

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LUCAS DA SILVA FLORES

**Ferramentas de Gestão de Projetos de
Desenvolvimento de Software: Um
Mapeamento Sistemático da Literatura**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Profa. Dra. Lucineia Heloisa Thom

Porto Alegre
2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^a. Patricia Helena Lucas Pranke

Pró-Reitoria de Ensino (Graduação e Pós-Graduação): Prof^a. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Rodrigo Machado

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Sometimes it is the people no one can imagine anything
of who do the things no one can imagine.”*

— ALAN TURING

AGRADECIMENTOS

Agradeço à minha orientadora, prof^a. Lucineia Heloisa Thom, e ao grupo de pesquisa da professora pela oportunidade de trabalhar com essas pessoas, pelo apoio e pela paciência durante todo o desenvolvimento deste trabalho. A todos os professores que fizeram com que eu me apaixonasse pela academia, pelo ato de ensinar e aprender. Aos meus pais, Vera Lúcia Lourenço da Silva e Jorge Moraes Flores, por todo o suporte, pelo carinho, pelo respeito, pelo amor e por serem a base de tudo que eu fiz até hoje. A todos os amigos que fiz durante a graduação e que estão comigo até hoje, ajudando em todos os momentos. Aos sócios da Vortigo, por sempre me apoiarem durante os últimos semestres da graduação com meus horários complicados e por todo suporte dentro da empresa.

RESUMO

No período de 2011 a 2021, o mercado mundial de *software* e serviços cresceu mais de 150%, atingindo 1,47 trilhões de dólares só em 2021. Com esse investimento, os desafios no desenvolvimento de *software* aumentam, em conjunto com a complexidade dos sistemas. Gerenciar projetos complexos de desenvolvimento de *software* também se torna um desafio. Com isso, as organizações procuram se adaptar ao mercado para finalizar projetos de forma rápida, com menos custos e maior eficiência, para atender às necessidades dos *stakeholders*. Essa crescente demanda por agilidade na entrega dos projetos apresenta as ferramentas de gestão de projetos de desenvolvimento de *software*, como *Microsoft Project*, *Planner* e *JIRA*. São ferramentas que auxiliam gestores de projeto a distribuir, priorizar e acompanhar as tarefas realizadas para atingir os objetivos do projeto. A escolha de ferramentas que farão parte da gestão de um projeto é importante, pois ferramentas que possuem problemas de usabilidade podem afetar negativamente um projeto de *software*, desmotivando o time a utilizar a ferramenta. Este trabalho apresenta um mapeamento sistemático da literatura sobre essas ferramentas, trazendo uma análise quantitativa em relação aos trabalhos encontrados e suas contribuições, buscando trabalhos sobre usabilidade e utilidade em relação a essas ferramentas. São destacados os tópicos mais encontrados na literatura, e como os trabalhos podem ser classificados dentro das atividades do processo de *software*. O tópico usabilidade se destaca como o tópico mais encontrado nos artigos selecionados. A atividade de projeto e implementação de *software* se destaca como a atividade que possui mais publicações sobre ferramentas de gestão de projetos de desenvolvimento de *software*. Por fim, a atividade de verificação de *software* se destaca dentro do período, entre 2011 e 2021, com tendência crescente de publicações sobre o tema.

Palavras-chave: Mapeamento Sistemático. Ferramentas de Gestão de Processos. Desenvolvimento de Software. Processo de Software. Usabilidade. Utilidade.

Software Development Project Management Tools: A Systematic Mapping of Literature

ABSTRACT

Between 2011 and 2021, the world market for software and services grew more than 150%, moving 1.47 trillion dollars in 2021 alone. With this investment, the challenges in software development increase, together with several factors, such as the complexity of the systems. Managing complex software development projects also becomes a challenge. With this, organizations seek to adapt to the market to complete projects quickly, with less cost and greater efficiency, to meet the needs of stakeholders. This growing demand for agility in project delivery introduces software development project management tools such as *Microsoft Project*, *Planner*, and *JIRA*. They are tools that help project managers to distribute, prioritize and monitor the tasks performed to achieve project objectives. The choice of tools that will be part of the project management is important because tools with usability problems can negatively affect a software project, demotivating the team to use the tool. This work presents a systematic mapping of the literature on these tools, bringing a quantitative analysis of the works found and their contributions, seeking works on the usability and usefulness of these tools. The topics most found in the literature are highlighted, and how the works can be classified within the software process activities. The usability topic stands out as the most found topic in the selected articles. The software design and implementation activity stand out as the activity that has more publications on software development project management tools. Finally, the software verification activity stands out within the period, between 2011 and 2021, with a growing trend of publications on the subject.

Keywords: Systematic Mapping. Project Managements Tools. Software Development. Software Process. Usability. Utility.

LISTA DE FIGURAS

Figura 1.1 Exemplo simplificado do mapeamento sistemático de literatura proposto no trabalho.	13
Figura 2.1 Estágios do modelo de processo de <i>software</i> em cascata	19
Figura 3.1 String de pesquisa aplicada em cada repositório	25
Figura 3.2 Critérios de exclusão aplicados aos artigos encontrados	26
Figura 3.3 Critérios de inclusão aplicados aos artigos encontrados	27
Figura 3.4 Exemplo de classificação dos artigos por ano de publicação	27
Figura 3.5 Exemplo de classificação dos artigos por tópico encontrado	28
Figura 3.6 Exemplo de classificação por processo dos artigos	28
Figura 4.1 Artigos por ano	30
Figura 4.2 Tópicos mais encontrados nos artigos de 2011	31
Figura 4.3 Tópicos mais encontrados nos artigos de 2012	32
Figura 4.4 Tópicos mais encontrados nos artigos de 2013	32
Figura 4.5 Tópicos mais encontrados nos artigos de 2014	33
Figura 4.6 Tópicos mais encontrados nos artigos de 2015	33
Figura 4.7 Tópicos mais encontrados nos artigos de 2016	34
Figura 4.8 Tópicos mais encontrados nos artigos de 2017	35
Figura 4.9 Tópicos mais encontrados nos artigos de 2018	36
Figura 4.10 Tópicos mais encontrados nos artigos de 2019	37
Figura 4.11 Tópicos mais encontrados nos artigos de 2020	38
Figura 4.12 Tópicos mais encontrados nos artigos de 2021	39
Figura 4.13 Os 10 tópicos mais encontrados nos artigos selecionados.	40
Figura 4.14 Linha do tempo de <i>AGILE SOFTWARE DEVELOPMENT</i>	40
Figura 4.15 Linha do tempo de <i>AGILE</i>	41
Figura 4.16 Linha do tempo de <i>PROJECT MANAGEMENT</i>	42
Figura 4.17 Linha do tempo de <i>RISK MANAGEMENT</i>	42
Figura 4.18 Linha do tempo de <i>SCRUM</i>	43
Figura 4.19 Linha do tempo de <i>SOFTWARE DEVELOPMENT</i>	44
Figura 4.20 Linha do tempo de <i>SOFTWARE ENGINEERING</i>	45
Figura 4.21 Linha do tempo de <i>SOFTWARE QUALITY</i>	45
Figura 4.22 Linha do tempo de <i>SOFTWARE</i>	46
Figura 4.23 Linha do tempo de <i>USABILITY</i>	47
Figura 4.24 Artigos por atividade do processo de <i>software</i>	47
Figura 4.25 Tópicos mais encontrados na atividade de especificação de <i>software</i>	48
Figura 4.26 Tópicos encontrados na atividade de projeto e implementação (1/2)	49
Figura 4.27 Tópicos encontrados na atividade de projeto e implementação (2/2)	50
Figura 4.28 Tópicos mais encontrados na atividade de validação de <i>software</i>	51
Figura 4.29 Tópicos mais encontrados na atividade de evolução de <i>software</i>	52
Figura 4.30 Linha do tempo da atividade de especificação de <i>software</i>	53
Figura 4.31 Linha do tempo da atividade de projeto e implementação de <i>software</i>	54
Figura 4.32 Linha do tempo da atividade de validação de <i>software</i>	56
Figura 4.33 Linha do tempo da atividade de evolução de <i>software</i>	57
Figura A.1 Tópicos mais encontrados nos artigos selecionados. (1/2).....	73
Figura A.2 Tópicos mais encontrados nos artigos selecionados. (2/2)	74

LISTA DE TABELAS

Tabela B.1	Tabela de artigos selecionados (1/36)	75
Tabela B.2	Tabela de artigos selecionados (2/36)	76
Tabela B.3	Tabela de artigos selecionados (3/36)	77
Tabela B.4	Tabela de artigos selecionados (4/36)	78
Tabela B.5	Tabela de artigos selecionados (5/36)	79
Tabela B.6	Tabela de artigos selecionados (6/36)	80
Tabela B.7	Tabela de artigos selecionados (7/36)	81
Tabela B.8	Tabela de artigos selecionados (8/36)	82
Tabela B.9	Tabela de artigos selecionados (9/36)	83
Tabela B.10	Tabela de artigos selecionados (10/36)	84
Tabela B.11	Tabela de artigos selecionados (11/36)	85
Tabela B.12	Tabela de artigos selecionados (12/36)	86
Tabela B.13	Tabela de artigos selecionados (13/36)	87
Tabela B.14	Tabela de artigos selecionados (14/36)	88
Tabela B.15	Tabela de artigos selecionados (15/36)	89
Tabela B.16	Tabela de artigos selecionados (16/36)	90
Tabela B.17	Tabela de artigos selecionados (17/36)	91
Tabela B.18	Tabela de artigos selecionados (18/36)	92
Tabela B.19	Tabela de artigos selecionados (19/36)	93
Tabela B.20	Tabela de artigos selecionados (20/36)	94
Tabela B.21	Tabela de artigos selecionados (21/36)	95
Tabela B.22	Tabela de artigos selecionados (22/36)	96
Tabela B.23	Tabela de artigos selecionados (23/36)	97
Tabela B.24	Tabela de artigos selecionados (24/36)	98
Tabela B.25	Tabela de artigos selecionados (25/36)	99
Tabela B.26	Tabela de artigos selecionados (26/36)	100
Tabela B.27	Tabela de artigos selecionados (27/36)	101
Tabela B.28	Tabela de artigos selecionados (28/36)	102
Tabela B.29	Tabela de artigos selecionados (29/36)	103
Tabela B.30	Tabela de artigos selecionados (30/36)	104
Tabela B.31	Tabela de artigos selecionados (31/36)	105
Tabela B.32	Tabela de artigos selecionados (32/36)	106
Tabela B.33	Tabela de artigos selecionados (33/36)	107
Tabela B.34	Tabela de artigos selecionados (34/36)	108
Tabela B.35	Tabela de artigos selecionados (35/36)	109
Tabela B.36	Tabela de artigos selecionados (36/36)	110

LISTA DE ABREVIATURAS E SIGLAS

TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
ABES	Associação Brasileira das Empresas de Software
BPM	Gerenciamento de Processos de Negócios
XP	Extreme Programming
US	Estória de Usuário

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Motivação.....	11
1.2 Objetivos	12
1.3 Organização do Trabalho.....	13
2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS.....	14
2.1 Gestão de Projetos.....	14
2.2 Ferramentas de Gestão de Projetos.....	15
2.3 Processo de Software	16
2.3.1 Especificação de Software	16
2.3.2 Projeto e implementação de Software.....	17
2.3.3 Validação de Software.....	17
2.3.4 Evolução de Software	18
2.3.5 Modelo de Processo de Software	18
2.4 Usabilidade e Utilidade.....	20
2.5 Desenvolvimento Ágil de Software	20
2.6 Trabalhos Relacionados.....	21
2.7 Resumo do Capítulo.....	22
3 METODOLOGIA DE PESQUISA	23
3.1 Perguntas de Pesquisa	23
3.2 Pesquisa e Seleção	24
3.2.1 Critérios de Exclusão	24
3.2.2 Critérios de Inclusão	25
3.3 Classificação e Visualização dos Dados.....	26
3.4 Resumo do Capítulo.....	29
4 RESULTADOS	30
4.1 Classificação Por Ano	30
4.2 Classificação Por Tópico.....	39
4.3 Classificação por Atividades do Processo de Software	47
4.3.1 Principais Tópicos por Atividade.....	48
4.3.2 Linha do Tempo por Atividade	53
4.4 Discussão e Perguntas de Pesquisa	57
4.5 Resumo do Capítulo.....	58
5 CONCLUSÃO	60
5.1 Contribuições e Limitações da Pesquisa	61
5.2 Trabalhos Futuros.....	61
REFERÊNCIAS.....	62
APÊNDICE A — TÓPICOS ENCONTRADOS MAIS DE UMA VEZ NOS ARTIGOS SELECIONADOS.....	73
APÊNDICE B — TABELA DOS ARTIGOS SELECIONADOS.....	75

1 INTRODUÇÃO

De acordo com Sommerville (2010), a gestão de projetos de *software* é parte essencial da engenharia de *software*. O processo de gerenciamento desses projetos pode ser complexo, conforme Silva et al. (2010). A qualidade do produto final desenvolvido depende de vários fatores, inclusive do processo aplicado durante o ciclo de vida do projeto de *software* desenvolvido.

Para auxiliar nesse gerenciamento, existem as *ferramentas de gestão de projetos de desenvolvimento de software*. Ferramentas como *Microsoft Project*, *Planner* e *JIRA* são exemplos de *softwares* que fazem parte do processo de gestão, auxiliando durante todo o ciclo de vida de um projeto de *software* (MISHRA; MISHRA, 2013).

Com a evolução dos projetos de *software* e demandas complexas, aliado a necessidade de entrega rápida e contínua do software, as ferramentas de gestão de projetos de *software* precisam ser úteis e fáceis de usar (MISHRA; MISHRA, 2013). Isso possibilita que as suas funcionalidades contribuam para a melhoria, agilidade e facilidade do processo de desenvolvimento de um novo sistema.

1.1 Motivação

De acordo com os dados da Associação Brasileira das Empresas de Software (ABES), em 2011 o mercado mundial de *software* e serviços atingiu o valor de 941 bilhões de dólares (ABES, 2012). Em 2021, o mercado mundial de *software* e serviços atingiu o valor de 1,47 trilhões de dólares, representando um crescimento de 156,22% entre 2011 e 2021 (ABES, 2022). Com esse investimento, os projetos de *software* e desafios na área se tornam mais complexos, mas ao mesmo tempo, precisam ser finalizados ou estarem funcionais de forma mais rápida, conforme Silva et al. (2010).

De acordo com Moreno, Afonso and Costa (2019), os desafios no desenvolvimento de *software* aumentam devido a diversos fatores, como a complexidade dos sistemas e a falta de profissionais capacitados. Gerenciar esses projetos de desenvolvimento de *software* também se torna uma tarefa complexa devido ao crescimento dos projetos. Conforme Moreno, Afonso and Costa (2019), as organizações procuram se adaptar ao mercado para finalizar projetos de forma rápida, com menos custos e maior eficiência, para atender às necessidades dos *stakeholders*.

Conforme Alomar et al. (2016), a crescente demanda por agilidade nas atividades da gestão de projetos introduz as ferramentas de gestão de projetos desenvolvimento de *software*. Essas ferramentas oferecem recursos que auxiliam o gestor do projeto a distribuir e priorizar tarefas, e gerenciar o andamento do projeto. Os autores também afirmam que a escolha de uma ferramenta de gestão de projetos de desenvolvimento de *software* é importante, pois essa ferramenta irá auxiliar, também, o gestor a atingir os objetivos do projeto, fornecendo soluções para gerir, planejar e acompanhar o projeto de desenvolvimento de *software*. De acordo com Alomar et al. (2016), ferramentas de gestão de projetos que possuem problemas de usabilidade, podem desmotivar o uso da ferramenta, afetando negativamente o progresso do projeto.

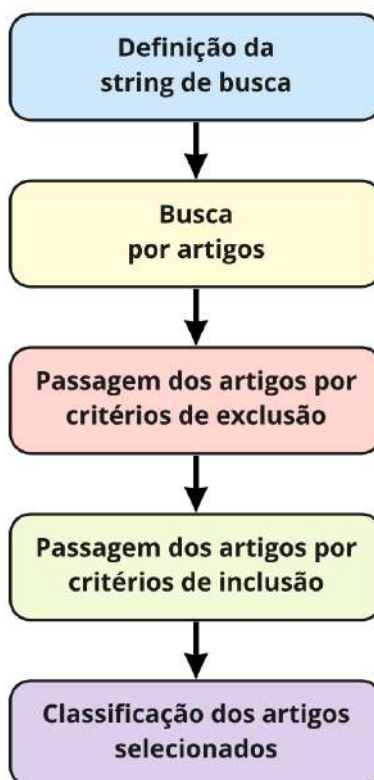
1.2 Objetivos

Este trabalho apresenta um mapeamento sistemático da literatura, o qual é ilustrado de forma simplificada na Figura 1.1. Inicialmente, é definida uma *string* de busca para procura de artigos nos repositórios escolhidos. Após a busca por artigos, esses trabalhos passam por critérios de exclusão e inclusão para definição dos artigos selecionados para a classificação. A partir dessa seleção, é feita a classificação dos trabalhos por ano, por tópico e, conforme definido por Sommerville (2010), por atividade do processo de *software*.

Com base no mapeamento sistemático proposto, o trabalho tem como objetivo apresentar contribuições sobre usabilidade e utilidade em conjunto com as ferramentas de gestão de projetos de *software*. Para isso, o trabalho apresenta análises quantitativas em relação aos tópicos mais encontrados nos trabalhos selecionados no mapeamento sistemático, reunindo a informação encontrada através das atividades do processo de *software*.

Ao associar as informações encontradas com as atividades do processo de *software*, é possível observar quais as atividades mais encontradas e estudadas na literatura e quais atividades foram menos exploradas na literatura, observando também tendências em relação às atividades do processo de *software*.

Figura 1.1: Exemplo simplificado do mapeamento sistemático de literatura proposto no trabalho.



Fonte: O Autor

1.3 Organização do Trabalho

O restante deste texto está organizado como segue. O Capítulo 2 apresenta a fundamentação teórica necessária para o entendimento do mapeamento sistemático da literatura apresentado neste trabalho. Também revisa o que existe na literatura sobre outros trabalhos relacionados e como eles contribuem para o assunto. A metodologia utilizada no mapeamento sistemático desse trabalho é apresentada no Capítulo 3, no qual aborda os critérios de inclusão de exclusão no processo que levou à seleção dos artigos e a descrição do processo de classificação dos artigos. O Capítulo 4 apresenta os resultados extraídos do mapeamento sistemático. O Capítulo 5 traz a conclusão do trabalho, limitações da pesquisa e perspectivas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Este capítulo apresenta a base teórica necessária para a compreensão desse trabalho, a definição de gestão de projeto, suas etapas e a definição de ferramentas de gestão de projetos. Apresenta também a definição de processos de *software*, e desenvolvimento ágil de *software*. Por fim, o capítulo apresenta os trabalhos relacionados ao mapeamento sistemático apresentado neste trabalho.

2.1 Gestão de Projetos

Conforme o *Project Management Institute* (PMI) PMI (2013), um projeto deve ter datas de início e fim, deve possuir custo definido, escopo claro e uma visão do que precisa ser construído, além de especificações que devem ser cumpridas.

Gestão de projetos é a aplicação de habilidade, conhecimento e ferramentas nas atividades do projeto (LEWIS, 2006). Essas atividades representam desde a inicialização do projeto, passando pelo planejamento, execução e monitoramento, até a finalização do projeto, conforme Sommerville (2010). O gestor de projeto é o facilitador, ou seja, a pessoa que possui a função de auxiliar o time a entender o que precisa ser feito. É um papel de liderança. A gestão de projetos, conforme (LEWIS, 2006), é dividida em etapas, apresentadas da seguinte forma:

- *Definição do problema*, etapa em que se identifica qual problema será solucionado pelo projeto desenvolvido. Nesta etapa se deve auxiliar o cliente a entender qual deve ser o resultado final esperado e se esse resultado será satisfatório.
- *Discussão sobre soluções*, etapa em que se discute as maneiras existentes, visando a solução do problema, quais as alternativas disponíveis, como diminuir o custo do projeto e se a solução irá de fato resolver o problema por completo ou apenas parte do problema.
- *Planejamento do projeto*, etapa em que se planeja o que precisa ser feito, por quem será feito, quanto custará e por quanto tempo será feito.
- *Execução do planejamento*, etapa em que as definições feitas na etapa de planejamento devem ser executadas. Esta etapa é crítica, pois erros nesta serão refletidos na execução do projeto, levando a problemas como aumento no custo e na data de entrega do projeto.

- *Monitoramento e controle do progresso*, etapa em que se acompanha o andamento do projeto. É ao longo desse acompanhamento que o planejamento é validado e reforçado, para ter certeza de que será cumprido. Problemas no planejamento são levantados durante esse acompanhamento e, caso possível, corrigidos durante o monitoramento do progresso.
- *Encerramento do projeto*, etapa em que se finaliza e se entrega o que foi planejado para o projeto em questão. Essa etapa também serve como conclusão do ciclo do projeto com o time, fazendo uma análise geral sobre os aspectos positivos, negativos, assim como eventuais aspectos de melhorias para futuros projetos.

2.2 Ferramentas de Gestão de Projetos

De acordo com Alomar et al. (2016), ferramentas de gestão de projetos são instrumentos que auxiliam, tanto o gestor do projeto quanto o time de desenvolvimento, a acompanhar, planificar e organizar o projeto. Existem vários tipos de ferramentas de gestão como *JIRA*, *Microsoft Project*, *Trello*, *@Task*, *Microsoft Azure*, entre outras. Essas ferramentas podem auxiliar e influenciar o projeto de diferentes formas.

Durante a definição do problema e a discussão sobre a solução, as ferramentas podem auxiliar na documentação do projeto, desde a informação extraída do cliente para a construção do projeto, até a documentação em relação a solução definida para a resolução do problema.

Nas etapas de planejamento e execução, a ferramenta pode auxiliar na definição de tarefas que precisam ser feitas, na divisão dessas tarefas entre o time de desenvolvimento e no acompanhamento do status de cada uma. Essa visão possibilita que o gestor entenda a situação do projeto em relação aos prazos definidos, fazendo com que a etapa de monitoramento e controle do projeto também seja auxiliada pela ferramenta.

Por fim, no encerramento do projeto, a ferramenta que passou por todas as etapas do projeto, possui histórico de tudo que foi feito desde o início do projeto. Esse histórico pode auxiliar na cerimônia de encerramento, onde se pode acompanhar o andamento do projeto desde que foi iniciado, levantar informações sobre quais tarefas foram mais complexas ou tomaram mais tempo, quais os problemas que o planejamento do projeto não levou em consideração e o que levou a possíveis atrasos e aumento de custo no projeto.

2.3 Processo de Software

De acordo Pressman (2005), processo de *software* é uma metodologia para atividades, tarefas e ações necessárias para o desenvolvimento de um software com qualidade. Conforme Sommerville (2010), processo de *software* é um grupo de atividades que conduzem o processo de produção e desenvolvimento de um produto de *software*. Essa produção pode ser de um novo *software*, ou de produtos que já existem mas precisam passar por modificações ou evoluções. Existe mais de um tipo de processo de *software*, mas todos os processos incluem quatro atividades fundamentais (SOMMERVILLE, 2010):

- *Especificação de software*, no qual as funcionalidades e restrições do funcionamento do produto são definidas, descrita na Seção 2.3.1;
- *Projeto e implementação de software*, produção do produto atendendo as especificações, descrita na Seção 2.3.2;
- *Validação de software*, validação do projeto para garantir que o produto atende às demandas do cliente, descrita na Seção 2.3.3;
- *Evolução de software*, evolução do produto para que atenda às necessidades de mudança do cliente, descrita na Seção 2.3.4.

Processos de *software* são complexos, dependem de pessoas para tomar decisões. Eles evoluem na tentativa de obter a melhor capacidade das pessoas e organizações envolvidas no desenvolvimento de um sistema. Para sistemas críticos, por exemplo, em que uma falha pode levar a perdas significativas de dinheiro, ameaças ao ser humano ou danos físicos, o processo precisa ser muito bem estruturado (DROBKA; NOFTZ; RAGHU, 2004).

2.3.1 Especificação de Software

A atividade de especificação de *software*, ou engenharia de requisitos, representa a etapa de levantamento de requisitos do sistema, em que se compreende e identifica restrições relacionadas operação e desenvolvimento do sistema. Esta etapa é crítica, pois erros causados no levantamento de requisitos podem ser propagados para as outras atividades do processo de *software*, gerando problemas na implementação e validação do sistema (SOMMERVILLE, 2010).

Esta atividade tem como objetivo a construção da documentação de requisitos do sistema. Estes requisitos são especificações acordadas com os responsáveis pelo projeto, especificações que satisfazem as necessidades do cliente.

2.3.2 Projeto e implementação de Software

A atividade de projeto e implementação de *software* é o estágio de desenvolvimento do sistema descrito. É o estágio em que a descrição dos requisitos definidos na atividade de especificação são convertidos na implementação do sistema de fato, em um produto funcional (SOMMERVILLE, 2010).

O projeto de *software* é a documentação da estrutura do *software*, ou seja, descrição dos modelos, estruturas, interfaces e componentes necessários para a implementação do *software*. Essa documentação pode ser feita através de diagramas de classes, e ocorre durante todo o desenvolvimento do sistema, que é feito de forma iterativa, passando por revisões constantes e correções. O desenvolvimento do projeto durante essa atividade é a programação do sistema, onde todos os itens descritos na documentação do projeto de *software* se tornam o produto funcional.

2.3.3 Validação de Software

A atividade de validação de *software*, também conhecida como verificação e validação, tem como objetivo a execução de testes em cima do sistema desenvolvido. Esses testes servem para validar que o sistema satisfaz todas as especificações levantadas pelo cliente na atividade de especificação do *software* (SOMMERVILLE, 2010).

A validação e verificação do sistema desenvolvido não ocorre apenas no final do projeto, após o desenvolvimento de todo o sistema, ocorre durante todo o processo de desenvolvimento. Como o desenvolvimento é um processo iterativo, a validação também, testando toda entrega de pequenas porções funcionais do sistema, assim como a validação da integração dessas novas entregas com o que já foi desenvolvido em iterações anteriores.

2.3.4 Evolução de Software

A atividade de evolução do *software* representa a evolução e manutenção de um sistema. Mudanças em *software* são consideradas alterações muito mais baratas do que alterações em *hardware*, por exemplo. Essas podem ser feitas a qualquer momento, durante ou após a finalização do desenvolvimento (SOMMERVILLE, 2010).

A evolução e manutenção de *software* deve ser vista como um processo contínuo, junto com o desenvolvimento do projeto. Por se tratar de um processo evolutivo, as especificações e requisitos do cliente podem ser alterados, prolongando o período de vida do produto, que segue se adaptando às necessidades do cliente.

Evoluir o *software* envolve também a resolução de débitos técnicos do projeto. Débito técnico ou dívida técnica, de acordo com Brown et al. (2010), representa a dívida que um projeto de *software* assume quando toma a decisão de priorizar a entrega, ao invés da qualidade. Isso acontece quando requisitos complexos do projeto de *software* são solucionados de maneira inadequada, exigindo que a solução seja revisitada no futuro para que o projeto possa evoluir da maneira correta.

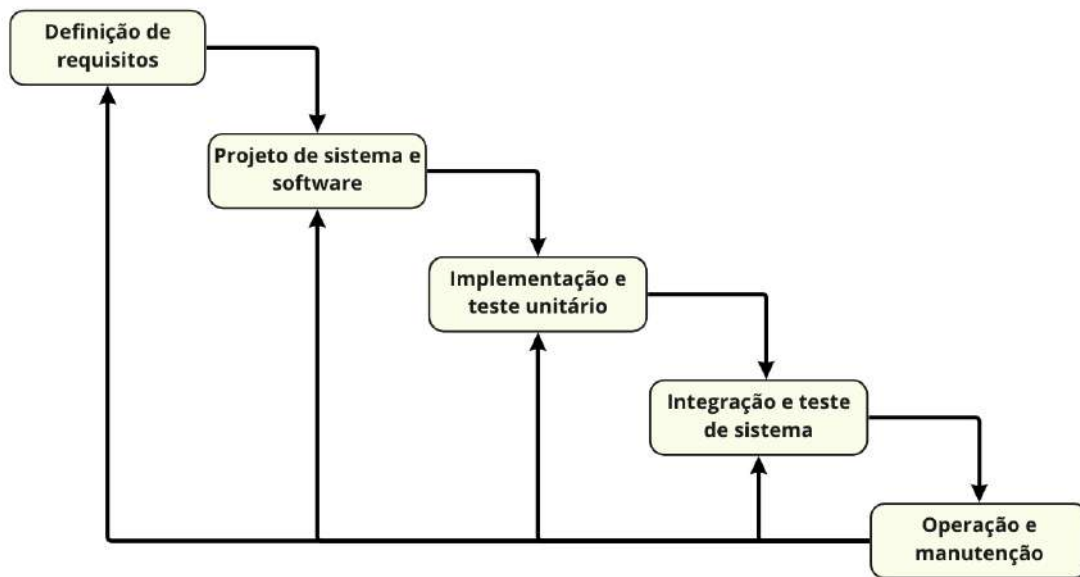
2.3.5 Modelo de Processo de Software

Os processos de *software*, de acordo com Sommerville (2010), são representados como modelos, e cada modelo apresenta suas próprias características. Esses modelos são estruturas que podem dividir as atividades, que representam uma visão macro do processo de software. Pensa-se que cada atividade complexa pode ser dividida em tarefas de menor complexidade, expandindo o processo de *software*.

