



**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**ELIELTON DA COSTA CARVALHO**

**ABORDAGENS PARA APOIAR O GERENCIAMENTO DE  
PROJETOS DE SOFTWARE NO CONTEXTO ÁGIL: UMA  
PESQUISA EXPLORATÓRIA E DESCRITIVA**

Belém  
2023

ELIELTON DA COSTA CARVALHO

**ABORDAGENS PARA APOIAR O GERENCIAMENTO DE  
PROJETOS DE SOFTWARE NO CONTEXTO ÁGIL: UMA  
PESQUISA EXPLORATÓRIA E DESCRITIVA**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Área de Concentração: Sistemas de Informação.

Orientador: Prof. Dr. Sandro Ronaldo Bezerra Oliveira.

Belém  
2023

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD  
Sistema de Bibliotecas da Universidade Federal do Pará  
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

---

D111a da Costa Carvalho, Eielton.  
ABORDAGENS PARA APOIAR O GERENCIAMENTO DE  
PROJETOS DE SOFTWARE NO CONTEXTO ÁGIL : UMA  
PESQUISA EXPLORATÓRIA E DESCRITIVA / Eielton da  
Costa Carvalho. — 2023.  
193 f. : il. color.

Orientador(a): Prof. Dr. Sandro Ronaldo Bezerra Oliveira  
Dissertação (Mestrado) - Universidade Federal do Pará,  
Instituto de Ciências Exatas e Naturais, Programa de Pós-  
Graduação em Ciência da Computação, Belém, 2023.

1. Gerência de Projetos de Software. 2. Desenvolvimento  
Ágil. 3. Abordagens de Gerenciamento de Projetos de  
Software. I. Título.

CDD 004.072

---

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**ELIELTON DA COSTA CARVALHO**

**ABORDAGENS PARA APOIAR O GERENCIAMENTO DE PROJETOS DE SOFTWARE NO  
CONTEXTO ÁGIL: UMA PESQUISA EXPLORATÓRIA E DESCRITIVA**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Pará como requisito para obtenção do título de Mestre em Ciência da Computação, defendida e aprovada em 18/01/2023, pela banca examinadora constituída pelos seguintes membros:

*Sandro Ronaldo B. Oliveira*

---

**PROF. DR. SANDRO RONALDO BEZERRA OLIVEIRA**  
Orientador – PPGCC/UFPA

*Carlos dos Santos Portela*

---

**PROF. DR. CARLOS DOS SANTOS PORTELA**  
Membro Externo – UFPA

*Isaac Souza Elgrably*

---

**PROF. DR. ISAAC SOUZA ELGRABLY**  
Membro Externo – CESUPA

Visto:

*Marcelle Pereira Mota*

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Marcelle Pereira Mota  
Coordenadora do PPGCC/UFPA

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por ter me proporcionado força, sabedoria, discernimento para a concretização de mais esta etapa em minha vida.

Agradeço a minha mãe, Maria Carvalho, por todo apoio mesmo nos momentos mais difíceis que passamos. Minha eterna gratidão.

Agradeço a minha família por todo incentivo e cuidado.

Agradeço meus amigos Lion, Andréia, Fran, Taís, Eduardo, Carol, Brenda e Elaize que sempre me incentivaram para que eu continuasse a busca pelos meus sonhos.

Agradeço meu ex-professor, Fabrício Garcia, que se tornou um grande amigo e parceiro de pesquisa. Minha gratidão pelo apoio durante essa jornada.

Agradeço meus colegas do Laboratório SPIDER por dividirem comigo seus conhecimentos e contribuírem para a realização deste trabalho.

Por fim, agradeço imensamente a meu orientador, Prof. Sandro Oliveira, por toda paciência, ensinamentos e amizade. Se consegui finalizar o mestrado, muito eu devo a ele. Gratidão!

## RESUMO

Gerenciar projetos não é uma tarefa trivial, tendo em vista a multiplicidade de fatores que precisam ser levados em consideração, principalmente no contexto de desenvolvimento ágil de software, onde as mudanças são ainda mais rápidas e frequentes. Com isso, as abordagens de gerenciamento de projetos tornaram-se indispensáveis no dia a dia das equipes de desenvolvimento ágil de software e ao longo do tempo passaram por mudanças e evoluções que as deram ainda mais relevância e destaque nesse contexto. No entanto, essas mudanças trouxeram desafios para as organizações ao escolherem a abordagem mais adequada para os objetivos e as necessidades do projeto. Vários estudos foram realizados com o intuito de demonstrar o quão é relevante para a indústria de desenvolvimento de software um gerenciamento bem feito e como as abordagens podem auxiliar nesse processo. Porém, embora os estudos concentrem-se em apresentar o desenvolvimento de novas abordagens de gerenciamento de projetos ou a avaliação das que já existem, não foi verificado um estudo que reúna de forma ampla essas abordagens, incluindo características como: pontos fortes, limitações e onde essas abordagens são aplicadas no gerenciamento de projetos de software dentro do contexto ágil. Assim, este trabalho tem como objetivo identificar quais são as abordagens de gerenciamentos de projetos usadas no desenvolvimento ágil de software. Para atingir o objetivo desta pesquisa, foi realizado um estudo exploratório e descritivo, mediante a condução de um Mapeamento Sistemático da Literatura, seguido de um *Survey* e, ao final, foi desenvolvido um catálogo com as abordagens identificadas na pesquisa. Com isso, espera-se contribuir com o estado da arte, gerentes de projetos e entusiastas na hora de selecionar uma abordagem que melhor adeque-se às suas necessidades diárias.

**PALAVRAS-CHAVE:** Gerência de Projetos de Software; Desenvolvimento Ágil; Abordagens de Gerenciamento de Projetos de Software.

## **ABSTRACT**

Managing projects is not a trivial task, given the multitude of factors that need to be taken into account, especially in the context of agile software development, where changes are even faster and more frequent. As a result, project management approaches have become indispensable in the daily lives of agile software development teams and over time they have undergone changes and evolutions that have given them even more relevance and prominence in this context. However, these changes, over time, brought challenges for organizations when choosing the most appropriate approach for the project's objectives and needs. Several studies have been carried out with the aim of demonstrating how relevant a well-done management is for the agile software development industry and how the approaches can help in this process. However, although studies focus on presenting the development of new project management approaches or the evaluation of existing ones, no study was found that broadly brings together these approaches, including characteristics such as: strengths, limitations and where these approaches are applied in software project management within the agile context. Thus, this work aims to identify which are the project management approaches used in agile software development. To achieve the objective of this research, an exploratory and descriptive study was carried out, by conducting a Systematic Mapping of Literature, followed by a Survey and, at the end, a catalog was developed with the approaches identified in the research. With this, it is expected to contribute with the state of the art, project managers and enthusiasts when selecting an approach that best suits their daily needs.

**KEYWORDS:** Software Project Management; Agile Development; Software Project Management Approaches

## LISTA DE FIGURAS

- Figura 1 – Passos metodológicos da pesquisa19
- Figura 2 – Passos adotados para o MSL32
- Figura 3 – Quantidade de trabalhos incluídos e excluídos por cada critério40
- Figura 4 – Quantidade de trabalhos incluídos e excluídos40
- Figura 5 – Distribuição dos artigos por ano41
- Figura 6 – Quantidade de trabalhos distribuídos por país42
- Figura 7 – Distribuição dos artigos por tipo de instituição43
- Figura 8 – Áreas do projeto onde as abordagens são aplicadas44
- Figura 9 – Fase do projeto onde as abordagens são aplicadas44
- Figura 10 – Tipo de abordagem identificada nos trabalhos selecionados45
- Figura 11 – Forma de avaliação das abordagens56
- Figura 12 – Metodologia utilizada na condução do survey60
- Figura 13 - Área de formação dos participantes63
- Figura 14 - Grau de formação dos participantes64
- Figura 15 - Quantidade de participantes com e sem certificações em gerência de projetos64
- Figura 16 - Nome e quantidade de certificações dos participantes65
- Figura 17 - Faixa etária dos participantes66
- Figura 18 - Tempo de experiência dos participantes66
- Figura 19 - Estado onde se localiza a empresa onde participante atua68
- Figura 20 - Tamanho das empresas68
- Figura 21 - Softwares de gestão de projetos utilizados pelos participantes70
- Figura 22 - Metodologias de gestão de projetos utilizadas pelos participantes71
- Figura 23 - Métodos de gestão de projetos utilizados pelos participantes73
- Figura 24 - Modelos de gestão de projetos utilizados pelos participantes74
- Figura 25 - Ferramentas de gestão de projetos utilizadas pelos participantes76
- Figura 26 - Técnicas de gestão de projetos utilizadas pelos participantes77
- Figura 27 - Frameworks de gestão de projetos utilizados pelos participantes78
- Figura 28 - Práticas de gestão de projetos utilizadas pelos participantes79
- Figura 29 - Abordagens que possuem pontos fortes80
- Figura 30 - Pontos fortes das abordagens citados pelos participantes80
- Figura 31 - Abordagens que possuem pontos fracos82
- Figura 32 - Pontos fracos das abordagens citados pelos participantes83

## LISTA DE QUADROS

- Quadro 1 - Estrutura PICOC definida para o trabalho33
- Quadro 2 - Subquestões de pesquisa33
- Quadro 3 - Critérios de seleção das fontes de busca34
- Quadro 4 - Fontes de busca selecionadas34
- Quadro 5 - Palavras chaves utilizadas para a elaboração da string de busca35
- Quadro 6 - String de busca utilizada36
- Quadro 7 - Critérios de inclusão38
- Quadro 8 - Critérios de exclusão38
- Quadro 9 – Artigos que apresentam abordagem do tipo “Software”46
- Quadro 10 - Artigos que apresentam abordagem do tipo “Metodologia”47
- Quadro 11 - Artigos que apresentam abordagem do tipo “Método”48
- Quadro 12 - Artigos que apresentam abordagem do tipo “Modelo”49
- Quadro 13 - Artigos que apresentam abordagem do tipo “Ferramenta”50
- Quadro 14 - Artigos que apresentam abordagem do tipo “Técnica”51
- Quadro 15 - Artigos que apresentam abordagem do tipo “Framework”52
- Quadro 16 - Artigos que apresentam abordagem do tipo “Prática”53

## LISTA DE ABREVIATURAS E SIGLAS

MSL	Mapeamento Sistemático da Literatura
GT	<i>Grounded Theory</i>
PMI	<i>Project Management Institute</i>
PMBOK	<i>Project Management Body of Knowledge</i>
XP	<i>Extreme Programming</i>
DSDM	<i>Dynamic System Development Model</i>
ASD	<i>Adaptive Software Development</i>
FDD	<i>Feature-Driven Development</i>
GSD	<i>Global Software Development</i>
SWOT	<i>Strengths, Weaknesses, Opportunities, Threats</i>
SPIDER	<i>Software Process Improvement: Development and Research</i>
ACM	<i>Association for Computing Machinery</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
SPESS	<i>Sprint Planning Decision Support System</i>
ESSE	<i>Early Software Size Estimation</i>
MEDIATION	<i>Metrics Driven Research Collaboration</i>
MBA	<i>Master of Business Administration</i>
PMP	<i>Project Management Professional</i>
PSM	<i>Professional Scrum Master</i>
PSPO	<i>Professional Scrum Product Owner</i>
COBIT	<i>Control Objectives for Information and related Technology</i>
ITIL	<i>Information Technology Infrastructure Library</i>
SFPC	<i>Security Fundamentals Professional Certification</i>
SEBRAE	Serviço Brasileiro de Apoio às Micro e Pequenas Empresas
CMMI	<i>Capability Maturity Model Integration</i>
LSTM	<i>Long Short-Term Memory</i>
ASAP	<i>Agile Software Planning Tool</i>
UP	<i>Unified Process</i>
API	<i>Application Programming Interface</i>
GPS	Gerenciamento de Projetos de Software

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>12</b>
1.1.	Contexto do Trabalho .....	12
1.2.	Motivação .....	14
1.3.	Justificativa .....	16
1.4.	Objetivos .....	17
1.4.1	Objetivo Geral .....	17
1.4.2	Objetivos Específicos .....	17
1.5.	Metodologia do Trabalho .....	18
1.6.	Estrutura do Trabalho.....	21
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>23</b>
2.1.	Gestão de Projetos.....	23
2.2.	Metodologia Ágil .....	24
2.3.	Gestão de Projeto Ágil .....	26
2.4.	Tipos de Abordagens Usadas Neste Trabalho .....	27
2.4.1.	Trabalhos Relacionados .....	28
<b>3</b>	<b>ABORDAGENS PARA APOIO AO GERENCIAMENTO DE PROJETOS DE SOFTWARE NO CONTEXTO ÁGIL: UM MAPEAMENTO SISTEMÁTICO DA LITERATURA</b> .....	<b>31</b>
3.1	<b>Metodologia do Mapeamento Sistemático da Literatura</b> .....	<b>31</b>
3.1.1	Planejamento do Mapeamento Sistemático da Literatura .....	31
3.2	<b>Resultados do Mapeamento Sistemático da Literatura</b> .....	<b>39</b>
3.2.1	Resultados Bibliométricos .....	39
3.2.2	Sub-Q1: Como foi a evolução do número de estudos publicados relacionados com o tópico desta pesquisa? .....	41
3.2.3	Sub-Q2: Quais os países que mais publicam trabalhos relacionados com o tópico desta pesquisa? .....	41
3.2.4	Sub-Q3: Qual o tipo de instituição que mais publicaram trabalhos relacionados com o tópico desta pesquisa?.....	42
3.2.5	Sub-Q4: Em qual área de conhecimento do GPS a abordagem é aplicada? .....	43
3.2.6	Sub-Q5: Em qual fase do GPS a abordagem é aplicada?.....	44
3.2.7	Sub-Q6: Qual é o tipo de abordagem? .....	45
3.2.8	Sub-Q7: Quais os pontos fortes e fracos das abordagens?.....	54
3.2.9	Sub-Q8: Como foi feita a avaliação das abordagens?.....	55
<b>4</b>	<b>PERCEPÇÃO SOBRE O USO DE ABORDAGENS DE GERENCIAMENTO DE PROJETOS DE SOFTWARE NO CONTEXTO ÁGIL: UMA APLICAÇÃO DE SURVEY</b> .....	<b>60</b>
4.1	<b>Metodologia do Survey</b> .....	<b>60</b>
4.1.1	Planejamento .....	61
4.1.2	Execução .....	61
4.1.3	Análise dos Dados.....	62
4.2	<b>Resultados do Survey</b> .....	<b>63</b>
4.2.1	Caracterização dos Participantes.....	63

4.2.2	Caracterização das Empresas .....	67
4.2.3	Abordagens de Gerenciamento de Projetos de Software .....	69
<b>5</b>	<b>CATÁLOGO DE ABORDAGENS DE GERENCIAMENTO DE PROJETOS DE SOFTWARE USADOS NO CONTEXTO DO DESENVOLVIMENTO ÁGIL .....</b>	<b>85</b>
<b>5.1</b>	<b>Softwares.....</b>	<b>85</b>
<b>5.2</b>	<b>Metodologias.....</b>	<b>87</b>
<b>5.3</b>	<b>Métodos.....</b>	<b>90</b>
<b>5.4</b>	<b>Modelos.....</b>	<b>92</b>
<b>5.5</b>	<b>Ferramentas.....</b>	<b>94</b>
<b>5.6</b>	<b>Técnicas.....</b>	<b>97</b>
<b>5.7</b>	<b>Frameworks.....</b>	<b>99</b>
<b>5.8</b>	<b>Práticas.....</b>	<b>101</b>
<b>6</b>	<b>CONCLUSÕES .....</b>	<b>103</b>
<b>6.1.</b>	<b>Visão Geral .....</b>	<b>103</b>
<b>6.2.</b>	<b>Resultados Obtidos .....</b>	<b>103</b>
<b>6.3.</b>	<b>Contribuições da Pesquisa.....</b>	<b>105</b>
<b>6.4.</b>	<b>Limitações da Pesquisa.....</b>	<b>105</b>
<b>6.5.</b>	<b>Trabalhos Futuros.....</b>	<b>107</b>
	<b>REFERÊNCIA BIBLIOGRÁFICAS.....</b>	<b>109</b>
	<b>APÊNDICES .....</b>	<b>119</b>

# 1 INTRODUÇÃO

Este capítulo apresenta: a contextualização, a justificativa, a motivação, os objetivos, a metodologia utilizada nesta pesquisa e, finalmente, a estrutura desta dissertação. O objetivo deste capítulo é prover uma visão geral do que foi desenvolvido nesse trabalho.

## 1.1. Contexto do Trabalho

Os projetos tornaram-se ativos primordiais para o sucesso das organizações, independentemente da área em que ela está inserida. Os projetos que são desenvolvidos com sucesso melhoram os lucros das organizações envolvidas e satisfazem os interesses dos clientes (FORSGREN; HUMBLE; KIM, 2018). Sobre os projetos de desenvolvimento de software, estes se tornaram a base para um produto de software bem desenvolvido. Nas últimas décadas essa área do conhecimento evoluiu rapidamente, tornou-se mais estruturada e robusta, essencial para atender as crescentes demandas e melhorias tecnológicas e também um pré-requisito para qualquer empresa que almeja entregar qualidade por meio dos seus produtos (SALEEM *et al.*, 2021).

Hoje os projetos exigem requisitos bem elaborados, isto é, requisitos precisos, um custo bem estimado e um cronograma bem ajustado para garantir o sucesso do desenvolvimento. No entanto, controlar as incertezas de um projeto, principalmente projetos de software, é uma tarefa que exige dos gerentes um aparato de abordagens que o auxiliem nas atividades, seja para comunicação com a equipe, para controlar o cronograma e até mesmo para mitigar os riscos do projeto (HALEEM *et al.*, 2022).

Kerzner (2013) chama a atenção para a importância da gestão de projetos no desenvolvimento de um planejamento correto, organizado e controlado dos recursos organizacionais a fim de cumprir metas de curto prazo, para cumprir metas específicas ou mesmo objetivos mais amplos. Os benefícios da realização de projetos bem-sucedidos podem ser entregues diretamente da criação de novos produtos e serviços, pela redução de certas despesas operacionais ou mesmo por meio de mudanças nas práticas comuns de trabalho, do redesenho de processos e da atualização de habilidades pessoais e profissionais (WARD; DANIEL, 2006).

A demanda por melhores resultados aumentou nos últimos anos devido a crescente pressão por agilidade nas entregas, redução de custos e flexibilidade de escopo. Essas demandas exigem mudanças nos processos de desenvolvimento de software e nas organizações como um todo, as quais exigem investimentos significativos e são de avaliação complexa (FRANCO; HIRAMA; CARVALHO, 2018). Essa demanda por processos mais ágeis foi a mola propulsora para o surgimento do desenvolvimento ágil de software, que surgiu no final da década de 1990 para lidar com as incertezas dos requisitos dos clientes, a evolução da tecnologia e os ambientes de negócios em constante mudança (MANIFESTO ÁGIL, 2001).

Existem muitas características do desenvolvimento ágil de software que afetam a tomada de decisão, incluindo sua natureza iterativa e incremental acelerada, suas funções de desenvolvedor orgânicas e flexíveis, e sua ênfase na auto-organização (DRURY-GROGAN; CONBOY; ACTON, 2017). Para auxiliar os gerentes de projetos e as equipes a tomarem as melhores decisões, existem uma série de abordagens que oferecem uma ampla gama de funções nas áreas de planejamento de projetos, monitoramento e avaliação contínua da implementação do projeto e avaliação final quando ele for concluído (KOSTALOVA; TETREVOVA; SVEDIK, 2015).

Neste contexto, as abordagens de gerenciamento de projetos ganham papel de destaque, uma vez que se tornaram indispensáveis no dia a dia dos gerentes de projetos. Essas abordagens evoluíram do tradicional para o ágil e global, mudando não só o foco de atuação, mas a forma como elas entregam resultados à equipe de desenvolvimento. Essas mudanças sofridas pelas abordagens tornaram-se um desafio para as organizações no que se refere a escolha da abordagem mais adequada para os objetivos e as necessidades do projeto.

A literatura tem relatado um número significativo de melhores práticas, softwares, guias, metodologias e padrões para orientar as equipes de desenvolvimento. No entanto, às vezes essas abordagens são muito teóricas e podem não ser relevantes para um determinado ambiente ou organização (MORANDINI *et al.*, 2021). Além disso, por conta dessa diversidade de abordagens, o processo de escolha de qual delas é a mais adequada para um determinado projeto torna-se uma tarefa complexa à medida que novas abordagens vão surgindo. Portanto, este trabalho tem como objetivo identificar quais são as abordagens de gerenciamento de projetos utilizadas para apoiar o desenvolvimento de software no contexto ágil.

Com o intuito de alcançar o objetivo, foi realizado inicialmente um Mapeamento Sistemático da Literatura (MSL), a fim de identificar na literatura especializada quais são as abordagens que estão sendo desenvolvidas e utilizadas no contexto ágil de desenvolvimento de software. Posteriormente, foi realizado um *survey* com gerentes de projetos que atuam na indústria de desenvolvimento de software, para verificar se o que foi encontrado na literatura é de fato o que está em uso no dia a dia dos profissionais. Ao final desta pesquisa foi elaborado um catálogo com as abordagens identificadas na literatura, contendo suas principais características. Dessa forma, os profissionais de desenvolvimento de software terão disponível um artefato que contribuirá para a tomada de decisão referente a qual (ou quais) abordagem é a mais indicada para seu contexto.

## 1.2. Motivação

Inegavelmente, a indústria de software alavancou-se nas últimas décadas. Porém, com essa evolução vieram novos desafios, e os desafios que a indústria de software enfrenta na atualidade, no que diz respeito à complexidade, tamanho e demanda de sistemas, são bem diferentes e maiores do que os desafios de 30, 20 anos atrás (GAROUSI *et al.*, 2018). No entanto, apesar dos problemas terem aumentado, o objetivo das organizações de desenvolvimento de software permanece o mesmo: entregar produtos de software no cronograma acordado, dentro do orçamento e com qualidade, o chamado Triângulo de Ferro (GALVAN-CRUZ *et al.*, 2021).

Garousi *et al.* (2018) afirmam que, apesar das melhorias, os projetos de desenvolvimento de software continuam a ter altas taxas de falhas, se comparados com projetos tradicionais. Há de se considerar ainda que os projetos de desenvolvimento de software possuem características específicas que os tornam complexos do ponto de vista gerencial (MENEZES JUNIOR; GUSMÃO; MOURA, 2019). Devido a isso, as organizações precisam analisar dados históricos, entender e controlar seus subprocessos e estabelecer metas de melhoria organizacional. Os gerentes de projetos devem, ainda, avaliar constantemente se as metas de qualidade e desempenho estão de acordo com o que foi previamente definido no planejamento do projeto (CERDEIRAL; SANTOS, 2019).

Planejar, executar, controlar e monitorar essa série de tarefas sem o auxílio de alguma abordagem, é praticamente impossível. Acrescenta-se ainda o fato de que a

competição no mercado é acirrada, que as mudanças são cada vez mais rápidas e que o desenvolvimento tecnológico é constante. Portanto, é fundamental conhecer abordagens eficientes de gestão de projetos que ajudem as organizações a se anteciparem e terem resultados cada vez mais satisfatórios frente à concorrência (BADEWI, 2016). Vários estudos foram realizados para demonstrar o valor da gestão de projetos (ZHAI; XIN; CHENG, 2009), (LAPPE; SPANG, 2014), (MIR; PINNINGTON, 2014), (CARVALHO; PATAH; BIDO, 2015), (JOSLIN; MÜLLER, 2015). Ao realizar uma busca nas bases científicas que tratam sobre gerenciamento de projetos, os pesquisadores depararam-se com uma “infinidade” de estudos que relatam o desenvolvimento, aplicação, avaliação e/ou comparação de abordagens voltadas ao gerenciamento de projetos de software.

Zabil *et al.* (2020) conduziram uma RSL (Revisão Sistemática da Literatura) para identificar onde o aprendizado de máquina tem sido utilizado no gerenciamento de projetos de software. Alaidaros *et al.* (2019) desenvolveram uma ferramenta de monitoramento do progresso do gerenciamento de projetos de software, a partir das limitações encontradas nas ferramentas já existentes. Hayat *et al.* (2019) investigaram a influência da metodologia Scrum no gerenciamento de projetos de software, destacando os pontos positivos e negativos do uso dessa metodologia. Por fim, Tam *et al.* (2020) propuseram, a partir de uma pesquisa com 216 profissionais, um modelo que reúne cinco fatores que influenciam no sucesso do gerenciamento de projetos no contexto de desenvolvimento ágil de software.

Embora os estudos concentrem-se em apresentar o desenvolvimento de novas abordagens de gerenciamento de projetos ou a avaliação das que já existem, não foi verificado um trabalho que reúna todas essas abordagens, incluindo seus pontos fortes e fracos, e onde são aplicadas no contexto do gerenciamento de projetos de software. Portanto, o fato de não haver um estudo que tenha identificado abordagens de gerenciamento de projetos e suas aplicações do desenvolvimento ágil de software, fez surgir o seguinte problema: **“Quais são as abordagens de gerenciamento de projetos utilizadas no para apoiar o desenvolvimento de software no contexto ágil?”**.

Neste trabalho, entende-se por abordagens quaisquer softwares, ferramentas, métodos, técnicas, modelos, tecnologias, práticas, padrões, guias, artefatos, metodologias, *frameworks*, processos, princípios, temas e papéis que são utilizados no gerenciamento de projetos de software. Vale destacar que, este trabalho está voltado para

o contexto ágil por entender que a publicação do Manifesto Ágil aumentou as taxas de sucesso de projetos de desenvolvimento de software, porém ainda há a necessidade de pesquisas que amenizem problemas existentes e contribuam para um gerenciamento eficaz e, conseqüentemente, para o desenvolvimento de softwares com níveis de qualidade cada vez maiores (TAM *et al.*, 2020).

### 1.3. Justificativa

O gerenciamento de projetos é inevitável no mundo de hoje, pois há uma necessidade de se obter melhorias contínuas por meio de diferentes tipos de projetos. O gerenciamento de tais atividades não é apenas uma necessidade para essa melhoria, mas também uma área que busca a própria melhoria (RADUJKOVIĆ; SJEKAVICA, 2017). No entanto, o gerenciamento de projetos não é uma tarefa trivial, tendo em vista a multiplicidade de fatores que precisam ser levados em consideração. Diante disso, é essencial que os projetos apliquem de forma eficaz técnicas, métodos, softwares, metodologias, ou seja, abordagens para alcançarem seus objetivos (MENEZES JUNIOR; GUSMÃO; MOURA, 2019).

Atualmente, as organizações têm sua produtividade associada majoritariamente às ferramentas e metodologias de gerenciamento de projetos (MIR; PINNINGTON, 2014). Bastos, Barcellos e Falbo (2018) relatam que nas últimas décadas foram propostas e desenvolvidas várias técnicas de gerenciamento de projetos. Varajão, Colomo-Palacios e Silva (2017) também compartilham uma informação semelhante, ao afirmarem que muitos guias de boas práticas, abrangendo processos e técnicas, foram desenvolvidos, cobrindo todos os aspectos do gerenciamento de projetos, desde o início até o encerramento.

A popularidade dessas abordagens é confirmada por Fortune *et al.* (2011), que realizaram um estudo e identificaram um aumento significativo no uso de metodologias e ferramentas de gerenciamento de projetos entre os profissionais da área. No entanto, o gerenciamento de projetos continua sendo um empreendimento altamente problemático (VARAJÃO; COLOMO-PALACIOS; SILVA, 2017). Nesse cenário, Filippetto, Lima e Barbosa (2020) contribuem dizendo que abordagens aplicadas em contextos ágeis são alternativas para a gestão de projetos ao buscar resultados de negócios mais rápidos e com maior qualidade. Segundo Parsi (2021), os gerentes de projetos não estão apenas percebendo o valor do desenvolvimento ágil, eles reconhecem que explorar novas

abordagens pode ajudar as organizações a se adaptarem mais rapidamente às novas realidades.

Observa-se que os desafios no desenvolvimento ágil foram e continuam sendo explorados, mas não houve nenhum trabalho abrangente sobre essas abordagens de gerenciamento de projetos e como elas são usadas e combinadas na prática, o que é um indicativo da necessidade de uma investigação adicional nesta área (SHRIVASTAVA; RATHOD, 2017). É importante, portanto, pesquisar as características dessas abordagens e mapeá-las para minimizar os desafios das decisões.

Adicionalmente, faz-se necessário, também, investigar como elas são usadas e as lições aprendidas a partir de seu uso, a fim de identificar oportunidades de melhorias (VARAJÃO; COLOMO-PALACIOS; SILVA, 2017). No entanto, devido a existência de muitas abordagens diferentes de gerenciamento de projetos, é difícil fazer uma forte diferenciação entre elas, principalmente por causa de seus relacionamentos mútuos. Além disso, torna-se importante também identificar seus principais pontos forte e fracos com base na sua aplicação em um contexto real de desenvolvimento de software (RADUJKOVIĆ; SJEKAVICA, 2017).

Com base no que foi apresentado, esta pesquisa justifica-se pela necessidade de conhecer quais são as principais abordagens de gerenciamento de projetos utilizadas para apoiar o desenvolvimento de software no contexto ágil. Além disso, a criação de um catálogo dessas abordagens justifica-se por ser um documento que reunirá em um único local as principais informações e, assim, permitirá que os profissionais da área tenham uma base para consultar e selecionar a abordagem mais adequada para o seu contexto.

## **1.4. Objetivos**

Este subcapítulo apresenta o objetivo principal desta pesquisa, assim como também os objetivos específicos.

### **1.4.1 Objetivo Geral**

Esta dissertação tem como objetivo identificar as abordagens de gerenciamento de projetos usadas no contexto ágil de desenvolvimento de software.

### **1.4.2 Objetivos Específicos**

Para atingir o objetivo geral, esta dissertação tem os seguintes objetivos específicos:

- Identificar as abordagens que constam relatadas na literatura por meio de um MSL;
- Verificar, por meio de um survey, se o que foi identificado na literatura está sendo usado na indústria de desenvolvimento de software;
- Criar um catálogo com as abordagens identificadas no MSL.

## 1.5. Metodologia do Trabalho

O campo da pesquisa científica é vasto e cada uma possui peculiaridades que a torna única. Então, é natural que as pesquisas necessitem de classificações, pois isso permite uma maior organização e, assim, proporciona um melhor entendimento a seu respeito (GIL, 2008). Mediante ao exposto, esta pesquisa é classificada:

- **Quanto à natureza:** apresenta-se como uma pesquisa aplicada, pois este trabalho propõe-se a solucionar um problema de um grupo específico e busca construir um conhecimento que possa ser aplicado de forma prática;
- **Quanto à abordagem do problema:** apresenta-se como uma pesquisa quantitativa e qualitativa, uma vez que foram coletados dados numéricos e dados textuais;
- **Quanto aos objetivos:** apresenta-se como uma pesquisa exploratória e descritiva, visto que, foi realizado um levantamento bibliográfico por meio de um MSL e, também, um levantamento de opiniões de um grupo de pessoas específicas, a fim de tornar o problema mais explícito;
- **Quanto aos procedimentos técnicos:** apresenta-se como uma pesquisa bibliográfica e de levantamento, pois a investigação foi iniciada com um MSL e, posteriormente, foi realizado um *survey* para interrogar diretamente um grupo de pessoas cujo a opinião pretendia-se conhecer.

Mediante esta classificação, o fluxo metodológico desenvolvido nesta dissertação de mestrado segue o modelado na Figura 1. Neste trabalho, inicialmente foi desenvolvido um MSL composto de três etapas, a saber: (i) Planejamento, (ii) Execução e (iii) Análise dos Dados. A etapa de planejamento consistiu na elaboração de um protocolo contendo as questões de pesquisa, bem como todas as restrições necessárias para alcançar o objetivo

da MSL, que foi base para etapa posterior. Na etapa de execução o protocolo construído no planejamento foi executado, isto é, a *string* de busca foi rodada nas bases selecionadas e os trabalhos foram selecionados de acordo com os critérios estabelecidos. Por fim, na etapa de análise os dados foram ajustados, tratados e organizados por meio da estatística descritiva.

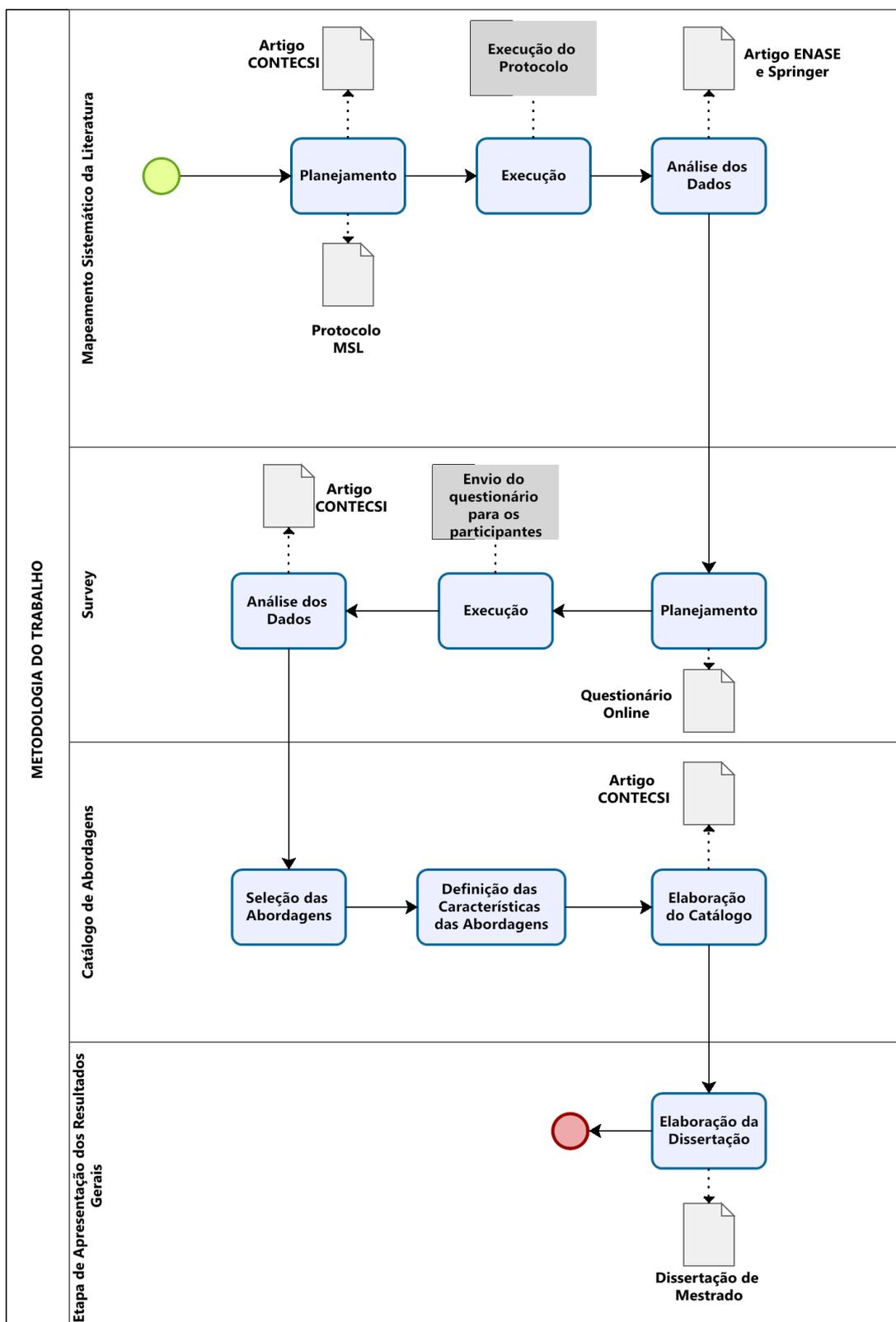
Assim como o MSL, o *survey* também foi conduzido mediante três etapas: (i) Planejamento, (ii) Execução e (iii) Análise dos Dados. Na etapa de planejamento foram realizadas atividades como: definição da população alvo, concepção da amostra, definição do método de coleta, construção do instrumento de coleta e realização de um teste piloto. Para a elaboração do instrumento de coleta foi utilizado o Microsoft Forms<sup>1</sup>.

Na etapa de execução, o instrumento de coleta dos dados foi enviado para os participantes que foram selecionados de acordo com a etapa de planejamento, assim como também foram enviados lembretes para reforçar a importância da pesquisa. Por fim, na etapa de análise os dados coletados foram submetidos a uma análise quantitativa e uma análise qualitativa. A análise quantitativa deu-se por meio da estatística descritiva, enquanto que a análise qualitativa foi realizada utilizando a codificação aberta e axial (STRAUSS; CORBIN, 2007).

Figura 1 – Passos metodológicos da pesquisa

---

<sup>1</sup> <https://forms.office.com/>



Fonte: O autor (2022)

No que diz respeito ao catálogo, este foi construído a partir das abordagens identificadas no MSL. Para cada abordagem foram realizadas pesquisas sobre suas principais características e a sua forma de utilização. A principal referência para cada

abordagem foi o próprio artigo selecionado no MSL. Contudo, para algumas abordagens outras buscas foram realizadas a fim de complementar o conteúdo abordado pelos autores dos trabalhos que desenvolveram e/ou aplicaram a abordagem. A partir disso, o catálogo foi estruturado em cinco tópicos, da seguinte forma:

- **Fornecedor / Desenvolvedor:** esse tópico apresenta os responsáveis pelo desenvolvimento e/ou manutenção da abordagem, seja pessoa física ou alguma empresa/organização;
- **Site de acesso:** esse tópico traz um *link* onde o leitor pode consultar mais informações sobre a abordagem, caso tenha interesse;
- **Descrição:** esse tópico apresenta a ferramenta, isto é, na descrição a abordagem é definida de acordo com o que foi identificado nos artigos do MSL, bem como informações adicionais retiradas de outras fontes complementares;
- **Finalidade:** esse tópico traz ao leitor o objetivo principal para o qual a abordagem foi concebida;
- **Como funciona:** esse tópico apresenta um pouco mais da descrição da abordagem e como ela pode ser usada pelos interessados.

## 1.6. Estrutura do Trabalho

Além deste capítulo, que trata sobre a introdução geral do trabalho realizado, do contexto, sua justificativa, sua motivação, seus objetivos e sua metodologia utilizada para a execução deste trabalho, a seguir são descritas as estruturas dos demais capítulos desta dissertação.

O Capítulo 2 apresenta a fundamentação teórica utilizada na construção desse trabalho, que inclui uma descrição detalhada dos tópicos relevantes à pesquisa e dos trabalhos relacionados.

O Capítulo 3 apresenta os resultados da condução do MSL sobre as abordagens de gerenciamento de projeto usadas no contexto de desenvolvimento ágil de software.

O Capítulo 4 apresenta os resultados da aplicação do *survey* realizado com gerentes de projetos de software.

O Capítulo 5 apresenta o catálogo das abordagens de gerenciamento de projetos que foram identificadas no MSL.

E, finalmente, o Capítulo 6 apresenta a conclusão, as contribuições deste trabalho, bem como suas limitações e a indicação de trabalhos futuros.



## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica desta dissertação, discutindo os seguintes tópicos:

- Gestão de projetos;
- Metodologia ágil;
- Gestão de projeto ágil;
- Tipo de abordagens usadas neste trabalho;
- Trabalhos Relacionados.

### 2.1. Gestão de Projetos

O homem, desde seu surgimento, tem mudado a paisagem por onde passa, seja por meio de grandes plantações de alimentos ou até mesmo por construções que desafiam a mente humana. Essas mudanças constantes requerem, em muitos casos, planejamento para que sejam executados adequadamente, ou seja, o homem sempre gerenciou seus projetos de forma empírica, natural e espontânea. Porém, o gerenciamento de projetos praticado antes de sua formalização não tinha as terminologias e as tecnologias próprias que o tornasse uma disciplina e o distinguisse das demais (CODAS, 1987).

O gerenciamento de projetos como é conhecido hoje surgiu apenas na década de 1960, impulsionado principalmente pela indústria bélica e aeroespacial americana (MARTINS, 2010). Na década seguinte surgem os primeiros softwares para auxiliar no planejamento, no controle e no tratamento dos projetos que aumentavam cada vez mais o volume de atividades a serem executadas, como é o caso do Projacs, Proplan e Ártemis (CODAS, 1987).

Outro personagem importante para o desenvolvimento e difusão do gerenciamento de projetos é o Instituto de Gerenciamento de Projetos (*Project Management Institute*, PMI). O PMI é um instituto fundado em 1969 com o propósito de criar padrões de práticas de gerenciamento de projetos que possam ser aplicadas em qualquer ramo de atividades. O instituto logrou êxito, pois até hoje suas definições e normas são seguidas por milhares de empresas que tem suas metas pautadas em projetos. Seu principal artefato é, sem dúvidas, o Guia de Conhecimento sobre Gerenciamento de Projetos (*Project Management Body of Knowledge*, PMBOK), um guia que reúne as melhores práticas de gerenciamento de projetos usadas e aprovadas ao longo de vários

anos. O guia é, atualmente, a principal referência para os pesquisadores, profissionais da indústria e entusiastas em geral do gerenciamento de projetos (HELDMAN, 2006).

O conceito de projeto e suas aplicações é dado por diversos autores, guias, normas e padrões. Algumas dessas definições que se destacam são dadas pelo PMBOK (2017, p. 13), o qual define projeto como “um esforço temporário empreendido para criar um produto, serviço ou resultado único”. Da mesma forma, a ISO 21500 conceitua projeto como um conjunto único de processos que consiste em atividades coordenadas e controladas com datas de início e fim. Vargas (2018), por sua vez, conceitua projeto como:

[...] um empreendimento não repetitivo, caracterizado por uma sequência lógica de eventos, com início, meio e fim, que se destina a atingir um objetivo claro e definido, sendo conduzido por pessoas dentro de parâmetros predefinidos de tempo, custo, recursos envolvidos e qualidade (VARGAS, 2018, p. 7).

É possível perceber a partir desses conceitos que, apesar de existirem muitas definições sobre o que é um projeto, todas afirmam a mesma coisa: um projeto não é repetitivo e precisa ter um limite para seu término. Caso essas condições não sejam atendidas de forma integral, o que deveria ser um projeto fica totalmente descaracterizado. O PMBOK (2017) também conceitua gerenciamento de projetos que, segundo o guia, é a aplicação de uma série de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto com o propósito de cumprir os requisitos estabelecidos.

O resultado de um projeto é alvo de interesse de uma série de pessoas, que são conhecidas no gerenciamento de projetos como *stakeholders*. Os *stakeholders* “são pessoas que, de alguma forma, serão (ou se consideram) afetadas pelo projeto, seja de forma positiva ou negativa” (KEELING; BRANCO, 2014, p. 4). Dentre esses interessados sobressai-se a figura do gerente de projetos, o profissional que “executa as funções de gestão, planejamento e controle” (MARTINS, 2010, p. 7). Esse profissional também “proporciona direção, coordenação e integração à equipe de projetos[...]”, e ainda “deve garantir que sejam feitas escolhas adequadas entre os requisitos de prazo, custo e desempenho do projeto” (LARSON; GRAY, 2016, p. 7).

## 2.2. Metodologia Ágil

À medida que a complexidade dos softwares aumenta, o limite de tempo e

orçamento tendem a aumentar na mesma proporção. Além do que, os clientes, pressionados pelo mercado cada vez mais competitivo, exigem manutenções constantes em seus produtos de software para que mantenham sempre a qualidade desejada. Nesse cenário, os métodos tradicionais de desenvolvimento estão em desvantagem frente aos métodos ágeis, pois possuem características que tornam o processo de desenvolvimento carregado e não conseguem responder de forma satisfatória às mudanças frequentes e inerentes ao mercado atual (CHAUDHARI; JOSHI; BHONGADE, 2021). Assim, a quantidade de empresas que ainda usam os métodos tradicionais como método de ciclo de vida de desenvolvimento de software é cada vez menor (SINHA; DAS, 2021).

Diante disso, as metodologias ágeis visam acabar ou pelo menos minimizar esses desafios. As metodologias ágeis ganharam força a partir de 2001, quando representantes da *Extreme Programming (XP)*, *Scrum*, *Dynamic System Development Model (DSDM)*, *Adaptive Software Development (ASD)*, *Crystal*, *Feature-Driven Development (FDD)*, *Pragmatic Programming* e outros simpatizantes dos métodos ágeis reuniram-se em Utah e lançaram o Manifesto Ágil. Esse manifesto é composto por quatro valores e 12 princípios. Os valores são (BECK *et al.*, 2001):

- **Indivíduos e interações** mais que processos e ferramentas;
- **Software em funcionamento** mais que documentação abrangente;
- **Colaboração com o cliente** mais que negociação de contratos;
- **Responder as mudanças** mais que seguir um plano.

