



**UNIVERSIDADE
FEDERAL
DE PERNAMBUCO**



Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Departamento de Eletrônica e Sistemas



Graduação em Engenharia Eletrônica

Robson Lima de Alencar

Estudo sobre a Especificação de Requisitos de Sistemas Embarcados com Metodologias Ágeis

Recife

2023

Robson Lima de Alencar

Estudo sobre a Especificação de Requisitos de Sistemas Embarcados com Metodologias Ágeis

Trabalho de Conclusão apresentado ao Curso de Graduação em Engenharia Eletrônica, do Departamento de Eletrônica e Sistemas, da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica.

Orientador(a): Prof. Fernanda Maria Ribeiro de Alencar, D.Sc.

Coorientador(a): Prof. Aêda Monalliza Cunha de Sousa, D.Sc.

Recife
2023

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Alencar, Robson Lima de.

Estudo sobre a Especificação de Requisitos de Sistemas Embarcados com
Metodologias Ágeis / Robson Lima de Alencar. - Recife, 2023.
81 : il., tab.

Orientador(a): Fernanda Maria Ribeiro de Alencar

Cooorientador(a): Aêda Monalliza Cunha de Sousa

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de
Pernambuco, Centro de Tecnologia e Geociências, Engenharia Eletrônica -
Bacharelado, 2023.

Inclui referências, apêndices, anexos.

1. Engenharia de Requisitos. 2. Sistemas Embarcados. 3. Metodologias
Ágeis. I. Alencar, Fernanda Maria Ribeiro de. (Orientação). II. Sousa, Aêda
Monalliza Cunha de. (Coorientação). III. Título.

620 CDD (22.ed.)

Robson Lima de Alencar

Estudo sobre a Especificação de Requisitos de Sistemas Embarcados com Metodologias Ágeis

Trabalho de Conclusão apresentado ao Curso de Graduação em Engenharia Eletrônica, do Departamento de Eletrônica e Sistemas, da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica.

Aprovado em: 12/05/2023

Banca Examinadora

Prof. Fernanda Maria Ribeiro de Alencar, D.Sc.
Universidade Federal de Pernambuco

Prof. Guilherme Nunes Melo, D.Sc.
Universidade Federal de Pernambuco

Prof. Hermano Andrade Cabral, Ph.D.
Universidade Federal de Pernambuco

*Aos meus pais Edmilson e
Milene que me incentivaram
desde o início e a minha
namorada Milena que me apoiou
em todos os momentos*

Agradecimentos

Gostaria de dedicar este trabalho em primeiro lugar à minha família, em especial aos meus pais Milene e Edmilson, e aos meus irmãos Richard e Emily, pelo apoio incondicional e incentivo ao longo de toda a minha vida. Suas palavras de encorajamento e motivação me ajudaram a superar os desafios e a manter o foco em meus objetivos.

Aos meus colegas de turma, que se tornaram grandes amigos no decorrer da graduação. A cooperação, o incentivo e a troca de experiências foram fundamentais para o meu crescimento pessoal e acadêmico. Sem a ajuda de vocês, este trabalho não teria sido possível.

Gostaria também de agradecer à minha namorada Milena, que sempre esteve ao meu lado durante todo o período de estudo e trabalho. Seu amor, compreensão e apoio foram essenciais para me manter motivado e focado em alcançar meus objetivos.

À Universidade Federal de Pernambuco por ter me proporcionado a formação no curso de Engenharia Eletrônica, a todos os professores do CTG e área 2, especialmente à professora Fernanda Alencar, cujo conhecimento, dedicação e orientação foram fundamentais para a realização deste trabalho. Sua capacidade de inspirar e guiar seus alunos é um exemplo a ser seguido por todos.

Por fim, agradeço a todos os que, direta ou indiretamente, contribuíram para a conclusão deste TCC. Seja por meio de discussões, compartilhamento de materiais ou apoio emocional, vocês foram fundamentais para a minha jornada acadêmica e para a realização deste trabalho.

O maior prazer na vida é fazer o
que os outros dizem que você não
pode fazer.

Nikola Tesla

Resumo do Trabalho de Conclusão de Curso apresentado ao Departamento de Eletrônica e Sistemas, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Eletrônica(Eng.)

Estudo sobre a Especificação de Requisitos de Sistemas Embarcados com Metodologias Ágeis

Robson Lima de Alencar

Os sistemas embarcados são extremamente importantes na sociedade moderna, pois estão presentes em uma ampla variedade de aplicações e dispositivos, desde os mais simples até os mais complexos. Na era da Internet das Coisas (IoT), ganham ainda mais importância, onde dispositivos conectados em rede exigem sistemas embarcados capazes de coletar, processar e transmitir dados de maneira rápida e confiável. A falta de clareza na especificação de requisitos em sistemas embarcados pode levar a problemas graves durante o processo de desenvolvimento, testes e implementação. Se forem utilizadas metodologias ágeis, essa abordagem pode ser ainda mais desafiadora. O objetivo desse estudo é de levantar as práticas da engenharia de requisitos que sejam aplicáveis a sistemas embarcados, além de verificar se esses sistemas podem ser desenvolvidos em metodologias ágeis. Foi realizada uma revisão sistemática da literatura com objetivo de levantar os estudos que tratam sobre esse tema e construir uma base de conhecimento relevante sobre a abordagem, de forma que torne possível chegar a conclusões. Os estudos mostraram que diversas técnicas da engenharia de requisitos podem ser utilizadas em sistemas embarcados, além disso, são abordadas diferentes etapas da engenharia de requisitos nesses sistemas. Concluiu-se que é possível especificar requisitos de sistemas embarcados, mas não há clareza ainda quando estes estão sendo desenvolvidos em metodologias ágeis.

Palavras-chave: Engenharia de requisitos; Metodologias ágeis; Sistemas embarcados.

Abstract of Course Conclusion Work, presented to Departament of Eletronic and Systems, as a partial fulfillment of the requirements for the degree of Bachelor of Electronic Engineering(Eng.)

Requirements Specification in Embedded Systems Development with Agile Methodologies

Robson Lima de Alencar

Embedded systems are extremely important in modern society as they are present in a wide variety of applications and devices, from the simplest to the most complex. In the era of the Internet of Things (IoT), they become even more important, where networked devices require embedded systems capable of collecting, processing, and transmitting data quickly and reliably. The lack of clarity in specifying requirements in embedded systems can lead to serious problems during the development, testing, and implementation process. If agile methodologies are used, this approach can be even more challenging. The objective of this study is to identify requirements engineering practices that are applicable to embedded systems, as well as to verify whether these systems can be developed using agile methodologies. A systematic literature review was conducted to identify studies that address this topic and to build a relevant knowledge base about the approach to reach conclusions. The studies showed that various requirements engineering techniques can be used in embedded systems, and different stages of requirements engineering are addressed in these systems. It was concluded that it is possible to specify requirements for embedded systems, but there is still no clarity when they are being developed using agile methodologies.

Keywords: Embedded systems; Requirements engineering; Agile methodologies.

Sumário

1	Introdução	13
1.1	Justificativa	14
1.2	Objetivo Geral	14
1.2.1	Objetivos específicos	15
1.3	Procedimentos Metodológicos	15
1.4	Organização do TCC	16
2	Fundamentação Teórica	17
2.1	Engenharia de Requisitos	18
2.2	Sistemas Embarcados	22
2.3	Metodologias ágeis	24
2.3.1	Scrum	26
2.3.2	Extreme Programming	27
2.3.3	Kanban	28
2.4	Considerações Finais	29
3	Revisão Sistemática da Literatura	30
3.1	Etapas da Revisão Sistemática da Literatura	31
3.1.1	Etapa de Planejamento	32
3.1.2	Etapa de Execução	35
3.1.3	Etapa de Análise	36
3.2	Considerações Finais	36

4	Análise dos Estudos	37
4.1	Resultados das Etapas da Revisão Sistemática	37
4.2	RQ1 - Quais etapas do processo de engenharia de requisitos estão presentes na especificação de sistemas embarcados?	47
4.3	RQ2 - Quais métodos, técnicas, processos e ferramentas da engenharia de requisitos são utilizados na especificação de sistemas embarcados?	53
4.4	RQ3: Sistemas embarcados podem ser desenvolvidos em metodologias ágeis? Se sim, como? Como são especificados?	59
4.5	Considerações Finais	62
5	Considerações finais	64
5.1	Conclusões	64
5.2	Dificuldades Encontradas	65
5.3	Trabalhos Futuros	66
	Referências	67
	Anexos	72
1	Lista de artigos finais da Revisão Sistemática da Literatura	72

Lista de Ilustrações

3.1	Etapas da Revisão Sistemática da Literatura	32
3.2	Etapas de execução da Revisão Sistemática da Literatura	35
4.1	Total por etapa da revisão sistemática	38
4.2	Total por <i>string</i> - Todos os resultados	39
4.3	Total por <i>string</i> - Após leitura de títulos	40
4.4	Total por <i>string</i> - Após leitura de resumos	40
4.5	Total por <i>string</i> - Resultado final	41
4.6	Total por ano - Todos os resultados	42
4.7	Total por ano - Resultado final	42
4.8	Total por base de dados - Todos os resultados	43
4.9	Total por base de dados - Após leitura de títulos	44
4.10	Total por basede dados - Após leitura de resumos	44
4.11	Total por base de dados - Resultado final	45
4.12	Total por tipo de estudo - Todos os Resultados	46
4.13	Total por tipo de estudo - Resultado final	46
4.14	RQ1 - Etapas do processo de Engenharia de Requisitos encontrados em sistemas embarcados	47
4.15	RQ2 - Técnicas da engenharia de requisitos na especificação de sistemas embarcados	54
4.16	RQ2 - Métodos e processos da engenharia de requisitos na especi- ficação de sistemas embarcados	56

4.17 RQ2 - Ferramentas da engenharia de requisitos na especificação de	
sistemas embarcados por ênfase	57

Capítulo 1

Introdução

A Engenharia de Requisitos (ER) é uma área essencial para o levantamento prévio da viabilidade e das necessidades dos *stakeholders* e das restrições sob as quais se encontra o sistema a ser desenvolvido (SOMMERVILLE, 2011). De modo geral, a má especificação de requisitos, com a ausência de documentação suficiente, pode ser um fator de insucesso dos projetos de sistemas.

Em projetos de sistemas embarcados (SE), esse processo de especificação de requisitos é ainda mais crítico, uma vez que é necessário lidar com restrições, por exemplo, de espaço e desempenho, além de requisitos específicos para o *hardware* envolvido. Por muitas vezes a ER é negligenciada em função da simplicidade aparente e da necessidade de agilidade do processo de desenvolvimento (SOUSA et al., 2015).

Nesse sentido, surgiram os métodos ágeis, abordagens de desenvolvimento de *software* que enfatizam a entrega incremental, a colaboração da equipe, o planejamento contínuo (SOMMERVILLE, 2011) e a necessidade de se estudar engenharia de requisitos em metodologias ágeis (LEFFINGWELL, 2011).

O estudo recente de Liebel (2023) mostrou que a adoção de práticas ágeis pode trazer benefícios significativos na Engenharia de Requisitos, como uma maior eficiência na entrega, redução de custos e erros, além de uma melhor colaboração entre as equipes envolvidas. Assim, quando se trata do desenvolvimento de sistemas embarcados, o uso de metodologias ágeis com Engenharia de Requisitos ainda é um desafio.

A motivação nesse estudo está na importância de se adotar Engenharia de Requisitos com práticas mais ágeis e o desenvolvimento de projetos de sistemas embarcados. Espera-se que, ao final deste trabalho, seja possível contribuir com o conhecimento sobre como especificar requisitos em sistemas embarcados, quando se utiliza métodos ágeis.

1.1 Justificativa

Uma das principais problemáticas na Engenharia de Requisitos, de modo geral em sistemas embarcados é como lidar com a falta de clareza e com a ambiguidade quando se especifica requisitos. Destaca-se que a fase de requisitos é a mais abstrata, ainda, dentro do desenvolvimento de *software* e que, normalmente, é realizada em linguagem natural. Diante desse cenário, é fundamental a realização de estudos que busquem compreender melhor a aplicação da Engenharia de Requisitos para o domínio de sistemas embarcados, como também buscar conhecimento sobre o desenvolvimento de sistemas embarcados utilizando-se metodologias ágeis. Deseja-se explorar a utilização dessas metodologias em sistemas embarcados. Para tanto, fez-se busca sistemática sobre estudos que tratem dessa temática. Com a análise dos resultados desses estudos, é possível identificar as melhores práticas de especificação nesse contexto específico. Dessa forma, contribuindo-se para a melhoria da qualidade dos sistemas embarcados desenvolvidos e para a redução de problemas como a falta de compreensão clara dos requisitos do sistema, atrasos na entrega, extrapolação de custos e dificuldades na manutenção.

1.2 Objetivo Geral

O objetivo geral deste trabalho é identificar como é feita a especificação de requisitos em sistemas embarcados, e compreender o seu desenvolvimento através de metodologias ágeis.

1.2.1 Objetivos específicos

Os objetivos específicos são mais detalhados e ajudam a atingir o objetivo geral. Dessa forma, propoe-se:

- Construir o conhecimento de base com o estado da arte sobre as práticas de Engenharia de Requisitos em projetos de sistemas embarcados; e também, sobre a utilização de metodologias ágeis no desenvolvimento de sistemas embarcados, através de uma revisão sistemática da literatura;
- Verificar quais são as metodologias ágeis mais utilizadas em projetos de sistemas embarcados e como elas são adaptadas à Engenharia de Requisitos nesse contexto;
- Analisar os resultados obtidos, comparando com abordagens mais tradicionais;
- Avaliar a eficácia das metodologias ágeis na melhoria da qualidade dos sistemas desenvolvidos para sistemas embarcados, considerando fatores como a clareza dos requisitos, entrega no prazo e manutenção.

1.3 Procedimentos Metodológicos

Trata-se de uma pesquisa exploratória que tem por finalidade proporcionar mais informações sobre como é realizada a especificação de requisitos nos sistemas embarcados que são desenvolvidos com metodologias ágeis. Para a condução dessa pesquisa, adotou-se a Revisão Sistemática da Literatura (RSL), que busca evidências de estudos já publicados na área, cujo o rigor do processo de busca é um fator que a distingue de outros tipos de revisões.

O método hipotético-dedutivo é um interessante mecanismo que foi utilizado. Assim, a hipótese considerada é a de que é possível se especificar requisitos em sistemas embarcados, quando esses são desenvolvidos através de métodos ágeis. Para isso, busca-se formas de fazê-lo.

Sobre a natureza da pesquisa, trata-se de uma pesquisa básica, ou seja, tem como foco principal gerar conhecimentos novos úteis para o avanço da ciência sem aplicações práticas previstas. A forma de abordagem desse estudo é o de pesquisa quantitativa, que trata tudo como quantificável e foca em traduzir os resultados em números e classificá-los para extrair o máximo de informação possível.

1.4 Organização do TCC

Além desse capítulo introdutório, o presente trabalho está organizado em quatro outros capítulos. No capítulo 2, trata-se a Fundamentação Teórica, na qual são abordados conceitos fundamentais sobre a área de Engenharia de Requisitos, Sistemas Embarcados e Metodologias Ágeis. No capítulo 3, descreve-se a Revisão sistemática da literatura realizada para coletar os principais estudos primários acerca de como se especificar os requisitos de sistemas embarcados desenvolvidos com metodologias ágeis. Esse capítulo é fundamental para compreender como os estudos foram coletados, selecionados e analisados. No capítulo 4, apresentam-se a análise dos estudos selecionados, apontando-se principais características, limitações e tendências futuras. Por fim, no capítulo 5 são apresentadas as principais conclusões, e algumas limitações deste estudo e possíveis direções para futuras pesquisas.

Capítulo 2

Fundamentação Teórica

A fundamentação teórica do trabalho está organizada em torno dos seguintes temas: Engenharia de Requisitos, Sistemas Embarcados e Metodologias Ágeis.

Na seção de Engenharia de Requisitos, são abordados conceitos e processos envolvidos na elicitação, análise e validação de requisitos de *software*, visando garantir que o produto final atenda às necessidades do usuário e do negócio. Também são exploradas técnicas e ferramentas utilizadas para a documentação e gestão de requisitos.

Na seção de Sistemas Embarcados, são apresentados conceitos e características específicas desse tipo de sistema, que geralmente é projetado para realizar tarefas específicas com recursos limitados. São também apresentados exemplos de aplicações que utilizam sistemas embarcados, como automóveis, dispositivos médicos, eletrodomésticos inteligentes, entre outros.

Na seção de Metodologias Ágeis, são discutidas as principais características desse tipo de abordagem de desenvolvimento de *software*, que valoriza a interação e colaboração entre as equipes e o cliente, a entrega incremental e contínua de funcionalidades, a adaptação a mudanças e a busca pela excelência técnica. São apresentados exemplos de metodologias ágeis, como Scrum, XP e Kanban.

2.1 Engenharia de Requisitos

A engenharia de requisitos continua sendo uma disciplina fundamental na área de desenvolvimento de *software* nos dias atuais (GUPTA et al., 2020). A engenharia de requisitos é um processo que visa identificar, documentar e gerenciar os requisitos de um sistema de *software*, tendo em vista as necessidades dos usuários e do negócio (MELEGATI et al., 2019) (CURCIO et al., 2018).

Além disso, a engenharia de requisitos é uma atividade interativa e contínua que deve ocorrer ao longo do ciclo de vida do *software*. Isso significa que, para garantir que os requisitos do sistema estejam corretos, completos, consistentes e precisos, a engenharia de requisitos envolve diversas atividades, como elicitação, análise e negociação de requisitos, documentação e validação de requisitos (CURCIO et al., 2018) (MAALEM e ZAROOUR, 2016)

Ademais, a Engenharia de Requisitos é uma fase crucial no processo de desenvolvimento de *software* e deve ser tratada como um processo contínuo que envolve a participação de todas as partes interessadas no sistema (ALSANOOSY et al., 2020). Isso destaca a importância da colaboração e comunicação efetiva entre a equipe de desenvolvimento e os *stakeholders* para garantir que os requisitos do sistema sejam atendidos de maneira efetiva (HOSSEINI et al., 2018).

Em resumo, a engenharia de requisitos continua sendo um processo vital no desenvolvimento de *software*, com uma abordagem sistemática e interativa que deve ser realizada ao longo do ciclo de vida do *software* para garantir que os requisitos do sistema atendam às necessidades dos usuários e do negócio (CURCIO et al., 2018).

Se tratando das etapas da engenharia de requisitos, a elicitação de requisitos é a primeira, que busca identificar e coletar as necessidades do sistema através de técnicas como entrevistas, questionários e observação de usuários (BURNAY et al., 2014).

A análise de requisitos é a segunda etapa, que envolve a revisão e validação dos requisitos coletados na etapa anterior, identificando inconsistências e omissões nos requisitos e utilizando técnicas como análise de casos de uso e prototipagem.