Um exemplo de modelo de processo de *software* é o modelo em cascata. No modelo em cascata, as atividades são separadas em estágios que seguem uma ordem restrita, formando assim a cascata, como mostra a Figura 2.1. No modelo cascata, os principais estágios do desenvolvimento são representados da seguinte forma:

- *Análise e definição de requisitos*, estágio de estabelecimento das restrições e metas do sistema a ser desenvolvido. Nesse estágio são definidas as especificações do sistema a partir do levantamento de requisitos.

Figura 2.1: Estágios do modelo de processo de *software* em cascata



Fonte: Reproduzido de Sommerville (2010)

- *Projeto de sistema e software*, estágio em que os requisitos são analisados para alocar as necessidades do sistema em relação à tecnologias que serão utilizadas no desenvolvimento. Essas necessidades também envolvem a definição da arquitetura do sistema.
- *Implementação e teste unitário*, estágio em que o projeto é desenvolvido e, em conjunto com o desenvolvimento, são feitos testes unitários. O teste unitário nesse estágio serve para a verificação de que cada unidade entregue corretamente as necessidades levantadas no estágio de definição do sistema.
- *Integração e teste de sistema*, estágio em que as unidades do sistema desenvolvidas, de forma iterativa na etapa de implementação, são integradas com o sistema por completo. Os testes de sistema nessa etapa servem para verificar se todas as unidades, quando utilizadas em conjunto como um sistema completo, satisfazem os requisitos do estágio de definição, e após essa validação, o sistema pode ser entregue ao cliente.
- *Operação e manutenção do sistema*, etapa em que o sistema entra em uso pelo cliente final. Nessa etapa, a manutenção do sistema ocorre conforme erros, que não foram encontrados durante os testes, são encontrados. Esses erros são corrigidos e melhorias são levantadas conforme as novas necessidades do cliente, descobertas a partir do uso do sistema.

2.4 Usabilidade e Utilidade

De acordo com Barbosa and Silva (2010), um sistema interativo possui uma interface, e utilizar um sistema interativo é realizar ações a partir de sua interface para alcançar um objetivo. Entre os critérios de qualidade que envolvem interação e interface está a usabilidade.

Conforme Barbosa and Silva (2010), usabilidade se refere a maneira como o uso de um sistema interativo pode ser afetado pelas características do usuário. Essas características envolvem a cognição do usuário, sua capacidade de entender as respostas do sistema e realizar ações na interface do sistema.

Enquanto a usabilidade reforça a preocupação em criar sistemas que sejam fáceis de usar, a utilidade representa a preocupação com as funcionalidades do sistema. Um sistema é útil quando suas funcionalidades atendem às necessidades do usuário, no contexto em que é utilizado.

2.5 Desenvolvimento Ágil de Software

Conforme Sommerville (2010), o desenvolvimento ágil de software é uma resposta à necessidade de produzir aplicações de forma rápida, sem que o cliente aguarde até a finalização do projeto ter algo funcional disponível. Com entregas menores e menos complexas, esses métodos auxiliam o gestor e o time de desenvolvimento a atingir pequenos objetivos que contribuem para a finalização do projeto. Métodos ágeis definem princípios e, apesar de existirem diversos métodos ágeis, os seguintes princípios devem ser seguidos:

- Envolver o cliente no processo de desenvolvimento do software, ajudando a priorizar as próximas entregas e a refinar os requisitos do sistema.
- Realizar entregas do sistema de forma incremental, onde o sistema é entregue ao cliente com as novas funcionalidades, que foram priorizadas em iterações anteriores.
- Levar em consideração o contexto pessoal da equipe de desenvolvimento, onde os membros encontram sua forma de trabalhar.
- Levar em consideração que os requisitos do sistema irão sofrer alterações, portanto o sistema precisa estar preparado para essas mudanças.

- Manter a simplicidade no processo e no desenvolvimento do sistema, trabalhando de forma que seja possível diminuir a complexidade do sistema.

Extreme Programming (XP) é o mais utilizado dos métodos ágeis. Em XP, os requisitos são descritos como estórias de usuário (US), que representam uma série de tarefas necessárias para a conclusão de funcionalidades do *software* (BECK, 1999). As práticas de XP respeitam os princípios dos métodos ágeis e trabalham de forma com que cada princípio seja levado em consideração nas iterações do processo:

- O envolvimento do cliente é atingido por conta do engajamento com a equipe de desenvolvimento, onde existe um representante do cliente que participa da construção do sistema, levantando as definições de aceitação para o sistema.
- O desenvolvimento incremental é atingido com cada *release* feita do sistema, que contém o conjunto de funcionalidades que foram descritas nas estórias de usuário priorizadas em cada iteração.
- O contexto pessoal da equipe é levado em consideração a partir da programação em pares, onde existe uma distribuição da responsabilidade pelo código escrito e proporciona um desenvolvimento sustentável do sistema.
- As possíveis mudanças do sistema são esperadas levando em consideração que a cada *release* do sistema, existe uma nova iteração que permite com que o cliente reflita sobre o que foi desenvolvido e solicite mudanças, que serão priorizadas na próxima entrega.
- A simplicidade no processo é atingida também com as entregas contínuas, o que permite que o projeto sempre esteja atualizado e refatorado para receber novas funcionalidades.

2.6 Trabalhos Relacionados

Diversos trabalhos envolvendo análise de ferramentas de gestão de projetos de desenvolvimento foram encontrados, com focos em diferentes aspectos. Dentre esses trabalhos, Chadli et al. (2016) e Mishra and Mishra (2013) são os que mais se aproximam do tema, com a comparação e classificação de ferramentas existentes.

Em Chadli et al. (2016), os autores realizam um mapeamento sistemático sobre ferramentas de gestão com foco no desenvolvimento global de *software*. Nesse estudo, os resultados da pesquisa são apresentados, identificando quais ferramentas estão sendo

utilizadas no desenvolvimento global de *software* e classificando em qual etapa do ciclo de desenvolvimento elas se encontram, além de apresentar como essas ferramentas suportam o modelo de colaboração 3C (comunicação, coordenação e cooperação). O estudo mostrou que das 102 ferramentas encontradas, 77% são ferramentas *standalone*, ou seja, não precisam de outras ferramentas para serem utilizadas. Já 8% das 102 ferramentas encontradas oferece uma série de outras ferramentas interativas com a ferramenta principal, para que possa suportar mais etapas do ciclo de vida do desenvolvimento de software. Os resultados encontrados indicam que ainda existem áreas no desenvolvimento global de *software* que não são cobertas pelas ferramentas disponíveis.

No trabalho de Mishra and Mishra (2013) é apresentada uma breve visão comparativa entre as ferramentas de gestão de projetos mais populares segundo os autores. Essa comparação leva em consideração aspectos como, por exemplo, se é uma plataforma local ou *web*, se possui rastreamento de tempo, descrição das tarefas, geração de relatórios, definição dos papéis dos usuários da ferramenta, entre outros. O artigo conclui que, entre as ferramentas estudadas, as ferramentas proprietárias *@Task*, *Clarizen* e *Gemini* (com suporte de *@Task*) são as mais completas, suportando 12 das 13 características levantadas pelos autores. No caso das ferramentas de código aberto, as mais completas são *GanttProject* e *OpenProj*.

2.7 Resumo do Capítulo

Neste capítulo foram apresentados conceitos necessários para a compreensão deste trabalho e também, trabalhos relacionados com o estudo apresentado. Foi definido o que é gestão de projetos e seus estágios, e como as ferramentas de gestão de projetos podem auxiliar durante todo o ciclo de vida de um projeto.

O capítulo apresentou também as definições de processo de *software* e suas atividades e como os processos podem ser classificados em modelos. As definições de mapeamento sistemático de literatura e os princípios dos métodos ágeis também foram apresentados neste capítulo.

Por fim, o capítulo apresentou os trabalhos de Chadli et al. (2016), com o mapeamento sistemático de ferramentas de gestão com foco em desenvolvimento global de *software* e Mishra and Mishra (2013), com uma breve visão comparativa entre 20 ferramentas de gestão de projetos.

3 METODOLOGIA DE PESQUISA

Este capítulo apresenta a metodologia utilizada na pesquisa, bem como a seleção e classificação dos artigos durante o processo do mapeamento sistemático. Diferente das revisões sistemáticas que visam levantar e sintetizar evidências, os estudos de mapeamento sistemáticos são produzidos a partir de classificação por categoria e contagem das contribuições a partir de cada categoria descrita. É uma pesquisa na literatura para entender quais tópicos de um determinado assunto estão sendo cobertos e onde esses estudos estão sendo publicados, para entender a relevância dessas contribuições (PETERSEN; VAKKALANKA; KUZNIARZ, 2015). Os critérios de seleção e classificação dos artigos encontrados na literatura são definidos neste trabalho de forma detalhada para que os estudos possam ser reproduzidos.

3.1 Perguntas de Pesquisa

De acordo com (PETERSEN; VAKKALANKA; KUZNIARZ, 2015), as perguntas de pesquisa devem se referir ao objetivo da pesquisa. Neste trabalho, o objetivo é identificar e analisar o que existe na literatura em relação a ferramentas de gestão de projetos de desenvolvimento de *software*, bem como encontrar critérios de usabilidade e utilidade que envolvem essas ferramentas e o processo de desenvolvimento de *software*. A partir desse objetivo, o trabalho define as seguintes perguntas de pesquisa:

- P01: O que existe na literatura sobre ferramentas de gestão de projetos de desenvolvimento de *software* nos últimos onze anos?
- P02: O que existe na literatura sobre *usabilidade* em relação a ferramentas de gestão de projetos de desenvolvimento de *software*?
- P03: O que existe na literatura sobre *utilidade* em relação a ferramentas de gestão de projetos de desenvolvimento de *software*?

A informação extraída na pesquisa combinada à classificação dos artigos encontrados ajudará a responder essas perguntas e concluir o estado da arte sobre o tema investigado neste trabalho.

3.2 Pesquisa e Seleção

A partir das perguntas de pesquisa levantadas na Seção 3.1, a *string* de busca é definida. Os termos da *string*, são definidos em inglês, para que a busca possa abranger o maior número de trabalhos no acervo dos repositórios pesquisados.

A partir da pergunta P01, *O que existe na literatura sobre ferramentas de gestão de projetos de desenvolvimento de software nos últimos onze anos?*, é possível extrair *ferramentas de gestão de projetos de desenvolvimento de software*, ou *software development project management tools*, o tema principal pesquisa. Ao observar as perguntas de pesquisa P02, *O que existe na literatura sobre usabilidade em relação à ferramentas de gestão de projetos de desenvolvimento de software?* e P03, *O que existe na literatura sobre utilidade em relação a ferramentas de gestão de projetos de desenvolvimento de software?*, é possível extrair os termos *utilidade* e *usabilidade*, ou seja, *utility* e *usability*.

Com essas informações extraídas das perguntas de pesquisa, a construção da *string* de busca se dá por "*software development*" AND (*project* OR *process*) AND "*management*" AND (*tool* OR "*software*") AND ("*usability*" OR "*utility*"), levando em consideração as variações dos termos nos artigos encontrados. A partir da *string* de busca considerada neste trabalho, é iniciado o processo de busca por artigos nos três repositórios escolhidos. São eles:

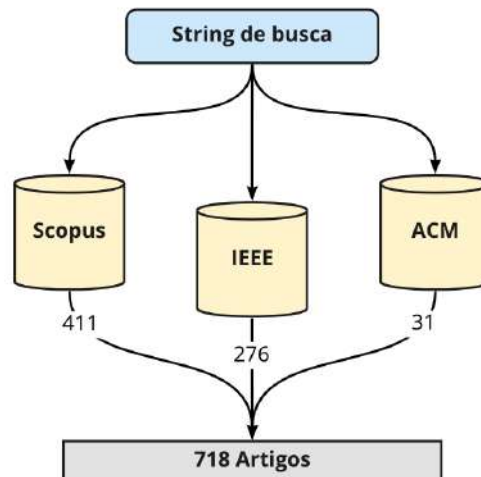
- Scopus: Banco de dados de resumos e citações organizados por especialistas (ELSEVIER, 2022).
- ACM DL: *Association for Computing Machinery Digital Library* (ACM, 2022).
- IEEE Xplore: *The Institute of Electrical and Electronics Engineers Digital Library* (IEEE, 2022).

Após a pesquisa por trabalhos, aplicando *string* de pesquisa em cada um dos repositórios citados, 718 artigos foram encontrados. Desses artigos, 411 são encontrados na Scopus, 276 na IEEE e 31 na ACM, como apresenta a Figura 3.1.

3.2.1 Critérios de Exclusão

A partir da seleção dos artigos, critérios de exclusão são aplicados. Esses critérios são utilizados para validação da data de publicação dos artigos, área de publicação e outros aspectos. Para este trabalho, foram definidos os seguintes critérios:

Figura 3.1: String de pesquisa aplicada em cada repositório



Fonte: O Autor

CE01: Artigos publicados antes de 2011 e após 2021.

CE02: Artigos que não pertencem à área de ciência da computação.

CE03: Artigos que não foram escritos em inglês.

CE04: Artigos com menos de quatro páginas.

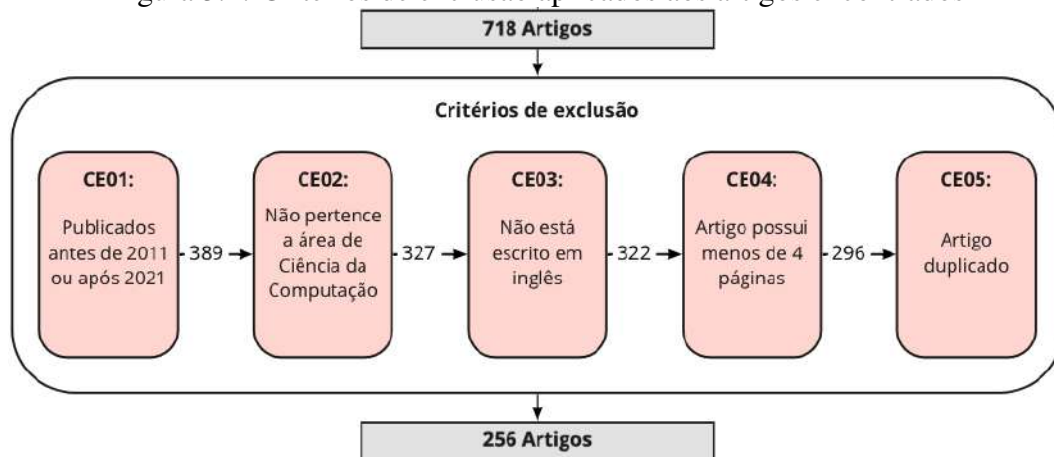
CE05: Artigos duplicados.

CE01 serve como limitador da data dos artigos encontrados, pois este trabalho tem como objetivo estudar apenas o período entre 2011 e 2021, ou seja, os últimos 11 anos, e com essa exclusão, 389 artigos são mantidos. CE02 e CE03 tem como objetivo excluir os artigos que não pertencem a área de ciência da computação e artigos que não foram escritos em inglês, e com a aplicação desses dois critérios, 322 artigos continuam na seleção. A partir da aplicação do CE04, que exclui os artigos que possuem menos de quatro páginas, 296 artigos seguem na seleção. Por fim, é aplicado o CE05, que exclui os artigos duplicados, e com isso 256 artigos seguem para a próxima etapa, como apresentado na Figura 3.2.

3.2.2 Critérios de Inclusão

Após a aplicação dos critérios de exclusão nos artigos selecionados, a validação do conteúdo dos trabalhos é feita para validar quais artigos estão dentro do tema e devem fazer parte da classificação proposta. Para a validação do conteúdo dos artigos, foram levantados os seguintes critérios de inclusão:

Figura 3.2: Critérios de exclusão aplicados aos artigos encontrados



Fonte: O Autor

CI01: Artigo possui título e palavras-chave dentro do tema da pesquisa.

CI02: Artigo possui *abstract* dentro do tema de pesquisa.

CI03: Artigo possui conclusão dentro do tema de pesquisa, em artigos onde apenas título, palavra-chave e *abstract* não foram o suficiente.

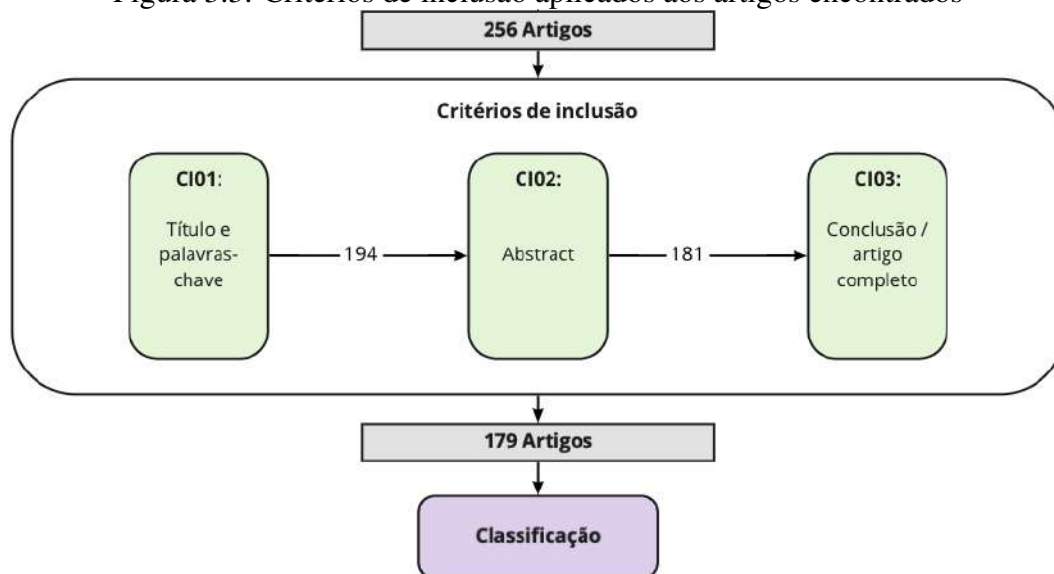
São considerados fora do tema os artigos que não possuem relação com o tema ou que, apesar de discutirem temas envolvendo o processo de desenvolvimento de *software*, não foram no processo em si. São artigos que não falam sobre metodologias, métricas ou melhorias no processo de *software*, e sim focam no *software* desenvolvido. Para que artigos, focados especificamente no produto gerado pelo trabalho, pudessem ser incluídos, esse produto deveria ser alguma ferramenta que melhora, facilita ou tem relação com o processo de desenvolvimento de *software*.

Após a aplicação da validação de conteúdo com os critérios de inclusão, o trabalho chega a 179 artigos selecionados, como mostra a Figura 3.3, finalizando o processo de seleção de artigos para classificação.

3.3 Classificação e Visualização dos Dados

Os artigos selecionados passam por três tipos de classificação definidas neste trabalho. Na classificação dos artigos por ano, todos os artigos são separados por ano de publicação, para que se possa ter uma visão geral de publicações por ano, como mostra a Figura 3.4. Após a classificação dos artigos por ano, é extraído de cada ano os tópicos mais encontrados, ou seja, as palavras-chave mais encontradas nos artigos selecionados.

Figura 3.3: Critérios de inclusão aplicados aos artigos encontrados



Fonte: O Autor

A partir dessas informações, é feito um resumo das contribuições feitas pelos trabalhos encontrados nos tópicos mais encontrados em cada ano.

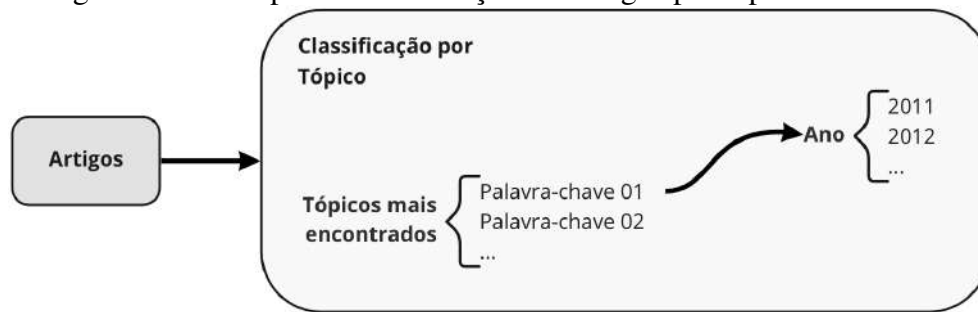
Figura 3.4: Exemplo de classificação dos artigos por ano de publicação



Fonte: O Autor

O segundo tipo de classificação apresentado é a classificação de artigo por tópico. Os tópicos neste trabalho são representados pelas palavras-chave encontradas nos trabalhos selecionados. Nessa classificação, os dez tópicos mais encontrados nos artigos são apresentados e, a partir desses tópicos, é feita a linha do tempo de cada tópico, como apresentado na Figura 3.5. A linha do tempo do tópico é a apresentação de quantos artigos sobre o tópico foram publicados por ano, dentro dos artigos selecionados. A partir dessa informação, é feita uma análise quantitativa em relação ao número de publicações dos dez tópicos mais encontrados nos trabalhos selecionados, observando tendências e relevância dos tópicos, considerando este mapeamento sistemático.

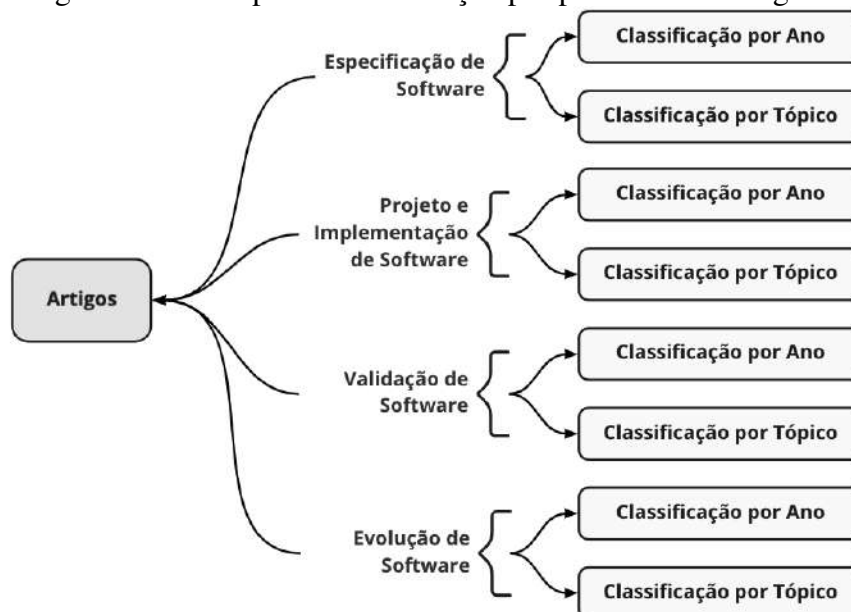
Figura 3.5: Exemplo de classificação dos artigos por tópico encontrado



Fonte: O Autor

O terceiro e último tipo de classificação dos artigos é pelas atividades do processo de *software*, apresentadas no Capítulo 2. Os artigos são classificados manualmente, a partir do conteúdo, em relação à atividade do processo de *software* a qual se referem. Após a classificação, para cada atividade são feitos dois tipos de classificação, como apresenta a Figura 3.6. Na classificação por tópico, é feito um resumo das contribuições dos artigos que possuem os tópicos mais encontrados em cada atividade. Na classificação por ano, é feita uma análise quantitativa em relação ao número de publicações por ano da atividade em questão, analisando tendência e popularidade da atividade considerando este mapeamento sistemático e um resumo das contribuições sobre a atividade no ano com mais publicações.

Figura 3.6: Exemplo de classificação por processo dos artigos



Fonte: O Autor

3.4 Resumo do Capítulo

Neste capítulo foi apresentada a metodologia utilizada neste trabalho. Foi apresentado como foi feito o mapeamento sistemático, a pesquisa nos repositórios da *Scopus*, IEE e ACM, e quais os critérios de exclusão e inclusão foram definidos.

O capítulo apresentou também os três tipos de classificação utilizados para a apresentação dos dados encontrados durante a pesquisa. São eles, classificação por ano de publicação, classificação por tópico (palavras-chave mais encontradas nos artigos) e por atividade do processo de *software*, onde são divididas por ano de publicação e tópico.

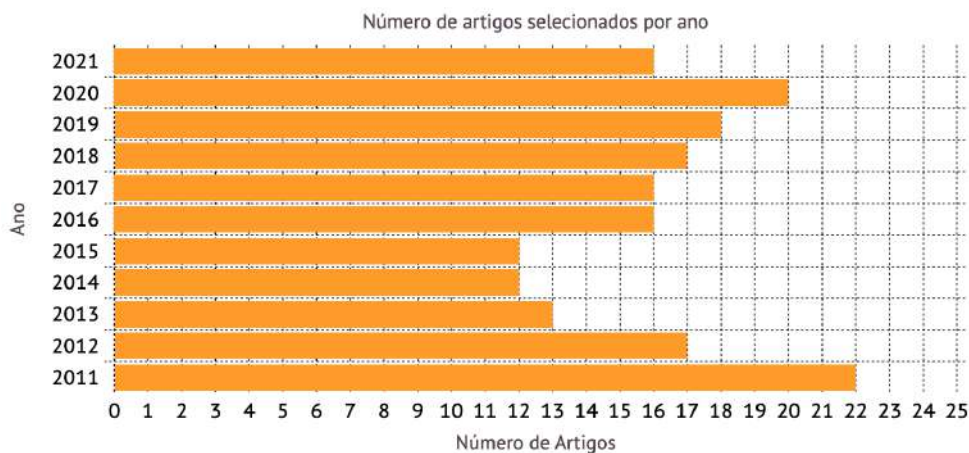
4 RESULTADOS

Este capítulo apresenta os resultados da classificação dos 179 artigos selecionados, após busca e passagem por critérios de exclusão e inclusão, definidos no Capítulo 3. A classificação dos artigos é dividida por ano, por tópico e por processo de software. Todas as informações extraídas no processo de classificação apresentado, fazem parte da resposta da pergunta de pesquisa P01 (*O que existe na literatura sobre ferramentas de gestão de projetos de desenvolvimento de software nos últimos onze anos?*).

4.1 Classificação Por Ano

Na classificação dos artigos por ano de publicação, apresentado na Figura 4.1, observa-se que 2011 foi o ano com mais publicações de artigos sobre ferramentas de gestão de projetos, representando 12,29% dos trabalhos selecionados.

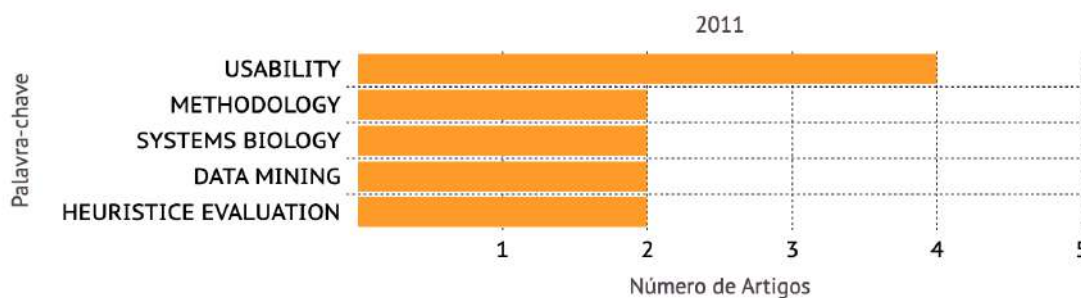
Figura 4.1: Artigos por ano



Fonte: O Autor

No ano de 2011, as palavras-chave mais encontradas nos artigos foram "*USABILITY*", "*METHODOLOGY*", "*SYSTEMS BIOLOGY*", "*DATA MINING*" e "*HEURISTICE EVALUATION*", exposto na Figura 4.2, com todas as palavras-chave que foram citadas mais de uma vez. Os artigos que citam "*USABILITY*" são Conger (2011), Juárez-Ramírez et al. (2011), Lanna and Amyot (2011) e Spencer, Zimmerman and Abramson (2011), e auxiliam na resposta da pergunta de pesquisa P02 (*O que existe na literatura sobre usabilidade em relação a ferramentas de gestão de projetos de desenvolvimento de software?*).

Figura 4.2: Tópicos mais encontrados nos artigos de 2011



Fonte: O Autor

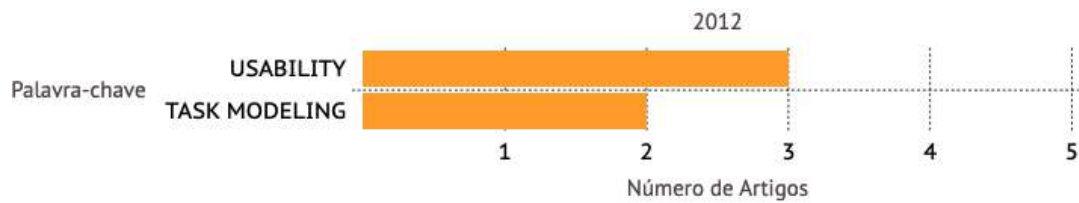
No artigo de Conger (2011) o foco é justamente a melhoria do processo de desenvolvimento de software, avaliando o estado do desenvolvimento de aplicações naquele momento (em 2011). O trabalho também recomenda melhorias para o processo de desenvolvimento de novas aplicações que corrigem erros históricos, como atenção ao gerenciamento de riscos, testes e detalhamento de práticas de trabalho no desenvolvimento de *software*.

Em Juárez-Ramírez et al. (2011) é indicado como o processo de desenvolvimento de aplicações móveis pode ser melhorado a partir de práticas de melhoria da usabilidade da interface dessas aplicações. O trabalho de Lanna and Amyot (2011) discute usabilidade a partir de *Source Code Management* (SCM), uma ferramenta que permite o versionamento de arquivos do projeto, onde são apresentadas novas funcionalidades para melhoria da ferramenta. Finalmente, em Spencer, Zimmerman and Abramson (2011) são apresentados os desafios e oportunidades na gestão de projetos em *e-science*.