Ao analisar os valores dispostos no manifesto, percebe-se que o foco não está na documentação e sim no produto que atenda as necessidades do cliente, seja entregue de forma rápida, mas sem perder qualidade. De acordo com Alves (2012), o que caracteriza os métodos ágeis é a forma incremental de desenvolvimento, isto é, as especificações de requisitos acontecem gradativamente, à medida que interações de curta duração são desenvolvidas e pequenas partes do sistema são entregues ao cliente.

No que tange aos princípios, o Manifesto Ágil baseia-se em 12, são eles (BECK *et al.*, 2001):

1. Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado;
2. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente;

3. Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo;
4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto;
5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho;
6. O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face;
7. Software funcionando é a medida primária de progresso;
8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
9. Contínua atenção à excelência técnica e bom design aumenta a agilidade;
10. Simplicidade--a arte de maximizar a quantidade de trabalho não realizado--é essencial;
11. As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis;
12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

Segundo Sinha e Das (2021), ser ágil é ser responsivo ao mercado e ao consumidor, ter a capacidade de responder de forma rápida às suas necessidades, mudando o que foi planejado conforme a circunstâncias possa exigir. Para os autores, os métodos ágeis adaptam-se muito bem a qualquer setor, mas são ainda mais adequados para o desenvolvimento de software, justamente pelas características que essa área possui.

### **2.3. Gestão de Projeto Ágil**

Um projeto de software pode ser definido como um conjunto de atividades que criam um resultado com valor e identificável. Em outras palavras, o gerenciamento de projetos nada mais é do que planejar, executar e monitorar essas atividades. Todavia, esse processo de gerenciamento ainda possui altos custos e altas taxas de falha, o que leva a pesquisadores e profissionais a se empenharem em desenvolver um gerenciamento cada vez mais eficaz e que consiga responder bem às mudanças (DYBA; DINGSOYR, 2015).

Um ambiente volátil acarreta em uma pressão na entrega de projetos de

tecnologia, em especial em projetos de software. As formas tradicionais de gerenciar não são mais adequadas para gerenciar esse tipo de projeto, uma vez que não lidam muito bem com mudanças. Assim, a gestão ágil de projetos ganhou destaque entre os gerentes, pois essa forma de gestão permite com que esses profissionais encarem esses desafios e superem as limitações do gerenciamento de projetos (JAYAWARDENA; EKANAYAKE, 2010).

É sabido que a forma ágil de gerenciar é diferente da tradicional. Contudo, isso não quer dizer que a definição de um projeto ou do gerenciamento de projetos difere significativamente entre projetos preditivos e ágeis, mas que eles são construídos sobre diferentes pressupostos. Há de se entender, ainda que o gerenciamento ágil de projetos não precisa, necessariamente, substituir o gerenciamento de projetos tradicional, pois eles podem se complementar, embora algumas práticas sejam contrastantes (ARTELT, 2021).

## 2.4. Tipos de Abordagens Usadas Neste Trabalho

Um fator crítico que afeta significativamente o sucesso de qualquer projeto é a abordagem aplicada para gerenciá-lo. A abordagem aplicada determina o sucesso na realização do próprio projeto, por isso uma boa escolha é uma das etapas mais importantes antes de iniciar qualquer projeto. É necessário enfatizar que podem ser aplicadas muitas abordagens diferentes de gerenciamento de projetos, que possuem algumas semelhanças, mas também diferenças significativas (LALIC *et al.*, 2021).

Hoje, o mundo é orientado a projetos, e todas as atividades são conduzidas como projetos, não existindo uma abordagem universal que sirva para todas as organizações (CIRIC; LALIC; GRACANIN, 2016). Para COBB (2015), dois fatores chaves influenciam na adoção de uma abordagem de gerenciamento de projetos: (A) a tecnologia que está mudando rapidamente; e (B) a natureza dos projetos que está mudando. O autor ainda afirma que a profissão de gerente de projeto está passando por mudanças e aqueles profissionais que virão precisam adquirir novas habilidades e conhecimentos sobre novas abordagens de gestão de projetos.

Assim, a seguir estão definidos os conceitos de cada tipo de abordagem usada nesta dissertação:

- **Software:** é um conjunto de composto por instruções de computador, estrutura de dados e documento associado (PRESSMAN, 2006);

- **Metodologia:** uma metodologia de gerenciamento de projetos é, na verdade, uma estrutura e um processo que uma determinada organização exige ou recomenda para gerenciar seus próprios projetos (WHITTON; LANGAN, 2018);
- **Método:** é definido como uma série de atividades racionais e sistematizadas que traça o caminho a ser seguido e auxilia na detecção de erros e tomadas de decisões (MATIAS-PEREIRA, 2010);
- **Modelo:** é a forma ideal, o paradigma, algo a ser seguido e tem como função criar outros como ele. Modelos podem ser definidos também como algo que se modifica a depender do lugar, momento histórico e posição social que são usados (GOUVEIA JR, 1999);
- **Ferramenta:** trata-se de um recurso que auxilia as pessoas e empresas nas tomada de decisões. Assim, ferramentas podem ser usadas nos mais variados negócios. Com ferramentas tem-se maior controle sobre os processos complexos (VENDRUSCOLO; HOFFMANN; FREITAS, 2012);
- **Técnica:** é um conjunto de processos ou preceitos que guiam determinadas ações, seja no campo da ciência, de negócios e até mesmo do entretenimento (ARAÚJO, 2000);
- **Framework:** é uma série de objetos que, de forma conjunta, colaboram com o objetivo de atender a responsabilidades para uma aplicação específica ou um domínio de aplicação (HELM *et al.*, 2000);
- **Prática:** são realizações em ambientes que precisam ser entendidas e absorvidas por quem tem contado com elas. Além disso, práticas para terem sucesso precisam ser encaradas como algo auto-organizável, após seu entendimento pelos interessados (GARFINKEL, 2006).

#### 2.4.1. Trabalhos Relacionados

Masso *et al.* (2020) conduziram uma RSL focada no gerenciamento dos riscos de um projeto. O intuito do trabalho é caracterizar e apresentar o estado da arte da área, bem como identificar lacunas e oportunidades de novas pesquisas. Nesse estudo, os autores chegaram à conclusão de que existe ainda uma deficiência na utilização de normas, modelos e softwares que possibilitem a minimização dos riscos negativos que cercam um projeto, além de ser mínima a contribuição científica que foque em abordagens capazes de mudar esse cenário. Para os autores, isso abre caminho para que

novas pesquisas sejam realizadas, buscando identificar lacunas e propor novas soluções que incorporem outras áreas do conhecimento. Apesar dos resultados apresentados por Masso *et al.* (2020) caracterizarem-se como resultados importantes para a área da gestão de projetos, uma vez que a área de riscos está inserida dentro de um contexto maior, o trabalho foca apenas nessa área especificamente. Este trabalho, porém, não foca apenas em uma área da gestão de projetos, o objetivo deste trabalho é identificar abordagens que sejam utilizadas em todas as áreas que compõem um projeto de desenvolvimento de software.

Bajta *et al.* (2018) realizaram um MSL para identificar e classificar estudos sobre abordagens de gerenciamento de projetos de software no contexto do Desenvolvimento Global de Software (*Global Software Development, GSD*). Os autores identificaram, a partir da análise de 84 artigos, os pontos fortes e fracos de cada abordagem e analisaram suas aplicações na indústria. Os resultados obtidos no trabalho indicam que o interesse em gerenciamento de projetos de software para o GSD tem aumentado desde 2006. Foi constatado ainda que os métodos relatados com mais frequência (40%) são aqueles usados para coordenação, planejamento e monitoramento, junto com técnicas de estimativa de tempo e custos. Nota-se que os autores realizaram uma pesquisa semelhante a que foi realizada nesta dissertação, porém apesar de terem analisado a aplicação das abordagens na indústria de desenvolvimento de software, a análise dessa aplicação deu-se apenas por meio da RSL. Enquanto isso, neste trabalho foi realizado um *survey* com profissionais da área a fim de coletar informações que serão, posteriormente, confrontadas com aquelas obtidas por meio do MSL e, assim, verificar se o que está sendo desenvolvido na academia, está sendo utilizado na indústria.

Chadli *et al.* (2016), por meio de um MSL, identificaram 102 ferramentas de gestão de projetos utilizadas no GSD. Os autores identificaram que 77% dessas ferramentas são autônomas, isto é, são ferramentas utilizadas de forma individual, sem a necessidade de outras para complementá-las. Os resultados também indicam que as áreas de gerenciamento de projetos não são adequadamente suportadas pelas ferramentas correspondentes e merecem mais atenção dos construtores de ferramentas. Os autores afirmam que as informações contidas no trabalho abrem oportunidades para avanços futuros tanto por parte de pesquisadores, quanto de desenvolvedores de ferramentas que podem fornecer novas ferramentas capazes de mitigar o efeito da globalização na gestão de projetos de software. Este trabalho, porém, difere-se do que foi realizado pelo autor

mencionado em vários aspectos, principalmente em relação ao tipo de equipe. Os autores focaram apenas em equipes globalmente distribuídas, enquanto este trabalho realizou o estudo com profissionais que atuam em diferentes contextos organizacionais.

Jahan *et al.* (2019) fizeram um levantamento por meio de um *survey* sobre as abordagens de gestão de projetos utilizadas na indústria de desenvolvimento de software do Paquistão. Os autores buscaram identificar as metodologias, os softwares e as práticas que mais frequentemente são adotados no país árabe. O estudo identificou que o Scrum é a metodologia de gestão de projetos mais utilizada para gerenciar os projetos; que o MS Project e o Jira são os softwares mais utilizados pelas equipes; e que para realizar a análise do desempenho dos projetos os gerentes de projetos utilizam com mais frequência a análise SWOT. Os autores tratam de abordagens de forma geral, o que torna o trabalho deles semelhante, nesse quesito, ao que está sendo proposto aqui. Contudo, apesar de identificarem a utilização de algumas metodologias ágeis pela indústria paquistanesa de software, o foco não está voltado exclusivamente para o desenvolvimento ágil de software. Nesse sentido, o presente trabalho difere-se do realizado por Jahan *et al.* (2019) pelo fato de que o foco está centrado unicamente no contexto do desenvolvimento ágil de software, levando em consideração a adoção massiva dessas práticas na indústria de software global (SHASTRI; HODA; AMOR, 2021).

Carvalho, Malcher e Santos (2020) realizaram um *survey* com 75 gerentes de projetos de software com o objetivo de identificar os aplicativos móveis mais utilizados no contexto de gerenciamento de projetos de software. Os autores constataram que os aplicativos mais utilizados pelos profissionais entrevistados são o Trello e o Jira, respectivamente. Além disso, os autores identificaram também que os aplicativos são mais utilizados para auxiliar no gerenciamento das comunicações e do cronograma. Foi possível verificar que os autores sugerem que sejam realizados estudos que ampliem o foco que foi dado no estudo conduzido por eles. Dessa forma, semelhante ao estudo conduzido pelos autores mencionados, este trabalho visa realizar um *survey* com profissionais da área de gestão de projetos de software. No entanto, este trabalho propõe-se a investigar outras abordagens utilizadas no gerenciamento de projetos de software e não fica restrito apenas aos aplicativos móveis.

### **3 ABORDAGENS PARA APOIO AO GERENCIAMENTO DE PROJETOS DE SOFTWARE NO CONTEXTO ÁGIL: UM MAPEAMENTO SISTEMÁTICO DA LITERATURA**

Este capítulo apresenta os resultados do MSL. Inicialmente é apresentada a metodologia empregada em sua condução, onde são descritos os três passos principais. Posteriormente, são apresentados de forma detalhada os resultados obtidos a partir dos estudos primários selecionados.

#### **3.1 Metodologia do Mapeamento Sistemático da Literatura**

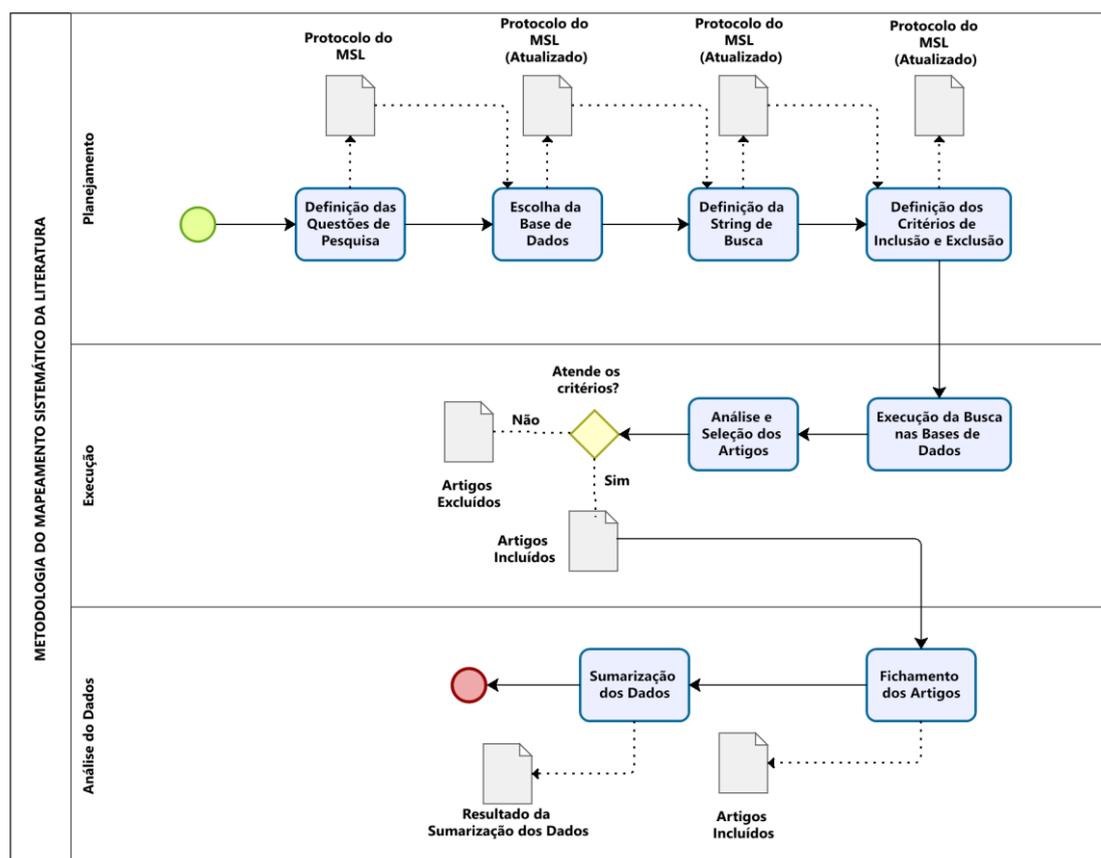
Com o objetivo de identificar as abordagens de gerenciamento de projeto usadas no contexto ágil de desenvolvimento de software e que constam relatadas na literatura especializada, foi realizado um MSL para obter informações sobre o tema. Esse mapeamento abrangeu um período de 20 anos (2001-2021) em bases de pesquisas de destaque internacional por apresentarem dados expressivos e com alto grau de qualidade exigidos em suas publicações. Assim, optou-se pelas bases da ACM Digital Libray e IEEE Xplore.

De posse dos dados levantados pelas questões de pesquisa e análise dos artigos revisados, pôde-se ter um melhor entendimento sobre as abordagens de gerenciamento de projetos. Além disso, os produtos desse MSL foram documentados e analisados para o uso direto nas demais etapas do projeto de pesquisa desta dissertação.

##### **3.1.1 Planejamento do Mapeamento Sistemático da Literatura**

O processo de planejamento do MSL foi realizado com o objetivo de identificar as etapas necessárias para a execução correta da pesquisa, análise e sumarização dos dados. Nesse contexto, foi possível identificar cada atividade necessária, bem como os artefatos que foram utilizados ou gerados pela mesma.

Figura 2 – Passos adotados para o MSL



Fonte: O autor (2022)

Destarte, o planejamento do MSL foi dividido em oito etapas, havendo interações com os artefatos gerados durante o processo que foram atualizados a cada parte da execução. A Figura 2 ilustra o processo por etapas, bem como os artefatos gerados para o cumprimento dos objetivos dessa pesquisa. A execução do MSL ocorreu de acordo com o planejamento supracitado com o objetivo de aprofundar os conhecimentos acerca das abordagens de gerenciamento de projetos. Essas etapas são descritas nas subseções a seguir.

### 3.1.1.1 Definição das Questões de Pesquisa

As questões de pesquisa deste estudo foram definidas seguindo a estrutura proposta por Kitchenham e Charters (2007), denominada PICOC (*Population, Intervention, Comparison, Outcomes e Context*). No entanto, este estudo busca apenas identificar abordagens de gerenciamento de projetos relacionadas na literatura e não se

preocupa em realizar comparações entre elas. Diante disso, o critério de “Comparação” não será utilizado. A estrutura deste trabalho está descrita no Quadro 1.

Quadro 1 - Estrutura PICOC definida para o trabalho

<b>População</b>	Gerência de Projetos.
<b>Intervenção</b>	Identificar, listar e descrever as abordagens de gerenciamento de projetos de software (GPS).
<b>Comparação</b>	Não se aplica.
<b>Resultados</b>	Abordagens utilizadas no gerenciamento de projetos de software, considerando ferramentas, métodos, técnicas, modelos, tecnologias, práticas, padrões, guias, artefatos, metodologias, frameworks, processos, princípios, temas e papéis.
<b>Contexto</b>	Industrial e acadêmico de desenvolvimento ágil de software.

Fonte: O autor (2022)

Assim, foi então definida a questão de pesquisa principal (QP) deste MSL: **“Quais são as abordagens de gerenciamento de projetos utilizadas no para apoiar o desenvolvimento de software no contexto ágil?”**. Para auxiliar na resposta da QP, subquestões (Sub-Q) foram definidas e detalhadas no Quadro 2.

Quadro 2 - Subquestões de pesquisa

<b>Sub-Q1</b>	Como foi a evolução do número de estudos publicados relacionados com o tópico desta pesquisa?
<b>Sub-Q2</b>	Quais os países que mais publicaram trabalhos relacionados com o tópico desta pesquisa?
<b>Sub-Q3</b>	Qual o tipo de instituição que mais publicou trabalhos relacionados com o tópico desta pesquisa?
<b>Sub-Q4</b>	Em qual área de conhecimento do GPS a abordagem é aplicada?
<b>Sub-Q5</b>	Em qual fase do GPS a abordagem é aplicada?
<b>Sub-Q6</b>	Qual é o tipo de abordagem?
<b>Sub-Q7</b>	Quais os pontos fortes e fracos das abordagens?

<b>Sub-Q8</b>	Como foi feita a avaliação das abordagens?
---------------	--

Fonte: O autor (2022)

### 3.1.1.2 Escolha das Bases de Dados

A seleção das fontes deu-se seguindo os critérios de seleção definidos no Quadro 3.

Quadro 3 - Critérios de seleção das fontes de busca

<b>Id.</b>	<b>Critério</b>
CS1	Disponibilidade para consultas web
CS2	Disponibilidade para busca de estudos por meio do portal de periódicos da CAPES, utilizando o domínio da UFPA.
CS3	Disponibilidade de estudos na íntegra através do portal de periódicos da CAPES, utilizando o domínio da UFPA ou a partir da utilização da <i>engine</i> de busca Google e/ou Google Scholar.
CS4	Disponibilidade de estudos em inglês.
CS5	Relevância da fonte.

Fonte: O autor (2022)

As fontes adotadas para este estudo são as que possuem artigos na íntegra, a fim de minimizar o retrabalho dos pesquisadores em buscar manualmente artigos que disponibilizam apenas título e *abstract*. No entanto, nos casos em que não foi possível baixar o artigo na íntegra, foram realizadas buscas manuais no Google<sup>2</sup> e Google Scholar<sup>3</sup> com o objetivo de encontrar a versão completa do artigo.

Seguindo os critérios de seleção definidos no Quadro 3, foram selecionadas as fontes apresentadas no Quadro 4.

Quadro 4 - Fontes de busca selecionadas

<b>Fonte de busca</b>	<b>Link</b>
-----------------------	-------------

<sup>2</sup> <http://www.google.com.br/>

<sup>3</sup> <http://scholar.google.com.br/>

ACM Digital Library	<a href="https://dl.acm.org/">https://dl.acm.org/</a>
IEEE Xplore	<a href="https://ieeexplore.ieee.org/">https://ieeexplore.ieee.org/</a>

Fonte: O autor (2022)

As duas fontes apresentadas no Quadro 4 foram selecionadas por possuírem um grande acervo de publicações na área de interesse deste estudo e também por indexarem anais de eventos importantes da computação. Importante salientar que não foram realizadas buscas no EI COMPENDEX e SCOPUS porque, de acordo com os resultados de Souza *et al.* (2018), há uma alta taxa de redundância de trabalhos nas respectivas bases.

### 3.1.1.3 Definição da *String* de Busca

Com base nas questões de pesquisa, algumas palavras-chave foram definidas a partir da estrutura PICOC. Porém, como já mencionado na seção 3.1.1.1, o critério “Comparação” foi descartado. Além disso, como o critério “Intervenção” trata das ações que serão realizadas, não há a necessidade de incluir palavras-chave para ele. Diante disso, temos as seguintes palavras-chave (Quadro 5).

Quadro 5 - Palavras chaves utilizadas para a elaboração da string de busca

<b>CRITÉRIO</b>	<b>PALAVRAS-CHAVE</b>
<b>População</b>	Inglês: <i>project management</i> Português: gerenciamento de projetos de software
<b>Resultado</b>	Inglês: <i>software, tool, method, technique, model, technology, practice, standard, guide, work product, methodology, framework, process, principle, theme and profile</i> Português: software, ferramenta, método, técnica, modelo, tecnologia, prática, padrão, guia, artefato, metodologia, framework, processo, princípio, tema e papel

CRITÉRIO	PALAVRAS-CHAVE
Contexto	Inglês: <i>software and agile</i> Português: software e ágil

Fonte: O autor (2022)

Os termos de pesquisa que conduziram este processo foram reunidos em uma *String* de Busca (SB). Esses termos foram escritos em inglês por ser largamente adotado em conferências e periódicos internacionais relacionados ao tema da pesquisa. Para a elaboração da SB as palavras-chave e seus sinônimos foram agrupados por meio do operador lógico OR. Enquanto isso, os conjuntos de termos foram concatenados por meio do operador lógico AND. O primeiro conjunto de termos da SB trata da população que, de alguma forma, é impactada pelas abordagens de gerenciamento de projetos. O segundo conjunto de termos trata dos resultados que se espera com a execução das buscas, ou seja, abordagens que são aplicadas na indústria de software e relatadas na literatura. Por fim, o terceiro conjunto de termos refere-se ao contexto no qual essas abordagens são aplicadas. A SB e os operadores lógicos utilizados podem ser visualizados no Quadro 6.

Quadro 6 - *String* de busca utilizada

String de Busca (SB)
<p>(“<i>project management</i>”) AND (“<i>software</i>”, “<i>tool</i>” OR “<i>method</i>” OR “<i>technique</i>” OR “<i>model</i>” OR “<i>technolog*</i>” OR “<i>practice</i>” OR “<i>pattern</i>” OR “<i>standard</i>” OR “<i>guide</i>” OR “<i>work product</i>” OR “<i>methodolog*</i>” OR “<i>framework</i>” OR “<i>process</i>” OR “<i>principle</i>” OR “<i>theme</i>” OR “<i>profile</i>”) AND (“<i>software</i>”) AND (“<i>agile</i>”)</p>

Fonte: O autor (2021)

Após a elaboração da SB, ela foi testada para verificar sua eficácia. O teste consistiu em rodar a *string* em uma das bases e verificar se três trabalhos retornariam. Esses trabalhos foram selecionados após a constatação de que eles atendem os critérios de inclusão definidos no Quadro 7. Os trabalhos são:

- **IEEE XPLORE**

CARNEIRO, L. B.; SILVA, A. C. C. L. M.; ALENCAR, L. H.. Scrum Agile Project Management Methodology Application for Workflow Management: a case study. *In*: 2018 IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND ENGINEERING MANAGEMENT (IEEM), 12., 2018, Bangkok. **2018 IEEE**

**International Conference on Industrial Engineering and Engineering Management (IEEM).** [S.L.]: IEEE, 2018. p. 938-942.

SILVA, Lilian Santos Ferreira da; OLIVEIRA, Sandro Ronaldo Bezerra. A Process Framework with Agile Practices for Implementation of Project Portfolio Management Process. *In: 2016 10TH INTERNATIONAL CONFERENCE ON THE QUALITY OF INFORMATION AND COMMUNICATIONS TECHNOLOGY (QUATIC), 10., 2016, Lisbon. 2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC).* [S.L.]: IEEE, 2016. p. 146-149.

HAYAT, Faisal; REHMAN, Ammar Ur; ARIF, Khawaja Saemad; WAHAB, Kanwal; ABBAS, Muhammad. The Influence of Agile Methodology (Scrum) on Software Project Management. *In: 2019 20TH IEEE/ACIS INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, ARTIFICIAL INTELLIGENCE, NETWORKING AND PARALLEL/DISTRIBUTED COMPUTING (SNPD), 20., 2019, Toyama. 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD).* [S.L.]: IEEE, 2019. p. 145-149.

- **ACM DIGITAL LIBRARY**

HOUSTON, Dan X.; BUETTNER, Douglas J.. Modeling user story completion of an agile software process. *In: THE 2013 INTERNATIONAL CONFERENCE, 8., 2013, New York, New York, USA. Proceedings of the 2013 International Conference on Software and System Process - ICSSP 2013.* New York, New York, USA: ACM Press, 2013. p. 88-97.

RAITH, Florian; RICHTER, Ingo; LINDERMEIER, Robert. How Project-management-tools are used in Agile Practice. *In: THE 21ST INTERNATIONAL DATABASE ENGINEERING & APPLICATIONS SYMPOSIUM, 21., 2017, New York, New York, USA. Proceedings of the 21st International Database Engineering & Applications Symposium on - IDEAS 2017.* New York, New York, USA: ACM Press, 2017. p. 30-39.

LYRA, Kamila Takayama; ALVES, Marcela Lopes; SILVA, Fernando H. Carvalho; SOUZA, Kathiani; ISOTANI, Seiji. An agile project management experience. *In: THE XXXII BRAZILIAN SYMPOSIUM, 32., 2018, New York, New York, USA. Proceedings of the XXXII Brazilian Symposium on Software Engineering - SBES '18.* New York, New York, USA: ACM Press, 2018. p. 240-249.

#### 3.1.1.4 Definição dos Critérios de Inclusão e Exclusão

Kitchenham e Charters (2007) afirmam que, para selecionar as publicações que retornam com a execução das buscas, é necessário definir e aplicar sobre esses trabalhos alguns critérios de inclusão (CI) e exclusão (CE). As publicações foram incluídas se

atendessem pelo menos um dos CI e excluídos quando se enquadraram em ao menos um dos CE. Os Quadros 7 e 8 apresentam os CI e os CE, respectivamente.

Quadro 7 - Critérios de inclusão

<b>Id.</b>	<b>Critérios de Inclusão</b>
CI1	Publicações que apresentam alguma abordagem de gerenciamento de projetos e que foram aplicadas no contexto industrial e/ou acadêmico de desenvolvimento ágil de software.
CI2	Publicações que realizaram a avaliação dessas abordagens.

Fonte: O autor (2022)

Quadro 8 - Critérios de exclusão

<b>Id.</b>	<b>Critérios de Exclusão</b>
CE1	Estudos que não atendam pelo menos um critério de inclusão.
CE2	Estudos repetidos encontrados em diferentes mecanismos de pesquisa. Nesse caso, apenas um estudo será considerado.
CE3	Estudos não disponíveis para download abertamente ou pelo IP institucional dos pesquisadores.
CE4	Estudos que não estejam escritos em inglês.
CE5	Publicações do tipo relatórios de workshops, pôster, apresentações, <i>keynotes speaker</i> , livros, teses e dissertações.
CE6	Artigos publicados fora do período definido para a busca.
CE7	Artigos que claramente não atendam as questões de pesquisa.

Fonte: O autor (2022)

### 3.1.1.5 Execução da Busca nas Bases de Dados

Para a execução deste MSL os seguintes recursos foram utilizados:

1. Dois pesquisadores (um aluno de mestrado e um professor Dr. em Engenharia de Software);
2. Acesso às fontes de pesquisa por meio do domínio da Universidade Federal do Pará;

3. Validações sobre documentos e procedimentos da realização do MSL através de reuniões com o coordenador do Projeto SPIDER e orientador desta dissertação, onde esta pesquisa está inserida.

#### 3.1.1.6 Análise e Seleção dos Artigos

Cada um dos estudos passou por um processo de seleção composto por quatro etapas:

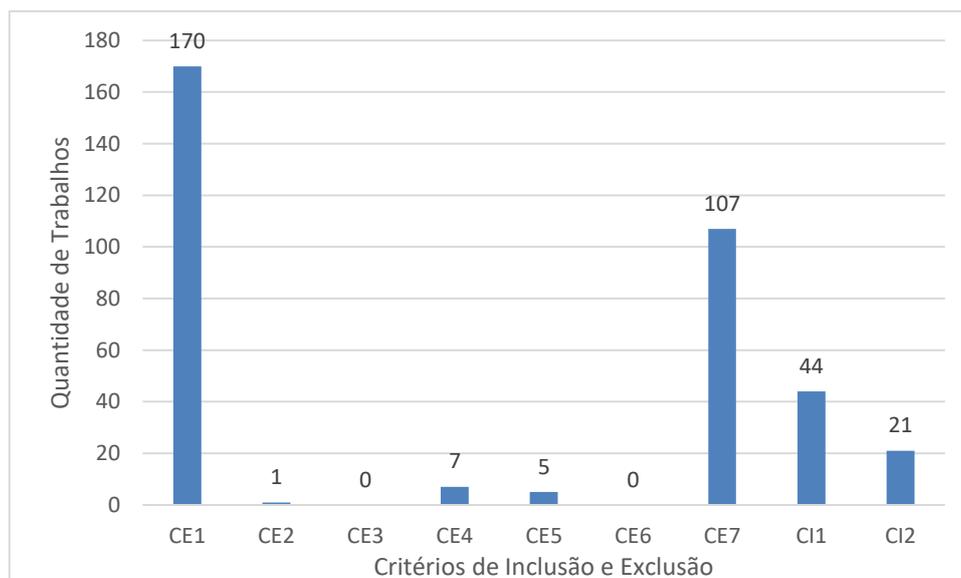
- (i) dois pesquisadores leram os títulos e resumos de todos os estudos e aplicaram os critérios de exclusão, esta etapa foi definida como pré-seleção;
- (ii) os mesmos pesquisadores discutiram sobre divergências na aplicação dos critérios de exclusão para chegar a um consenso;
- (iii) os pesquisadores leram o título e resumo, e o texto completo, se necessário, dos estudos selecionados na primeira etapa para aplicar os critérios de inclusão;
- (iv) os pesquisadores discutiram sobre divergências na aplicação de critérios de exclusão para chegar a um consenso. O processo descrito resultou em 65 estudos primários, identificados neste trabalho por meio do código (EP00 – Estudo Primário e o número sequencial), onde suas referências estão disponíveis no Apêndice 1.

## 3.2 Resultados do Mapeamento Sistemático da Literatura

### 3.2.1 Resultados Bibliométricos

Ao aplicar a *string* de busca nas bases selecionadas, foram retornados um total de 2825 trabalhos, sendo que desse total, 1486 retornaram da ACM Digital Library e 1339 retornaram da IEEE Xplore. Após a busca foram removidos 161 trabalhos duplicados, restando, portanto, 2664 trabalhos totais. Após a leitura dos títulos e resumos de cada um dos trabalhos, foram selecionados 355 trabalhos, 251 da ACM e 104 da IEEE. O passo seguinte foi ler os trabalhos na íntegra e aplicar os critérios estabelecidos e descritos na seção 3.1.1.4. A Figura 3 mostra a quantidade de trabalhos incluídos e excluídos por cada um dos critérios utilizados.

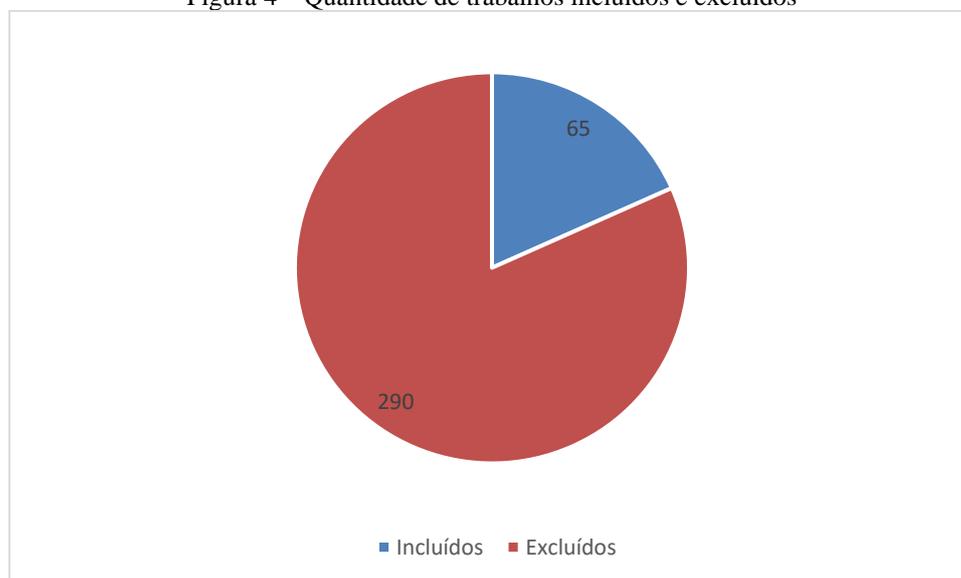
Figura 3 – Quantidade de trabalhos incluídos e excluídos por cada critério



Fonte: O autor (2022)

A Figura 4 mostra a quantidade de trabalhos incluídos e excluídos após a leitura dos trabalhos na íntegra. Dentre os 65 trabalhos selecionados, 40 são provenientes da ACM e 25 da IEEE.

Figura 4 – Quantidade de trabalhos incluídos e excluídos

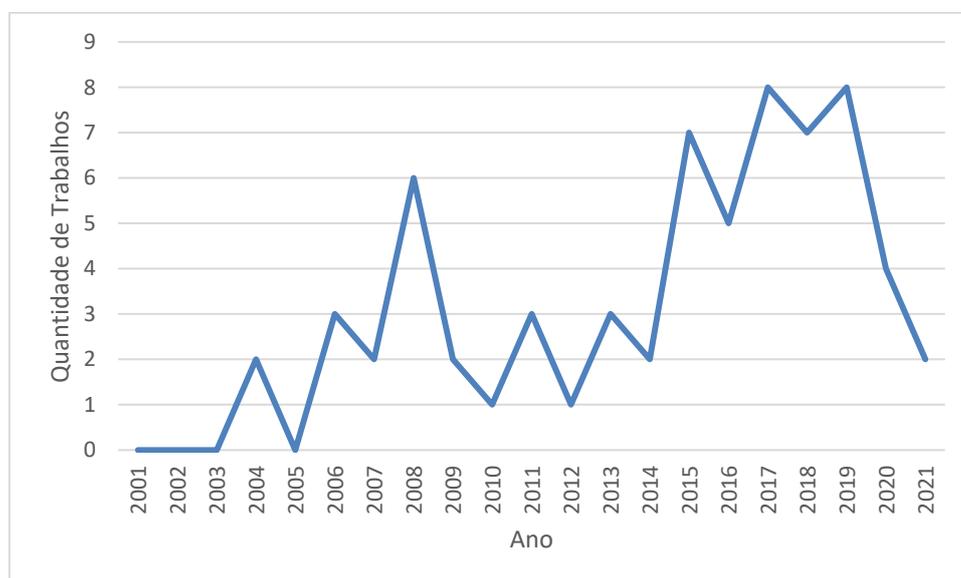


Fonte: O autor (2022)

**3.2.2 Sub-Q1:** Como foi a evolução do número de estudos publicados relacionados com o tópico desta pesquisa?

O MSL buscou trabalhos entre os anos de 2001 e 2021. Porém, os 65 estudos primários selecionados estão distribuídos entre os anos de 2004 e 2021, como mostra a Figura 5. Ainda com base na Figura 5, é possível notar que, apesar de uma queda na quantidade de estudo após 2008, a tendência é de crescimento no número de publicações relacionadas ao tema deste trabalho, com um pico mais acentuado a partir de 2015. Porém, nota-se ainda que os anos de 2020 e 2021 apresentaram uma queda expressiva no número de publicações.

Figura 5 – Distribuição dos artigos por ano



Fonte: O autor (2022)

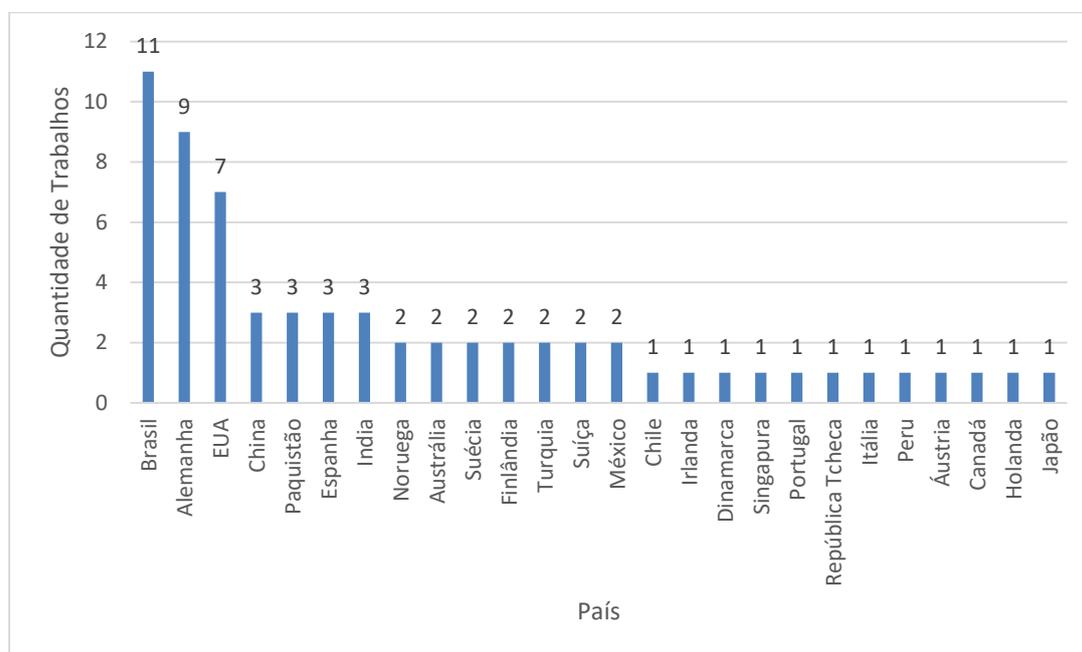
De acordo com Yanow e Good (2020), houve uma redução no número de publicações em 2020, pois muitas universidades e empresas precisaram reduzir suas atividades de pesquisas, haja vista que os laboratórios foram fechados durante o período da pandemia de COVID-19. Há de se levar em consideração também que o número de estudos em 2021 é menor em consequência do período de execução deste trabalho.

**3.2.3 Sub-Q2:** Quais os países que mais publicam trabalhos relacionados com o tópico desta pesquisa?

No que tange aos países que mais publicaram trabalhos na área, a Figura 6 aponta que o Brasil, Alemanha e Estados Unidos foram os países que mais publicaram trabalhos

relacionados com o tema desta pesquisa, com 11, 9 e 7 trabalhos, respectivamente. Por outro lado, países como Holanda, Japão e Peru tiveram apenas 1 trabalho publicado cada.

Figura 6 – Quantidade de trabalhos distribuídos por país

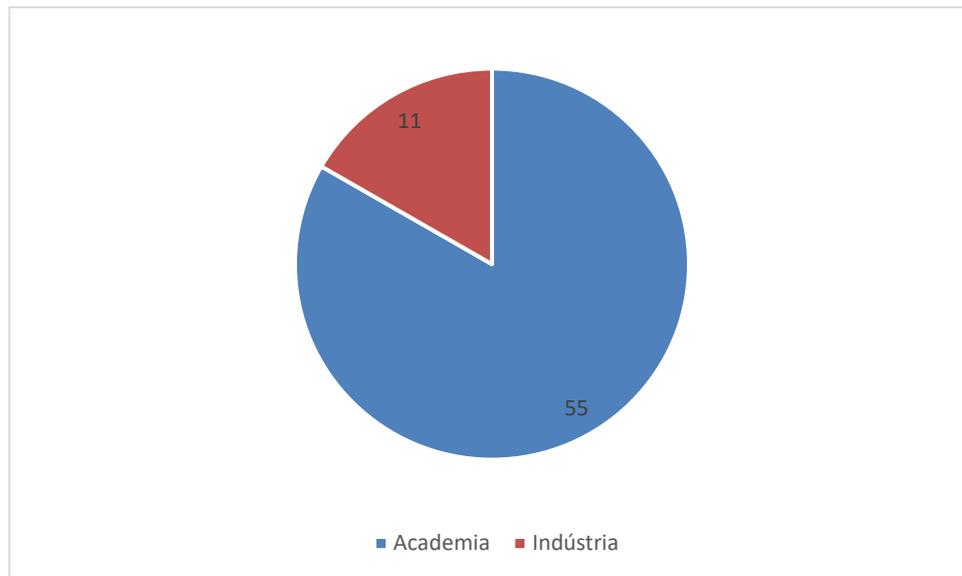


Fonte: O autor (2022)

### 3.2.4 Sub-Q3: Qual o tipo de instituição que mais publicaram trabalhos relacionados com o tópico desta pesquisa?

Outro dado levantado a partir dos trabalhos selecionados trata-se do tipo de instituição a qual o autor principal do trabalho é proveniente. Como pode ser observado na Figura 7, a academia é responsável pela maioria dos trabalhos publicados que estão relacionados ao tema, com um total de 54 trabalhos. Enquanto isso, a indústria publicou um total de 11 trabalhos referentes ao tema desta pesquisa.

Figura 7 – Distribuição dos artigos por tipo de instituição

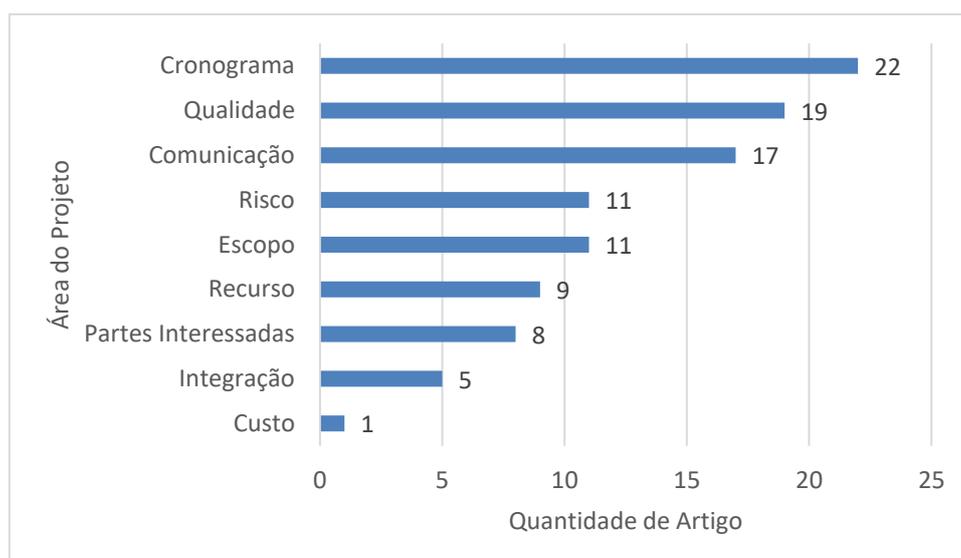


Fonte: O autor (2022)

### 3.2.5 Sub-Q4: Em qual área de conhecimento do GPS a abordagem é aplicada?

A Figura 8 apresenta as áreas do projeto onde as abordagens identificadas são usadas. Com base nessa figura, é possível verificar que o foco das abordagens está principalmente voltado para as áreas de cronograma (22 estudos), qualidade (19 estudos) e comunicação (17 estudos), como é o caso dos seguintes estudos primários: [EP61], [EP59] e [EP30], respectivamente.