A documentação de requisitos é a terceira etapa, que consiste em documentar os requisitos em um formato claro e compreensível. É fundamental garantir que os requisitos estejam completos, corretos, verificáveis e consistentes, e técnicas como especificação formal e modelagem de requisitos podem ser utilizadas. A validação de requisitos é a quarta etapa, que visa verificar se os requisitos definidos atendem às expectativas do cliente. Para isso, são utilizadas técnicas como revisões de requisitos, testes de aceitação e validação de protótipos. Por fim, o gerenciamento de requisitos é a última etapa, que consiste em manter os requisitos atualizados e gerenciar mudanças ao longo do tempo, garantindo que os requisitos continuem atendendo às necessidades do cliente. Essa etapa é importante para o sucesso do projeto e pode incluir o uso de ferramentas de gerenciamento de requisitos (SOMMERVILLE, 2011).

Para facilitar o gerenciamento de requisitos, diversas técnicas e ferramentas podem ser utilizadas, como prototipagem, modelagem de requisitos, diagramas de caso de uso e ferramentas de rastreabilidade de requisitos. Existem várias técnicas para apoiar cada uma das etapas. Algumas técnicas amplamente utilizadas incluem:

- Entrevistas: uma técnica comum de elicitación de requisitos, na qual um analista de requisitos realiza entrevistas com *stakeholders* do sistema para obter informações sobre seus requisitos (BURNAY et al., 2014);
- Brainstorming: uma técnica colaborativa para geração de ideias, onde os *stakeholders* trabalham juntos para gerar ideias e discutir os requisitos do sistema (YANG et al., 2023); (MICH et al., 2023);
- Casos de Uso: uma técnica para representar e documentar requisitos em termos de interações entre atores e o sistema (SOMÉ, 2006);
- Prototipagem: uma técnica de design iterativo em que modelos iniciais do sistema são desenvolvidos para validar e refinar os requisitos (KANG et al., 2023);

- Análise de Requisitos Orientada a Objetos (OOA): uma técnica de modelagem de requisitos que usa conceitos de orientação a objetos para descrever as interações entre objetos no sistema; (MAREW et al., 2009);
- Diagramas de Fluxo de Dados (DFD): uma técnica para visualizar o fluxo de informações em um sistema, identificando entradas, saídas e processos envolvidos (CHEEMA et al., 2023);
- Inspeção de Requisitos: uma técnica formal de revisão de requisitos, na qual um grupo de revisores examina documentos de requisitos para detectar problemas de qualidade (TAKAYA et al., 2016).

Essas são apenas algumas das técnicas disponíveis para a engenharia de requisitos, e cada uma tem suas vantagens e limitações. A seleção da técnica apropriada depende do contexto do projeto e dos requisitos específicos do sistema. Em resumo, a engenharia de requisitos é uma disciplina fundamental no processo de desenvolvimento de *software* e deve ser tratada como um processo contínuo e iterativo. O uso de técnicas e ferramentas adequadas pode ajudar a garantir a qualidade dos requisitos do sistema e, conseqüentemente, a qualidade do *software* entregue ao cliente.

Cada etapa da engenharia de requisitos requer uma abordagem cuidadosa e a utilização de técnicas específicas para garantir a qualidade dos requisitos do sistema. Ao seguir um processo estruturado de engenharia de requisitos, é possível garantir que o sistema desenvolvido atenda às necessidades e expectativas do cliente, além de minimizar os riscos e custos associados ao desenvolvimento de *software* (AHMAD et al., 2023).

A elaboração de requisitos pode se apresentar como um grande desafio, principalmente em projetos que envolvem sistemas complexos. Para auxiliar na definição de requisitos, é utilizada a engenharia de requisitos, que compreende um conjunto de práticas que abrange desde a identificação até a documentação dos requisitos de um sistema computacional (SOMMERVILLE, 2011). A Engenharia de Requisitos pode ser tratada como:

- Uma condição ou capacidade necessária para um usuário resolver um problema ou alcançar um objetivo;
- Uma condição ou capacidade que deve ser alcançada ou estar presente em um sistema ou componente de sistema para satisfazer um contrato, norma, especificação ou outro documento formalmente imposto;
- Uma representação documentada de uma condição ou capacidade como nos dois itens anteriores. (IEEE, 1990)

Obter resultados satisfatórios na especificação de requisitos requer a utilização de métodos bem definidos, como destaca a engenharia de requisitos. Segundo outra definição, a engenharia de requisitos é um ramo da engenharia de sistemas que engloba o descobrimento, desenvolvimento, análise, qualificação, comunicação e gestão de requisitos em diferentes níveis de abstração, com o objetivo de definir o sistema de forma adequada. (HULL et al., 2010)

O analista de requisitos é uma figura-chave na engenharia de requisitos, sendo responsável por identificar, elicitar, documentar, validar e gerenciar os requisitos de um sistema. Segundo Sommerville (SOMMERVILLE, 2011), o analista de requisitos é o intermediário entre os *stakeholders* e a equipe de desenvolvimento, tendo a função de entender as necessidades dos usuários e traduzi-las em requisitos claros e precisos. Para isso, o analista de requisitos utiliza diversas técnicas, como entrevistas, questionários, prototipagem, entre outras.

Além de ser um facilitador da comunicação entre as partes interessadas, o analista de requisitos também deve ser capaz de avaliar a viabilidade e a adequação dos requisitos propostos, levando em consideração as restrições técnicas, orçamentárias e de prazo (RAGO et al., 2019). Para garantir a qualidade dos requisitos, o analista deve realizar atividades de validação, como revisões técnicas e testes de aceitação (MAHAJU et al., 2023).

Outra função importante do analista de requisitos é gerenciar os requisitos ao longo do ciclo de vida do sistema. Isso inclui rastrear mudanças nos requisitos,

avaliar o impacto dessas mudanças, manter a rastreabilidade dos requisitos e gerenciar conflitos entre os requisitos (SOMMERVILLE, 2011). O analista também deve garantir a integridade e a consistência dos requisitos, evitando ambiguidades, inconsistências e omissões.

Em resumo, o analista de requisitos desempenha um papel fundamental na engenharia de requisitos, sendo responsável por identificar e documentar as necessidades dos usuários e traduzi-las em requisitos claros e precisos. Para isso, o analista utiliza diversas técnicas de elicitación e validação de requisitos, além de ser capaz de gerenciar os requisitos ao longo do ciclo de vida do sistema (MAHAJU et al., 2023).

2.2 Sistemas Embarcados

Um sistema embarcado é um sistema com capacidades computacionais, destinado à execução de uma ou mais atividades específicas dentro de um sistema maior, com dedicação exclusiva para esse sistema. Sem dúvida, os sistemas embarcados trouxeram progressos relevantes para a sociedade, introduzindo o conceito da Internet das Coisas ou “Internet of Things” (IoT) e impactando amplamente no design, na construção, na implantação e na evolução dos sistemas como um todo (AL-FUQAHA et al., 2015).

Os sistemas embarcados são sistemas computacionais que possuem uma ou mais funcionalidades específicas e são destinados a realizar uma tarefa em particular. Eles estão presentes em diversos equipamentos eletrônicos utilizados no dia a dia, como em automóveis, dispositivos médicos, eletrodomésticos, entre outros. A crescente demanda por soluções cada vez mais inteligentes e automatizadas em diferentes setores da indústria impulsionou o desenvolvimento de sistemas embarcados (SILVA, 2019).

O estudo de sistemas embarcados é fundamental para acompanhar as tendências do mercado tecnológico, tendo em vista que esses sistemas estão presentes em uma grande variedade de dispositivos e máquinas que fazem parte do cotidiano das pessoas. Conhecer as principais características e recursos desses sistemas, bem como

suas aplicações práticas, pode ampliar as possibilidades de atuação profissional na área de tecnologia da informação (MIRITZ¹, 2016).

Além disso, o desenvolvimento de sistemas embarcados requer um conjunto específico de habilidades técnicas, como conhecimentos em eletrônica, programação, arquitetura de sistemas, entre outros. A aquisição dessas habilidades pode proporcionar uma ampla gama de oportunidades de carreira, desde o desenvolvimento de produtos para indústrias específicas até a pesquisa em áreas como robótica e inteligência artificial (PRATES, 2012).

Dessa forma, o estudo de sistemas embarcados se torna uma área de grande interesse para profissionais e estudantes da área de tecnologia da informação. A busca por conhecimentos e habilidades relacionados a sistemas embarcados pode proporcionar novas perspectivas de atuação profissional e contribuir para o desenvolvimento de soluções tecnológicas cada vez mais avançadas e eficientes (PRATES, 2012).

Nos últimos anos, tem havido um aumento na aplicação de sistemas embarcados em diferentes campos, como saúde, transporte, segurança e agricultura. A implementação de sistemas embarcados na área da saúde, por exemplo, tem permitido o monitoramento remoto de pacientes e a coleta de dados de saúde em tempo real. Na área de transporte, os sistemas embarcados têm sido utilizados em veículos autônomos para aprimorar a segurança e a eficiência no tráfego. Na agricultura, esses sistemas têm sido aplicados em máquinas agrícolas para otimizar a produtividade e reduzir o desperdício de recursos (NASCIMENTO, 2020).

Um exemplo de sistema embarcado amplamente utilizado é o microcontrolador Arduino, que é capaz de controlar diversos dispositivos eletrônicos, como sensores de temperatura, motores e iluminação (MCROBERTS, 2011).

Outro exemplo de sistema embarcado é o Raspberry Pi, que é um computador de placa única, desenvolvido para fins educacionais e projetos de computação de baixo custo. Ele pode ser usado para uma variedade de aplicações, como servidores de mídia, consoles de jogos, dispositivos de streaming de vídeo e sistemas de automação residencial (KURNIAWAN, 2019).

Esses são apenas alguns exemplos de sistemas embarcados que estão presentes em diversos aspectos da vida cotidiana. A evolução contínua da tecnologia tem permitido o desenvolvimento de sistemas cada vez mais avançados e eficientes, com novas aplicações e possibilidades. Em suma, sistemas embarcados são parte fundamental de nossa sociedade e seu uso tende a crescer cada vez mais. O conhecimento e desenvolvimento desses sistemas devem ser conduzidos de forma consciente e sustentável, a fim de garantir sua eficiência e contribuir para um futuro mais tecnológico e responsável.

2.3 Metodologias ágeis

Os métodos ágeis de desenvolvimento surgiram com o objetivo de atender às necessidades do mercado em constante mudança, em que a rapidez na entrega de um produto de qualidade é essencial. Esse modelo de desenvolvimento se baseia em ciclos curtos de trabalho, em que a equipe de desenvolvimento é altamente colaborativa e autogerenciável, com foco em atender às necessidades do cliente de forma eficiente (FADEL e SILVEIRA, 2010).

Um dos principais motivos para o uso de métodos ágeis é a capacidade de adaptar-se rapidamente às mudanças no mercado e as necessidades dos clientes. O ambiente de negócios é altamente dinâmico e, com o uso de métodos ágeis, as empresas podem ajustar suas estratégias e prioridades com base nas mudanças do mercado. Além disso, os métodos ágeis permitem que os desenvolvedores entreguem produtos de alta qualidade em um curto espaço de tempo, o que é essencial em um mercado em que a velocidade é um fator crítico de sucesso (FADEL e SILVEIRA, 2010).

Outro motivo para o uso de métodos ágeis é a possibilidade de melhorar a comunicação e a colaboração entre as equipes de desenvolvimento e os clientes. O modelo ágil enfatiza a comunicação constante entre a equipe de desenvolvimento e o cliente, permitindo que as necessidades do cliente sejam entendidas e incorporadas ao produto de forma mais efetiva. Isso também ajuda a reduzir o retrabalho, já que

as mudanças de requisitos podem ser incorporadas ao produto de forma mais rápida e eficiente (PRIKLADNICKI et al., 2014).

Por fim, os métodos ágeis promovem uma cultura de melhoria contínua, em que a equipe de desenvolvimento é incentivada a identificar e corrigir falhas de forma rápida e eficiente. Essa abordagem ajuda a reduzir o risco de fracasso do projeto, já que as falhas são identificadas e corrigidas precocemente (PRIKLADNICKI et al., 2014).

Em suma, os métodos ágeis de desenvolvimento são uma alternativa viável para empresas que buscam maior agilidade e eficiência no desenvolvimento de *software*. Essa abordagem permite que as empresas adaptem-se rapidamente às mudanças do mercado e das necessidades dos clientes, entregando produtos de alta qualidade em um curto espaço de tempo, além de promover uma cultura de melhoria contínua.

Se referindo mais a área de *software*, metodologias ágeis são um conjunto de métodos que se concentram em um processo iterativo e incremental para entregar *software* de alta qualidade de forma rápida e flexível (SANCHES, 2019). Essas metodologias surgiram como uma resposta às limitações dos métodos tradicionais de desenvolvimento de *software*, que muitas vezes eram lentos, burocráticos e inflexíveis em relação às mudanças de requisitos do cliente.

A adoção de métodos ágeis em desenvolvimento de *software* tem aumentado significativamente nos últimos anos, sendo amplamente adotados em empresas de todos os tamanhos e setores. Isso se deve ao fato de que os métodos ágeis são mais eficientes em relação aos custos e prazos de entrega, oferecem uma maior flexibilidade para mudanças nos requisitos do cliente e proporcionam uma maior satisfação do cliente (NUNES, 2017). No entanto, a adoção de métodos ágeis de desenvolvimento de *software* também pode apresentar desafios, como a necessidade de uma comunicação eficaz e colaborativa entre os membros da equipe, a necessidade de uma gestão de projeto mais flexível e a necessidade de uma equipe altamente qualificada e experiente.

Em resumo, elas são uma abordagem eficiente e flexível para o desenvolvimento

de *software*, que permite uma entrega mais rápida e eficiente de *software* de alta qualidade. A adoção desses métodos pode ser uma vantagem competitiva para as empresas, desde que sejam realizados de forma adequada e com uma equipe qualificada.

2.3.1 Scrum

Scrum é uma metodologia ágil de desenvolvimento de *software* que tem sido amplamente adotada em diversas empresas ao redor do mundo. Ela se baseia em ciclos curtos de trabalho, chamados de *sprints*, onde o objetivo é entregar uma funcionalidade completa do produto ao final de cada *sprint* (STOPA e RACHID, 2019).

O Scrum é baseado em três papéis principais: o Product Owner, responsável por definir as funcionalidades do produto e priorizar as atividades de desenvolvimento; o Scrum Master, responsável por garantir a aplicação das práticas e princípios do Scrum e apoiar a equipe de desenvolvimento; e a equipe de desenvolvimento, responsável por realizar as atividades de desenvolvimento e entregar o produto (STOPA e RACHID, 2019).

Uma das principais motivações para a adoção do Scrum é a flexibilidade que a metodologia proporciona, permitindo que a equipe se adapte facilmente a mudanças nos requisitos do produto e prioridades de desenvolvimento. Além disso, o Scrum também promove a comunicação constante entre a equipe de desenvolvimento e o cliente, permitindo que o produto final seja mais alinhado às expectativas do cliente (HRON e OBWEGESER, 2022).

Outra motivação para a adoção do Scrum é a possibilidade de aumentar a eficiência e a qualidade do desenvolvimento de *software*. Ao dividir o trabalho em *sprints* curtos e focar em entregar funcionalidades completas, a equipe é capaz de avaliar constantemente o progresso do projeto e identificar possíveis melhorias no processo de desenvolvimento (DIMES, 2014).

Em resumo, o Scrum é uma metodologia ágil de desenvolvimento de *software*

que tem sido adotada em diversas empresas ao redor do mundo. A flexibilidade, a comunicação constante com o cliente e a possibilidade de aumentar a eficiência e a qualidade do desenvolvimento são algumas das principais motivações para a adoção do Scrum.

2.3.2 Extreme Programming

A metodologia Extreme Programming (XP) é um conjunto de práticas de desenvolvimento ágil de *software* que enfatiza a comunicação constante, o *feedback* rápido e a adaptação às mudanças durante todo o processo de desenvolvimento. Criada em 1996 por Kent Beck, a XP se concentra na qualidade do código, no envolvimento do cliente e na entrega contínua de pequenas funcionalidades (BEC, 2002).

Uma das principais características da XP é a programação em par, onde dois programadores trabalham juntos em um mesmo computador, para aumentar a qualidade do código e promover o compartilhamento de conhecimento. Outra prática importante é a integração contínua, onde o código é integrado ao repositório, várias vezes ao dia, para evitar conflitos e identificar problemas rapidamente (BEC, 2002).

A XP também enfatiza a importância dos testes automatizados, que são escritos antes mesmo do código ser desenvolvido, garantindo que o *software* funcione corretamente mesmo após mudanças no código. Além disso, a metodologia incentiva a utilização de histórias de usuário para definir as funcionalidades que serão desenvolvidas, tornando o processo mais centrado no cliente (SHRIVASTAVA et al., 2021).

Dentre os benefícios da adoção da metodologia XP, destaca-se a redução de custos e tempo de desenvolvimento, além da entrega mais rápida e frequente de funcionalidades para o cliente. No entanto, a XP também requer uma equipe altamente colaborativa e comprometida, além de um forte envolvimento do cliente no processo de desenvolvimento (SHRIVASTAVA et al., 2021).

2.3.3 Kanban

A metodologia Kanban é um método ágil de gestão de projetos que utiliza um quadro visual para gerenciar o fluxo de trabalho. Criado originalmente para a indústria automotiva, o Kanban tem sido amplamente adotado em diversas áreas, incluindo desenvolvimento de *software*, gerenciamento de projetos e serviços de TI (ZAYAT e SENVAR, 2020).

O método Kanban é baseado em três princípios fundamentais: visualizar o trabalho, limitar o trabalho em progresso e gerenciar o fluxo. Isso é feito através de um quadro que mostra o fluxo de trabalho em colunas, com cada tarefa sendo representada por um cartão. O objetivo é maximizar a eficiência, eliminando desperdícios e reduzindo o tempo de ciclo de trabalho (ZAYAT e SENVAR, 2020).

Uma das principais vantagens do método Kanban é a sua adaptabilidade. Como não há papéis ou processos rígidos a serem seguidos, o Kanban permite que as equipes se ajustem de forma flexível às mudanças nos requisitos do projeto ou nas condições de trabalho. Além disso, o método enfatiza a colaboração e a comunicação, o que pode levar a uma maior transparência e engajamento da equipe (ALAIDAROS et al., 2021).

No entanto, o sucesso do método Kanban depende da sua correta implementação. É importante que a equipe esteja ciente dos princípios do método e que haja uma cultura de melhoria contínua, para que o fluxo de trabalho seja otimizado e os resultados sejam alcançados de forma eficiente (ALAIDAROS et al., 2021).