No ano de 2012, com 17 artigos selecionados, a palavra-chave mais citada também é "*USABILITY*", seguida por "*TASK MODELING*" que apareceu duas vezes, como mostra a Figura 4.3 com as duas palavras-chave que apareceram mais de uma vez. "*USABILITY*" foi encontrada nos artigos McCalden (2012), Al-Qaimari and Agi (2012) e Pierce (2012), auxiliando na resposta da pergunta de pesquisa P02 (*O que existe na literatura sobre usabilidade em relação à ferramentas de gestão de projetos de desenvolvimento de software?*).

Em Al-Qaimari and Agi (2012) é apresentada uma proposta de processo para auxiliar organizações na melhoria da usabilidade no desenvolvimento dos seus sistemas a partir de design centrado no usuário. O trabalho de McCalden (2012) discute usabilidade na concepção de ferramentas espaciais, para diminuir o risco durante o desenvolvimento de *software*. E em Pierce (2012) a discussão é centrada em sistemas complexos, ou sistemas de sistemas, e como melhorar a usabilidade desses sistemas.

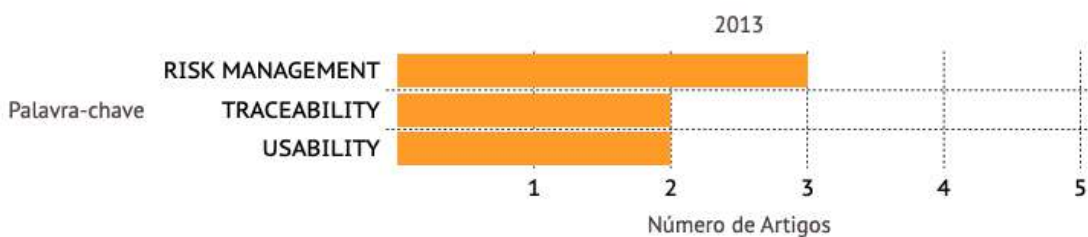
Figura 4.3: Tópicos mais encontrados nos artigos de 2012



Fonte: O Autor

Em 2013, com 13 artigos selecionados, três desses artigos trazem a palavra-chave "*RISK MANAGEMENT*", seguido por duas menções à "*TRACEABILITY*" e "*USABILITY*", representado na Figura 4.4. Os artigos que citam "*RISK MANAGEMENT*" são Tényi et al. (2013), Lindholm and Host (2013) e Zhou, Gifford and Ratakonda (2013).

Figura 4.4: Tópicos mais encontrados nos artigos de 2013



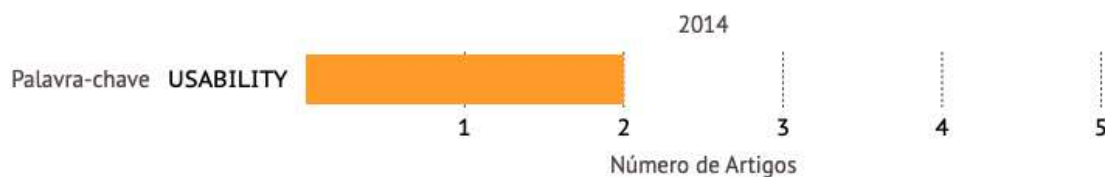
Fonte: O Autor

No trabalho de Tényi et al. (2013), o processo de gestão de riscos é focado no processo regulatório do ambiente médico, na criação de um processo em que *hardware*, *software* e riscos no uso dos equipamentos sejam integrados para validação de riscos durante todo o processo de desenvolvimento das aplicações. Em Lindholm and Host (2013), testes de usabilidade são adicionados ao processo de gestão de riscos de aplicações médicas. Zhou, Gifford and Ratakonda (2013) discute a relação entre risco e custo de projeto de software, apresentando um método para ajustar o custo de projeto, levando em consideração os riscos do lado do cliente e escolhas que podem ser feitas durante o orçamento do projeto para diminuir os riscos de desenvolvimento.

O ano de 2014 possui 12 artigos selecionados. A palavra-chave mais encontrada nos artigos é, como observa-se na Figura 4.5, "*USABILITY*", encontrada em Moorthy, Ibrahim and Mahrin (2014) e Tabassum et al. (2014), auxiliando na resposta da pergunta de pesquisa P02 (*O que existe na literatura sobre usabilidade em relação à ferramentas de gestão de projetos de desenvolvimento de software?*).

O trabalho de Moorthy, Ibrahim and Mahrin (2014) apresenta uma pesquisa sobre

Figura 4.5: Tópicos mais encontrados nos artigos de 2014

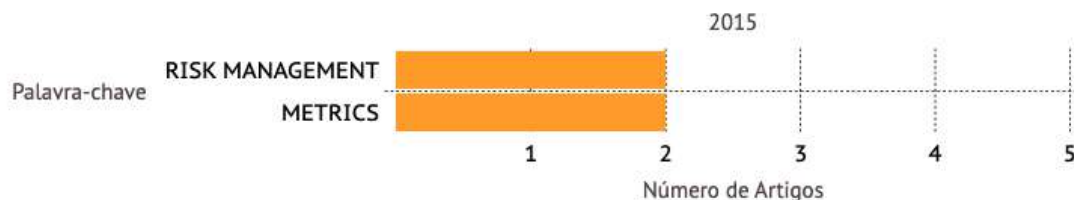


Fonte: O Autor

identificação de riscos de usabilidade, fazendo gestão de riscos desde o início do desenvolvimento de um *software*. Em Tabassum et al. (2014) o foco é a dependência entre requisitos não funcionais do projeto, determinando interdependência entre esses requisitos para evitar problemas de conflito.

No ano de 2015, os 12 artigos selecionados apresentam duas palavras-chave encontradas mais de uma vez. "*RISK MANAGEMENT*" e "*METRICS*", são encontradas duas vezes, como mostra a Figura 4.6. Wanderley et al. (2015) apresenta as duas palavras-chave, enquanto Jha and Patnaik (2015) e Lindholm (2015) apresentam "*METRICS*" e "*RISK MANAGEMENT*" respectivamente.

Figura 4.6: Tópicos mais encontrados nos artigos de 2015



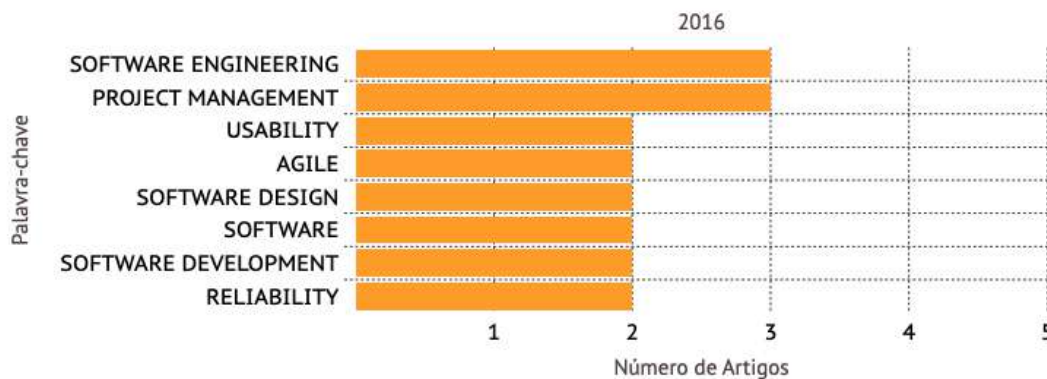
Fonte: O Autor

Em Wanderley et al. (2015) o foco é o desenvolvimento de múltiplos projetos de *software* ao mesmo tempo, necessitando de gestão e métricas de riscos para validação constante do escopo do produto e outros aspectos. A discussão em Jha and Patnaik (2015) foca em modelos de estimação de *bugs*, baseado no conjunto de métricas de *Chidamber & Kemerer* (CK), para *softwares* desenvolvidos com a abordagem orientada a objetos. Finalmente, Lindholm (2015) traz o uso da perspectiva dos usuários testando e utilizando dispositivos médicos para ajudar na gestão de riscos.

No ano de 2016, 16 artigos estão entre os selecionados. Oito palavras-chave são encontradas mais de uma vez, como apresentado na Figura 4.7. Dessas oito palavras, "*USABILITY*", "*SOFTWARE DESIGN*", "*AGILE*", "*SOFTWARE DEVELOPMENT*", "*SOFTWARE*", "*SOFTWARE DEVELOPMENT*" e "*RELIABILITY*" são encontradas duas vezes.

"SOFTWARE ENGINEERING" é encontrada três vezes, nos trabalhos de Alomar et al. (2016), Heitzenrater and Simpson (2016) e Z, Villavicencio and Izquierdo (2016). "PROJECT MANAGEMENT" é também encontrada três vezes, em Oliveira et al. (2016), Porubän and Baciková (2016) e Crisan, Gardy and Munzner (2016).

Figura 4.7: Tópicos mais encontrados nos artigos de 2016



Fonte: O Autor

Em Alomar et al. (2016) é feita uma pesquisa para avaliar quatro ferramentas de gestão de projetos (*JIRA*, *AgileZen*, *VersionOne* e *ZebraPlan*.) com base em critérios de avaliação de usabilidade. Heitzenrater and Simpson (2016) falam sobre validação de segurança para casos de uso indevido ou abuso da tecnologia em relação a ameaças reais aos sistemas, propondo funções de utilidade econômica dentro do processo de desenvolvimento desses casos para abordar os desafios existentes. No trabalho de Z, Villavicencio and Izquierdo (2016), é apresentado um *framework* para avaliação de ferramentas de prototipagem a baixo custo.

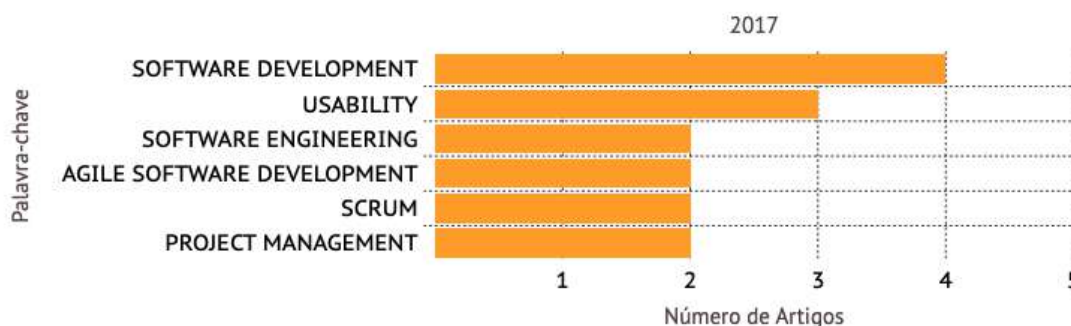
Em relação à *PROJECT MANAGEMENT*, ou gestão de projetos, Oliveira et al. (2016) apresenta uma pesquisa sobre os impactos do uso de metodologias ágeis no desenvolvimento de *software* e na adoção do Scrum em um cenário real, para entender se essa abordagem pode contribuir e ajudar na gestão de riscos dos projetos e na qualidade desses *softwares*. O trabalho de Oliveira et al. (2016) é também o único artigo, entre os artigos selecionados, que apresenta o tópico *UTILITY*, sendo o único artigo a auxiliar na resposta da pergunta de pesquisa P03 *O que existe na literatura sobre utilidade em relação a ferramentas de gestão de projetos de desenvolvimento de software?*.

Porubän and Baciková (2016) traz um estudo sobre o ensino de engenharia de *software*, apresentando uma nova abordagem para criar um ambiente real, no qual o aluno consegue ter uma visão do progresso do *software* desenvolvido, também avaliando a usabilidade dessa abordagem com um grupo de alunos. Em Crisan, Gardy and Munzner

(2016) é apresentada uma visualização de projeto, baseada em problemas, para identificar as necessidades dos usuários, oferecendo orientação baseada em técnicas de desenvolvimento de *software* para identificar e avaliar restrições externas.

O ano de 2017 possui 16 artigos selecionados. As palavras-chave encontradas mais de uma vez nesses artigos são *USABILITY*, *SOFTWARE ENGINEERING*, *AGILE SOFTWARE DEVELOPMENT*, *SCRUM*, *PROJECT MANAGEMENT* e *SOFTWARE DEVELOPMENT*, que foi encontrada quatro vezes, como mostra a Figura 4.8. *SOFTWARE DEVELOPMENT* é encontrada em Mäkiäho, Vartiainen and Poranen (2017), Smaradottir (2017), Andrade and Sadaoui (2017) e Zanatta et al. (2017).

Figura 4.8: Tópicos mais encontrados nos artigos de 2017

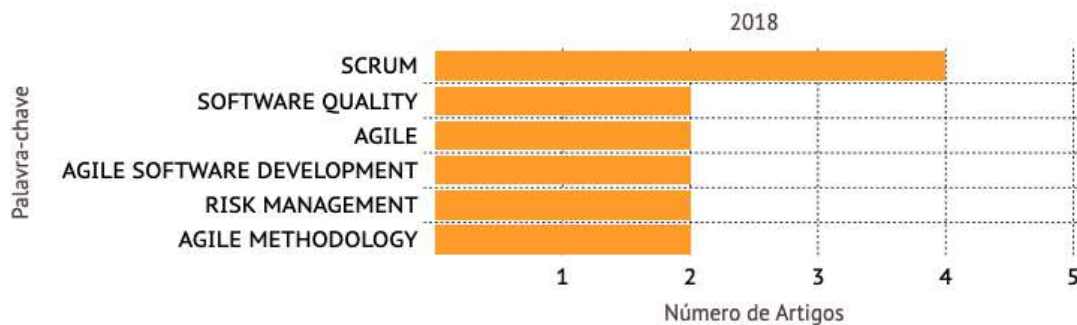


Fonte: O Autor

Em Mäkiäho, Vartiainen and Poranen (2017), o trabalho apresenta a *Metrics Monitoring Tool* (MMT), uma ferramenta desenvolvida por alunos da graduação e pós-graduação para auxiliar times responsáveis por projetos de *software* a monitorar o andamento do projeto a partir de métricas que tornam a utilização da ferramenta mais fácil e efetiva. No trabalho de Smaradottir (2017) o foco são os passos do *design* centrado no usuário em tecnologias de informação voltadas para a saúde. Andrade and Sadaoui (2017) apresentam uma nova forma de utilizar *Key Performance Indicators* (KPIs) para descentralizar a informação e abordar possíveis lacunas gerenciais existentes nas empresas. Em Zanatta et al. (2017), o foco são as boas práticas para minimizar as barreiras em *crowdsourcing* de *software*, que é a distribuição de uma tarefa para uma multidão de pessoas.

O ano de 2018 traz 17 artigos selecionados. As palavras-chave encontradas mais de uma vez são "*SOFTWARE QUALITY*", "*AGILE*", "*AGILE SOFTWARE DEVELOPMENT*", "*RISK MANAGEMENT*", "*AGILE METHODOLOGY*", e "*SCRUM*", que é encontrada quatro vezes, como mostra a Figura 4.9.

Figura 4.9: Tópicos mais encontrados nos artigos de 2018



Fonte: O Autor

"SCRUM" é encontrada nos artigos de Mousaei and Gandomani (2018), Rosenberger and Tick (2018), Kikitamara and Noviyanti (2018) e Setiadi and Premapasha (2018). Em Mousaei and Gandomani (2018) é apresentado um modelo de gestão de riscos utilizando o *framework Scrum*, para aprimorar a qualidade do produto desenvolvido. Rosenberger and Tick (2018) discute a sexta versão do *Project Management Body Of Knowledge* (PMBOK) e se ele suporta o trabalho com projetos que utilizam metodologias ágeis, como o *Scrum*, fazendo comparações entre os princípios ágeis e onde eles conflitam com o PMBOK. No trabalho de Kikitamara and Noviyanti (2018) tem o conceito de um modelo de *User Experience* (UX) na prática do *Scrum*, comparando possíveis modelos de trabalho. Em Setiadi and Premapasha (2018), a aplicação do *Scrum* é sobre o processo de desenvolvimento de aplicações que possuem transações online, na gestão de um hospital.

No ano de 2019, com 18 artigos selecionados, as palavras-chave mais encontradas são "SOFTWARE QUALITY", "SOFTWARE DEVELOPMENT", encontradas duas vezes, "USABILITY" e "SOFTWARE ENGINEERING" encontradas três vezes, como mostra a Figura 4.10. "USABILITY" é encontrada nos trabalhos de Deraman and Salman (2019), Espinoza et al. (2019), Mylly, Rajanen and Iivari (2019), complementando a resposta da pergunta de pesquisa P02 (*O que existe na literatura sobre usabilidade em relação à ferramentas de gestão de projetos de desenvolvimento de software?*), e "SOFTWARE ENGINEERING" em Sharma and Kumar (2019), Slhoub, Nembhard and Carvalho (2019) e Sarinho (2019).

O trabalho de Deraman and Salman (2019) fala sobre o gerenciamento da avaliação de usabilidade no ambiente de desenvolvimento ágil, com a adição dessa avaliação no processo ágil. Em Espinoza et al. (2019), o foco é usabilidade no desenvolvimento de uma ferramenta de gestão de projeto, reforçado por aprendizado de máquina. Mylly, Rajanen and Iivari (2019) traz heurísticas de usabilidade para o desenvolvimento de jogos.

Figura 4.10: Tópicos mais encontrados nos artigos de 2019



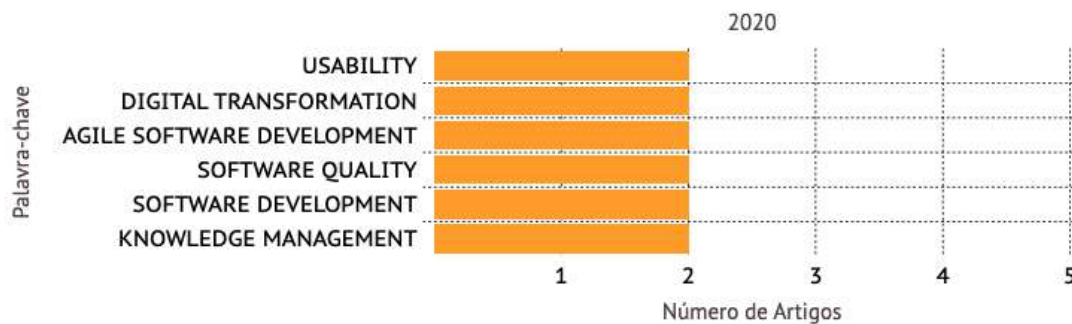
Fonte: O Autor

No tópico "*SOFTWARE ENGINEERING*", Sharma and Kumar (2019) apresenta a planificação dos lançamentos do sistema sendo desenvolvido, com a ajuda de processamento de linguagem natural. Em Slhoub, Nembhard and Carvalho (2019) é apresentado um conjunto de métricas de *software*, inspiradas no paradigma *Goal-Questions-Metrics* (GQM), que auxiliam no acompanhamento de qualidade, durante o processo de desenvolvimento de um produto. O trabalho de Sarinho (2019) discute um jogo de tabuleiro com a proposta de ensinar conceitos de engenharia de *software*, na tentativa de aplicar uma experiência mais prática.

Em 2020, 20 artigos foram selecionados. Nesses artigos, as palavras-chave "*USABILITY*", "*DIGITAL TRANSFORMATION*", "*AGILE SOFTWARE DEVELOPMENT*", "*SOFTWARE QUALITY*", "*SOFTWARE DEVELOPMENT*", "*KNOWLEDGE MANAGEMENT*" são encontradas duas vezes, como mostra a Figura 4.11. Essas palavras são encontradas nos artigos de Cunha et al. (2020), Lima et al. (2020), Ruiz and Hasselman (2020), Kalinowski et al. (2020), Rahmat and Hanifah (2020), Evans et al. (2020a), Alvear-Suárez, Vara-Mesa and Bollati (2020), Lauesen (2020), Moyo and Mnkandla (2020) Dilorenzo et al. (2020), e Arcos-Medina and Mauricio (2020), que possui "*AGILE SOFTWARE DEVELOPMENT*" e "*SOFTWARE QUALITY*".

Em "*USABILITY*", auxiliando na resposta da pergunta de pesquisa P02 (*O que existe na literatura sobre usabilidade em relação à ferramentas de gestão de projetos de desenvolvimento de software?*), o artigo de Rahmat and Hanifah (2020) fala sobre testes de usabilidade no processo ágil com *Kanban*, no ciclo de desenvolvimento de um sistema de gerenciamento para clubes. No trabalho de Evans et al. (2020a) o foco é também na parte de teste, explorando a usabilidade de ferramentas de teste e relacionando a usabilidade das ferramentas com a adoção delas no processo de validação de *software*.

Figura 4.11: Tópicos mais encontrados nos artigos de 2020



Fonte: O Autor

Em "*DIGITAL TRANSFORMATION*", o trabalho de Ruiz and Hasselman (2020) traz uma proposta de *framework* para automação dos processos de especificação de requisitos de um *software*, para que o projeto de *software* possa ser desenhado durante a discussão sobre os requisitos. Em Kalinowski et al. (2020) existe uma proposta de transformação digital a partir pesquisa e desenvolvimento ágil e a apresentação dos resultados dessa proposta aplicadas em uma empresa.

Em "*AGILE SOFTWARE DEVELOPMENT*", Arcos-Medina and Mauricio (2020) discute a influência da aplicação de práticas ágeis na qualidade do *software*, com base em oito características de qualidade de acordo com a norma ISO/IEC 25010. No trabalho de Dilorenzo et al. (2020) é proposta uma taxonomia para criar relações semânticas entre histórias de usuário e facilitar o reuso do que foi desenvolvido nessas USs.

No tópico "*SOFTWARE QUALITY*", além do artigo de Arcos-Medina and Mauricio (2020) que já foi citado nessa seção, ainda há o trabalho de Moyo and Mnkandla (2020) que propõe uma metodologia de desenvolvimento ágil de *software* com práticas de segurança. Essas práticas permitem que a validação de segurança do projeto seja apresentada em mais atividades do desenvolvimento ágil.

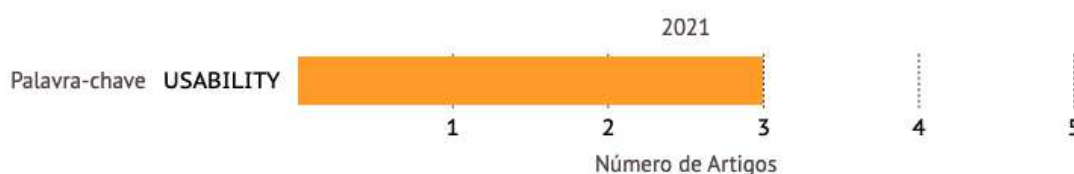
Em "*SOFTWARE DEVELOPMENT*", o trabalho de Evans et al. (2020a) apresenta, em uma pesquisa exploratória, o impacto de novas metodologias de desenvolvimento de *software* em Porto Rico. Lauesen (2020) apresenta o resultado de uma investigação de quatro incidentes que aconteceram com projetos de TI na Dinamarca, buscando a origem dos problemas e possíveis soluções, como por exemplo, problemas que seriam resolvidos com testes de usabilidade.

Por fim, em "*KNOWLEDGE MANAGEMENT*", o trabalho de Cunha et al. (2020) apresenta uma pesquisa sobre como novos membros de um time de desenvolvimento con-

seguem trabalhar com a dificuldade na passagem de conhecimento. Em Lima et al. (2020) existe uma discussão sobre uma análise quantitativa na identificação das dificuldades no processo de passagem de conhecimento.

Em 2021, 16 artigos selecionados. A palavra-chave mais encontrada, com três ocorrências, foi "*USABILITY*", como mostra na Figura 4.12. Esse tópico apareceu nos trabalhos de Morocho et al. (2021), Smirnov et al. (2021) e Ljungberg et al. (2021), auxiliando na resposta da pergunta de pesquisa P02 (*O que existe na literatura sobre usabilidade em relação à ferramentas de gestão de projetos de desenvolvimento de software?*).

Figura 4.12: Tópicos mais encontrados nos artigos de 2021



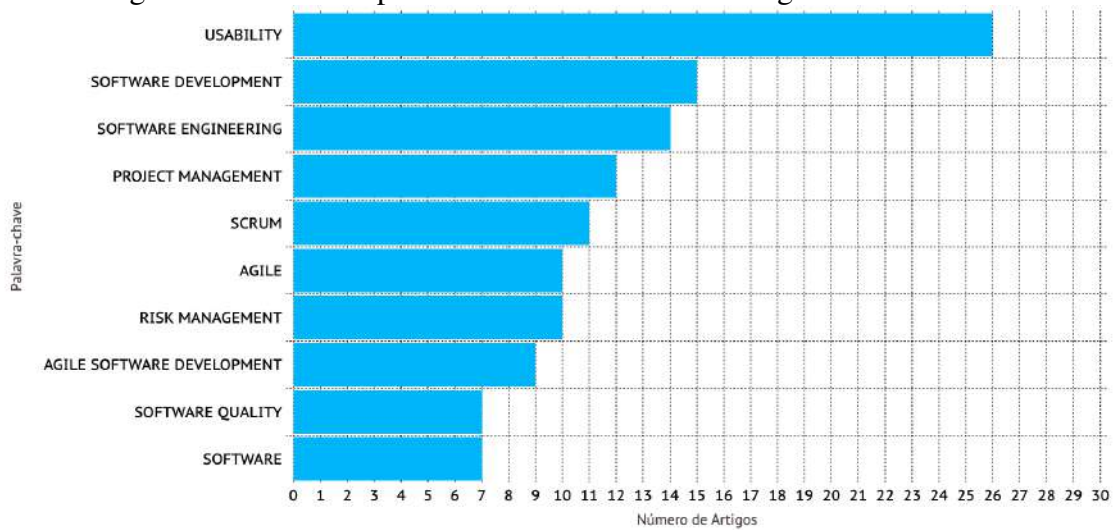
Fonte: O Autor

No trabalho de Morocho et al. (2021) é apresentada a implementação de um guia de riscos para televisão digital terrestre interativa, utilizando objetos de aprendizagem, passando por testes e análise de usabilidade. Em Smirnov et al. (2021), é apresentada a ferramenta *Revizor*, criada para automatizar mudanças frequentes no código-fonte de um sistema, destacando as linhas no código onde essa mudança pode ser aplicada. Ljungberg et al. (2021) apresenta um estudo de caso sobre implantação orientada por dados de análise de programas.

4.2 Classificação Por Tópico

Os artigos selecionados possuem 711 tópicos, e 91 (12,80% do total) são encontrados mais de uma vez. Esta seção apresenta os 10 tópicos mais recorrentes nos trabalhos selecionados, como apresentado na Figura 4.13. A partir desses 10 tópicos, o presente trabalho faz uma análise crítica baseada no número de publicações dos artigos, durante o período entre 2011 e 2021, criando uma linha do tempo do tópico, relacionado ao tema do trabalho, *ferramentas de gestão de projetos de desenvolvimento de software*. O gráfico completo, com todos os tópicos encontrados mais de uma vez nos artigos selecionados, é apresentado no Apêndice A.

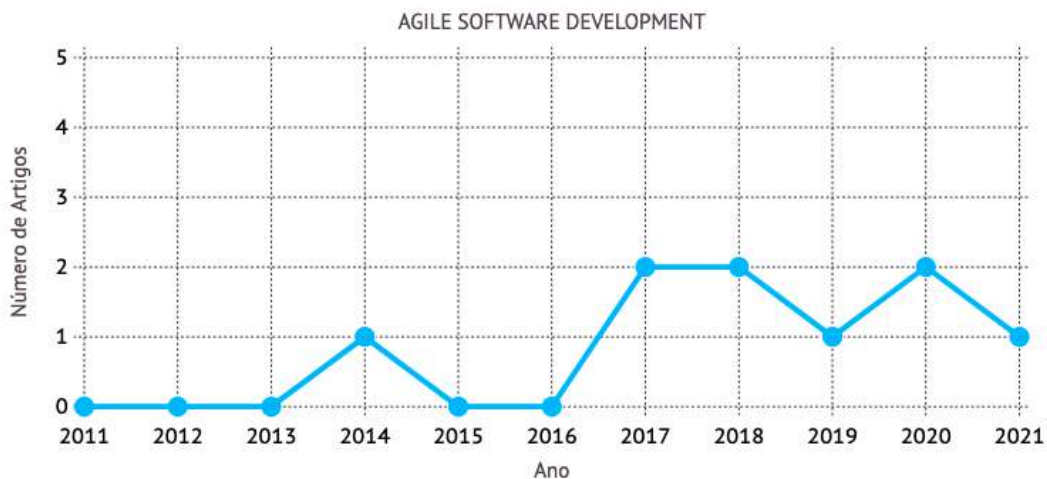
Figura 4.13: Os 10 tópicos mais encontrados nos artigos selecionados.



Fonte: O Autor

AGILE SOFTWARE DEVELOPMENT, ou desenvolvimento ágil de software, é encontrado nove vezes. Em 2020 foi um dos tópicos mais frequente nos artigos. Pela Figura 4.14 observa-se que os anos em que esse tópico mais aparece são os anos de 2017, 2018 e 2020, tendo duas ocorrências em cada ano. Foi encontrado uma vez em 2014, 2019 e 2021, mas não houve ocorrência nos demais anos pesquisados.