Figura 8 – Áreas do projeto onde as abordagens são aplicadas

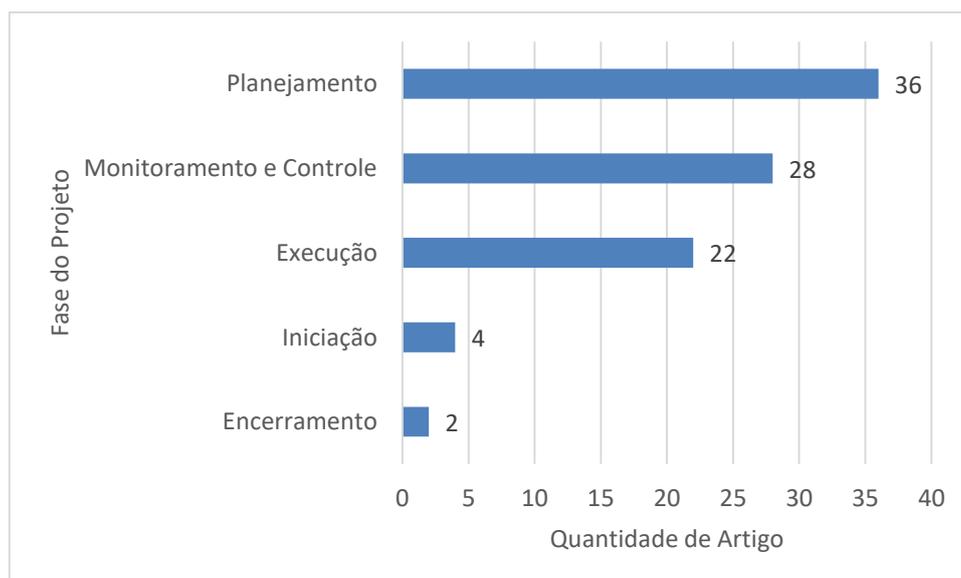


Fonte: O autor (2022)

### 3.2.6 Sub-Q5: Em qual fase do GPS a abordagem é aplicada?

Com relação às fases do projeto em que são utilizadas as abordagens identificadas neste trabalho, foi possível notar que as abordagens dão mais ênfase à fase de planejamento do projeto (36 estudos), além do monitoramento e controle (28 estudos) do mesmo, como pode ser exemplificado por [EP64] e [EP10], respectivamente.

Figura 9 – Fase do projeto onde as abordagens são aplicadas



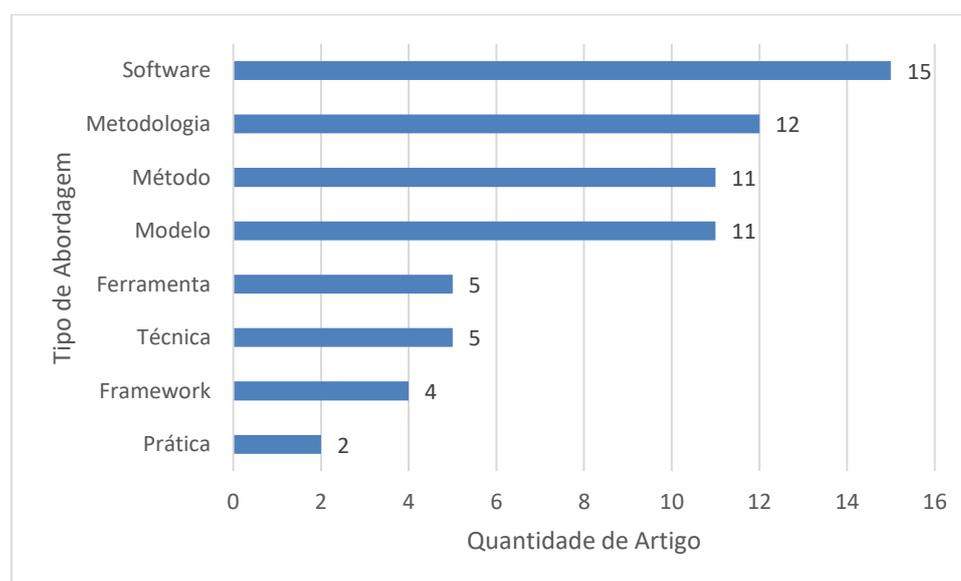
Fonte: O autor (2022)

É importante notar que alguns estudos focam em mais de uma fase do projeto ao mesmo tempo, como é o caso, por exemplo, dos seguintes estudos: [EP07, EP12, EP28, EP54, EP65]. O mesmo vale para as áreas do projeto, pois, assim como nas fases, há estudos que são focados em mais de uma área simultaneamente, por exemplo: [EP04, EP34, EP39, EP47, EP57].

### 3.2.7 Sub-Q6: Qual é o tipo de abordagem?

A partir dos estudos selecionados foi possível identificar oito tipos de abordagens, sendo elas: software, metodologia, método, modelo, ferramenta, técnica, *framework* e prática. A Figura 10 ilustra os tipos de abordagens identificadas, bem como a quantidade de estudos que trata de cada uma delas.

Figura 10 – Tipo de abordagem identificada nos trabalhos selecionados



Fonte: O autor (2022)

Ainda com base na Figura 10, é possível verificar que “software” é o tipo de abordagem mais comum utilizada para apoiar o gerenciamento de projetos de software, sendo essa abordagem identificada em 15 estudos. Logo em seguida, abordagens do tipo “metodologia” é a que mais se destaca, desenvolvida e/ou utilizada em 12 estudos. Por fim, dentre os tipos de abordagens que são mais utilizadas, “método” aparece em terceiro lugar, sendo identificado esse tipo de abordagem em 11 estudos primários.

Vale destacar que, abordagens do tipo “*framework*” e “prática” são as menos utilizadas, de acordo com os dados extraídos dos estudos primários. Tais abordagens foram observadas apenas em quatro e dois estudos, respectivamente.

### ❖ Software

Como mencionado anteriormente, software é o tipo de abordagem mais comum utilizada para apoiar o gerenciamento de projetos de software. O Quadro 9 apresenta os estudos que apresentam esse tipo de abordagem.

Quadro 9 – Artigos que apresentam abordagem do tipo “Software”

<b>Estudos que apresentam abordagem “Software”</b>
[EP01], [EP02], [EP06], [EP07], [EP10], [EP14], [EP16], [EP21], [EP30], [EP40], [EP43], [EP44], [EP52], [EP61], [EP62]

Fonte: O autor (2022)

Morgan e Maurer [EP44] relatam em seu estudo o uso do MasePlanner. MasePlanner trata-se de um software voltado para o planejamento da comunicação no projeto. É um software que, segundo os autores, oferece suporte às interações, facilitando a comunicação não verbal. O MasePlanner disponibiliza às equipes um ambiente digital que oferece suporte ao gerenciamento de informações, além de interações naturais. Com relação ao suporte para interação natural, o MasePlanner permite que os artefatos de planejamento sejam criados, editados e organizados de maneira semelhante às reuniões de planejamento em papel.

Mais recentemente, Alhazmi e Huang [EP01] desenvolveram o Sistema de Apoio à Decisão de Planejamento de Sprint (SPESS). Esse software tem como objetivo auxiliar os gestores no planejamento de Sprint. O SPESS baseia-se, principalmente, em três fatores: a competência do desenvolvedor, a antiguidade do desenvolvedor e a dependência da tarefa. Esta ferramenta visa atribuir as tarefas de cada Sprint aos desenvolvedores garantindo que cada membro da equipe contribua com o máximo de seu potencial, e o planejamento do projeto seja otimizado para o menor tempo possível.

Os demais estudos que utilizam software estão voltados para a comunicação [EP06, EP10, EP30, EP40, EP52], qualidade [EP02, EP07, EP14, EP16, EP62], cronograma [EP07, EP14, EP43, EP61], escopo [EP07] e partes interessadas [EP21]. Vale

destacar que um software não necessariamente atende apenas a uma área do projeto. Begosso *et al.* [EP07], por exemplo, apresentam o SimScrumF, que trata-se de um jogo que foca na promoção do engajamento de alunos no processo de aprendizagem dos conceitos da metodologia Scrum, amplamente utilizada para gerenciar projetos de desenvolvimento de software ao redor do mundo. O jogo aborda o processo de gerenciamento do escopo definido para o projeto, visando entregar um produto com qualidade, dentro do cronograma pré-estabelecido.

### ❖ Metodologia

Atrás somente dos softwares, as metodologias foram as abordagens mais utilizadas para apoiar o gerenciamento de projetos. O Quadro 10 apresenta os estudos que têm como foco esse tipo de abordagem.

Quadro 10 - Artigos que apresentam abordagem do tipo “Metodologia”

<b>Estudos que apresentam abordagem “Metodologia”</b>
[EP04], [EP09], [EP12], [EP13], [EP17], [EP20], [EP28], [EP37], [EP39], [EP47], [EP53], [EP54]

Fonte: O autor (2022)

Castillo-Barrera *et al.* [PS13] afirmam que antes de iniciar a execução do projeto é necessário ter realizado previamente uma análise e também uma síntese das informações que o permeiam. Para tanto, os autores apresentam o BloomSoft, uma metodologia adaptada da Taxonomia de Bloom e utilizada em conjunto com o Scrum, que visa apoiar as equipes no planejamento da construção, integração e teste do software que será desenvolvido no projeto.

Não só isso, pois os autores também ressaltam que a metodologia possibilita ter uma forma ágil de classificar a complexidade das histórias de usuários com base nos verbos nelas identificados e, concomitantemente, determinar a partir dessa classificação o estágio de Desenvolvimento de Software para o qual ele pertence. Assim, a equipe de desenvolvimento tem a possibilidade de classificar as histórias em etapas e, com isso, fazer um melhor planejamento para cada sprint.

Bierwolf *et al.* [EP09] relatam o uso da metodologia DevOps na gestão de projetos de software. O DevOps visa mitigar os riscos, em particular para alcançar um ambiente de produção estável, seguro e confiável. Por meio dessa forma de comunicação

e colaboração todas as partes interessadas gerenciam as incertezas que surgem, por exemplo, devido a mudanças na tecnologia ou no ambiente. O trabalho também realiza uma comparação entre os resultados da utilização de abordagens tradicionais (cascata, prototipação e espiral) em relação às abordagens de DevOps. A comparação está relacionada, principalmente, a aspectos como controle e mitigação de incertezas e riscos.

Há também uma metodologia que varia do Scrum e é apresentada por Baptista [EP04], o uScrum, metodologia descrita no estudo citado, que gerencia a incerteza e o desconhecido, permitindo que a equipe envolvida no projeto reaja rapidamente às mudanças nas mais diversas condições. O uScrum permite que a equipe priorize efetivamente o trabalho regular ao lado do trabalho criativo mais difícil. A metodologia prioriza as tarefas em iterações mensais com base em sua importância, urgência e pontualidade e, assim como o Scrum, também são chamadas de sprints.

#### ❖ Método

Outro tipo de abordagem que se destaca são os métodos, pois esse tipo de abordagem foi identificado em 11 dos 65 estudos selecionados e, junto com os modelos, é o terceiro mais utilizado para apoiar a gestão de projetos de software. O Quadro 11 apresenta os estudos que tratam de métodos.

Quadro 11 - Artigos que apresentam abordagem do tipo “Método”

<b>Estudos que apresentam abordagem “Método”</b>
[EP03], [EP05], [EP08], [EP11], [EP15], [EP23], [EP26], [EP31], [EP50], [EP56], [EP64]

Fonte: O autor (2022)

Haugen [EP26] descreve o uso do *Planning Poker*. O objetivo do estudo é verificar se a introdução desse método na estimativa de histórias de usuários melhora o desempenho da estimativa em comparação com a estimativa de grupo não estruturada. Nesse processo, o cliente primeiro explica cada história de usuário ao grupo de desenvolvedores. Os desenvolvedores então discutem o trabalho envolvido na implementação até o ponto em que todos sentem que têm informações suficientes para estimar o esforço necessário. Todos os desenvolvedores então estimam a história do usuário de forma independente e revelam suas estimativas simultaneamente. Em seguida, os desenvolvedores com as estimativas mais baixas e mais altas justificam suas

estimativas e o grupo continua a discussão para decidir sobre uma estimativa coletiva, possivelmente realizando uma ou mais rodadas adicionais de estimativas individuais.

Zhang *et al.* [EP64] abordam em seu estudo sobre *Early Software Size Estimation* (ESSE), um método que pode extrair recursos semânticos de requisitos de linguagem natural automaticamente e construir modelos de estimativa de tamanho para o projeto. O ESSE faz uma análise semântica dos documentos de especificação de requisitos por extração de informações e disseminação de ativação. Então, características relacionadas à complexidade são extraídas dos resultados da análise semântica. Além disso, o ESSE extrai recursos locais e recursos globais para fazer análise semântica em nível de palavra. Então, usando esses recursos, direcionadores de tamanho e tamanhos reais de dados históricos do projeto, o modelo de estimativa de tamanho pode ser estabelecido por algoritmos de regressão. Finalmente, a ESSE pode estimar o tamanho de um novo projeto usando este modelo de estimativa de tamanho.

Os demais estudos que utilizam algum método de gerenciamento de projetos de software estão preocupados em gerenciar a qualidade [EP05, EP50], cronograma [EP11], recursos [EP31, EP56], escopo [EP03] e os riscos do projeto [EP23]. Além disso, esses mesmos estudos estão voltados para as seguintes fases do projeto: planejamento [EP11, EP23, EP31], execução [EP50] e monitoramento e controle [EP03, EP05, EP23, EP31, EP56].

## ❖ Modelo

Além dos métodos, foi possível identificar, por meio dos estudos selecionados, 11 modelos que visam apoiar a gestão de projetos de software no contexto ágil. O Quadro 12 apresenta os estudos que focam nesse tipo de abordagem.

Quadro 12 - Artigos que apresentam abordagem do tipo “Modelo”

<b>Estudos que apresentam abordagem “Modelo”</b>
[EP22], [EP25], [EP27], [EP29], [EP34], [EP36], [EP46], [EP48], [EP55], [EP58], [EP59]

Fonte: O autor (2022)

Godoy *et al.* [EP22] apresentam uma nova ferramenta de desenvolvimento de software baseada nas metodologias ágeis Scrum e Kanban e adaptada à tendência atual do ambiente global de software. O modelo *Blueprint* propõe um gerenciamento de

projetos leve que é combinado com uma organização de equipes para incentivar e facilitar a comunicação entre equipes em diferentes locais. O *Blueprint* introduz importantes adaptações ao Scrum e Kanban para reduzir a burocracia indesejada e facilitar o GSD.

Perkusich *et al.* [EP48] desenvolveu um Modelo Probabilístico com Rede Bayesiana. Como o nome sugere, o modelo é uma rede Bayesiana que representa um projeto de desenvolvimento de software gerenciado em essência com a metodologia Scrum. Os Scrum Masters devem usá-lo para identificar problemas do projeto e orientar a equipe para melhorar as chances de sucesso do projeto. O modelo produz dados com valores de probabilidade que representam o status atual dos principais fatores do projeto. Deve ser usado para identificar problemas e priorizar áreas de melhoria. A priorização de áreas de melhoria deve ser uma atividade colaborativa e o modelo deve ser utilizado apenas como fonte de informação para orientar a discussão.

Os demais estudos que tratam de algum modelo focam no gerenciamento do cronograma do projeto [EP34, EP36, EP58]. Além dos estudos citados, há aqueles voltados para gestão de risco [EP27, EP46], comunicação [EP25], escopo [EP29], *stakeholders* [EP55] e qualidade [EP59]. Esses mesmos estudos ainda focam em diferentes fases do projeto, como: planejamento [EP25, EP46, EP58, EP59], execução [EP36, EP55] e monitoramento e controle [EP27, EP29].

## ❖ Ferramenta

Com base nos estudos selecionados, foram identificadas cinco ferramentas que auxiliam a gestão de projetos de software. É importante destacar que, este trabalho nomeou como ferramentas as abordagens que também foram tratadas assim nos EP's, ou seja, não foram mudadas as nomenclaturas para manter a forma como os autores descreveram em seus respectivos trabalhos. O Quadro 13 apresenta esses estudos.

Quadro 13 - Artigos que apresentam abordagem do tipo “Ferramenta”

<b>Estudos que apresentam abordagem “Ferramenta”</b>
[EP18], [EP38], [EP42], [EP49], [EP63]

Fonte: O autor (2022)

Vivian *et al.* [EP63] expõem em seu estudo um Painel para Discussões de Trabalho em Equipe Online. A ferramenta é um *dashboard* que extrai e comunica a

distribuição dos papéis da equipe e informações emocionais dos membros em tempo real. O painel é composto por uma série de elementos: participação da equipe e distribuição de papéis, análise de sentimento da equipe e individual e emoções da equipe e individuais. Ele fornece análise em tempo real das discussões de trabalho em equipe e visualiza as emoções dos membros da equipe, os papéis que assumiram e o sentimento geral da equipe durante o curso de um projeto colaborativo.

Mateescu *et al.* [EP42] apresentam uma ferramenta chamada aWall. É uma ferramenta ágil de colaboração em equipe para grandes sistemas de parede multitoque. O aWall foi projetado com base em pesquisas empíricas de usuários usando novos conceitos de interação e visualização para apoiar e promover o estilo de trabalho ágil altamente colaborativo e comunicativo. A ferramenta é baseada em tecnologia web e pode ser utilizada tanto em ambientes colocalizados quanto distribuídos.

Segundo Mateescu *et al.* [EP42], a ferramenta pode ser crucial para equipes ágeis, uma vez que o processo ágil depende de intensa interação, colaboração e comunicação aberta constante entre os membros da equipe. Os demais estudos que tratam de ferramentas voltaram sua atenção para o monitoramento e controle do cronograma do projeto e para o planejamento do escopo. Fehlmann e Kranich [EP18], por exemplo, por meio de um Gráfico de *Burn-Up* de Abordagem Bayesiana, conseguiram fornecer estimativas de quanto tempo adicional é necessário para concluir o trabalho planejado.

## ❖ Técnica

Seguindo o que foi verificado a respeito das ferramentas, a partir dos estudos selecionados, também foram identificadas cinco técnicas que visam apoiar a gestão de projetos de software, conforme apresentado no Quadro 14.

Quadro 14 - Artigos que apresentam abordagem do tipo “Técnica”

<b>Estudos que apresentam abordagem “Técnica”</b>
[EP33], [EP35], [EP41], [EP51], [EP60]

Fonte: O autor (2022)

Stapel *et al.* [PS60] apresentam o *Flow Mapping*, uma técnica do Método FLOW para planejar e orientar a comunicação em projetos de desenvolvimento distribuído. Para atingir esses objetivos, a técnica está centrada na visualização de um mapa FLOW. Um

mapa FLOW é um modelo FLOW especial (ou seja, visualização de participantes do projeto, documentos e fluxos de informações) estendido por recursos para melhorar a conscientização em equipes distribuídas. Segundo os autores, ao utilizar a abordagem *Flow Mapping* a comunicação de um projeto distribuído pode ser planejada em um dia útil.

Kroll *et al.* [EP35] usaram uma técnica de atribuição baseada em algoritmo genético. A técnica foi usada para atribuir tarefas em um projeto GSD. A técnica usa um simulador GSD baseado em fila para avaliar a função de aptidão. Os resultados baseados em um estudo de caso múltiplo (aplicando a técnica a dados de três projetos do mundo real) mostram que a abordagem pode ser tão boa ou melhor do que as atribuições de tarefas dos gerentes de projeto.

#### ❖ *Framework*

Quatro estudos apresentam algum *framework* que apoiam o gerenciamento de projetos de software no contexto ágil, como pode ser visto no Quadro 15.

Quadro 15 - Artigos que apresentam abordagem do tipo “Framework”

<b>Estudos que apresentam abordagem “Framework”</b>
[EP19], [EP24], [EP32], [EP45]

Fonte: O autor (2022)

Silva e Oliveira [PS19] desenvolveram um *framework* ágil de gerenciamento de portfólio de projetos. O *framework* refere-se a uma abordagem flexível de gestão de portfólio, sugerindo reuniões mais rápidas e dinâmicas, com foco principalmente na interação e comprometimento dos envolvidos no processo. Algumas práticas ágeis que podem ser utilizadas em cada atividade deste *framework* são: planejar a gestão do portfólio, identificar novas propostas, analisar os projetos candidatos, compor o portfólio de projetos e monitorar o portfólio.

Guerreiro *et al.* [EP24] apresentam o Eagle, um *framework* que suporta uma forma sistemática de definir, medir e visualizar as práticas dos membros das equipes de desenvolvimento de software seguindo princípios ágeis. Especificamente, a estrutura fornece arquitetura de micro-serviços com base no ecossistema “Governify” para gerenciar adequadamente. A estrutura fornece um ecossistema de ferramentas para que

as organizações definam suas melhores práticas para acompanhar e acompanhar a adesão de suas equipes e membros, a fim de conhecer suas armadilhas e melhorar o projeto ao longo do tempo.

Jain e Suman [EP32] expõem em seu estudo sobre um *framework* de gerenciamento de projetos para o GSD. O *GSD Project Management Framework*, como os autores chamam a abordagem, assimila as áreas de conhecimento do PMBOK com as áreas de conhecimento necessárias para uma gestão eficaz do GSD. Ele orientaria o gerente de projeto sobre os aspectos a serem considerados ao executar projetos distribuídos. A estrutura apresentada abrange gerenciamento de viabilidade e risco, gerenciamento de equipe virtual, gerenciamento de conhecimento, gerenciamento de escopo e recursos, gerenciamento de desempenho e gerenciamento de integração GSD.

### ❖ Prática

Conforme mostra a Figura 10, as práticas foram os tipos de abordagens menos identificados nos estudos selecionados, com um total de dois estudos que abordam apenas algumas delas, conforme apresentado no Quadro 16.

Quadro 16 - Artigos que apresentam abordagem do tipo “Prática”

Estudos que apresentam abordagem “Prática”
[EP57], [EP65]

Fonte: O autor (2022)

Schreiber *et al.* [EP57] discutem uma prática chamada *Metrics Driven Research Collaboration* (MEDIATION). A prática visa garantir que todos os participantes do projeto tenham um objetivo comum contínuo: o sucesso do projeto. De acordo com a prática estabelecida, a equipe do projeto deve se concentrar nos requisitos mais importantes e verificar continuamente se o produto de software está em conformidade com o escopo definido e as métricas correspondentes. A prática estabelece ainda que o status, os desafios e o progresso do projeto sejam sempre transparentes para todos os membros da equipe.

Zhang *et al.* [EP65] implementaram uma prática chamada *Fireteam*, uma prática que se concentra em pequenas equipes na indústria de software. *Fireteam* nada mais é do que um estilo de trabalho em equipe. A prática define de dois a cinco membros para lidar

com a divisão de trabalho e questões de coordenação em equipes de desenvolvimento tradicionais. Resumidamente, a prática visa reduzir as despesas gerais de gerenciamento de projetos e melhorar a produtividade, por meio da institucionalização da prática de pequenas equipes em toda a organização, para solucionar problemas decorrentes de aspectos humanos e sociais, como amizade, talento, habilidade e comunicação.

### 3.2.8 Sub-Q7: Quais os pontos fortes e fracos das abordagens?

#### ❖ Pontos Fortes

Quando um pesquisador/desenvolvedor propõe-se a desenvolver uma determinada abordagem, ele busca alguns meios que o ajudem a romper barreiras específicas do contexto em que está inserido. Com relação às abordagens que visam apoiar a gestão de projetos de software, identificadas a partir dos estudos selecionados, não é diferente. Todas elas têm algumas vantagens e, embora sejam, em alguns casos, diferentes entre si, no final das contas, todos têm o mesmo objetivo: apoiar de forma eficiente as equipes de desenvolvimento de software.

Algumas abordagens, apesar de virtuais tentam aproximar-se o máximo possível do mundo real [EP38], como é o caso da ferramenta desenvolvida por Liskin e Schneider [EP38]. Da mesma forma, as abordagens identificadas sempre tentam reduzir o esforço exigido pela equipe para concluir uma tarefa do projeto [EP34, EP56, EP58, EP59]. Essas abordagens são consideradas robustas o suficiente para identificar problemas no projeto e corrigi-los a tempo, evitando maiores danos aos recursos, cronograma e, conseqüentemente, ao projeto [EP48].

Destacam-se também as abordagens que proporcionam grande facilidade no planejamento das comunicações, como é o caso do *Flow Mapping* apresentado por Stapel *et al.* [PS60], que permite o planejamento da comunicação em até um dia. Além disso, existe abordagem que permite a redução do tempo de Sprint, ao trabalhar com Scrum, sem que o projeto perca qualidade, mesmo que a redução seja mínima, conforme relatado por Alhazmi e Huang [EP01] e Kroll *et al.* [EP35].

#### ❖ Pontos Fracos

Um dos pontos a se levar em consideração quando falamos de qualquer tipo de abordagem, seja qual for a finalidade para a qual foi desenvolvida, são as suas limitações. As limitações são inerentes a qualquer trabalho, especialmente quando se trata de abordagens focadas em equipe. Em relação às abordagens de gerenciamento de projetos de software, há uma série de limitações identificadas em alguns estudos selecionados e que acabam estendendo-se aos demais, em algumas situações.

Inicialmente, foi possível observar que existem abordagens que focam, dentre outras coisas, no reparo constante do projeto e dos produtos de trabalho gerados por ele. No entanto, o ato de corrigir defeitos constantemente pode trazer esforços adicionais para os desenvolvedores, conforme relatado por Tang [EP34]. Existem também abordagens que dependem inteiramente de um compromisso sólido e igual de todos os membros da equipe. Se a equipe não estiver 100% focada e comprometida, o projeto naturalmente tende a falhar [EP12, EP28, EP47, EP54], pois funcionários menos engajados podem realizar atividades de forma inadequada, mesmo com o auxílio de alguma abordagem, conforme relatam Godoy *et al.* [EP22].

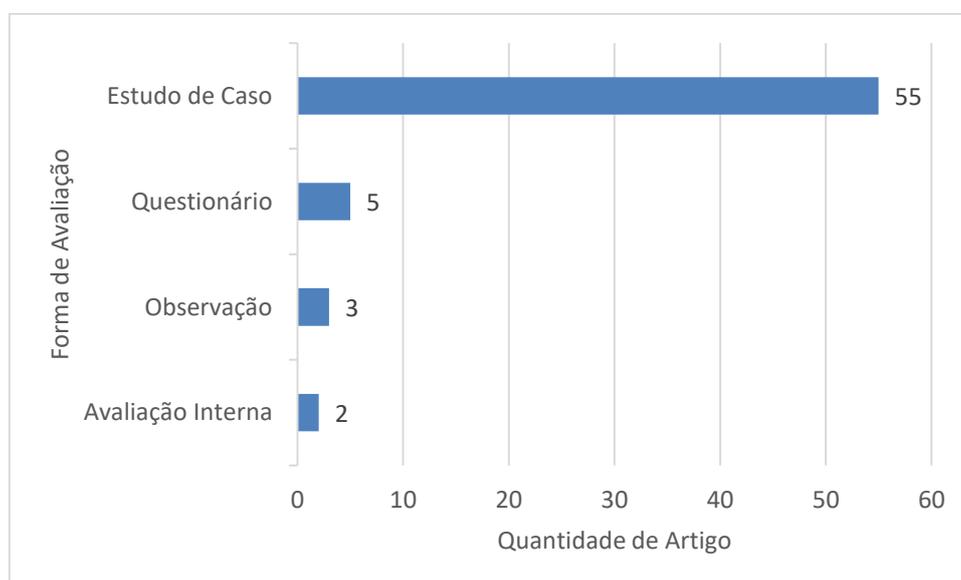
Algumas abordagens não são tão recomendadas para planejar um pequeno número de tarefas, pois foram desenvolvidas para lidar com grandes volumes de dados, como o SPESS, desenvolvido por Alhazmi e Huang [EP01]. Outras abordagens não monitoram constantemente o projeto e isso limita a visão da equipe sobre o andamento do que está sendo desenvolvido, conforme Stapel *et al.* [EP60].

No que se refere especificamente aos softwares, muitos *plugins* que são desenvolvidos para compô-los acabam caindo em desuso rapidamente, como é o caso dos *plugins* desenvolvidos para o Redmine, relatados no estudo de Dowling e McGrath [EP16]. Para os autores, embora exista uma comunidade ativa de desenvolvedores trabalhando no Redmine e criando *plugins* valiosos, isso é uma “faca de dois gumes”, visto que nem sempre os *plugins* são compatíveis com as versões mais recentes, o que acarreta em problemas durante a atualização e falhas na funcionalidade de recursos que as equipes usavam anteriormente.

### **3.2.9 Sub-Q8:** Como foi feita a avaliação das abordagens?

Como pode ser observado na Figura 11, a forma mais utilizada pelos autores dos estudos para avaliar suas abordagens foi o estudo de caso, com um total de 55 estudos que utilizaram esse método. Trapa e Rao [EP61], por exemplo, aplicaram sua abordagem a um projeto chamado Cronos, onde histórias de usuários foram divididas em tarefas e inseridas no software. Cada tarefa foi atribuída a dois desenvolvedores que foram responsáveis por atribuir a estimativa de tempo para cada tarefa pela qual eram responsáveis. Em seguida, foram gerados relatórios com informações sobre o tempo de conclusão do projeto e os impactos das adições de novas funcionalidades que passaram a existir.

Figura 11 – Forma de avaliação das abordagens



Fonte: O autor (2022)

Alguns estudos utilizaram questionários para obter *feedback* sobre a abordagem desenvolvida, cinco estudos mais precisamente, conforme ilustrado na Figura 11. Bastarrica *et al.* [EP05] formularam duas questões de pesquisa para identificar as vantagens e desvantagens de usar sua abordagem. A coleta de dados envolveu um questionário estruturado que foi disponibilizado ao presidente e a um profissional técnico de cada empresa. O objetivo do questionário era capturar as práticas reais e desejadas em relação ao gerenciamento de projetos.

Em relação às avaliações observacionais, podemos destacar o estudo de Mateescu *et al.* [EP42] que, a partir da observação do uso da abordagem proposta, conseguiram identificar pontos positivos e pontos de melhoria, bem como conclusões sobre o quão eficiente a abordagem proposta pode ser em um projeto. Os autores

verificaram que o aWall aproveita a resolução da tecnologia web e, assim, supera as possibilidades das ferramentas desktop existentes. Eles observaram que cada reunião ágil tem sua tarefa principal e objetivo específico, mas também precisa de muitas informações de suporte e produtos de trabalho.

A Figura 11 também mostra que dois estudos realizaram avaliações internas, ou seja, os próprios desenvolvedores das abordagens as avaliaram com base em suas próprias métricas, sem necessariamente disponibilizar as abordagens para terceiros. Bruegge *et al.* [EP11] avaliaram a viabilidade e o desempenho de sua abordagem a partir de dois experimentos de classificação. O primeiro experimento classificou as tarefas de acordo com a atividade a que pertenciam. No segundo experimento a abordagem foi empregada para classificar o status das tarefas, ou seja, o mecanismo de aprendizado de máquina prevê se essas tarefas já foram concluídas ou não.

## **Análise dos Resultados Obtidos**

Esta seção apresenta nossas principais conclusões e impressões sobre os resultados do MSL.

O primeiro item a ser observado refere-se aos resultados sobre quais áreas e fases do gerenciamento de projetos as abordagens enfatizam. As áreas de cronograma, qualidade e comunicação são as áreas que mais receberam atenção dos autores. Algo semelhante pode ser observado em Carvalho, Malcher e Santos (2020), onde os autores identificaram, por meio de uma pesquisa de opinião com gerentes de projetos de software, que as áreas de cronograma e comunicação são as duas áreas que mais preocupam esses profissionais.

Especificamente sobre o cronograma, área mais enfatizada nos estudos, foi possível observar que o foco nessa área dá-se pelo fato de que estimar o tempo e o esforço do projeto são tarefas cruciais e influenciam diretamente nos resultados do projeto [PS64]. Em relação às fases, observou-se que os autores dão maior ênfase ao planejamento do projeto, pois as tarefas precisam ser alocadas aos membros da equipe, levando em consideração a precedência das tarefas, equilibrando a carga de trabalho dos membros da equipe e garantindo a qualidade (LIN *et al.*, 2014).

Em relação ao tipo de abordagem, foi possível verificar que o software é o tipo de abordagem mais utilizado para apoiar a gestão de projetos. Resultados como esse

podem ser considerados comuns, considerando que atualmente diversos sistemas que automatizam processos estão tornando-se cada vez mais comuns. Tais sistemas facilitam o trabalho de pessoas em diferentes campos de aplicação (USPENSKIY *et al.*, 2019). Shaikh *et al.* (2018) também afirmam que softwares de gerenciamento de projetos são amplamente utilizados porque incluem diversos recursos úteis que facilitam o trabalho da equipe, principalmente do gerente de projetos. Dentre os recursos que os autores citam, destacam-se o gerenciamento de tarefas, monitoramento em tempo real, *chatbox*, notificações e alertas.

Além dos softwares, as metodologias de gestão de projetos também se destacaram. Durante a análise dos estudos selecionados foi possível identificar 12 estudos que tratam desse tipo de abordagem. Dentre essas metodologias, uma que se destaca é o Scrum. Essa metodologia foi utilizada diretamente em quatro estudos [EP12, EP28, EP47, EP54] e indiretamente, por meio de adaptações, em dois estudos [EP04, EP07]. Uma das razões que torna o Scrum uma metodologia altamente difundida diz respeito ao foco que a abordagem tem nas equipes, pois as equipes ágeis são os pilares de melhoria e bom desempenho de um projeto. Como qualquer método de desenvolvimento ágil, o Scrum segue uma abordagem colaborativa e guiada para o desenvolvimento de software, refletindo os princípios do Manifesto Ágil (MICHAEL *et al.*, 2021).

Algo que se destacou durante a análise dos estudos foi como as abordagens focam no comprometimento e engajamento da equipe. Entretanto, algumas abordagens impõem um processo de comunicação e interação constante, utilizando esses itens como forma de avaliar o desempenho dos membros da equipe. No entanto, é preciso ter cuidado com essa questão específica, pois existem abordagens que podem dar a falsa impressão de que um colaborador não está ajudando ou não está engajado no projeto. Isso ocorre porque esse tipo de abordagem baseia-se e tira suas conclusões de interações constantes. No entanto, nem sempre alguém que ajuda no projeto interage com tanta frequência, mas pode contribuir com mais informações técnicas [EP63].

Ainda sobre as limitações, verificou-se que algumas abordagens não são recomendadas para planejar um pequeno número de tarefas, pois foram desenvolvidas para lidar com grandes volumes de dados. Nesse caso, o ideal é que a organização invista em ferramentas que a monitorem constantemente, principalmente quando se trata de um projeto de grande porte [EP01].

Com relação às avaliações das abordagens realizadas pelos autores, foi possível perceber um índice considerável de estudos que utilizaram o estudo de caso como forma de avaliação, índice que corresponde a 86% dos estudos analisados. Esse índice permite verificar que esse tipo de avaliação consegue gerar resultados mais satisfatórios e confiáveis, pois a abordagem é submetida a um ambiente real e a partir de sua utilização são extraídos dados que permitem o planejamento e condução de melhorias, além da redução de possíveis limitações que a abordagem possa apresentar.

Com base nos resultados elencados e descritos neste MSL é possível afirmar que, embora tenha ocorrido uma queda no número de estudos nos anos de 2020 e 2021 (este último ano justificado por ser o ano de realização do MSL), há interesse por parte da academia e da indústria em desenvolver pesquisas voltadas à gestão de projetos de software, especialmente pesquisas que focam em abordagens que visem facilitar esse processo.

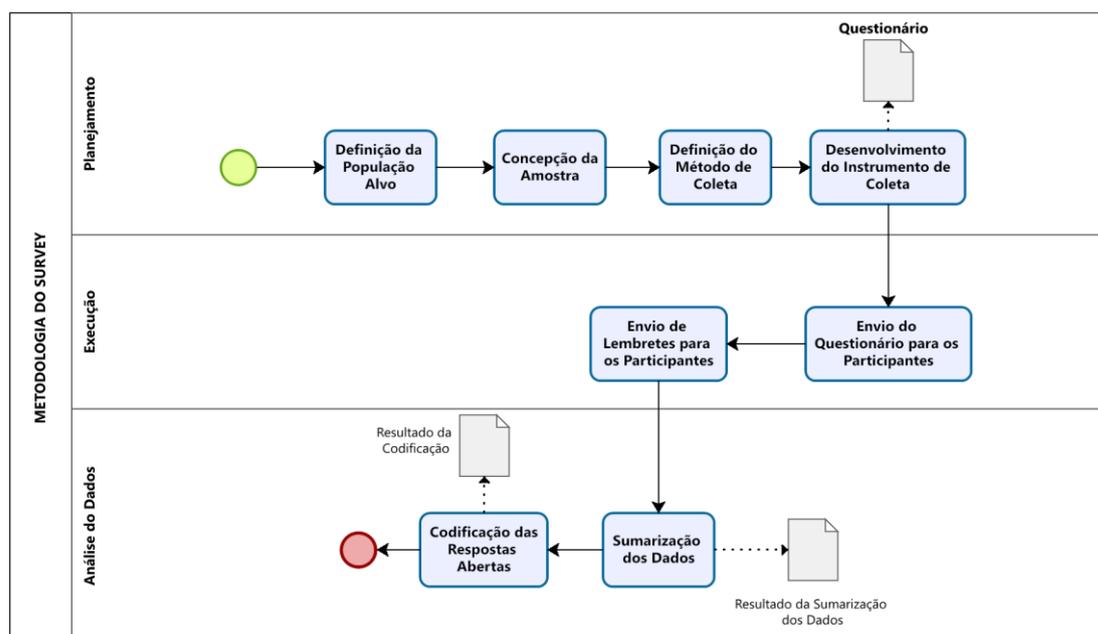
## 4 PERCEPÇÃO SOBRE O USO DE ABORDAGENS DE GERENCIAMENTO DE PROJETOS DE SOFTWARE NO CONTEXTO ÁGIL: UMA APLICAÇÃO DE SURVEY

Este capítulo apresenta os resultados do *survey*, sendo apresentado inicialmente os passos que foram utilizados para desenvolvê-lo. Posteriormente, são apresentados os resultados obtidos mediante a resposta dos participantes.

### 4.1 Metodologia do Survey

O ser humano, em sua busca incessante pelo conhecimento, desenvolve e aprimora técnicas que o permite explorar a imensidão desconhecida que lhe rodeia (GIL, 2008). Um método que possibilita a realização de estudos no campo científico é a pesquisa de opinião ou *survey*. Esse método tomou força enquanto método de pesquisa a partir de sua utilização por Karl Marx e Max Weber. Marx realizou um levantamento com 25.000 trabalhadores franceses com o intuito de verificar o grau de exploração que esses trabalhadores sofriam por seus patrões. Da mesma forma, Weber fez uso da pesquisa de opinião para realizar um estudo sobre a ética protestante (BABBIE, 1999). GIL (2008) diz um *survey* é caracterizado pela interrogação direta das pessoas as quais deseja-se conhecer algum comportamento, gosto ou opinião. A Figura 12 apresenta as etapas que foram seguidas durante a condução deste *survey*.

Figura 12 – Metodologia utilizada na condução do *survey*



Fonte: O autor (2022)

Um *survey* pode ser conduzido tanto de forma presencial, como de forma remota, sendo essa última maneira denominada de *survey* online. O *survey* online é um método que tem como características o conhecimento direto da realidade dos respondentes, a economia de recursos e a rapidez na obtenção dos dados (PRODANOV; FREITAS, 2013). Assim, este *survey* teve como objetivo coletar a opinião dos gerentes de projetos sobre as abordagens identificadas no MSL. Para conseguir atingir esse objetivo, o *survey* foi conduzido seguindo três etapas, a saber: (i) Planejamento, (ii) Execução e (iii) Análise dos Dados. A seguir, cada uma das etapas será detalhada.

#### 4.1.1 Planejamento

O planejamento do *survey* passou pela revisão da literatura, a fim de obter conhecimento sobre o método que seria adotado. Além disso, na etapa de planejamento foram definidos a população alvo do estudo, a amostra, o método de coleta e desenvolvido o instrumento de coleta. A população alvo da pesquisa foi formada por gerentes de projetos de software, porém aplicar um *survey* a todos os gerentes de projetos é algo inviável. Diante disso, foi definida uma amostra a partir da técnica de cotas, que consiste na escolha de participantes a partir de um determinado critério (HOPPING *et al.*, 2018).

Para coletar as respostas dos participantes, um questionário online foi usado como instrumento de coleta, sendo que o questionário era composto por quatro seções: a primeira seção consistia na apresentação da pesquisa, bem como seus objetivos, como forma de estimular a pessoa a respondê-lo; a segunda seção apresentava questões de caracterização dos participantes; a terceira seção apresentava questões de caracterização das empresas nas quais os participantes atuavam; e, por fim, a quarta seção trazia as perguntas referentes às abordagens de gerenciamento de projetos.

#### 4.1.2 Execução

O questionário foi encaminhado para um total de 106 pessoas e ficou disponível durante o período de 17 de maio de 2022 a 19 de junho de 2022. Durante esse período de pouco mais de um mês, três lembretes foram enviados para os participantes, reforçando a importância das respostas para o sucesso da pesquisa. A seleção dos participantes deu-

se por meio do LinkedIn e eles foram selecionados após a constatação de que em seus perfis contava o cargo de gerente de projetos ou gerente de projetos de software. Para selecionar esses participantes, os seguintes filtros foram usados:

- **Localidade:** Brasil;
- **Setor:** Tecnologia, Informação e Mídia; Tecnologia, Informação e Internet; Desenvolvimento de Software;
- **Serviço:** Desenvolvimento de Software; Tecnologia da Informação; Gestão de Projetos.

Finalizado o prazo estabelecido para a coleta de dados, foram obtidas 30 respostas, o que corresponde a 28,30% dos 106 participantes. Esse percentual pode ser considerado um valor relevante, haja vista o baixo número de respostas obtidas por meio de *surveys* online (SMITH *et al.*, 2013).

#### 4.1.3 Análise dos Dados

A análise dos dados consistiu em uma análise quantitativa e qualitativa, isto é, para sumarizar e descrever os dados obtidos foi utilizada a estatística descritiva. Nessa etapa da pesquisa, os dados foram organizados, tratados e dispostos em gráficos e tabelas, como forma de condensar as informações obtidas. Enquanto isso, para fins de análise qualitativa, foram usados dois procedimentos: a codificação aberta e a codificação axial (STRAUSS; CORBIN, 2007).

Na codificação aberta os dados foram trabalhados minuciosamente. Para tal, foram realizadas comparações e categorizações dos dados, a fim de identificar códigos e categorias que, posteriormente, foram relacionados entre si. Na codificação axial, os códigos e categorias identificados na codificação aberta foram analisados para identificar possíveis relações. Esses relacionamentos são denominados conectores e, por sua vez, apontam causas e efeitos, condições e estratégias de ação (SCHOTS, 2010).

A análise qualitativa deu-se a partir dos dados obtidos das questões abertas, ou seja, aquelas questões as quais o participante citou os pontos fortes e fracos de cada abordagem, segundo suas experiências.

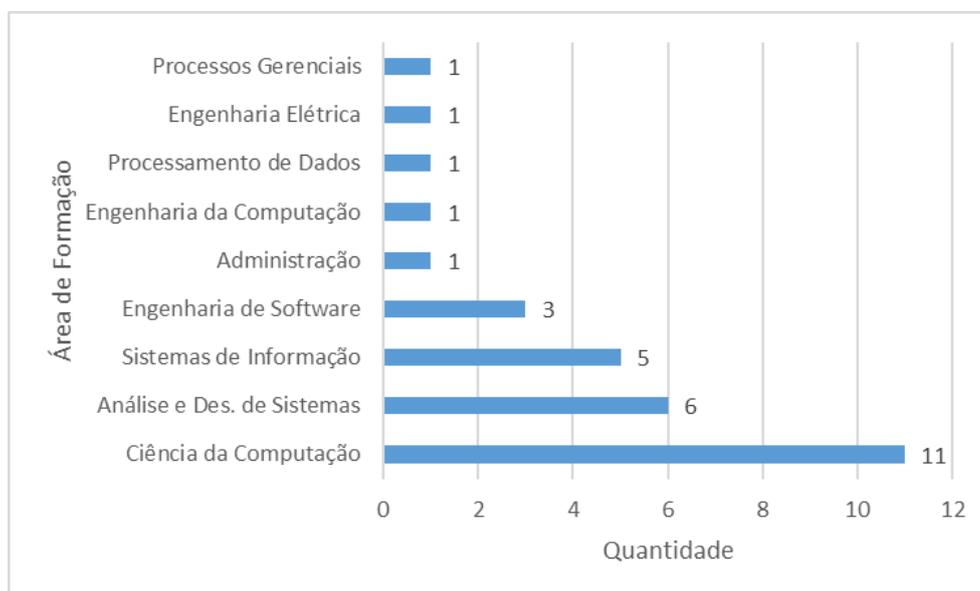
## 4.2 Resultados do *Survey*

Esta seção apresenta os resultados obtidos com a execução desta pesquisa. Além disso, para cada resultado mais significativo é apresentado uma discussão baseada na literatura.