Em resumo, a metodologia Kanban é uma opção interessante para equipes que buscam aumentar a eficiência do seu fluxo de trabalho e melhorar a comunicação e colaboração entre os membros da equipe. Com a sua adaptabilidade e ênfase na melhoria contínua, o Kanban pode ser uma ferramenta valiosa para equipes em diversos setores e projetos.

2.4 Considerações Finais

Neste capítulo foram apresentados os principais fundamentos referentes às três áreas de pesquisa envolvidas: Engenharia de Requisitos; Sistemas Embarcados; e, Metodologias Ágeis.

No processo de engenharia de requisitos que inclui um estudo de viabilidade, o levantamento e análise, documentação (especificação), validação e gerenciamento daquilo que se quer que o sistema faça e da definição das restrições em sua operação e implementação. Sendo assim, uma etapa de fundamental importância para o sucesso do desenvolvimento. Apesar do processo de ER ter sido proposto no contexto nos métodos tradicionais de desenvolvimento de *software*, vê-se hoje sua necessidade em outros métodos, como nos métodos ágeis, e no desenvolvimento de sistemas.

Hoje, apesar dos métodos terem sido propostos para uso em desenvolvimento de sistemas pequenos e médio portes, onde se tem pequenas equipes, observa-se sistemas cada vez mais complexos sendo construídos com métodos ágeis. Exatamente nesse sentido é que se quer examinar como se faz a especificação de sistemas embarcados, que são desenvolvidos com esses métodos.

Capítulo 3

Revisão Sistemática da Literatura

Esse trabalho tem como objetivo explorar a especificação de requisitos em sistemas embarcados, como essa especificação é realizada quando o método de desenvolvimento ágil é utilizado. Tomou-se como hipótese que é possível se especificar requisitos em sistemas embarcados, quando desenvolvidos em métodos ágeis.

Assim, realizou-se uma revisão sistemática da literatura com base em KITCHENHAM et al. (2009), para buscar evidência em estudos publicados, que apresentassem as formas de especificar requisitos de sistemas embarcados, quando se usa metodologias ágeis. Observando-se os resultados dessas práticas para identificar as potencialidades e limitações.

Antes de se iniciar a revisão sistemática, foi realizado um estudo exploratório abrangendo os termos de engenharia de requisitos, sistemas embarcados e metodologias ágeis, bem como a relação existente entre eles.

A realização do estudo exploratório permitiu uma visão mais ampla sobre as diferentes abordagens de engenharia de requisitos que podem ser aplicadas em projetos de sistemas embarcados, bem como a forma como as metodologias ágeis podem ser utilizadas em conjunto com essas abordagens, para obter um melhor desempenho do processo de desenvolvimento.

Dessa forma, a partir desse estudo exploratório, tornou-se possível identificar com mais clareza quais são as lacunas existentes na área de engenharia de requisitos em sistemas embarcados e quais os possíveis benefícios que podem ser alcançados

quando estes são desenvolvidos com metodologias ágeis. Assim, a revisão sistemática realizada foi embasada em um conhecimento mais amplo e crítico sobre o tema, o que proporcionou uma análise mais profunda e objetiva dos estudos selecionados.

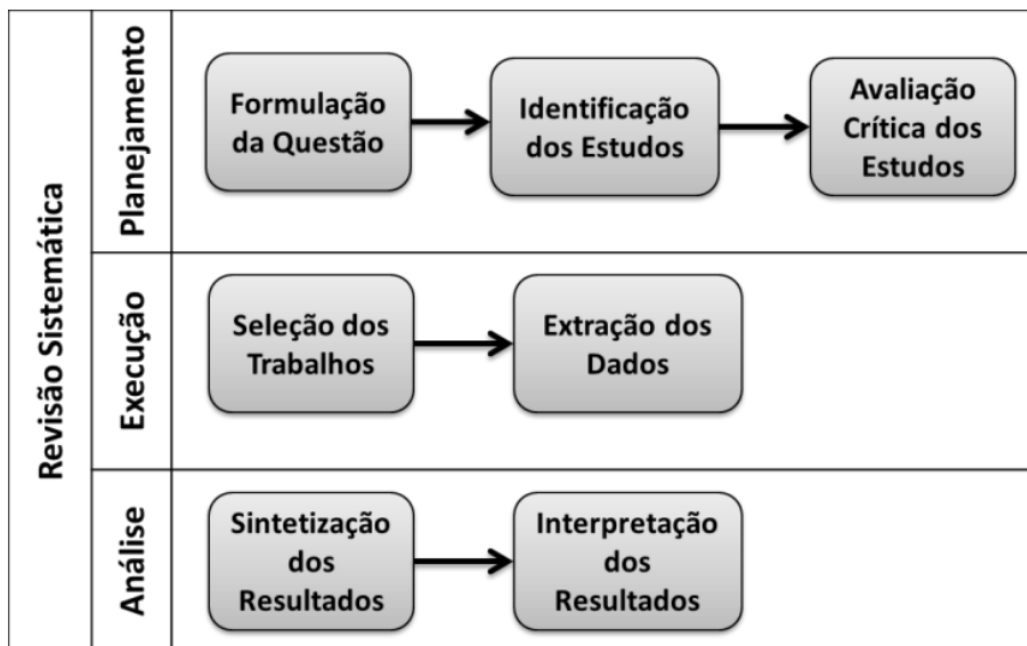
3.1 Etapas da Revisão Sistemática da Literatura

A revisão sistemática da literatura é uma técnica amplamente utilizada na pesquisa científica para coletar, avaliar e sintetizar evidências relevantes sobre um determinado tema. Ela é considerada uma ferramenta importante para auxiliar na tomada de decisões em diversas áreas, incluindo a saúde, a educação e a tecnologia (KITCHENHAM et al., 2009).

A motivação para realizar uma revisão sistemática da literatura pode ser variada. Por exemplo, a necessidade de responder a uma pergunta específica de pesquisa, avaliar a qualidade dos estudos existentes, identificar lacunas na literatura e propor novas áreas de investigação. Além disso, a revisão sistemática também pode ajudar a identificar áreas de consenso e de controvérsia em uma determinada área. (PAUL et al., 2021).

É importante que a revisão sistemática seja realizada com base em critérios objetivos e transparentes, de forma a garantir a confiabilidade e validade dos resultados. Além disso, é utilizada em diversas áreas, incluindo medicina, enfermagem, psicologia, educação, engenharia, entre outras. É uma técnica que vem sendo cada vez mais valorizada na produção científica, uma vez que permite a organização sistemática das informações e o estabelecimento de uma base sólida para o desenvolvimento de novas pesquisas.

Para realizar esta Revisão sistemática da Literatura, utilizamos foi utilizado um modelo cujo processo envolve diversas atividades agrupadas em três fases principais: planejamento; execução; e análise (KITCHENHAM et al., 2009) (BIOLCHINI et al., 2005). A Figura 3.1 ilustra o processo.

Figura 3.1: Etapas da Revisão Sistemática da Literatura

Fonte: (SOUSA et al., 2015)

3.1.1 Etapa de Planejamento

Com esse estudo tendo sido feito, o primeiro passo foi de identificar perguntas chave para trazer respostas à hipótese inicialmente levantada. Assim, foram selecionadas algumas perguntas para a execução da pesquisa, conforme observa-se a seguir.

- RQ1: Quais etapas do processo de engenharia de requisitos estão presentes na especificação de sistemas embarcados?
- RQ2: Quais métodos, técnicas, processos e ferramentas da engenharia de requisitos são utilizados na especificação de sistemas embarcados?
- RQ3: Sistemas embarcados podem ser desenvolvidos em metodologias ágeis? Se sim, como? Como são especificados?

A primeira pergunta de pesquisa (RQ1) tem como objetivo identificar quais são as etapas do processo de engenharia de requisitos que estão presentes em projetos de sistemas embarcados. É importante entender como essas etapas são aplicadas

nestes projetos em questão, a fim de determinar as diferenças em relação a outros tipos de projetos de *software*.

A segunda pergunta de pesquisa (RQ2) busca identificar quais são os métodos, técnicas, processos e ferramentas da engenharia de requisitos, utilizados em projetos de sistemas embarcados. Com essa informação, é possível avaliar a efetividade dessas abordagens e determinar quais são as melhores práticas para o processo.

A terceira pergunta de pesquisa (RQ3) tem como objetivo identificar se sistemas embarcados podem ser desenvolvidos em metodologias ágeis. Além disso, deseja-se encontrar informações sobre como estes são desenvolvidos e como são especificados nesse cenário. É importante entender a aplicabilidade das metodologias ágeis em sistemas embarcados, pois conforme já mencionado anteriormente, eles apresentam desafios específicos comparado a outros tipos de projetos de *software*.

Para realizar a revisão sistemática, foram construídas *strings* de pesquisa, que serviram como base para a busca nas bases de dados disponíveis, tais como: ACM, IEEE, Scopus e Periódicos Capes.

Para a elaboração dessas *strings*, foram considerados os conceitos fundamentais da engenharia de requisitos, sistemas embarcados e metodologias ágeis. Além disso, foram utilizados sinônimos e termos relacionados a esses conceitos, para garantir que a pesquisa fosse ampla e abrangente, permitindo a inclusão do maior número possível de trabalhos relevantes.

Esse processo de seleção de termos permitiu a criação de *strings* de pesquisa mais abrangentes e precisas, que foram utilizadas para buscar por métodos, técnicas, ferramentas e outros temas significativos para a pesquisa. A escolha criteriosa desses termos foi fundamental para garantir que a busca fosse eficiente e que os resultados encontrados fossem pertinentes e úteis para a pesquisa.

Os termos utilizados estão listados a seguir.

- “embedded systems” OR “critical systems” OR “real time systems” OR “embedded *software*” OR “computational systems”;
- “requirements elicitation” OR “requirements engineering” OR “requirements

specification” OR “requirements management”;

- “Agile methodology” OR “Agile method” OR “Agile approach” OR “Scrum” OR “Lean” OR “Kanban” OR “Extreme programming” OR “Dynamic System Development Model”.

Através dos termos, foram elaboradas *strings* de busca para usar nas bases de dados. As *strings* estão listadas a seguir.

- (“embedded systems” OR “critical systems” OR “real time systems” OR “embedded *software*” OR “computational systems”) AND (“requirements elicitation” OR “requirements engineering” OR “requirements specification” OR “requirements management”);
- (“embedded systems” OR “critical systems” OR “real time systems” OR “embedded *software*” OR “computational systems”) AND (“requirements elicitation” OR “requirements engineering” OR “requirements specification” OR “requirements management”) AND (“Agile methodology” OR “Agile method” OR “Agile approach” OR “Scrum” OR “Lean” OR “Kanban” OR “Extreme programming” OR “Dynamic System Development Model”).

Levando em consideração as *strings* de busca já definidas, o próximo passo foi pesquisar essas *strings* nas bases de dados escolhidas.

Com as *strings* de busca desenvolvidas e aplicadas nas bases de dados, foi possível identificar um grande número de trabalhos relacionados ao tema deste trabalho. Entretanto, é importante ressaltar que a simples coleta de um grande número de trabalhos não é suficiente para atingir os objetivos propostos. É necessário realizar uma filtragem rigorosa para que apenas os artigos relevantes e de qualidade sejam selecionados para análise completa.

Foram definidos critérios de inclusão e exclusão, que foram aplicados na etapa de triagem inicial dos resultados. Esses critérios foram estabelecidos com base nos

objetivos específicos e na questão de pesquisa do trabalho, buscando incluir trabalhos que abordassem a engenharia de requisitos e metodologias ágeis em sistemas embarcados, e excluindo aqueles que não estivessem diretamente relacionados.

Critérios de inclusão:

- Estudos relacionados com engenharia, ciência ou tecnologia;
- Estudos publicados entre 2010 e 2022;
- Estudos escritos em inglês ou português.

Critérios de exclusão:

- Estudos não escritos em inglês ou português;
- Literatura cinza;
- Trabalhos incompletos ou artigos curtos.

3.1.2 Etapa de Execução

Mesmo após a redução dos resultados através da aplicação dos critérios de inclusão e exclusão, é comum que a quantidade de artigos ainda seja elevada. Por isso, é necessário realizar uma nova etapa de filtragem para reduzir ainda mais os resultados e tornar a leitura completa dos artigos viável. O passo a passo seguido nessa etapa foi o apresentado na Figura 3.2.

Figura 3.2: Etapas de execução da Revisão Sistemática da Literatura



Fonte: (Autor, 2023)

Por fim, com a lista final dos artigos que passaram por todas as etapas, foi realizada a extração de informações.

3.1.3 Etapa de Análise

A fase de análise de dados é uma das etapas mais importantes de uma revisão sistemática. É nesta fase que os dados relevantes são identificados e coletados a partir dos artigos selecionados na etapa anterior. A extração de informações é um processo sistemático e envolve a análise cuidadosa de cada artigo em busca das informações relevantes para o estudo em questão.

A forma que foi feita a extração foi voltada para responder às perguntas de pesquisas (RQ1, RQ2 e RQ3), inicialmente planejadas. Foi feita uma leitura completa de cada um dos 67 (sessenta e sete) estudos, cujos resultados são discutidos no próximo capítulo.

3.2 Considerações Finais

Neste capítulo foi apresentado o método que foi utilizado com objetivo de extrair informações sobre as práticas atuais de especificação de requisitos em sistemas embarcados, envolvendo também informações de quando estes sistemas são desenvolvidos em metodologias ágeis.

A revisão sistemática é um processo rigoroso e bem estruturado, que envolve a seleção cuidadosa dos estudos a serem incluídos na análise, a avaliação crítica dos dados, a síntese dos resultados e a interpretação dos achados. Tudo tendo por base um protocolo definido, capaz de ser reaplicado, na etapa de planejamento, foram definidas as questões de pesquisa, as estratégias de busca, as fontes de buscas, os critérios de inclusão e exclusão e as etapas de execução e análise.

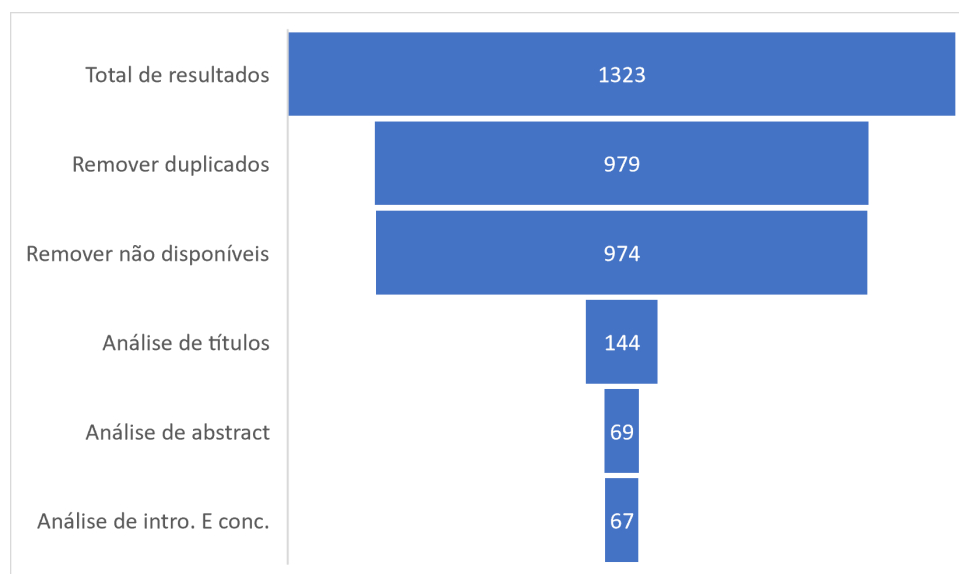
Capítulo 4

Análise dos Estudos

Neste capítulo, tem-se quatro subseções, onde na primeira são apresentados resultados coletados em cada uma das etapas da RSL, trazendo-se informações relevantes a serem consideradas. As outras três subseções tratam sobre cada uma das três perguntas de pesquisa delineadas no capítulo anterior.

4.1 Resultados das Etapas da Revisão Sistemática

A metodologia utilizada trouxe resultados diretos a serem visualizados. A Figura 4.1 mostra a quantidade de estudos em cada etapa da revisão sistemática. Na primeira etapa da metodologia, foram introduzidas as *strings* nas bases de dados, aplicando os critérios de seleção, e reunindo todos os resultados, totalizando 1323 estudos.

Figura 4.1: Total por etapa da revisão sistemática

Fonte: (Autor, 2023)

Ao aplicar um filtro para remover todos os estudos duplicados, foi possível perceber uma grande redução nos resultados, trazendo 979 estudos. A pesquisa passou também por uma etapa onde os estudos encontrados nas bases de dados deveriam ser agrupados para serem lidos. Ao tentar baixar todos os artigos, alguns deles se encontravam indisponíveis para *download*, e estes tiveram que ser removidos da revisão sistemática. No total, foram apenas 5 artigos que se encontraram indisponíveis, o que faz a revisão sistemática seguir com 974 estudos.

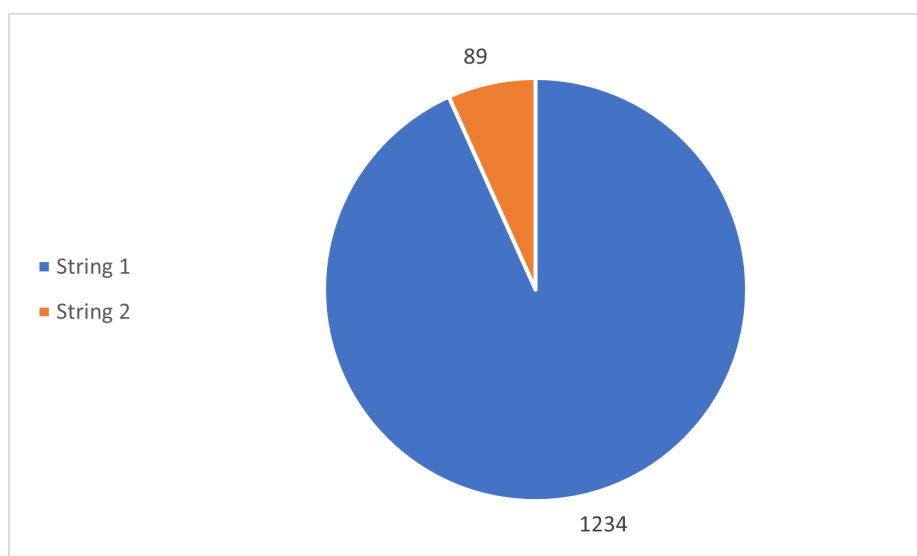
Durante a realização da revisão sistemática, foi realizada uma análise inicial dos títulos dos estudos identificados na busca. Dos 974 estudos com títulos lidos, apenas 144 passaram para a fase de análise de resumos. A leitura dos resumos trouxe apenas 69 resultados que trataram do tema especificado. Por último, foi feita uma análise de introdução e conclusão para eliminar estudos ainda não relacionados com o tema, e essa análise final eliminou apenas dois resultados, trazendo para a etapa de extração 67 estudos.