Figura 4.14: Linha do tempo de *AGILE SOFTWARE DEVELOPMENT*

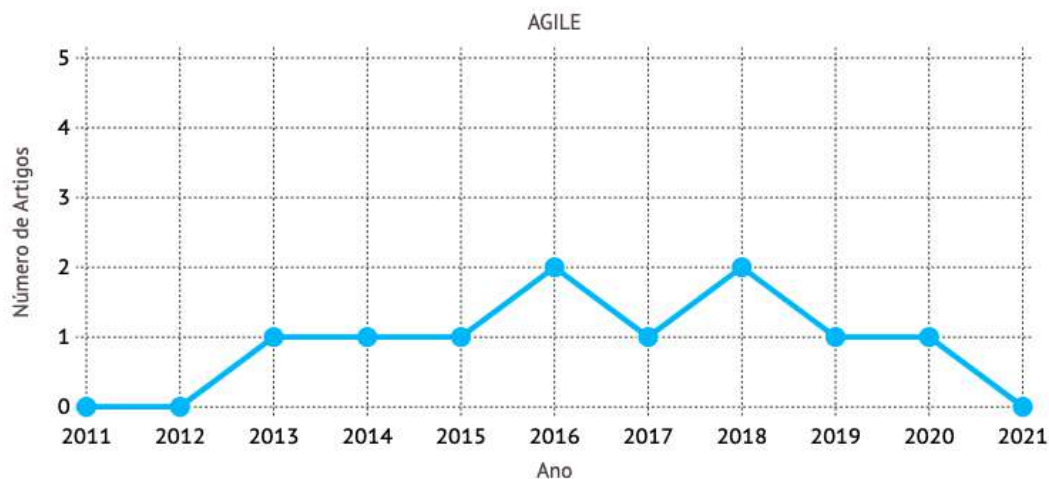


Fonte: O Autor

O desenvolvimento ágil de software apresenta baixa popularidade até 2016, com apenas um artigo publicado em 2014. Após esse período, nenhum ano deixou de ter publicações sobre o tema, indicando que houve tendência crescente em relação ao tema nesse período. Para completar a análise do tema, é interessante acompanhar também a linha do tempo do tópico *AGILE* na Figura 4.15.

AGILE, ou ágil, no qual os artigos se referem aos princípios dos métodos ágeis, foi encontrado dez vezes. 2016 e 2018 são os anos em que o tópico foi mais explorado, com duas publicações por ano. O tópico também é encontrado uma vez nos anos de 2013, 2014, 2017, 2019 e 2020, e não possui ocorrências nos demais anos pesquisados, como mostra a Figura 4.15.

Figura 4.15: Linha do tempo de *AGILE*

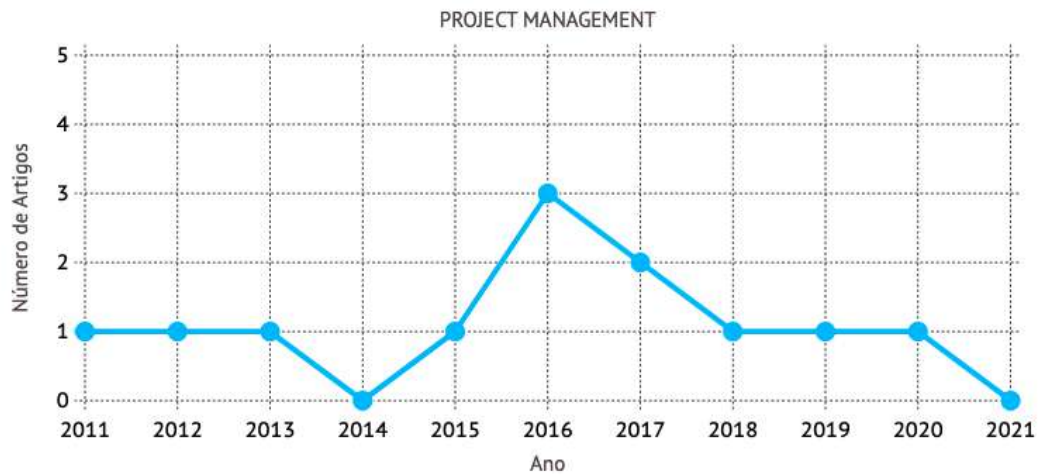


Fonte: O Autor

Observa-se a redução da popularidade de *AGILE* em 2011 e 2012, sendo que após estes anos apenas 2021 não possui publicações sobre o tópico, indicando possível exaustão do tema. Os artigos que possuem *AGILE* ou *AGILE SOFTWARE DEVELOPMENT* discutem questões relacionadas a métodos ágeis e desenvolvimento ágil de software, exemplificado pelos artigos Arcos-Medina and Mauricio (2020), Mousaei and Gandomani (2018), e Dilorenzo et al. (2020) em *AGILE SOFTWARE DEVELOPMENT* e El-Najar, Ahmad and Alkandari (2016), Hossain et al. (2020) e Hacaloglu and Demirors (2019) em *AGILE*. A comparação entre os dois gráficos indica em 2018 a maior tendência de trabalhos sobre o tópico, por ser o único ano que possui o maior número de publicações das duas figuras.

PROJECT MANAGEMENT, ou gestão de projeto, aparece 12 vezes nos tópicos dos artigos selecionados, sendo o quarto tópico mais encontrado. Esse tópico é visto de forma mais recorrente no ano de 2016, chegando a três menções. Foi encontrado duas vezes em 2017, não teve menções nos anos de 2014 e 2021, e foi encontrado apenas uma vez nos demais anos estudados, como apresenta a Figura 4.16.

A Figura 4.16 indica que o ano de maior popularidade de gestão de projetos foi 2016, ano em que o tópico possui três artigos selecionados nesse trabalho. Observa-se

Figura 4.16: Linha do tempo de *PROJECT MANAGEMENT*

Fonte: O Autor

também que nos outros anos estudados, o tópico sempre teve trabalhos, com exceção de 2014 e 2021, em que a falta de trabalhos no último ano estudado indica a perda de interesse em gestão de projetos.

RISK MANAGEMENT, ou gestão de riscos, tem um total de dez menções nos trabalhos encontrados. 2013 é o ano em que o tópico é mais citado, sendo o tópico mais encontrado no ano, com um total de três menções. O tópico aparece duas vezes em 2015 e 2018, uma vez em 2012 e 2014, e não é citado nos outros anos, como mostra a Figura 4.17.

Figura 4.17: Linha do tempo de *RISK MANAGEMENT*

Fonte: O Autor

A gestão de riscos tem um gráfico com picos de popularidade e perda de interesse com o passar dos anos, como indica a Figura 4.17. O período entre 2011 e 2013 apresenta

o maior crescimento em relação ao tópico, com três trabalhos encontrados em 2013, ano que apresenta a maior tendência sobre gestão de riscos. Após esse período, apesar de os anos 2016 e 2017 não possuírem artigos selecionados, houve um aumento da popularidade do tópico em 2018, mas novamente o tópico passa por dois anos sem publicações. Por fim, 2021 indica uma possível volta de popularidade em relação a gestão de riscos, e assim como aconteceu em 2012 e 2014, o próximo ano pode trazer ainda mais trabalhos sobre o tópico.

SCRUM é um *framework* que auxilia pessoas, times e organizações na solução de problemas complexos em projetos, no geral, de desenvolvimento de *software* (SCHWABER; SUTHERLAND, 2014). É o quinto tópico mais encontrado nos artigos, sendo a maior ocorrência em 2014, em que foi encontrado quatro vezes e também o tópico mais citado do ano. Em 2017 apareceu duas vezes, e não foi citado nos anos de 2014, 2015, 2019 e 2020. Nos outros anos, o tópico não foi citado, como apresenta a Figura 4.18.

Figura 4.18: Linha do tempo de *SCRUM*



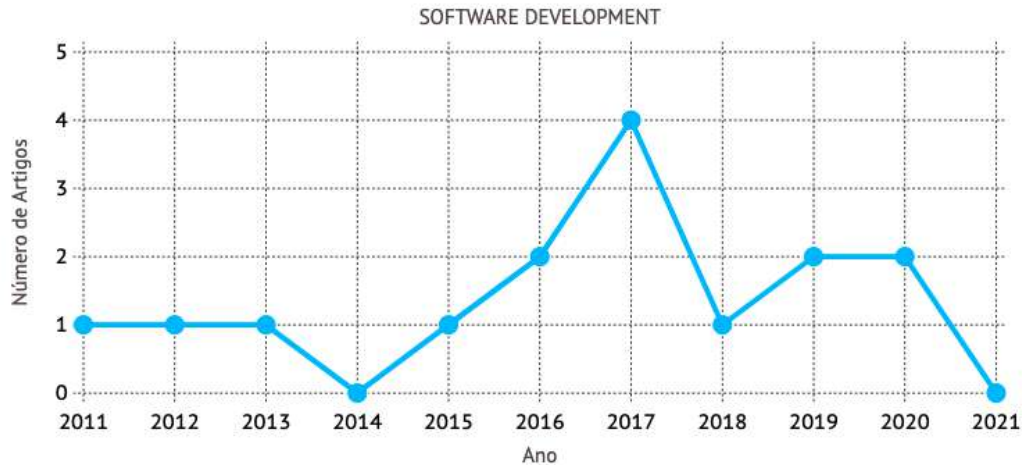
Fonte: O Autor

O gráfico apresentado na Figura 4.18 sobre o tópico *SCRUM* indica a tendência em 2018 sobre o assunto. Após dois anos sem publicações, o período entre 2015 e 2018 apresenta o crescimento da popularidade do tópico, chegando a quatro publicações em 2018. Essa tendência também indica possível esgotamento sobre o assunto, em que após seu pico de popularidade, passou dois anos sem publicações, voltando a ter um novo artigo publicado em 2021.

SOFTWARE DEVELOPMENT, ou desenvolvimento de *software*, é o segundo tópico mais encontrado nos artigos selecionados, encontrado 15 vezes. Em 2017, o tópico foi encontrado quatro vezes, sendo também o tópico mais citado no ano. Nos anos de

2016, 2019 e 2020 o tópico foi encontrado duas vezes, não foi citado em 2014 e 2021 e surgiu uma vez nos demais anos estudados, apresentado na Figura 4.19.

Figura 4.19: Linha do tempo de *SOFTWARE DEVELOPMENT*



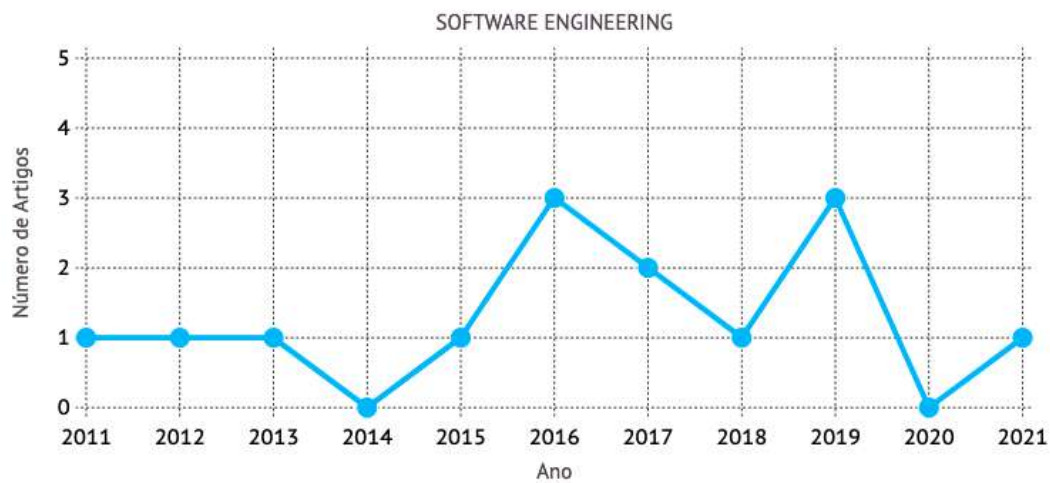
Fonte: O Autor

O gráfico sobre desenvolvimento de *software*, apresentado na Figura 4.19, apresenta o pico de popularidade sobre o tópico em 2017, com quatro artigos publicados. A tendência em relação ao tópico apresenta um crescimento no período entre 2014 e 2017. Após esse período, o número de publicações cai, com apenas uma publicação em 2018. O tópico se mantém estável até 2021, ano em que não teve publicações, indicando uma possível exaustão em relação ao tópico.

SOFTWARE ENGINEERING, ou engenharia de *software*, é o terceiro tópico mais encontrado nos artigos selecionados. O tópico é citado três vezes em 2016 e 2019, sendo também o tópico mais encontrado nesses anos, em conjunto com outros tópicos. Foi encontrado duas vezes em 2017, não foi citado em 2014 e 2020, e foi citado uma vez nos demais anos estudados, como mostra a Figura 4.20.

Em engenharia de *software*, o gráfico apresentado na Figura 4.20 indica que o tópico foi tendência em 2016 e 2019, atingindo três publicações nesses anos. Observa-se um crescimento na popularidade do tópico entre 2014 e 2016. Após o pico de 2019, o tópico não teve artigos publicados em 2020, mas voltou a ter publicações em 2021, o que pode indicar uma nova tendência em relação ao tema nos próximos anos, como aconteceu em 2014 e 2015.

SOFTWARE QUALITY, ou qualidade de *software* é encontrado sete vezes nos artigos selecionados. 2018, 2019 e 2020 são os anos em que o tópico mais aparece, encontrado duas vezes. No ano de 2011 o tópico surgiu uma vez e não foi citado nos demais

Figura 4.20: Linha do tempo de *SOFTWARE ENGINEERING*

Fonte: O Autor

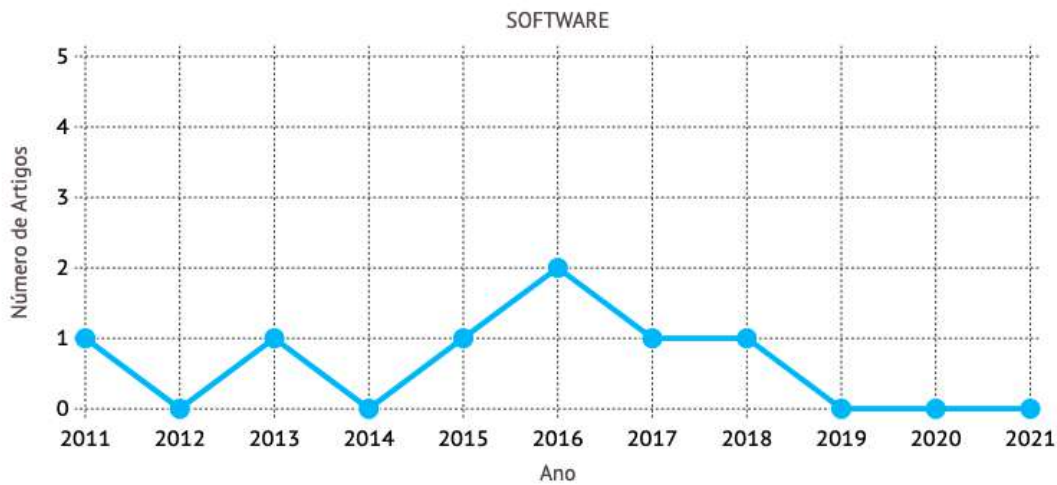
anos estudados, como mostra a Figura 4.21.

Figura 4.21: Linha do tempo de *SOFTWARE QUALITY*

Fonte: O Autor

O gráfico apresentado na Figura 4.21 indica a baixa popularidade de *SOFTWARE QUALITY* até o ano de 2017. O tópico teve seu maior número de publicações entre 2018 e 2020, se mantendo estável com duas publicações por ano. O tópico termina o período estudado com perda de popularidade em 2021, com nenhum trabalho publicado, segundo o mapeamento sistemático apresentado neste trabalho.

SOFTWARE é encontrado sete vezes nos artigos selecionados. O ano em que o tópico mais aparece é 2016, com duas citações. Em 2011, 2013, 2015, 2017 e 2018, o tópico aparece uma vez, e nos demais anos estudados, o tópico não é encontrado, como mostra a Figura 4.22.

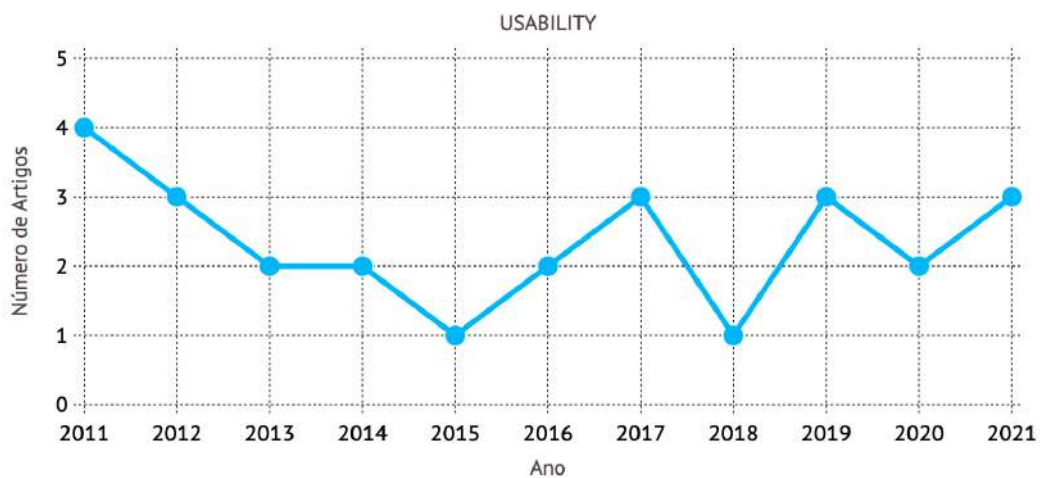
Figura 4.22: Linha do tempo de *SOFTWARE*

Fonte: O Autor

O gráfico de publicações por ano sobre *SOFTWARE*, apresentado na Figura 4.22, tem uma interpretação um pouco diferente quando se observa apenas esse tópico. É claro o pico de popularidade sobre o tópico em 2016, com duas publicações, mas deve se levar em consideração que o termo *SOFTWARE* pode ser encontrado em conjunto com *QUALITY*, *ENGINEERING* ou *DEVELOPMENT*, como visto na Figura 4.13 que apresenta os tópicos mais encontrados nos trabalhos.

USABILITY, ou usabilidade, é o tópico mais encontrado nos artigos selecionados, presente em todos os anos estudados. O tópico é encontrado cinco vezes em 2011, sendo também o tópico mais citado no ano. Em 2012, 2017, 2019 e 2020, o tópico aparece três vezes, e nos anos de 2012, 2019 e 2021, também foi o tópico mais encontrado. Em 2013, 2014, 2016 e 2020, o tópico apareceu duas vezes, sendo o tópico mais citado no ano de 2014. Nos anos de 2015 e 2018, o tópico foi citado uma vez, como mostra a Figura 4.23.

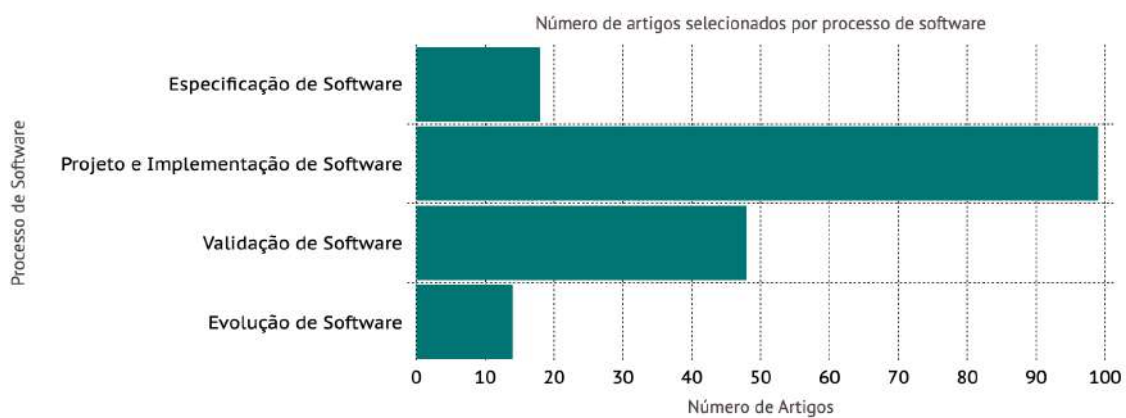
O gráfico de *USABILITY*, apresentado na Figura 4.23, indica que o tópico se mostrou relevante ao longo do período considerado neste mapeamento sistemático. Teve seu pico de popularidade em 2011, com quatro publicações e, após esse pico, passou por uma queda, com uma publicação em 2015. Houve mais um pico de popularidade em 2017, outro em 2019 e mais um em 2021. O período entre 2019 e 2021 indica tendência em relação ao tópico nos próximos anos, pois não deixou de ser relevante, ou seja, ainda não houve exaustão em relação à *USABILITY* quando associado à ferramentas de gestão de projetos de desenvolvimento de *software*.

Figura 4.23: Linha do tempo de *USABILITY*

Fonte: O Autor

4.3 Classificação por Atividades do Processo de Software

A classificação de artigos por atividade do processo de *software* apresenta a atividade de projeto e implementação de *software* com o maior número publicações entre as atividades. A atividade representa 55,56% dos trabalhos selecionados, somando 99 artigos. A atividade de especificação de *software* apresenta 18 trabalhos encontrados, validação de *software* apresenta 48 artigos encontrados e evolução de *software* apresenta 14 artigos, dentro dos artigos selecionados.

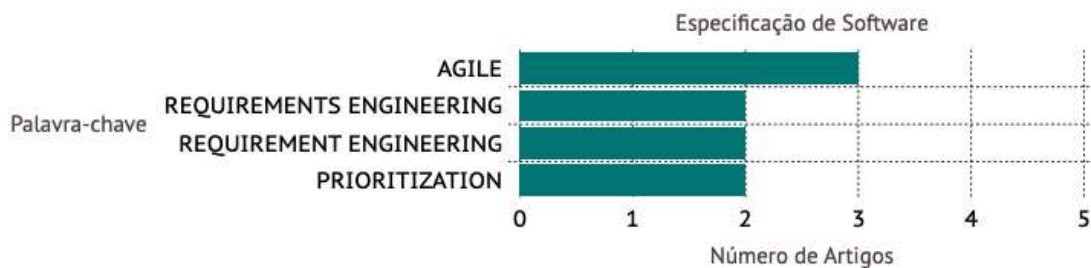
Figura 4.24: Artigos por atividade do processo de *software*

Fonte: O Autor

4.3.1 Principais Tópicos por Atividade

A atividade de especificação de *software* apresenta os seguintes tópicos encontrados mais de uma vez: *REQUIREMENTS ENGINEERING*, *REQUIREMENT ENGINEERING*, *PRIORITIZATION* e *AGILE*. *AGILE* foi o tópico mais encontrado, atingindo três ocorrências, como mostra a Figura 4.25. A tendência por publicações sobre a atividade de especificação teve seu maior crescimento no período entre 2017 e 2020, com 3 artigos em 2020 e nenhuma publicação em 2017.

Figura 4.25: Tópicos mais encontrados na atividade de especificação de *software*



Fonte: O Autor

Observa-se que os tópicos *REQUIREMENTS ENGINEERING* e *REQUIREMENT ENGINEERING* se referem ao levantamento de requisitos na atividade de especificação, no contexto em que aparecem nos trabalhos de El-Najar, Ahmad and Alkandari (2016), Ullah, Iqbal and Khan (2011), Fouad et al. (2011) e Felfernig et al. (2018). Apesar de *REQUIREMENTS ENGINEERING* e *REQUIREMENT ENGINEERING* possuírem o mesmo significado e aparecerem no mesmo contexto, o presente trabalho considera separadamente todos os tópicos encontrados.

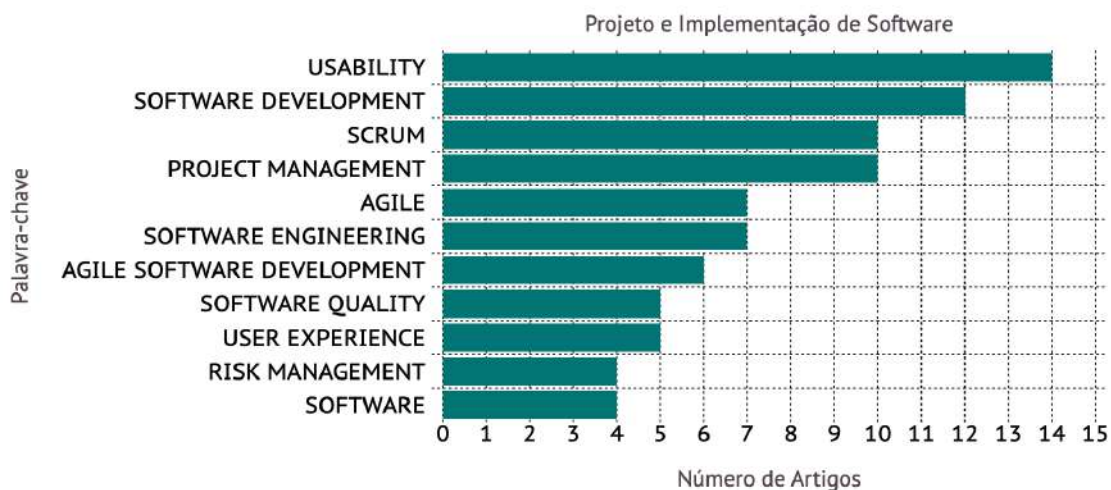
AGILE, o tópico mais encontrado, é explorado nos artigos de El-Najar, Ahmad and Alkandari (2016), Hossain et al. (2020) e Hacaloglu and Demirors (2019). O trabalho de El-Najar, Ahmad and Alkandari (2016) discute a falha na comunicação com o cliente de projetos de *software* e explora métodos existentes de comunicação, classificados em dois tipos: reuniões presenciais e via *softwares* de comunicação. Esses tipos de comunicação são comparados a partir de fatores como custo, confiabilidade e cobertura do projeto.

Em Hossain et al. (2020) o foco é na reutilização de requisitos, onde foram levantados 48 desafios na área e 43 técnicas para usar na implementação da reutilização de recursos em projetos globais de desenvolvimento de *software*, em que os times trabalham de forma distribuída. O trabalho de Hacaloglu and Demirors (2019) investiga o

uso da métrica de tamanho funcional (FSM), a qual é utilizada para estimar as atividades planejadas em um projeto de *software*, em projetos de desenvolvimento ágil.

A atividade do processo de *software* que mais obteve artigos selecionados foi a atividade de projeto e implementação. Com 99 artigos selecionados e mais de 500 tópicos encontrados, 39 surgem mais de uma vez, como mostra o gráfico completo de tópicos encontrados, na Figura 4.26 e na Figura 4.27.

Figura 4.26: Tópicos encontrados na atividade de projeto e implementação (1/2)



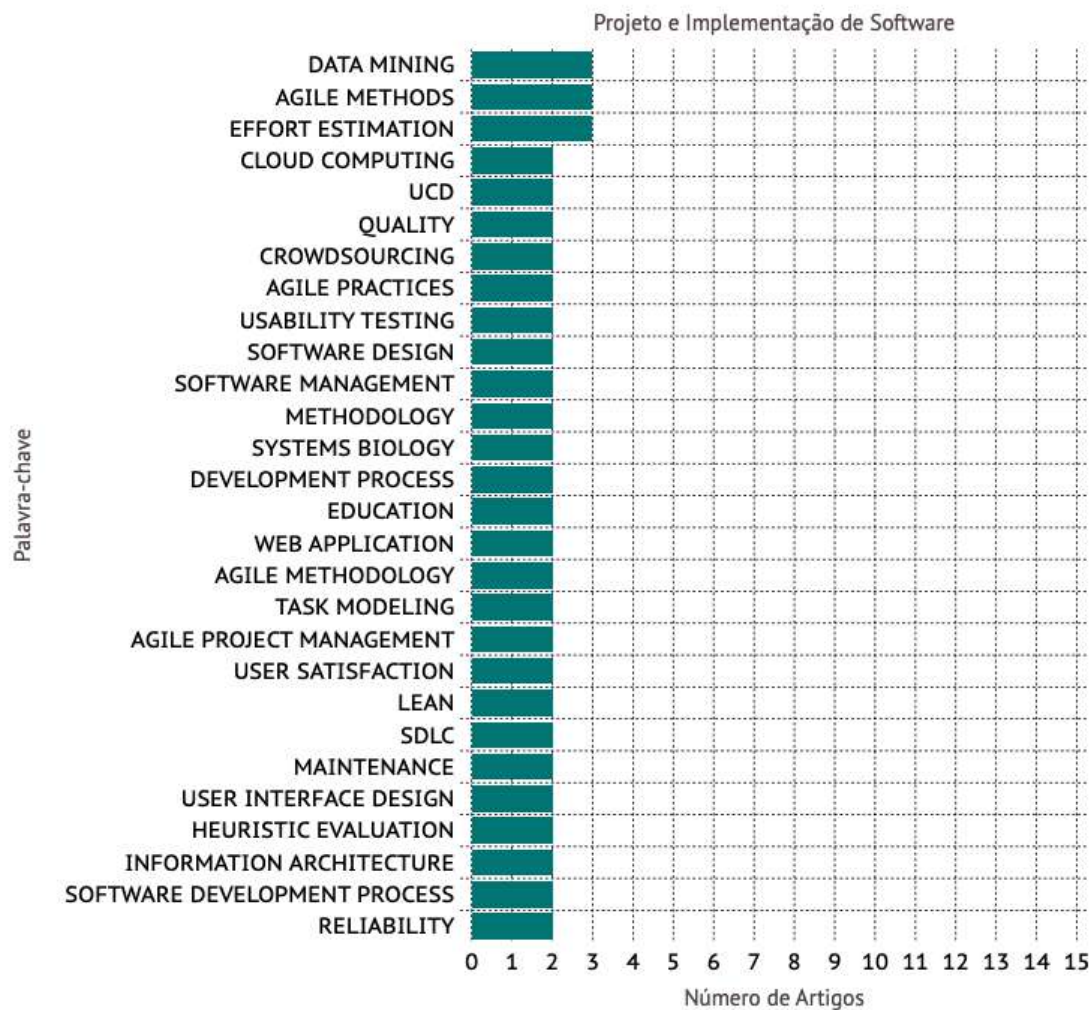
Fonte: O Autor

Observa-se entre os principais tópicos, *SOFTWARE DEVELOPMENT* e *SOFTWARE ENGINEERING*, indicando a relação com a implementação e desenvolvimento do *software*. Também são encontrados os tópicos *SCRUM*, *PROJECT MANAGEMENT* e *AGILE*, que representam a relação com o projeto de *software*, onde existe o acompanhamento e organização do projeto.