### 4.2.1 Caracterização dos Participantes

A primeira seção de perguntas do questionário apresentava questões de caracterização dos participantes. A primeira questão questionava os participantes quanto a sua formação. Como pode ser observado na Figura 13, 11 participantes são graduados em Ciência da Computação, seguido de Análise e Desenvolvimento de Sistemas com 6 participantes formados na área, Sistemas de Informação com 5 participantes que possuem essa formação e, por fim, 3 participantes informaram ser formados em Engenharia de Software. Ainda de acordo com a Figura 13, Processos Gerenciais, Engenharia Elétrica, Processamento de Dados, Engenharia da Computação e Administração possuem um participante formado nas respectivas áreas.

Figura 13 - Área de formação dos participantes

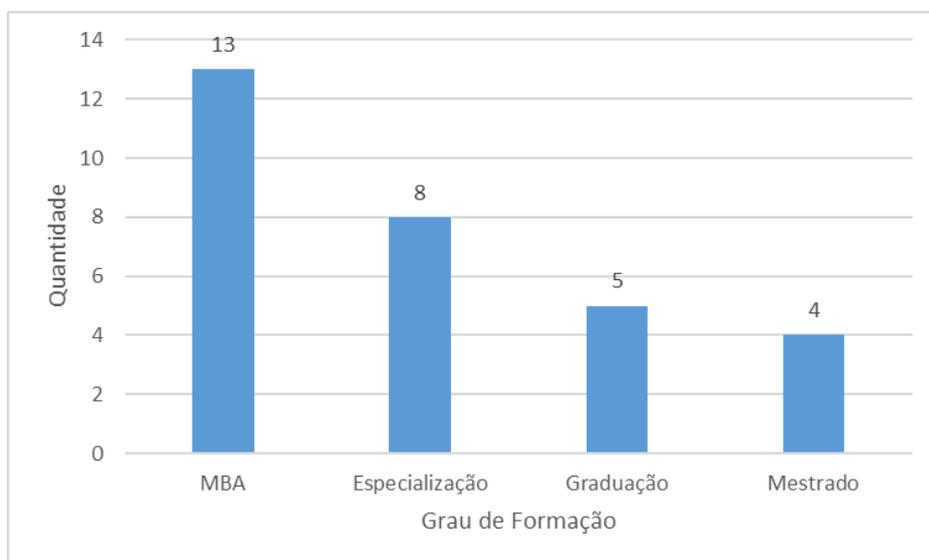


Fonte: O autor(2022)

Na questão 2 os participantes foram questionados quanto ao seu grau de formação. A Figura 14 mostra que a maioria possui o grau de *Master of Business Administration* (MBA), com um total de 13 participantes com esse grau de formação. Logo em seguida, 8 participantes informaram possuir o grau de especialização, seguido

de graduação e mestrado com 5 e 4 participantes com esse grau de formação, respectivamente.

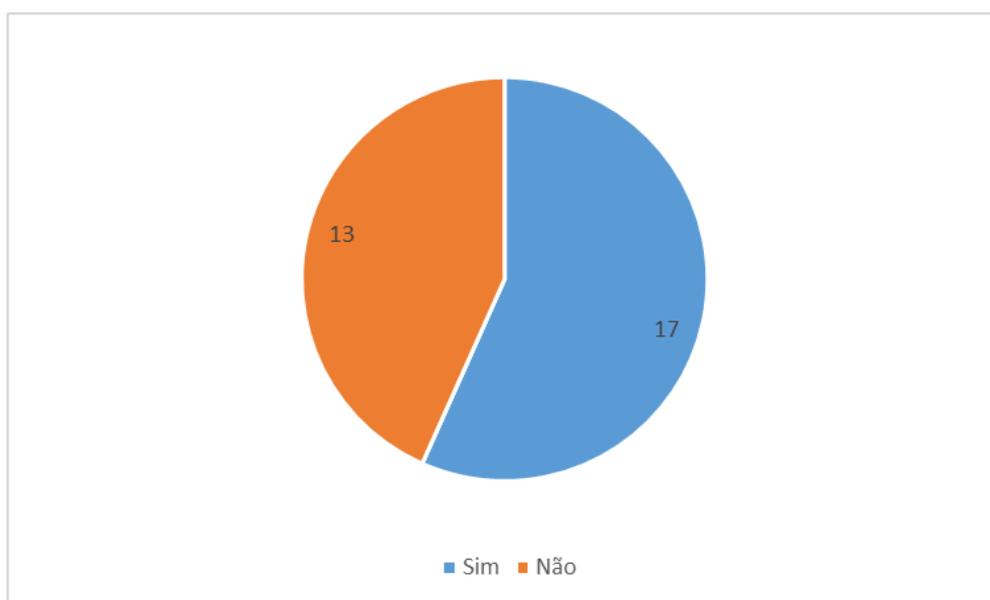
Figura 14 - Grau de formação dos participantes



Fonte: O autor (2022)

Foi perguntado aos participantes também se eles possuíam alguma certificação em gerência de projetos ou em áreas correspondentes. De acordo com a Figura 15, 17 participantes possuem pelo menos uma certificação em gerência de projetos. Enquanto isso, 13 participantes não possuem certificação alguma.

Figura 15 - Quantidade de participantes com e sem certificações em gerência de projetos

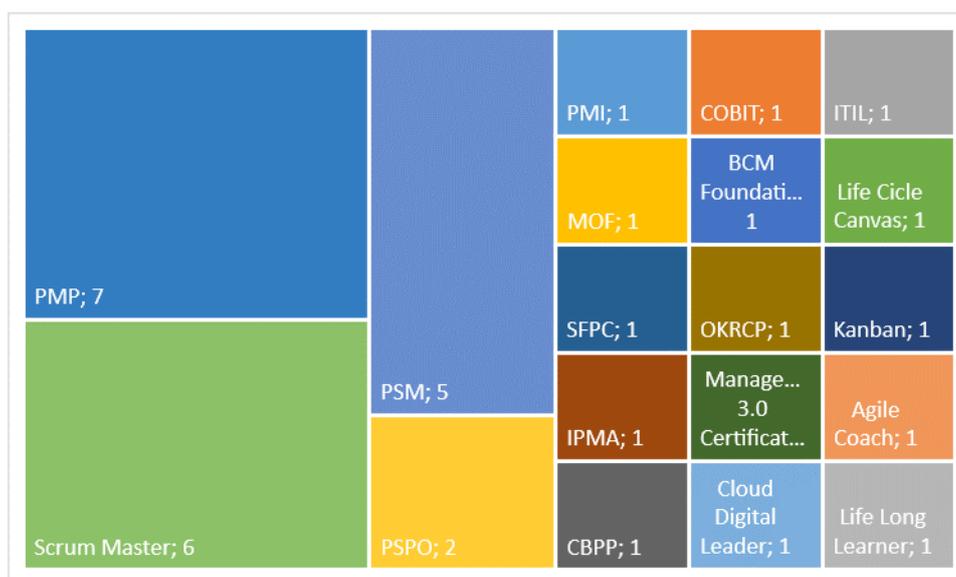


Fonte: O autor (2022)

Ainda sobre as certificações dos participantes, a Figura 16 apresenta quais são essas certificações, bem como a quantidade de participantes que as possuem. Nota-se a partir da Figura 16 que a certificação “*Project Management Professional - PMP*” é a certificação que a maioria dos participantes possuem, com um total de 7. Logo em seguida aparece a certificação de Scrum Master, sendo que 6 participantes informaram possuí-la. A certificação “*Professional Scrum Master - PSM*” aparece logo em seguida com um total de 5 participantes que a possuem.

É importante destacar que as certificações de Scrum Master e PSM possuem uma diferença, pois enquanto a primeira é uma certificação emitida por diversos órgãos diferentes, a PSM é emitida apenas pela Scrum.org. A certificação “*Professional Scrum Product Owner – PSPO*” foi citada ainda por dois participantes. Finalmente, certificações do PMI, *Control Objectives for Information and related Technology (COBIT)*, *Information Technology Infrastructure Library (ITIL)* e *Security Fundamentals Professional Certification (SFPC)* foram citadas apenas uma vez, como visto na Figura 16.

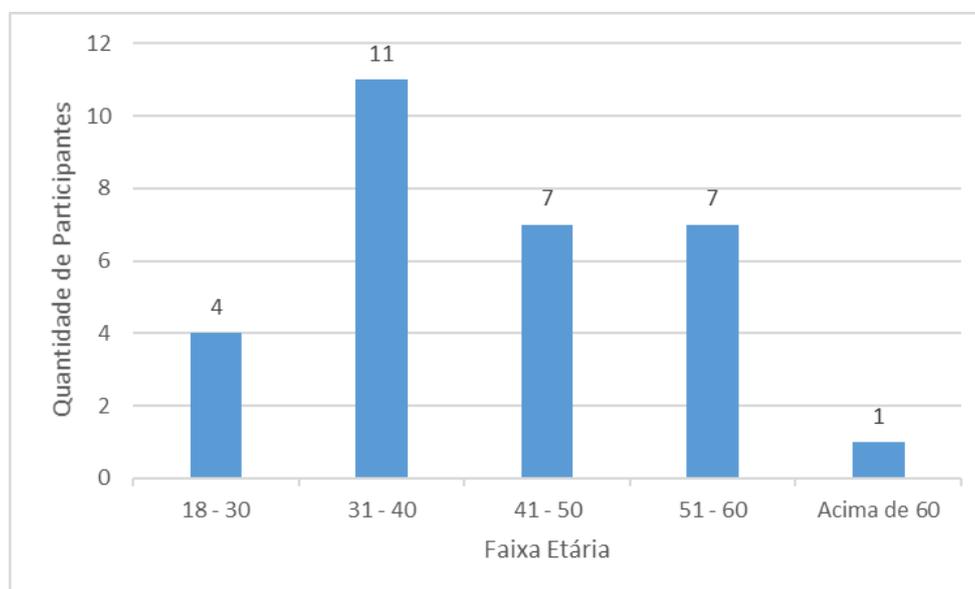
Figura 16 - Nome e quantidade de certificações dos participantes



Fonte: O autor (2022)

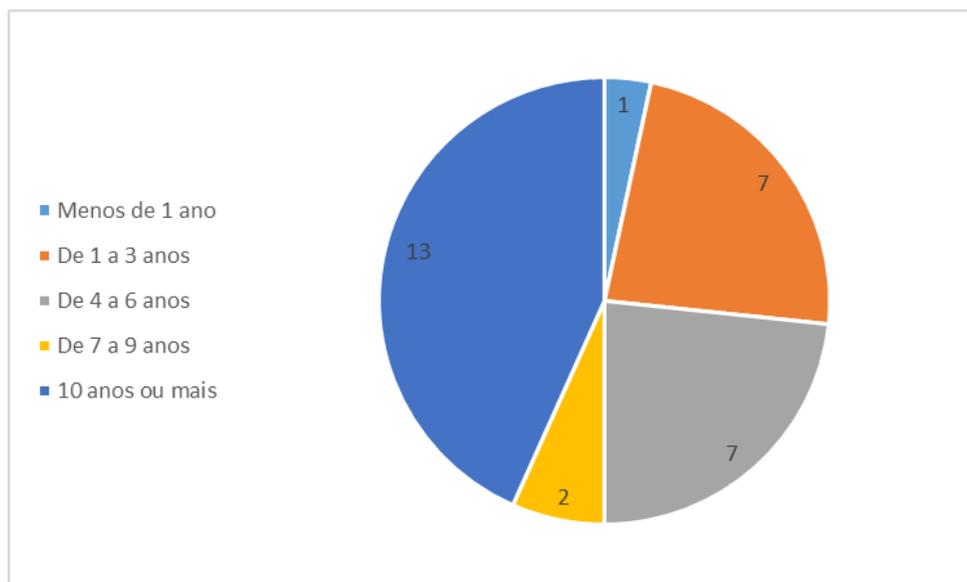
Outra informação coletada diz respeito à faixa etária dos participantes da pesquisa. A Figura 17 mostra que, dos 30 participantes, 11 estão dentro da faixa etária de 31 a 40 anos. 7 participantes possuem idades que variam de 41 a 50 anos. Outros 7 participantes têm idades entre 51 a 60 anos, enquanto 4 participantes possuem idades na faixa etária que corresponde de 18 a 30 anos. Por fim, apenas 1 participante informou possuir idade superior a 60 anos.

Figura 17 - Faixa etária dos participantes



Fonte: O autor (2022)

Figura 18 - Tempo de experiência dos participantes



Fonte: O autor (2022)

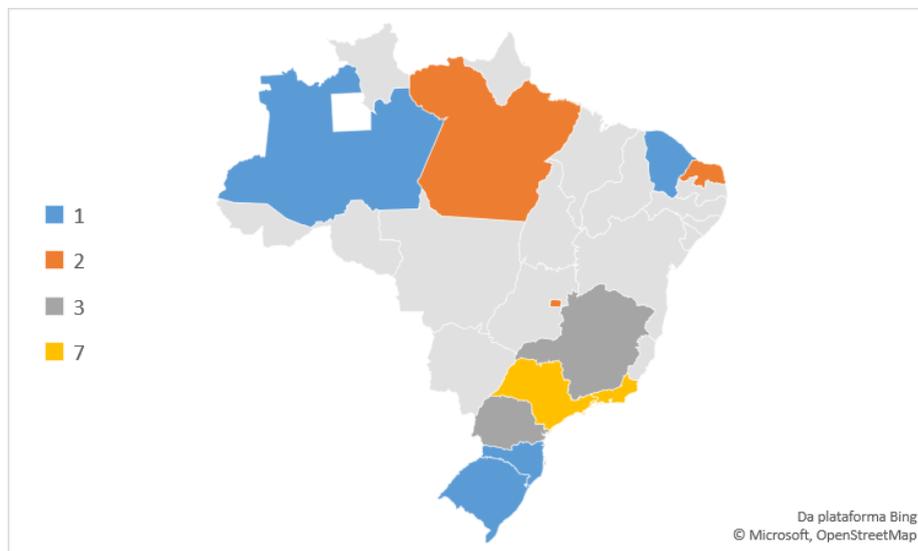
Fechando a seção 2 do questionário, foi perguntado aos participantes o tempo de experiência que eles tinham atuando como gerentes de projetos de software. Verifica-se na Figura 18 que, dentre os 30 participantes da pesquisa, 13 possuem 10 anos ou mais de experiência na área. 7 participantes informaram possuir experiência que variam entre quatro e seis anos. Mais 7 participantes possuem entre um e três anos de experiência, 2 possuem entre sete e nove anos de experiência na área e apenas 1 informou ter menos de um ano de experiência como gerente de projetos de software.

#### 4.2.2 Caracterização das Empresas

Com relação às empresa onde os participantes atuam, a Figura 19 mostra a distribuição dessas empresas por Estado da Federação. Como pode ser observado na figura, todas as regiões possuem pelo menos uma empresa representada, sendo que a maioria das empresas estão concentradas na região Sudeste do Brasil. São Paulo e Rio de Janeiro são os Estados com o maior quantitativo de empresas representadas, com um total de 7 empresas cada. Ainda na região Sudeste, o Estado de Minas Gerais possui 3 empresas onde os participantes atuam. Logo atrás do Sudeste, a região Sul possui 5 empresas onde os participantes informaram atuar, sendo o Paraná o Estado com o maior número de empresas, 3 no total. As regiões Norte e Nordeste aparecem empatadas nesse quesito, com 3 empresas em cada região, sendo que desse total, 2 empresas localizam-se no Pará

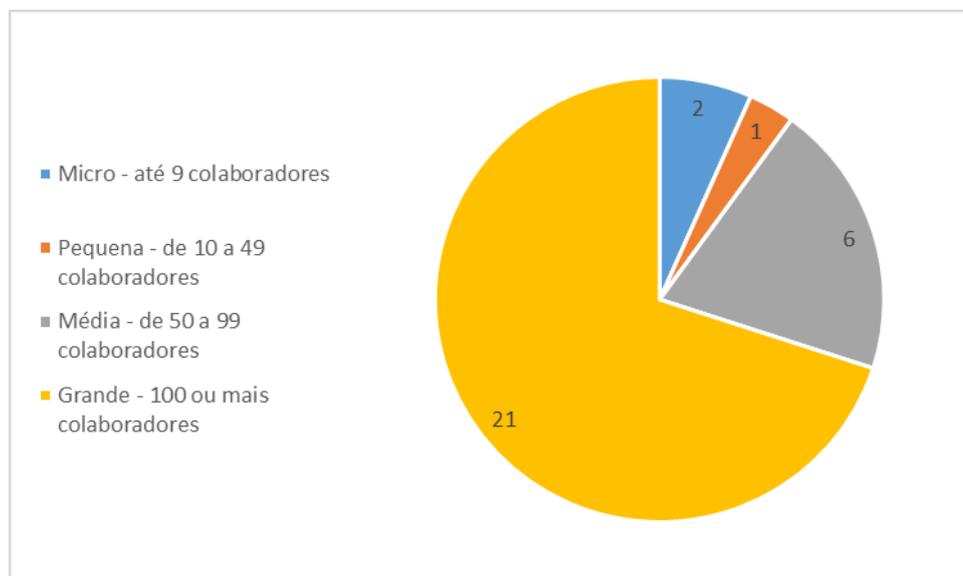
e no Rio Grande do Norte, e 1 empresa no Amazonas e no Ceará. Há ainda 2 empresas da região Centro-Oeste, ambas localizam-se no Distrito Federal.

Figura 19 - Estado onde se localiza a empresa onde participante atua



Fonte: O autor (2022)

Figura 20 - Tamanho das empresas



Fonte: O autor (2022)

Para encerrar a seção 3, foi questionado aos participantes o tamanho da empresa onde eles atuam. Salienta-se que o tamanho da empresa foi definido de acordo com o Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE).

Nota-se por meio da Figura 20 que 21 empresas são consideradas grandes empresas, ou seja, possuem 100 ou mais colaboradores. Logo depois, 6 empresas são consideradas médias, pois possuem entre 50 e 99 colaboradores. 2 empresas são consideradas microempresas, haja vista possuírem até 9 colaboradores. Por fim, apenas 1 empresa é considerada pequena, pois possui de 10 a 49 colaboradores.

#### 4.2.3 Abordagens de Gerenciamento de Projetos de Software

A última seção de perguntas apresentava questões referentes ao uso de abordagens de gerenciamento de projetos, sendo que cada pergunta foi separada de acordo com o tipo de abordagem identificada no MSL realizada. Os dados obtidos para cada tipo de abordagem estão descritos a seguir.

##### 4.2.3.1 Softwares

O primeiro tipo de abordagem usado no gerenciamento de projetos de software trata-se de softwares, sendo que, de acordo com a Figura 21, o Jira é o software mais utilizado pelos gerentes de projetos, com um total de 22 citações. Após o Jira, o Git foi o software mais citado pelos participantes da pesquisa. Vale destacar ainda o Redmine e o Slack, com um total de 10 e nove citações, nessa ordem.

O Jira é um software de gestão ágil de projetos e oferece uma série de recursos, como por exemplo: suporte ao Scrum, quadros Kanban, gerenciamento de tarefas e roteiros, além do gerenciamento de problemas (MONTGOMERY; LÜDERS; MAALEJ, 2022). Os benefícios que tornam o Jira um dos softwares mais usados no gerenciamento de projetos, de acordo com Montgomery, Lüders, Maalej (2022), é a inclusão de históricos de alterações de problemas, uma rede de rastreabilidade e vinculação de problemas, além da possibilidade de configurá-lo de acordo com a necessidade do cliente e do projeto. De acordo com Datanyze<sup>4</sup>, Slintel<sup>5</sup> e Enlyft<sup>6</sup>, o Jira é de longe o software

---

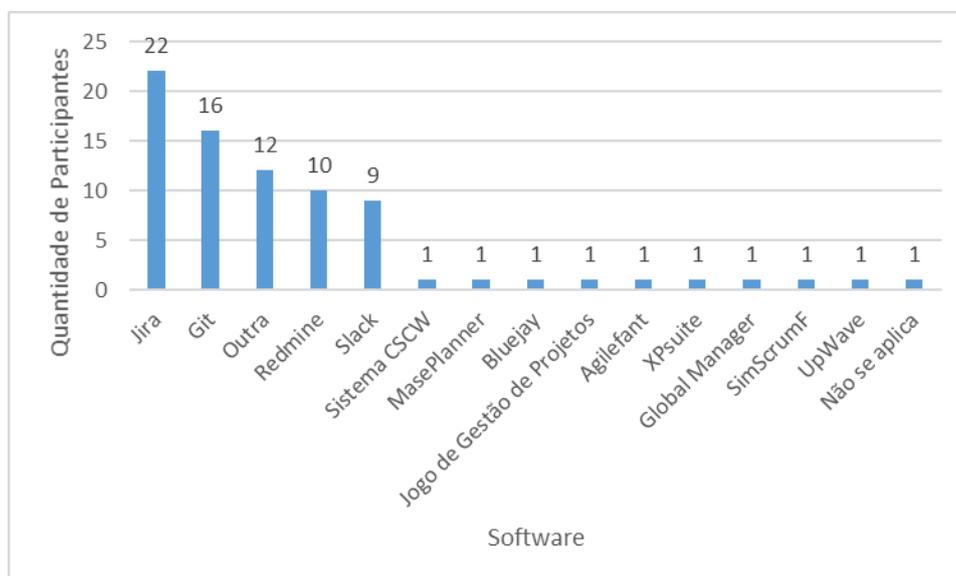
<sup>4</sup> <https://www.datanyze.com/market-share/project-management--217/jira-market-share>

<sup>5</sup> <https://www.slintel.com/tech/bug-and-issue-tracking/atlassian-jira-market-share>

<sup>6</sup> <https://enlyft.com/tech/products/atlassian-jira>

mais popular quando o assunto é gerenciamento de problemas e projetos, apesar de ainda ser pouco explorado em pesquisas de Engenharia de Software.

Figura 21 - Softwares de gestão de projetos utilizados pelos participantes



Fonte: O autor (2022)

Com relação ao Git, trata-se de um sistema de controle de versão distribuído popular que fornece flexibilidade e robustez para projetos de desenvolvimento de software (RÍOS; EMBURY; ERASLAN, 2022). O Git pode ser usado dentro do projeto como o gerenciador de artefatos, uma vez que controla as versões desses artefatos. No entanto, há autores que afirmam que o Git pode ir além, como é o caso Darussalam e Widyani (2021) que adaptaram o Git para gerenciar o cronograma do projeto. Os autores verificaram que o Git oferece uma série de operações que podem ser usadas no monitoramento do cronograma do projeto, mas que essa funcionalidade ainda é pouco explorada pelas equipes de desenvolvimento.

A Figura 21 mostra ainda alguns softwares poucos citados, mas que fazem parte do dia a dia de alguns participantes, como é o caso do XPsuite, UpWave, Agilefant e, ainda, alguns jogos de gestão de projetos. Vale destacar também o número considerável de participantes que usam outros softwares além daqueles listados no questionário, sendo esses softwares em sua maioria desenvolvidos ou adaptados pela própria equipe de desenvolvimento, de acordo com a necessidade do projeto e da empresa onde será aplicado.

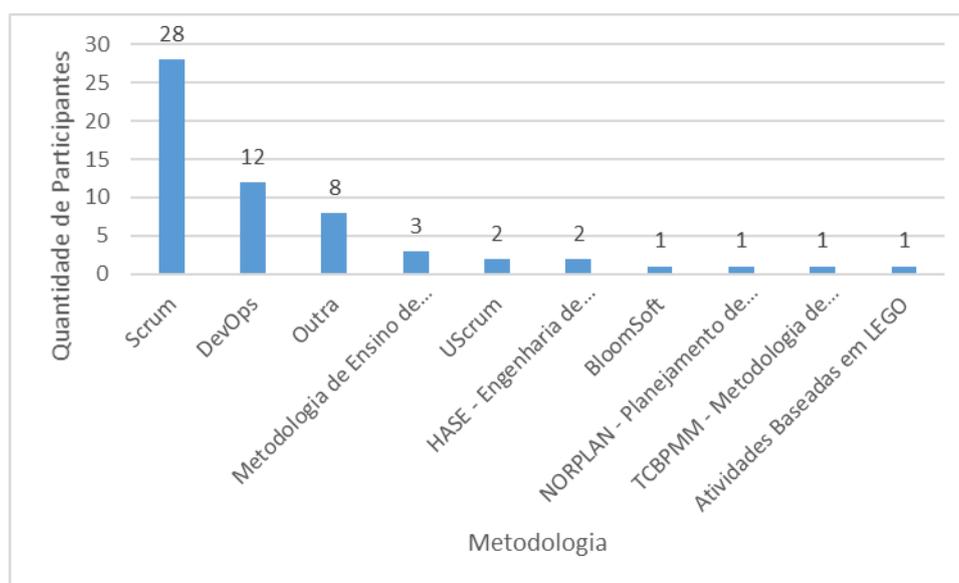
#### 4.2.3.2 Metodologias

As metodologias são outro tipo de abordagem usadas no gerenciamento de projetos de software, dentre as quais o Scrum é a mais usada, com um total de 28 participantes que relataram utilizá-la. Com 12 participantes informando que a usam, a metodologia DevOps ficou em segundo lugar entre as metodologias mais usadas pelos gerentes de projetos. Destacam-se também as metodologias de ensino de gerenciamento de projetos (3 citações), o UScrum e o Hase, com duas citações cada.

Com as exigências de um mercado cada vez mais conectado e uma alta volatilidade, é natural que as organizações busquem por agilidade em seus processos, visando uma entrega em menos tempo, mas mantendo a qualidade. Não é por acaso que o Scrum é uma das metodologias mais usadas quando estamos falando em gerenciamento de projetos ágeis. A Figura 22 mostra que os participantes da pesquisa seguem essa tendência.

Segundo Taufiq, Raharjo, Wahbi (2020), o Scrum é uma das abordagens mais populares em uso no desenvolvimento ágil de software. O Scrum possui três pilares principais que uma organização deve possuir, são transparência, inspeção e adaptação. Ainda de acordo com os autores, há uma tendência na utilização do Scrum, DevOps e Kanban combinados.

Figura 22 - Metodologias de gestão de projetos utilizadas pelos participantes



Fonte: O autor (2022)

Com relação ao DevOps, que foi a segunda metodologia mais citada pelos participantes, trata-se de uma metodologia que tem aumentado sua popularidade nos últimos anos e tem se tornado uma abordagem amplamente utilizada em empresas de desenvolvimento de software, como relatam Marrero e Astudillo (2021).

O DevOps pode ser definido como um processo que trabalha com práticas de agilidade dentro de uma cultura que preza pela colaboração como meio de melhoria do desenvolvimento e da entrega de softwares. Essa metodologia concentra-se nas equipes de desenvolvimento e operações, visando a troca de conhecimento entre seus membros para alcançar altos índices de qualidade no produto final (SMEDS; NYBOM; PORRES, 2015).

Metodologias como o BloomSoft, NORPLAN, TCBPMM e atividades baseadas em Lego também foram citadas uma vez cada pelos participantes. Além disso, assim como no caso dos softwares, há metodologias usadas pelos gerentes de projetos que foram desenvolvidas pela própria equipe que a utiliza. Com relação às metodologias, um total de oito participantes informaram que usam outro tipo de metodologia além das listadas no questionário.

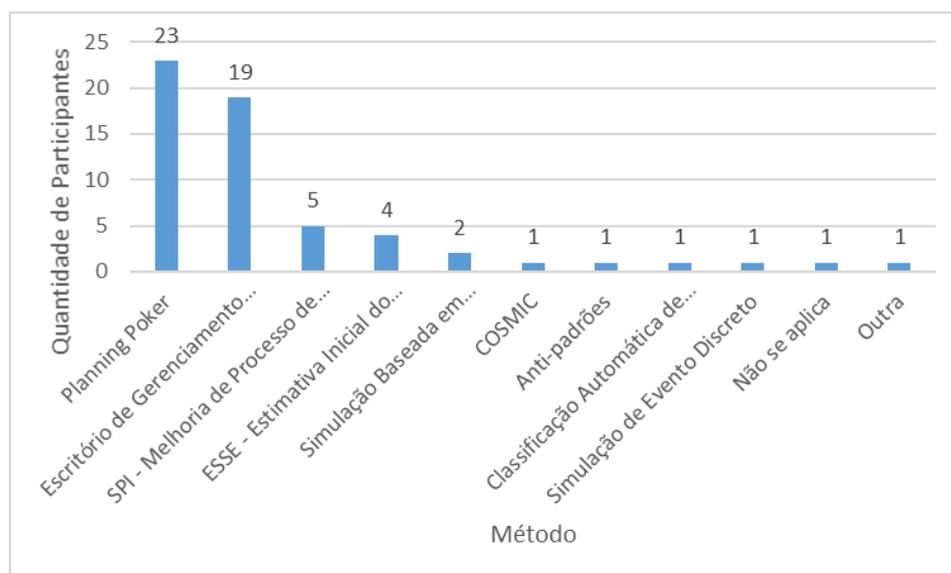
#### 4.2.3.3 Métodos

A Figura 23 apresenta os métodos de gestão de projetos mais usados pelos gerentes de projetos de software que participaram do *survey*. De acordo com a figura, o *Planning Poker* é o método mais utilizado pelos participantes, sendo mencionado por 23 participantes. Seguindo o *Planning Poker*, o Escritório de Gerenciamento de Projetos é o segundo método mais usado pelos participantes, com um total de 19 citações.

Dentre os métodos citados pelos participantes, destaca-se o *Planning Poker*, como observado na Figura 23. Esse método é considerado uma forma de incentivar discussões entre as equipes de desenvolvimento de software e é usada com frequência para estimar o esforço necessário para tarefas de um projeto e só encerrado quando todas as tarefas foram estimadas (LEINONEN, 2016). Para Sudarmaningtyas e Mohamed (2020), a maior vantagem do *Planning Poker* é o poder de decisão dado aos desenvolvedores, especialmente quando se trata de estimativas de esforços.

O *State of Agile Report*, publicado em 2021, afirma o *Planning Poker* é usado por 58% das organizações software para estimar esforços de projetos (VERSIONE, 2021). Complementando o que as pesquisas já constataram, a literatura também ratifica que o *Planning Poker* é um método que pode fornecer estimativas mais confiáveis, se comparado a outros métodos (MOLØKKEN-ØSTVOLD; HAUGEN; BENESTAD, 2008; MAHNIČ; HOVELJA, 2012; FERNANDÉZ-DIEGO *et al.*, 2020).

Figura 23 - Métodos de gestão de projetos utilizados pelos participantes



Fonte: O autor (2022)

No que diz respeito aos escritórios de gerenciamento de projetos, citado 19 vezes pelos participantes, os autores Rincon-Gonzalez, Diaz-Piraquive, Diez-Silva (2019) realizaram uma pesquisa de campo onde puderam constatar que, ao fazer uso desse método, as empresas tiveram uma melhora na entrega do escopo do projeto, maior cumprimento dos prazos do cronograma, maior número de projetos fechados dentro do orçamento, maior grau de cumprimento das especificações de qualidade e um uso otimizado dos recursos do projeto. Para os autores, empresas que incorporam os escritórios de gerenciamento de projetos tendem a se comportar melhor, comparada às empresas que não adotam esse método.

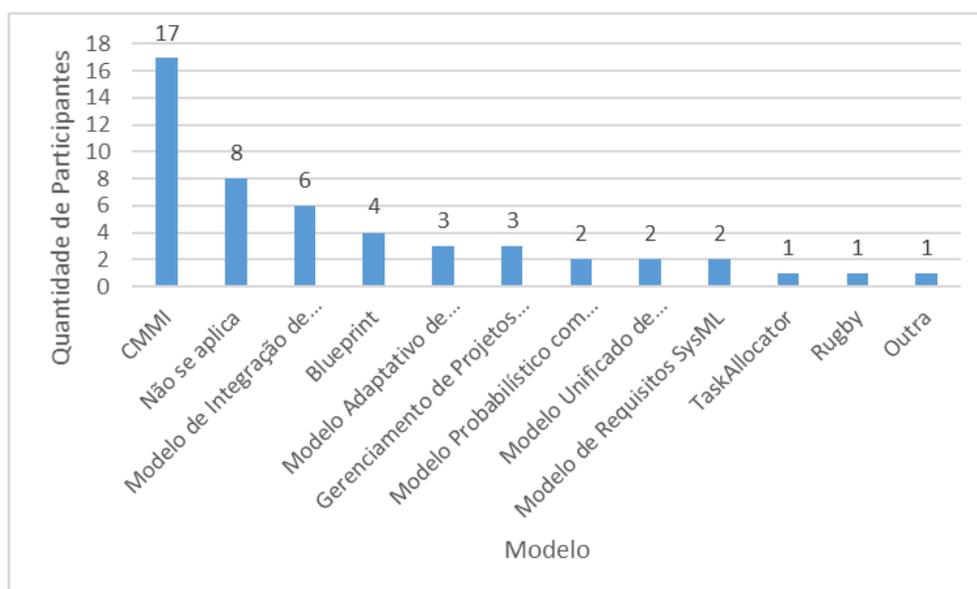
Há ainda métodos de Melhoria de Processos de Softwares, sendo esse método citado por cinco participantes; métodos de Estimativa Inicial de Software e métodos de Simulação Baseados em Agentes, sendo que esses dois foram citados por quatro e dois participantes, respectivamente. É importante mencionar também alguns métodos que

foram mencionados por apenas um participante cada, mas que de alguma forma contribuem para a melhoria da área, como é o caso dos métodos Antipadrões, dos métodos de Classificação Automática de Tarefas e métodos de Simulação de Evento Discreto.

#### 4.2.3.4 Modelos

A Figura 24 apresenta os modelos de gerenciamento de projetos de software citados pelos participantes do *survey*, dentre os quais o *Capability Maturity Model Integration* (CMMI) foi o que obteve o maior número de menções, com um total de 17. Logo em seguida aparecem os Modelos de Integração de Gerenciamento de Riscos, com seis citações. O *Blueprint* foi mencionado por quatro participantes e os Modelos Adaptativos de Diagnóstico de Saúde para o Desenvolvimento Ágil de Software, bem como o Gerenciamento de Projetos Baseado no Monitor de Projetos Empíricos foram mencionados três vezes cada um.

Figura 24 - Modelos de gestão de projetos utilizados pelos participantes



Fonte: Os autores (2022)

Influenciado pelas duras competições do mercado, muitas organizações de software precisam alcançar melhores resultados em seus processos gerenciais. Além disso, atualmente, pessoas, processos e tecnologias são os principais personagens do desenvolvimento de software. Sabendo disso, as organizações passaram a dar grande importância a eles. É nesse contexto que entra o CMMI, um modelo de melhoria de

processos de software bastante difundido, onde suas avaliações aumentam ano após ano. O CMMI é tido como um modelo que pode agregar ainda mais valor a uma organização, principalmente se levarmos em consideração seus ótimos resultados no aumento da produtividade (14%), redução do retrabalho (70%), aumento da lucratividade (57%), taxa de entrega no prazo (97%), redução do defeito (20%) e maior precisão da estimativa (17%) (SQME, 2019).

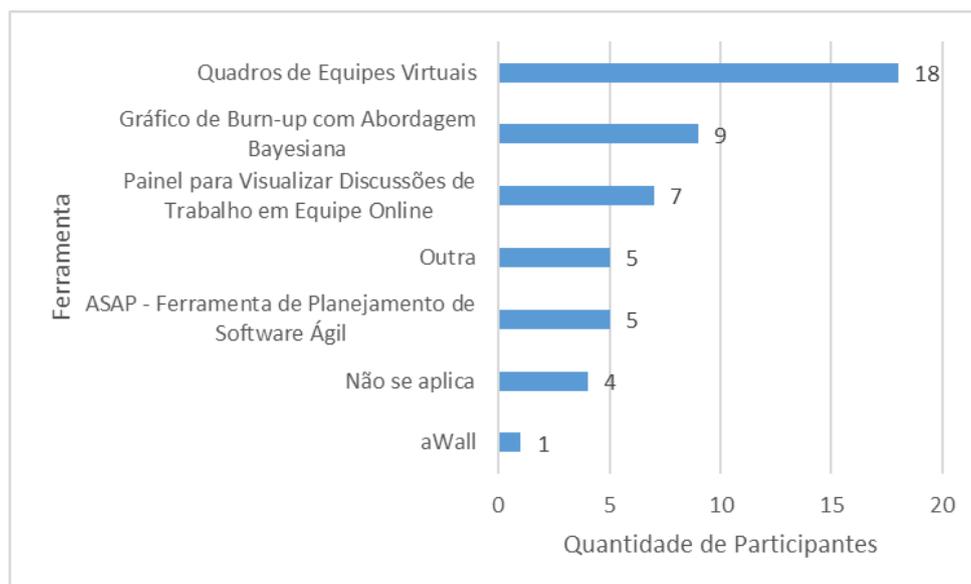
Outros modelos que estão destacados na Figura 24 tratam-se de Modelos Probabilísticos que usam Redes Bayesiana, Modelo Unificado de Rastreabilidade e Modelo de Requisitos SysML, cada um desses modelos obteve duas citações cada. Por fim, o *TaskAllocator*, Rugby e um modelo próprio da equipe foram citados uma única vez cada um deles.

#### 4.2.3.5 Ferramentas

No que tange às ferramentas de gerenciamento de projetos de software, o Quadro de Equipes Virtuais foi a mais citada pelos participantes, com um total de 18 citações, como mostra a Figura 25. Logo após vem o Gráfico de *Burn-up* com Abordagem Bayesiana, com nove citações. Há ainda sete participantes que disseram usar Painéis para Visualizar Discussões de Trabalho em Equipe Online.

Os projetos de software enfatizam a comunicação como um dos principais pilares de um projeto bem sucedido. A literatura aponta que a satisfação do cliente é o principal indicador de um projeto de qualidade e eles atribuem esse sucesso a uma comunicação bem feita e gerenciada. Nesse quesito, os Quadros de Equipes Virtuais entram como um facilitador da comunicação entre as equipes, principalmente em equipes distribuídas geograficamente.

Figura 25 - Ferramentas de gestão de projetos utilizadas pelos participantes



Fonte: Os autores (2022)

Segundo Liskin e Schneider (2012), usar abordagens visuais significa adicionar mais informações ao projeto. Para os autores, os Quadros de Equipes Virtuais são atraentes para as equipes, uma vez que permitem a visualização simples das histórias de usuários que precisam ser desenvolvidas. Azizyan, Magarian, Kajko-Matsson (2011) também descobriram em uma pesquisa que o atributo mais importante de uma ferramenta de gerenciamento de projetos é a facilidade de uso.

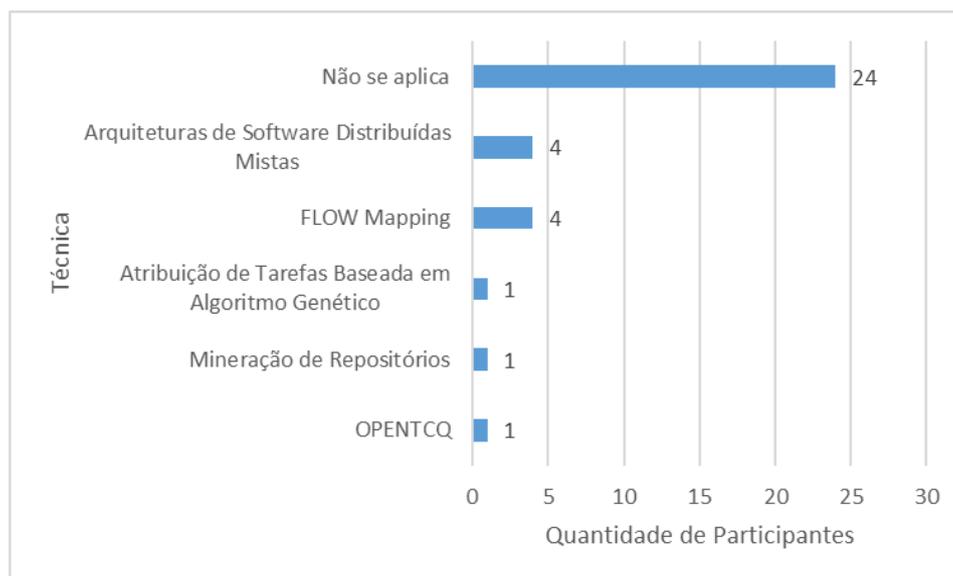
Ainda de acordo com a Figura 25, dentre os 30 participantes, cinco relataram que usam Ferramentas de Planejamento de Software Ágil e um usa a ferramenta aWall no gerenciamento de projetos. Ao observar ainda a Figura 25, é possível notar que quatro participantes disseram não usar nenhuma Ferramenta e cinco deles usam outras ferramentas desenvolvidas internamente pela equipe.

#### 4.2.3.6 Técnicas

Diferentemente das abordagens descritas nas seções anteriores, as técnicas de gerenciamento de projetos de software obtiveram poucas menções pelos participantes, sendo que 24 deles disseram não usar nenhuma delas, como pode ser observado na Figura 26. Por outro lado, técnicas de Arquiteturas Distribuídas e o *FLOW Mapping* foram as mais citadas, com quatro citações cada uma.

As empresas necessitam cada vez mais de uma visão ampla para tomar decisões críticas de negócios e, assim, entregar soluções orientadas aos seus clientes. Para que haja decisões mais assertivas, as empresas necessitam de dados dos seus clientes e esses dados estão distribuídos em várias fontes. Pensando nesse aspecto, as empresas começaram a adotar arquiteturas de softwares distribuídas.

Figura 26 - Técnicas de gestão de projetos utilizadas pelos participantes



Fonte: O autor (2022)

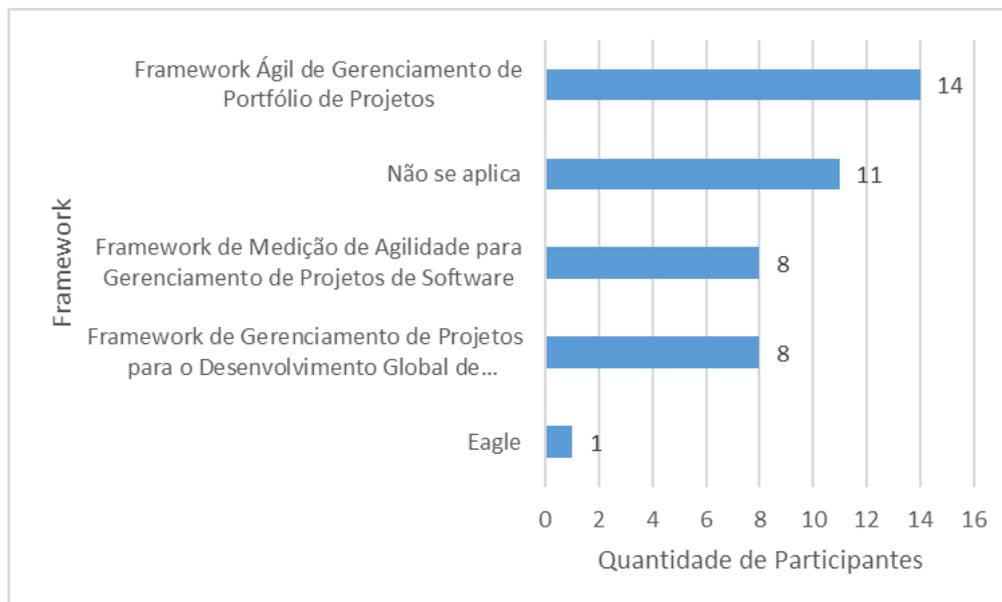
A arquitetura distribuída fornece informações apropriadas sobre o cliente para o qual o software está sendo desenvolvido. Esse tipo de arquitetura permite também que o software tenha uma melhor adequação às necessidades do cliente, pois essa técnica permite ir além dos requisitos e implementar funcionalidades que poderiam passar despercebida pelo cliente, como relatam Kargili *et al.* (2021). As técnicas de Atribuição de Tarefas Baseadas em Algoritmos Genéticos foram citadas uma vez, assim como também as técnicas de Mineração de Repositórios e o OpentCQ, como mostra a Figura 26.