Foi possível ver que dos 1323 estudos, sobraram apenas 67, resultando num aproveitamento de aproximadamente 5%. Se formos considerar somente os resultados não duplicados, o aproveitamento dos estudos retornados pelas bases de dados sobe

para 6.8%.

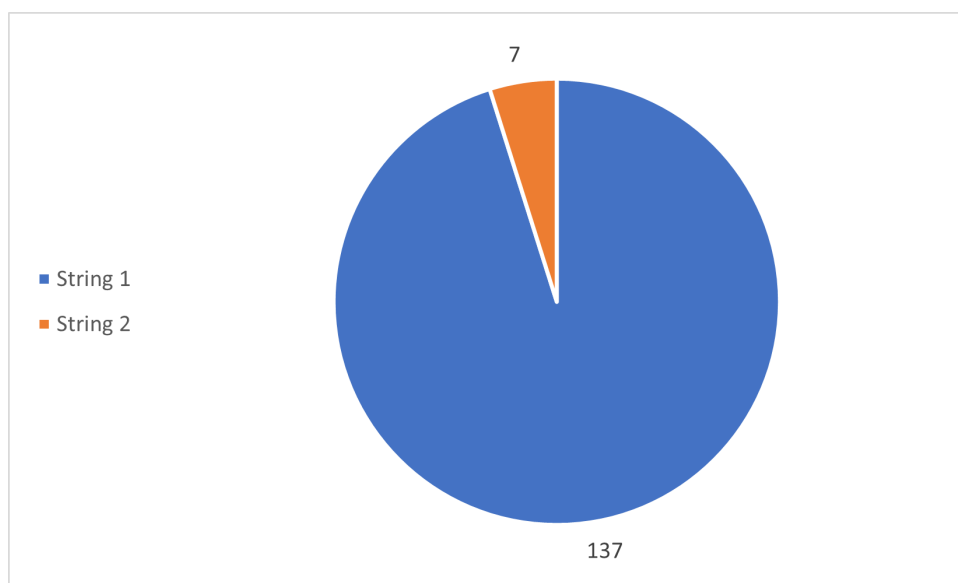
É importante ressaltar também outras informações sobre os resultados. Foi possível perceber que a grande maioria dos resultados foram referentes a *string* 1. A Figura 4.2 mostra que dos 1323 resultados, somente 89 vieram da *string* 2, o que representa aproximadamente 6.7% dos resultados.

Figura 4.2: Total por *string* - Todos os resultados



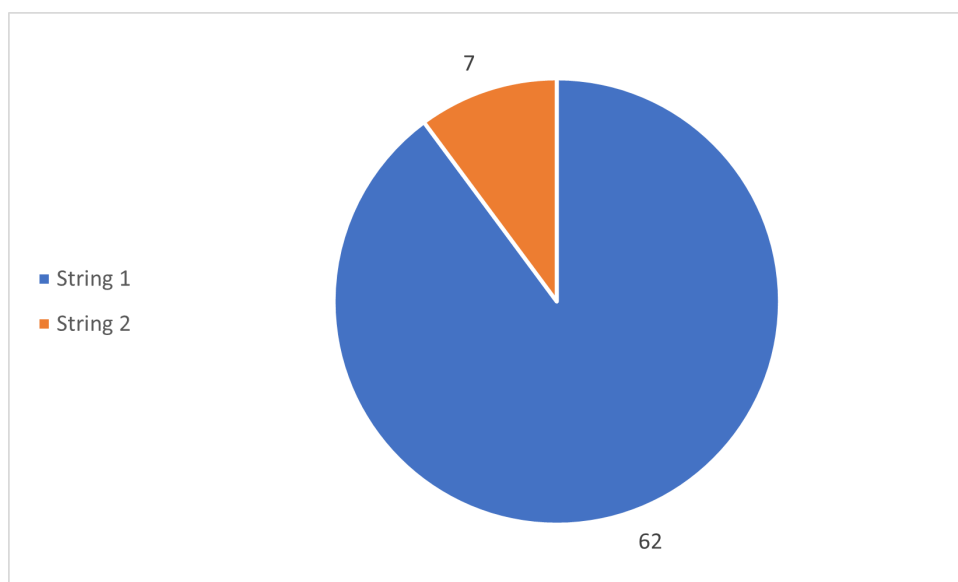
Fonte: (Autor, 2023)

Após remover os duplicados e os arquivos não encontrados para *download*, a leitura dos títulos já reduziram os estudos de forma considerável. A Figura 4.3 mostra os a quantidade de estudos por *string*.

Figura 4.3: Total por *string* - Após leitura de títulos

Fonte: (Autor, 2023)

Considerando os estudos após a leitura dos resumos, a Figura 4.4 mostra os resultados por *string*.

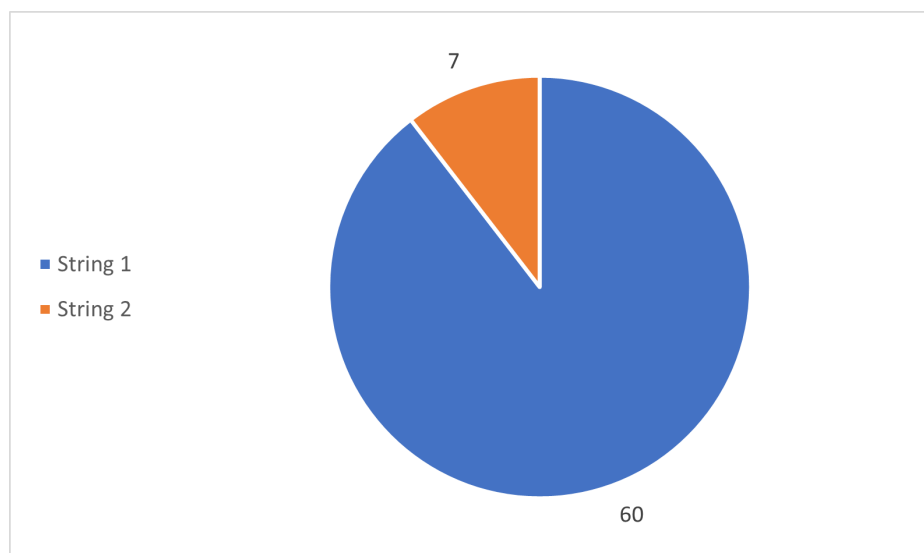
Figura 4.4: Total por *string* - Após leitura de resumos

Fonte: (Autor, 2023)

Por fim, houve a etapa da leitura de introdução e conclusão. Trazendo somente os resultados finais após todas as etapas da revisão sistemática, dos 67 estudos, 60

representaram a *string* 1, e 7 estudos representaram a *string* 2, mostrando que os resultados iniciais foram bem próximos dos resultados finais quando são separados por *string*. A Figura 4.5 ilustra o valor total final de estudos separados por *string*.

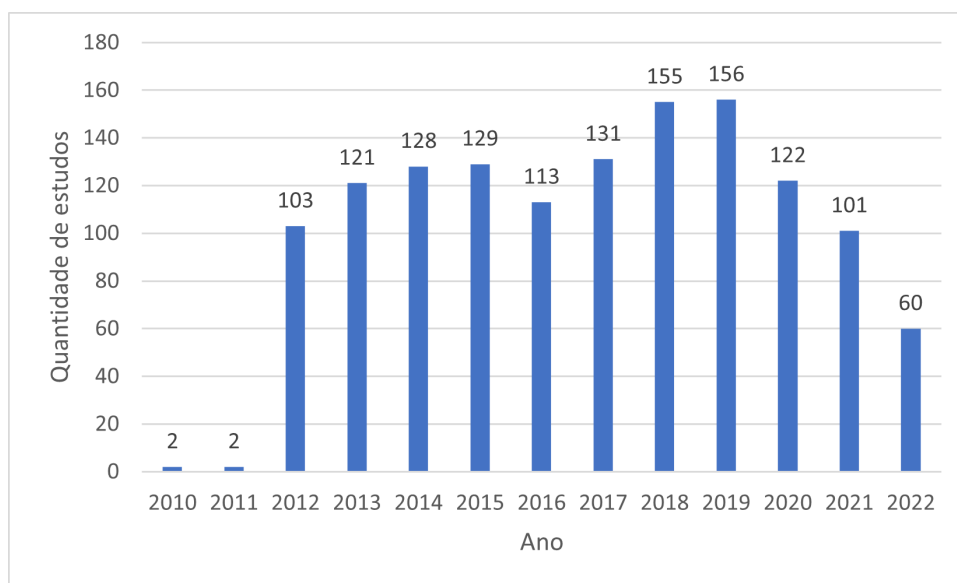
Figura 4.5: Total por *string* - Resultado final



Fonte: (Autor, 2023)

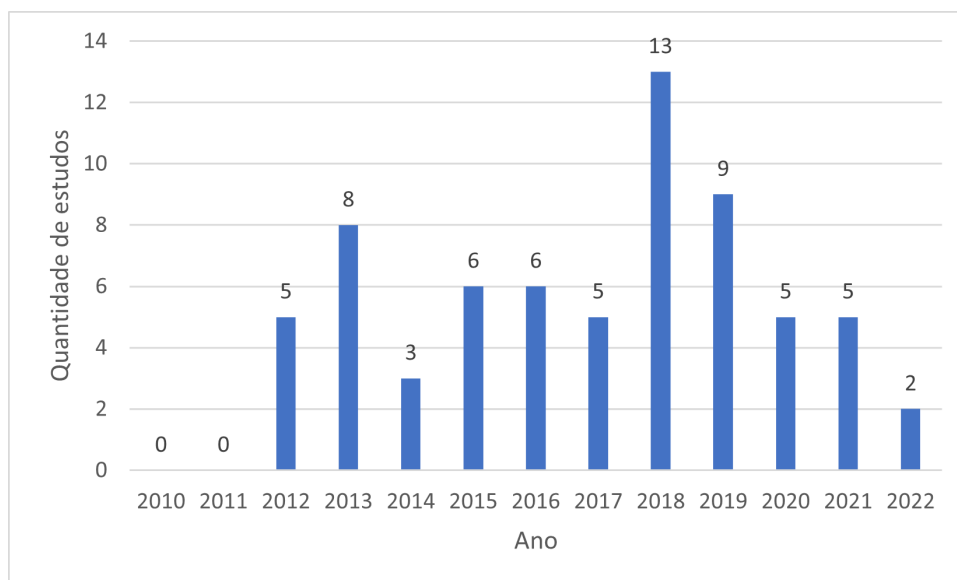
O ano de publicação dos estudos incluídos em uma revisão sistemática é importante para avaliar a relevância e atualidade das evidências apresentadas. Estudos mais antigos podem ter limitações, enquanto estudos mais recentes podem ser mais precisos, mas ainda não terem sido amplamente avaliados. Portanto, a avaliação do ano de publicação pode fornecer informações importantes para a interpretação dos resultados e para definição de recomendações baseadas em evidências atualizadas.

A Figura 4.6 mostra a quantidade de resultados no início da pesquisa, categorizando por ano de publicação do estudo. É possível ver que não houve grande concentração em anos recentes ou antigos, o que mostra que o interesse pelo conteúdo está se passando de forma constante ao longo dos anos. Vale lembrar também que os anos de 2010 e 2011 só tiveram 2 estudos cada, o que pode indicar que o interesse pelo tema não vem de antes. Mas vale lembrar também que um dos critérios de inclusão envolvia estudos de 2010 até 2022.

Figura 4.6: Total por ano - Todos os resultados

Fonte: (Autor, 2023)

A Figura 4.7 mostra a mesma comparação trazendo para os resultados finais. O que se percebe é que houve um pico no ano de 2018, mas para os demais anos, os resultados se mantiveram de forma constante.

Figura 4.7: Total por ano - Resultado final

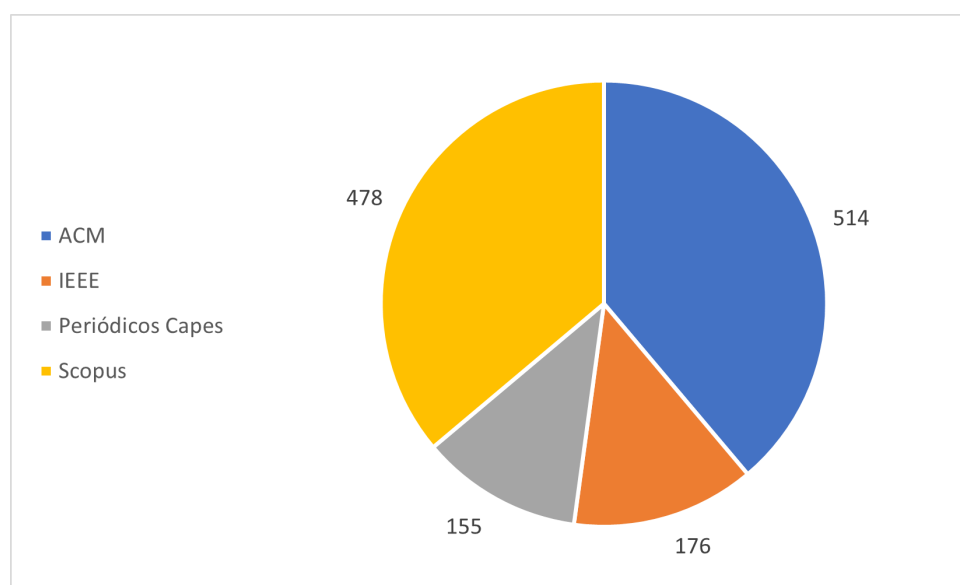
Fonte: (Autor, 2023)

A categorização dos estudos por base de dados é importante para incluir todas

as evidências relevantes na análise da revisão sistemática. Diferentes bases de dados podem ter estudos únicos e heterogeneidade nos resultados. Essa abordagem ajuda na definição de recomendações baseadas em evidências para tomadas de decisão e confiança nos resultados.

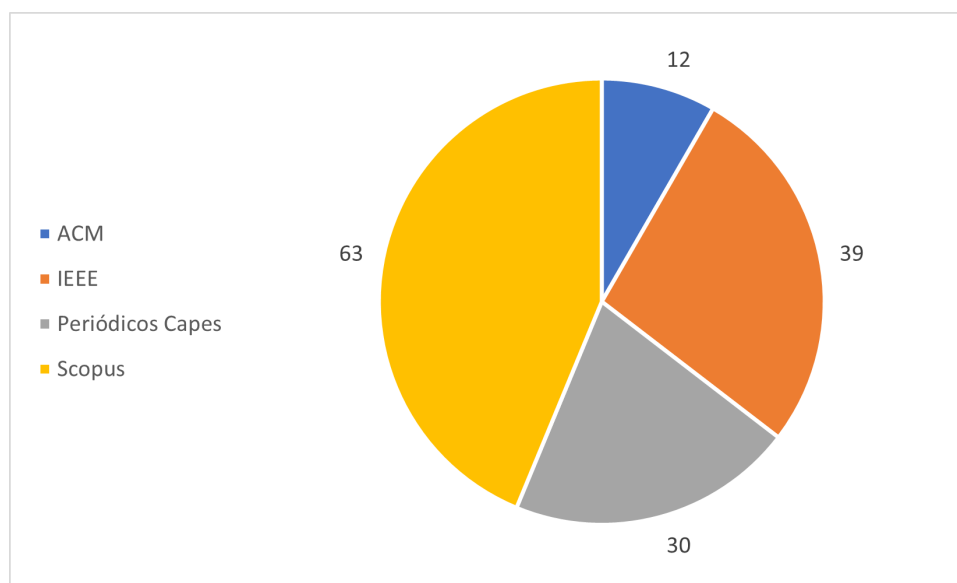
A Figura 4.8 mostra a quantidade de estudos por base, se tratando de todos os estudos encontrados. Majoritariamente os estudos se encontraram nas bases ACM e Scopus. As bases ACM e Scopus trouxeram 514 e 478 resultados, respectivamente, e representaram cerca de 75% dos resultados. A base IEEE trouxe 176 estudos, e a base de Periódicos da Capes trouxe 155.

Figura 4.8: Total por base de dados - Todos os resultados



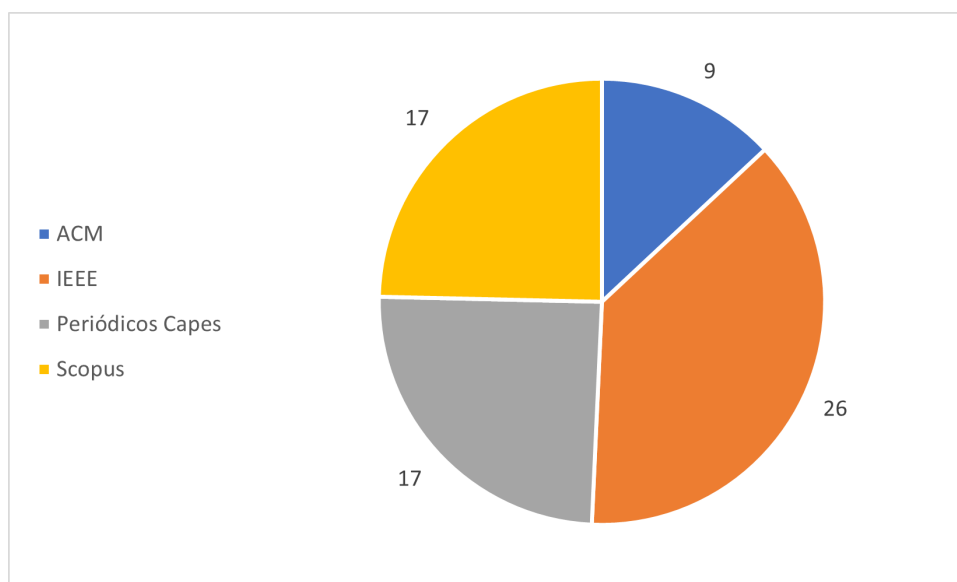
Fonte: (Autor, 2023)

Após a remoção dos duplicados e dos arquivos não encontrados para *download*, e também após a leitura dos títulos, os dados resultaram na Figura 4.9, que mostra as quantidades de artigos para cada uma das bases de dados. É possível ver que a representatividade da base Scopus aumenta muito nessa fase, enquanto a base ACM cai muito nos resultados.

Figura 4.9: Total por base de dados - Após leitura de títulos

Fonte: (Autor, 2023)

Considerando os estudos após a leitura dos resumos, a Figura 4.10 mostra os resultados por bases de dados.

