scrum, Extreme Programming e entre outros)
- c) Desenvolvimento misto

5 – Quais tipos de *software* são desenvolvidos na organização em que você trabalha?

- a) Aplicações web
- b) Aplicativos mobile
- c) Aplicações corporativas (Ex: CRM)
- d) Outras

6 – Qual a sua função na organização? (Permite mais de uma seleção)

- ☐ Analista de negócio
- ☐ Analista de sistemas
- ☐ Analista de testes
- ☐ CEO
- ☐ Consultor de sistemas
- ☐ Desenvolvedor/Programador de *software*
- ☐ Designer
- ☐ Gerente de Projetos
- ☐ Product Owner
- ☐ Scrum Master
- ☐ Suporte
- ☐ Outro. Qual? _____

Práticas em requisitos de *software*:

7 – Quais das práticas em requisitos de *software* listadas a seguir, são utilizadas pela organização? E em que fase geralmente são utilizadas? (Permite mais de uma seleção).

	Elicitar	Documentar	Validar	Gerenciar
Análise de personas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aplicação de questionários	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Chuva de ideias (Brainstorming)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Controle de mudanças/ Versionamento de requisitos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criação de casos de teste/ Teste de aceitação	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criação de casos de uso/ Artefatos criados com a UML	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Documentação de requisitos não funcionais	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entrevistas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Escrita de requisitos em forma de histórias de usuário	[]	[]	[]	[]
Etnografia	[]	[]	[]	[]
Grupo focal	[]	[]	[]	[]
Histórias de usuário são fornecidas pelo cliente	[]	[]	[]	[]
Observação de um sistema existente	[]	[]	[]	[]
Priorização dos requisitos	[]	[]	[]	[]
Prototipação	[]	[]	[]	[]
Rastreabilidade de requisitos	[]	[]	[]	[]
Realização de apresentações /Workshops	[]	[]	[]	[]
Repositório para os artefatos de especificação de requisitos de <i>software</i>	[]	[]	[]	[]
Reuso de artefatos de especificação de requisitos de <i>software</i>	[]	[]	[]	[]
Técnica INVEST	[]	[]	[]	[]
Técnica JAD	[]	[]	[]	[]
Técnica SMART	[]	[]	[]	[]
Trawling	[]	[]	[]	[]
Uso de <i>softwares</i> para o preparo da documentação do sistema/gestão dos requisitos	[]	[]	[]	[]
Uso de templates para documentação	[]	[]	[]	[]
Validação dos requisito	[]	[]	[]	[]
WebSite Design	[]	[]	[]	[]

8 – Que outra (s) práticas não listadas na questão anterior, você considera como melhores práticas em requisitos de *software*?

APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO DO *ROADMAP* PROPOSTO

QUESTIONÁRIO DE AVALIAÇÃO DO *ROADMAP*

Sobre a adoção do *roadmap* para a implantação da prática de “Uso de *software* para gestão dos requisitos” na empresa:

1 – O *roadmap* proporcionou a oportunidade de implantação de ferramentas e melhorias nas práticas em requisitos de *software*?

2 – Você considera que a utilização deste *roadmap* pode conduzir as empresas de desenvolvimento de sistemas de informação na utilização de boas práticas em requisitos de *software*? Comente.

3 – Quais as suas sugestões para a melhoria deste *roadmap*?