#### 4.2.3.7 Frameworks

No que se refere aos *frameworks*, é possível verificar na Figura 27 que *Frameworks* Ágeis de Gerenciamento de Projetos são os mais usados pelos participantes,

sendo citados por um total de 14 participantes. A Figura 27 mostra ainda que oito participantes usam *Frameworks* de Medição de Agilidade.

Figura 27 - Frameworks de gestão de projetos utilizados pelos participantes



Fonte: O autor (2022)

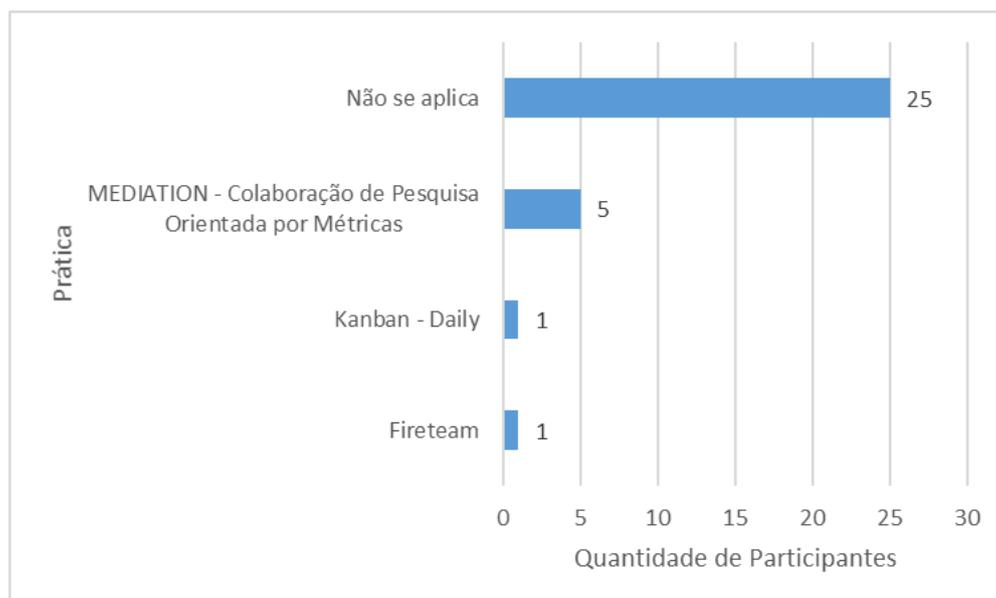
Sobre os *Frameworks* Ágeis de Gerenciamento de Projetos algumas vantagens podem ser destacadas, como por exemplo: a possibilidade de integração entre as pessoas; resposta rápida a imprevistos e mudanças; colaboração das partes interessadas. Além disso, ainda destacam-se por promover mais independência e produtividade para a equipe, podendo haver uma otimização de tempo e gestão de rentabilidade, já que não será necessário preocupar-se com contratos e documentações extensos e complicados, nem com ferramentas que engessam os processos de gerenciamento. Podemos destacar ainda aquele que pode ser considerado o principal benefício, um atendimento de qualidade às necessidades do cliente, cuja a atenção deve estar focada nas pessoas que adquirem seus produtos e/ou serviços (SILVA; OLIVEIRA, 2016).

Há ainda oito participantes que usam *Frameworks* de Gerenciamento de Projetos para o GSD. Nota-se também na Figura 27 que, um quantitativo considerável de participantes informou não usar nenhum *framework*, com um total de 11 participantes e apenas um participante usa o Eagle.

#### 4.2.3.8 Práticas

Com relação às práticas de gerenciamento de projetos de software, é possível observar, com base na Figura 28, que a maioria dos participantes não usam nenhuma prática, com um total de 25 participantes. Porém, dentre aqueles que usam, cinco usam a Colaboração de Pesquisa Orientada por Métricas, um usa o *Kanban Daily* e outro participante informou que usa a prática de *Fireteam*.

Figura 28 - Práticas de gestão de projetos utilizadas pelos participantes



Fonte: O autor (2022)

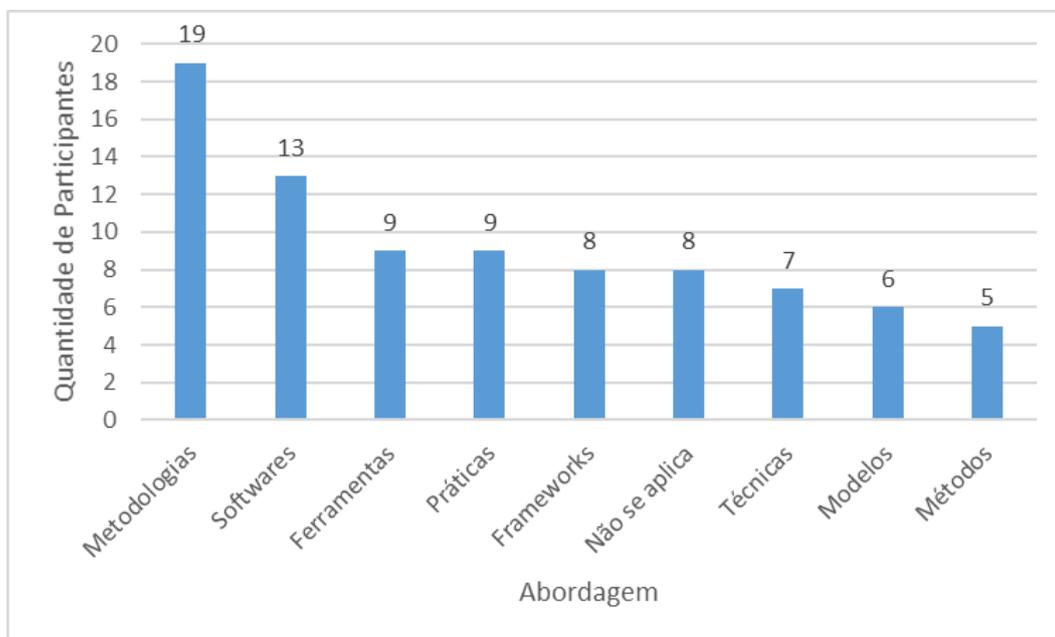
Apesar da maioria dos participantes não usarem nenhuma prática de gerenciamento de projetos das que foram listadas, uma prática ainda pouco explorada merece destaque: a prática de *Fireteam*. Uma equipe de desenvolvimento de software, a depender do tamanho do projeto, pode ser uma equipe enorme, o que dificulta o gerenciamento do pessoal. Pensando nessa questão, o *Fireteam* surge como uma prática na qual equipes muito grandes são desmembradas em equipes menores autogerenciáveis que variam entre três e quatro membros (ZHANG *et al.*, 2020).

#### 4.2.3.9 Pontos Fortes da Abordagens

Encerrando a seção de perguntas sobre as abordagens de gerenciamento de projetos de software, os participantes foram questionados sobre os pontos fortes e fracos dessas abordagens. Como pode ser observado na Figura 29, 19 participantes apontaram que os pontos fortes das abordagens concentram-se principalmente nas metodologias.

Além disso, 13 participantes marcaram os softwares como sendo a abordagem que apresenta a maior quantidade de pontos positivos. Ferramentas e Práticas obtiveram o mesmo número, ambas foram marcadas por nove participantes.

Figura 29 - Abordagens que possuem pontos fortes

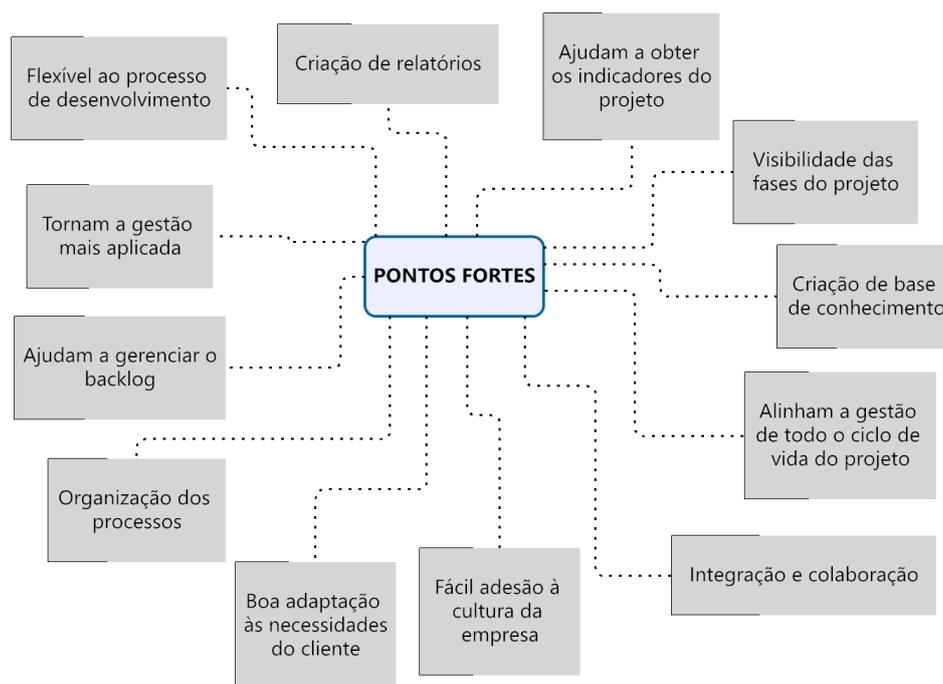


Fonte: O autor (2022)

Ainda em relação aos pontos positivos elencados pelos participantes da pesquisa, oito veem pontos positivos nos *Frameworks*, sete apontaram pontos positivos nas Técnicas, seis marcaram os modelos e cinco citaram os métodos. Destaca-se ainda que, oito participantes marcaram a opção “Não se aplica”, pois não veem pontos positivos nas abordagens elencadas.

Foi solicitado também aos participantes que descrevessem os principais pontos positivos das abordagens. A partir das respostas dos participantes foi realizada uma análise qualitativa e obtidos a partir dessa análise 12 pontos positivos extraídos das respostas, como ilustra a Figura 30.

Figura 30 - Pontos fortes das abordagens citados pelos participantes



Fonte: O autor (2022)

É possível observar na Figura 30 que os participantes relataram que, as abordagens de gestão de projetos de software permitem o **gerenciamento do backlog** e a **criação de relatórios**, o que por sua vez permite aos gestores obter de forma mais detalhada os **indicadores do projeto**. Os indicadores do projeto são informações que podem auxiliar esses profissionais a tomarem decisões mais assertivas e que serão cruciais para o sucesso do projeto.

As abordagens de gestão de projetos dão para os gestores uma **maior visibilidade das etapas do projeto**, o que permite saber quais são as etapas que foram concluídas, o prazo de conclusão, assim como identificar aquelas que ainda estão pendentes, tornando o gerenciamento mais eficiente e controlado e, conseqüentemente, torna os **processos mais organizados**. A organização do projeto, facilitada pelas abordagens, auxilia também em projetos que ainda serão desenvolvidos, uma vez que essas abordagens permitem e exploram a criação de **bases de conhecimentos** que podem ser usadas como apoio em outros projetos da empresa.

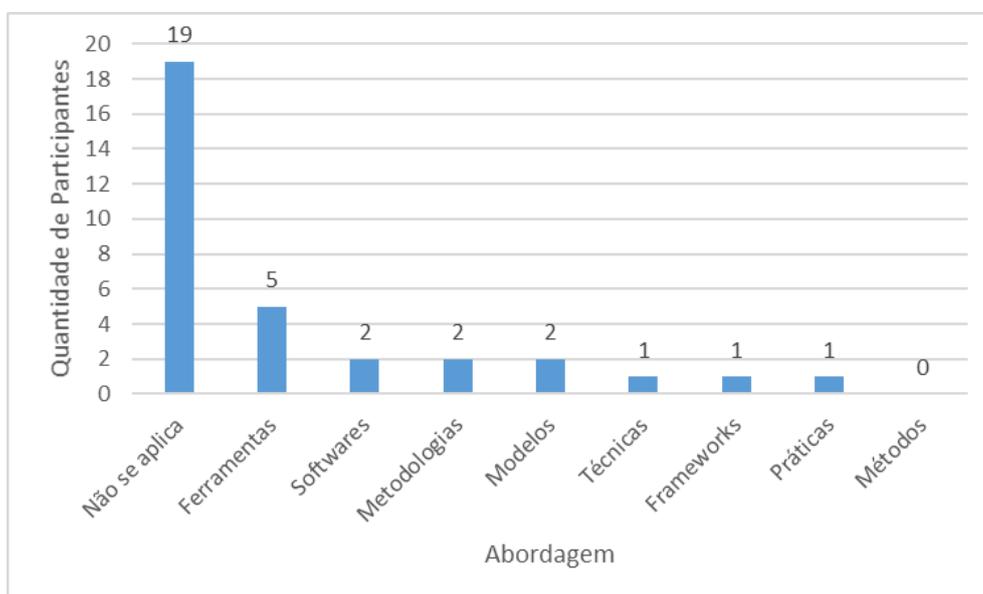
Os participantes destacaram também a importância das abordagens durante todo o ciclo de vida do projeto, pois permitem o **alinhamento de todas as fases do projeto** de forma unificada. O alinhamento entre as fases do projeto, proporcionado por essas abordagens, torna a **gestão mais aplicada**, uma vez que todas as fases foram relacionadas entre si e a equipe passa a conhecer as prioridades e urgências do projeto.

A Figura 30 ainda mostra que as abordagens de gestão de projetos têm uma **fácil adesão à cultura das empresas**, pois se **adaptam às necessidades da equipe e do cliente**. Além disso, essas abordagens são bem quistas pelas equipes, uma vez que **são flexíveis ao processo de desenvolvimento de software**, permitindo uma maior **integração e colaboração** entre os times de desenvolvimento.

#### 4.2.3.10 Pontos Fracos das Abordagens

Assim como foi perguntado aos participantes sobre os pontos fortes das abordagens de gestão de projetos, foi também questionado a eles os pontos fracos dessas mesmas abordagens. A Figura 31 mostra que uma ampla maioria não considera pontos fracos nas abordagens, com um total de 19 participantes que marcaram a opção “Não se aplica”. Por outro lado, dentre aqueles que enxergam pontos negativos nas abordagens, cinco apontaram pontos fracos nas Ferramentas, dois nos Softwares, Metodologias e Modelos. Técnicas, Frameworks e Práticas foram apontados uma única vez cada. Enquanto isso, não foi verificado pontos negativos nos Métodos de gestão de projetos.

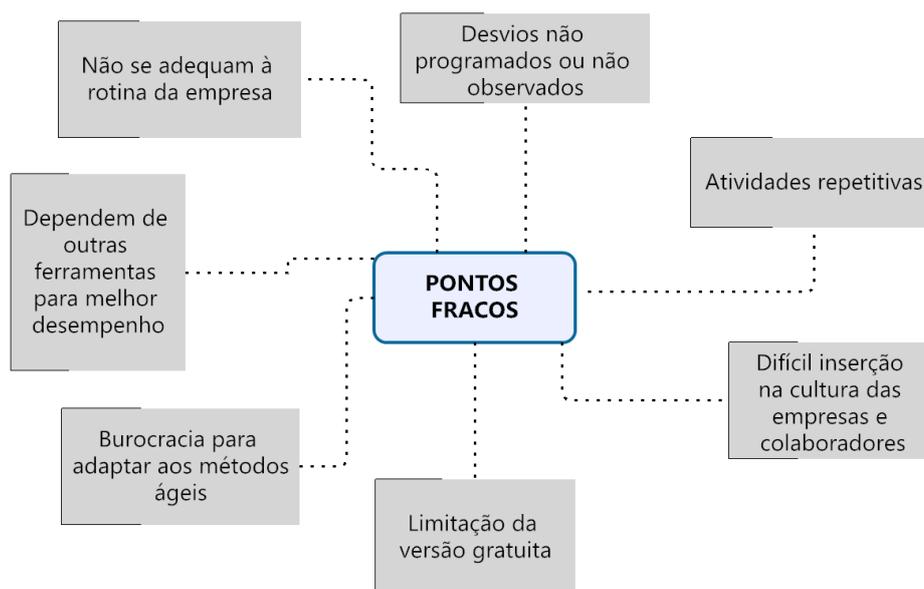
Figura 31 - Abordagens que possuem pontos fracos



Fonte: O autor (2022)

Com relação às respostas abertas, os participantes relataram alguns pontos negativos das abordagens, que foram reunidos em um total de sete códigos, a partir das codificações descritas na seção 4.1.3, como mostra a Figura 32.

Figura 32 - Pontos fracos das abordagens citados pelos participantes



Fonte: O autor (2022)

De acordo com os participantes da pesquisa, algumas abordagens são **difíceis de inserir na cultura das empresas** e de seus colaboradores, pois **não se adequam às rotinas da empresa**. As empresas de desenvolvimento de software têm uma característica peculiar em relação as demais, a alta volatilidade e a rapidez do mercado, o que exige a adoção de métodos de desenvolvimento mais ágeis. Por outro lado, como alguns participantes relataram que algumas abordagens **apresentam um excesso de burocracia** para se adaptarem aos métodos ágeis adotados nessas organizações. A burocracia identificada pelos participantes pode estar diretamente ligada à questão da não adequação às rotinas da empresa.

As abordagens apresentam também, segundo os participantes da pesquisa, alguns **desvios não programados**, isto é, as abordagens não são fiéis ao que acontece diariamente durante a execução de um projeto. Esses desvios relatados pelos participantes podem influenciar diretamente no resultado do projeto, uma vez que é com base nessas ferramentas que muitos indicadores do projeto são extraídos. Outro ponto a se destacar diz respeito às atividades repetitivas. Alguns projetos exigem das equipes a execução de **atividades repetitivas** e, de acordo com algumas respostas dos participantes, as abordagens não conseguem dar o suporte necessário a essas atividades repetitivas, o que acaba levando os gerentes de projetos a controlarem essas atividades de forma mais manual.

Muitas abordagens são desenvolvidas para um fim específico e não necessariamente para gerenciar todo o projeto, o que leva os gestores a terem que usar mais de uma abordagem. Por exemplo, alguns softwares de gestão de projetos **possuem dependências** de outros para funcionarem de forma mais eficiente. A dependência entre abordagens foi citada pelos participantes da pesquisa como um ponto negativo, uma vez que exige a aquisição de mais abordagens para suprir as demandas do projeto. No entanto, é importante destacar que algumas abordagens possuem essa **limitação apenas em suas versões gratuitas**, haja vista que grande parte delas precisam ser compradas para disponibilizarem todas suas funcionalidades, principalmente no que se refere aos softwares.

## 5 CATÁLOGO DE ABORDAGENS DE GERENCIAMENTO DE PROJETOS DE SOFTWARE USADOS NO CONTEXTO DO DESENVOLVIMENTO ÁGIL

Esta seção apresenta o catálogo de abordagens que foram identificadas no MSL. Esta seção apresenta, para cada categoria, três abordagens diferentes. As demais abordagens podem ser encontradas no Apêndice 3. As abordagens descritas neste trabalho foram selecionadas com base no número de citações nos trabalhos do MSL, ou seja, as três mais citadas constam neste capítulo. A partir disso, o catálogo foi estruturado em cinco tópicos, da seguinte forma:

- **Fornecedor / Desenvolvedor:** esse tópico apresenta os responsáveis pelo desenvolvimento e/ou manutenção da abordagem, seja pessoa física ou alguma empresa/organização;
- **Site de acesso:** esse tópico traz um *link* onde o leitor pode consultar mais informações sobre a abordagem, caso tenha interesse;
- **Descrição:** esse tópico apresenta a ferramenta, isto é, na descrição a abordagem é definida de acordo com o que foi identificado nos artigos do MSL, bem como informações adicionais retiradas de outras fontes complementares;
- **Finalidade:** esse tópico traz ao leitor o objetivo principal para o qual a abordagem foi concebida;
- **Como funciona:** esse tópico apresenta um pouco mais da descrição da abordagem e como ela pode ser usada pelos interessados.

Importante destacar que algumas abordagens já são mais difundidas que outras e, portanto, possuem mais informações disponíveis para consulta. Enquanto isso, algumas foram desenvolvidas e apresentadas apenas nos artigos selecionados no MSL. Além disso, vale mencionar que, grande parte dos *links* disponibilizados para mais informações remetem para o artigo de onde a abordagem foi retirada. Isso acontece pois a maioria das abordagens foram frutos de pesquisas acadêmicas e que ainda não foram implementadas massivamente nas indústrias de desenvolvimento de software.

### 5.1 Softwares

- **T3 Software**

**Fornecedor / Desenvolvedor:** Valerie Trapa e Santhpur Rao

**Site de acesso:** <https://ieeexplore.ieee.org/document/1667585>

**Descrição:** A ferramenta T3 é um software leve baseado em JUnit que fornece visibilidade do andamento do projeto para dar suporte ao desenvolvimento rápido. Usando visualizações centradas na história, as equipes ágeis podem avaliar rapidamente o status, determinar dependências e identificar pontos problemáticos. O T3 para relata o status realista do projeto e melhora a comunicação entre os membros da equipe distribuídos geograficamente.

**Finalidade:** O T3 tem como finalidade abordar questões sistemáticas em uma ampla gama de projetos. Além disso, ele busca auxiliar na fácil identificação da complexidade funcional do projeto e a experiência dos desenvolvedores.

**Como funciona:** T3 é um software que calcula a porcentagem de métricas completas do projeto. Além disso, fornece uma visão geral, por meio de relatórios, sobre o tempo de implementação de cada uma das histórias de usuários e do tempo total de implementação de todo o projeto, com base nessas histórias de usuários.

- **MasePlanner**

**Fornecedor / Desenvolvedor:** Roberto Morgan e Frank Maurer

**Site de acesso:** <https://ieeexplore.ieee.org/document/4031753>

**Descrição:** MasePlanner é um software de planejamento ágil que oferece suporte para o planejamento baseado em cartão, facilitando a comunicação não verbal. O MasePlanner foi desenvolvido para melhorar o processo de planejamento, fornecendo às equipes um ambiente digital que oferece suporte ao gerenciamento de informações.

**Finalidade:** O MasePlanner tem como finalidade oferecer suporte a reuniões de planejamento, fornecendo um espaço de trabalho comum semelhante a uma mesa para criar e organizar cartões de história.

**Como funciona:** O MasePlanner suporta a criação de cartões de história e iterações permitindo sua organização. Ele permite o planejamento baseado em cartão que fornece conhecimento espacial da localização dos cartões. O MasePlanner foi projetado como uma ferramenta de suporte e, como tal, requer que exista um canal de voz de alta qualidade para que ocorra uma verdadeira colaboração. O MasePlanner permite que os

artefatos de planejamento sejam criados, editados e organizados de maneira semelhante às reuniões de planejamento em papel.

- **Sistema de Apoio à Decisão de Planejamento de Sprint (SPESS)**

**Fornecedor / Desenvolvedor:** Alhejab Alhazmi e Shihong Huang

**Site de acesso:** <https://ieeexplore.ieee.org/document/8479063>

**Descrição:** O SPESS é um software auxiliar os gestores no planejamento de Sprint. O SPESS é baseado principalmente em três fatores: a competência do desenvolvedor, a antiguidade do desenvolvedor e a dependência da tarefa.

**Finalidade:** Este software visa atribuir as tarefas de cada Sprint aos desenvolvedores garantindo que cada membro da equipe contribua com o máximo de seu potencial, e o planejamento do projeto seja otimizado para o menor tempo possível.

**Como funciona:** O SPESS reduz do tempo da sprint em até um dia. Isso se dá pelo fato de o SPESS ser aplicado considerando que todos os desenvolvedores têm a competência necessária para trabalhar em qualquer tarefa. A razão da diferença neste caso particular é devido ao seu método que depende da igualdade do número de tarefas atribuídas a cada desenvolvedor. Além da antiguidade dos desenvolvedores, o SPESS também leva em consideração dois fatores adicionais de planejamento de software - os níveis de competência dos desenvolvedores e a dependência das tarefas. A competência dos desenvolvedores é um dos principais fatores para determinar a quais desenvolvedores adequados são atribuídos as tarefas que correspondem às suas habilidades. O fator de dependência da tarefa é usado para determinar a ordem em que as tarefas são desenvolvidas. Entre cada iteração no Sprint, o SPESS verifica o status de trabalho dos desenvolvedores e prepara o número de desenvolvedores com os quais trabalhar para a próxima iteração. Como resultado, a ferramenta SPESS atinge seu objetivo principal de que os desenvolvedores trabalhem nas tarefas de acordo com seu nível de habilidade com o tempo de acabamento ideal.

## 5.2 Metodologias

- **BloomSoft**

**Fornecedor / Desenvolvedor:** Edgar Castillo-Barrera, Monica Amador-García, Héctor Pérez-González, Francisco Martínez-Pérez e Francisco Torres-Reyes

**Site de acesso:** <https://ieeexplore.ieee.org/document/8645899>

**Descrição:** A metodologia consiste em um ajuste da Taxonomia de Bloom, ou seja, em uma mudança de hierarquia do nível de Aplicação, que na Taxonomia de Bloom está no nível 3, mas de acordo com o processo de Desenvolvimento de Software passa para o nível 5. A mudança baseia-se na seguinte análise: para construir, integrar e testar um software é necessário ter realizado previamente uma análise e também uma síntese da informação, que corresponde em Engenharia de Software às fases de Design e Programação, respectivamente.

**Finalidade:** Permite ter uma forma ágil de classificar a complexidade das histórias de usuário a partir dos verbos nelas identificadas e, ao mesmo tempo, determinar com essa classificação o estágio de Desenvolvimento de Software a que pertence. Desta forma, a equipe de desenvolvimento pode classificar as histórias em etapas e com isso fazer um melhor planejamento dessas histórias a serem realizadas em cada Sprint.

**Como funciona:** As etapas do processo de Desenvolvimento de Software ou ciclo de vida clássico são: (1) Pré-análise/Planejamento, (2) Análise/Requisitos, (3) Projeto, (4) Desenvolvimento, (5) Testes e (6) Implementação (PRESSMAN, 2015). Considerando e analisando as etapas mencionadas, é feito um ajuste da Taxonomia de Bloom, que consiste em uma mudança de hierarquia do nível de Aplicação, que na Taxonomia de Bloom está no nível 3, mas conforme o processo de Desenvolvimento de Software passa para o nível 5. A mudança baseia-se na seguinte análise: para construir, integrar e testar um programa de Software, é necessário ter feito previamente uma análise e uma síntese da informação, que corresponde em Engenharia de Software às fases de Design e Programação respectivamente. A proposta segue uma analogia com o ciclo de vida do Software, ou seja, o nível de Aplicação no ciclo de vida do Software é ajustado à fase de Teste, gerando um melhor entendimento e permitindo uma determinação da complexidade das histórias de usuários. Para complementar a identificação da complexidade, foram adicionados verbos em cada nível da taxonomia relacionada à área de Engenharia de Software.

- **DevOps**

**Fornecedor / Desenvolvedor:** John Allspaw e Paul Hammond

**Site de acesso:** <https://www.redhat.com/pt-br/topics/devops>

**Descrição:** DevOps vem da junção das palavras Development e Operations que foi mencionado pela primeira vez em 2009 em uma palestra chamada de "10+ Deploys Per Day", por uma dupla que trabalhava no Flickr.

**Finalidade:** O DevOps permite que funções anteriormente isoladas – desenvolvimento, operações de TI, engenharia da qualidade e segurança – atuem de forma coordenada e colaborativa para gerar produtos melhores e mais confiáveis. Ao adotar uma cultura de DevOps em conjunto com as práticas e ferramentas de DevOps, as equipes ganham a capacidade de responder melhor às necessidades dos clientes, aumentar a confiança nos aplicativos que constroem e cumprir as metas empresariais mais rapidamente.

**Como funciona:** Com a implementação de um modelo de DevOps, as equipes de desenvolvimento e operações não ficam mais separadas. Às vezes, essas duas equipes são combinadas em uma só. Os engenheiros trabalham durante o ciclo de vida inteiro do aplicativo, da fase de desenvolvimento e testes à fase de implantação e operações, e desenvolvem várias qualificações não limitadas a uma única função. Em alguns modelos de DevOps, as equipes de controle de qualidade e segurança também podem aumentar sua integração com o desenvolvimento, as operações e todo o ciclo de vida dos aplicativos. Quando a segurança é a prioridade de todos em uma equipe de DevOps, isso é denominado às vezes DevSecOps. Essas equipes usam práticas para automatizar processos que historicamente sempre foram manuais e lentos. Eles usam uma pilha de tecnologia e ferramentas que os ajudam a operar e desenvolver aplicativos de modo rápido e confiável. Essas ferramentas também ajudam os engenheiros a realizar tarefas independentemente (por exemplo, implantação de código ou provisionamento de infraestrutura) que normalmente exigiriam a ajuda de outras equipes, e isso aumenta ainda mais a velocidade da equipe.

- **Scrum**

**Fornecedor / Desenvolvedor:** Scrum.org / Jeff Sutherland

**Site de acesso:** <https://www.scrum.org/>

**Descrição:** O Scrum é uma estrutura que ajuda as equipes a trabalharem juntas. Semelhante a uma equipe de *rugby* (de onde vem o nome) treinando para o grande jogo,

o Scrum estimula as equipes a aprenderem com as experiências, a se organizarem enquanto resolvem um problema e a refletirem sobre os êxitos e fracassos para melhorarem sempre.

**Finalidade:** O Scrum tem como finalidade ser uma estrutura leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptáveis para problemas complexos.

**Como funciona:** A metodologia Scrum propõe que um projeto seja dividido em diversos (pequenos) ciclos de atividades, chamados de sprint, com reuniões frequentes para que a equipe possa alinhar o que vem fazendo e pensar formas de melhorar o processo com agilidade. Essa metodologia propõe que o projeto seja acompanhado sempre bem de perto e passe por mudanças de planejamento o tempo todo, de forma livre e pouco engessada.

### 5.3 Métodos

- **Planning Poker**

**Fornecedor / Desenvolvedor:** Mike Cohn

**Site de acesso:** <https://www.planningpoker.com/>

**Descrição:** O Planning Poker garante que todos os desenvolvedores participem do processo de estimativa e que a opinião de todos seja ouvida, independentemente de estarem entre as pessoas mais influentes do grupo. Também pode reduzir o problema potencial da chamada ancoragem. Ancoragem é o impacto que alguém que apresenta uma estimativa ou expectativa de custo tem nas estimativas subsequentes, mesmo que a estimativa/expectativa seja irrealista.

**Finalidade:** A finalidade do Planning Poker é estimar de forma mais rápida e consensual um conjunto de tarefas que precisam ser desenvolvidas em uma sprint.

**Como funciona:** O Planning Poker consiste na obtenção da estimativa através de um jogo de cartas, o que faz com que todos os membros que fazem parte da equipe participem da estimativa, colocando a sua visão de complexidade, levando em consideração variáveis como tempo e o esforço para pontuar um cartão e após isso chegar a um denominador comum na equipe através de consenso. O Poker no nome vem por conta de as estimativas serem feitas com um baralho. Não um baralho comum, mas um que usa a Sequência de Fibonacci.

- **COSMIC**

**Fornecedor / Desenvolvedor:** Jean-Marc Desharnais, Luigi Buglione e Buğra Kocatürk

**Site de acesso:** <https://dl.acm.org/doi/abs/10.1145/2181101.2181117>

**Descrição:** O COSMIC é um método padronizado de medição do tamanho funcional do software, seja qual for o seu domínio de aplicação. O método COSMIC também foi aceito pela ISO / IEC JTC1 / SC7 como um padrão internacional (ISO / IEC 19761).

**Finalidade:** O COSMIC tem como finalidade reduzir o risco de o projeto sair do escopo definido, controlar o custo e a qualidade do produto desenvolvido e ajudar na conclusão oportuna do projeto.

**Como funciona:** O COSMIC pode ser aplicado desde as fases iniciais, necessitando, pelo menos, de requisitos de alto nível a serem refinados durante o ciclo de vida do projeto. No processo de medição, os movimentos de dados são contados, os quais podem ser definidos como “componentes funcionais básicos que movem um único tipo de grupo de dados”. Existem 4 subtipos de movimentos de dados: Entry (E), eXit (X), Read (R) e Write (W). A soma das movimentações de dados dentro do escopo de medição estabelecido retorna o número de Pontos de Função COSMIC (CFP).

- **FASTDash**

**Fornecedor / Desenvolvedor:** Jacob T. Biehl, Mary Czerwinski, Greg Smith e George G. Robertson

**Site de acesso:** <https://dl.acm.org/doi/10.1145/1240624.1240823>

**Descrição:** FASTDash é uma nova visualização interativa que busca melhorar a percepção das atividades da equipe usando uma representação espacial da base de código compartilhada que destaca as atividades atuais dos membros da equipe.

**Finalidade:** O FASTDash visa contribuir com um sistema de relatórios de conscientização subjacente que pode ser usado para suportar uma infinidade de ambientes de programação, sistemas de repositório e futuras ferramentas de visualização; um novo esquema de codificação de observação que pode ser usado de forma mais ampla para estudos de comportamento de grupo em espaços de trabalho compartilhados.

**Como funciona:** Um desenvolvedor pode determinar rapidamente quais membros da equipe têm arquivos de origem verificados, quais arquivos estão sendo visualizados e

quais métodos e classes estão sendo alterados no momento. A visualização pode ser anotada, permitindo que os programadores complementem as informações da atividade com detalhes de status adicionais. Ele fornece conhecimento imediato de possíveis situações de conflito, como dois programadores editando o mesmo arquivo de origem. O FASTDash foi desenvolvido por meio de design centrado no usuário, incluindo pesquisas, entrevistas em equipe e observação *in situ*.

## 5.4 Modelos

- **Blueprint**

**Fornecedor / Desenvolvedor:** Cristiano P. Godoy, Lanier M. Santos, Andre F. Cruz, Rafael S. Zerbini, Elisangela P. Silva e Cícero A. L. Pahins

**Site de acesso:** <https://dl.acm.org/doi/10.1109/ICGSE.2019.00014>

**Descrição:** O Blueprint é um modelo de desenvolvimento de software baseada nas metodologias ágeis Scrum e Kanban e adaptada à tendência atual do ambiente global de software. O modelo Blueprint é uma ferramenta inspirada em Scrum e Kanban que gerencia entidades de negócios do projeto antes de seu desenvolvimento. O Blueprint apresenta adaptações importantes para Scrum e Kanban para reduzir burocracias indesejadas e para facilitar o desenvolvimento de software global.

**Finalidade:** O modelo Blueprint propõe um gerenciamento de projeto leve que é combinado com uma organização de equipe para incentivar e facilitar a comunicação entre equipes em locais diferentes.

**Como funciona:** O Blueprint propõe uma hierarquia de equipe que pode se adaptar melhor às mudanças nos requisitos de software durante a execução do projeto. O Blueprint define sua hierarquia de equipe como subequipes. O número de membros por subequipe não é definido pelo Blueprint, mas como uma analogia ao Scrum, subequipes devem se concentrar em pequenos grupos. Além disso, o número de subequipes pode mudar durante a execução de um projeto. O Blueprint segue a noção de que o número de membros não representa qualidade ou produtividade. No modelo Blueprint, cada subequipe tem um líder definido como um aspecto da execução do trabalho com responsabilidade. Cada líder de subequipe é responsável pelas atividades e resultados de seu grupo. Ele também é responsável por integrar as tarefas do grupo ao projeto,

coordenando esse processo com os demais líderes. Todos os líderes da subequipe respondem ao líder principal.

- **Rugby**

**Fornecedor / Desenvolvedor:** Stephan Krusche, Lucas Alperowitz, Bernd Bruegge e Martin O. Wagner

**Site de acesso:** <https://dl.acm.org/doi/10.1145/2593812.2593818>

**Descrição:** O Rugby é um modelo de processo ágil que inclui fluxos de trabalho para a entrega contínua de software. Ele permite que os funcionários de meio período trabalhem em uma organização baseada em projetos com vários projetos para a entrega rápida de protótipos e produtos. O Rugby foi projetado para ser usado em organizações baseadas em projetos com vários projetos.

**Finalidade:** O Rugby apresenta uma metodologia leve para desenvolver e lançar protótipos rápidos e aprender com comentários de feedback em ciclos paralelos rápidos.

**Como funciona:** O Rugby é composto por até oito desenvolvedores, um líder de equipe e um líder de projeto. A equipe do projeto é auto-organizada, multifuncional e, portanto, responsável por todos os aspectos de desenvolvimento e entrega de software. O líder do projeto e o líder da equipe cumprem um papel semelhante ao de um Scrum master enquanto estão em um relacionamento de mestre-aprendiz. Enquanto o gerente de projeto já é experiente, o líder da equipe é um desenvolvedor experiente. Assim, ele está familiarizado com a infraestrutura e os aspectos organizacionais do Rugby. Uma tarefa do líder da equipe é organizar a primeira reunião da equipe e garantir que a equipe organize todas as reuniões seguintes de forma estruturada.

- **TaskAllocator**

**Fornecedor / Desenvolvedor:** Saad Shafiq, Atif Mashkoor, Christoph Mayr-Dorn e Alexander Egyed

**Site de acesso:** <https://ieeexplore.ieee.org/document/9461028>

**Descrição:** O TaskAllocator é um modelo que aproveita a arquitetura LSTM (*long short-term memory*), uma variante da rede neural recorrente artificial (RNN) e é frequentemente usada para problemas de classificação de texto.

**Finalidade:** O TaskAllocator tem como finalidade ajudar os gerentes de projeto a alocar a tarefa recém-adicionada ou modificada para a função mais adequada na equipe.

**Como funciona:** O TaskAllocator aprende com os recursos textuais das alocações de tarefas anteriores e prevê a função aproximada das tarefas recebidas. Uma das características distintivas do TaskAllocator é sua arquitetura flexível. Atualmente, o TaskAllocator está explorando a memória de curto prazo para previsões.

## 5.5 Ferramentas

- **aWall**

**Fornecedor / Desenvolvedor:** Magdalena Mateescu, Martin Kropp, Roger Burkhard, Carmen Zahn e Dario Vischi

**Site de acesso:** <https://dl.acm.org/doi/10.1145/2817721.2835072>

**Descrição:** Uma ferramenta ágil de colaboração em equipe para grandes sistemas de parede multitoque. O aWall foi projetado com base em pesquisas empíricas de usuários usando novos conceitos de interação e visualização para apoiar e promover o estilo de trabalho ágil altamente colaborativo e comunicativo. O aplicativo é baseado em tecnologia web e pode ser usado tanto em ambiente colocalizado quanto distribuído. Os protótipos implementados foram validados com usuários finais em uma oficina de usuários. Na demonstração, os usuários podem experimentar os conceitos de interação e visualização na prática.

**Finalidade:** O objetivo é propiciar um processo ágil de intensa interação, colaboração e comunicação aberta constante entre todos os membros da equipe.

**Como funciona:** O aWall oferece suporte à colaboração ágil da equipe e promove o estilo ágil de trabalho por meio de um sistema de parede multitoque extragrande. O aWall tira proveito da resolução 4K e da tecnologia web e com isso supera as possibilidades das ferramentas de desktop existentes. A ferramenta diferencia três aspectos importantes para ferramentas de colaboração ágil: a necessidade de ferramentas para sustentar a atividade principal da equipe ágil (visão de ação), as necessidades de comunicação da equipe ágil e, em terceiro lugar, suas necessidades de atualizar informações que podem ser distribuídas entre os membros da equipe e artefatos (visão de informações). As visualizações de ações foram cuidadosamente projetadas para sustentar a maioria das interações que precisam ser realizadas durante uma reunião específica. A visualização de

informações fornece informações extras necessárias para uma reunião e, com isso, contribui indiretamente para o alcance da meta. As visualizações de informações representam a memória transacional da equipe. Qualquer sistema de memória precisa permitir mudanças. Portanto, as informações representadas nessas visualizações podem ser manipuladas a qualquer momento.

- **Quadro de Equipes Virtuais**

**Fornecedor / Desenvolvedor:** Olga Liskin e Kurt Schneider

**Site de acesso:** <https://ieeexplore.ieee.org/document/6337316>

**Descrição:** Um quadro de equipes virtuais é uma ferramenta para visualizar a estrutura entre histórias de usuários. Abandonar estruturas fixas para representar histórias do usuário dá aos usuários liberdade para especificar os atributos do cartão da história por si próprios. A posição de um cartão no quadro branco ou em relação a outros cartões, bem como muitos outros atributos, pode ter um significado e, portanto, fornecer informações adicionais que estendem o conteúdo puro.

**Finalidade:** O quadro de equipes virtuais tem como finalidade proporcionar maior visibilidade do status do projeto e das tarefas para equipes que trabalham geograficamente distribuídas.

**Como funciona:** Inicialmente, pode ser implementado um catálogo de atributos visuais e uma ferramenta de quadro de equipe que ofereça suporte a esses atributos. Após isso, uma avaliação pode ser conduzida para verificar o benefício e o esforço de usar a abordagem do quadro de equipe para diferentes tarefas e a comparamos com ferramentas baseadas em tabelas.

- **ASAP (*Agile SoftwAre Planning Tool*)**

**Fornecedor / Desenvolvedor:** Uffe Kock Wiil e Rasmus Rosenqvist Petersen

**Site de acesso:** <https://dl.acm.org/doi/abs/10.1145/1379092.1379101>

**Descrição:** Um Espaço no ASAP é uma grande superfície 2D usada para organizar os elementos de informação de acordo com seus relacionamentos. O Cartão de Tarefas é o elemento básico de informação usado no ASAP. Ele representa o equivalente em papel com uma grade na qual cada célula recebe um significado específico. O valor das células pode ser alterado pelo usuário. Os elementos de informação são conhecidos em todos os

domínios de estruturação de hipertexto. Os Cartões de Tarefas são criados com um gesto de arrastar dentro do Espaço. Um cartão de tarefas é excluído ao pressionar a “cruz vermelha” no canto superior direito.

**Finalidade:** A primeira versão do ASAP mostrou que uma seleção cuidadosamente escolhida (combinação) de conceitos existentes de vários domínios de estruturação de hipertexto pode fornecer suporte para a gestão do conhecimento complexo tarefa que está planejando.

O novo conceito de Separador foi introduzido como um novo mecanismo de estruturação. O Separador é principalmente interessante em relação ao domínio de estruturação espacial ao trabalhar com certos tipos de tarefas de gestão do conhecimento (como planejamento de projetos).

**Como funciona:** O ASAP substitui os cartões de tarefas em papel por cartões eletrônicos e fornece uma grande superfície de tela para se parecer com a superfície da mesa. A configuração típica de implantação para ASAP é um SMART Board. A atividade de planejamento Blitz é composta de dez etapas gerais.

Etapa 1: Reúna os participantes. Os planejadores podem digitar seus nomes o mais rápido possível. Os nomes são usados ao distribuir tarefas entre os membros da equipe e são visíveis na Visualização de Tarefas Distribuídas.

Etapa 2: Faça um *brainstorming* das tarefas. Os planejadores podem criar novos cartões de tarefas com um simples gesto. Os planejadores podem digitar os nomes das tarefas recém-criadas na célula TASK.

Etapa 3: Organize as tarefas. Os planejadores podem organizar os cartões de tarefas na superfície 2D do ASAP, posicionando-os ao lado dos cartões de tarefas relacionados.

Etapa 4: Revise as tarefas. Os planejadores têm uma visão geral fácil de compreender dos Cartões de Tarefas dispostos como resultado da Etapa 3. Os planejadores podem reorganizar os Cartões de Tarefas nesta etapa da mesma maneira que na Etapa 3, se assim o desejarem.

Etapa 5: Estime e marque as tarefas. Os planejadores podem inserir estimativas para cada tarefa (digitando a estimativa na célula TEMPO). Os planejadores podem atribuir tarefas aos membros da equipe (digitando o nome da pessoa na célula PESSOA).

Etapa 6: Classifique as tarefas. Os planejadores podem classificar as tarefas em iterações sequenciais usando os separadores, colocando os cartões de tarefas em relação aos separadores. Os planejadores podem ter uma visão geral das iterações usando o plano de projeto gerado automaticamente.