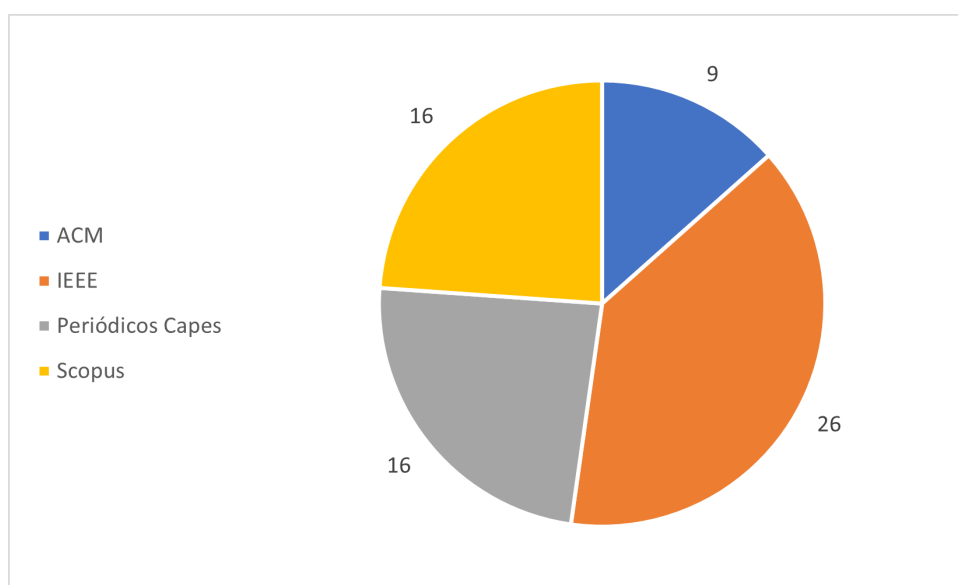
Figura 4.10: Total por base de dados - Após leitura de resumos

Fonte: (Autor, 2023)

Por fim, a leitura das introduções e conclusões trouxeram os resultados finais. Observando a Figura 4.11, é possível ver que, apesar de ser minoritário nos estudos

totais, a base IEEE liderou em quantidade de resultados finais. Trazendo 26 resultados finais, dos 176 iniciais, apresentou uma taxa de aproveitamento de 14,7%, quase triplicando a taxa geral. A base Scopus trouxe 16 resultados finais, o que mostra boa representatividade. Scopus trouxe também 16 resultados finais, e a base ACM, que apresentou a maior quantidade de resultados iniciais, foi a que menos trouxe resultados após todas as etapas da revisão sistemática da literatura.

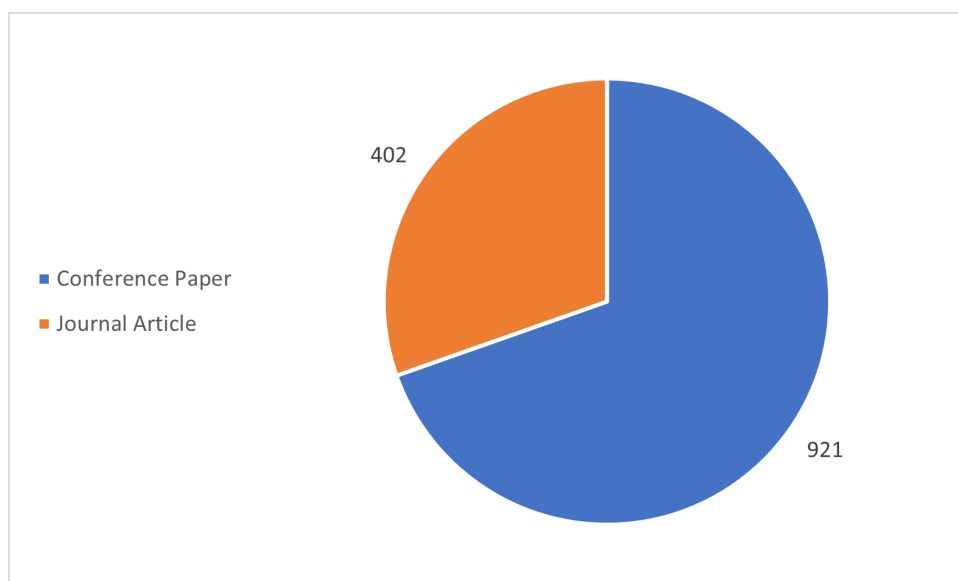
Figura 4.11: Total por base de dados - Resultado final



Fonte: (Autor, 2023)

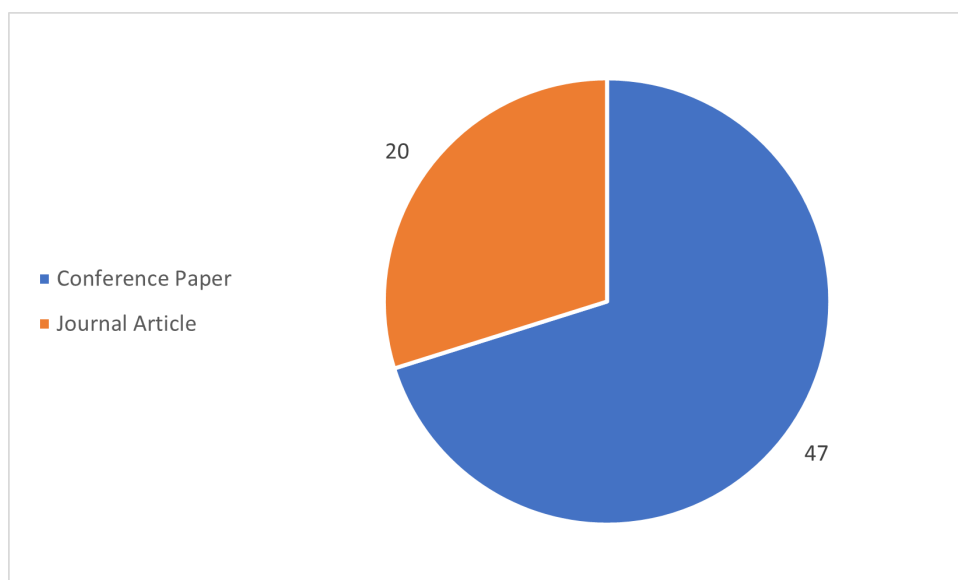
Distinguir entre “trabalhos apresentados em conferências” e “artigos de periódicos” é importante na revisão sistemática, pois os primeiros podem ser trabalhos preliminares com metodologia menos rigorosa, enquanto os últimos são publicações mais completas e revisadas por pares.

A Figura 4.12 mostra a quantidade de trabalhos apresentados em conferências e artigos de periódicos em suas proporções. É possível ver que cerca de 70% dos resultados iniciais são trabalhos apresentados em conferências, e aproximadamente 30% são artigos de periódicos.

Figura 4.12: Total por tipo de estudo - Todos os Resultados

Fonte: (Autor, 2023)

Trazendo os mesmos resultados para os estudos finais, a Figura 4.13 mostra que a proporção se mantém. Cerca de 70% dos estudos finais foram de trabalhos apresentados em conferências, e 30% foram de artigos de periódicos.

Figura 4.13: Total por tipo de estudo - Resultado final

Fonte: (Autor, 2023)

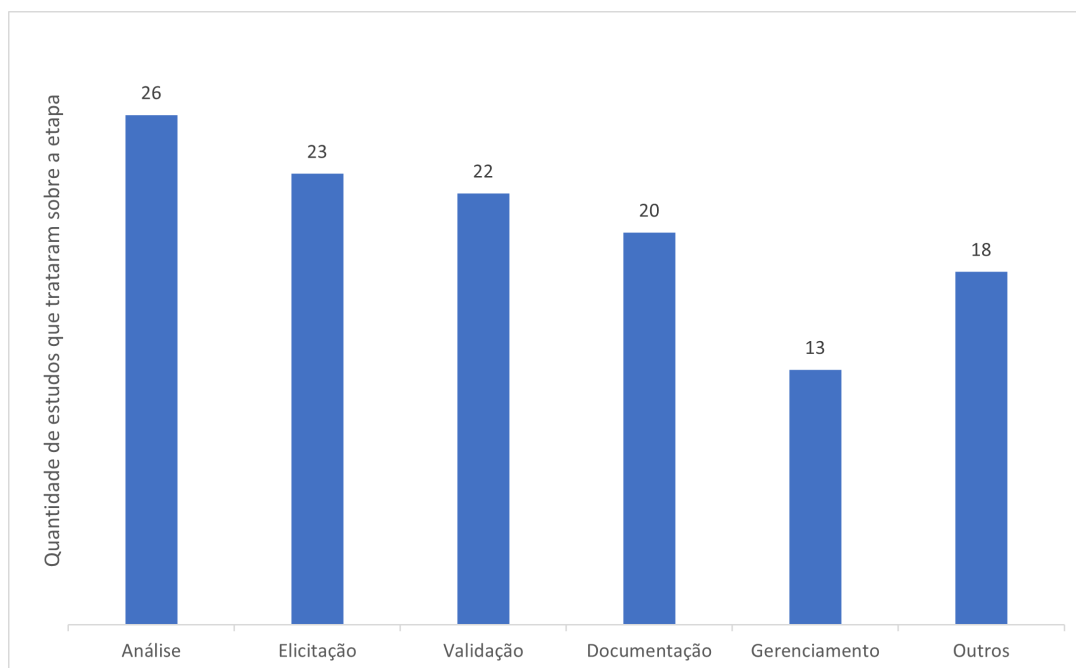
Apesar da maior concentração de estudos ser de trabalhos apresentados em con-

ferências, também foram obtidos diversos resultados de artigos em periódicos. É importante considerar que muitos trabalhos de alta qualidade são apresentados em conferências e podem eventualmente ser publicados em revistas científicas, o que não significa que uma maior concentração de estudos, sendo trabalhos apresentados em conferências, demonstram menor qualidade.

4.2 RQ1 - Quais etapas do processo de engenharia de requisitos estão presentes na especificação de sistemas embarcados?

Com a fase da extração, foi possível ter resultados para responder às perguntas RQ1 a RQ3. A Figura 4.14 mostra as etapas da engenharia de requisitos que foram encontradas nos artigos que se referem a projetos de sistemas embarcados, onde um mesmo artigo pode ter tratado de mais de uma etapa.

Figura 4.14: RQ1 - Etapas do processo de Engenharia de Requisitos encontrados em sistemas embarcados



Fonte: (Autor, 2023)

É possível ver que a etapa mais presente nos estudos se refere a Análise de Requisitos: 26 estudos trataram sobre essa etapa em projetos de sistemas embarcados. Porém, não houve uma concentração em poucas etapas, a distribuição seguiu um certo limite de uniformidade. Elicitação de Requisitos trouxe 23, Validação de Requisitos trouxe 22, Documentação de Requisitos trouxe 20 e Gerenciamento de Requisitos, 13.

É importante destacar que dos 67 estudos lidos, 26 não trouxeram abordagens voltadas para etapas da Engenharia de Requisitos em Sistemas embarcados. Dos estudos restantes, aproximadamente metade tratava de ao menos 3 etapas simultaneamente, aumentando a quantidade de resultados.

Se tratando de análise de requisitos, alguns estudos tiveram destaques. O estudo “*Model-Driven Requirements Engineering for Embedded Systems Development*” trata sobre a importância da especificação completa e compreensível de requisitos para o desenvolvimento de sistemas embarcados, que apresentam complexidade e desafios em relação aos recursos limitados e à necessidade de lidar com *tradeoffs* entre diferentes propriedades e objetivos de qualidade. O estudo apresenta uma extensão para a Linguagem de Definição e Análise de Requisitos (RDAL) para especificação de requisitos não-funcionais (*Non-Functional Requirements* - NFRs).

Pode-se citar também o estudo “*The Manager Perspective on Requirements Impact on Automotive Systems Development Speed*”, que aborda como a velocidade de desenvolvimento para sistemas automotivos intensos pode ser impactada pelo gerenciamento de requisitos.

O artigo “*Requirements engineering of industrial automation systems: Adapting the CESAR requirements meta model for safety-critical smart grid software*” trata sobre a importância da engenharia de requisitos para o desenvolvimento de sistemas industriais críticos de segurança, como as *smart-grids*. O objetivo é formalizar os requisitos de forma a automatizar e ajudar nas fases posteriores do ciclo de vida do desenvolvimento do sistema. O estudo enfatiza a importância da rastreabilidade entre os requisitos e os componentes do modelo do sistema IEC 61499 para gerar

casos de teste automáticos e análise formal dos requisitos.

Trazendo também o estudo “*A novel requirement analysis approach for periodic control systems*” trata de uma abordagem de análise de requisitos para sistemas de controle periódicos (PCSs) em sistemas embarcados em tempo real. A abordagem proposta visa automatizar a validação das especificações informais de requisitos por meio da construção de um modelo intermediário SPARDL com sintaxe e semântica formal a partir dos documentos de requisitos normalizados.

Um último exemplo para análise de requisitos é o estudo “*Testing Requirement Analysis Method Based on Signal Path for High-reliability Embedded software*” trata de uma proposta de método de análise de requisitos de teste para *software* embarcado de alta confiabilidade. O método proposto é baseado em análise de caminho de sinal e é destinado a lidar com limitações na análise de requisitos de *software* que são frequentemente descritos e particionados a partir do módulo de design.

Se tratando de elicitação, também houveram alguns resultados. Iniciando pelo estudo “*Enabling model-based requirements engineering in a complex industrial System of Systems environment*”, trata de um novo método para elicitação, especificação e gerenciamento de requisitos em ambientes complexos de sistemas embarcados, como os sistemas de sistemas (SoSs) da Indústria 4.0 e dos Sistemas Ciber-Físicos (CPS). O objetivo é fornecer uma abordagem padronizada e formalizada para modelagem de sistemas de acordo com especificações arquiteturais detalhadas, permitindo a verificação de rastreabilidade e interoperabilidade entre sistemas. O estudo menciona a introdução de um processo alinhado aos padrões arquiteturais, e é avaliado por meio de um estudo de caso real.

O estudo “*Petri Nets-based method to elicit component-interaction related safety requirements in safety-critical systems*” trata sobre uma abordagem baseada em Redes de Petri para a elicitação de requisitos de segurança relacionados à interação entre componentes em sistemas críticos de segurança.

Por último, o estudo “*A Metamodel to Guide a Requirements Elicitation Process for Embedded Systems*” busca em em uma revisão sistemática da literatura algo

que possa apoiar a elicitação de conceitos relacionados a sistemas embarcados e um processo de elicitação de requisitos que utilize esse metamodelo como guia. O resultado é um metamodelo com 118 entidades, que representam os conceitos de elicitação de requisitos para sistemas embarcados, e um processo com 49 ações para guiar a identificação e definição de requisitos. A conclusão é que o metamodelo pode ajudar os profissionais de desenvolvimento de sistemas embarcados a obterem as informações necessárias para o desenvolvimento adequado desses sistemas e a melhorar suas práticas de requisitos.

Trazendo também o tópico de validação, dois estudos podem ter destaque. O estudo “*The observer-based technique for requirements validation in embedded real-time systems*” trata sobre a validação de requisitos em sistemas embarcados por meio da técnica de observação baseada em modelo. A pesquisa propõe uma abordagem leve de validação que usa construtos de observação formalmente definidos para especificação de requisitos usando a linguagem *Timed Abstract State Machine* (TASM). A técnica é capaz de validar comportamentos funcionais e propriedades não funcionais (como tempo), de sistemas embarcados em tempo real. O estudo apresenta ainda a *Events Monitoring Logic* (EvML), para facilitar a especificação de observadores e exemplifica sua aplicação em um sistema industrial de Trava-Desatrava de Veículos.

O estudo “*Requirement Validation for Embedded Systems in Automotive Industry Through* ” trata sobre a importância da validação de requisitos para sistemas embarcados na indústria automotiva e propõe uma abordagem baseada em modelos, para validar esses requisitos. O objetivo é detectar erros nos requisitos o mais cedo possível no ciclo de vida do projeto, evitando problemas no desenvolvimento posterior e reduzindo o tempo e o custo de desenvolvimento. O estudo destaca que a validação de requisitos é fundamental para garantir a qualidade do produto final e que a falta de atenção a essa fase, pode levar a projetos fracassados e produtos de baixa qualidade. A abordagem proposta utiliza técnicas formais, como redes de Petri, e é aplicada em dois estudos de caso, na indústria automotiva. O estudo

demonstra que, a abordagem baseada em modelos, pode ser eficaz na validação de requisitos para sistemas embarcados na indústria automotiva.

O estudo “*FAT-AES: Systematic Methodology of Functional Testing for Automotive Embedded software*” trata sobre o desenvolvimento de uma metodologia sistemática de testes funcionais para *software* embarcado automotivo, com o objetivo de estruturar, otimizar e planejar os testes com critérios reais automotivos, atendendo aos requisitos de especificação.

Sobre documentação, também houveram diversos resultados. O estudo “*A Restricted Natural Language Based Use Case Modeling Methodology for Real-Time Systems*” trata sobre a importância da modelagem e validação de propriedades relacionadas ao tempo, durante as fases de documentação e análise de requisitos, para o desenvolvimento bem-sucedido de sistemas em tempo real em termos de custo, qualidade e produtividade.

O artigo “*Identification of a reusable requirements structure for embedded products in a dynamic market environment*” trata da identificação de uma estrutura de requisitos reutilizável para produtos embarcados em um ambiente de mercado dinâmico. Ele aborda a importância da reutilização de requisitos para aumentar a eficiência e qualidade dos processos de requisitos, e como é necessário uma estrutura bem definida para organizar os requisitos antes de sua documentação e reutilização. O estudo propõe uma nova forma de agrupar os requisitos, e utiliza técnicas existentes de modelagem de requisitos, para criar uma estrutura de especificação de requisitos reutilizável. O estudo foi aplicado a um projeto de desenvolvimento de inversores solares, e a estrutura proposta foi considerada bem aceita pelos usuários e suportou a reutilização de requisitos.

Gerenciamento de requisitos trouxe um destaque em específico: “*A model-based workflow from specification until validation of timing requirements in embedded software systems*”. O estudo trata de uma abordagem para integrar o gerenciamento de requisitos (RM) com o desenvolvimento baseado em modelos (MDD) e ferramentas de validação de tempo, em sistemas embarcados. O objetivo é preencher a lacuna

existente entre essas áreas e propor um fluxo de trabalho integrado, que permita gerenciar requisitos de tempo em ferramentas de RM, especificá-los em ferramentas de MDD e validá-los em ferramentas de análise de tempo.

Mais um estudo tratando de gerenciamento de requisitos foi o “*A model-based framework encompassing a complete workflow from specification until validation of timing requirements in embedded software systems*” aborda a importância da gestão de requisitos, em particular requisitos de temporização, em sistemas embarcados complexos e críticos. Ele propõe uma abordagem integrada baseada em modelos, que abrange todo o fluxo de trabalho, desde a especificação até a validação de requisitos de temporização, utilizando ferramentas de gestão de requisitos, desenvolvimento baseado em modelos e análise de temporização especializada. O objetivo é melhorar a qualidade do *software* e preencher a lacuna na falta de uma estrutura de trabalho completa para o desenvolvimento de *software* embarcado de alta qualidade.