USABILITY, o tópico mais encontrado nessa atividade, presente nos trabalhos de Branham, Moxley and Ross (2015), Cayola and Macías (2018), Spichkova et al. (2016), Al-Busaidi et al. (2017), e Ramakrishnan and Gunter (2017). Também é encontrado nos trabalhos de Alomar et al. (2016), Conger (2011), Espinoza et al. (2019) Juárez-Ramírez et al. (2011), Mylly, Rajanen and Iivari (2019), Pierce (2012), Smaradottir (2017), Spencer, Zimmerman and Abramson (2011) e Ljungberg et al. (2021), que foram listados e apresentados na Seção 4.1.

Branham, Moxley and Ross (2015) discutem sobre consequências e benefícios em utilizar processos de validação de usabilidade não apenas na etapa de validação de um *software*, mas durante todo o processo de desenvolvimento de *software*. Em Cayola and Macías (2018) é apresentada uma abordagem para indicar métodos de usabilidade, de

Figura 4.27: Tópicos encontrados na atividade de projeto e implementação (2/2)



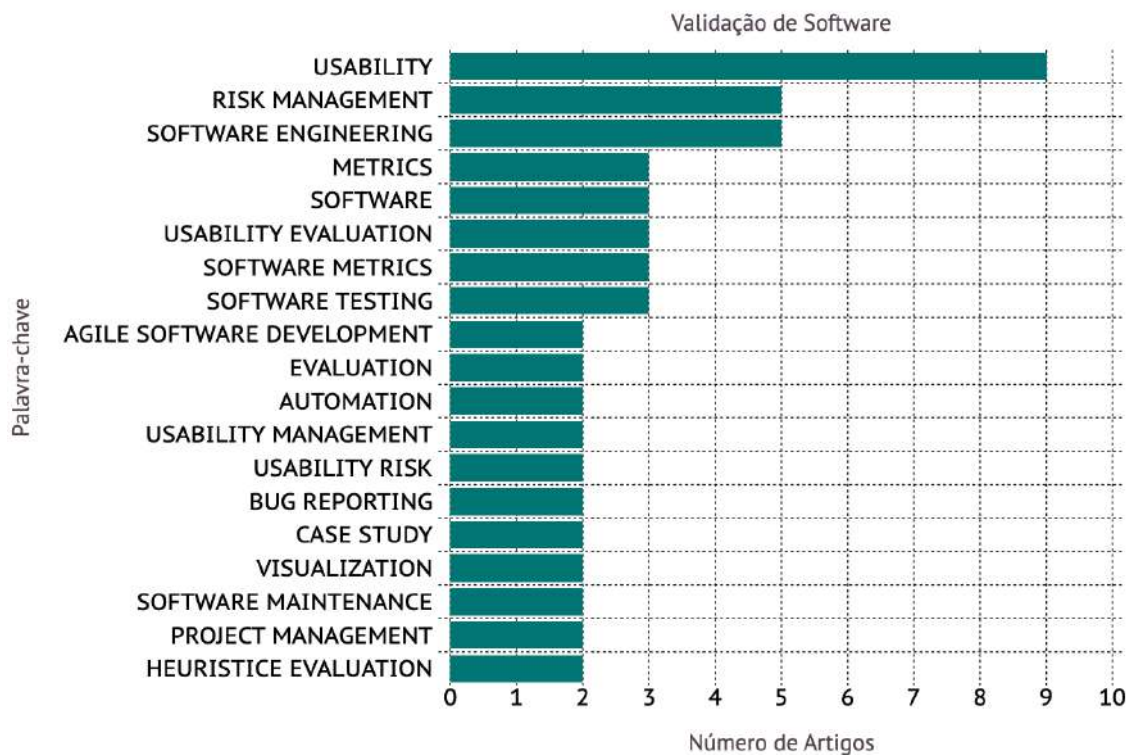
Fonte: O Autor

forma automática, a partir de informações sobre o estágio de desenvolvimento do projeto. No trabalho de Spichkova et al. (2016) é proposto um *framework* que facilita a modelagem dos principais componentes de uma plataforma, utilizando computação em nuvem, com foco em recursos de confiabilidade e usabilidade. Al-Busaidi et al. (2017) apresenta um projeto de sistema para usuários com deficiência visual. Em Ramakrishnan and Gunter (2017), os autores definem dez princípios para a criação de software para a ciência, com foco na usabilidade do software. Os princípios apontam diferenças entre o desenvolvimento comercial de software e o ciclo de vida do software em ambientes científicos.

A segunda atividade do processo de *software* que mais obteve artigos foi a validação de *software*. Nos 48 artigos selecionados, são encontrados 255 tópicos. Os tópicos encontrados mais de uma vez são apresentados na Figura 4.28. Além de *USABILITY*, que aparece nove vezes, representando a etapa de validação de *software* observa-se *RISK*

MANAGEMENT, *USABILITY EVALUATION*, *SOFTWARE TESTING*, *SOFTWARE METRICS* e *METRICS*, que indicam validações de segurança, levantamento de métricas do projeto e testes em geral.

Figura 4.28: Tópicos mais encontrados na atividade de validação de *software*



Fonte: O Autor

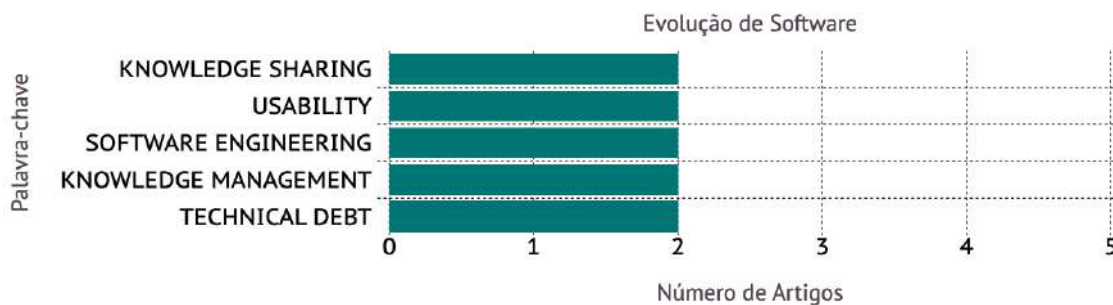
USABILITY é encontrado nos trabalhos de Evans et al. (2020b) e Al-azawi, Ayesha and Obaidy (2013). É encontrado também em Deraman and Salman (2019), Lanna and Amyot (2011), Morocho et al. (2021), Smirnov et al. (2021), Rahmat and Hanifah (2020), Moorthy, Ibrahim and Mahrin (2014) e Lindholm and Host (2013), trabalhos que já foram listados na Seção 4.1.

Em Evans et al. (2020b) é apresentada uma pesquisa sobre ferramentas de teste, validando a usabilidade dessas ferramentas e mostrando que apenas a usabilidade não é suficiente para que o uso dessas ferramentas represente o sucesso do projeto. O trabalho também discute possíveis problemas no uso a longo prazo dessas ferramentas. O trabalho de Al-azawi, Ayesha and Obaidy (2013) apresenta uma investigação sobre fase de avaliação de jogos, com foco em heurísticas que podem facilitar o processo de validação para qualquer projeto desenvolvimento de jogos.

Evolução de *software* foi a atividade encontrada no menor número de artigos. Os 14 artigos encontrados sobre a atividade de evolução apresentam 69 tópicos. Desses 69,

cinco tópicos aparecem mais de uma vez, como é apresentado na Figura 4.29. Além de *USABILITY*, que aparece em três das quatro atividades, os tópicos, *KNOWLEDGE SHARING*, *KNOWLEDGE MANAGEMENT* e *TECHNICAL DEBT* surgem em contexto de evolução de projeto, desde a passagem de conhecimento sobre o projeto para outros desenvolvedores, até a resolução de débitos técnicos do projeto.

Figura 4.29: Tópicos mais encontrados na atividade de evolução de *software*



Fonte: O Autor

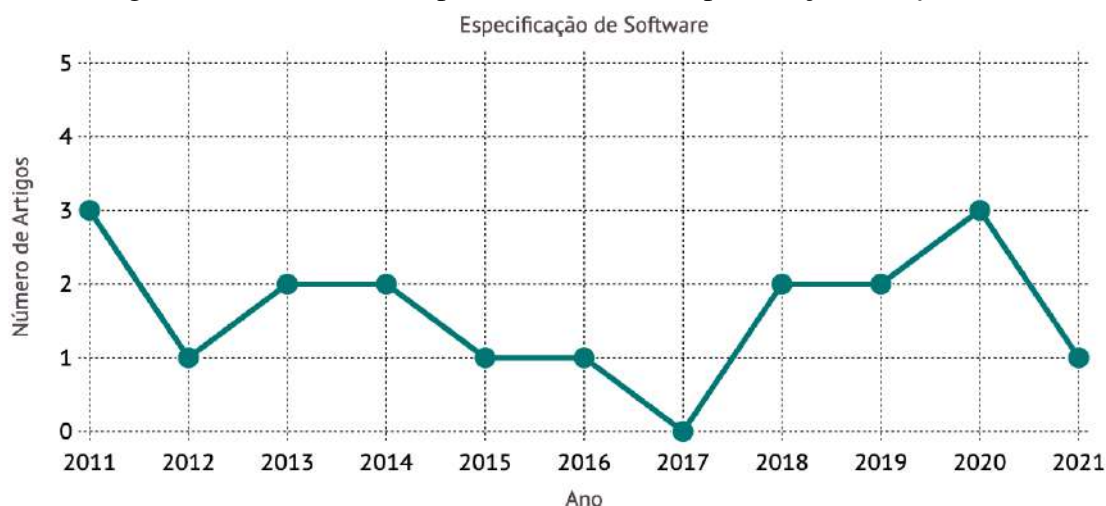
Os tópicos encontrados mais de uma vez são compartilhados por oito artigos da seguinte forma: em Al-Qaimari and Agi (2012) é encontrado *USABILITY*; em Lima et al. (2020) são encontrados *KNOWLEDGE MANAGEMENT* e *KNOWLEDGE SHARING*; em McCalden (2012) é encontrado *USABILITY*; em Sharma and Kumar (2019) é encontrado *SOFTWARE ENGINEERING*; Lage et al. (2019) é encontrado *TECHNICAL DEBT*; em Zanatta et al. (2017) é encontrado *SOFTWARE ENGINEERING*; em Prabhu et al. (2011) são encontrados *KNOWLEDGE MANAGEMENT* e *KNOWLEDGE SHARING*; e em Zazworka et al. (2013) é encontrado *TECHNICAL DEBT*. Desses trabalhos, apenas Lage et al. (2019), Prabhu et al. (2011) e Zazworka et al. (2013) ainda não foram citados. Os outros trabalhos se encontram na Seção 4.1.

Lage et al. (2019) apresenta uma caracterização inicial de débitos técnicos de usabilidade em projetos de *software*, para tentar cobrir o maior número de problemas de usabilidade. Em Prabhu et al. (2011) é apresentado o impacto do gerenciamento de conhecimento na qualidade do *software* desenvolvido, e uma pesquisa com 70 gerentes de projeto para validação. O trabalho de Zazworka et al. (2013) é um estudo de caso sobre identificação eficaz de débito técnico, comparando débitos encontrados de forma automática e de forma humana.

4.3.2 Linha do Tempo por Atividade

Na atividade de especificação de *software*, dos 18 artigos selecionados, três foram publicados em 2011 e mais três em 2020, que são os anos com mais artigos publicados sobre o tema, como mostra a Figura 4.30. O ano com menor número de publicações foi 2017, que não teve artigos.

Figura 4.30: Linha do tempo da atividade de especificação de *software*



Fonte: O Autor

Além do pico de publicações em 2011, o gráfico presente na Figura 4.30 indica também o crescimento da popularidade em relação à atividade de especificação de *software* no período entre 2017 e 2020. Em 2021, com apenas um artigo, indica que a popularidade sobre o assunto caiu novamente, e como o período entre 2016 e 2017, pode não ter artigos no próximo ano ou possuir mais artigos publicados, como período entre 2012 e 2013.

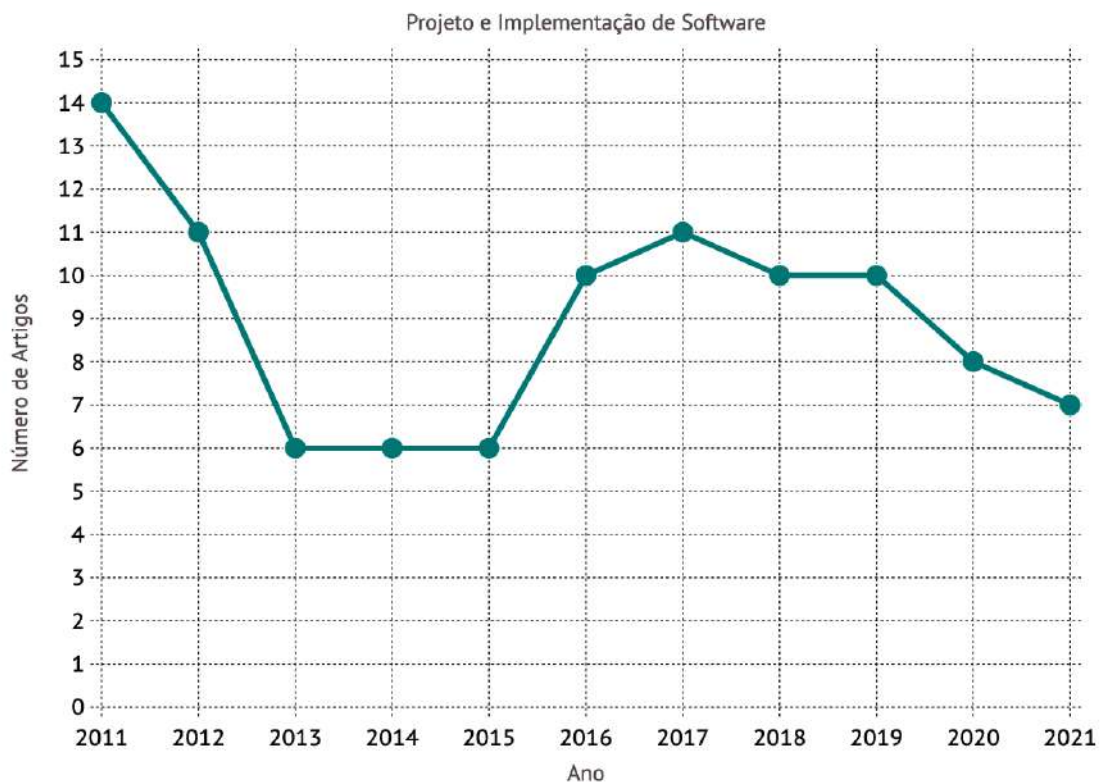
Em 2011, os três trabalhos encontrados são os artigos de Fouad et al. (2011), Wilkie et al. (2011) e Ullah, Iqbal and Khan (2011). Em Fouad et al. (2011) temos um estudo sobre a incorporação de requisitos dentro de uma arquitetura orientada a modelos, com ênfase no valor da notação de modelagem de processo de negócios para analistas de negócio. O trabalho de Wilkie et al. (2011) discute a utilização de técnicas para estimativa de custo e tempo, em relação a projetos de desenvolvimento de *software*. Ullah, Iqbal and Khan (2011) traz uma pesquisa sobre questões envolvendo a elicitação de requisitos não funcionais em projetos de *software*.

Em 2020 aparecem os artigos de Hossain et al. (2020) e Ruiz and Hasselman (2020), que foram citados na Seção 4.1, e o trabalho de Salamea et al. (2020). Salamea et

al. (2020) apresenta um estudo sobre estimativa e priorização de requisitos de qualidade no desenvolvimento de *software*.

A atividade de projeto e implementação de *software*, com 99 artigos selecionados, teve 14 artigos no ano de 2011, o ano com mais publicações sobre o tema. Nenhum dos anos estudados deixou de ter contribuições sobre o tema, como mostra a Figura 4.31, mas 2013, 2014 e 2016 foram os anos com menos publicações, seis em cada ano, respectivamente.

Figura 4.31: Linha do tempo da atividade de projeto e implementação de *software*



Fonte: O Autor

A linha do tempo da atividade de projeto e implementação, apresentada na Figura 4.31, indica que o maior pico de popularidade sobre a atividade foi em 2011, com 14 artigos. No período entre 2011 e 2013 existe uma queda expressiva em relação às publicações sobre a atividade, com seis artigos publicados em 2013. Entre 2015 e 2017 a atividade ganha força novamente, se mantendo estável até 2019, onde tem novamente uma queda na popularidade e deixa de ser tendência. Observa-se também que o gráfico indica bastante espaço na pesquisa sobre atividade de projeto e implementação de *software*, pois no período entre 2013 e 2015, com menos publicações por ano, a atividade ainda possui seis publicações por ano.

Dos 14 artigos selecionados no ano de 2011, os trabalhos de Conger (2011), Juárez-Ramírez et al. (2011) e Spencer, Zimmerman and Abramson (2011) já foram apresentados na Seção 4.1. Os 11 trabalhos restantes são de Botzenhardt, Meth and Maedche (2011), Glavinic, Ljubic and Kukec (2011), Gonçalves and Santos (2011), Maier et al. (2011), Lacheiner and Ramler (2011), Viikki and Palviainen (2011), Zhang et al. (2011), Nunes, Constantine and Kazman (2011), Agarwal (2011), Williams (2011) e Linehan et al. (2011).

Botzenhardt, Meth and Maedche (2011) falam sobre a integração da gestão de produto com o design do produto no desenvolvimento de aplicações de *software*, explorando fatores de sucesso. Em Glavinic, Ljubic and Kukec (2011) são apresentadas formas de suportar usabilidade universal em *software* com foco em telas sensíveis ao toque. O trabalho de Gonçalves and Santos (2011) apresenta um *software* para criar protótipos, de baixa fidelidade, de interfaces em projetos de desenvolvimento ágil.

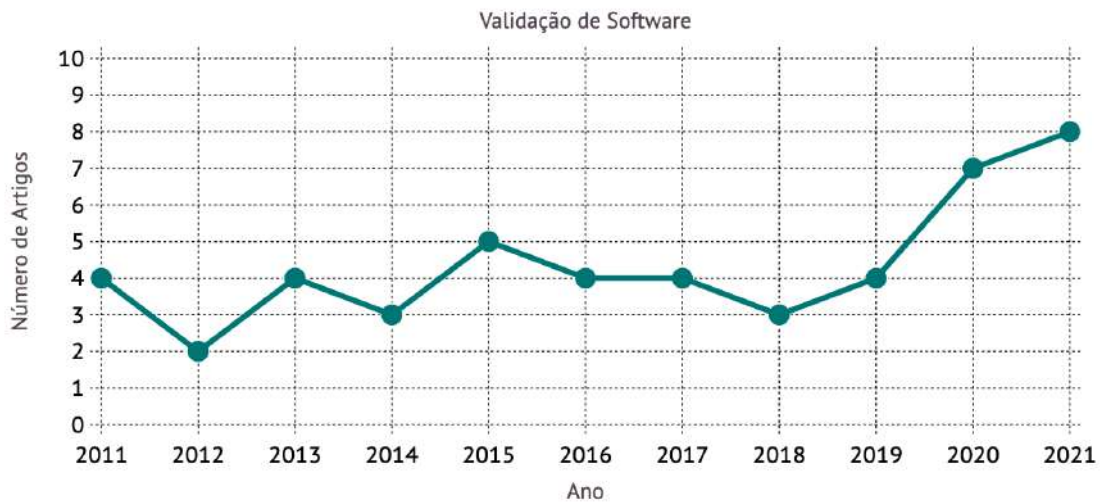
Maier et al. (2011) discute a aplicação da distribuição de conhecimento para biologia de sistemas. O trabalho de Lacheiner and Ramler (2011) apresenta o gerenciamento do ciclo de vida de aplicativos, para melhoria e evolução do processo de *software*. Viikki and Palviainen (2011) explora a integração do design centrado no ser humano com o desenvolvimento de *software* na indústria de automação. No trabalho de Zhang et al. (2011) é apresentada uma maneira de fazer alocação de recursos baseado em estatística para computação em nuvem. Em Nunes, Constantine and Kazman (2011) é apresentada uma proposta de adaptação de *use-case point* (UCP) focada no desenvolvimento ágil, estimando o custo de projetos de *software* de forma interativa durante o processo de desenvolvimento.

Agarwal (2011) apresenta uma visualização animada do processo de desenvolvimento de *software*, a partir de um simulador, para auxiliar gestores de projeto na tomada de decisões a partir das variáveis do projeto. Em Williams (2011) são apresentados sete hábitos de engenheiros de *software* que podem impactar a evolução do projeto, melhorando a relação do engenheiro com o projeto, como proatividade e melhoria contínua. Por fim, no trabalho de Linehan et al. (2011) é apresentada uma versão controlada de linguagem natural em Inglês, baseado na semântica do vocabulário de negócio e de regras de negócio, para produção e processamento de eventos dentro de um projeto de *software*.

A atividade de validação de *software*, segunda atividade com mais artigos selecionados, teve o maior número de artigos por ano em 2021, com oito trabalhos escolhidos, apresentado na Figura 4.32. Nenhum ano do período estudado deixou de ter publicações

sobre o tema, mas 2012, o ano com menos publicações, teve apenas dois artigos. Essa atividade apresenta crescimento constante da popularidade, indicando pela Figura 4.32, que o interesse por validação de *software* está aumentando, ou seja, é uma tendência.

Figura 4.32: Linha do tempo da atividade de validação de *software*



Fonte: O Autor

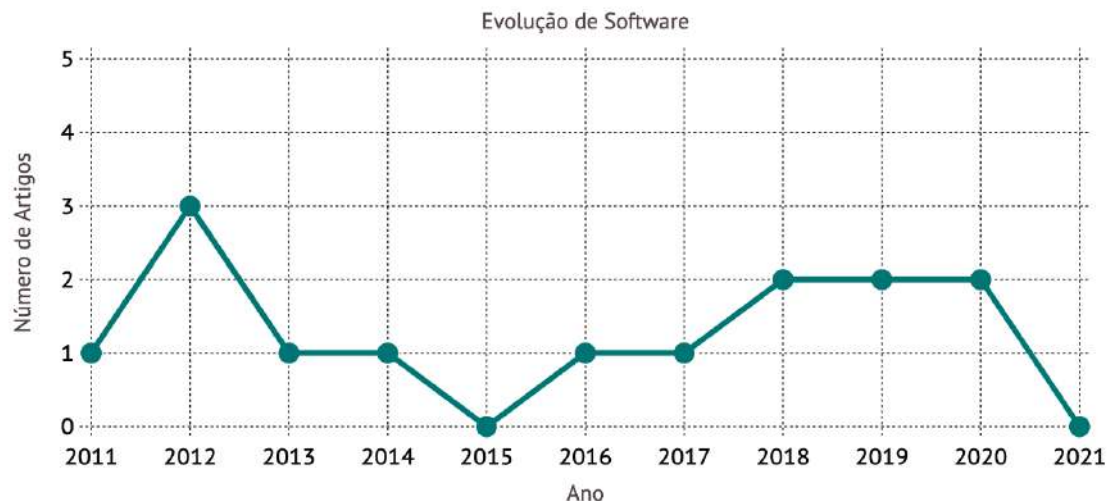
No ano de 2021, os trabalhos de Morocho et al. (2021) e Smirnov et al. (2021), já foram citados na Seção 4.1. Além desse, são encontrados também os artigos de Ali et al. (2021), Chakravarty and Singh (2021), Tao et al. (2021), Majeed et al. (2021), Reynolds et al. (2021), Imran, Ciborowska and Damevski (2021).

Ali et al. (2021) apresenta um modelo de trabalho compatível com *Capability Maturity Model Integration (CMMI)* para estabelecer a integridade da gestão de configurações em *software*, para pequenas e médias empresas, para validação de clareza, usabilidade e utilidade dos modelos. O trabalho de Chakravarty and Singh (2021) mostra um estudo sobre métricas de qualidade de *software* em ambientes de desenvolvimento ágil, com essas métricas focando na qualidade, no processo e nos parâmetros do produto, sendo influenciadas por fatores como a funcionalidade, usabilidade e confiabilidade do produto. Em Tao et al. (2021) é apresentado um método de validação de usabilidade para plataformas de *E-learning*, baseado em avaliação abrangente *fuzzy*. Majeed et al. (2021) mostra um estudo comparativo de ferramentas *open source* para automação de testes, usando as ferramentas *Selenium*, *Katalon Studio* e *Test Project*. O trabalho de Reynolds et al. (2021) discute design centrado no usuário para visualização de relatórios de vulnerabilidade de *software*. Por fim, Imran, Ciborowska and Damevski (2021) descreve o levantamento de perguntas de acompanhamento para relatórios de *bugs* com informações incompletas, na tentativa de agilizar o processo do levantamento de informação para resolver e corrigir

problemas de *software*.

A atividade de evolução de *software*, com 14 artigos selecionados, teve seu pico de publicações no ano de 2012, com três trabalhos encontrados. 2015 e 2021 foram os anos com menos publicações no tema, com zero publicações, como mostra a Figura 4.33.

Figura 4.33: Linha do tempo da atividade de evolução de *software*



Fonte: O Autor

O gráfico apresentado na Figura 4.33 indica que evolução é a atividade menos popular entre as atividades do processo de *software*. Indica também que apesar da estabilidade em publicações no período entre 2018 e 2020, perdeu força em 2021.

Dos três artigos selecionados em 2012, os trabalhos de Al-Qaimari and Agi (2012) e McCalden (2012) já foram listados na Seção 4.1. O trabalho de Fagerholm and Münch (2012) apresenta a visão do projeto a partir da experiência dos desenvolvedores, mostrando a importância da visão do desenvolvedor sobre o que foi desenvolvido, documentado e sobre decisões de projeto, para evolução de projetos de *software*.

4.4 Discussão e Perguntas de Pesquisa

A análise quantitativa e o resumo das contribuições dos trabalhos selecionados auxiliam na resposta da pergunta de pesquisa P01 (*O que existe na literatura sobre ferramentas de gestão de projetos de desenvolvimento de software nos últimos onze anos?*). Para sintetizar essa informação, a Seção 4.1 apresenta quais os tópicos mais encontrados por ano de publicação dos trabalhos selecionados. Com as informações apresentadas na Seção 4.2, sobre quais os dez tópicos mais encontrados nos artigos selecionados, é possí-

vel observar quais os tópicos mais encontrados na literatura sobre ferramentas de gestão de projetos de desenvolvimento de software, mas não significa que esses tópicos são a ênfase dos artigos em que são encontrados.

Em relação à pergunta de pesquisa P02 (*O que existe na literatura sobre usabilidade em relação a ferramentas de gestão de projetos de desenvolvimento de software?*), é possível afirmar, pelas informações apresentadas na Seção 4.2, que usabilidade é o tópico mais encontrado na literatura sobre ferramentas de gestão de projetos de desenvolvimento de software, considerando o mapeamento sistemático apresentado neste trabalho.

Além disso, dos onze anos estudados, no período entre 2011 e 2021, seis desses anos possuem usabilidade como o tópico mais encontrado no ano. É importante ressaltar que, mesmo os trabalhos que possuem usabilidade como palavra-chave, podem não ter esta palavra-chave como ênfase do trabalho. O tópico usabilidade foi encontrado em todas as atividades do processo de software, ou seja, pôde ser relacionado com especificação, projeto, implementação, validação e evolução de software.

Ao comparar o número de trabalhos contendo usabilidade e utilidade como palavra-chave, é possível afirmar que usabilidade foi mais relevante que utilidade, considerando o mapeamento sistemático apresentado neste trabalho. Enquanto o tópico usabilidade foi encontrado em 29 dos 179 trabalhos selecionados, o tópico utilidade foi encontrado em apenas um trabalho, que apresenta um estudo sobre os impactos do uso de metodologias ágeis no desenvolvimento de software, respondendo a pergunta de pesquisa P03 (*O que existe na literatura sobre utilidade em relação a ferramentas de gestão de projetos de desenvolvimento de software?*).

4.5 Resumo do Capítulo

Neste capítulo foi apresentado o resultado do mapeamento sistemático pela classificação dos artigos. A informação obtida, através do mapeamento sistemático, se refere a trabalhos que possuem relação com ferramentas de gestão de projetos de desenvolvimento de *software*, usabilidade e utilidade.