Etapa 7: Marque o esqueleto ambulante, a versão mais antiga e a receita mais antiga. Os planejadores podem usar Separadores para indicar marcos importantes do projeto, bem como esqueleto ambulante, versão inicial e receita inicial.

Etapa 8: Identifique outras versões. Os planejadores podem adicionar separadores adicionais para dar suporte a esta etapa.

Etapa 9: Otimize o plano para se adequar às prioridades do projeto. Os planejadores podem mover os cartões de tarefas para melhor corresponder às prioridades do projeto.

Etapa 10: Capture a saída. Atualmente, existem duas opções para capturar a saída. Os planejadores podem armazenar o plano do projeto, permitindo que ele seja recuperado e revisado posteriormente. Os planejadores podem exportar o projeto em formato XML para ser usado por ferramentas de gerenciamento de projetos de terceiros (se desejado).

## 5.6 Técnicas

- **FLOW Mapping**

**Fornecedor / Desenvolvedor:** Kai Stapel, Eric Knauss, Kurt Schneider e Nico Zazworka

**Site de acesso:** <https://ieeexplore.ieee.org/abstract/document/6063167>

**Descrição:** Um FLOW Mapping é uma técnica FLOW especial, ou seja, visualização dos participantes do projeto, documentos e fluxos de informações, estendido por recursos para melhorar a conscientização em equipes distribuídas.

**Finalidade:** FLOW Mapping é uma técnica usada para planejar e orientar a comunicação em projetos de desenvolvimento distribuído de software.

**Como funciona:** O FLOW Mapping funciona com metáfora do estado da informação para distinguir entre informação sólida, ou seja, acessível a longo prazo, acessível repetível e compreensível por terceiros, ou fluida, ou seja, informação que viola um dos critérios sólidos. Representantes típicos de informações sólidas são documentos formais ou e-mails formais. A informação fluida, por outro lado, é o conhecimento na mente das pessoas, ou mídia que depende fortemente do conhecimento das pessoas para uma

interpretação correta, como notas informais, e-mails informais ou o conteúdo de conversas, por exemplo, por telefone.

- **OPENTCQ**

**Fornecedor / Desenvolvedor:** Jamshaid Iqbal Janjua, Aatka Ali e Muhammad Umar Chaudhry

**Site de acesso:** <https://dl.acm.org/doi/10.1145/2967878.2967920>

**Descrição:** A maioria dos modelos e técnicas para o gerenciamento de mudanças compartilham uma estratégia comum, e o processo ou análise de cada iteração é ignorado, por isso encontrou a falha para incorporar a mudança. O formulário de solicitação de mudança é documentado para cada vez que o requisito de mudança surge. Uma estratégia sólida de estimativa de requisitos está faltando para lidar com a mudança. O OPENTCQ, processa cada iteração antes de sua integração ao produto como um todo. Essa técnica é, na verdade, customizada a partir do *Unified Process* (UP) e do *Agile Model*.

**Finalidade:** A OPENTCQ se propõe a dimensionar o modelo tradicional para atingir o nível de agilidade.

**Como funciona:** A OPENTCQ permite ter uma forma ágil de classificar a complexidade das histórias de usuário a partir dos verbos nelas identificadas e, ao mesmo tempo, determinar com essa classificação o estágio de desenvolvimento de software a que pertence. Desta forma, a equipe de desenvolvimento pode classificar as histórias em etapas e com isso fazer um melhor planejamento dessas histórias a serem realizadas para cada Sprint.

- **Arquiteturas de Software Distribuídas Mistas**

**Fornecedor / Desenvolvedor:** Jorge Valenzuela Posadas

**Site de acesso:** <https://ieeexplore.ieee.org/document/8079698>

**Descrição:** A arquiteturas mista são usados para atender as reais exigências do governo, porém quando há picos de demanda em prazos específicos, a abordagem de resposta de microserviços melhora a economia de custos de infraestrutura e responde de forma mais ágil por sua escalabilidade granular.

**Finalidade:** A arquitetura mista permite trazer e compartilhar informações em escala massiva de lugares remotos onde realizam projetos socioprodutivos em diversos setores.

**Como funciona:** Para implementar ambas as arquiteturas, usa-se o .Net framework, seleciona-se ASP.Net Web API para o desenvolvimento de microsserviços, pois oferece a opção de *deploy* em *On-Premise Server* e depois, a opção de *deploy* em Cloud migrando para ASP.Net Core ou utilizando framework de microsserviços puro *Azure Service Fabric*, sendo a última opção a melhor maneira de implantar em Cloud. Para aplicação monolítica, seleciona-se ASP.Net MVC porque permitem criar uma base de código tradicional e usar serviços WCF (baseado em XML) por meio de contratos de serviço de proxy. Ambos os aplicativos podem ser implantados em servidores IIS e a arquitetura de microsserviços pode ser implantada nos Serviços de Nuvem do Azure. MySQL e Oracle podem ser usados como banco de dados relacional na arquitetura de microsserviços e apenas Oracle pode ser usado na arquitetura monolítica.

## 5.7 Frameworks

- **Eagle**

**Fornecedor / Desenvolvedor:** Alejandro Guerreiro, Rafael Fresno, An Ju, Armando Fox, Pablo Fernández, Carlos Müller e A. Ruiz-Cortés

**Site de acesso:** <https://dl.acm.org/doi/abs/10.1145/3338906.3341181>

**Descrição:** A abordagem permite um esclarecimento ao tornar o status, os desafios e o progresso do projeto transparentes para todos os participantes a qualquer momento. Recursos podem ser atribuídos aos problemas que causam estagnação na conformidade métrica. Além disso, os pesquisadores se concentraram em desafios relevantes, resolvendo o caso de negócios. O foco nos valores de negócios por meio de métricas reduziu essencialmente o risco de falha.

**Finalidade:** O Eagle tem como finalidade dar maior transparência para as atividades do projeto.

**Como funciona:** A fim de rastrear a aderência de uma equipe às diferentes práticas, desenvolvemos um painel interativo que é gerado automaticamente a partir do TPA4 que tanto os membros da equipe quanto os gerentes de projeto podem usar. O painel permite selecionar um período de tempo para analisar, por exemplo, uma iteração que foi insatisfatória para detectar as práticas que foram violadas. Além disso, uma anotação sobre o gráfico é possível para destacar comentários persistentes ao longo da linha do tempo para análise posterior. Além disso, cada gráfico está vinculado a uma tabela

complementar acima dele com o grau de cumprimento dos TPs para cada dia; caso o objetivo não seja cumprido é destacado com fundo vermelho e com fundo verde caso contrário. Caso sejam necessárias mais informações, os usuários podem obter uma lista de evidências que suportam esse grau de cumprimento.

- **Framework de Gerenciamento de Projetos para o Desenvolvimento Global de Software**

**Fornecedor / Desenvolvedor:** Ritu Jain, Ugrasen Suman

**Site de acesso:** <https://dl.acm.org/doi/10.1145/3178315.3178329>

**Descrição:** Esse framework assimila as áreas de conhecimento do PMBOK com as áreas de conhecimento necessárias para uma gestão eficaz do GSD. Ele orientaria o gerente de projetos da GSD sobre os aspectos a serem considerados na execução de projetos distribuídos. Essa estrutura também atuaria como uma linha de base para pesquisadores para investigações adicionais no domínio de gerenciamento de projetos GSD.

**Finalidade:** Esse framework tem como finalidade proporcionar um ambiente de desenvolvimento mais coerente com a realidade da equipe e mais robusto para desenvolver projetos atuais.

**Como funciona:** A estrutura proposta abrange gerenciamento de viabilidade e risco, gerenciamento de equipe virtual, gerenciamento de conhecimento, gerenciamento de escopo e recursos, gerenciamento de desempenho e gerenciamento de integração GSD. O framework garante uma operação bem-sucedida de projetos GSD, pois auxilia no gerenciamento de equipe virtual robusto, garantindo coesão de equipe, configuração de equipe, motivação, alocação de tarefas e gerenciamento de conflitos.

- **Framework de Medição de Agilidade para Gerenciamento de Projetos de Software**

**Fornecedor / Desenvolvedor:** Mourad Mounir, Akram Salah, Amr Kamel e Hanan Moussa

**Site de acesso:** <https://dl.acm.org/doi/abs/10.1145/3436829.3436853>

**Descrição:** A principal noção dessa abordagem é que para que um processo seja verdadeiramente ágil, os valores ágeis devem ser alcançados independentemente de quais práticas ágeis são usadas e sem ditado para as escolhas das práticas. Concentrar-se apenas

nos valores ágeis também torna muito simples e fácil conduzir a avaliação sem seguir uma abordagem complexa. Isso apoiará a ideia de fazer a avaliação de agilidade mais de uma vez em diferentes períodos apenas para garantir que haja uma melhoria na agilidade ao longo do cronograma do projeto.

**Finalidade:** Esse framework se concentra em medir a eficácia do Agile e a medição da agilidade é incluída para garantir que a causa e o efeito do Agile sejam medidos corretamente no desenvolvimento de sistemas críticos.

**Como funciona:** Esse framework de medição não combina todas as abordagens já usadas para medir a agilidade do processo ou a eficácia do Agile em sistemas críticos. Essa abordagem de medição de agilidade vai depender da medição da conquista dos valores culturais Agile pelo processo atual adotado. Os princípios e práticas ágeis adotados têm seu efeito no alcance dos valores ágeis, porém recomenda-se a liberdade de escolha das práticas a serem adotadas, pois cada projeto ou organização possui suas próprias restrições, ambiente e ferramentas específicas. A principal noção dessa abordagem é que para que um processo seja verdadeiramente ágil, os valores ágeis devem ser alcançados independentemente de quais práticas ágeis são usadas e sem ditado para as escolhas das práticas. Concentrar-se apenas nos valores ágeis também torna muito simples e fácil conduzir a avaliação sem seguir uma abordagem complexa. Isso apoiará a ideia de fazer a avaliação de agilidade mais de uma vez em diferentes períodos apenas para garantir que haja uma melhoria na agilidade ao longo do cronograma do projeto.

## 5.8 Práticas

- **MEDIATION - Colaboração de Pesquisa Orientada por Métricas**

**Fornecedor / Desenvolvedor:** Marc Schreiber, Bodo Kraft e Albert Zündorf

**Site de acesso:** <https://ieeexplore.ieee.org/document/7964364>

**Descrição:** De acordo com o nome, a definição de métricas é a tarefa mais importante em uma colaboração de pesquisa; a definição de métricas traz definições de objetivos comuns em foco. Cada métrica está vinculado aos requisitos globais do projeto. A colaboração de pesquisa deve elaborar um consenso sobre um pequeno número de métricas de alto nível que colocam os requisitos mais importantes em foco. A princípio, todos os participantes devem escolher um requisito e definir uma métrica correspondente por completo. A métrica deve: representar uma meta comum de negócios de alto nível, ser avaliado

automaticamente, corresponder ao status do projeto, ter um limite inicial e, ter um rico conjunto de dados de entrada e saída de exemplo.

**Finalidade:** A abordagem tem como finalidade o esclarecimento da equipe ao tornar o status, os desafios e o progresso do projeto transparentes para todos os participantes a qualquer momento.

**Como funciona:** A abordagem garante continuamente que todos os participantes do projeto tenham um objetivo comum: o sucesso do projeto. Depois de aplicar o MEDIATION o desenvolvimento e o progresso tornaram-se mais eficazes. Recursos podem ser atribuídos aos problemas que causam estagnação na conformidade métrica. Todos os participantes foram convencidos de que o produto de software desenvolvido aumenta o valor do negócio. Além disso, os pesquisadores se concentraram em desafios relevantes, resolvendo o caso de negócios. O foco nos valores de negócios por meio de métricas reduziu essencialmente o risco de falha. Por meio da abordagem MEDIATION, a equipe do projeto concentra-se nos requisitos mais importantes e verifica continuamente se o produto de software está em conformidade com as métricas correspondentes.

- **Fireteams**

**Fornecedor / Desenvolvedor:** He Zhang, Huang Huang, Dong Shao e Xin Huang

**Site de acesso:** <https://dl.acm.org/doi/10.1145/3368089.3417068>

**Descrição:** A fim de reduzir as despesas gerais de gerenciamento de projetos e melhorar a produtividade, uma empresa global de tecnologia da informação e comunicação institucionaliza uma prática de equipe pequena em toda a organização, chamados Fireteams, para resolver os problemas decorrentes de aspectos humanos e sociais, como amizade, talento, habilidade e comunicações.

**Finalidade:** Os Fireteams tem como objetivo gerar menos sobrecarga de comunicação e gerenciamento, maior agilidade e simultaneidade e melhor habilidade pessoal, com equipes reduzidas.

**Como funciona:** Os Fireteams funcionam a partir da divisão de equipes em equipes menores (geralmente 4 a 5 pessoas), o que garante maior controle para os gerentes, menos despesas gerais de comunicação, maior agilidade e simultaneidade e capacidade pessoal aprimorada e, conseqüentemente, aumentam a produtividade dos esquadrões.

## 6 CONCLUSÕES

Este capítulo apresenta as principais conclusões desta pesquisa. Além disso, serão apresentadas as principais contribuições deste trabalho, assim como as limitações identificadas e os trabalhos futuros a serem executados a partir do estudo realizado.

### 6.1. Visão Geral

A gerência de projetos é parte crucial para um desenvolvimento de software bem sucedido, pois permite o desenvolvimento controlado e com alto grau de qualidade, sem extrapolar o prazo e o orçamento estabelecido. Contudo, é necessário que os profissionais da área possuam ferramentas capazes de auxiliá-los no processo de gerenciamento, conforme aponta Carvalho, Malcher e Santos (2020).

Assim, esta dissertação apresentou uma pesquisa exploratória e descritiva, que foi conduzida mediante a realização de um MSL, aplicação de *survey* e elaboração de um catálogo, onde ambos têm como objetivo identificar as abordagens de gerenciamento de projetos utilizadas na indústria de desenvolvimento ágil de software.

Com base nos resultados, foi possível observar que os softwares são as abordagens mais comuns dentre todas as demais, seguido das metodologias. Verificamos alguns pontos positivos relacionados ao uso dessas abordagens, como por exemplo: reduzir o esforço necessário por parte da equipe; proporcionar uma maior facilidade no que tange o planejamento das comunicações; e reduzir do tempo da Sprint, quando se trabalha com o Scrum. Além disso, identificamos alguns pontos negativos, como podemos destacar: há abordagens que dependem integralmente de um comprometimento igual de todos os integrantes da equipe e há abordagens que requerem muitas tarefas na fase de inicial do projeto.

### 6.2. Resultados Obtidos

Durante o desenvolvimento desta pesquisa de mestrado alguns artefatos foram gerados e publicados, contribuindo, assim, para o estado da arte da área de gerência de projetos de software. Dentre os artefatos gerados, temos:

- Um artigo completo publicado no ano de 2021 na “*18th International Conference on Information Systems and Technology Management*”, intitulado “*Analysis of a*

*Systematic Literature Review Protocol on Agile Approaches Used for Software Project Management*”, no qual foi realizado a análise de protocolos de revisão sistemática e comparado com o protocolo desenvolvido para o MSL apresentado nesta dissertação;

- Dois artigos publicados no de 2022 na “*19th International Conference on Information Systems and Technology Management*”, intitulados “*A Survey Research on The Use of Project Management Approaches in Agile Software Development*” e “*Catalog of Software Project Management Approaches Used in the Context of Agile Development*”. O primeiro artigo refere-se ao *survey* realizado com gerentes de projetos de software, com o intuito de verificar se as abordagens identificadas no MSL são usadas na indústria de software. O segundo, por sua vez, trata-se do catálogo contendo as abordagens identificadas no MSL, bem como suas características;
- Um artigo publicado no ano de 2022 na “*17th International Conference on Evaluation of Novel Approaches to Software Engineering*”, intitulado “*The Diversity of Approaches to Support Software Project Management in the Agile Context: Trends, Comparisons and Gaps*”, no qual foi descrito os resultados do mapeamento sobre as abordagens de gerenciamento de projetos de software. Este artigo foi eleito como um dos melhores trabalhos de sua trilha, sendo premiado com a publicação de uma versão estendida e será indexado na Springer.

Abaixo constam as referências dos trabalhos publicados:

CARVALHO, Elielton; OLIVEIRA, Sandro. The Diversity of Approaches to Support Software Project Management in the Agile Context: Trends, Comparisons and Gaps. In: 17TH INTERNATIONAL CONFERENCE ON EVALUATION OF NOVEL APPROACHES TO SOFTWARE ENGINEERING, 2022, Online Streaming, --- Select a Country ---. **17th International Conference on Evaluation of Novel Approaches to Software Engineering**. [S. l.]: SCITEPRESS - Science and Technology Publications, 2022. Disponível em: <https://doi.org/10.5220/0011063500003176>. Acesso em: 16 dez. 2022.

CARVALHO, ELIELTON DA COSTA; OLIVEIRA, SANDRO RONALDO BEZERRA. ANALYSIS OF A SYSTEMATIC LITERATURE REVIEW PROTOCOL ON AGILE APPROACHES USED FOR SOFTWARE PROJECT MANAGEMENT. In: CONTECSI INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT. **CONTECSI International Conference on Information Systems and Technology Management**. [S. l.]: TECSI. Disponível em: <https://doi.org/10.5748/18contecsi/pse/prm/6733>. Acesso em: 16 dez. 2021.

CARVALHO, ELIELTON DA COSTA; OLIVEIRA, SANDRO RONALDO BEZERRA. A SURVEY RESEARCH ON THE USE OF PROJECT MANAGEMENT APPROACHES IN AGILE SOFTWARE DEVELOPMENT. *In: CONTECSI INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT. CONTECSI International Conference on Information Systems and Technology Management*. [S. l.]: TECSI. Disponível em: <https://doi.org/10.5748/18contecsi/pse/prm/6733>. Acesso em: 16 dez. 2022.

CARVALHO, ELIELTON DA COSTA; OLIVEIRA, SANDRO RONALDO BEZERRA. CATALOG OF SOFTWARE PROJECT MANAGEMENT APPROACHES USED IN THE CONTEXT OF AGILE DEVELOPMENT. *In: CONTECSI INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT. CONTECSI International Conference on Information Systems and Technology Management*. [S. l.]: TECSI. Disponível em: <https://doi.org/10.5748/18contecsi/pse/prm/6733>. Acesso em: 16 dez. 2022.

### **6.3. Contribuições da Pesquisa**

A primeira contribuição deste trabalho trata-se da divulgação de um protocolo que pode ser replicado por outros trabalhos semelhantes, pois, como definido por Kitchenham e Charters (2007), um protocolo de MSL deve ser passível de replicação.

Outra contribuição deste trabalho foi detalhar as características das abordagens de gerenciamento utilizados por gerentes de projetos de software a partir de um MSL. Verificou-se, com isso, que a maioria das abordagens estão direcionadas para o gerenciamento da comunicação, do cronograma e da qualidade dos projetos. Além disso, as abordagens que possuem o maior número de soluções são os softwares, as metodologias e os métodos, respectivamente. Com isso, nota-se que as abordagens de gerenciamento seguem a tendência da informatização, onde hoje os serviços mais essenciais são realizados com o auxílio de algum tipo de software.

Os resultados deste trabalho podem ser úteis para que organizações e os profissionais que atuam no mercado de desenvolvimento de software conheçam as principais abordagens de gerenciamento de projetos de software, bem como a forma como eles podem ser utilizados. Também podem ajudar os desenvolvedores e pesquisadores na definição de requisitos que possam atualizar, melhorar e até criar novas soluções com base no que foi apresentado neste trabalho.

### **6.4. Limitações da Pesquisa**

Quando estamos falando de pesquisas que envolvem participação de terceiros, algumas limitações são ainda mais evidentes. Em relação aos participantes da pesquisa, o questionário foi encaminhado para os perfis pessoais dos gerentes de projetos que constavam no LinkedIn. Apesar da carta de apresentação solicitar que os gerentes de projetos respondessem, não há 100% de garantia que todos os respondentes ocupassem o cargo de gerente. Todavia, é importante destacar que, em alguns casos, o dono da empresa ou até mesmo um desenvolvedor sênior acaba desempenhando a função de um gerente de projetos, apenas não usa essa nomenclatura (JAHAN *et al.*, 2019).

Alguns participantes responderam as questões abertas de forma muito direta, o que gerou uma dificuldade a mais na hora de analisar esses dados. Porém, essa dificuldade já era esperada, uma vez que, de acordo com Soares (2019), pesquisas que envolvem codificações são extremamente trabalhosas. Ainda sobre as codificações, essa etapa pode ter levado em consideração a experiência dos pesquisadores envolvidos.

Sobre o viés inerente às pesquisas qualitativas, Pinto e Santos (2012) afirmam que esse viés pode limitar a pesquisa, uma vez que os pesquisadores estão expostos a diversas maneiras de interpretar os dados. Os autores continuam dizendo que, por causa da subjetividade nas análises e interpretações, ao analisar novamente os mesmos dados, outros pesquisadores podem enxergar algo que pode ter passado despercebido na primeira análise.

Com todas essas limitações que, como pode ser observado, são comuns em pesquisas que envolvem procedimentos de codificação, os resultados deste trabalho foram verificados e ajustados após a análise de um pesquisador com ampla experiência em pesquisas qualitativas em Engenharia de Software, o que minimizou alguns dos problemas descritos e comentados nos parágrafos anteriores.

Com relação às limitações do MSL, pode-se destacar o viés que esse tipo de pesquisa possui, pois ao analisar um trabalho, um determinado pesquisador poderia incluí-lo, enquanto um segundo pesquisador poderia excluí-lo. No entanto, para minimizar o viés inerente ao MSL, os achados foram validados pelo orientador desta pesquisa, que possui mais de 14 anos de experiência com pesquisas dessa linha. Ainda sobre as limitações do MSL tem-se as seguintes ameaças à validade, bem como as formas utilizadas para mitigá-las:

- **Validade de construção**

Para minimizar o risco de que o MSL não trouxesse os estudos que respondessem às questões de pesquisa, foi realizado um teste com a *string* de busca. Foram selecionados manualmente seis estudos que provaram atender aos objetivos da pesquisa e, em seguida, verificou-se se, ao executar a *string* nas bases, esses mesmos estudos retornariam, o que de fato aconteceu.

- **Validade interna**

Durante o processo de extração os estudos foram classificados com base no julgamento dos autores desta dissertação. Estudos que dependem do julgamento dos autores podem carregar consigo um viés que precisa ser mitigado ao máximo. Pensando nisso, durante todo o processo de análise dos estudos foram realizadas reuniões semanais para discutir e chegar a um consenso sobre quais estudos realmente deveriam ser selecionados.

- **Validade externa**

Era possível que o MSL não retornasse todos os estudos relevantes sobre abordagens que apoiam o gerenciamento de projetos de software. Para mitigar esse risco foram identificados e usados estudos semelhantes a este, sendo a base para o desenvolvimento do MSL apresentado nesta dissertação.

- **Validade de conclusão**

Este trabalho tem sua validade de conclusão pautada no seguintes argumentos: O MSL foi conduzido seguindo estritamente um protocolo que, por sua vez, foi elaborado baseado nas melhores práticas definidas e testadas na literatura. Enquanto isso, o survey obteve uma amostra substancial, se levarmos em consideração a quantidade média de participantes que costumam responder esse tipo de pesquisa. Além disso, o survey conseguiu obter respostas de participantes que atuam em todas as regiões do Brasil, com idades diferentes, bem como com níveis de experiência distintos. Com isso, as respostas obtidas conseguiram mapear de forma satisfatória a realidade do tema desta dissertação.

## **6.5. Trabalhos Futuros**

A partir dos resultados foi possível verificar o quanto e como as abordagens de gerenciamento de projetos são importantes para o desenvolvimento de software com mais qualidade, além de poder enxergar algumas limitações que ainda devem ser mitigadas, o que dá, portanto, oportunidades aos pesquisadores da área para desenvolverem mais na área.

A partir desta pesquisa, alguns trabalhos futuros podem ser desenvolvidos a fim de contribuir com o estado da arte em gerenciamento de projetos de software. Entre eles, destacam-se: (i) identificar outras características das abordagens de gerenciamento a partir das necessidades dos participantes e realizar uma nova pesquisa de opinião; (ii) investigar formas de mitigação para alguns dos problemas identificados neste trabalho; (iii) elaborar um roteiro para a escolha da abordagem mais adequada para um determinado projeto ou organização; e (iv) desenvolver e aplicar uma nova abordagem de gerenciamento de projeto de software, levando em consideração os resultados descritos nesta dissertação.

## REFERÊNCIA BIBLIOGRÁFICAS

AL Aidaros, Hamzah; Omar, Mazni; Romli, Rohaida; Hussein, Adnan. The Development and Evaluation of A Progress Monitoring Prototype Tool for Software Project Management. *In: 2019 FIRST INTERNATIONAL CONFERENCE OF INTELLIGENT COMPUTING AND ENGINEERING (ICOICE)*, 2019, Hadhramout, Yemen. **2019 First International Conference of Intelligent Computing and Engineering (ICOICE)**. [S. l.]: IEEE, 2019. ISBN 9781728144870. Disponível em: <https://doi.org/10.1109/icoice48418.2019.9035146>. Acesso em: 19 dez. 2022.

ALVES, Junqueira Junior. **Estudo Comparativo entre Métodos Ágeis e Tradicionais de Fabricação de Software**. 2012. Disponível em: <https://docplayer.com.br/1087982-Estudo-comparativo-entre-metodos-ageis-e-tradicionais-de-fabricacao-de-software-trabalho-de-conclusao-de-curso-engenharia-da-computacao.html>. Acesso em: 18 dez. 2022.

ALVES, Junqueira Junior. **Estudo Comparativo entre Métodos Ágeis e Tradicionais de Fabricação de Software**. 2012. Disponível em: <https://docplayer.com.br/1087982-Estudo-comparativo-entre-metodos-ageis-e-tradicionais-de-fabricacao-de-software-trabalho-de-conclusao-de-curso-engenharia-da-computacao.html>. Acesso em: 04 nov. 2019.

ARAÚJO, Saint-Clair Cardoso de. **Métodos de pesquisa**. Universidade Católica de Brasília, 2000.

ARTELT, Sascha. Analysing the Impact of Agile Project Management on Organisations. *In: 2021 IEEE EUROPEAN TECHNOLOGY AND ENGINEERING MANAGEMENT SUMMIT (E-TEMS)*, 2021, Dortmund, Germany. **2021 IEEE European Technology and Engineering Management Summit (E-TEMS)**. [S. l.]: IEEE, 2021. Disponível em: <https://doi.org/10.1109/e-tems51171.2021.9524863>. Acesso em: 18 dez. 2022.

AZIZYAN, G.; MAGARIAN, M. K.; KAJKO-MATSSON, M.. Survey of Agile Tool Usage and Needs. *In: 2011 AGILE CONFERENCE*, 2011, Salt Lake City, UT, USA. **2011 AGILE Conference**. [S. l.]: IEEE, 2011. Disponível em: <https://doi.org/10.1109/agile.2011.30>. Acesso em: 19 ago. 2022.

BABBIE, E.. Métodos de pesquisas de survey. 1. ed. Belo Horizonte: **Editora UFMG**, 1999.

BADEWI, Amgad. The impact of project management (PM) and benefits management (BM) practices on project success: Towards developing a project benefits governance framework. **International Journal of Project Management**, v. 34, n. 4, p. 761-778, 2016.

BAJTA, Manal El et al. Software project management approaches for global software development: a systematic mapping study. **Tsinghua Science And Technology**, [S.l.], v. 23, n. 6, p. 690-714, dez. 2018.

- BASTOS, Eric Casagrande; BARCELLOS, Monalessa Perini; FALBO, Ricardo de Almeida. Using Semantic Documentation to Support Software Project Management. **Journal On Data Semantics**, v. 7, p. 107-132, jun. 2018.
- BECK, K. et al. **Manifesto for Agile Software Development**, 2001. Disponível em: <http://agilemanifesto.org/>. Acesso: 05 nov. 2022.
- CARVALHO, Elielton da Costa; MALCHER, Paulo R. Campelo; SANTOS, Rodrigo Pereira dos. A Survey Research on the Use of Mobile Applications in Software Project Management. In: SBQS'20: 19TH BRAZILIAN SYMPOSIUM ON SOFTWARE QUALITY, 19., 2020, New York, NY, USA. **19th Brazilian Symposium on Software Quality**. New York, NY, USA: ACM, 2020. p. 1-10.
- CARVALHO, Elielton da Costa; MALCHER, Paulo R. Campelo; SANTOS, Rodrigo Pereira dos. A Survey Research on the Use of Mobile Applications in Software Project Management. In: SBQS'20: 19TH BRAZILIAN SYMPOSIUM ON SOFTWARE QUALITY, São Luís Brazil. **SBQS'20: 19th Brazilian Symposium on Software Quality**. New York, NY, USA: ACM, 2020. Disponível em: <https://doi.org/10.1145/3439961.3439963>. Acesso em: 14 dez. 2022.
- CARVALHO, Marly Monteiro de; PATAH, Leandro Alves; DE SOUZA BIDO, Diógenes. Project management and its effects on project success: Cross-country and cross-industry comparisons. **International Journal of Project Management**, v. 33, n. 7, p. 1509-1522, out. 2015. Disponível em: <https://doi.org/10.1016/j.ijproman.2015.04.004>. Acesso em: 19 dez. 2022.
- CERDEIRAL, Cristina; SANTOS, Gleison. Software project management in high maturity: a systematic literature mapping. **Journal of Systems and Software**, v. 148, p. 56-87, fev. 2019.
- CHADLI, Saad Yasser et al. Software project management tools in global software development: a systematic mapping study. **Springerplus**, [S.l.], v. 5, n. 1, p. 1-38, 24 nov. 2016.
- CHAUDHARI, Ashvini R.; JOSHI, Shashank D.; BHONGADE, Rushikesh S. Analytical Study of Success Rate of I.T Projects Developed using Agile Methodology. In: 2021 SECOND INTERNATIONAL CONFERENCE ON ELECTRONICS AND SUSTAINABLE COMMUNICATION SYSTEMS (ICESC), 2021, Coimbatore, India. **2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)**. [S. l.]: IEEE, 2021. Disponível em: <https://doi.org/10.1109/icesc51422.2021.9533020>. Acesso em: 18 dez. 2022.
- CIRIC, Danijela; LALIC, Bojan; GRACANIN, Danijela. Managing innovation: Are project management methods enemies or allies. **International Journal of Industrial Engineering and Management**, v. 7, n. 1, p. 31, 2016. Disponível em: [http://ijiemjournal.uns.ac.rs/images/journal/volume7/ijiem\\_vol7\\_no1\\_5.pdf](http://ijiemjournal.uns.ac.rs/images/journal/volume7/ijiem_vol7_no1_5.pdf). Acesso em: 18 dez. 2022.
- CODAS, Manuel M. Benitez. Gerência de projetos: uma reflexão histórica. **Revista de Administração de Empresas**, [S.l.], v. 27, n. 1, p. 33-37, mar. 1987.

DARUSSALAM, I. M.; WIDYANI, Y.. GitMonitor: a Software for Monitoring Project Schedules with Git Data. In: 2021 INTERNATIONAL CONFERENCE ON DATA AND SOFTWARE ENGINEERING (ICODSE), 2021, Bandung, Indonesia. **2021 International Conference on Data and Software Engineering (ICoDSE)**. [S. l.]: IEEE, 2021. Disponível em: <https://doi.org/10.1109/icodse53690.2021.9648433>. Acesso em: 19 ago. 2022.

DRURY-GROGAN, Meghann L.; CONBOY, Kieran; ACTON, Tom. Examining decision characteristics & challenges for agile software development. **Journal Of Systems And Software**, [S.l.], v. 131, p. 248-265, set. 2017.

DYBA, Tore; DINGSOYR, Torgeir. Agile Project Management: From Self-Managing Teams to Large-Scale Development. In: 2015 IEEE/ACM 37TH IEEE INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2015, Florence, Italy. **2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE)**. [S. l.]: IEEE, 2015. ISBN 9781479919345. Disponível em: <https://doi.org/10.1109/icse.2015.299>. Acesso em: 19 dez. 2022.

FERNÁNDEZ-DIEGO, M.; MÉNDEZ, E. R.; GUEVARA, F. G. L.; ABRAHÃO, S.; INFRAN, E.. An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review. **IEEE Access**, v. 8, p. 166768-166800, 2020. Disponível em: <https://doi.org/10.1109/access.2020.3021664>. Acesso em: 19 ago. 2022.

FILIPPETTO, Alessandro Souza; LIMA, Robson; BARBOSA, Jorge Luis Victória. A risk prediction model for software project management based on similarity analysis of context histories. **Information and Software Technology**, v. 131, p. 1-18, 19 nov. 2020.

FORSGREN, Nicole; HUMILDE, Jez; KIM, Gene. **Acelerar: A ciência do software enxuto e DevOps**: construindo e dimensionando organizações de tecnologia de alto desempenho. Portland, OR: IT Revolution Press, 2018.

FORTUNE, Joyce *et al.* Looking again at current practice in project management. **International Journal of Managing Projects in Business**, v. 4, n. 4, p. 553-572, 13 set. 2011. Disponível em: <https://doi.org/10.1108/17538371111164010>. Acesso em: 19 dez. 2022.

FRANCO, Eduardo Ferreira; HIRAMA, Kechi; CARVALHO, Marly M.. Applying system dynamics approach in software and information system projects: a mapping study. **Information And Software Technology**, [S.L.], v. 93, p. 58-73, jan. 2018.

GALVAN-CRUZ, Sergio; MORA, Manuel; LAPORTE Claude Y.; DURAN-LIMON, Hector. Reconciliation of scrum and the project management process of the ISO/IEC 29110 standard-Entry profile—an experimental evaluation through usability measures. **Software Quality Journal**, 3 abr. 2021. Disponível em: <https://doi.org/10.1007/s11219-021-09552-3>. Acesso em: 19 dez. 2022.

GARFINKEL, H. **Studios en Etnometodología**. Barcelona: Anthropos, 2006.

GAROUSI, Vahid; TARHAN, Ayça; PFAHL, Dietmar; COŞKUNÇAY, Ahmet; DEMIRÖRS, Onur. Correlation of critical success factors with success of software

projects: an empirical investigation. **Software Quality Journal**, v. 27, n. 1, p. 429-493, 27 out. 2018. Disponível em: <https://doi.org/10.1007/s11219-018-9419-5>. Acesso em: 19 dez. 2022.

GIL, Antônio Carlos. Métodos e técnicas de pesquisa social. 6. ed. São Paulo: **Atlas**, 2008. 216 p.

GOUVEIA JR, Amauri. O conceito de modelo e sua utilização nas ciências do comportamento: breves notas introdutórias. **Estudos de Psicologia (Campinas)**, v. 16, n. 1, p. 13-16, abr. 1999. Disponível em: <https://doi.org/10.1590/s0103-166x1999000100002>. Acesso em: 18 dez. 2022.

HALEEM, Mohd; FAROOQUI, Md Faizan; FAISAL, Md. Tackling Requirements Uncertainty in Software Projects: A Cognitive Approach. **International Journal of Cognitive Computing in Engineering**, v. 2, p. 180-190, jun. 2021. Disponível em: <https://doi.org/10.1016/j.ijcce.2021.10.003>. Acesso em: 19 out. 2022.

HAYAT, Faisal *et al.* The Influence of Agile Methodology (Scrum) on Software Project Management. *In: 2019 20TH IEEE/ACIS INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, ARTIFICIAL INTELLIGENCE, NETWORKING AND PARALLEL/DISTRIBUTED COMPUTING (SNPD)*, 2019, Toyama, Japan. **2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)**. [S. l.]: IEEE, 2019. ISBN 9781728116518. Disponível em: <https://doi.org/10.1109/snpd.2019.8935813>. Acesso em: 19 dez. 2022.

HELDMAN, Kim. **Gerência de projetos**: guia para o exame oficial do PMI. Gulf Professional Publishing, 2006.

HELM, Richard; GAMMA, Erich; VLISSIDES, John; RALPH, Johnson. **Design Patterns Elements of Reusable Object-Oriented Software**. [S. l.]: Pearson Education, 2000. ISBN 9812358757.

HOPPING, C.; HELLARD, B.; HOLLAND, M.. Best Project Management Software. **It Pro London**, Londres, v. 9, n. 1, p. 1-11, set. 2018.

JAHAN, M. S. et al. Software Project Management and Its Tools in Practice in IT Industry of Pakistan. *In: 2019 2ND INTERNATIONAL CONFERENCE ON COMPUTING, MATHEMATICS AND ENGINEERING TECHNOLOGIES (ICOMET)*, 2., 2019, Sukkur. **Proceedings [...]**. Sukkur: IEEE, 2019. p. 1-6.

JAYAWARDENA, Dinusha S.; EKANAYAKE, Lesly L. Adaptation analysis of Agile Project Management for managing IT projects in Sri Lanka. *In: 2010 INTERNATIONAL CONFERENCE ON ADVANCES IN ICT FOR EMERGING REGIONS (ICTER)*, 2010, Colombo, Sri Lanka. **2010 International Conference on Advances in ICT for Emerging Regions (ICTer)**. [S. l.]: IEEE, 2010. ISBN 9781424490417. Disponível em: <https://doi.org/10.1109/ictcr.2010.5643278>. Acesso em: 18 dez. 2022.

JOSLIN, Robert; MÜLLER, Ralf. Relationships between a project management methodology and project success in different project governance

contexts. **International journal of project management**, v. 33, n. 6, p. 1377-1392, 2015.

KARGILI, O. B.; ARIK, A. O.; BEKLER, M.; KOSE, O. U., AKTAS M. S.. A Novel Distributed Software Architecture for Managing Customer Behavior Data: A Case Study in Banking Sector. In: **2021 21ST INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND ITS APPLICATIONS (ICCSA)**, 2021, Cagliari, Italy. 2021 21st International Conference on Computational Science and Its Applications (ICCSA). [S. l.]: IEEE, 2021. Disponível em: <https://doi.org/10.1109/iccsa54496.2021.00037>. Acesso em: 19 ago. 2022.

KEELING, R.; BRANCO, R. H. F.. **Gestão de projetos: uma abordagem global**. 3. ed. São Paulo: Saraiva, 2014.

KERZNER, Harold. **Project Management: a systems approach to planning, scheduling, and controlling**. 11. ed. Hoboken, Nova Jersey, Eua: John Wiley & Sons, 2013. 1264 p.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. **Staffordshire**, UK. 2007.

KOSTALOVA, Jana; TETREVOVA, Libena; SVEDIK, Jan. Support of Project Management Methods by Project Management Information System. *Procedia - Social And Behavioral Sciences*, [S.L.], v. 210, p. 96-104, dez. 2015.

LALIC, B.; CIRIC, D.; SAVKOVIC, M.; RAKIC, S.; MARJANOVIC, U. Exploring the use of game-based learning in agile project management education. In: **2021 19TH INTERNATIONAL CONFERENCE ON EMERGING ELEARNING TECHNOLOGIES AND APPLICATIONS (ICETA)**, 2021, Košice, Slovakia. **2021 19th International Conference on Emerging eLearning Technologies and Applications (ICETA)**. [S. l.]: IEEE, 2021b. Disponível em: <https://doi.org/10.1109/iceta54173.2021.9726599>. Acesso em: 18 dez. 2022.

LAPPE, Marc; SPANG, Konrad. Investments in project management are profitable: A case study-based analysis of the relationship between the costs and benefits of project management. **International Journal of Project Management**, v. 32, n. 4, p. 603-612, 2014.

LARSON, E. W.; GRAY, C. F. **Gerenciamento de projetos: o processo gerencial**. 6. ed. Porto Alegre: AMGH Editora, 2016.

LEINONEN, J.. Evaluating Software Development Effort Estimation Process. In: **Agile Software Development Context**. University of Oulu, 2016.

LIN, Xianchai; WANG, Mei; ZUO, Yajing; LI, Mingge; LIN, Xiaofeng; ZHU, Siping; ZHENG, Yongxin; YU, Minbin; LAMOUREUX, Ecosse L. Health Literacy, Computer Skills and Quality of Patient-Physician Communication in Chinese Patients with Cataract. **PLoS ONE**, v. 9, n. 9, p. e107615, 16 set. 2014. Disponível em: <https://doi.org/10.1371/journal.pone.0107615>. Acesso em: 14 dez. 2022.

LISKIN, O.; SCHNEIDER, K.. Improving Project Communication with Virtual Team Boards. In: **2012 SEVENTH IEEE INTERNATIONAL CONFERENCE ON GLOBAL SOFTWARE ENGINEERING WORKSHOP (ICGSEW)**, 2012, Porto Alegre, Rio Grande do Sul, Brazil. 2012 Seventh IEEE International Conference on Global Software Engineering Workshop (ICGSEW). [S. l.]: IEEE, 2012. Disponível em: <https://doi.org/10.1109/icgsew.2012.10>. Acesso em: 19 ago. 2022.

MAHNIČ, V.; HOVELJA, T.. On using planning poker for estimating user stories. **Journal of Systems and Software**, v. 85, n. 9, p. 2086-2095, set. 2012. Disponível em: <https://doi.org/10.1016/j.jss.2012.04.005>. Acesso em: 19 ago. 2022.

MANIFESTO ÁGIL. Agile manifesto. Haettu, v. 14, p. 2012, 2001.

MARRERO, L.; ASTUDILLO, H.. DevOps-RAF: An assessment framework to measure DevOps readiness in software organizations. In: **2021 40TH INTERNATIONAL CONFERENCE OF THE CHILEAN COMPUTER SCIENCE SOCIETY (SCCC)**, 2021, La Serena, Chile. 2021 40th International Conference of the Chilean Computer Science Society (SCCC). [S. l.]: IEEE, 2021. Disponível em: <https://doi.org/10.1109/sccc54552.2021.9650363>. Acesso em: 19 ago. 2022.

MARTINS, José Carlos Cordeiro. **Gerenciamento de projetos de desenvolvimento de Software com PMI, RUP e UML**. 5. ed. Rio de Janeiro: Brasport, 2010.

MASSO, Jhon et al. Risk management in the software life cycle: a systematic literature review. **Computer Standards & Interfaces**, [S.l.], v. 71, p. 103431, ago. 2020.

MATIAS-PEREIRA, J.. **Manual de metodologia da pesquisa científica**. (2ª ed.). São Paulo: Atlas, 2010.

MENEZES JUNIOR, Júlio; GUSMÃO, Cristine; MOURA, Hermano. Risk factors in software development projects: a systematic literature review. **Software Quality Journal**, v. 27, p. 1149-1174, set. 2019.

MICHAEL, Dennis; DAZKI, Erick; SANTOSO, Handri; INDRAJIT, Richardus Eko. Scrum Team Ownership Maturity Analysis on Achieving Goal. In: **2021 SIXTH INTERNATIONAL CONFERENCE ON INFORMATICS AND COMPUTING (ICIC)**, 2021, Jakarta, Indonesia. **2021 Sixth International Conference on Informatics and Computing (ICIC)**. [S. l.]: IEEE, 2021. Disponível em: <https://doi.org/10.1109/icic54025.2021.9632969>. Acesso em: 15 dez. 2022.

MIR, Farzana Asad; PINNINGTON, Ashly H.. Exploring the value of project management: linking project management performance and project success. **International Journal Of Project Management**, [S.L.], v. 32, n. 2, p. 202-217, fev. 2014.

MOLØKKEN-ØSTVOLD, K.; HAUGEN, N. C.; BENESTAD, H. C.. Using planning poker for combining expert estimates in software projects. **Journal of Systems and Software**, v. 81, n. 12, p. 2106-2117, dez. 2008. Disponível em: <https://doi.org/10.1016/j.jss.2008.03.058>. Acesso em: 19 ago. 2022.

MONTGOMERY, L.; LÜDERS, C.; MAALEJ, W.. An Alternative Issue Tracking Dataset of Public Jira Repositories. In: **2022 IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)**. IEEE, 2022. p. 73-77.

MORANDINI, Marcelo; COLETI, Thiago Adriano; OLIVEIRA, Edson; CORRÊA, Pedro Luiz Pizzigatti. Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams. **Computer Science Review**, v. 39, p. 100314, fev. 2021. Disponível em: <https://doi.org/10.1016/j.cosrev.2020.100314>. Acesso em: 19 dez. 2022.

PARSI, Novid. The Next Agile Awakening: four agile leaders discuss new possibilities in a world of sudden change. *PM Network*, Filadélfia, Pensilvânia, v. 3, n. 35, p. 36-43, 1 maio 2021. Disponível em: <https://www.pmi.org/learning/library/agile-leaders-discuss-possibilities-post-pandemic-13077>. Acesso em: 2 jun. 2021.