As respostas a RQ1 trouxeram como reflexões:

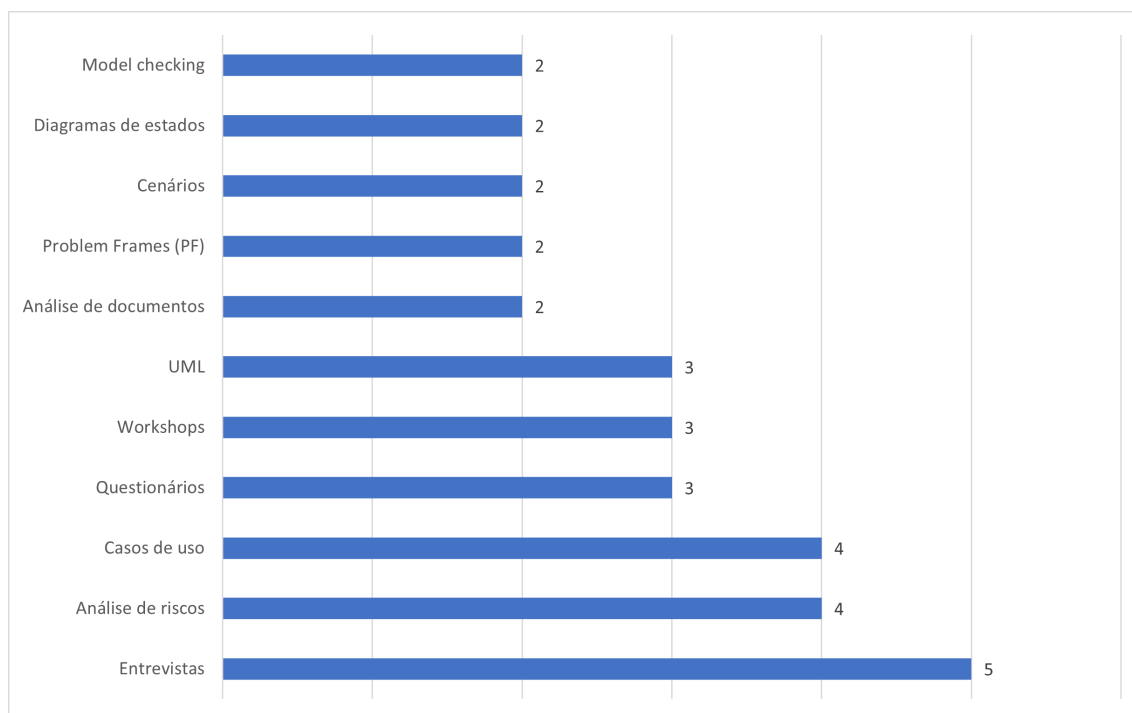
- A utilização de ferramentas para especificação de requisitos não funcionais;
- A forma como os requisitos são gerenciados impacta na velocidade de desenvolvimento do sistema;
- A formalização dos requisitos é muito importante para tentar automatizar as validações, por isso a rastreabilidade é essencial;
- É útil padronizar e formalizar um método para elicitação;
- Deve ser considerado no levantamento de requisitos, a interação entre os componentes do sistema;
- Vale considerar a utilização de um metamodelo como guia para elicitação;
- A validação dos requisitos devem envolver aspectos funcionais e não funcionais;
- A detecção de erros deve ser feita o mais rápido possível, isso pode ser feito considerando os requisitos de forma constante no desenvolvimento;

- A estruturação e o planejamento dos testes devem considerar critérios bem definidos;
- Vale considerar uma estrutura reutilizável de requisitos para documentação;
- É de alta importância considerar requisitos de tempo;
- Pode ser considerado o desenvolvimento baseado em modelos.

4.3 RQ2 - Quais métodos, técnicas, processos e ferramentas da engenharia de requisitos são utilizados na especificação de sistemas embarcados?

Para responder a RQ2, os resultados foram separados em três análises diferentes: técnicas, métodos e processos, e ferramentas. A Figura 4.15 mostra graficamente as técnicas encontradas na engenharia de requisitos, para especificar sistemas embarcados. É importante destacar que houveram artigos individuais que trataram de temas relevantes para a questão, mas que não se repetiram em nenhum outro estudo, portanto, estes não foram mostrados nas representações gráficas. Dentre eles: Componentização de requisitos, *Requeriments Definition and analysis language* (RDAL), Técnicas de priorização, Estudos de caso, Técnica baseada em observadores, Redes de Petri, Rede Bayesiana e Validação cruzada.

Figura 4.15: RQ2 - Técnicas da engenharia de requisitos na especificação de sistemas embarcados



Fonte: (Autor, 2023)

Em relação ao tópico Casos de uso, o estudo “*software Requirement Specification Based on a Gray Box for Embedded Systems: A Case Study of a Mobile Phone Camera Sensor Controller*” trata de uma proposta de método de especificação de requisitos, baseado em uma abordagem de caixa cinza (gray box), que busca um equilíbrio entre os modelos de caixa preta (black box) e caixa branca (white box), para sistemas embarcados. O estudo menciona que o modelo de caso de uso é uma das abordagens mais utilizadas para especificar requisitos funcionais, mas que a abordagem de caixa preta, pode omitir informações importantes para o desenvolvimento de sistemas embarcados, que não divulgam a maioria das lógicas de implementação fora do sistema. O estudo propõe uma abordagem que busca um nível adequado de modelagem do sistema embarcado para definir os requisitos, e também apresenta uma proposta de mecanismo para geração automática de *interface* de programação de aplicativos, para cada componente, com base no modelo criado.

Sobre análise de riscos, o artigo “*What Requirements Knowledge Do Developers*

Need to Manage Change in Safety-Critical Systems?” aborda a análise de riscos relacionados a mudanças propostas em sistemas críticos de segurança, que podem estar relacionados a sistemas embarcados. Os desenvolvedores precisam avaliar o impacto que essas mudanças teriam nos controles de segurança existentes e, para isso, é necessário o uso de informações sobre riscos, pressupostos ambientais e requisitos de segurança.

Os estudos sobre questionários, entrevistas e *workshops* não abordaram nada de diferente do que já é debatido na engenharia de *software*. Os questionários foram trazidos como uma técnica de pesquisa quantitativa para coletar informações de grandes números de pessoas, as entrevistas foram abordadas de forma qualitativa, permitindo coletar informações mais profundas e detalhadas, e os *workshops* foram abordados trazendo as partes interessadas para debates, em busca da especificação dos requisitos.

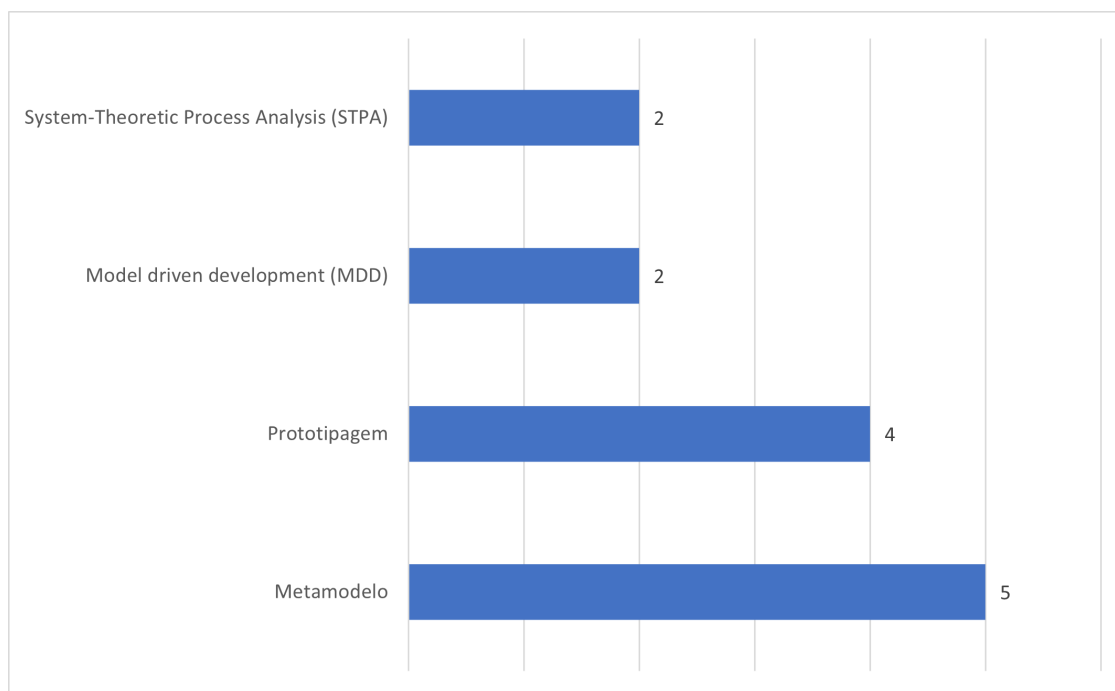
Em relação a técnicas, é possível considerar:

- A utilização de entrevistas, questionários, e *workshops* para levantamento dos requisitos em sistemas embarcados;
- Os casos de uso podem ser utilizados assim como no desenvolvimento de *software* para especificação de requisitos funcionais;
- É importante considerar o impacto de mudanças nos requisitos ao sistema, principalmente considerando fatores de segurança;
- UML pode ser considerado de forma adaptada para alguns sistemas embarcados.

Se tratando de métodos, a Figura 4.16 mostra os resultados obtidos pela revisão sistemática. Da mesma forma que as técnicas, os métodos também tiveram resultados individuais não mostrados no gráfico: Abordagem baseada em modelos (MDA), *Model-based system engineering* (MBSE), *Model-Driven engineering* (MDE), Ontologias, *SOFL structured object oriented formal language*, Abordagem STAMP

(*Systems theoretic accident modeling and process*), *Aspect-Oriented software Development* (AOSD), Modelagem orientada a aspectos (AOM), Testes automatizados, Testes de segurança.

Figura 4.16: RQ2 - Métodos e processos da engenharia de requisitos na especificação de sistemas embarcados



Fonte: (Autor, 2023)

Um estudo que trata sobre metamodelos em questão de análise foi o “*Towards a Metamodel for a Requirements Engineering Process of Embedded Systems*”. Trata do desenvolvimento de um metamodelo, para orientar o processo de engenharia de requisitos de sistemas embarcados, com o objetivo de minimizar problemas relacionados à compreensão inadequada dos requisitos. Os autores propõem um metamodelo que define conceitos e relacionamentos, que devem ser considerados no desenvolvimento de um sistema embarcado, bem como um processo preciso de engenharia de requisitos.

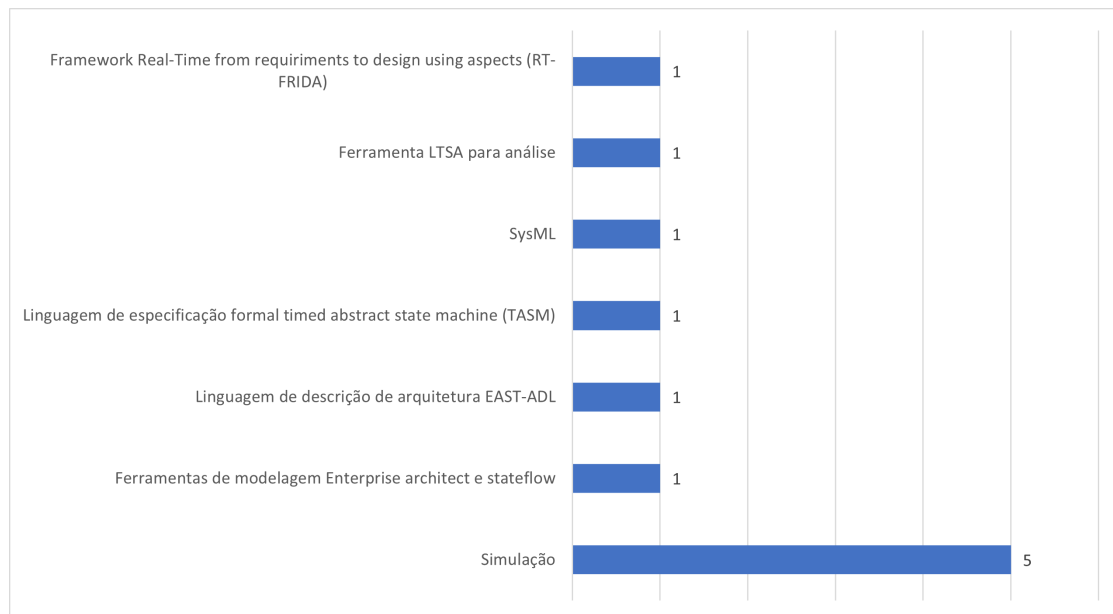
Complementando a análise, considerando metamodelos e já destacando um ponto em UML, o estudo “*A Restricted Natural Language Based Use Case Modeling Methodology for Real-Time Systems*” trata de uma metodologia de modelagem de casos de

uso, baseada em linguagem natural restrita (RUCM4RT), para especificar requisitos funcionais de sistemas em tempo real, juntamente com restrições relacionadas ao tempo. Ele propõe o uso de um perfil UML para Modelagem e Análise de Sistemas Embarcados e em Tempo Real (MARTE) e um mecanismo de formalização baseado em metamodelos, chamado UCMeta4RT, para formalizar automaticamente modelos de casos de uso. O estudo argumenta que a modelagem e validação de propriedades relacionadas ao tempo, nas fases de especificação e análise de requisitos, é importante para o desenvolvimento bem-sucedido de sistemas em tempo real, em termos de custo, qualidade e produtividade. O estudo também apresenta dois estudos de caso do mundo real, para avaliar a solução proposta.

Sobre os métodos, é possível concluir que metamodelos podem ser utilizados na fase de requisitos, em sistemas embarcados. Além disso, a prototipagem já é válida e utilizada, e este estudo só reforça a utilização.

A Figura 4.17 mostra as ferramentas encontradas para especificação de requisitos em sistemas embarcados.

Figura 4.17: RQ2 - Ferramentas da engenharia de requisitos na especificação de sistemas embarcados por ênfase



Fonte: (Autor, 2023)

No tópico de simulação, o artigo “*Towards a Workflow for Model-Based Testing*

of Embedded Systems” apresenta uma *workflow* de teste baseada em modelos para sistemas embarcados, que começa com a especificação de requisitos por meio de uma linguagem específica de domínio (DSL), similar ao Gherkin. Essa linguagem ajuda a extrair informações sobre os elementos do modelo e, posteriormente, a criar um modelo completo do sistema sob teste (SUT), usando uma plataforma de simulação específica de domínio. O estudo avalia essa *workflow* usando um subsistema do sistema de controle de trens, desenvolvido pela Alstom Transport AB na Suécia, e mostra que o *script* de teste gerado a partir do modelo SUT é executado com sucesso em uma plataforma de execução de *software* em loop (SIL) e os resultados dos testes são visualizados como uma sequência de etapas de teste passadas e falhadas.

Ainda tratando de simulação, o estudo “*A novel requirement analysis approach for periodic control systems*” propõe uma abordagem de análise de requisitos para sistemas de controle periódicos (PCSs), em sistemas embarcados em tempo real. A abordagem proposta inclui a validação automatizada das especificações informais de requisitos, por meio da construção de um modelo intermediário SPARDL, com sintaxe e semântica formais a partir de documentos de requisitos normalizados. Esse modelo pode ser transformado em código executável para simulação e validação do comportamento do sistema, por meio de testes.

É possível perceber sobre ferramentas:

- As principais ferramentas que envolvem requisitos em sistemas embarcados abrangem simulação;
- É possível construir modelos intermediários para validação de requisitos em sistemas embarcados;
- É possível utilizar ferramentas similares às utilizadas na engenharia de *software*, como o DSL, que é similar ao Gherkin.

4.4 RQ3: Sistemas embarcados podem ser desenvolvidos em metodologias ágeis? Se sim, como? Como são especificados?

Partindo agora para a RQ3, foi possível obter alguns resultados. É importante mencionar que todos os artigos referentes a *string* 1 e *string* 2 foram lidos, mas apenas os resultados da *string* 2 trouxeram abordagens relativas à metodologias ágeis. E mesmo considerando os sete artigos trazidos pela *string* 2, nem todos abordaram a pergunta em questão, mas quatro resultados tiveram destaque. O primeiro é um questionário sobre utilização de metodologias ágeis para desenvolvimento de sistemas críticos de segurança, que apontaram que problemas de segurança nem sempre são identificados no início, e os requisitos devem ser avaliados e ao longo do desenvolvimento. O segundo deles é uma revisão sistemática sobre metodologias ágeis para *software* embarcado, onde é abordada a importância do XP e do scrum. O terceiro traz abordagens relativas a testes automatizados em sistemas críticos de segurança, incluindo sistemas embarcados. O último deles aborda uma documentação padrão para sistemas embarcados utilizando a metodologia scrum.

O estudo “*Challenges and Opportunities in Agile Development in Safety Critical Systems: A Survey*” trata das dificuldades e oportunidades da aplicação de metodologias ágeis no desenvolvimento de sistemas críticos de segurança, incluindo sistemas embarcados. O objetivo do estudo foi explorar a relação entre três atividades-chave em engenharia de segurança e uma abordagem ágil - desenvolvimento de requisitos de segurança, análise de perigos e desenvolvimento de casos de segurança. O estudo apresenta os resultados de uma pesquisa realizada com profissionais da área e discute brevemente as implicações para a integração dessas atividades na abordagem ágil. Segundo o estudo, Sistemas críticos de segurança, que podem envolver sistemas embarcados, podem ser desenvolvidos integralmente ou parcialmente usando metodologias ágeis. Além disso, houve reconhecimento de que os problemas de segurança nem sempre são identificados no início do ciclo de vida. Houve forte apoio para a

avaliação regular do progresso do caso de segurança, durante o desenvolvimento.

O estudo “*Applying Agile Methods to Embedded software Development: A Systematic Review*” trata da aplicação de metodologias ágeis no desenvolvimento de *software* embarcado, com o objetivo de explorar a possibilidade de adotar essas práticas em um contexto que possui características distintas do desenvolvimento de *software* não embarcado. O estudo realiza uma revisão sistemática, para fornecer um guia inicial para pesquisadores e profissionais da indústria interessados nessa abordagem. São trazidas características do *software* embarcado que trazem desafios para a aplicação do ágil:

- As habilidades dos membros da equipe devem ser altos, e é difícil ter pessoas muito capacitadas;
- Dependências de *hardware*: O *software* fica dependente do *hardware*;
- Pressões de fora – Mercado, concorrência trazem falta de tempo e orçamento;
- Recursos limitados;
- Mudança de requisitos, mudança de *hardware*;
- Requisitos de desempenho.