A classificação dos artigos por ano de publicação apresentou que em seis dos onze anos estudados, o tópico *USABILITY* foi o tópico mais encontrado nos artigos selecionados. Apresentou também que 2011 foi o ano com o maior número de artigos encontrados, representando 12,29% do total. Nos anos em que *USABILITY* não foi o tópico mais encontrado, os tópicos mais encontrados foram *RISK MANAGEMENT* em 2013 e

2014, *SOFTWARE ENGINEERING* e *PROJECT MANAGEMENT* em 2016, *SOFTWARE DEVELOPMENT* em 2017 e *SCRUM* em 2018. Para cada um desses tópicos, foram apresentados resumos das contribuições dos artigos selecionados em cada ano. O resumo das contribuições dos trabalhos encontrados, a partir dos tópicos dos trabalhos, apresenta que nem sempre o tópico é o foco dentro do contexto do artigo apresentado.

Na classificação por tópico mais encontrado, o capítulo apresentou a linha do tempo dos dez tópicos mais encontrados em todos os artigos selecionados. *USABILITY* foi o tópico mais encontrado, surgindo em 26 artigos dos 176 selecionados. A análise de popularidade e tendência de cada um dos dez tópicos mais encontrados foi feita, apresentando que *AGILE* e *AGILE SOFTWARE DEVELOPMENT* tiveram seu pico de popularidade juntos em 2018. *PROJECT MANAGEMENT* cresceu entre 2014 e 2016, mas, após esse período, teve queda na popularidade até ter zero publicações em 2021. *RISK MANAGEMENT* teve seu maior pico de popularidade em 2013.

SCRUM teve um grande crescimento na popularidade entre 2015 e 2018, e nesse período, foi tendência, até cair para zero publicações em 2019. *SOFTWARE DEVELOPMENT* teve mais publicações em 2017 e após esse período, o tema entrou em exaustão. *SOFTWARE ENGINEERING* teve dois picos de popularidade, em 2016 e 2019, perdendo força em 2020. *SOFTWARE QUALITY* se manteve estável entre 2018 e 2020, e, após esse período, não teve publicações. *SOFTWARE* apresentou aumento da popularidade entre 2014 e 2016, estabilizando após esse período, perdendo força nos anos seguintes, chegando a zero publicações de 2019 a 2021. *USABILITY* não deixou de ter publicações em nenhum ano, com pico de quatro publicações em 2011.

Neste capítulo também foi apresentada a classificação por atividade do processo de *software*. A atividade de especificação de *software* teve 15 artigos selecionados, e dois picos de publicações, com três artigos em 2011 e 2020. Além disso, o tópico mais encontrado na atividade foi *AGILE*. A atividade de projeto e implementação de *software* foi a mais popular, com 99 artigos encontrados e *USABILITY* como tópico mais encontrado. A atividade também teve seu pico de publicações em 2014, com 14 artigos publicados.

A atividade de validação de *software* apresentou 18 artigos selecionados e o tópico mais encontrado foi *USABILITY*. Além disso, a atividade se apresentou como tendência para os próximos anos, crescendo entre 2018 e 2021. Por fim, a atividade de evolução do *software* teve 14 artigos selecionados, os tópicos mais encontrados foram *KNOWLEDGE SHARING*, *USABILITY*, *SOFTWARE ENGINEERING*, *KNOWLEDGE MANAGEMENT* e *TECHNICAL DEBT*.

5 CONCLUSÃO

Este trabalho apresentou um mapeamento sistemático da literatura sobre ferramentas de desenvolvimento de *software* no período de 2011 a 2021. Com base no mapeamento sistemático realizado, é possível constatar que durante esse período, a usabilidade foi constantemente estudada, seja em relação ao início de um projeto, quando se realiza o levantamento dos requisitos necessários para o desenvolvimento do *software*, seja nas etapas de validação e evolução, onde existem testes de usabilidade que podem ser realizados. Também é possível observar que, diferente de usabilidade, a utilidade não teve destaque no mapeamento realizado, sendo identificada em apenas um artigo, que apresenta um estudo sobre os impactos do uso de metodologias ágeis no desenvolvimento de *software*.

A partir das classificações realizadas no contexto do mapeamento sistemático, foi constatado que a falta de publicações na atividade de evolução de *software* demonstra potencial de evolução do tema, pois diariamente *softwares* novos são construídos e precisam estar preparados para novas necessidades encontradas pelos *stakeholders*. No presente trabalho foi observado que, na atividade de evolução, uma grande quantidade de publicações discutem documentação e transferência de conhecimento, o que reflete o atual momento do mercado de TI, pois evoluir *software* de forma planejada é menos custoso que construir *softwares* novos do zero.

A atividade de especificação de *software* foi a segunda atividade com menor número de publicações, atividade que também apresenta potencial de investigação e evolução. Na concepção de um novo produto de *software*, a atividade de especificação é a etapa em que se entende quais as necessidades do cliente. Isso é importante, pois erros nessa atividade afetam diretamente o custo do projeto e o tempo definido para a entrega.

Por fim, a atividade de validação de *software* apresentou crescimento de popularidade nos últimos anos, indicando tendência para o futuro. Esse interesse também reflete o mercado de TI, pois o processo de validação e verificação de *software* é bastante custoso. Assim, esforços para a automação dessa atividade auxiliam na diminuição do custo de projetos de *software*.

5.1 Contribuições e Limitações da Pesquisa

O mapeamento sistemático, apresentado neste trabalho trouxe quais os tópicos mais encontrados na literatura, sobre ferramentas de gestão de projetos de desenvolvimento de *software*, no período investigado. Este mapeamento trouxe informações sobre como as contribuições dos trabalhos selecionados se apresentam através das atividades de processo de *software*, trazendo uma análise quantitativa das palavras-chave encontradas nesses trabalhos. O trabalho trouxe também a relevância do tópico usabilidade, em relação a ferramentas de desenvolvimento de *software*, e a falta de publicações sobre o tópico utilidade em relação a essas ferramentas. Entretanto, o mapeamento sistemático apresentado no trabalho e a metodologia de pesquisa apresentada passam por validação apenas de uma pessoa.

Além dos critérios de inclusão, que foram validados de forma manual, validando se os artigos estavam no contexto do tema, a classificação dos artigos por atividade do processo de *software* também foi feita de forma manual. Por terem sido feitas de forma manual, pode haver divergências na seleção dos artigos por outros pesquisadores. Além disso, as análises quantitativas apresentadas neste trabalho levam em consideração apenas as palavras-chave encontradas nos trabalhos selecionados, e não o contexto em que aparecem no conteúdo dos trabalhos.

5.2 Trabalhos Futuros

Com os resultados obtidos neste trabalho, é possível que novos estudos sejam apresentados em relação às ferramentas de gestão de projetos de *software*. Este trabalho destaca como possibilidades de estudo os seguintes temas:

- Projeto de construção de ferramenta de gestão de desenvolvimento de *software* baseado nas melhorias encontradas.
- Pesquisa com usuários sobre satisfação com as ferramentas de gestão de desenvolvimento de *software* utilizadas.
- Como as ferramentas de gestão de desenvolvimento de *software* podem auxiliar na documentação e passagem de conhecimento para times que possuem alta rotatividade.

REFERÊNCIAS

ABES. **Mercado Brasileiro de Software – Panorama e Tendências 2012**. ABES – Associação Brasileiras das Empresas de Software, 2012. Available from Internet: <https://abessoftware.com.br/wp-content/uploads/anterior/Arquivos/2012_Publicacao_Mercado_ABES.pdf>.

ABES. **Mercado Brasileiro de Software – Panorama e Tendências 2022**. ABES – Associação Brasileiras das Empresas de Software, 2022. Available from Internet: <<https://abessoftware.com.br/download/45142/>>.

ACM. 2022. Available from Internet: <<https://dl.acm.org/>>.

AGARWAL, R. Software development process animation. In: **Proceedings of the 49th Annual Southeast Regional Conference**. New York, NY, USA: Association for Computing Machinery, 2011. (ACM-SE '11), p. 221–226. ISBN 9781450306867. Available from Internet: <<https://doi.org/10.1145/2016039.2016098>>.

AL-AZAWI, R.; AYESH, A.; OBAIDY, M. A. Generic evaluation framework for games development methodology. In: **2013 Third International Conference on Communications and Information Technology (ICCIT)**. [S.l.: s.n.], 2013. p. 55–60.

AL-BUSAIDI, A. et al. User experiences system design for visually impaired and blind users in oman. In: **2017 6th International Conference on Information and Communication Technology and Accessibility (ICTA)**. [S.l.: s.n.], 2017. p. 1–5. ISSN 2379-4402.

AL-QAIMARI, G.; AGI, L. Improving the position of usability activities in the organisation by reusing existing knowledge and assessing performance. In: . [s.n.], 2012. p. 594–598. Cited By 2. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84904187078&partnerID=40&md5=e304dc18bb3b0ee23699a38b6cf9d252>>.

ALI, I. et al. Cmmi compliant workflow models to establish configuration management integrity in software smes. **Intelligent Automation and Soft Computing**, v. 27, n. 3, p. 605–623, 2021. Cited By 0. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102878953&doi=10.32604/2fiASC.2021.014639&partnerID=40&md5=96f4db7ab5d09cff57eb5a6bcf561998>>.

ALOMAR, N. et al. Usability engineering of agile software project management tools. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 9746, p. 197–208, 2016. Cited By 5. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84977507148&doi=10.1007/978-3-319-40409-7_20&partnerID=40&md5=d3e6a21f5ec1a53c526dd92482eb0621>.

ALVEAR-SUÁREZ, A.; VARA-MESA, J. M.; BOLLATI, V. Impact of the new software development methodologies in puerto rico: An exploratory survey. In: **2020 15th Iberian Conference on Information Systems and Technologies (CISTI)**. [S.l.: s.n.], 2020. p. 1–6. ISSN 2166-0727.

ANDRADE, P. R. Martins de; SADAQUI, S. Improving business decision making based on kpi management system. In: **2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. [S.l.: s.n.], 2017. p. 1280–1285.

ARCOS-MEDINA, G.; MAURICIO, D. The influence of the application of agile practices in software quality based on iso/iec 25010 standard. **International Journal of Information Technologies and Systems Approach**, v. 13, n. 2, p. 27–53, 2020. Cited By 1. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083586287&doi=10.4018%2fIJITSA.2020070102&partnerID=40&md5=85629dab8149a7c063c4c2443a405af6>>.

BARBOSA, S. D. J.; SILVA, B. S. da. **Interacao humano-computador**. [S.l.]: Elsevier, 2010.

BECK, K. **Extreme Programming Explained: Embrace Change**. USA: Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN 0201616416.

BOTZENHARDT, A.; METH, H.; MAEDCHE, A. Cross-functional integration of product management and product design in application software development: Exploration of success factors. In: . [s.n.], 2011. v. 3, p. 2310–2327. Cited By 9. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84884629029&partnerID=40&md5=a8e98c36deaa3261013023f3574cb9b9>>.

BRANHAM, C.; MOXLEY, J.; ROSS, V. My reviewers: Participatory design amp; crowd-sourced usability processes. In: . [s.n.], 2015. Cited By 2. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84961718024&doi=10.1145%2f2775441.2775482&partnerID=40&md5=7b1645609992ec4b6770b7efec7665a4>>.

BROWN, N. et al. Managing technical debt in software-reliant systems. In: . [S.l.: s.n.], 2010. p. 47–52.

CAYOLA, L.; MACÍAS, J. Systematic guidance on usability methods in user-centered software development. **Information and Software Technology**, v. 97, p. 163–175, 2018. Cited By 20. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85041524360&doi=10.1016%2fj.infsof.2018.01.010&partnerID=40&md5=81257a412a2af88efdf7d2f6e7f694b9>>.

CHADLI, S. Y. et al. Software project management tools in global software development: a systematic mapping study. **SpringerPlus**, v. 5, 12 2016.

CHAKRAVARTY, K.; SINGH, J. A study of quality metrics in agile software development. **Advances in Intelligent Systems and Computing**, v. 1311 AISC, p. 255–266, 2021. Cited By 0. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85104812916&doi=10.1007%2f978-981-33-4859-2_26&partnerID=40&md5=2614da21f84a7807e18a86566691cadd>.

CONGER, S. Software development life cycles and methodologies: Fixing the old and adopting the new. **International Journal of Information Technologies and Systems Approach**, v. 4, n. 1, p. 1–22, 2011. Cited By 5. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85001686703&doi=10.4018%2fjitsa.2011010101&partnerID=40&md5=aa7e697d9998eae19289f24c0d38558>>.

CRISAN, A.; GARDY, J.; MUNZNER, T. On regulatory and organizational constraints in visualization design and evaluation. In: . [s.n.], 2016. v. 24-October-2016, p. 1–9. Cited By 6. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84995476581&doi=10.1145%2f2993901.2993911&partnerID=40&md5=0eb557ae849719fae493c0fc3fdac872>>.

CUNHA, R. et al. How do newcomers learn work process in global software development (gsd)?: A survey study from the perspective of newly project leaders. In: . [s.n.], 2020. p. 71–75. Cited By 0. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85093120594&doi=10.1145%2f3372787.3389302&partnerID=40&md5=6ceedabfe2f9bd114c65edfae361aa3c>>.

DERAMAN, A.; SALMAN, F. Managing usability evaluation practices in agile development environments. **International Journal of Electrical and Computer Engineering**, v. 9, n. 2, p. 1288–1297, 2019. Cited By 8. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059987128&doi=10.11591%2fijece.v9i2.pp.1288-1297&partnerID=40&md5=929420a13968e793432a6dab3fe6fe04>>.

DILORENZO, E. et al. Enabling the reuse of software development assets through a taxonomy for user stories. **IEEE Access**, v. 8, p. 107285–107300, 2020. ISSN 2169-3536.

DROBKA, J.; NOFTZ, D.; RAGHU, R. Piloting xp on four mission-critical projects. **IEEE Software**, v. 21, n. 6, p. 70–75, 2004.

EL-NAJAR, T.; AHMAD, A.; ALKANDARI, M. Client communication: A major issue in agile development. **International Journal of Software Engineering and its Applications**, v. 10, n. 12, p. 113–130, 2016. Cited By 1. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85009274865&doi=10.14257%2fijseia.2016.10.12.10&partnerID=40&md5=945b4236323406bb2630952e62d963a3>>.

ELSEVIER. **Scopus: O maior banco de dados da literatura revisada por pares**. 2022. Available from Internet: <<https://www.elsevier.com/pt-br/solutions/scopus>>.

ESPINOZA, J. et al. Usability in the development of a project management software reinforced with machine learning. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 11584 LNCS, p. 361–378, 2019. Cited By 0. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85069839484&doi=10.1007%2f978-3-030-23541-3_26&partnerID=40&md5=197a065c147238d2ee19533d09d97758>.

EVANS, I. et al. Test tools: an illusion of usability? In: **2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)**. [S.l.: s.n.], 2020. p. 392–397.

EVANS, I. et al. Test tools: An illusion of usability? In: . [s.n.], 2020. p. 392–397. Cited By 3. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091782851&doi=10.1109%2fICSTW50294.2020.00070&partnerID=40&md5=b4582b3695504366b9c6639aa7657038>>.

FAGERHOLM, F.; MÜNCH, J. Developer experience: Concept and definition. In: **2012 International Conference on Software and System Process (ICSSP)**. [S.l.: s.n.], 2012. p. 73–77.

FELFERNIG, A. et al. Towards utility-based prioritization of requirements in open source environments. In: **2018 IEEE 26th International Requirements Engineering Conference (RE)**. [S.l.: s.n.], 2018. p. 406–411. ISSN 2332-6441.

FOUAD, A. et al. Embedding requirements within model-driven architecture. **Software Quality Journal**, v. 19, n. 2, p. 411–430, 2011. Cited By 11. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-79952815851&doi=10.1007%2fs11219-010-9122-7&partnerID=40&md5=3c5c2df93c7e35592779961292332d53>>.

GLAVINIC, V.; LJUBIC, S.; KUKEC, M. Supporting universal usability of mobile software: Touchscreen usability meta-test. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 6767 LNCS, n. PART 3, p. 26–35, 2011. Cited By 1. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-79960319282&doi=10.1007%2f978-3-642-21666-4_4&partnerID=40&md5=84600710a7f501e8003725e557867622>.

GONÇALVES, J.; SANTOS, C. Polvo - software for prototyping of low-fidelity interfaces in agile development. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 6761 LNCS, n. PART 1, p. 63–71, 2011. Cited By 7. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-79960306751&doi=10.1007%2f978-3-642-21602-2_7&partnerID=40&md5=beebaf13d4ae913883a256c16424ab2a>.

HACALOGLU, T.; DEMIRORS, O. Measureability of functional size in agile software projects: Multiple case studies with cosmic fsm. In: . [s.n.], 2019. p. 204–211. Cited By 4. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85075973406&doi=10.1109%2fSEAA.2019.00041&partnerID=40&md5=499fbbdadabceb6df85635c6d38aad03>>.

HEITZENRATER, C.; SIMPSON, A. Misuse, abuse, and reuse: Economic utility functions for characterising security requirements. In: . [s.n.], 2016. p. 572–581. Cited By 3. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85009069587&doi=10.1109%2fARES.2016.90&partnerID=40&md5=09999ec857d3985b2a559a1b899668fd>>.

HOSSAIN, S. et al. Requirements re-usability in global software development: A systematic mapping study. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 12252 LNCS, p. 960–974, 2020. Cited By 0. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85092257400&doi=10.1007%2f978-3-030-58811-3_68&partnerID=40&md5=02f0d0d49b94efa10d7b94e8583a3503>.

IEEE. 2022. Available from Internet: <<https://ieeexplore.ieee.org/Xplore/home.jsp/>>.

IMRAN, M. M.; CIBOROWSKA, A.; DAMEVSKI, K. Automatically selecting follow-up questions for deficient bug reports. In: **2021 IEEE/ACM 18th International**

Conference on Mining Software Repositories (MSR). [S.l.: s.n.], 2021. p. 167–178. ISSN 2574-3864.

JHA, P.; PATNAIK, K. S. Empirical bug estimation model: Replicating ideas with analysis and implication of bugs in ck metric suite. **International Journal of Software Engineering and its Applications**, v. 9, n. 12, p. 267–284, 2015. Cited By 0. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84954405849&doi=10.14257%2fijseia.2015.9.12.24&partnerID=40&md5=bf934a4656dbfbedabf8cd3b24b3743a>>.

JUÁREZ-RAMÍREZ, R. et al. Engineering the development process for user interfaces: Toward improving usability of mobile applications. **Communications in Computer and Information Science**, v. 167 CCIS, n. PART 2, p. 65–79, 2011. Cited By 2. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-79960045446&doi=10.1007%2f978-3-642-22027-2_7&partnerID=40&md5=aaa65abe7f1a0c1e5d7eb0d72c19bab8>.

KALINOWSKI, M. et al. Towards lean r amp;d: An agile research and development approach for digital transformation. In: **2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**. [S.l.: s.n.], 2020. p. 132–136.

KIKITAMARA, S.; NOVIYANTI, A. A. A conceptual model of user experience in scrum practice. In: **2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)**. [S.l.: s.n.], 2018. p. 581–586.

LACHEINER, H.; RAMLER, R. Application lifecycle management as infrastructure for software process improvement and evolution: Experience and insights from industry. In: . [s.n.], 2011. p. 286–293. Cited By 18. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-82955234110&doi=10.1109%2fSEAA.2011.51&partnerID=40&md5=9e2c5e5faf144c62119d3f4656d774d4>>.

LAGE, L. D. F. et al. Usability technical debt in software projects: A multi-case study. In: . [s.n.], 2019. v. 2019-Septemer. Cited By 2. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074277075&doi=10.1109%2fESEM.2019.8870180&partnerID=40&md5=7a97ae24771a9610a1e0592ce0be482c>>.

LANNA, M.; AMYOT, D. Spotting the difference. **Software - Practice and Experience**, v. 41, n. 6, p. 607–626, 2011. Cited By 6. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-79953793992&doi=10.1002%2fspe.1013&partnerID=40&md5=629a3e639d36f9095561c38beda2f4e2>>.

LAUESEN, S. It project failures, causes and cures. **IEEE Access**, v. 8, p. 72059–72067, 2020. ISSN 2169-3536.

LEWIS, J. **Fundamentals of Project Management, Third Edition**. USA: Amacom, 2006. ISBN 9780814408797.

LIMA, F. et al. Identifying difficulties in the knowledge transfer process in global software: A qualitative analysis. In: . [s.n.], 2020. p. 84–90. Cited By 0. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85099586298&partnerID=40&md5=10a41819ab156e0467a0ea2986689d04>>.

LINDHOLM, C. Involving user perspective in a software risk management process. **Journal of Software: Evolution and Process**, v. 27, n. 12, p. 953–975, 2015. Cited By 2. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84955706910&doi=10.1002%2fsmr.1753&partnerID=40&md5=f7a167d527a6522e87a13028dd2254e0>>.

LINDHOLM, C.; HOST, M. Introducing usability testing in the risk management process in software development. In: . [s.n.], 2013. p. 5–11. Cited By 2. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84885203274&doi=10.1109%2fSEHC.2013.6602470&partnerID=40&md5=ba432a6baa9751e3b5757a30b20f48d0>>.

LINEHAN, M. H. et al. Controlled english language for production and event processing rules. In: **Proceedings of the 5th ACM International Conference on Distributed Event-Based System**. New York, NY, USA: Association for Computing Machinery, 2011. (DEBS '11), p. 149–158. ISBN 9781450304238. Available from Internet: <<https://doi.org/10.1145/2002259.2002281>>.

LJUNGBERG, A. et al. Case study on data-driven deployment of program analysis on an open tools stack. In: **2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)**. [S.l.: s.n.], 2021. p. 208–217.

MAIER, D. et al. Knowledge management for systems biology a general and visually driven framework applied to translational medicine. **BMC Systems Biology**, v. 5, 2011. Cited By 45. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-79952201056&doi=10.1186%2f1752-0509-5-38&partnerID=40&md5=1d30e623641b4037c7d5f1f0c875111b>>.

MAJEED, B. et al. Comparative study of open source automation testing tools: Selenium, katalon studio amp; test project. In: **2021 International Conference on Innovative Computing (ICIC)**. [S.l.: s.n.], 2021. p. 1–6.

MCCALDEN, A. Usability-driven evolution of a space instrument. In: . [s.n.], 2012. v. 8449. Cited By 0. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84871691760&doi=10.1117%2f12.925319&partnerID=40&md5=2a0ac7367bc16960d126eab425469533>>.

MISHRA, A.; MISHRA, D. Software project management tools: A brief comparative view. **ACM SIGSOFT Software Engineering Notes**, v. 38, 05 2013.

MOORTHY, J. S.; IBRAHIM, S. B.; MAHRIN, M. Identifying usability risk: A survey study. In: . [s.n.], 2014. p. 148–153. Cited By 3. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84929379059&doi=10.1109%2fMySec.2014.6986005&partnerID=40&md5=d03afd68a6e99608fe91c29ac801c123>>.

MORENO, W.; AFONSO, P.; COSTA, H. Challenges and solutions of project management in distributed software development. In: **2019 XLV Latin American Computing Conference (CLEI)**. [S.l.: s.n.], 2019. p. 1–10.

MOROCHO, V. et al. Implementation of a basic risk guide for interactive digital terrestrial television using learning objects. **Advances in Intelligent Systems and Computing**, v. 1326 AISC, p. 202–215, 2021. Cited By 0.

Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85104881242&doi=10.1007%2f978-3-030-68080-0_15&partnerID=40&md5=77f435a64ddd83ce1a9104c13eebd900>.

MOUSAEI, M.; GANDOMANI, T. A new project risk management model based on scrum framework and prince2 methodology. **International Journal of Advanced Computer Science and Applications**, v. 9, n. 4, p. 442–449, 2018. Cited By 8. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85049512832&doi=10.14569%2fIJACSA.2018.090461&partnerID=40&md5=2629d9c183a78b24dbb12a86b65a728c>>.

MOYO, S.; MNKANDLA, E. A novel lightweight solo software development methodology with optimum security practices. **IEEE Access**, v. 8, p. 33735–33747, 2020. ISSN 2169-3536.

MYLLY, S.; RAJANEN, M.; IIVARI, N. Usable usability heuristics for game developers. In: . [s.n.], 2019. Cited By 0. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091282398&partnerID=40&md5=fe3a2d0c2cd5f2776327c393917037eb>>.

MÄKIAHO, P.; VARTIAINEN, K.; PORANEN, T. Mmt: A tool for observing metrics in software projects. **International Journal of Human Capital and Information Technology Professionals**, v. 8, n. 4, p. 27–37, 2017. Cited By 4. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85028732529&doi=10.4018%2fIJHCITP.2017100103&partnerID=40&md5=686959836233c91a602d79ecec23340e>>.

NUNES, N.; CONSTANTINE, L.; KAZMAN, R. iucp: Estimating interactive-software project size with enhanced use-case points. **IEEE Software**, v. 28, n. 4, p. 64–73, July 2011. ISSN 1937-4194.

OLIVEIRA, J. et al. Is scrum useful to mitigate project's risks in real business contexts? **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 9790, p. 422–437, 2016. Cited By 0. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84978219395&doi=10.1007%2f978-3-319-42092-9_33&partnerID=40&md5=f9c867e3e249920903557db82819ce92>.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and Software Technology**, v. 64, p. 1–18, 2015. ISSN 0950-5849. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/S0950584915000646>>.

PIERCE, R. User assistance for complex systems. In: . [s.n.], 2012. p. 9–16. Cited By 0. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84869008460&doi=10.1145%2f2379057.2379060&partnerID=40&md5=c534a9f56c2df8a68c4de385b4604fc8>>.

PMI (Ed.). **A Guide to the Project Management Body of Knowledge (PMBOK Guide)**. 5. ed. Newtown Square, PA: Project Management Institute, 2013. ISBN 978-1-935589-67-9.

PORUBÄN, J.; BACIKOVÁ, M. Live simulation of software processes in programming courses. In: **2016 International Conference on Emerging eLearning Technologies and Applications (ICETA)**. [S.l.: s.n.], 2016. p. 283–288.

PRABHU, N. et al. Impact of knowledge management on offshore software development: An exploratory study. In: . [s.n.], 2011. p. 121–128. Cited By 6. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84872783344&doi=10.1109%2fICoAC.2011.6165160&partnerID=40&md5=7bc3e2666158fed4bee798835dcf7a13>>.

PRESSMAN, R. **Software Engineering: A Practitioner's Approach**. sixth. [S.l.]: McGraw-Hill, 2005.

RAHMAT, A.; HANIFIAH, N. Usability testing in kanban agile process for club management system. In: . [s.n.], 2020. Cited By 0. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85101573616&doi=10.1109%2fICIDM51048.2020.9339668&partnerID=40&md5=73d063b30c46fe8ac1f04da038f4a11f>>.

RAMAKRISHNAN, L.; GUNTER, D. Ten principles for creating usable software for science. In: . [s.n.], 2017. p. 210–218. Cited By 4. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85043753708&doi=10.1109%2feScience.2017.34&partnerID=40&md5=3b0a320305de36cbbb7a365ad7a78ee6>>.

REYNOLDS, S. et al. User-centered design of visualizations for software vulnerability reports. In: . [s.n.], 2021. p. 68–78. Cited By 0. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85123849896&doi=10.1109%2fVizSec53666.2021.00013&partnerID=40&md5=4767ae230b6b7c3fbb4e1d7d2c1d3f38>>.

ROSENBERGER, P.; TICK, J. Suitability of pmbok 6th edition for agile-developed it projects. In: **2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)**. [S.l.: s.n.], 2018. p. 000241–000246. ISSN 2471-9269.

RUIZ, M.; HASSELMAN, B. Can we design software as we talk?: A research idea. **Lecture Notes in Business Information Processing**, v. 387 LNBIP, p. 327–334, 2020. Cited By 0. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85086300855&doi=10.1007%2f978-3-030-49418-6_22&partnerID=40&md5=1024014331b4c6fff6551948b4fe0b00>.

SALAMEA, M. J. et al. Systematic literature study on estimation and prioritization of quality requirements in software development. In: **2020 15th Iberian Conference on Information Systems and Technologies (CISTI)**. [S.l.: s.n.], 2020. p. 1–7. ISSN 2166-0727.

SARINHO, V. Masters of the process: A board game proposal for teaching software management and software development process. In: . [s.n.], 2019. p. 532–536. Cited By 2. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85073201311&doi=10.1145%2f3350768.3352459&partnerID=40&md5=9067cd6964fbc7518beeaf5416986145>>.

SCHWABER, K.; SUTHERLAND, J. *The scrum guide*. 2014.

SETIADI, T.; PREMAPASHA, S. B. Scrum implementation for online transaction processing (oltp) in hospital management. In: **2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA)**. [S.l.: s.n.], 2018. p. 1–6.

SHARMA, S.; KUMAR, D. Agile release planning using natural language processing algorithm. In: **2019 Amity International Conference on Artificial Intelligence (AICAI)**. [S.l.: s.n.], 2019. p. 934–938.

SILVA, F. Q. da et al. Challenges and solutions in distributed software development project management: A systematic literature review. In: **2010 5th IEEE International Conference on Global Software Engineering**. [S.l.: s.n.], 2010. p. 87–96.

SLHOUB, K.; NEMBHARD, F.; CARVALHO, M. A metrics tracking program for promoting high-quality software development. In: **2019 SoutheastCon**. [S.l.: s.n.], 2019. p. 1–8. ISSN 1558-058X.

SMARADOTTIR, B. The steps of user-centered design in health information technology development: Recommendations from a phd research study. In: . [s.n.], 2017. p. 116–121. Cited By 8. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85017298928&doi=10.1109%2fCSCI.2016.0029&partnerID=40&md5=4c01ee2a2fb9a3891818b8e8efb3daed>>.

SMIRNOV, O. et al. Revizor: A data-driven approach to automate frequent code changes based on graph matching. In: **2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)**. [S.l.: s.n.], 2021. p. 1242–1246. ISSN 2643-1572.