PINTO, M. R.; SANTOS, L. L. S.. A Grounded Theory como abordagem metodológica: relatos de uma experiência de campo. **Organizações & Sociedade**, v. 19, p. 417-436, 2012.

PMBOK. **Um guia do conhecimento em gerenciamento de projetos**. 6. ed. Newtown Square: PMI, 2017. 763 p.

PRESSMAN, R. **Engenharia de software**. São Paulo: McGraw-Hill, 2006.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. Porto Alegre: AMGH, 2015.

PRODANOV, C. C.; FREITAS, E. C.. Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico. 2. ed. Novo Hamburgo: **Feevale**, 2013. E-book (276).

RADUJKOVIĆ, Mladen; SJEKAVICA, Mariela. Project Management Success Factors. **Procedia Engineering**, v. 196, p. 607-615, 2017. Disponível em: <https://doi.org/10.1016/j.proeng.2017.08.048>. Acesso em: 19 dez. 2022.

RINCON-GONZALEZ, C.; DIAZ-PIRAQUIVE, F. N.; DIEZ-SILVA, M.. Bibliometric and impact analysis of the Project Management Offices in Colombia. In: **2019 CONGRESO INTERNACIONAL DE INNOVACIÓN Y TENDENCIAS EN INGENIERIA (CONIITI)**, 2019, BOGOTA, Colombia. 2019 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI). [S. l.]: IEEE, 2019. Disponível em: <https://doi.org/10.1109/coniiti48476.2019.8960626>. Acesso em: 19 ago. 2022.

RÍOS, J. C. C.; EMBURY, S. M.; ERASLAN, S.. A unifying framework for the systematic analysis of Git workflows. *Information and Software Technology*, v. 145, p. 106811, maio 2022. Disponível em: <https://doi.org/10.1016/j.infsof.2021.106811>. Acesso em: 19 ago. 2022.

SALEEM, Nazish; MATHRANI, Sanjay; SAJJAD, Aymen. Exploring the Evolutionary Characteristics of Project Management Approaches at Different Levels of Operations. In: **2021 IEEE ASIA-PACIFIC CONFERENCE ON COMPUTER SCIENCE AND DATA ENGINEERING (CSDE)**, 2021, Brisbane, Australia. 2021

IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). [S. l.]: IEEE, 2021. Disponível em: <https://doi.org/10.1109/csde53843.2021.9718471>. Acesso em: 19 out. 2022.

SCHOTS, N. C. L.. UMA ABORDAGEM PARA A IDENTIFICAÇÃO DE CAUSAS DE PROBLEMAS UTILIZANDO GROUNDED THEORY. **Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia de Sistemas e Computação**, Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa em Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010, 206 f.. Disponível em: <https://www.cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/2134>. Acesso em: 18 ago. 2022.

SHAIKH, Tabish Hanif; KHAN, Faisal Latif; SHAIKH, NoorAli Alimuddin; SHAH, Haidarali Nadir; PIRANI, Zainab. Survey of Web-Based Project Management System. In: 2018 INTERNATIONAL CONFERENCE ON SMART SYSTEMS AND INVENTIVE TECHNOLOGY (ICSSIT), 2018, Tirunelveli, India. **2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)**. [S. l.]: IEEE, 2018. ISBN 9781538658734. Disponível em: <https://doi.org/10.1109/icssit.2018.8748784>. Acesso em: 15 dez. 2022.

SHASTRI, Yogeshwar; HODA, Rashina; AMOR, Robert. The role of the project manager in agile software development projects. **Journal Of Systems And Software**, [S.l.], v. 173, p. 110871, mar. 2021.

SHRIVASTAVA, Suprika Vasudeva; RATHOD, Urvashi. A risk management framework for distributed agile projects. **Information And Software Technology**, [S.L.], v. 85, p. 1-15, maio 2017.

SILVA, L. S. F.; OLIVEIRA, S. R. B.. A Process Framework with Agile Practices for Implementation of Project Portfolio Management Process. In: **2016 10TH INTERNATIONAL CONFERENCE ON THE QUALITY OF INFORMATION AND COMMUNICATIONS TECHNOLOGY (QUATIC)**, 2016, Lisbon. 2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC). [S. l.]: IEEE, 2016. Disponível em: <https://doi.org/10.1109/quatic.2016.037>. Acesso em: 19 ago. 2022.

SINHA, Abhiup; DAS, Pallabi. Agile Methodology Vs. Traditional Waterfall SDLC: A case study on Quality Assurance process in Software Industry. In: 2021 5TH INTERNATIONAL CONFERENCE ON ELECTRONICS, MATERIALS ENGINEERING & NANO-TECHNOLOGY (IEMENTECH), 2021, Kolkata, India. **2021 5th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)**. [S. l.]: IEEE, 2021. Disponível em: <https://doi.org/10.1109/iementech53263.2021.9614779>. Acesso em: 18 dez. 2022.

SMEDS, J.; NYBOM, K.; PORRES, I.. DevOps: a definition and perceived adoption impediments. In: **International conference on agile software development**. Springer, Cham, 2015. p. 166-177.

SMITH, E.; LOFTIN, R.; MURPHY-HILL, E.; BIRD, C.; ZIMMERMANN, T.. Improving developer participation rates in surveys. In: **2013 6TH INTERNATIONAL WORKSHOP ON COOPERATIVE AND HUMAN ASPECTS OF SOFTWARE**

**ENGINEERING (CHASE)**, 2013, San Francisco, CA, USA. 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 2013.

SOARES, E. B.. Trajetórias de acadêmicas no Brasil e em Portugal: um olhar sobre o gênero a partir de uma Grounded Theory. 2019. **Tese de Doutorado**.

SOUZA, Mauricio R. de A., VEADO, Lucas, MOREIRA, Renata Teles, FIGUEIREDO, Eduardo, COSTA, Heitor. A systematic mapping study on game-related methods for software engineering education. **Information and Software Technology**, v. 95, p. 201-218, mar. 2018. Disponível em: <https://doi.org/10.1016/j.infsof.2017.09.014>. Acesso em: 13 dez. 2022.

SQME. Take Organizational Performance To The Next Level With CMMI V2.0. 2019. Disponível em: <https://sqme.com/cmmi-blog-01/#:~:text=Take%20Organizational%20Performance%20to%20the,satisfaction%20and%20other%20key%20drivers>. Acesso em: 19 ago. 2022.

STRAUSS, A.; CORBIN, J.. **Basics of Qualitative Research: techniques and procedures for developing grounded theory**. 3. ed. Nova York: Sage Publications, 2007. 400 p.

SUDARMANINGTYAS, P.; MOHAMED, R. B.. Extended Planning Poker: A Proposed Model. In: **2020 7TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY, COMPUTER, AND ELECTRICAL ENGINEERING (ICITACEE)**, 2020, Semarang, Indonesia. 2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). [S. l.]: IEEE, 2020. Disponível em: <https://doi.org/10.1109/icitacee50144.2020.9239165>. Acesso em: 19 ago. 2022.

TAM, Carlos *et al.* The factors influencing the success of on-going agile software development projects. **International Journal of Project Management**, v. 38, n. 3, p. 165-176, abr. 2020. Disponível em: <https://doi.org/10.1016/j.ijproman.2020.02.001>. Acesso em: 19 dez. 2022.

TAUFIQ, A.; RAHARJO, T.; WAHBI, A.. Scrum Evaluation to Increase Software Development Project Success: A Case Study of Digital Banking Company. In: **2020 INTERNATIONAL CONFERENCE ON ADVANCED COMPUTER SCIENCE AND INFORMATION SYSTEMS (ICACISIS)**, 2020, Depok, Indonesia. 2020 International Conference on Advanced Computer Science and Information Systems (ICACISIS). [S. l.]: IEEE, 2020. Disponível em: <https://doi.org/10.1109/icacsis51025.2020.9263235>. Acesso em: 19 ago. 2022.

USPENSKIY, Mikhail B.; SMIRNOV, Sergey V.; LOGINOVA, Aleksandra V.; SHIROKOVA, Svetlana V.. Modelling of Complex Project Management System in the Field of Information Technologies. In: **2019 III INTERNATIONAL CONFERENCE ON CONTROL IN TECHNICAL SYSTEMS (CTS)**, 2019, St. Petersburg, Russia. **2019 III International Conference on Control in Technical Systems (CTS)**. [S. l.]: IEEE, 2019. ISBN 9781728125107. Disponível em: <https://doi.org/10.1109/cts48763.2019.8973245>. Acesso em: 15 dez. 2022.

VARAJÃO, João; COLOMO-PALACIOS, Ricardo; SILVA, Hélio. ISO 21500: 2012 and pmbok 5 processes in information systems project management. *Computer Standards & Interfaces*, [S.L.], v. 50, p. 216-222, fev. 2017.

VARGAS, R. V.. **Gerenciamento de projetos**: estabelecendo diferenciais competitivos. 9. ed. Rio de Janeiro: Brasport, 2018.

VENDRUSCOLO, Bruno; HOFFMANN, Valmir Emil; FREITAS, Carlos Alberto. A INFLUÊNCIA DE FERRAMENTAS DE GESTÃO ESTRATÉGICA E DE STAKEHOLDERS NO DESEMPENHO DE ORGANIZAÇÕES DO RAMO PARTICULAR DE SAÚDE DO DISTRITO FEDERAL. **Revista Ibero-Americana de Estratégia**, v. 11, n. 1, p. 30-61, 1 abr. 2012. Disponível em: <https://doi.org/10.5585/riae.v11i1.1799>. Acesso em: 18 dez. 2022.

VERSIONONE, CollabNet. 15th annual state of agile report. collab. net, 2021.

WARD, John; DANIEL, Elizabeth. **Benefits Management**: delivering value from is and it investments. Chichester, West Sussex, Uk: John Wiley & Sons, 2006. 424 p.

WHITTON, Nicola; LANGAN, Mark. Fun and games in higher education: an analysis of UK student perspectives. **Teaching in Higher Education**, v. 24, n. 8, p. 1000-1013, 8 nov. 2018. Disponível em: <https://doi.org/10.1080/13562517.2018.1541885>. Acesso em: 18 dez. 2022.

YANOW, Stephanie K.; GOOD, Michael F. Nonessential Research in the New Normal: The Impact of COVID-19. **The American Journal of Tropical Medicine and Hygiene**, v. 102, n. 6, p. 1164-1165, 3 jun. 2020. Disponível em: <https://doi.org/10.4269/ajtmh.20-0325>. Acesso em: 14 dez. 2022.

ZABIL, Mohamed; MAHDI, Mohamed Najah; AZMI, Muhammad Sufyian Mohd; CHENG, Lim Kok; YUSOF, Azlan; AHMAD, Abdul Rahim. Software Project Management Using Machine Learning Technique - A Review. *In: 2020 8TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY AND MULTIMEDIA (ICIMU)*, 2020, Selangor, Malaysia. **2020 8th International Conference on Information Technology and Multimedia (ICIMU)**. [S. l.]: IEEE, 2020. ISBN 9781728173108. Disponível em: <https://doi.org/10.1109/icimu49871.2020.9243543>. Acesso em: 19 dez. 2022.

ZHAI, Li; XIN, Yanfei; CHENG, Chaosheng. Understanding the value of project management from a stakeholder's perspective: Case study of mega-project management. **Project Management Journal**, v. 40, n. 1, p. 99-109, 2009.

ZHANG, H.; HUANG, H.; SHAO, D.; HUANG, X.. Fireteam: a small-team development practice in industry. *In: ESEC/FSE '20: 28TH ACM JOINT EUROPEAN SOFTWARE ENGINEERING CONFERENCE AND SYMPOSIUM ON THE FOUNDATIONS OF SOFTWARE ENGINEERING*, Virtual Event USA. ESEC/FSE '20: 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. New York, NY, USA: ACM, 2020. Disponível em: <https://doi.org/10.1145/3368089.3417068>. Acesso em: 19 ago. 2022.

# APÊNDICES

## Apêndice 1

### Corpus of Literature: Approaches to Support Software Project Management in The Agile Context

- [EP01] Alhazmi, A., Huang, S. (2018). A Decision Support System for Sprint Planning in Scrum Practice. In *SoutheastCon 2018*. IEEE. <https://doi.org/10.1109/secon.2018.8479063>
- [EP02] Angioni, M., Carboni, D., Melis, M., Pinna, S., Sanna, R., Soro, A. (2004). XPSuite: tracking and managing XP projects in the IDE. In *the 2004 workshop*. ACM Press. <https://doi.org/10.1145/1151433.1151440>
- [EP03] Baia, D. d. M. (2015). An Integrated Multi-Agent-Based Simulation Approach to Support Software Project Management. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE)*. IEEE. <https://doi.org/10.1109/icse.2015.290>
- [EP04] Baptista, J. (2008). Agile documentation with uScrum. In *the 26th annual ACM international conference*. ACM Press. <https://doi.org/10.1145/1456536.1456596>
- [EP05] Bastarrica, M. C., Perovich, D., Marín, J., Rioseco, L. (2017). Process-based project management and SPI. In *ICSSP 2017: International Conference on the Software and Systems Process 2017*. ACM. <https://doi.org/10.1145/3084100.3084115>
- [EP06] Beale, M. (2016). Designing an Agile Game for Technical Communication Classrooms. In *SIGDOC '16: The 34th ACM International Conference on the Design of Communication*. ACM. <https://doi.org/10.1145/2987592.2987615>
- [EP07] Begosso, L. R., Franco, L. H. B., da Cunha, D. S., Begosso, L. C. (2019). SimScrumF: a game for supporting the process of teaching Scrum. In *the 9th International Conference*. ACM Press. <https://doi.org/10.1145/357419.3357426>
- [EP08] Biehl, J. T., Czerwinski, M., Smith, G., Robertson, G. G. (2007). FASTDash: A Visual Dashboard for Fostering Awareness in Software Teams. In *the SIGCHI Conference*. ACM Press. <https://doi.org/10.1145/1240624.1240823>
- [EP09] Bierwolf, R., Frijns, P., van Kemenade, P. (2017). Project management in a dynamic environment: Balancing stakeholders. In *2017 IEEE European Technology and Engineering Management Summit (E-TEMS)*. IEEE. <https://doi.org/10.1109/etems.2017.8244226>
- [EP10] Borrego, G., Salazar-Lugo, G., Parra, M., Palacio, R. (2019). Slack's Knowledge Classification Mechanism for Architectural Knowledge Condensation. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. <https://doi.org/10.1109/csci49370.2019.00212>
- [EP11] Bruegge, B., David, J., Helming, J., Koegel, M. (2009). Classification of tasks using machine learning. In *the 5th International Conference*. ACM Press. <https://doi.org/10.1145/1540438.1540455>
- [EP12] Carneiro, L. B., Silva, A. C. C. L. M., Alencar, L. H. (2018). Scrum Agile Project Management Methodology Application for Workflow Management: A Case Study. In *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE. <https://doi.org/10.1109/ieem.2018.8607356>
- [EP13] Castillo-Barrera, F. E., Amador-Garcia, M., Perez-Gonzalez, H. G., Martinez-Perez, F. E., Torres-

- Reyes, F. J. (2018). Adapting Bloom's Taxonomy for an Agile Classification of the Complexity of the User Stories in SCRUM. In *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*. IEEE. <https://doi.org/10.1109/conisoft.2018.8645899>
- [EP14] Chasanidou, D., Elvesæter, B., Berre, A.-J. (2016). Enabling team collaboration with task management tools. In *OpenSym '16: The International Symposium on Open Collaboration*. ACM. <https://doi.org/10.1145/2957792.2957799>
- [EP15] Desharnais, J.-M., Buglione, L., Kocatürk, B. (2011). Using the COSMIC method to estimate Agile user stories. In *the 12th International Conference*. ACM Press. <https://doi.org/10.1145/2181101.2181117>
- [EP16] Dowling, P., McGrath, K. (2015). Using Free and Open Source Tools to Manage Software Quality: An agile process implementation. *Queue*, 13(4), 20–27. <https://doi.org/10.1145/2756506.2767182>
- [EP17] Farid, W. M., Mitropoulos, F. J. (2013). NORPLAN: Non-functional Requirements Planning for agile processes. In *IEEE SOUTHEASTCON 2013*. IEEE. <https://doi.org/10.1109/secon.2013.6567463>
- [EP18] Fehlmann, T. M., Kranich, E. (2017). A new approach for continuously monitoring project deadlines in software development. In *IWSM/Mensura '17: 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement*. ACM. <https://doi.org/10.1145/3143434.3143439>
- [EP19] Silva, L. S. F., Oliveira, S. R. B. (2016). A Process Framework with Agile Practices for Implementation of Project Portfolio Management Process. In *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)*. IEEE. <https://doi.org/10.1109/quatic.2016.037>
- [EP20] Gama, K. (2019). An Experience Report on Using LEGO-based Activities in a Software Engineering Course. In *SBES 2019: XXXIII Brazilian Symposium on Software Engineering*. ACM. <https://doi.org/10.1145/3350768.3353817>
- [EP21] Garcia, C., Guerrero, A., Zeitsoff, J., Korlakunta, S., Fernandez, P., Fox, A., Ruiz-Cortes, A. (2021). Bluejay: A Cross-Tooling Audit Framework For Agile Software Teams. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. IEEE. <https://doi.org/10.1109/icse-seet52601.2021.00038>
- [EP22] Godoy, C. P., Cruz, A. F., Silva, E. P., Santos, L. M., Zerbini, R. S., Pahins, C. A. L. (2019). Blueprint Model: A new Approach to Scrum Agile Methodology. In *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*. IEEE. <https://doi.org/10.1109/icgse.2019.00014>
- [EP23] Gruhn, V., von Brisinski, N. S. (2020). How to reduce risk effectively in fixed price software development. In *ICSE '20: 42nd International Conference on Software Engineering*. ACM. <https://doi.org/10.1145/3377813.3381361>
- [EP24] Guerrero, A., Fresno, R., Ju, A., Fox, A., Fernandez, P., Muller, C., Ruiz-Cortés, A. (2019). Eagle: a team practices audit framework for agile software development. In *ESEC/FSE '19: 27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM. <https://doi.org/10.1145/3338906.3341181>
- [EP25] Hanakawa, N., Okura, K. (n.d.). A Project Management Support Tool using Communication for Agile Software Development. In *11th Asia-Pacific Software Engineering Conference*. IEEE. <https://doi.org/10.1109/apsec.2004.8>

- [EP26] Haugen, N. C. (2006). An Empirical Study of Using Planning Poker for User Story Estimation. In *AGILE 2006 (AGILE'06)*. IEEE. <https://doi.org/10.1109/agile.2006.16>
- [EP27] Hayat, F., Anwar, M. W., Azam, F., Kiran, A. (2019). A SYSML-Based Approach for Requirements Risk Management and Change Control. In *ICIME 2019: 2019 11th International Conference on Information Management and Engineering*. ACM. <https://doi.org/10.1145/3373744.3373751>
- [EP28] Hayat, F., Rehman, A. U., Arif, K. S., Wahab, K., Abbas, M. (2019). The Influence of Agile Methodology (Scrum) on Software Project Management. In *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE. <https://doi.org/10.1109/npd.2019.8935813>
- [EP29] Helming, J., Koegel, M., Naughton, H. (2009). Towards traceability from project management to system models. In *2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE)*. IEEE. <https://doi.org/10.1109/tfse.2009.5069576>
- [EP30] Hildenbrand, T., Geisser, M., Kude, T., Bruch, D., Acker, T. (2008). Agile Methodologies for Distributed Collaborative Development of Enterprise Applications. In *2008 International Conference on Complex, Intelligent and Software Intensive Systems*. IEEE. <https://doi.org/10.1109/cisis.2008.105>
- [EP31] Houston, D. X., Buettner, D. J. (2013). Modeling user story completion of an agile software process. In *the 2013 International Conference*. ACM Press. <https://doi.org/10.1145/2486046.2486063>
- [EP32] Jain, R., Suman, U. (2018). A Project Management Framework for Global Software Development. *ACM SIGSOFT Software Engineering Notes*, 43 (1), 1–10. <https://doi.org/10.1145/3178315.3178329>
- [EP33] Janjua, J. I., Ali, A., Chaudhry, M. U. (2016). OPENTCQ: Towards Change Management in Hybrid Agile Model. In *ICCCNT '16: 7th International Conference on Computing Communication and Networking Technologies*. ACM. <https://doi.org/10.1145/2967878.2967920>
- [EP34] Tang, J. (2008). An adaptive model of health diagnosis for agile software development. In *2008 International Conference on Machine Learning and Cybernetics (ICMLC)*. IEEE. <https://doi.org/10.1109/icmlc.2008.4620486>
- [EP35] Kroll, J., Friboim, S., Hemmati, H. (2017). An Empirical Study of Search-Based Task Scheduling in Global Software Development. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. IEEE. <https://doi.org/10.1109/icse-seip.2017.30>
- [EP36] Krusche, S., Alperowitz, L., Bruegge, B., Wagner, M. O. (2014). Rugby: an agile process model based on continuous delivery. In *the 1st International Workshop*. ACM Press. <https://doi.org/10.1145/2593812.2593818>
- [EP37] Lin, J., Yu, H., Shen, Z., Miao, C. (2014). Studying task allocation decisions of novice agile teams with data from agile project management tools. In *ASE '14: ACM/IEEE International Conference on Automated Software Engineering*. ACM. <https://doi.org/10.1145/2642937.2642959>
- [EP38] Liskin, O., Schneider, K. (2012). Improving Project Communication with Virtual Team Boards. In *2012 Seventh IEEE International Conference on Global Software Engineering Workshop (ICGSEW)*. IEEE. <https://doi.org/10.1109/icgsew.2012.10>
- [EP39] Lyra, K. T., Alves, M. L., Silva, F. H. C., Souza, K., Isotani, S. (2018). An agile project management experience: points of view of

- graduate students. In *the XXXII Brazilian Symposium*. ACM Press. <https://doi.org/10.1145/3266237.3266248>
- [EP40] Márquez, R., Vizcaíno, A., García, F. O., Manjavacas, A. (2020). GLOBAL-MANAGER: a serious game for providing training in project manager skills. In *ICGSE '20: 15th IEEE/ACM International Conference on Global Software Engineering*. ACM. <https://doi.org/10.1145/3372787.3390443>
- [EP41] Masticola, S. P. (2007). Lightweight Risk Mitigation for Software Development Projects Using Repository Mining. In *Fourth International Workshop on Mining Software Repositories*. IEEE. <https://doi.org/10.1109/msr.2007.16>
- [EP42] Mateescu, M., Kropp, M., Burkhard, R., Zahn, C., Vischi, D. (2015). aWall: A Socio-Cognitive Tool for Agile Team Collaboration using Large Multi-Touch Wall Systems. In *the 2015 International Conference*. ACM Press. <https://doi.org/10.1145/2817721.2835072>
- [EP43] Molléri, J. S., Gonzalez-Huerta, J., Henningsson, K. (2018). A Legacy Game for Project Management in Software Engineering Courses. In *ECSEE'18: European Conference of Software Engineering Education 2018*. ACM. <https://doi.org/10.1145/3209087.3209094>
- [EP44] Morgan, R., Maurer, F. (2006). MasePlanner: A Card-Based Distributed Planning Tool for Agile Teams. In *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*. IEEE. <https://doi.org/10.1109/icgse.2006.261226>
- [EP45] Mounir, M., Salah, A., Kamel, A., Moussa, H. (2020). Framework to Measure Agile Software Process Effectiveness in Critical Systems Development. In *ICSIE 2020: 2020 9th International Conference on Software and Information Engineering*. ACM. <https://doi.org/10.1145/3436829.3436853>
- [EP46] Nyfjord, J., Kajko-Mattsson, M. (2008). Outlining a Model Integrating Risk Management and Agile Software Development. In *2008 34th Euromicro Conference Software Engineering and Advanced Applications (SEAA)*. IEEE. <https://doi.org/10.1109/saa.2008.77>
- [EP47] Paasivaara, M., Lassenius, C. (2011). Scaling Scrum in a Large Distributed Project. In *2011 5th International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE. <https://doi.org/10.1109/sem.2011.49>
- [EP48] Perkusich, M., de Almeida, H. O., Perkusich, A. (2013). A model to detect problems on scrum-based software development projects. In *the 28th Annual ACM Symposium*. ACM Press. <https://doi.org/10.1145/2480362.2480560>
- [EP49] Petersen, R. R., Wiil, U. K. (2008). Asap: a planning tool for agile software development. In *the nineteenth ACM conference*. ACM Press. <https://doi.org/10.1145/1379092.1379101>
- [EP50] Picha, P., Brada, P. (2019). Software process anti-pattern detection in project data. In *the 24th European Conference*. ACM Press. <https://doi.org/10.1145/361149.3361169>
- [EP51] Posadas, J. V. (2017). Application of mixed distributed software architectures for social-productive projects management in peru. In *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*. IEEE. <https://doi.org/10.1109/intercon.2017.8079698>
- [EP52] Raith, F., Richter, I., Lindermeier, R. (2017). How Project-management-tools are used in Agile Practice: Benefits, Drawbacks and Potentials. In *the 21st International Database Engineering & Applications Symposium*. ACM Press. <https://doi.org/10.1145/3105831.3105865>
- [EP53] Rajeev, B. V., Hejib, V. (2018). Control based management to self organizing agile teams: a case study. In *ICSE '18: 40th International Conference on*

- Software Engineering*. ACM. <https://doi.org/10.1145/3196369.3196394>
- [EP54] Romano, B. L., Silva, A. D. (2015). Project Management Using the Scrum Agile Method: A Case Study within a Small Enterprise. In *2015 12th International Conference on Information Technology - New Generations (ITNG)*. IEEE. <https://doi.org/10.1109/itng.2015.139>
- [EP55] Sajja, P. R. P., Chaudhary, B. D. (2010). A tool for software development driven by customer interaction. In *the 3rd India software engineering conference*. ACM Press. <https://doi.org/10.1145/1730874.1730887>
- [EP56] Salmanoglu, M., Hacaloglu, T., Demirors, O. (2017). Effort estimation for agile software development: comparative case studies using COSMIC functional size measurement and story points. In *IWSM/Mensura '17: 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement*. ACM. <https://doi.org/10.1145/3143434.3143450>
- [EP57] Schreiber, M., Kraft, B., Zundorf, A. (2017). Metrics Driven Research Collaboration: Focusing on Common Project Goals Continuously. In *2017 IEEE/ACM 4th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)*. IEEE. <https://doi.org/10.1109/ser-ip.2017..6>
- [EP58] Shafiq, S., Mashkooor, A., Mayr-Dorn, C., Egyed, A. (2021). TaskAllocator: A Recommendation Approach for Role-based Tasks Allocation in Agile Software Development. In *2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE)*. IEEE. <https://doi.org/10.1109/icssp-icgse52873.2021.00014>
- [EP59] Soares, F. S. F., Meira, S. R. L. (2015). An agile strategy for implementing CMMI project management practices in software organizations. In *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE. <https://doi.org/10.1109/cisti.2015.7170402>
- [EP60] Stapel, K., Knauss, E., Schneider, K., Zazworka, N. (2011). FLOW Mapping: Planning and Managing Communication in Distributed Teams. In *2011 6th IEEE International Conference on Global Software Engineering (ICGSE)*. IEEE. <https://doi.org/10.1109/icgse.2011.9>
- [EP61] Trapa, V., Rao, S. (2006). T3 - Tool for Monitoring Agile Development. In *AGILE 2006 (AGILE'06)*. IEEE. <https://doi.org/10.1109/agile.2006.49>
- [EP62] Vähäniitty, J., Rautiainen, K. T. (2008). Towards a conceptual framework and tool support for linking long-term product and business planning with agile software development. In *the 1st international workshop*. ACM Press. <https://doi.org/10.1145/1370720.1370730>
- [EP63] Vivian, R., Tarmazdi, H., Falkner, K., Falkner, N., Szabo, C. (2015). The Development of a Dashboard Tool for Visualising Online Teamwork Discussions. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE)*. IEEE. <https://doi.org/10.1109/icse.2015.170>
- [EP64] Zhang, C., Tong, S., Mo, W., Zhou, Y., Xia, Y., Shen, B. (2016). ESSE: An Early Software Size Estimation Method Based on Auto-Extracted Requirements Features. In *Internetware '16: The Eighth Asia-Pacific Symposium on Internetware*. ACM. <https://doi.org/10.1145/2993717.2993733>
- [EP65] Zhang, H., Huang, H., Shao, D., Huang, X. (2020). Fireteam: a small-team development practice in industry. In *ESEC/FSE '20: 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM. <https://doi.org/10.1145/3368089.3417068>



**Apêndice 2****PROTOCOLO DE MAPEAMENTO****Abordagens de gerenciamento de projetos utilizadas na indústria de desenvolvimento ágil de software: um mapeamento sistemático da literatura****Documento: Protocolo de Mapeamento Sistemático da Literatura****Histórico de Revisões**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
24/04/2021	0.1	Início da elaboração do protocolo	Elielton Carvalho
29/04/2021	0.2	Revisão	Sandro Bezerra

13/05/2021	0.3	Ajustes após revisão	Elielton Carvalho
------------	-----	----------------------	----------------------

## 1.CONTEXTO

A história da gerência de projetos remete à antiguidade e continua útil até os dias atuais, pois desde a época da construção das pirâmides egípcias até mais recentemente com a colocação da Estação Espacial Internacional em órbita, a ideia de projetos é utilizada (PMBOK, 2017). Apesar do longo período de utilização da gerência de projetos, foi apenas na década de 1960 que ela passou a ser encarada como uma disciplina, principalmente por ter sido largamente implementada na indústria americana de armas e de tecnologia espacial (MARTINS, 2010).

Com os desafios impostos pela chegada do século XXI e, conseqüentemente, com o avanço tecnológico, os conceitos e implementações da gerência de projetos alcançaram outro patamar. Nesse mesmo período, as organizações começaram a perceber que a concorrência aumenta exponencialmente com o passar do tempo e que seus projetos precisam ser bem gerenciados. Gerenciar o que está sendo desenvolvido passou a ser uma necessidade e não mais um diferencial, haja vista que a concorrência obriga as empresas a se reinventarem para que possam “sobreviver” no mercado capitalista (KERZNER, 2017).

Assim como as demais áreas, o desenvolvimento de software também se beneficia do gerenciamento de projetos, pois cada dia mais cresce a demanda por produtos de software que tenham qualidade, mas que sejam desenvolvidos no menor prazo de tempo possível e sem extrapolar o orçamento pré-estabelecido. Dessa forma, a indústria de software lida constantemente com pressões para que adotem em seus trabalhos o desenvolvimento ágil de software, afim de que respondam as expectativas dos clientes com relação às entregas mais frequentes e às mudanças constantes dos requisitos (ISLAM; STORER, 2020).

Por conta dessa alta demanda, Pressman e Maxin (2016) afirmam que as indústrias de desenvolvimento de software foram transformadas em grandes laboratórios de projetos. Para Andrade e Tait (2012), a gerência de projetos busca garantir a qualidade

do software que está sendo desenvolvido sem comprometer negativamente a empresa responsável por esse desenvolvimento. Aliado à gerência de projetos, a utilização dos métodos ágeis proporciona às empresas uma maior flexibilidade e adaptação à dinâmica atual do mercado consumidor, uma vez que esses métodos permitem a customização dos serviços e produtos que são desenvolvidos (MAS; MESQUIDA; PACHECO, 2020).

O Corpo de Conhecimento em Gerenciamento de Projetos (*Project Management Body of Knowledge - PMBOK*) (2017) é enfático ao afirmar que o gerenciamento de projetos utiliza ferramentas para auxiliar nas atividades que são desenvolvidas dentro de uma organização. Essas ferramentas, porém, não se limitam apenas a um tipo, podendo ser um software, uma metodologia, um padrão, dentre outros tipos de abordagens. Nesse sentido, este Mapeamento Sistemático da Literatura (MSL) busca identificar as abordagens de gerenciamento de projetos que são utilizadas no desenvolvimento ágil de software. Pois apesar de existirem muitas abordagens conhecidas pelos profissionais e pesquisadores da área, ainda existem lacunas a serem preenchidas, principalmente na identificação de outras ferramentas que podem estar sendo utilizadas com êxito, mas que acabam passando despercebida por não haver um estudo que condense e centralize essas informações.

Portanto, este MSL é parte de um projeto de dissertação de mestrado vinculado ao Projeto SPIDER com o objetivo de identificar quais as abordagens de gerenciamento de projetos que constam relatadas na literatura especializada e que são utilizadas no contexto de desenvolvimento ágil de software. Espera-se que, a partir dos resultados deste MSL seja possível o desenvolvimento de um catálogo que reúna todas as abordagens que forem identificadas.

## 2.OBJETIVOS

Este estudo tem como objetivo identificar as abordagens utilizadas no gerenciamento de projetos no contexto de desenvolvimento ágil de software e que constam relatadas na literatura especializada por meio de um MSL, mediante três etapas: planejamento, execução e análise dos dados. Para formalizar o objetivo deste estudo, foi utilizado o *Goal-Question-Metric* (GQM) definido por Basili (1992). Assim, este estudo busca:

**Analisar:** estudos primários, por meio de um MSL.

**Com o propósito de:** identificar as abordagens utilizadas no gerenciamento de projetos de software (GPS) que constam relatadas na literatura especializada.

**Com relação:** a definição, uso e avaliação de abordagens, no contexto de gerenciamento de projetos, utilizadas no desenvolvimento de software.

**Do ponto de vista de:** pesquisadores, organizações e profissionais da área de gerenciamento de projetos de software.

**No contexto:** industrial e acadêmico de desenvolvimento ágil software.

### 3. QUESTÕES DE PESQUISA

As questões de pesquisa deste estudo foram definidas seguindo a estrutura proposta por Kitchenham e Charters (2007), denominada PICOC (*Population, Intervention, Comparison, Outcomes e Context*). No entanto, este estudo busca apenas identificar abordagens de GPS relatadas na literatura e não se preocupa em realizar comparações entre elas. Diante disso, o critério de “Comparação” não será utilizado. A estrutura deste trabalho está descrita no Quadro 1.

Quadro 17 - Estrutura PICOC definida para o trabalho

<b>População</b>	Gerência de Projetos.
<b>Intervenção</b>	Identificar, listar e descrever as abordagens de GPS.
<b>Comparação</b>	Não se aplica.
<b>Resultados</b>	Abordagens utilizadas no gerenciamento de projetos de software, considerando ferramentas, métodos, técnicas, modelos, tecnologias, práticas, padrões, guias, artefatos, metodologias, frameworks, processos, princípios, temas e papéis.
<b>Contexto</b>	Industrial e acadêmico de desenvolvimento ágil de software.

Fonte: O autor (2021)

Assim, foi então definida a questão de pesquisa principal (QP) deste MSL: “**Quais são as abordagens de gerenciamento de projetos que constam relatadas na literatura especializada e que são utilizadas no contexto de desenvolvimento ágil de software?**”. Para auxiliar na resposta da QP, subquestões (Sub-Q) foram definidas e detalhadas no Quadro 2.

Quadro 18 - Subquestões de pesquisa

<b>Sub-Q1</b>	Como foi a evolução do número de estudos publicados relacionados com o tópico desta pesquisa?
<b>Sub-Q2</b>	Quais os países que mais publicaram trabalhos relacionados com o tópico

	desta pesquisa?
<b>Sub-Q3</b>	Qual o tipo de instituição que mais publicou trabalhos relacionados com o tópico desta pesquisa?
<b>Sub-Q4</b>	Em qual área de conhecimento do GPS a abordagem é aplicada?
<b>Sub-Q5</b>	Em qual fase do GPS a abordagem é aplicada?
<b>Sub-Q6</b>	Qual é o tipo de abordagem?
<b>Sub-Q7</b>	Quais os pontos fortes e fracos das abordagens?
<b>Sub-Q8</b>	Como foi feita a avaliação das abordagens?

Fonte: O autor (2021)

#### 4.ESCOPO DA PESQUISA

Como escopo deste estudo, alguns critérios de seleção, bem como algumas restrições foram definidas, a fim de que este MSL ocorra de forma adequada e a sua realização seja viável.

##### 4.1.Critérios de seleção das fontes de buscas

A seleção das fontes ocorrerá seguindo os critérios de seleção definidos no Quadro 3.

Quadro 19 - Critérios de seleção das fontes de busca

<b>Id.</b>	<b>Critério</b>
CS1	Disponibilidade para consultas web
CS2	Disponibilidade para busca de estudos por meio do portal de periódicos da CAPES, utilizando o domínio da UFPA.
CS3	Disponibilidade de estudos na íntegra através do portal de periódicos da CAPES, utilizando o domínio da UFPA ou a partir da utilização da <i>engine</i> de busca Google e/ou Google Scholar.
CS4	Disponibilidade de estudos em inglês.
CS5	Relevância da fonte.

Fonte: O autor (2021)

As fontes adotadas para este estudo são as que possuem artigos na íntegra, a fim de minimizar o retrabalho dos pesquisadores em buscar manualmente artigos que disponibilizam apenas título e *abstract*. No entanto, caso ocorra de retornar um artigo o

qual não seja possível baixá-lo na íntegra, serão realizadas buscas manuais nos *sites* do Google (<http://www.google.com.br/>) e Google Scholar (<http://scholar.google.com.br/>) com o objetivo de encontrar a versão completa do artigo. Agora, se mesmo assim o trabalho não for localizado, ele será desconsiderado.

#### 4.2. Definição das fontes de buscas

Seguindo os critérios de seleção definidos na seção 4.1, foram selecionadas as fontes apresentadas no Quadro 4.

Quadro 20 - Fontes de busca selecionadas

Fonte de busca	Link
ACM Digital Library	<a href="https://dl.acm.org/">https://dl.acm.org/</a>
IEEE Xplore	<a href="https://ieeexplore.ieee.org/">https://ieeexplore.ieee.org/</a>

Fonte: O autor (2021)

As duas fontes apresentadas no Quadro 4 foram selecionadas por possuírem um grande acervo de publicações na área de interesse deste estudo e também por indexarem anais de eventos importantes da computação.

#### 4.3. Restrições

Para este MSL só serão consideradas as fontes que permitam a pesquisa de forma gratuita ou por meio do portal de periódicos da CAPES, pois não poderá incorrer ônus aos pesquisadores. Apenas os estudos que estiverem de acordo com os critérios de seleção, assim como com os critérios de inclusão e exclusão, serão considerados. A pesquisa abrange publicações sobre abordagens de GPS no período 2001 a 2021. Esse período foi definido tendo em vista que o contexto deste MSL está voltado para o desenvolvimento ágil de software e foi no ano de 2001 que surgiu a declaração de valores e princípios essenciais para esse contexto, o Manifesto Ágil (MANIFESTO ÁGIL, 2001).

#### 4.4. Tipo dos Estudos

Serão considerados estudos primários do tipo: experimentais (ou *Empirical Studies*); teóricos (estudos conceituais baseados em um entendimento de uma área); e, relatos de experiência, na forma de artigos completos de periódicos ou conferências.

## 5. ESTRATÉGIAS DE PESQUISA

### 5.1. Métodos de buscas nas fontes

Existem dois métodos que podem ser utilizados para realizar as buscas nas bases: método de busca manual e método de busca automatizada. Sendo assim, esses dois métodos serão utilizados, caso haja necessidade. A busca automatizada ocorrerá por meio de uma *string* de buscas, formada, por sua vez, por uma série de palavras-chave e seus respectivos sinônimos. Enquanto isso, a busca manual levará em consideração o título do trabalho e o nome de seus autores, os quais serão inseridos nos mecanismos de busca web do Google (<http://www.google.com.br/>) e Google Scholar (<http://scholar.google.com.br/>). A busca manual será realizada caso os estudos não estejam disponíveis na íntegra.

### 5.2. Palavras-chave e sinônimos

Com base nas questões de pesquisa, algumas palavras-chave foram definidas a partir da estrutura PICOC. Porém, como já mencionado na seção 3, o critério “Comparação” foi descartado. Além disso, como o critério “Intervenção” trata das ações que serão realizadas, não há a necessidade de incluir palavras-chave para ele. Diante disso, temos as seguintes palavras-chave (Quadro 5).

Quadro 21 - Palavras chaves utilizadas para a elaboração da *string* de busca

CRITÉRIO	PALAVRAS-CHAVE
<b>População</b>	Inglês: <i>project management</i>  Português: gerenciamento de projetos de software
<b>Resultado</b>	Inglês: <i>tool, method, technique, model, technology, practice, standard, guide, work product, methodology, framework, process, principle, theme and profile</i>  Português: ferramenta, método, técnica, modelo, tecnologia, prática, padrão, guia, artefato, metodologia, framework, processo, princípio, tema

	e papel
<b>Contexto</b>	Inglês: <i>software and agile</i> Português: software e ágil

Fonte: O autor (2021)

### 5.3.String de busca

Os termos de pesquisa que conduziram este processo foram reunidos em uma *String* de Busca (SB). Esses termos foram escritos em inglês por ser largamente adotado em conferências e periódicos internacionais relacionados ao tema da pesquisa. Para a elaboração da SB, as palavras-chave e seus sinônimos foram agrupados por meio do operador lógico OR. Enquanto isso, os conjuntos de termos foram concatenados por meio do operador lógico AND. O primeiro conjunto de termos da SB trata da população que, de alguma forma, é impactada pelas abordagens de gerenciamento de projetos. O segundo conjunto de termos trata dos resultados que se espera com a execução das buscas, ou seja, abordagens que são aplicadas na indústria de software e relatadas na literatura. Por fim, o terceiro conjunto de termos se refere ao contexto no qual essas abordagens são aplicadas. A SB e os operadores lógicos utilizados podem ser visualizados no Quadro 6.

Quadro 22 - *String* de busca utilizada

<b>String de Busca (SB)</b>
<i>(“project management”) AND (“software”, “tool” OR “method” OR “technique” OR “model” OR “technolog*” OR “practice” OR “pattern” OR “standard” OR “guide” OR “work product” OR “methodolog*” OR “framework” OR “process” OR “principle” OR “theme” OR “profile”) AND (“software”) AND (“agile”)</i>

Fonte: O autor (2021)

Após a elaboração da SB, ela foi testada para verificar sua eficácia. O teste consistiu em rodar a *string* em uma das bases e verificar se três trabalhos retornariam.

Esses trabalhos foram selecionados após a constatação de que eles atendem os critérios de inclusão definidos no Quadro 7. Os trabalhos são:

- **IEEE XPLORE**

CARNEIRO, L. B.; SILVA, A. C. C. L. M.; ALENCAR, L. H.. Scrum Agile Project Management Methodology Application for Workflow Management: a case study. *In: 2018 IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND ENGINEERING MANAGEMENT (IEEM)*, 12., 2018, Bangkok. **2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)**. [S.L.]: IEEE, 2018. p. 938-942.

SILVA, Lilian Santos Ferreira da; OLIVEIRA, Sandro Ronaldo Bezerra. A Process Framework with Agile Practices for Implementation of Project Portfolio Management Process. *In: 2016 10TH INTERNATIONAL CONFERENCE ON THE QUALITY OF INFORMATION AND COMMUNICATIONS TECHNOLOGY (QUATIC)*, 10., 2016, Lisbon. **2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)**. [S.L.]: IEEE, 2016. p. 146-149.

HAYAT, Faisal *et al.* The Influence of Agile Methodology (Scrum) on Software Project Management. *In: 2019 20TH IEEE/ACIS INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, ARTIFICIAL INTELLIGENCE, NETWORKING AND PARALLEL/DISTRIBUTED COMPUTING (SNPD)*, 20., 2019, Toyama. **2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)**. [S.L.]: IEEE, 2019. p. 145-149.

- **ACM DIGITAL LIBRARY**

HOUSTON, Dan X.; BUETTNER, Douglas J.. Modeling user story completion of an agile software process. *In: THE 2013 INTERNATIONAL CONFERENCE*, 8., 2013, New York, New York, USA. **Proceedings of the 2013 International Conference on Software and System Process - ICSSP 2013**. New York, New York, USA: ACM Press, 2013. p. 88-97.

RAITH, Florian; RICHTER, Ingo; LINDERMEIER, Robert. How Project-management-tools are used in Agile Practice. *In: THE 21ST INTERNATIONAL DATABASE ENGINEERING & APPLICATIONS SYMPOSIUM*, 21., 2017, New York, New York, USA. **Proceedings of the 21st International Database Engineering & Applications Symposium on - IDEAS 2017**. New York, New York, USA: ACM Press, 2017. p. 30-39.