O artigo traz XP como a mais utilizada, seguida do scrum. O desafio é que em desenvolvimento embarcado, o *hardware* geralmente é desenvolvido simultaneamente com o *software* relacionado e, portanto, não disponível até o final do projeto. O XP tem a capacidade de fazer um progresso significativo antes da disponibilidade do *hardware*. O Scrum pode ser combinado com XP.

Outro estudo obtido foi o “*Patterns for Development of Safety-Critical Systems with Agile: Trace Safety Requirements and Perform Automated Testing*”, que trata de como aplicar metodologias ágeis no desenvolvimento de sistemas críticos de segurança, incluindo sistemas embarcados, com foco em garantir a rastreabilidade dos requisitos de segurança e realizar testes automatizados. Para o estudo, Sistemas críticos de segurança (incluindo sistemas embarcados), podem ser desenvolvidos em

metodologias ágeis. O estudo trata sobre a validação de requisitos em sistemas que são desenvolvidos usando metodologias ágeis, utilizando testes automatizados. Os testes funcionavam de forma que a primeira vez que ele é implementado, deve ser testado manualmente, mas as demais vezes pode ser testado inteiramente com o sistema, de forma automatizada. Esses testes ocorrem de forma incremental, assim como o desenvolvimento. Eles aplicaram os testes automatizados, baseados em *software* referidos como Framework SBST, para automação de testes, para apoiar o projeto de um controlador lógico programável industrial.

O estudo “*A Generic Model for software Documentation and Its Application in Embedded Systems Developed with Scrum*” apresenta uma abordagem para documentação de *software* em sistemas embarcados, desenvolvidos com a metodologia ágil Scrum. O objetivo do trabalho é apresentar um modelo genérico de documentação de *software* e seu estudo de caso, aplicando a metodologia Scrum. O modelo proposto fornece diretrizes para os engenheiros escolherem e estruturarem a documentação de *software* de forma adequada, considerando as particularidades do desenvolvimento de sistemas embarcados. O trabalho traz um estudo de caso, aplicando scrum para sistemas embarcados em um servidor WEB embarcado. O scrum é levado da fase da especificação de requisitos até a análise, design e controle do sistema.

Com estes estudos, pode-se reunir algumas informações sobre metodologias ágeis em sistemas embarcados:

- É interessante incluir no processo ágil validações constantes do sistema, para encontrar defeitos o mais rápido possível, já que nem sempre estes surgem na fase da especificação;
- Existem alguns problemas relacionados a desenvolver sistemas embarcados com metodologias ágeis, mas a metodologia XP, com algumas adaptações pode ser considerada útil. O Scrum muito bem adaptado ao projeto também pode ser útil;

- É possível incluir testes automatizados em sistemas embarcados desenvolvidos com metodologias ágeis;
- A depender do sistema a ser desenvolvido, pode-se considerar uma documentação padronizada, e incluir a fase da especificação no processo ágil.

4.5 Considerações Finais

Neste capítulo foi apresentada a análise realizada a partir da coleta das evidências, através da Revisão Sistemática da Literatura.

Como ameaça à validade das informações, podem ser categorizadas quatro categorias: ameaças de validade de construção, internas, externas e de conclusão (WOHLIN et al., 2012).

Sobre validade de construção, foram seguidas as diretrizes tratadas por KITCHENHAM et al. (2009) para seguir um protocolo confiável. O protocolo seguiu o padrão encontrado em outros já publicados, e as etapas definidas foram revisitadas diversas vezes até que se tivesse certeza.

Tratando de Validade interna, algumas decisões referentes à inclusão e exclusão de estudos podem ter sido tomadas de forma subjetiva, já que alguns artigos não eram tão claros com suas abordagens. De qualquer forma, a pesquisa foi conduzida de forma a mitigar quaisquer problemas de viés no entendimento do conteúdo.

Abordando Validade externa: Se tratando da aplicação em metodologias ágeis, é difícil concluir que é possível generalizar os resultados pela pequena quantidade de estudos finais. Contudo, se tratando de especificação de requisitos em sistemas embarcados, a pesquisa foi feita utilizando uma metodologia rigorosa, além de haver uma boa quantidade de resultados com características em comum.

Se tratando da Validade de conclusão: a revisão sistemática da literatura é um método que já supõe que nem todos os estudos, que são relacionados ao tema, conseguirão ser identificados e levados até a fase final. Com isso, é possível que alguns estudos excluídos ou nem encontrados para esta revisão poderiam ter sido

incluídos. Para reduzir os impactos desse fator, foi feito estudo prévio sobre os termos de busca e foi tomado cuidado ao definir o processo de análise e os critérios de inclusão e exclusão.

Capítulo 5

Considerações finais

As considerações finais desse trabalho foram separadas em três tópicos, de forma que o primeiro trata do que foi possível concluir com esse trabalho, o segundo se trata das dificuldades encontradas na realização desse trabalho, e o terceiro trata sobre possíveis trabalhos futuros a serem realizados.

5.1 Conclusões

Através do estudo foi possível construir conhecimento de base, num bom nível, sobre as práticas de engenharia de requisitos em sistemas embarcados. Além disso, foi criado grande conhecimento sobre metodologias ágeis e sua utilização em sistemas embarcados. Com o conhecimento obtido através do estudo, foi possível realizar uma revisão sistemática de qualidade, utilizando critérios rigorosos de pesquisa.

Diante dos resultados obtidos na pesquisa, pode-se concluir que existem diversos métodos e técnicas da engenharia de *software* que podem ser aplicados para especificar requisitos em sistemas embarcados. Dentre eles, entrevistas, casos de uso, questionários e *workshops* se destacam por ter tido mais resultados em comum. Além disso, foi possível concluir que diferentes etapas da engenharia de requisitos são também abordadas em sistemas embarcados, envolvendo análise, elicitação, documentação e também validação. Essa conclusão é relevante para aumentar as abordagens sobre especificação de requisitos em sistemas embarcados e tornar mais

claro quais técnicas podem ser usadas e como os requisitos são especificados em si.

Além disso, houve outra busca por resultados: sistemas embarcados são desenvolvidos em metodologias ágeis? Como a quantidade de estudos obtidos foi baixa, não foi possível tomar conclusões precisas. O que se conclui é que, em alguns casos, é possível utilizar metodologias ágeis em sistemas embarcados, e além disso, existem possibilidades de se especificar requisitos nesses sistemas, mas não é possível generalizar essas abordagens. Essa conclusão abre margem para novos estudos explorando o que foi obtido sobre XP ou scrum, em sistemas embarcados.

O objetivo principal do trabalho foi de identificar formas de especificação de requisitos em sistemas embarcados. Com os resultados obtidos, pode-se concluir que sim, é possível, e existem técnicas frequentemente utilizadas para especificar requisitos em sistemas embarcados. Se tratando do segundo ponto, que envolve o seu desenvolvimento através de metodologias ágeis, o estudo não permitiu tomar conclusões concretas, visto que a quantidade de resultados foi muito baixa.

Com base nisso, conclui-se que o presente trabalho representa uma importante contribuição para a engenharia de requisitos e os estudos em sistemas embarcados. O estudo pode ser visitado por grupos de pesquisa, profissionais da área e até curiosos sobre a área.

5.2 Dificuldades Encontradas

As maiores dificuldades encontradas no desenvolvimento desse estudo foram voltadas à realização da extração dos dados. Toda a atenção e dedicação exigidas nessa fase da pesquisa, tornou muito mais cansativo e difícil para a finalização. A seleção de estudos em alguns momentos teve que ser feita com base em um pouco de subjetividade, o que também foi um ponto de dificuldade. Mas de forma geral, foi possível realizar o estudo com grande satisfação.

5.3 Trabalhos Futuros

Como trabalhos futuros, busca-se realizar trabalhos no mesmo tema, de forma alternativa. Visto que, se tratando de metodologias ágeis, não é possível tomar conclusões, é possível que com algumas alterações no método de pesquisa, outros estudos, que talvez abordem sobre o tema, possam ser encontrados.

Outras abordagens também podem ser trabalhadas, como a utilização dos métodos encontrados para especificar requisitos de sistemas embarcados. Pode-se aprofundar nas técnicas mais encontradas, visando buscar por mais detalhes e até outras técnicas também utilizadas.

Referências

AHMAD, K., ABDELRAZEK, M., ARORA, C., BANO, M., e GRUNDY, J. (2023). Requirements engineering for artificial intelligence systems: A systematic mapping study. <https://www.sciencedirect.com/science/article/pii/S0950584923000307>. Acesso em: 12/05/2023.

AL-FUQAHA, A., GUIZANI, M., MOHAMMADI, M., ALEDHARI, M., e AYYASH, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *Communications Surveys & Tutorials, IEEE*, 17(4):2347–2376.

ALCIDAROS, H., OMAR, M., e ROMLI, R. (2021). The state of the art of agile kanban method: challenges and opportunities. *Independent Journal of Management & Production*, 12(8):2535–2550.

ALSANOOSY, T., SPICHKOVA, M., e HARLAND, J. (2020). Exploratory analysis of cultural influences on requirements engineering activities based on stakeholders' profile. <https://www.sciencedirect.com/science/article/pii/S1877050920319542>. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 24th International Conference KES2020.

BEC, K. (2002). Extreme programming. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=-8d6824ea0826c2d90cb737002ce4c58d77f475d0> Acesso em: 12/05/2023.

BIOLCHINI, J., MIAN, P. G., NATALI, A. C. C., e TRAVASSOS, G. H. (2005). Systematic review in software engineering. Technical report, PESC – COPPE/UFRJ.

BURNAY, C., JURETA, I. J., e FAULKNER, S. (2014). What stakeholders will or will not say: A theoretical and empirical study of topic importance in requirements engineering elicitation interviews.

<https://www.sciencedirect.com/science/article/pii/S0306437914000908>. Acesso em: 12/05/2023.

CHEEMA, S. M., TARIQ, S., e PIRES, I. M. (2023). A natural language interface for automatic generation of data flow diagram using web extraction techniques. <https://www.sciencedirect.com/science/article/pii/S131915782300006X>. Acesso em: 12/05/2023.

CURCIO, K., NAVARRO, T., MALUCELLI, A., e REINEHR, S. (2018). Requirements engineering: A systematic mapping study in agile software development. <https://www.sciencedirect.com/science/article/pii/S0164121218300141>. Acesso em: 12/05/2023.

DIMES, T. (2014). *Scrum Essencial*. Babelcube Inc., 1. edição.

FADEL, A. C. e SILVEIRA, H. D. M. (2010). Metodologias ágeis no contexto de desenvolvimento de software: Xp, scrum e lean. *Monografia do Curso de Mestrado FT-027-Gestão de Projetos e Qualidade da Faculdade de Tecnologia-UNICAMP*, 98:101.

GUPTA, V., FERNANDEZ-CREHUET, J. M., HANNE, T., e TELESKO, R. (2020). Fostering product innovations in software startups through freelancer supported requirement engineering. <https://www.sciencedirect.com/science/article/pii/S2590123020300815>. Acesso em: 12/05/2023.

HOSSEINI, M., SHAHRI, A., PHALP, K., e ALI, R. (2018). Engineering transparency requirements: A modelling and analysis framework. <https://www.sciencedirect.com/science/article/pii/S0306437916305282>. Information Systems Engineering: selected papers from CAiSE 2016.

HRON, M. e OBWEGESER, N. (2022). Why and how is scrum being adapted in practice: A systematic review. <https://www.sciencedirect.com/science/article/pii/S0164121221002077>. Acesso em: 12/05/2023.

HULL, E., JACKSON, K., e DICK, J. (2010). *Requirements Engineering*. Springer Science Business Media.

IEEE (1990). IEEE standard for software engineering: IEEE std 610-1990.

KANG, B., CRILLY, N., NING, W., e KRISTENSSON, P. O. (2023). Prototyping to elicit user requirements for product development: Using

head-mounted augmented reality when designing interactive devices.

<https://www.sciencedirect.com/science/article/pii/S0142694X22000679>. Acesso em: 12/05/2023.

KITCHENHAM, B., PEARL BRERETON, O., BUDGEN, D., TURNER, M., BAILEY, J., e LINKMAN, S. (2009). Systematic literature reviews in software engineering – a systematic literature review.

<https://www.sciencedirect.com/science/article/pii/S0950584908001390>. Special Section - Most Cited Articles in 2002 and Regular Research Papers.

KURNIAWAN, A. (2019). *Introduction to Raspberry Pi*, páginas 1–25. Apress, Berkeley, CA.

LEFFINGWELL, D. (2011). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley Professional, 1st. edição.

LIEBEL, G. e KNAUSS, E. (2023). Aspects of modelling requirements in very-large agile systems engineering.

<https://www.sciencedirect.com/science/article/pii/S0164121223000237>. Acesso em: 12/05/2023.

MAALEM, S. e ZAROOUR, N. (2016). Challenge of validation in requirements engineering.

<https://www.sciencedirect.com/science/article/pii/S2352664516300025>. Special issue on Pattern Analysis and Intelligent Systems – With revised selected papers of the PAIS conference.

MAHAJU, S., CARVER, J. C., e BRADSHAW, G. L. (2023). Human error management in requirements engineering: Should we fix the people, the processes, or the environment?

<https://www.sciencedirect.com/science/article/pii/S0950584923000770>. Acesso em: 12/05/2023.

MAREW, T., LEE, J.-S., e BAE, D.-H. (2009). Tactics based approach for integrating non-functional requirements in object-oriented analysis and design.

<https://www.sciencedirect.com/science/article/pii/S0164121209000533>. SI: YAU.

MCROBERTS, M. (2011). *Arduino básico*. Novatec Editora, São Paulo, 1. edição.

MELEGATI, J., GOLDMAN, A., KON, F., e WANG, X. (2019). A model of requirements engineering in software startups.

<https://www.sciencedirect.com/science/article/pii/S095058491930028X>. Acesso em: 12/05/2023.

- MICH, L., SAKHNINI, V., e BERRY, D. (2023). To group or not to group? group sizes for requirements elicitation. <https://www.sciencedirect.com/science/article/pii/S0950584923000836>. Acesso em: 12/05/2023.
- MIRITZ¹, L. A. D. (2016). Programação de sistemas embarcados usando microcontroladores: um estudo de caso.
- NASCIMENTO, M. H. D. S. (2020). Análise de dados coletados por dispositivos de internet das coisas aplicada a sistemas embarcados. Instituto Federal de Ciência e Tecnologia de Pernambuco.
- NUNES, R. D. (2017). A implantação das metodologias ágeis de desenvolvimento de software scrum e extreme programming(xp): uma alternativa para pequenas empresas do setor de tecnologia da informação. *ForScience*, 4(2).
- PAUL, J., LIM, W. M., O'CASS, A., HAO, A. W., e BRESCIANI, S. (2021). Scientific procedures and rationales for systematic literature reviews (spar-4-slr). *International Journal of Consumer Studies*, 45(4):O1–O16.
- PRATES, L. D. A. (2012). Diretrizes para desenvolvimento de software para sistemas embarcados.
- PRIKLADNICKI, R., WILLI, R., e MILANI, F. (2014). *Métodos ágeis para desenvolvimento de software*. Bookman Editora.
- RAGO, A., DIAZ-PACE, J. A., e MARCOS, C. (2019). Do concern mining tools really help requirements analysts? an empirical study of the vetting process. <https://www.sciencedirect.com/science/article/pii/S0164121219301359>. Acesso em: 12/05/2023.
- SANCHES, F. D. A. (2019). Metodologias Ágeis no desenvolvimento de software. *Boletim do Gerenciamento*, 8(8):35–44.
- SHRIVASTAVA, A., JAGGI, I., KATOCH, N., GUPTA, D., e GUPTA, S. (2021). A systematic review on extreme programming. In *Journal of Physics: Conference Series*, volume 1969, página 012046. IOP Publishing.
- SILVA, W. A. E. (2019). Sistema embarcado de diagnóstico veicular integrado a ambientes inteligentes. Dissertação de mestrado, Universidade Federal do Amazonas, Manaus (AM). Dissertação de Mestrado em Engenharia Elétrica.
- SOMMERVILLE, I. (2011). *Engenharia de software*. Pearson Education, São Paulo, 9. edição.

- SOMÉ, S. S. (2006). Supporting use case based requirements engineering. <https://www.sciencedirect.com/science/article/pii/S0950584905000285>. Acesso em: 12/05/2023.
- SOUSA, A., AGRA, C., MELO, J. J. D., e ALENCAR, F. M. R. (2015). Elicitação e especificação de requisitos em sistemas embarcados: Uma revisão sistemática. In *Workshop em Engenharia de Requisitos*.
- STOPA, G. R. e RACHID, C. L. (2019). Scrum: Metodologia ágil como ferramenta de gerenciamento de projetos. *CES Revista*, 33(1):302–323.
- TAKAYA, S., KAMISHIMA, Y., MACHIDA, H., WATANABE, D., e ASAYAMA, T. (2016). Determination of in-service inspection requirements for fast reactor components using system based code concept. <https://www.sciencedirect.com/science/article/pii/S0029549316301431>. Acesso em: 12/05/2023.
- WOHLIN, C., RUNESON, P., HOST, M., OHLSSON, M., REGNELL, B., e WESSLEN, A. (2012). *Experimentation in software engineering: An introduction*. Springer, 2. edição.
- YANG, Y., LIU, B., WANG, J., CHEN, Y., e REN, Y. (2023). An improved multi-objective brainstorming algorithm with the application of rapeseed germination characteristics optimization. <https://www.sciencedirect.com/science/article/pii/S0168169923002533>. Acesso em: 12/05/2023.
- ZAYAT, W. e SENVAR, O. (2020). Framework study for agile software development via scrum and kanban. *International journal of innovation and technology management*, 17(04):2030002.

Anexo 1

Lista de artigos finais da Revisão Sistemática da Literatura

Mengjiao Shen, Wenrong Yang, Guoping Rong, and Dong Shao. 2012. Applying agile methods to embedded software development: a systematic review. In Proceedings of the Second International Workshop on Software Engineering for Embedded Systems (SEES '12). IEEE Press, 30–36.