SOMMERVILLE, I. **Software Engineering**. 9. ed. Harlow, England: Addison-Wesley, 2010. ISBN 978-0-13-703515-1.

SPENCER, D.; ZIMMERMAN, A.; ABRAMSON, D. Special theme: Project management in e-science: Challenges and opportunities. **Computer Supported Cooperative Work**, v. 20, n. 3, p. 155–163, 2011. Cited By 18. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-80052965391&doi=10.1007%2fs10606-011-9140-4&partnerID=40&md5=267d9e2e46aaf64b99f40038503ca10f>>.

SPICHKOVA, M. et al. Towards modelling and implementation of reliability and usability features for research-oriented cloud computing platforms. **Communications in Computer and Information Science**, v. 703, p. 158–178, 2016. Cited By 5. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85019088513&doi=10.1007%2f978-3-319-56390-9_8&partnerID=40&md5=8259bc52eebe37ff10d20657b1a2371f>.

TABASSUM, M. R. et al. Determining interdependency among non-functional requirements to reduce conflict. In: **2014 International Conference on Informatics, Electronics Vision (ICIEV)**. [S.l.: s.n.], 2014. p. 1–6.

TAO, R. et al. The usability evaluation method of e-learning platform based on fuzzy comprehensive evaluation. **Communications in Computer and Information Science**, v. 1438, p. 292–304, 2021. Cited By 0. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85019088513&doi=10.1007%2f978-3-319-56390-9_8&partnerID=40&md5=8259bc52eebe37ff10d20657b1a2371f>.

0-85113486277&doi=10.1007%2f978-3-030-81635-3_24&partnerID=40&md5=d948ca982b717cc0723587e41cadae05>.

TÉNYI, B. et al. Experience with an integrated risk management process in the medical regulatory environment. **Communications in Computer and Information Science**, v. 364 CCIS, p. 345–353, 2013. Cited By 0. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84904605952&doi=10.1007%2f978-3-642-39179-8_31&partnerID=40&md5=64e367b8cb0ccae86477f7d6f8b80a0b>.

ULLAH, S.; IQBAL, M.; KHAN, A. M. A survey on issues in non-functional requirements elicitation. In: **International Conference on Computer Networks and Information Technology**. [S.l.: s.n.], 2011. p. 333–340. ISSN 2223-6317.

VIIKKI, K.; PALVIAINEN, J. Integrating human-centered design into software development: An action research study in the automation industry. In: **2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications**. [S.l.: s.n.], 2011. p. 313–320. ISSN 2376-9505.

WANDERLEY, M. et al. Proposal of risk management metrics for multiple project software development. In: . [s.n.], 2015. v. 64, p. 1001–1009. Cited By 13. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84962897573&doi=10.1016%2fj.procs.2015.08.619&partnerID=40&md5=9d9df308155d0d0d564733e758688d61>>.

WILKIE, F. et al. The value of software sizing. **Information and Software Technology**, v. 53, n. 11, p. 1236–1249, 2011. Cited By 23. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-80051669402&doi=10.1016%2fj.infsof.2011.05.008&partnerID=40&md5=04e0743d1ada12169024fd25590b5b4a>>.

WILLIAMS, L. Seven habits of highly impactful empirical software engineers. In: **Proceedings of the 7th International Conference on Predictive Models in Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2011. (Promise '11). ISBN 9781450307093. Available from Internet: <<https://doi.org/10.1145/2020390.2020391>>.

Z, J. B.; VILLAVICENCIO, M.; IZQUIERDO, E. Evaluation of prototyping tools based on categories and parameters. In: **2016 IEEE Colombian Conference on Communications and Computing (COLCOM)**. [S.l.: s.n.], 2016. p. 1–6.

ZANATTA, A. et al. Barriers faced by newcomers to software-crowdsourcing projects. **IEEE Software**, v. 34, n. 2, p. 37–43, 2017. Cited By 23. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85017134027&doi=10.1109%2fMS.2017.32&partnerID=40&md5=7d00ba334786c244b2485347bfe68d71>>.

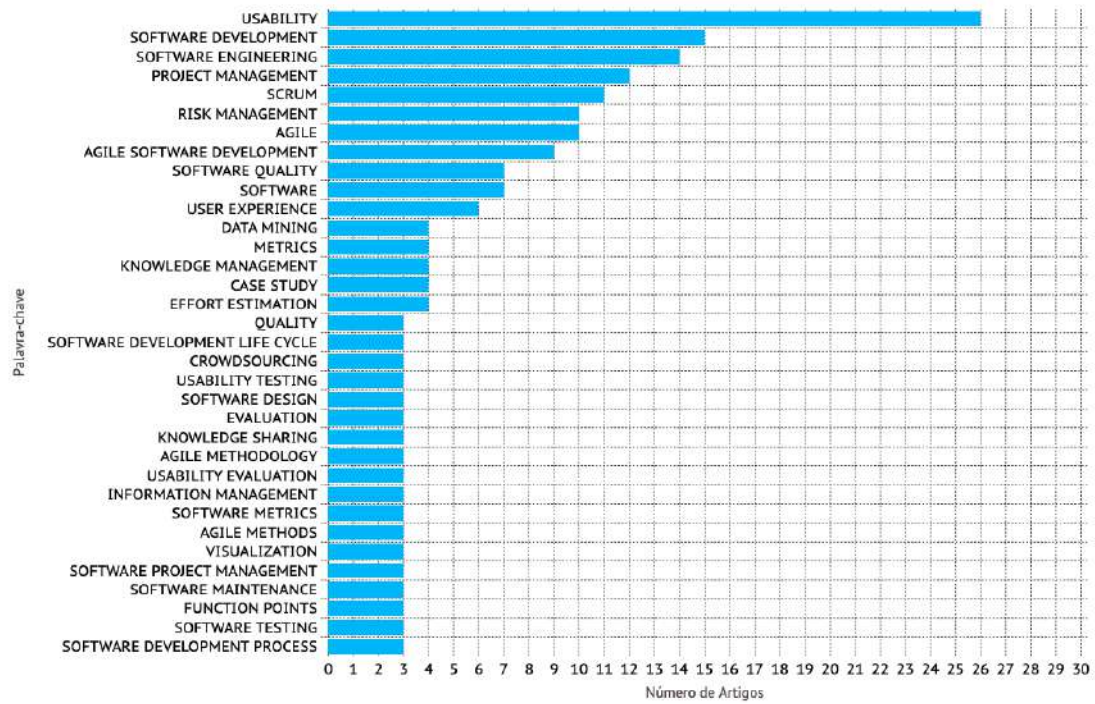
ZAZWORKA, N. et al. A case study on effectively identifying technical debt. In: . [s.n.], 2013. p. 42–47. Cited By 74. Available from Internet: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84877272255&doi=10.1145%2f2460999.2461005&partnerID=40&md5=a7df3f61ef237a907f85f5f9f44c149a>>.

ZHANG, Z. et al. A statistical based resource allocation scheme in cloud. In: **2011 International Conference on Cloud and Service Computing**. [S.l.: s.n.], 2011. p. 266–273.

ZHOU, N.; GIFFORD, W. M.; RATAKONDA, K. Toward a flexible bidding strategy with preference analysis. In: **2013 IEEE International Conference on Services Computing**. [S.l.: s.n.], 2013. p. 619–626.

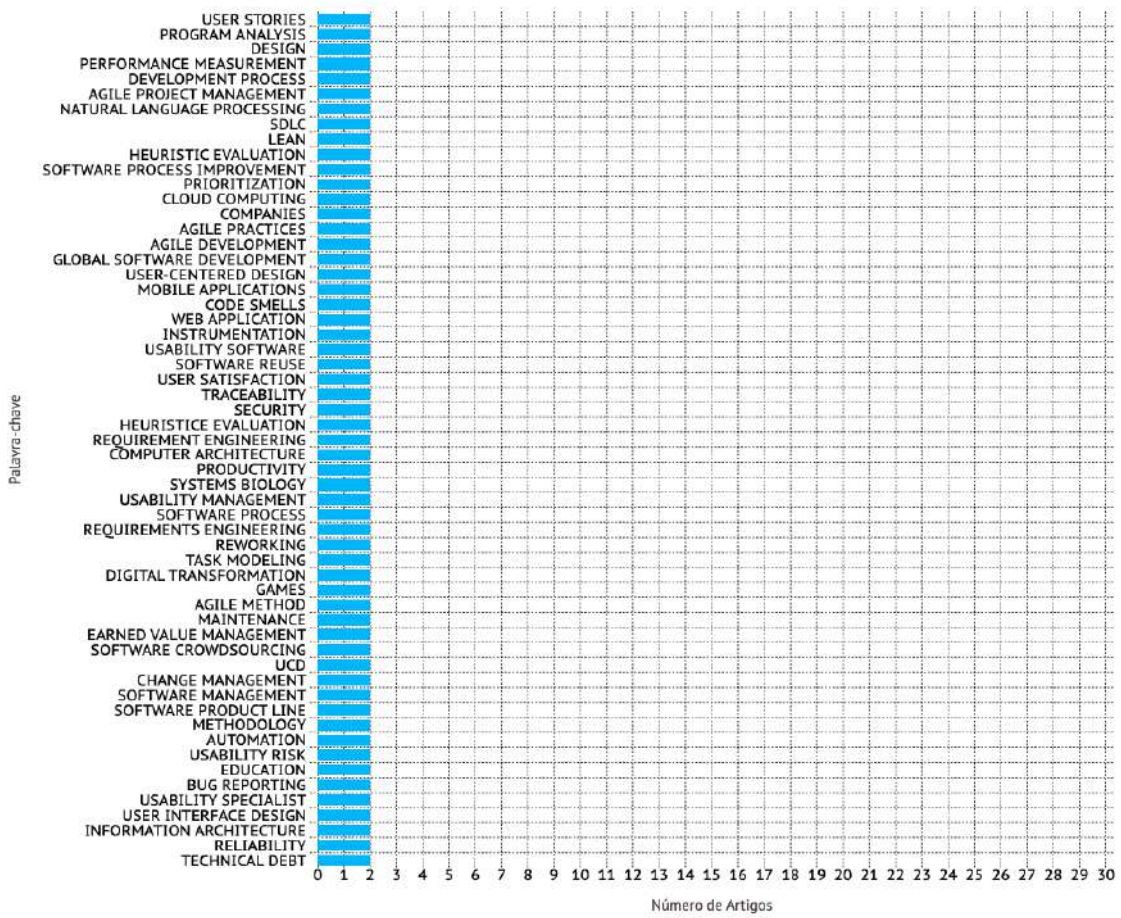
APÊNDICE A — TÓPICOS ENCONTRADOS MAIS DE UMA VEZ NOS ARTIGOS SELECIONADOS

Figura A.1: Tópicos mais encontrados nos artigos selecionados. (1/2)



Fonte: O Autor

Figura A.2: Tópicos mais encontrados nos artigos selecionados. (2/2)



Fonte: O Autor

APÊNDICE B — TABELA DOS ARTIGOS SELECIONADOS

Tabela B.1: Tabela de artigos selecionados (1/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
24	SCOPUS	Evolução de Software	Improving the position of usability activities in the organisation by reusing existing knowledge and assessing performance	Al-Qaimari G., Agi L.	2012	Capability assessment; Knowledge reuse; Metrics; Organizational tasks; Usability
25	SCOPUS	Validação de Software	Cmmi compliant workflow models to establish configuration management integrity in software smes	Ali I., Khan M., Mehmood W., Nisar W., Aslam W., Saleem M.Q., Omer M.K., Shafiq M.	2021	Capability maturity model integration; Software configuration management; Software process improvement; Software SMEs
26	SCOPUS	Projeto e Implementação de Software	Usability engineering of agile software project management tools	Alomar N., Almobarak N., Alkoblan S., Alhozaimy S., Alharbi S.	2016	Agile; Software engineering; Software project management; Usability; User experience
27	SCOPUS	Projeto e Implementação de Software	The influence of the application of agile practices in software quality based on ISO/IEC 25010 standard	Arcos-Medina G., Mauricio D.	2020	Agile practices; Agile software development; Conceptual model; Quality attributes; Software quality

Fonte: Os Autores

Tabela B.2: Tabela de artigos selecionados (2/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
33	SCOPUS	Projeto e Implementação de Software	Cross-functional integration of product management and product design in application software development: Exploration of success factors	Botzenhardt A., Meth H., Maedche A.	2011	Cross-functional integration; IS development; Product design; Product management
34	SCOPUS	Projeto e Implementação de Software	My reviewers: Participatory design & crowd-sourced usability processes	Branham C., Moxley J., Ross V.	2015	Digital writing tools; Project management; Usability; User experience
37	SCOPUS	Projeto e Implementação de Software	Systematic guidance on usability methods in user-centered software development	Cayola L., Macías J.A.	2018	CASE tool; Method recommendation; Software quality; Usability; User-centered development; User-centered project management
38	SCOPUS	Validação de Software	A Study of Quality Metrics in Agile Software Development	Chakravarty K., Singh J.	2021	Agile method; Agile metrics; Software quality metrics
40	SCOPUS	Projeto e Implementação de Software	Software Development Life Cycles and Methodologies: Fixing the Old and Adopting the New	Conger S.	2011	IT Service Management; Methodology; Software Development Life Cycle; Usability; User Centered Design; User Satisfaction

Fonte: Os Autores

Tabela B.3: Tabela de artigos selecionados (3/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
42	SCOPUS	Projeto e Implementação de Software	How do newcomers learn work process in global software development (GSD)?: A survey study from the perspective of newly project leaders	Cunha R.F.V., Souza F.B.S., Lima F.O., Bonifacio B.A.	2020	knowledge management; knowledge transfer; newcomers; newcomers in GSD; onboarding and socialization of newcomers; wiki
44	SCOPUS	Validação de Software	AGILUS: A method for integrating usability evaluations on agile software development	de Freitas R.C., Rodrigues L.A., Jr., da Cunha A.M.	2016	Agile development; Integration of HCI and agile methods; Scrum framework; Usability evaluation
47	SCOPUS	Validação de Software	Managing usability evaluation practices in agile development environments	Deraman A.B., Salman F.A.	2019	Agile software development; Usability; Usability evaluation; Usability management
48	SCOPUS	Validação de Software	A change management model and its application in software development projects	Efe P., Demirors O.	2019	Change management; Earned Value Management; Performance measurement; Reworking; Software project management
49	SCOPUS	Evolução de Software	Measuring change in software projects through an earned value lens	Efe P., Demirors O., Benetallah B.	2018	Change management; Earned Value Management; Performance Measurement; Reworking; Software project management

Fonte: Os Autores

Tabela B.4: Tabela de artigos selecionados (4/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
50	SCOPUS	Especificação de Software	Client communication: A major issue in agile development	El-Najar T., Ahmad A., Alkandari M.	2016	Agile; Client communication; Communication problem; Requirement engineering; Software development; Waterfall
51	SCOPUS	Projeto e Implementação de Software	An exploratory study on usage of process mining in agile software development	Erdem S., Demirörs O.	2017	Agile software development; Process conformance checking; Process discovery; Process mining
52	SCOPUS	Projeto e Implementação de Software	Obtaining Data from the Third-Party Systems for Software Development Process Analysis	Ergasheva S., Gudkov M., Kruglov A., Succi G., Vasques X., Zagidullina A.	2020	Development tools; management utilities; productivity; software development tracking
53	SCOPUS	Projeto e Implementação de Software	Usability in the Development of a Project Management Software Reinforced with Machine Learning	Espinoza J., Loarte P., Paz F., Flores L.	2019	Design; Nielsen usability guidelines; PM4r; Project management; Software development; Usability
58	SCOPUS	Especificação de Software	Embedding requirements within Model-Driven Architecture	Fouad A., Phalp K., Kanyaru J.M., Jeary S.	2011	Business process modelling notation; Computation-Independent Model; Model-Driven Architecture; Requirements Engineering specification; Use Cases

Fonte: Os Autores

Tabela B.5: Tabela de artigos selecionados (5/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
59	SCOPUS	Projeto e Implementação de Software	SecSDM: A usable tool to support it undergraduate students in secure software development	Futcher L., Von Solms R.	2012	Risk management; SecSDM; Secure software development; User satisfaction
60	SCOPUS	Projeto e Implementação de Software	Reconciliation of scrum and the project management process of the ISO/IEC 29110 standard-Entry profile—an experimental evaluation through usability measures	Galvan-Cruz S., Mora M., Laporte C.Y., Duran-Limon H.	2021	Agility-rigor reconciliation problem; Electronic Process Guide; ISO/IEC 29110 standard-Entry profile; Project Management process; Scrum; Within-subjects experiment design
62	SCOPUS	Projeto e Implementação de Software	FAMAP: A framework for developing m-health apps	García-Magariño I., Gonzalez Bedia M., Palacios-Navarro G.	2018	App; m-health; Mobile application; Multi-agent system; Software development; Well-being
63	SCOPUS	Projeto e Implementação de Software	On the socially aware development of self-Adaptive ubiquitous computing applications	Geihs K., Niemczyk S., Roßnagel A., Witsch A.	2014	Acm ccs: Software and its engineering-software creation and management-designing software; Human-centered computing-Ubiquitous and mobile computing; Self-Adaptive applications; Socially aware computing; Software development process
64	SCOPUS	Projeto e Implementação de Software	Mining the conceptual model of open source CMS using a reverse engineering approach	Gkantouna V., Sioutas S., Sourla G., Tsakalidis A., Tzimas G.	2013	Content management systems; Data mining; Joomla!; Model-driven development; Reverse engineering; Web application; WebML

Fonte: Os Autores

Tabela B.6: Tabela de artigos selecionados (6/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
65	SCOPUS	Projeto e Implementação de Software	Supporting universal usability of mobile software: Touchscreen usability meta-test	Glavinic V., Ljubic S., Kukec M.	2011	mobile software; touchscreen interaction; universal usability; usability testing
66	SCOPUS	Projeto e Implementação de Software	POLVO - Software for prototyping of low-fidelity interfaces in agile development	Gonçalves J., Santos C.	2011	Agile Methods; Interaction Design; Low-fidelity Prototype; Scrum
69	SCOPUS	Validação de Software	Applying source code analysis techniques: A case study for a large mission-critical software system	Haralambiev H., Boychev S., Lilov D., Kraichev K.	2011	case study; implementation quality; software maintenance; software metrics; source code analysis
70	SCOPUS	Projeto e Implementação de Software	Misuse, abuse, and reuse: Economic utility functions for characterising security requirements	Heitzenrater C., Simpson A.	2016	Software engineering; Development process; Economic factors; Modelling tools; Secure software engineering; Security community; Security requirements; Surround systems; Utility functions; Software design

Fonte: Os Autores

Tabela B.7: Tabela de artigos selecionados (7/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
73	SCOPUS	Projeto e Implementação de Software	Systematic literature reviews in agile software development: A tertiary study	Hoda R., Salleh N., Grundy J., Tee H.M.	2017	Agile software development; Mapping study; Systematic literature reviews; Tertiary study
74	SCOPUS	Especificação de Software	Requirements Re-usability in Global Software Development: A Systematic Mapping Study	Hossain S.S., Arafat Y., Amin T., Bhuiyan T.	2020	Agile; Distributed software development; Global Software Development (GSD); Large-scale; Requirement re-usability; Systematic mapping study
75	SCOPUS	Projeto e Implementação de Software	Assessing product development agility	Houston D.X., Rosemergy S.W.	2016	Agile method; Agile practice; Global software development; Information system development; Software development
76	SCOPUS	Validação de Software	Usability specialists as boundary spanners - An appraisal of usability specialists' work in multiparty distributed open source software development effort	Iivari N.	2013	boundary object; boundary spanning; children; Usability specialist
77	SCOPUS	Especificação de Software	University of Santo Tomas Department of Filipino Virtual Front Desk	J. Domingo M., L. Pua L.C., S. Pua W.P., S. Talag H.C.	2021	Document Management System; Virtual Helpdesk

Fonte: Os Autores

Tabela B.8: Tabela de artigos selecionados (8/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
79	SCOPUS	Validação de Software	Empirical bug estimation model: Replicating ideas with analysis and implication of bugs in CK metric suite	Jha P., Sridhar Patnaik K.	2015	Bugs; C-K suite; Metrics; Multicollinearity; Object-oriented approach; PCA
80	SCOPUS	Projeto e Implementação de Software	Engineering the development process for user interfaces: Toward improving usability of mobile applications	Juárez-Ramírez R., Licea G., Barriba I., Izquierdo V., Ángeles A.	2011	mobile applications; task and user analysis; Usability
81	SCOPUS	Validação de Software	Measuring software testing efficiency using two-way assessment technique	Kapur P.K., Singh G., Sachdeva N., Tickoo A.	2015	Analytical Hierarchical Process; Software Testing; Two-Way Assessment; Utility Value
83	SCOPUS	Projeto e Implementação de Software	Service-oriented methodology for systems development	Keith M., Demirkan H., Goul M.	2013	design science; modularization; project management; service orientation
86	SCOPUS	Projeto e Implementação de Software	Development of a rule-based system to enhance the data consistency and usability of COBie datasheets	Kumar V., Teo A.L.E.	2021	building information modeling; COBie; COBie data drops; information management

Fonte: Os Autores

Tabela B.9: Tabela de artigos selecionados (9/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
88	SCOPUS	Validação de Software	Spotting the difference	Lanna M., Amyot D.	2011	Eclipse; file comparison; graphical user interface (GUI); Java; usability
90	SCOPUS	Projeto e Implementação de Software	A dependency-sharing tool for global software engineering	Lee D., Milewski A.E., Rosca D.	2013	dependencies; maintenance; traceability
92	SCOPUS	Evolução de Software	Identifying difficulties in the knowledge transfer process in global software: A qualitative analysis	Lima F.O., Mattos V.C.S., Cunha R.F.V., Teixeira A.S., Ayres M.S., Pascareli L.H., Pires F.B., Bonifácio B.A.	2020	Grounded Theory; Knowledge Management; Knowledge Sharing; Qualitative Analysis
93	SCOPUS	Projeto e Implementação de Software	Involving user perspective in a software risk management process	Lindholm C.	2015	medical device development; risk management; software development; usability testing; user perspective
97	SCOPUS	Projeto e Implementação de Software	Enhancing decision-making in user-centered web development: a methodology for card-sorting analysis	Macías J.A., Culén A.L.	2021	Card sorting; Human-computer interaction; Information architecture; Quantitative analysis; User experience; User-centered web application development

Fonte: Os Autores

Tabela B.10: Tabela de artigos selecionados (10/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
98	SCOPUS	Projeto e Implementação de Software	Knowledge management for systems biology a general and visually driven framework applied to translational medicine	Maier D., Kalus W., Wolff M., Kalko S.G., Roca J., Marin de Mas I., Turan N., Cascante M., Falciani F., Hernandez M., Villà-Freixa J., Losko S.	2011	article; chronic obstructive lung disease; computer program; data mining; human; information processing; knowledge base; methodology; pathophysiology; systems biology; Data Collection; Data Mining; Humans; Knowledge Bases; Pulmonary Disease, Chronic Obstructive; Software; Systems Biology
99	SCOPUS	Validação de Software	MMT: A tool for observing metrics in software projects	Mäkiäho P., Vartiainen K., Poranen T.	2017	Design Science Research; Evaluation; Metrics; Monitoring; Project Management; Software Development; Student Courses; Teaching
102	SCOPUS	Projeto e Implementação de Software	A development process for usable large scale interactive critical systems: Application to satellite ground segments	Martinie C., Palanque P., Navarre D., Barboni E.	2012	formal methods; model-based design; safety management; Software engineering; task modeling; training
103	SCOPUS	Evolução de Software	Usability-driven evolution of a space instrument	McCalden A.	2012	Instrumentation; Interface; Language; Persona; Simulation; Space; System engineering; Usability
104	SCOPUS	Projeto e Implementação de Software	An agile implementation within a medical device software organisation	McHugh M., McCaffery F., Coady G.	2014	Agile; AV-Model; Hybrid; IEC 62304; Medical; SDLC

Fonte: Os Autores

Tabela B.11: Tabela de artigos selecionados (11/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
105	SCOPUS	Especificação de Software	Analyzing the requirements to implement a data analysis model for software process improvement	Mejía J., Íñiguez F., Muñoz M.	2019	big data; data analysis; information management; knowledge management; Software Process Improvement; software repositories
108	SCOPUS	Validação de Software	Implementation of a Basic Risk Guide for Interactive Digital Terrestrial Television Using Learning Objects	Morocho V., Cárdenas P., Illescas L., Maza-Cordova J., Achig R.	2021	DTT; Ginga; iDTT; Interactivity; ISDBT-TB; LUA; NCL; Return channel; Set Top Box; Usability
109	SCOPUS	Validação de Software	A new project risk management model based on Scrum framework and Prince2 methodology	Mousaei M., Gandomani T.J.	2018	Agile risk management; Agile software development; Prince2; Risk management; Risk management hybrid model; Scrum
111	SCOPUS	Projeto e Implementação de Software	Usable usability heuristics for game developers	Mylly S., Rajanen M., Iivari N.	2019	Game Development; Games; Heuristic evaluation; Usability
114	SCOPUS	Projeto e Implementação de Software	Is scrum useful to mitigate project's risks in real business contexts?	Oliveira J., Vinhas M., Da Costa F., Nogueira M., Ribeiro P., Machado R.J.	2016	Project management; Project monitoring; Risk; Scrum; Software; Utility

Fonte: Os Autores

Tabela B.12: Tabela de artigos selecionados (12/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
116	SCOPUS	Projeto e Implementação de Software	Design thinking integrated in agile software development: A systematic literature review	Pereira J.C., Russo R.D.F.S.M.	2018	Agile Methodology; Design Thinking; Information Systems; Project Management; Software
117	SCOPUS	Projeto e Implementação de Software	User assistance for complex systems	Pierce R.	2012	Best practices; Computer documentation; Development process; Information architecture; Information development; Personas; Software development; Software documentation; Task modeling; Task-based design; Usability; User assistance; User technologies
118	SCOPUS	Validação de Software	Vision 2020: The future of software quality management and impacts on global user acceptance	Poston R., Calvert A.	2015	Global user acceptance; Software development life cycle; Software quality management; User acceptance testing
120	SCOPUS	Projeto e Implementação de Software	Ontology based risks management model for agile software development	Prakash B., Viswanathan V.	2018	Agile software development; Natural language processing; Ontology; Risk management; Software quality
121	SCOPUS	Projeto e Implementação de Software	ARP-GWO: an efficient approach for prioritization of risks in agile software development	Prakash B., Viswanathan V.	2021	Agile software development; Meta-heuristic algorithms; Risk management; Risk prioritization; Swarm intelligence techniques

Fonte: Os Autores

Tabela B.13: Tabela de artigos selecionados (13/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
123	SCOPUS	Projeto e Implementação de Software	Non-response, social exclusion, and false acceptance: gatekeeping tactics and usability work in free-libre open source software development	Rajanen M., Iivari N., Lanamäki A.	2015	Boundary management; FLOSS; Gatekeeping; Usability specialist
124	SCOPUS	Validação de Software	An Empirical Investigation into Industrial Use of Software Metrics Programs	Ram P., Rodríguez P., Oivo M., Bagnato A., Abherve A., Choraś M., Kozik R.	2020	Decision-making; Metrics program; Process metrics
127	SCOPUS	Especificação de Software	Can we design software as we talk?: A research idea	Ruiz M., Hasselman B.	2020	Digital transformation; Requirements engineering; Software development life cycle; User stories generation
128	SCOPUS	Projeto e Implementação de Software	A practitioner perspective on integrating agile and user centred design	Salah D., Paige R., Cairns P.	2014	Agile software development processes; Agile user centred design integration; User centred design
129	SCOPUS	Validação de Software	On implementing usability evaluation activities within agile environment: A plan based-Process	Salman F.A., Deraman A., Jalil M.A., Fahmy S.	2017	Agile development process; Agile planning; Usability evaluation; Usability management

Fonte: Os Autores

Tabela B.14: Tabela de artigos selecionados (14/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
132	SCOPUS	Projeto e Implementação de Software	Managing user experience (UX) design practice: Approaches and considerations	Sharma A., Sharma A.	2017	Design management; Design practice; Design process; Design roles; HCI; Practice setup; Usability practice; User experience design
133	SCOPUS	Validação de Software	Navigating the maze - Journey towards an optimal	Shenvi A.A.	2014	Medical device; Quality management system; Regulations
135	SCOPUS	Projeto e Implementação de Software	The steps of user-centered design in health information technology development: Recommendations from a PhD research study	Smaradottir B.F.	2017	Health Information Technology; Software Development; Usability; User-centered Design
136	SCOPUS	Projeto e Implementação de Software	Special theme: Project Management in e-Science: Challenges and Opportunities	Spencer D., Zimmerman A., Abramson D.	2011	Agile development; Agile management; Collaboratories; Cyberinfrastructure; E-infrastructures; E-Science; Facilitation of e-science teams; Interdisciplinary collaboration; Leadership; Project Management; Software development; Team-building; Usability
137	SCOPUS	Projeto e Implementação de Software	Towards modelling and implementation of reliability and usability features for research-oriented cloud computing platforms	Spichkova M., Schmidt H.W., Yusuf I.I., Thomas I.E., Androulakis S., Meyer G.R.	2016	Cloud computing; Modelling; Reliability; Usability; Visualisation