Muhammad Nouman Zafar, Wasif Afzal, and Eduard Enoiu. 2021. Towards a workflow for model-based testing of embedded systems. In Proceedings of the 12th International Workshop on Automating TEST Case Design, Selection, and Evaluation (A-TEST 2021). Association for Computing Machinery, New York, NY, USA, 33–40. <https://doi.org/10.1145/3472672.3473956>

Hajer Berhouma. 2021. A Generic Model for Software Documentation and its Application in Embedded Systems Developed with Scrum. In Proceedings of the 9th International Conference on Software and Information Engineering (IC-SIE '20). Association for Computing Machinery, New York, NY, USA, 33–36. <https://doi.org/10.1145/3436829.3436858>

Mourad Mounir, Akram Salah, Amr Kamel, and Hanan Moussa. 2021. Framework to Measure Agile Software Process Effectiveness in Critical Systems Development. In Proceedings of the 9th International Conference on Software and

Information Engineering (ICSIE '20). Association for Computing Machinery, New York, NY, USA, 25–32. <https://doi.org/10.1145/3436829.3436853>

O. Doss and T. P. Kelly. 2016. Challenges and Opportunities in Agile Development in Safety Critical Systems: A Survey. *SIGSOFT Softw. Eng. Notes* 41, 2 (March 2016), 30–31. <https://doi.org/10.1145/2894784.2894798>

Hafiza Maria Maqsood, Eduardo Martins Guerra, Xiaofeng Wang, and Andrea Bondavalli. 2020. Patterns for Development of Safety-Critical Systems with Agile: Trace Safety Requirements and Perform Automated Testing. In *Proceedings of the European Conference on Pattern Languages of Programs 2020 (EuroPLoP '20)*. Association for Computing Machinery, New York, NY, USA, Article 3, 1–6. <https://doi.org/10.1145/3424771.3424800>

Daniel Russo, Gerolamo Taccogna, Paolo Ciancarini, Angelo Messina, and Giancarlo Succi. 2018. Contracting agile developments for mission critical systems in the public sector. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS '18)*. Association for Computing Machinery, New York, NY, USA, 47–56. <https://doi.org/10.1145/3183428.3183435>

A. Noyer, P. Iyengar, E. Pulvermuller, J. Engelhardt, F. Pramme and G. Bikker, "A model-based workflow from specification until validation of timing requirements in embedded software systems," 10th IEEE International Symposium on Industrial Embedded Systems (SIES), Siegen, Germany, 2015, pp. 1-4, doi: 10.1109/SIES.2015.7185056.

T. Pereira, A. Sousa, R. Silva, D. Albuquerque, F. Alencar and J. Castro, "A Metamodel to Guide a Requirements Elicitation Process for Embedded Systems," 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC), Coimbra, Portugal, 2018, pp. 101-109, doi: 10.1109/QUATIC.2018.00023.

D. Kildishev and A. Khoroshilov, "Developing Requirements Management Tool for Safety-Critical Systems," 2019 Actual Problems of Systems and Software Engi-

neering (APSSE), Moscow, Russia, 2019, pp. 50-57,
doi: 10.1109/APSSE47353.2019.00013.

H. Zhang, T. Yue, S. Ali, J. Wu and C. Liu, "A Restricted Natural Language Based Use Case Modeling Methodology for Real-Time Systems,"2017 IEEE/ACM 9th International Workshop on Modelling in Software Engineering (MiSE), Buenos Aires, Argentina, 2017, pp. 5-11, doi: 10.1109/MiSE.2017.9.

Jiale Zhou, "Requirements development and management of embedded real-time systems,"2014 IEEE 22nd International Requirements Engineering Conference (RE), Karlskrona, Sweden, 2014, pp. 479-484, doi: 10.1109/RE.2014.6912302.

D. Blouin and H. Giese, "Combining Requirements, Use Case Maps and AADL Models for Safety-Critical Systems Design,"2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Limassol, Cyprus, 2016, pp. 266-274, doi: 10.1109/SEAA.2016.15.

T. Pereira, D. Albuquerque, A. Sousa, F. Alencar and J. Castro, "Towards a Metamodel for a Requirements Engineering Process of Embedded Systems,"2016 VI Brazilian Symposium on Computing Systems Engineering (SBESC), João Pessoa, Brazil, 2016, pp. 93-100, doi: 10.1109/SBESC.2016.022.

M. Li, F. Batmaz, L. Guan, A. Grigg, M. Ingham and P. Bull, "Model-based systems engineering with requirements variability for embedded real-time systems,"2015 IEEE International Model-Driven Requirements Engineering Workshop (MoDRE), Ottawa, ON, Canada, 2015, pp. 1-10, doi: 10.1109/MoDRE.2015.7343874.

M. A. Wehrmeister and G. R. Berkenbrock, "AMoDE-RT: Advancing Model-Driven Engineering for embedded real-time systems,"16th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2013), Paderborn, Germany, 2013, pp. 1-7,
doi: 10.1109/ISORC.2013.6913233.

J. On, S. Woo and M. Lee, "A Graphical Language to Integrate Process Algebra and State Machine Views for Specification and Verification of Distributed Real-Time Systems,"2012 IEEE 36th Annual Computer Software and Applications Conference

Workshops, Izmir, Turkey, 2012, pp. 218-223, doi: 10.1109/COMPSACW.2012.48.

M. Bajer, "Embedded software development in research environment: A practical guide for non-experts," 2014 3rd Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2014, pp. 66-71, doi: 10.1109/MECO.2014.6862660.

A. Hazra, P. Ghosh and P. Dasgupta, "Reliability annotations to formal specifications of context-sensitive safety properties in embedded systems," Proceeding of the 2012 Forum on Specification and Design Languages, Vienna, Austria, 2012, pp. 36-43.

G. Loniewski, E. Borde, D. Blouin and E. Insfran, "Model-Driven Requirements Engineering for Embedded Systems Development," 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, Santander, Spain, 2013, pp. 236-243, doi: 10.1109/SEAA.2013.48.

S. M. A. Shah, D. Sundmark, B. Lindström and S. F. Andler, "Robustness Testing of Embedded Software Systems: An Industrial Interview Study," in IEEE Access, vol. 4, pp. 1859-1871, 2016, doi: 10.1109/ACCESS.2016.2544951.

J. Zhou, Y. Lu and K. Lundqvist, "The observer-based technique for requirements validation in embedded real-time systems," 2014 IEEE 1st International Workshop on Requirements Engineering and Testing (RET), Karlskrona, Sweden, 2014, pp. 47-54, doi: 10.1109/RET.2014.6908679.

C. Binder, K. Polanec, B. Brankovic, C. Neureiter, G. Lastro and A. Lüder, "Enabling model-based requirements engineering in a complex industrial System of Systems environment," 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vasteras, Sweden, 2021, pp. 1-6, doi: 10.1109/ETFA45728.2021.9613700.

S. M. Ågren, E. Knauss, R. Heldal, P. Pelliccione, G. Malmqvist and J. Bodén, "The Manager Perspective on Requirements Impact on Automotive Systems Development Speed," 2018 IEEE 26th International Requirements Engineering Conference (RE), Banff, AB, Canada, 2018, pp. 17-28, doi: 10.1109/RE.2018.00-55.

R. Sinha, S. Patil, C. Pang, V. Vyatkin and B. Dowdeswell, "Requirements engineering of industrial automation systems: Adapting the CESAR requirements meta model for safety-critical smart grid software," IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society, Yokohama, Japan, 2015, pp. 002172-002177, doi: 10.1109/IECON.2015.7392423.

J. Doerr, A. Hess and M. Koch, "RE and Society - A Perspective on RE in Times of Smart Cities and Smart Rural Areas," 2018 IEEE 26th International Requirements Engineering Conference (RE), Banff, AB, Canada, 2018, pp. 100-111, doi: 10.1109/RE.2018.00020.

J. Zhou, K. Hänninen, K. Lundqvist, Y. Lu, L. Provenzano and K. Forsberg, "An environment-driven ontological approach to requirements elicitation for safety-critical systems," 2015 IEEE 23rd International Requirements Engineering Conference (RE), Ottawa, ON, Canada, 2015, pp. 247-251, doi: 10.1109/RE.2015.7320431.

E. S. Grant, "Requirements engineering for safety critical systems: An approach for avionic systems," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 2016, pp. 991-995, doi: 10.1109/CompComm.2016.7924853.

M. Goodrum, J. Cleland-Huang, R. Lutz, J. Cheng and R. Metoyer, "What Requirements Knowledge Do Developers Need to Manage Change in Safety-Critical Systems?," 2017 IEEE 25th International Requirements Engineering Conference (RE), Lisbon, Portugal, 2017, pp. 90-99, doi: 10.1109/RE.2017.65.

M. Kaessmeyer, D. S. V. Moncada and M. Schurius, "Evaluation of a Systematic Approach in Variant Management for Safety-Critical Systems Development," 2015 IEEE 13th International Conference on Embedded and Ubiquitous Computing, Porto, Portugal, 2015, pp. 35-43, doi: 10.1109/EUC.2015.12.

S. P. Nanda and E. S. Grant, "A Survey of Formal Specification Application to Safety Critical Systems," 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, 2019, pp. 296-302, doi: 10.1109/INFOCT.2019.8711369.

L. Ordinez, O. Alimenti, E. Rinland, M. Gomez and J. Marchetti, "Modeling and Specifying Requirements for Cyber-Physical Systems," in *IEEE Latin America Transactions*, vol. 11, no. 1, pp. 625-632, Feb. 2013, doi: 10.1109/TLA.2013.6502874.

A. Shahrokni and R. Feldt, "Industrial Challenges with Quality Requirements in Safety Critical Software Systems," 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, Santander, Spain, 2013, pp. 78-81, doi: 10.1109/SEAA.2013.57.

K. N. Hodel, J. Reinaldo Da Silva, L. R. Yoshioka, J. F. Justo and M. M. D. Santos, "FAT-AES: Systematic Methodology of Functional Testing for Automotive Embedded Software," in *IEEE Access*, vol. 10, pp. 74259-74279, 2022, doi: 10.1109/ACCESS.2021.3128431.

Dagný Hauksdóttir, Niels Henrik Mortensen, Poul Erik Nielsen, Identification of a reusable requirements structure for embedded products in a dynamic market environment, *Computers in Industry*, Volume 64, Issue 4, 2013, Pages 351-362, ISSN 0166-3615, <https://doi.org/10.1016/j.compind.2012.10.008>.

Danjiang Zhu, Huobin Tan, Shuzhen Yao, Petri Nets-based method to elicit component-interaction related safety requirements in safety-critical systems, *Computers Electrical Engineering*, Volume 71, 2018, Pages 162-172, ISSN 0045-7906, <https://doi.org/10.1016/j.compeleceng.2018.07.019>.

Noyer, A., Iyengar, P., Engelhardt, J. et al. A model-based framework encompassing a complete workflow from specification until validation of timing requirements in embedded software systems. *Software Qual J* 25, 671–701 (2017). <https://doi.org/10.1007/s11219-016-9323-9>

Dixit, M.G., Ramesh, S. Dasgupta, P. Time-budgeting: a component based development methodology for real-time embedded systems. *Form Asp Comp* 26, 591–621 (2014). <https://doi.org/10.1007/s00165-012-0273-0>

Ågren, S.M., Knauss, E., Heldal, R. et al. The impact of requirements on systems development speed: a multiple-case study in automotive. *Requirements Eng* 24,

315–340 (2019). <https://doi.org/10.1007/s00766-019-00319-8>

Park, S. Software Requirement Specification Based on a Gray Box for Embedded Systems: A Case Study of a Mobile Phone Camera Sensor Controller. *Computers* 2019, 8, 20. <https://doi.org/10.3390/computers8010020>

Edith Zavala, Xavier Franch, Jordi Marco, Alessia Knauss, Daniela Damian, SACRE: Supporting contextual requirements' adaptation in modern self-adaptive systems in the presence of uncertainty at runtime, *Expert Systems with Applications*, Volume 98, 2018, Pages 166-188, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2018.01.009>.

Venter, C. A Critical Systems Approach to Elicit User-Centric Business Intelligence Business Requirements. *Syst Pract Action Res* 32, 481–500 (2019). <https://doi.org/10.1007/s11213-018-9468-5>

Muhammad Rashid, Muhammad Waseem Anwar, Aamir M. Khan, Toward the tools selection in model based system engineering for embedded systems—A systematic literature review, *Journal of Systems and Software*, Volume 106, 2015, Pages 150-163, ISSN 0164-1212, <https://doi.org/10.1016/j.jss.2015.04.089>.

Westman, J., Nyberg, M. Providing tool support for specifying safety-critical systems by enforcing syntactic contract conditions. *Requirements Eng* 24, 231–256 (2019). <https://doi.org/10.1007/s00766-017-0286-6>

D. Iqbal, A. Abbas, M. Ali, M. U. S. Khan and R. Nawaz, "Requirement Validation for Embedded Systems in Automotive Industry Through Modeling,"in *IEEE Access*, vol. 8, pp. 8697-8719, 2020, doi: 10.1109/ACCESS.2019.2963774.

Y. Yang, Q. Zu, W. Ke, M. Zhang and X. Li, "Real-Time System Modeling and Verification Through Labeled Transition System Analyzer,"in *IEEE Access*, vol. 7, pp. 26314-26323, 2019, doi: 10.1109/ACCESS.2019.2899761.

Chen, D., Feng, L., Qureshi, T.N. et al. An architectural approach to the analysis, verification and validation of software intensive embedded systems. *Computing* 95, 649–688 (2013). <https://doi.org/10.1007/s00607-013-0314-4>

Roque, A.d.S., Pohren, D., Freitas, E.P. et al. An Approach to Address Safety as

Non-Functional Requirements in Distributed Vehicular Control Systems. *J Control Autom Electr Syst* 30, 700–715 (2019). <https://doi.org/10.1007/s40313-019-00483-w>

Wang, Z., Pu, G., Li, J. et al. A novel requirement analysis approach for periodic control systems. *Front. Comput. Sci.* 7, 214–235 (2013). <https://doi.org/10.1007/s11704-013-2008-1>

Z. Chen, I. Li, H. Zhang, S. Preum, J. A. Stankovic and M. Ma, "CitySpec: An Intelligent Assistant System for Requirement Specification in Smart Cities,"2022 IEEE International Conference on Smart Computing (SMARTCOMP), Helsinki, Finland, 2022, pp. 32-39, doi: 10.1109/SMARTCOMP55677.2022.00020.

J. Yang, X. Chen and L. Yin, "Eliciting Timing Requirements for Cyber-Physical Systems: a Multiform Time based Approach,"2021 International Symposium on Theoretical Aspects of Software Engineering (TASE), Shanghai, China, 2021, pp. 199-206, doi: 10.1109/TASE52547.2021.00024.

Singh, NK, Lawford, M, Maibaum, TSE, Wassying, A. A formal approach to rigorous development of critical systems. *J Softw Evol Proc.* 2021; 33:e2334. <https://doi.org/10.1002/smr.2334>

Z. Jin, X. Chen, Z. Li and Y. Yu, "RE4CPS: Requirements Engineering for Cyber-Physical Systems,"2019 IEEE 27th International Requirements Engineering Conference (RE), Jeju, Korea (South), 2019, pp. 496-497, doi: 10.1109/RE.2019.00072.

S. Fritz, F. Weber and J. Ovtcharova, "A Guideline for the Requirements Engineering Process of SMEs Regarding to the Development of CPS,"2019 8th International Conference on Industrial Technology and Management (ICITM), Cambridge, UK, 2019, pp. 85-94, doi: 10.1109/ICITM.2019.8710732.

S. u. Rehman, C. Allgaier and V. Gruhn, "Security Requirements Engineering: A Framework for Cyber-Physical Systems,"2018 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2018, pp. 315-320, doi: 10.1109/FIT.2018.00062.

M. T. Span, L. O. Mailloux, M. R. Grimaila and W. B. Young, "A Systems Security Approach for Requirements Analysis of Complex Cyber-Physical Systems," 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Glasgow, UK, 2018, pp. 1-8, doi: 10.1109/CyberSecPODS.2018.8560682.

A. A. A. Saeed and S. -W. Lee, "Non-functional Requirements Trade-Off in Self-Adaptive Systems," 2018 4th International Workshop on Requirements Engineering for Self-Adaptive, Collaborative, and Cyber Physical Systems (RESACS), Banff, AB, Canada, 2018, pp. 9-15, doi: 10.1109/RESACS.2018.00007.

K. Czarnecki, "Requirements Engineering in the Age of Societal-Scale Cyber-Physical Systems: The Case of Automated Driving," 2018 IEEE 26th International Requirements Engineering Conference (RE), Banff, AB, Canada, 2018, pp. 3-4, doi: 10.1109/RE.2018.00-57.

Y. Ma, J. Wang, Y. Qu and T. Zhang, "Testing Requirement Analysis Method Based on Signal Path for High-reliability Embedded Software," 2018 International Computers, Signals and Systems Conference (ICOMSSC), Dalian, China, 2018, pp. 601-607, doi: 10.1109/ICOMSSC45026.2018.8941625.

M. Winokur, Z. Lando and A. Modai, "Seamless requirements flow down from system to software components in a cyber physical system," 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2018, pp. 1-4, doi: 10.1109/MECO.2018.8405971.

Christos Tsigkanos, Nianyu Li, Zhi Jin, Zhenjiang Hu, and Carlo Ghezzi. 2018. On early statistical requirements validation of cyber-physical space systems. In Proceedings of the 4th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS '18). Association for Computing Machinery, New York, NY, USA, 13–18. <https://doi.org/10.1145/3196478.3196485>

A. L. Borgne, N. Belloir, J. -M. Bruel and T. Nguyen, "Formal Requirements Engineering for Smart Industries: Toward a Model-Based Graphical Language," 2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trus-

ted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDDCom/IoP/SmartWorld), Toulouse, France, 2016, pp. 1028-1032, doi: 10.1109/UIC-ATC-ScalCom-CBDDCom-IoP-SmartWorld.2016.0160.

Iyas Ibriwesh, Khairi Shazwan Dollmat, Sin-Ban Ho, Ian Chai, and Chuie-Hong Tan. 2017. A Controlled Experiment on Requirements Elicitation in Electronic Markets. In Proceedings of the 8th International Conference on E-Education, E-Business, E-Management and E-Learning (IC4E '17). Association for Computing Machinery, New York, NY, USA, 56–60. <https://doi.org/10.1145/3026480.3026496>

A. R. Mat and A. Bujang Masli, "SOFL-based approach for requirements analysis of brain tumor treatment system,"2012 International Conference on Computer Information Science (ICCIS), Kuala Lumpur, Malaysia, 2012, pp. 885-888, doi: 10.1109/ICCISci.2012.6297151.

D. Mougouei, M. Moghtadaei and S. Moradmand, "A goal-based modeling approach to develop security requirements of fault tolerant security-critical systems,"2012 International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia, 2012, pp. 200-205, doi: 10.1109/ICCCE.2012.6271180.

Roberto Avila Paldês, Edna Dias Canedo, Fernando de Albuquerque Guimarães, and Angélica Toffano Seidel Calazans. 2021. Functional Requirements Elicitation in IoT Systems: a follow-up study. In Proceedings of the XIX Brazilian Symposium on Software Quality (SBQS '20). Association for Computing Machinery, New York, NY, USA, Article 14, 1–10. <https://doi.org/10.1145/3439961.3439975>

Kostas Kontogiannis, Daniel Amyot, and John Mylopoulos. 2021. Software techniques for engineering cyber-physical systems. In Proceedings of the 31st Annual International Conference on Computer Science and Software Engineering (CASCON '21). IBM Corp., USA, 289–290.