Fonte: Os Autores

Tabela B.15: Tabela de artigos selecionados (15/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
138	SCOPUS	Projeto e Implementação de Software	Practical application of a user-centered design management strategy	Stevens J., Laferriere K., Jones J.	2015	Agile; HSI; Human factors engineering; Human-system interfaces; Management strategy
139	SCOPUS	Projeto e Implementação de Software	Software development for autonomous and social robotics systems	Sun C., Zhang J., Liu C., King B.C.B., Zhang Y., Galle M., Spichkova M., Simic M.	2019	Anthropomorphic robots; Interactive computer systems; Multimedia services; Multimedia systems; Robotics; Software design; Autonomous humanoid robots; Behavioural assumptions; Historical data; Industrial environments; Physical contacts; Physical interactions; Social robotics; WEB application; Human robot interaction
141	SCOPUS	Validação de Software	The Usability Evaluation Method of E-learning Platform Based on Fuzzy Comprehensive Evaluation	Tao R., Zhu L., Wen Q., Shi Y., Feng X.	2021	Analytic hierarchy process; E-learning platform; Fuzzy comprehensive evaluation; Software usability
142	SCOPUS	Validação de Software	Experience with an Integrated Risk Management Process in the Medical Regulatory Environment	Tényi B., Csík A., Monoki I., Tegzes F.	2013	medical software development; risk management; usability engineering
143	SCOPUS	Especificação de Software	A systematic literature review of use case specifications research	Tiwari S., Gupta A.	2015	Evolution; Guidelines; Quality; Systematic reviews; Use case specifications; Use case templates

Fonte: Os Autores

Tabela B.16: Tabela de artigos selecionados (16/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
144	SCOPUS	Projeto e Implementação de Software	Agile technology development to improve scenario-based learning exercises	Turnbull B., Ormrod D., Moustafa N., Micallef N.	2019	Cyber-security; Online social network; Scenario-based-learning; Social-media; Software-engineering; War-game
145	SCOPUS	Projeto e Implementação de Software	Recommender System for ConFIGuration Management Process of Entrepreneurial Software Designing Firms	Uz Zaman M.W., Hafeez Y., Hussain S., Anwaar H., Yang S., Ali S., Abbasi A.A., Song O.-Y.	2021	ConFIGuration management; Feature selection; Recommender system; Software reuse
148	SCOPUS	Validação de Software	Proposal of Risk Management Metrics for Multiple Project Software Development	Wanderley M., Menezes J., Jr., Gusmão C., Lima F.	2015	metrics; multiple project management; risk management; software engineering
149	SCOPUS	Especificação de Software	The value of software sizing	Wilkie F.G., McChesney I.R., Morrow P., Tuxworth C., Lester N.G.	2011	Empirical software engineering; Function points; NESMA; Project planning; Size metrics; Software size estimation
151	SCOPUS	Validação de Software	Testing driven development of mobile applications using automatic bug management systems	Zaragoza M.G., Kim H.-K., Bae I.-H., Lee J.-H.	2018	Automatic bug management; Mobile applications; Testing driven development

Fonte: Os Autores

Tabela B.17: Tabela de artigos selecionados (17/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
154	SCOPUS	Validação de Software	De-Flake Your Tests: Automatically Locating Root Causes of Flaky Tests in Code at Google	Ziftci C., Cavalcanti D.	2020	Debuggers; Debugging aids; Diagnostics; Flaky tests; Software maintenance; Test management; Tracing
155	IEEE	Evolução de Software	Agile Release Planning Using Natural Language Processing Algorithm	S. Sharma; D. Kumar	2019	Natural Language Processing; Algorithm; Software Engineering; Agile Methodology; Release Planning; R Programming
156	IEEE	Validação de Software	A fuzzy-based methodology to assess software usability risk	A. D. Kartika; K. Surendro	2016	risk assessment; usability software; usability risk; fuzzy; AHP; design-reality gap
158	IEEE	Projeto e Implementação de Software	An empirical study on the importance of quality among offshore outsourced software development firms in Sri Lanka	C. D. Manawadu; S. G. S. Fernando; K. Schmidt; H. J. Perera	2013	Quality; Off-shore outsourcing; Sri Lanka; Software Development
159	IEEE	Projeto e Implementação de Software	Integration of User Centered Design and Software Development Process	S. Dhandapani	2016	UCD; SDLC; Waterfall model; Iterative Model; Incremental model; Agile Model; Prototype Model

Fonte: Os Autores

Tabela B.18: Tabela de artigos selecionados (18/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
160	IEEE	Projeto e Implementação de Software	Models for Software Development Cost with Multiple Market Factors	J. Hu; Z. Yang	2012	inside information;additional utility;field;martingale
161	IEEE	Evolução de Software	Developer experience: Concept and definition	F. Fagerholm; J. Münch	2012	Developer experience;software development environment;high-performing teams;human factors;software psychology
162	IEEE	Projeto e Implementação de Software	Determining the Factors Affecting the Accuracy of Effort Estimates for Different Application and Task Types	S. Bukhari; A. A. Malik	2012	estimation accuracy;effort estimation;empirical study;project management
163	IEEE	Validação de Software	Software Quality Metrics – Research, Analysis and Recommendation	T. Mladenova	2020	Software Quality Testing;testing metrics;quality metrics;quality measurements
164	IEEE	Projeto e Implementação de Software	Transitioning from Plan-Driven to Lean in a Global Software Engineering Organization: A Practice-Centric View	M. S. Roopa; R. Kumar; V. S. Mani	2018	Global software engineering;Lean;Agile;software engineering practices

Fonte: Os Autores

Tabela B.19: Tabela de artigos selecionados (19/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
165	IEEE	Projeto e Implementação de Software	EmoD: An End-to-End Approach for Investigating Emotion Dynamics in Software Development	K. P. Neupane; K. Cheung; Y. Wang	2019	Emotion awareness, emotion dynamics, emotion intensity, software project team, time series database
166	IEEE	Validação de Software	Usability Testing Methodology: Effectiveness of Heuristic Evaluation in E-Government Website Development	A. Sivaji; A. Abdullah; A. G. Downe	2011	Usability Testing Methodology;Heuristic Evaluation;Defect Detection;Iterative Testing
168	IEEE	Especificação de Software	Software requirement analysis template with automation aided system	M. A. Jabar; F. Sidi; A. Azizan; A. A. A. Ghani	2012	requirement analysis;requirement elicitation;usability software;requirement determination
169	IEEE	Projeto e Implementação de Software	User experiences system design for visually impaired and blind users in Oman	A. Al-Busaidi; P. Kumar; C. Jayakumari; P. Kurian	2017	User Experience;accessibility;usability;Evaluation methods;Interactive media;user Interface design;Human Computer Interface;Voice reorganization;visually impaired and blind users
170	IEEE	Projeto e Implementação de Software	Combining Text Mining and Visualization Techniques to Study Teams' Behavioral Processes	S. A. Licorish; S. G. Macdonell	2014	text mining;visualization;linguistic analysis

Fonte: Os Autores

Tabela B.20: Tabela de artigos selecionados (20/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
171	IEEE	Projeto e Implementação de Software	Towards Lean R&D: An Agile Research and Development Approach for Digital Transformation	M. Kalinowski; S. T. Batista; H. Lopes; S. Barbosa; M. Poggi; T. Silva; H. Villamizar; J. Chueke; B. Teixeira; J. A. Pereira; B. Ferreira; R. Lima; G. da Silva Cardoso; A. F. Teixeira; J. A. Warrak; M. Fischer; A. Kuramoto; B. Itagyba; C. Salgado; C. Pelizaro; D. Lemes; M. S. da Costa; M. Waltemberg; O. Lopes	2020	digital transformation; agile methods; lean; research and development; continuous experimentation
172	IEEE	Validação de Software	Revizor: A Data-Driven Approach to Automate Frequent Code Changes Based on Graph Matching	O. Smirnov; A. Lobanov; Y. Golubev; E. Tikhomirova; T. Bryksin	2021	Codes; Prototypes; History; Usability; Open source software; Software engineering; Software development management
173	IEEE	Validação de Software	Usability Testing in Kanban Agile Process for Club Management System	A. Rahmat; N. A. M. Hanifah	2020	club management system; kanban; usability
174	IEEE	Evolução de Software	Usability Technical Debt in Software Projects: A Multi-Case Study	L. C. d. F. Lage; M. Kalinowski; D. Trevisan; R. Spinola	2019	Technical Debt; Usability Debt; Case Study
175	IEEE	Projeto e Implementação de Software	Suitability of PMBOK 6 th edition for agile-developed IT Projects	P. Rosenberger; J. Tick	2018	agile project management; PMBOK; Scrum; agile practices; agile; IT project management

Fonte: Os Autores

Tabela B.21: Tabela de artigos selecionados (21/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
176	IEEE	Validação de Software	Identifying usability risk: A survey study	J. T. Sambantha Moorthy; S. bin Ibrahim; M. N. Mahrin	2014	usability;usability risk;risk management;risk identification
177	IEEE	Validação de Software	Test Tools: an illusion of usability?	I. Evans; C. Porter; M. Micallef; J. Harty	2020	software testing;test tools;automation;usability;quality in use;user experience
178	IEEE	Validação de Software	A Decision Support System for Global Team Management: Expert Evaluation	S. Beecham; N. Carroll; J. Noll	2012	Global Software Development;Distributed Software;Decision Support System;Software Process;Global Teaming Model;Expert Evaluation
179	IEEE	Projeto e Implementação de Software	Impact of the new Software development methodologies in Puerto Rico: An exploratory survey	A. Alvear-Suárez; J. M. Vara-Mesa; V. Bollati	2020	software development;agile methodologies;hybrid methodologies
183	IEEE	Projeto e Implementação de Software	How to Select the Appropriate Pattern of Human-Computer Interaction?: A Case Study with Junior Programmers	V. Silva-Rodríguez; S. E. Nava-Muñoz; F. E. Martínez-Pérez; H. G. Pérez-González	2018	HCI Patterns;Human-Computer Interaction;User Interface Design;Software Requirement;Software Engineering

Fonte: Os Autores

Tabela B.22: Tabela de artigos selecionados (22/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
184	IEEE	Projeto e Implementação de Software	Template based HDF5 satellite digital data product generation software	N. Gaur; S. Mahammad; V. Sharma; D. Dhar; R. Ramakrishnan	2012	HDF5 data product;template based software design
185	IEEE	Validação de Software	Enhancing the Effectiveness of Usability Evaluation by Automated Heuristic Evaluation System	A. Sivaji; S. -T. Soo; M. R. Abdullah	2011	Usability Management System;(UMS);Automation;Heuristice Evaluation;(HE)
186	IEEE	Validação de Software	Evaluation of prototyping tools based on categories and parameters	Jimmy Banchon Z; M. Villavicencio; E. Izquierdo	2016	evaluation;tools;prototype;low cost;software engineering;software development process
187	IEEE	Projeto e Implementação de Software	Identification of Inaccurate Effort Estimates in Agile Software Development	F. Raith; I. Richter; R. Lindermeier; G. Klincker	2013	agile;effort estimation;scrum;planning poker;media supported interaction
188	IEEE	Projeto e Implementação de Software	Integrating UCD into Scrumban for better and faster usability design	D. I. Sensuse; D. Satria; A. A. Pratama; I. A. Wulandari; M. Mishbah; H. Noprisson	2017	agile;scrum;ucd;scrumban;ui;system usability scale

Fonte: Os Autores

Tabela B.23: Tabela de artigos selecionados (23/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
189	IEEE	Projeto e Implementação de Software	Live simulation of software processes in programming courses	J. Porubän; M. Baciková	2016	Companies;Software;Education;Programming profession;Project management;Semantics
190	IEEE	Especificação de Software	Towards Utility-Based Prioritization of Requirements in Open Source Environments	A. Felfernig; M. Stettinger; M. Atas; R. Samer; J. Nerlich; S. Scholz; J. Tiihonen; M. Raatikainen	2018	Prioritization;Requirements Engineering;Constraint Satisfaction;Model based Diagnosis;Reconfiguration;Re-Planning;Multi Attribute Utility Theory;Group Decision Making;Group Recommender Systems;Open Source;OSS
191	IEEE	Projeto e Implementação de Software	A Conceptual Model of User Experience in Scrum Practice	S. Kikitamara; A. A. Noviyanti	2018	User Experience;Agile Methodology;Scrum
192	IEEE	Projeto e Implementação de Software	Scrum Implementation for Online Transaction Processing (OLTP) in Hospital Management	T. Setiadi; S. B. Premapasha	2018	E-class hospital;Online Transaction Processing (OLTP);Scrum;Evaluation
193	IEEE	Projeto e Implementação de Software	Integrating requirements engineering and user experience design in Product life cycle Management	P. Anitha; B. Prabhu	2012	Requirements Engineering (RE);User Experience Design (UXD);Product Life Cycle Management (PLM)

Fonte: Os Autores

Tabela B.24: Tabela de artigos selecionados (24/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
194	IEEE	Projeto e Implementação de Software	Supporting component presence notifications in software development	R. R. Palacio; G. Padilla; A. L. Morán; J. Cortez; A. Vizcaíno	2013	Component-based Software Engineering; Software Engineering; Notifications; Collaborative Systems
195	IEEE	Validação de Software	Comparative Study of Open Source Automation Testing Tools: Selenium, Katalon Studio & Test Project	B. Majeed; S. K. Toor; K. Majeed; M. N. A. Chaudhary	2021	Open source Automation testing tools; Selenium tool; Katalon studio; Test project tool; Evaluation of open source automation tools
199	IEEE	Projeto e Implementação de Software	Improving business decision making based on KPI management system	P. R. Martins de Andrade; S. Sadaoui	2017	E-business; Decision Support Systems; KPI Management; Software Development; Business Metrics
200	IEEE	Validação de Software	Unit Test Code Generation Tool Support for Lower Level Programming Languages	R. Bandara; I. Perera	2020	Google mock; Google test; unit test; unit test code; unit test specification
202	IEEE	Projeto e Implementação de Software	Emotion-based social computing platform for streaming big-data: Architecture and application	L. Zhang; J. Zhao; K. Xu	2016	Quality of service; Computer architecture; Logic gates; Reliability

Fonte: Os Autores

Tabela B.25: Tabela de artigos selecionados (25/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
203	IEEE	Especificação de Software	Determining interdependency among non-functional requirements to reduce conflict	M. R. Tabassum; Md. Saeed Siddik; M. Shoyaib; S. M. Khaled	2014	Usability;Scalability;Security;Software systems;Software reliability
204	IEEE	Projeto e Implementação de Software	Use of the PlantPredict Application Programming Interface for Automating Energy Prediction-Based Analyses	S. Kaplan; R. Callaway; L. Ngan; K. Passow	2018	performance modeling;energy simulation;application programming interface;Python
208	IEEE	Validação de Software	Generic evaluation framework for games development methodology	R. Al-azawi; A. Ayes; M. A. Obaidy	2013	Games;Usability;Presses;Information technology;Information management;Security;Educational institutions
209	IEEE	Especificação de Software	Context-Sensitive Traceability Controlling	A. Mazak; H. Kargl	2013	Requirements management;traceability;cognitive engineering;quality control;decision theory
210	IEEE	Validação de Software	Introducing usability testing in the risk management process in software development	C. Lindholm; M. Höst	2013	Usability;risk management;usability testing;case study;software

Fonte: Os Autores

Tabela B.26: Tabela de artigos selecionados (26/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
211	IEEE	Projeto e Implementação de Software	A Tool for Modeling Software Development Contexts in Small Software Organizations	D. Ortega; L. Silvestre; M. C. Bastarrica; S. F. Ochoa	2012	Software development context;model-driven engineering;modeling tool;software process tailoring
212	IEEE	Evolução de Software	Barriers Faced by Newcomers to Software-Crowdsourcing Projects	A. L. Zanatta; I. Steinmacher; L. S. Machado; C. R. B. de Souza; R. Prikladnicki	2017	software crowdsourcing;crowdsourcing;software engineering;software development;Topcoder
213	IEEE	Especificação de Software	Systematic Literature study on estimation and prioritization of quality requirements in software development	M. J. Salamea; L. González-Palacio; M. Oriol; C. Farré	2020	quality requirements;non-functional requirement;estimation;prioritization;agilism
214	IEEE	Projeto e Implementação de Software	Organizational and project process definition: A lightweight framework, supporting tool, and industry experience	X. Cui	2016	process definition;process tailoring;organizational standard process;project's defined process;life cycle model
215	IEEE	Evolução de Software	Exploring Reuse Levels in ERP Projects in Search of an Effort Estimation Approach	O. Demirörs; N. Küçükateş Ömür	2018	enterprise resource planning;effort estimation;function points;reuse

Fonte: Os Autores

Tabela B.27: Tabela de artigos selecionados (27/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
216	IEEE	Projeto e Implementação de Software	A Quality Framework for Evaluating Grammatical Structure of User Stories to Improve External Quality	S. Jiménez; R. Juárez-Ramírez	2019	Quality;User Stories;Grammatical Structure
217	IEEE	Validação de Software	Some Aspects of Test Data Management Strategy	S. Kasturi	2020	Data Masking;Data Encryption;Data Architecture;Data Provisioning;Data sub setting;Data Breach;Data Refresh;Synthetics Data
218	IEEE	Validação de Software	Representations of user feedback in an agile, collocated software team	M. J. Lee; A. J. Ko	2012	User-centered design;feedback;issue tracking
219	IEEE	Projeto e Implementação de Software	Contextual representations for enterprise model application (C.R.E.M.A.)	N. Wintrich; M. Meißner	2015	Contextual Perspectives;Enterprise Models;Business Process Models;Users;Complexity;IEM;Process Assistant
221	IEEE	Projeto e Implementação de Software	Application Lifecycle Management as Infrastructure for Software Process Improvement and Evolution: Experience and Insights from Industry	H. Lacheiner; R. Ramler	2011	application lifecycle management;AL;software engineering environment;process management

Fonte: Os Autores

Tabela B.28: Tabela de artigos selecionados (28/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
222	IEEE	Validação de Software	A Metrics Tracking Program for Promoting High-Quality Software Development	K. Slhoub; F. Nembhard; M. Carvalho	2019	software quality;software metrics;software measurement;assessment;coding standards;documentation standards;software engineering
223	IEEE	Evolução de Software	Impact of knowledge management on offshore software development: An exploratory study	N. V. A. Prabhu; R. Latha; K. Sankaran; G. Kannabiran	2011	software quality;knowledge transfer and integration;knowledge sharing;knowledge management
224	IEEE	Projeto e Implementação de Software	Integrating discount usability in scrum development process in Ethiopia	D. Teka; Y. Dittrich; M. Kifle	2017	Agile methods;Scrum;discount usability;culture;local IT personnel;heuristic evaluation;user pair testing
225	IEEE	Projeto e Implementação de Software	Integrating Human-Centered Design into Software Development: An Action Research Study in the Automation Industry	K. Viikki; J. Palviainen	2011	Human-centered design;software engineering;process integration;HCD-SE integration;strategic usability;automation design

Fonte: Os Autores

Tabela B.29: Tabela de artigos selecionados (29/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
226	IEEE	Projeto e Implementação de Software	A light-weight incremental effort estimation model for use case driven projects	K. Qi; B. W. Boehm	2017	effort estimation;use case analysis;use case driven process;model calibration;data normalization;software size metrics;model-based analysis;incremental estimation model;use case points;function points;unified modeling language;object-oriented modeling;project management;automated analysis
229	IEEE	Projeto e Implementação de Software	Supporting Virtual Meetings in Distributed Scrum Teams	G. Rodriguez; A. Soria; M. Campo	2012	Agile Project Management;Scrum;Virtual Meetings
230	IEEE	Evolução de Software	Argumentation schemes for the reuse of argumentation information in collaborative risk management	R. C. B. Pozzebon; L. A. L. Silva; L. M. Fontoura; J. A. Campbell	2014	Argumentation Schemes;Collaborative Risk Management;Software Risk Management;Dialogue Game;Argumentation
232	IEEE	Validação de Software	Deriving high-priority acceptance test cases using utility trees: A case study	P. Cruz; H. Astudillo	2017	Software;Proposals;Software testing;Password;Software engineering;Computer architecture
233	IEEE	Projeto e Implementação de Software	A statistical based resource allocation scheme in cloud	Z. Zhang; H. Wang; L. Xiao; L. Ruan	2011	Cloud Computing;Resource Management;Load Balancing;Statistical
234	IEEE	Especificação de Software	A survey on issues in non-functional requirements elicitation	S. Ullah; M. Iqbal; A. M. Khan	2011	Requirement Engineering;Non-Functional Requirements;Literature Survey;Approaches;Techniques;Issues

Fonte: Os Autores

Tabela B.30: Tabela de artigos selecionados (30/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
235	IEEE	Especificação de Software	Measureability of Functional Size in Agile Software Projects: Multiple Case Studies with COSMIC FSM	T. Hacaloglu; O. Demirors	2019	agile, functional size, COSMIC
236	IEEE	Especificação de Software	Toward a Flexible Bidding Strategy with Preference Analysis	N. Zhou; W. M. Gifford; K. Ratakonda	2013	Custom application development;bidding;productivity;team size;team collaboration;indifference curve;risk management
238	IEEE	Projeto e Implementação de Software	Ten Principles for Creating Usable Software for Science	L. Ramakrishnan; D. Gunter	2017	usability;escience
239	IEEE	Projeto e Implementação de Software	IT Project Failures, Causes and Cures	S. Lauesen	2020	Project management;project failure;accident prevention;software development
240	IEEE	Validação de Software	Code Quality Improvement for All: Automated Refactoring for Scratch	P. Techapalokul; E. Tilevich	2019	block-based languages;software refactoring;Scratch;code quality;code smells;program analysis;end-user programming;introductory curriculum

Fonte: Os Autores

Tabela B.31: Tabela de artigos selecionados (31/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
241	IEEE	Especificação de Software	Effective Domain Modeling for Mobile Business AHMS (Adaptive Human Management Systems) requirements	H. Kim	2014	Mobile business domain engineering; Mobile domain analysis; Domain reuse; Component-based Software Development; Business Domain model; Mobile business domain architecture
244	IEEE	Projeto e Implementação de Software	Information Visualization for Agile Software Development	J. Paredes; C. Anslow; F. Maurer	2014	Agile software development; information visualization; software visualization; information radiator; big visible charts; knowledge sharing; systematic mapping
245	IEEE	Projeto e Implementação de Software	Case Study on Data-Driven Deployment of Program Analysis on an Open Tools Stack	A. Ljungberg; D. Åkerman; E. Söderberg; G. Lundh; J. Sten; L. Church	2021	program analysis; usability; case study
246	IEEE	Validação de Software	Preliminary Evaluation of a Guided Usability Defect Report Form	N. S. M. Yusop; J. Grundy; J. -G. Schneider; R. Vasa	2018	Bugzilla; JIRA; expert judgment; usability defect reporting
247	IEEE	Projeto e Implementação de Software	Application of requirements prioritization decision rules in software product line evolution	M. Inoki; T. Kitagawa; S. Honiden	2014	requirements definition; requirements prioritization; decision rules; software product line; core assets; software evolution

Fonte: Os Autores

Tabela B.32: Tabela de artigos selecionados (32/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
248	IEEE	Projeto e Implementação de Software	iUCP: Estimating Interactive-Software Project Size with Enhanced Use-Case Points	N. Nunes; L. Constantine; R. Kazman	2011	design tools and techniques. distribution; maintenance; enhancement. usability testing
251	IEEE	Especificação de Software	Semi-Automated Extraction of New Requirements from Online Reviews for Software Product Evolution	J. Buchan; M. Bano; D. Zowghi; P. Volabouth	2018	Software requirements; Feature request; Software product line; Online reviews
255	IEEE	Projeto e Implementação de Software	Generative Patterns for Designing Multiple User Interfaces	T. Nguyen; J. Vanderdonck; A. Seffah	2016	Cross-device pattern; Generative pattern; Mobile service development; Multiple User Interfaces; User interface pattern.
256	IEEE	Projeto e Implementação de Software	Challenges and strategies for running controlled crowdsourcing experiments	J. Ramírez; M. Baez; F. Casati; L. Cernuzzi; B. Benatallah	2020	Crowdsourcing; Task Design; Controlled experiments; Crowdsourcing Platforms
257	IEEE	Projeto e Implementação de Software	The implementation experience of an advanced service repository for supporting service-oriented architecture	C. Garcia; W. Roberti; D. Schreiber; M. Paludo; A. Malucelli; S. Reinehr	2012	SOA; systematic reuse; service repository; uddi

Fonte: Os Autores

Tabela B.33: Tabela de artigos selecionados (33/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
259	IEEE	Evolução de Software	Quantifying and Mitigating Turnover-Induced Knowledge Loss: Case Studies of Chrome and a Project at Avaya	P. C. Rigby; Y. C. Zhu; S. M. Donadelli; A. Mockus	2016	Quantitative Risk Management; Mining Software Repositories; Knowledge Distribution; Truck Factor; Successors; Turnover
260	IEEE	Validação de Software	Comparison of software repositories for their usability in software process reconstruction	M. Janković; M. Bajec	2015	Software; Companies; Ontologies; Data mining; Documentation
262	IEEE	Validação de Software	Software metrics for cooperative scrum based ontology analysis	W. Mohsen; M. Aref; K. ElBahnasy	2017	ontology quality; software metrics; team performance; architectural metric
263	IEEE	Validação de Software	User-Centered Design of Visualizations for Software Vulnerability Reports	S. L. Reynolds; T. Mertz; S. Arzt; J. Kohhammer	2021	Human centered computing; Visualization; Visualization application domains; Visual analytics; Security and privacy; Software and application security; Systems security; Vulnerability management; Vulnerability scanners
264	IEEE	Validação de Software	Automatically Selecting Follow-up Questions for Deficient Bug Reports	M. M. Imran; A. Ciborowska; K. Damevski	2021	follow-up questions; bug reporting; bug triage

Fonte: Os Autores

Tabela B.34: Tabela de artigos selecionados (34/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
265	IEEE	Projeto e Implementação de Software	Crowdsourcing Software Task Assignment Method for Collaborative Development	D. Yu; Z. Zhou; Y. Wang	2019	Crowdsourcing;software development;collaborative candidate group;Hungarian algorithm;active time
266	IEEE	Projeto e Implementação de Software	A Novel Lightweight Solo Software Development Methodology With Optimum Security Practices	S. Moyo; E. Mnkandla	2020	Agility degree;quality practices;security practice;software development methodology;software product;software quality;software security;solo developer
267	IEEE	Projeto e Implementação de Software	Software Crowdsourcing Task Allocation Algorithm Based on Dynamic Utility	D. Yu; Y. Wang; Z. Zhou	2019	Capability measurement;Kuhn-Munkres algorithm;skill matching;software complexity;software crowdsourcing;task assignment
269	IEEE	Projeto e Implementação de Software	Birds of a Feather Gel Together: Impact of Team Homogeneity on Software Quality and Team Productivity	N. Qamar; A. A. Malik	2019	Five factor model;personality assessment;social aspects of software development;software developer personality traits;software quality;team homogeneity index;team productivity
271	IEEE	Evolução de Software	Enabling the Reuse of Software Development Assets Through a Taxonomy for User Stories	E. Dilorenzo; E. Dantas; M. Perkusich; F. Ramos; A. Costa; D. Albuquerque; H. Almeida; A. Perkusich	2020	Agile software development;software reuse;user stories;information retrieval;technology acceptance model

Fonte: Os Autores

Tabela B.35: Tabela de artigos selecionados (35/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
273	IEEE	Validação de Software	Rate-Based Queueing Simulation Model of Open Source Software Debugging Activities	C. Lin; Y. Li	2014	Queueing theory;rate-based simulation;open source software (OSS);bug reporting;report judgment;bug fixing;optimal version-updating time;non-homogeneous continuous time Markov chain (NHCTMC);multi-attribute utility theory (MAUT)
280	ACM	Projeto e Implementação de Software	Software Development Process Animation	Agarwal R	2011	animation, agent-based simulation, software process
282	ACM	Projeto e Implementação de Software	Software Development Risk Management: Using Machine Learning for Generating Risk Prompts	Joseph HR	2015	risk prompts, software management, software risk, machine learning
284	ACM	Projeto e Implementação de Software	Enhancing Architecture-Implementation Conformance with Change Management and Support for Behavioral Mapping	Zheng Y,Taylor RN	2012	software architecture;architecture change management;architecture-implementation mapping
285	ACM	Projeto e Implementação de Software	Seven Habits of Highly Impactful Empirical Software Engineers	Williams L	2011	

Fonte: Os Autores

Tabela B.36: Tabela de artigos selecionados (36/36)

N	Encontrado em	Categoria	Título	Autores	Ano	Palavras-chave
286	ACM	Projeto e Implementação de Software	Masters of the Process: A Board Game Proposal for Teaching Software Management and Software Development Process	Sarinho VT	2019	board games, software development process, education, software engineering, software management
287	ACM	Projeto e Implementação de Software	Managing Design-Time Uncertainty	Famelis M,Chechik M	2017	
288	ACM	Validação de Software	On Regulatory and Organizational Constraints in Visualization Design and Evaluation	Crisan A,Gardy JL,Munzner T	2016	Design; Project management; Research and development management; Software design; Design studies; External constraints; Organizational constraints; Premature termination; Problem-based; User's needs; Visualization designs; Visualization research; Visualization
289	ACM	Projeto e Implementação de Software	A Survey of Software Log Instrumentation	Chen B,Jiang ZM	2021	software logging, instrumentation, Systematic survey
291	ACM	Evolução de Software	A Case Study on Effectively Identifying Technical Debt	Zazworka N,Spínola RO,Vetro' A,Shull F,Seaman C	2013	automatic static analysis, code smells, software maintenance, technical debt
292	ACM	Projeto e Implementação de Software	Controlled English Language for Production and Event Processing Rules	Linehan MH,Dehors S,Rabinovich E,Fournier F	2011	sbvr, semantics of business vocabulary and business rules, controlled natural language, structured english, event processing language, complex event processing

Fonte: Os Autores