LYRA, Kamila Takayama *et al.* An agile project management experience. *In: THE XXXII BRAZILIAN SYMPOSIUM*, 32., 2018, New York, New York, USA. **Proceedings of the XXXII Brazilian Symposium on Software Engineering - SBES '18**. New York, New York, USA: ACM Press, 2018. p. 240-249.

## 6. CRITÉRIOS DE SELEÇÃO DE PUBLICAÇÕES

Kitchenham e Charters (2007) afirmam que, para selecionar as publicações que retornam com a execução das buscas, é necessário definir e aplicar sobre esses trabalhos

alguns critérios de inclusão (CI) e exclusão (CE). As publicações foram incluídas se atendessem pelo menos um dos CI e excluídos quando se enquadraram em ao menos um dos CE. O Quadro 7 e 8 apresentam os CI e os CE, respectivamente.

Quadro 23 - Critérios de inclusão

<b>Id.</b>	<b>Critérios de Inclusão</b>
CI1	Publicações que apresentam alguma abordagem de gerenciamento de projetos e que foram aplicadas no contexto industrial e/ou acadêmico de desenvolvimento ágil de software.
CI2	Publicações que realizaram a avaliação dessas abordagens.

Fonte: O autor (2021)

Quadro 24 - Critérios de exclusão

<b>Id.</b>	<b>Critérios de Exclusão</b>
CE1	Estudos que não atendam pelo menos um critério de inclusão.
CE2	Estudos repetidos encontrados em diferentes mecanismos de pesquisa. Nesse caso, apenas um estudo será considerado.
CE3	Estudos não disponíveis para download abertamente ou pelo IP institucional dos pesquisadores.
CE4	Estudos que não estejam escritos em inglês.
CE5	Publicações do tipo relatórios de workshops, pôster, apresentações, <i>keynotes speaker</i> , livros, teses e dissertações.
CE6	Artigos publicados fora do período definido para a busca.
CE7	Artigos que claramente não atendam as questões de pesquisa.

Fonte: O autor (2021)

## 7.PROCESSO DE SELEÇÃO DOS ESTUDOS PRIMÁRIOS

Para a execução deste MSL, os seguintes recursos serão utilizados:

4. Dois pesquisadores (um aluno de mestrado e um professor Dr. em Engenharia de Software);
5. Acesso às fontes de pesquisa por meio do domínio da Universidade Federal do Pará;
6. Validações sobre documentos e procedimentos da realização do MSL através de reuniões com o coordenador do Projeto SPIDER e orientador do projeto de dissertação, onde esta pesquisa está inserida, o Prof. Dr. Sandro Ronaldo Bezerra Oliveira;

Durante a condução deste MSL, os estudos primários serão identificados conforme o seguinte processo:

1. A partir da leitura de título, palavra chave e resumo, os artigos deverão passar por uma pré-seleção para verificar se estão enquadrados no contexto do MSL;
2. A partir da leitura do corpo do artigo inteiro, estes deverão ser avaliados quanto aos critérios de inclusão e exclusão, e o resultado deve ser registrado;
3. Os pesquisadores responsáveis pela seleção dos artigos devem entrar em consenso, quando necessário. Isso ocorre quando não há uma unanimidade na inclusão de um artigo e pelo menos um pesquisador decide não incluir.
4. Na fase de consenso, em caso de discordância sobre a inclusão de algum estudo, o mesmo deverá ser incluído.
5. Os estudos primários identificados serão posteriormente lidos em totalidade e então será aplicada a avaliação de qualidade e a estratégia de extração de dados, conforme descrito nas subseções seguintes.

## **8.CRITÉRIOS DE QUALIDADE DOS ESTUDOS PRIMÁRIOS**

Quando se trata de MSL, além dos critérios de inclusão e exclusão, é preciso levar também em consideração alguns critérios que avaliam a qualidade dos trabalhos selecionados (KITCHENHAM, 2004). Essa qualidade é medida pela relevância e pelo valor científico do trabalho. Vale destacar que, realizar a avaliação dos trabalhos retornados também é um critério de inclusão e exclusão, mas de forma mais detalhada.

Desse modo, como processo adicional de validação dos estudos, serão aplicados critérios de qualidade que permitirão a identificação de trabalhos que ainda podem ser desconsiderados da pesquisa e observar individualmente os estudos a fim de obter comparações durante a síntese de dados. Os Critérios de Qualidade (CQ) apresentados no Quadro 9, foram adaptados de Kitchenham e Charters (2007) e, Kitchenham, Budgen e Brereton (2015), uma vez que descreviam critérios abrangentes o suficiente para cobrir o escopo das publicações a serem considerados, com alterações para adequarem-se aos objetivos e questões de pesquisa deste MSL.

Quadro 25 - Critérios de qualidade

<b>Código</b>	<b>Critério de Qualidade</b>	<b>Foco de Análise</b>
---------------	------------------------------	------------------------

CQ1	<ul style="list-style-type: none"> <li>• Os objetivos ou questões do estudo são claramente definidos (incluindo justificativas para a realização do estudo)?</li> <li>• O tipo de estudo está definido claramente?</li> </ul>	Introdução
CQ2	<ul style="list-style-type: none"> <li>• Existe uma clara descrição do contexto no qual a pesquisa foi realizada?</li> <li>• O trabalho é bem/adequadamente referenciado (apresenta trabalhos relacionados ou semelhantes e baseia-se em modelos e teorias da literatura)?</li> </ul>	Desenvolvimento
CQ3	<ul style="list-style-type: none"> <li>• O estudo relata de forma clara e não ambígua os resultados?</li> <li>• Os objetivos ou questões do estudo são alcançados?</li> </ul>	Conclusão
CQ4	<ul style="list-style-type: none"> <li>• O estudo apresenta primariamente ou secundariamente abordagens de gerenciamento de projetos que foram utilizadas no desenvolvimento ágil de software?</li> <li>• O estudo endereça explicitamente as abordagens ao GPS?</li> </ul>	Questão de Investigação
CQ5	<ul style="list-style-type: none"> <li>• Existe uma descrição sobre a(s) organização(ões) onde foi conduzido o estudo?</li> </ul>	Relato de Experiência Industrial

Fonte: O autor (2021)

## 9. ESTRATÉGIA DE EXTRAÇÃO DE INFORMAÇÃO

### ➤ Qual é o tipo de abordagem?

Deve ser identificado o tipo de abordagem de GPS que foi utilizada no estudo, por exemplo: software, metodologia, técnica, etc.

### ➤ Em qual área de conhecimento do GPS a abordagem é aplicada?

Deve ser identificada a área de conhecimento onde a abordagem foi utilizada, como por exemplo: comunicação, cronograma, riscos, dentre outras.

### ➤ Em qual fase do GPS a abordagem é aplicada?

Deve ser identificado em qual fase do projeto a abordagem foi aplicada, ou seja, se foi na iniciação, planejamento, execução, monitoramento e controle ou encerramento.

### ➤ Qual(is) o(s) ponto(s) forte(s) e fraco(s) da abordagem?

Deve ser identificado pontos relevantes que tratem das principais vantagens e desvantagens em utilizar a abordagem identificada.

### ➤ Como foi feita a avaliação da abordagem?

Deve ser identificada qual foi ou quais foram os métodos utilizados para avaliar o uso da abordagem identificada.

➤ **Como foi a evolução do número de estudos publicados relacionados com o tópico desta pesquisa?**

Deve ser identificado se o número de estudos relacionados ao tem cresceu, diminuiu ou se manteve estável.

➤ **Quais os países que mais publicam trabalhos relacionados com o tópico desta pesquisa?**

Deve ser identificado os países que mais realizam trabalhos na área de pesquisa abordada neste trabalho.

➤ **Qual o tipo de instituição que mais publicam trabalhos relacionados com o tópico desta pesquisa?**

Deve ser identificado qual o tipo de instituição que mais publicam trabalhos relacionados com a área de pesquisa deste estudo.

## REFERÊNCIAS

ANDRADE, S. C.; TAIT, T. F. C.. Uma aplicação do guia PMBOK na gestão de projetos de software. **Revista Brasileira de Computação Aplicada**, Passo Fundo, v. 4, n. 1, p. 2-11, mar. 2012.

BASILI, V. R. **Software modeling and measurement: the Goal/Question/Metric paradigm**. 1992.

BEECHAM, S. *et al.* Motivation in Software Engineering: a systematic literature review. **Information And Software Technology**, [S.L.], v. 50, n. 9-10, p. 860-878, ago. 2008.

COSTA, C. S.. **Uma abordagem baseada em evidências para Gerenciamento de Projetos no Desenvolvimento Distribuído de Software**. 2010. 168 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2010.

KERZNER, H.. **Gestão de projetos: as melhores práticas**. 3. ed. Porto Alegre: Bookman, 2017.

KITCHENHAM, B. **Procedures for performing systematic reviews**. Keele, UK, Keele University, 33(2004):1–26, 2004.

KITCHENHAM, B.; BUDGEN, D.; BRERETON, P.. **Evidence-based software engineering and systematic reviews**. CRC press, 2015.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. Staffordshire, UK. 2007.

LOPES, V. P.; TRAVASSOS, G. H.. Experimentação em Engenharia de Software: glossário de termos. In: VI EXPERIMENTAL SOFTWARE ENGINEERING LATIN AMERICAN WORKSHOP, 6., 2009, São Carlos, Sp, Bra. **Anais [...]** . São Carlos, Sp, Bra: Eselaw, 2009. p. 42-51.

MANIFESTO ÁGIL. Agile manifesto. **Haettu**, v. 14, p. 2012, 2001.

MARTINS, J. C. C.. **Gerenciamento de projetos de desenvolvimento de Software com PMI, RUP e UML**. 5. ed. Rio de Janeiro: Brasport, 2010.

PMBOK. **Um guia do conhecimento em gerenciamento de projetos**. 6. ed. Newtown Square: PMI, 2017. 763 p.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH Editora, 2016.

MAS, Antònia; MESQUIDA, Antoni-Lluís; PACHECO, Marcos. Supporting the deployment of ISO-based project management processes with agile metrics. **Computer Standards & Interfaces**, [S.L.], v. 70, p. 103405-103415, jun. 2020.

### Apêndice 3

#### CATÁLOGO DE ABORDAGENS DE GERENCIAMENTO DE PROJETOS USADAS NO CONTEXTO DO DESENVOLVIMENTO ÁGIL

##### SOFTWARES

NOME DA ABORDAGEM	FORNECEDOR/DESENVOLVEDOR	SITE DE ACESSO
T3 Software	Valerie Trapa e Santhpur Rao	<a href="https://ieeexplore.ieee.org/document/1667585">https://ieeexplore.ieee.org/document/1667585</a>
<b>DESCRIÇÃO</b>		
<p>A ferramenta T3 é um software leve baseado em JUnit que fornece visibilidade do andamento do projeto para dar suporte ao desenvolvimento rápido. Usando visualizações centradas na história, as equipes ágeis podem avaliar rapidamente o status, determinar dependências e identificar pontos problemáticos. O T3 para relata o status realista do projeto e melhora a comunicação entre os membros da equipe distribuídos geograficamente.</p>		
<b>FINALIDADE</b>		
<p>O T3 tem como finalidade abordar questões sistemáticas em uma ampla gama de projetos. Além disso, ele busca auxiliar na fácil identificação da complexidade funcional do projeto e a experiência dos desenvolvedores.</p>		
<b>COMO FUNCIONA</b>		
<p>T3 é um software que calcula a porcentagem de métricas completas do projeto. Além disso, fornece uma visão geral, por meio de relatórios, sobre o tempo de implementação de cada uma das histórias de usuários e do tempo total de implementação de todo o projeto, com base nessas histórias de usuários.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Sistemas CSCW	Tobias Hildenbrand, Michael Geisser, Thomas Kude, Denis Bruch, Thomas Acker	<a href="https://ieeexplore.ieee.org/document/4606732">https://ieeexplore.ieee.org/document/4606732</a>
<b>DESCRIÇÃO</b>		
Os sistemas CSCW ou de trabalho cooperativo suportado por computador (CSCW) referem-se a grupos de software que trabalham simultaneamente em um objetivo comum. A aplicação dos CSWS pode compensar a falta de proximidade física para promover a aplicabilidade do XP e do ASD em geral em ambientes distribuídos. Além disso, a abordagem auxilia na comunicação da equipe e na integração com o cliente.		
<b>FINALIDADE</b>		
Os Sistemas CSCW, também chamados de Sistemas Groupware, têm como finalidade apoiar a comunicação do projeto, em especial aqueles distribuídos. Os sistemas CSCW visam a transferência de informações mais rápida, melhor utilização do conhecimento tácito, agilização dos processos de desenvolvimento e redução da sobrecarga administrativa.		
<b>COMO FUNCIONA</b>		
Os sistemas CSCW equipam os membros do grupo com meios de comunicação e interfaces para um ambiente de trabalho compartilhado, fomentando assim a consciência do grupo, ou seja, o conhecimento sobre as atividades atuais de outros membros do grupo. As equipes são treinadas para coordenar seus processos de trabalho com mais eficiência.		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
MasePlanner	Roberto Morgan e Frank Maurer	<a href="https://ieeexplore.ieee.org/document/4031753">https://ieeexplore.ieee.org/document/4031753</a>
<b>DESCRIÇÃO</b>		
<p>MasePlanner é um software de planejamento ágil que oferece suporte para o planejamento baseado em cartão, facilitando a comunicação não verbal. O MasePlanner foi desenvolvido para melhorar o processo de planejamento, fornecendo às equipes um ambiente digital que oferece suporte ao gerenciamento de informações.</p>		
<b>FINALIDADE</b>		
<p>O MasePlanner tem como finalidade oferecer suporte a reuniões de planejamento, fornecendo um espaço de trabalho comum semelhante a uma mesa para criar e organizar cartões de história.</p>		
<b>COMO FUNCIONA</b>		
<p>O MasePlanner suporta a criação de cartões de história e iterações permitindo sua organização. Ele permite o planejamento baseado em cartão que fornece conhecimento espacial da localização dos cartões. O MasePlanner foi projetado como uma ferramenta de suporte e, como tal, requer que exista um canal de voz de alta qualidade para que ocorra uma verdadeira colaboração. O MasePlanner permite que os artefatos de planejamento sejam criados, editados e organizados de maneira semelhante às reuniões de planejamento em papel.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Sistema de Apoio à Decisão de Planejamento de Sprint (SPESS)	Alhejab Alhazmi e Shihong Huang	<a href="https://ieeexplore.ieee.org/document/8479063">https://ieeexplore.ieee.org/document/8479063</a>
<b>DESCRIÇÃO</b>		
O SPESS é um software auxiliar os gestores no planejamento de Sprint. O SPESS é baseado principalmente em três fatores: a competência do desenvolvedor, a antiguidade do desenvolvedor e a dependência da tarefa.		
<b>FINALIDADE</b>		
Este software visa atribuir as tarefas de cada Sprint aos desenvolvedores garantindo que cada membro da equipe contribua com o máximo de seu potencial, e o planejamento do projeto seja otimizado para o menor tempo possível.		
<b>COMO FUNCIONA</b>		
O SPESS reduz do tempo da sprint em até um dia. Isso se dá pelo fato de o SPESS ser aplicado considerando que todos os desenvolvedores têm a competência necessária para trabalhar em qualquer tarefa. A razão da diferença neste caso particular é devido ao seu método que depende da igualdade do número de tarefas atribuídas a cada desenvolvedor. Além da antiguidade dos desenvolvedores, o SPESS também leva em consideração dois fatores adicionais de planejamento de software - os níveis de competência dos desenvolvedores e a dependência das tarefas. A competência dos desenvolvedores é um dos principais fatores para determinar a quais desenvolvedores adequados são atribuídos as tarefas que correspondem às suas habilidades. O fator de dependência da tarefa é usado para determinar a ordem em que as tarefas são desenvolvidas. Entre cada iteração no Sprint, o SPESS verifica o status de trabalho dos desenvolvedores e prepara o número de desenvolvedores com os quais trabalhar para a próxima iteração. Como resultado, a ferramenta SPESS atinge seu objetivo principal de que os desenvolvedores trabalhem nas tarefas de acordo com seu nível de habilidade com o tempo de acabamento ideal.		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Bluejay	ISA Group , Universidade de Sevilla	<a href="https://www.governify.io/quickstart/auditing-agile">https://www.governify.io/quickstart/auditing-agile</a>
<b>DESCRIÇÃO</b>		
<p>Bluejay é um software composto por um conjunto de componentes que permite modelar, monitorar e auditar práticas de equipes (TP). Ele é software de código aberto que usa as APIs de várias ferramentas para coletar dados brutos, sintetizá-los em medições de práticas de equipes e apresentar painéis para auditar as práticas de equipes (TP). Ele fornece um ecossistema de ferramentas para os instrutores definirem suas próprias práticas recomendadas para seguir e rastrear a aderência das equipes a fim de aprender sobre os riscos e melhorar com o tempo. Especificamente, o framework fornece uma arquitetura de microsserviços baseada no ecossistema Governify.</p>		
<b>FINALIDADE</b>		
<p>O Bluejay fornece um ecossistema de ferramentas para os instrutores definirem suas próprias práticas e rastrear a adesão das equipes a elas, permitindo que elas possam ser melhoradas ao longo do tempo.</p>		
<b>COMO FUNCIONA</b>		
<p>Primeiro, é realizada a modelagem do processo pelo qual os instrutores definem um conjunto de TP, criando um Contrato de Prática de Equipe (TPA) para as equipes cumprirem.</p> <p>Em segundo lugar, é iniciado um processo de monitoramento para cada TPA; como parte desse processo, a Bluejay calcula o cumprimento do TPA com base nas métricas calculadas com as informações coletadas das diferentes APIs REST; as informações reais de atendimento e os valores das métricas são posteriormente renderizados em um painel gráfico disponível para instrutores e equipes.</p> <p>Terceiro, os instrutores podem usar o painel para auditar as equipes e determinar se elas aderem às práticas.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Slack	Slack Technologies, Inc.	<a href="https://slack.com/intl/pt-br/">https://slack.com/intl/pt-br/</a>
<b>DESCRIÇÃO</b>		
Slack é uma plataforma de comunicação comercial em que pessoas e grupos podem se comunicar e compartilhar tarefas e arquivos. Ela é adotada por diversas empresas visando aumentar não apenas a produtividade como também ampliar a comunicação entre funcionários.		
<b>FINALIDADE</b>		
O Slack tem como finalidade transformar a forma como as organizações se comunicam, reunindo pessoas para trabalhar como uma equipe unificada em um app de comunicação baseada em canais.		
<b>COMO FUNCIONA</b>		
<p>O primeiro passo é estar em um workspace (área de trabalho). Para entrar em um workspace é necessário ser convidado ou criar um.</p> <p>Com um workspace criado, é possível editar o grupo de trabalho convidando pessoas para participar dos chats.</p> <p>Em seguida é possível criar os canais, que são grupos de mensagens identificados pelo símbolo # seguido do nome do grupo. Por exemplo: se quiser criar um grupo apenas com integrantes de um time de vendas, basta adicionar um canal, escolher os integrantes que fazem parte da equipe e nomear o grupo como #vendas.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Jira Software	Atlassian Corporation	<a href="https://www.atlassian.com/br/software/jira">https://www.atlassian.com/br/software/jira</a>
<b>DESCRIÇÃO</b>		
<p>O Jira é um software de gerenciamento de projetos desenvolvido pela Atlassian. Esse software reúne todas as funcionalidades necessárias para o controle e organização de diferentes tipos de projetos, equipes e ambientes de desenvolvimento. Oferece recursos úteis para a criação de fluxos de trabalho, sendo considerado uma das melhores ferramentas para gerenciar metodologias ágeis.</p>		
<b>FINALIDADE</b>		
<p>A princípio, Jira foi desenvolvido como um rastreador de bugs e itens. Hoje, o Jira Software é uma poderosa ferramenta de gerenciamento para todos os tipos de casos de uso, desde gerenciamento de casos de teste e requisitos, até desenvolvimento ágil de software.</p>		
<b>COMO FUNCIONA</b>		
<p>O Jira Software pode ser configurado para se adequar a qualquer tipo de projeto. As equipes podem começar com um template de projeto ou criar o próprio fluxo de trabalho personalizado. Os itens do Jira, também conhecidos como tarefas, rastreiam cada trabalho que precisa passar pelas etapas do fluxo de trabalho até a conclusão. Permissões personalizáveis permitem que os administradores determinem quem pode ver e fazer quais ações. Com todas as informações do projeto em vigor, os relatórios podem ser gerados para rastrear o progresso, a produtividade e garantir que nada fique para trás.</p> <p>Para equipes que praticam metodologias ágeis, o Jira Software oferece painéis Scrum e Kanban prontos para uso. Os quadros são centros de gerenciamento de tarefas, em que as tarefas são mapeadas para fluxos de trabalho personalizáveis. Os quadros oferecem transparência em todo o trabalho em equipe e visibilidade sobre o status de cada item de trabalho. Os recursos de rastreamento de tempo e os relatórios de desempenho em tempo real (gráficos de burn-up/down, relatórios de sprint, gráficos de velocidade) permitem que as equipes monitorem de perto a produtividade ao longo do tempo.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Jogo Legado para Gerenciamento de Projetos	Jefferson Seide Molléri, Javier Gonzalez-Huerta e Kennet Henningsson	<a href="https://dl.acm.org/doi/abs/10.1145/3209087.3209094">https://dl.acm.org/doi/abs/10.1145/3209087.3209094</a>
<b>DESCRIÇÃO</b>		
Este software trata-se de um jogo legado para expor os alunos a um projeto de desenvolvimento de software e aos diferentes modelos de processo de software. O jogo representa alguns dos desafios a serem enfrentados ao lidar com o planejamento e gestão de um projeto de software da vida real sem exigir que os alunos desenvolvam o sistema de software.		
<b>FINALIDADE</b>		
Estimular os alunos a adquirir conhecimentos (teóricos e práticos) que lhes permitam resolver desafios semelhantes aos que enfrentarão em projetos de software do mundo real.		
<b>COMO FUNCIONA</b>		

Cada equipe apresenta um plano de projeto.

A equipe deve manter um documento descrevendo um plano de projeto de acordo com os objetivos de aprendizagem do curso.

Durante a primeira rodada, a equipe apresenta uma versão inicial deste plano. Em cada rodada subsequente, uma versão atualizada é enviada.

Se os alunos demonstrarem um plano de projeto sólido, que adere às boas práticas e corresponda às necessidades do projeto, eles recebem bônus para o próximo turno do jogo.

Os alunos também podem ser penalizados se o plano de projeto omitir certas atividades de desenvolvimento necessárias (por exemplo, testes) ou práticas (por exemplo, retrospectiva de sprint).

O jogo consiste principalmente nos alunos jogando dados que representam as incertezas de um projeto de desenvolvimento de software.

Os professores orientam os alunos durante toda a etapa, relacionando essas incertezas a exemplos reais, por exemplo, uma penalidade na implementação pode ser causada por um design não atualizado ou falta de rastreabilidade de requisitos.

Os valores de incerteza são então adicionados aos bônus/penalidades marcados pelo projeto da equipe. Este resultado representa o quanto o processo real se desvia do plano original.

Os professores informam as equipes sobre novos eventos que estão ocorrendo. Eles representam mudanças que provavelmente ocorrerão em um processo de software do mundo real, como novos requisitos ou limitações de recursos.

Os professores desempenham o papel de clientes ou proprietários de produtos, incentivando os alunos a negociar alguns dos novos desafios.

Cada equipe atualiza seu plano de projeto. Finalmente, as equipes são solicitadas a relatar as mudanças no plano do projeto de acordo com o desvio e novos eventos (Passo 5). Eles também são sugeridos para fazer quaisquer melhorias que considerem necessárias para o sucesso do projeto.

O jogo termina e, em seguida, os alunos são solicitados a fornecer suas reflexões sobre o projeto, juntamente com feedback sobre a experiência de jogo e o aprendizado das lições.

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Jogo de Gestão de Projetos	Matthew Beale	<a href="https://dl.acm.org/doi/10.1145/2987592.2987615">https://dl.acm.org/doi/10.1145/2987592.2987615</a>
<b>DESCRIÇÃO</b>		
Este é um jogo para explorar o gerenciamento ágil de projetos e contribuir para a compreensão dos alunos sobre as habilidades e objetivos do estilo ágil de gerenciamento de projetos, bem como construir para seu conhecimento de como um projeto de um mês com várias partes interessadas funciona desde o conceito até a entrega.		
<b>FINALIDADE</b>		
Este jogo tem como finalidade compensar a falta de proximidade física entre os integrantes da equipe, para promover a aplicabilidade do XP e do ASD em ambientes distribuídos. Além disso, a abordagem auxilia na comunicação da equipe e na integração com o cliente.		
<b>COMO FUNCIONA</b>		
Este é um jogo cooperativo que pede a quatro jogadores que se reúnam e usem a metodologia Scrum para resolver diferentes cenários. Eles fazem isso retirando cartas do Scrum e, em seguida, colocando trabalhadores ao redor do tabuleiro de jogo para alocar recursos para lidar com essas tarefas durante a próxima rodada. O jogo é jogado ao longo de um número X de rodadas, em que X muda com base no cenário que os jogadores escolheram para aquele jogo. O objetivo de ter vários cenários possíveis é duplo: mostrar aos jogadores como o gerenciamento de projetos ágil pode ser aplicado a várias situações e atuar como uma configuração de dificuldade para o jogo (ou seja, alguns cenários serão mais difíceis do que outros). Para vencer, os jogadores devem atingir o número necessário de pontos de vitória antes que o rastreador de deadline chegue ao fim.		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Agilefant	Jarno Vähäniitty e Kristian Rautiainen	<a href="http://www.agilefant.org/">http://www.agilefant.org/</a>
<b>DESCRIÇÃO</b>		
<p>Agilefant é um software de gerenciamento de backlog de código aberto aprimorada com recursos de manipulação de requisitos hierárquicos. Agilefant fornece três níveis de planejamento diferentes para gerenciamento de backlog: backlogs de produto, projeto e iteração. Além dos backlogs tradicionais de lista simples, os requisitos podem ser expressos em forma de árvore, a fim de manter a rastreabilidade e a transparência.</p> <p>Além do backlog e do gerenciamento de requisitos, o Agilefant contém diferentes visualizações de composição para os itens em diferentes backlogs, o que pode ser útil para organizações de desenvolvimento de pequeno e médio porte. Essas visualizações de composição incluem recursos de controle de tempo, gerenciamento de trabalho pessoal e gerenciamento de portfólio.</p>		
<b>FINALIDADE</b>		
O Agilfante tem como finalidade permitir que os usuários façam um planejamento mais alto, mantendo o controle de detalhes e atualizações diárias.		
<b>COMO FUNCIONA</b>		
<p>Os gerentes de projeto podem acompanhar os principais marcos, bem como pequenos detalhes com a visualização dinâmica em árvore da história.</p> <p>Os usuários têm a flexibilidade de trabalhar com Scrum, Kanban, SAFe, Waterfall ou GTD. Agilefant oferece uma solução escalável.</p> <p>Agilefant suporta integração com JIRA e Trello.</p> <p>Os usuários podem priorizar projetos em andamento de forma eficaz usando arrastar e soltar.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
XPSuite	CIA, The Company	<a href="https://download.cnet.com/XPSuite/3000-2206_4-10175345.html">https://download.cnet.com/XPSuite/3000-2206_4-10175345.html</a>
<b>DESCRIÇÃO</b>		
<p>o XPSuite, uma ferramenta composta por duas partes: XPSwiki, uma ferramenta para gerenciar projetos XP e XP4IDE, um plug-in para integrar o XPSwiki com o IDE IntelliJ-Idea. O XPSuite foi desenvolvido para dar mais independência e produtividade para as equipes, melhorar a comunicação, preferindo conversas pessoais, a outras formas de comunicação, além de levar uma boa prática o feedback.</p>		
<b>FINALIDADE</b>		
<p>O XPSuite foi desenvolvido para atender às necessidades reais das empresas de software para coletar requisitos e cronogramas em formato eletrônico. A função do XP4IDE é dupla: por um lado, permite ao desenvolvedor visualizar, diretamente integrado ao IDE, dados do processo, histórias de usuário, tarefas, testes de aceitação e artefatos (classes, arquivos de documentação, casos de teste , etc.), tudo gerenciado e salvo no servidor XPSwiki. Por outro lado, ele mede o tempo específico gasto em Estórias de Usuário, Tarefas e Artefatos e envia essas informações para um servidor, automatizando a atividade de rastreamento.</p>		
<b>COMO FUNCIONA</b>		
<p>Gerentes, clientes e desenvolvedores podem acessar o XPSuite por meio de um navegador da web. O XPSuite é construído em cima de um mecanismo Wiki Escrito em Smalltalk chamado Swiki. Wiki é um sistema de gerenciamento de conteúdo muito simples e eficaz. Os usuários podem navegar nas páginas do Wiki e, se tiverem os privilégios apropriados, também podem editar as páginas diretamente da Web usando um pequeno conjunto de regras de formatação simples. O Swiki também permite adicionar estrutura às páginas, definindo formulários de entrada para elas. Desta forma, em um Swiki, podem-se ter páginas restringidas por uma determinada estrutura, que são usadas para conter informações estruturadas, bem como páginas de formato livre, adicionadas livremente às páginas existentes, como notas de Post-it.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Global Manager	Rubén Márquez, Aurora Vizcaíno, Félix Oscar García e Antonio Manjavacas	<a href="https://dl.acm.org/doi/abs/10.1145/3372787.3390443">https://dl.acm.org/doi/abs/10.1145/3372787.3390443</a>
<b>DESCRIÇÃO</b>		
<p>O Global Manager é um jogo onde os participantes podem receber formação e adquirir soft skills importantes através de situações simuladas, especialmente no que diz respeito às situações relacionadas à comunicação, coordenação e controle de um projeto Desenvolvimento Global de Software (<i>Global Software Development – GSD</i>). Além disso, os jogadores podem treinar a resolução de problemas, sendo submetidos a um determinado nível de estresse, como em um projeto real.</p>		
<b>FINALIDADE</b>		
<p>O jogo tenta desenvolver várias habilidades em seus jogadores, proporcionando-lhes, ao mesmo tempo, uma experiência envolvente, agradável e atraente. As habilidades desenvolvidas estão relacionadas à coordenação, comunicação e controle. O objetivo deste jogo é permitir que alunos e profissionais treinem em certas habilidades necessárias para a função de gerente de projeto em ambientes distribuídos. O jogador terá que enfrentar a gestão de um projeto GSD fictício desde o seu início até a entrega do produto e tentar resolver as várias dificuldades e obstáculos que possam surgir.</p>		
<b>COMO FUNCIONA</b>		

Os jogadores primeiro escolhem o tipo de projeto que desejam gerenciar e, em seguida, configuram uma série de parâmetros iniciais que irão influenciar a evolução do projeto GSD. Em primeiro lugar, os jogadores devem escolher o número de sites, o país em que o cliente está localizado e o idioma comum para a comunicação. Por exemplo, vamos imaginar que um jogador escolhe 3 sites, que o cliente estará no Reino Unido e o inglês será o idioma principal. Em seguida, o jogador deve definir a configuração de cada site, e 3 pequenas caixas irão aparecer abaixo desta configuração. Essas caixas mostram uma definição do país em que o projeto está localizado, o tamanho da equipe, o nível médio do idioma comum e, claro, o site principal. O jogador pode, por exemplo, estabelecer que o site principal está no Reino Unido, com alto nível de inglês, que outro site está na Coreia do Sul, com alto nível de inglês, e que o último site é na Espanha, onde o nível de inglês é médio, pois supõe-se que a equipe de trabalho não falará esse idioma com fluência. Na última etapa da configuração, o tipo de comunicação que existe entre cada um dos sites e com o cliente correspondente deve ser gerenciado. Por exemplo, telefone ou Skype para comunicação síncrona e e-mail ou fóruns no caso de comunicação assíncrona. O jogador pode optar pela comunicação por e-mail ou via fóruns entre o Reino Unido e a Coreia do Sul, devido à diferença de horário entre esses países, o que dificultaria a comunicação síncrona. O jogador também pode decidir fazer uma comunicação por e-mail entre o Reino Unido e a Espanha, uma vez que este último grupo de trabalho não fala inglês fluentemente, e a comunicação por telefone pode ser complicada para eles.

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Git	Linus Torvalds	<a href="https://git-scm.com/">https://git-scm.com/</a>
<b>DESCRIÇÃO</b>		
<p>O Git é um sistema de controle de versão distribuído e amplamente adotado pelas empresas de desenvolvimento de software. O Git é um projeto de código aberto maduro e com manutenção ativa desenvolvido em 2005 por Linus Torvalds, o famoso criador do kernel do sistema operacional Linux.</p>		
<b>FINALIDADE</b>		
<p>O Git tem como finalidade manter o histórico dos arquivos e das modificações realizadas nesses arquivos, pois muitas vezes mudamos arquivos em grupo, num movimento único. Dessa forma, pode-se voltar atrás e recuperar o estado do sistema como ele era antes, e comparar as mudanças, para encontrar bug, para estudar otimizações.</p>		
<b>COMO FUNCIONA</b>		
<p>O Git trata seus dados como um conjunto de imagens de um sistema de arquivos em miniatura. Toda vez que você fizer uma alteração, ou salvar o estado de seu projeto no Git, ele basicamente tira uma foto de todos os seus arquivos e armazena uma referência para esse conjunto de arquivos. Para ser eficiente, se os arquivos não foram alterados, o Git não armazena o arquivo novamente, apenas um link para o arquivo idêntico ao anterior já armazenado.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Redmine	Jean-Philippe Lang	<a href="https://www.redmine.org/">https://www.redmine.org/</a>
<b>DESCRIÇÃO</b>		
<p>Redmine é um software livre, gerenciador de projetos baseados na web e ferramenta de gerenciamento de bugs. Ele contém calendário e gráficos de Gantt para ajudar na representação visual dos projetos e seus deadlines (prazos de entrega). Ele pode também trabalhar com múltiplos projetos.</p>		
<b>FINALIDADE</b>		
<p>O principal objetivo do Redmine é auxiliar as equipes a gerenciar melhor um projeto, independentemente da metodologia utilizada.</p>		
<b>COMO FUNCIONA</b>		
<p>O Redmine vem com módulos padrão e, dependendo da aplicação, podem ser adicionados novos, com a ajuda de plugins. O Redmine permite o gerenciamento de tarefas. Esse gerenciamento de tarefas diz respeito não só àquilo que será feito. Dependendo da complexidade de um projeto, as tarefas podem ser agrupadas segundo características em comum. No Redmine é possível criar tarefas que podem ser de três tipos: <i>bug</i>, <i>feature</i> ou <i>support</i>. O Redmine ainda tem suporte para o Gantt, que é responsável por fornecer uma visão geral da relação entre as tarefas e o tempo de execução destas. Permite o acompanhamento detalhado dos projetos, bem como sinaliza se há alguma atividade com o risco iminente de não ser entregue no prazo estipulado. Além disso, o Gantt indica as possíveis razões que podem estar contribuindo para que uma tarefa não cumpra a <i>deadline</i> estipulada.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
SimScrumF	Luiz Ricardo Begosso, Luís Henrique Buzzo Franco, Douglas Sanches da Cunha e Luiz Carlos	<a href="https://dl.acm.org/doi/abs/10.1145/3357419.3357426">https://dl.acm.org/doi/abs/10.1145/3357419.3357426</a>
<b>DESCRIÇÃO</b>		
<p>O SimScrumF é um jogo baseado no RPG <i>–Role-Playing Game</i>, onde o aluno controla as ações de um personagem imerso em algum mundo bem definido. A arquitetura do jogo é multilíngue, para facilitar o aprendizado de alunos de diversos países, e inicialmente estão disponíveis os seguintes idiomas: inglês, português, francês e espanhol; em breve novos idiomas estarão disponíveis. Os alunos podem jogar online e a interação dentro do jogo permite que outros alunos vejam o perfil de todos os jogadores online através do ranking como forma de estimular a competitividade dos jogadores. Alguns frameworks utilizados na implementação desta ferramenta são Java 8, Spring, MySQL e Angular; sua licença é baseada em Apache GPL 2.0 Free Software Foundation.</p>		
<b>FINALIDADE</b>		
<p>O SimScrumF tem como foco promover o engajamento dos alunos no processo de aprendizagem dos conceitos ágeis e da metodologia Scrum.</p>		
<b>COMO FUNCIONA</b>		
<p>O aluno fica imerso em uma situação real de desenvolvimento de software, utilizando a mesma terminologia da equipe de software, e utiliza as mesmas configurações e mecânicas de jogo de um jogo de RPG. O ambiente do jogo tem semelhanças com jogos de papel e caneta, como elementos narrativos, desenvolvimento de personagens do jogador, complexidade, bem como rejogabilidade e imersão. A ideia por trás do jogo é apoiar o ensino da metodologia Scrum, inserindo o aluno em um ambiente virtual que simula as etapas de desenvolvimento de um produto de software em equipe. Enquanto o aluno estiver jogando, ele assumirá todos os papéis que representam o time Scrum: Product Owner, Scrum Master e membro do Time de Desenvolvimento.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Upwave	Upwave Technologies	<a href="https://www.upwave.io/">https://www.upwave.io/</a>
<b>DESCRIÇÃO</b>		
<p>O Upwave é um software de gerenciamento de projetos online, que ajuda as equipes a colaborar e organizar o trabalho do projeto. A ferramenta visa preencher a lacuna entre os modelos de gerenciamento de projetos, as melhores práticas para gerenciamento de projetos e formas simples de gerenciamento de tarefas. Ele também fornece uma interface visual e simples, promovendo a eficiência e motivação do usuário.</p>		
<b>FINALIDADE</b>		
<p>O objetivo do Upwave é ajudar as equipes a colaborar e organizar o trabalho do projeto. A ferramenta visa preencher a lacuna entre os modelos de gerenciamento de projetos, as melhores práticas para gerenciamento de projetos e formas simples de gerenciamento de tarefas. Ele também fornece uma interface visual e simples, promovendo a eficiência e motivação do usuário.</p>		
<b>COMO FUNCIONA</b>		
<p>O Upwave permite diferentes tipos de visualização do projeto, isto é, com o ele é possível alternar facilmente entre diferentes visualizações do seu projeto, incluindo quadro visual, tabela, linha do tempo e calendário. O software ainda permite monitorar o status e o progresso dos projetos em um portfólio, com isso é possível gerenciar, relatar e tomar medidas em projetos com sua própria visualização de portfólio personalizada, oferecendo acesso fácil a todos os dados relevantes. O Upwave permite criar equipes para vários departamentos, grupos de projetos ou parceiros externos. Isso contribui para uma maior clareza das tarefas, projetos e pessoas. Por fim, o software melhora a colaboração ao ter todas as informações em um só lugar.</p>		

## **METODOLOGIAS**

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
BloomSoft	Edgar Castillo-Barrera, Monica Amador-García, Héctor Pérez-González, Francisco Martínez-Pérez e Francisco Torres-Reyes	<a href="https://ieeexplore.ieee.org/document/8645899">https://ieeexplore.ieee.org/document/8645899</a>
<b>DESCRIÇÃO</b>		
<p>A metodologia consiste em um ajuste da Taxonomia de Bloom, ou seja, em uma mudança de hierarquia do nível de Aplicação, que na Taxonomia de Bloom está no nível 3, mas de acordo com o processo de Desenvolvimento de Software passa para o nível 5. A mudança baseia-se na seguinte análise: para construir, integrar e testar um software é necessário ter realizado previamente uma análise e também uma síntese da informação, que corresponde em Engenharia de Software às fases de Design e Programação, respectivamente.</p>		
<b>FINALIDADE</b>		
<p>Permite ter uma forma ágil de classificar a complexidade das histórias de usuário a partir dos verbos nelas identificadas e, ao mesmo tempo, determinar com essa classificação o estágio de Desenvolvimento de Software a que pertence. Desta forma, a equipe de desenvolvimento pode classificar as histórias em etapas e com isso fazer um melhor planejamento dessas histórias a serem realizadas em cada Sprint.</p>		
<b>COMO FUNCIONA</b>		

As etapas do processo de Desenvolvimento de Software ou ciclo de vida clássico são: (1) Pré-análise/Planejamento, (2) Análise/Requisitos, (3) Projeto, (4) Desenvolvimento, (5) Testes e (6) Implementação. Considerando e analisando as etapas mencionadas, é feito um ajuste da Taxonomia de Bloom, que consiste em uma mudança de hierarquia do nível de Aplicação, que na Taxonomia de Bloom está no nível 3, mas conforme o processo de Desenvolvimento de Software passa para o nível 5. A mudança baseia-se na seguinte análise: para construir, integrar e testar um programa de Software, é necessário ter feito previamente uma análise e uma síntese da informação, que corresponde em Engenharia de Software às fases de Design e Programação respectivamente. A proposta segue uma analogia com o ciclo de vida do Software, ou seja, o nível de Aplicação no ciclo de vida do Software é ajustado à fase de Teste, gerando um melhor entendimento e permitindo uma determinação da complexidade das histórias de usuários. Para complementar a identificação da complexidade, foram adicionados verbos em cada nível da taxonomia relacionada à área de Engenharia de Software.

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
DevOps	John Allspaw e Paul Hammond	<a href="https://www.redhat.com/p-t-br/topics/devops">https://www.redhat.com/p-t-br/topics/devops</a>
<b>DESCRIÇÃO</b>		
DevOps vem da junção das palavras <i>Development</i> e <i>Operations</i> que foi mencionado pela primeira vez em 2009 em uma palestra chamada de "10+ Deploys Per Day", por uma dupla que trabalhava no Flickr.		
<b>FINALIDADE</b>		
O DevOps permite que funções anteriormente isoladas – desenvolvimento, operações de TI, engenharia da qualidade e segurança – atuem de forma coordenada e colaborativa para gerar produtos melhores e mais confiáveis. Ao adotar uma cultura de DevOps em conjunto com as práticas e ferramentas de DevOps, as equipes ganham a capacidade de responder melhor às necessidades dos clientes, aumentar a confiança nos aplicativos que constroem e cumprir as metas empresariais mais rapidamente.		
<b>COMO FUNCIONA</b>		
<p>Com a implementação de um modelo de DevOps, as equipes de desenvolvimento e operações não ficam mais separadas. Às vezes, essas duas equipes são combinadas em uma só. Os engenheiros trabalham durante o ciclo de vida inteiro do aplicativo, da fase de desenvolvimento e testes à fase de implantação e operações, e desenvolvem várias qualificações não limitadas a uma única função.</p> <p>Em alguns modelos de DevOps, as equipes de controle de qualidade e segurança também podem aumentar sua integração com o desenvolvimento, as operações e todo o ciclo de vida dos aplicativos. Quando a segurança é a prioridade de todos em uma equipe de DevOps, isso é denominado às vezes DevSecOps.</p> <p>Essas equipes usam práticas para automatizar processos que historicamente sempre foram manuais e lentos. Eles usam uma pilha de tecnologia e ferramentas que os ajudam a operar e desenvolver aplicativos de modo rápido e confiável. Essas ferramentas também ajudam os engenheiros a realizar tarefas independentemente (por exemplo, implantação de código ou provisionamento de infraestrutura) que normalmente exigiriam a ajuda de outras equipes, e isso aumenta ainda mais a velocidade da equipe.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Scrum	Scrum.org / Jeff Sutherland	<a href="https://www.scrum.org/">https://www.scrum.org/</a>
<b>DESCRIÇÃO</b>		
<p>O Scrum é uma estrutura que ajuda as equipes a trabalharem juntas. Semelhante a uma equipe de rugby (de onde vem o nome) treinando para o grande jogo, o Scrum estimula as equipes a aprenderem com as experiências, a se organizarem enquanto resolvem um problema e a refletirem sobre os êxitos e fracassos para melhorarem sempre.</p>		
<b>FINALIDADE</b>		
<p>O Scrum tem como finalidade ser uma estrutura leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptáveis para problemas complexos.</p>		
<b>COMO FUNCIONA</b>		
<p>A metodologia Scrum propõe que um projeto seja dividido em diversos (pequenos) ciclos de atividades, chamados de sprint, com reuniões frequentes para que a equipe possa alinhar o que vem fazendo e pensar formas de melhorar o processo com agilidade. Essa metodologia propõe que o projeto seja acompanhado sempre bem de perto e passe por mudanças de planejamento o tempo todo, de forma livre e pouco engessada.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
NORPLAN	Weam M. Farid e Frank J. Mitropoulos	<a href="https://ieeexplore.ieee.org/document/6567463">https://ieeexplore.ieee.org/document/6567463</a>
<b>DESCRIÇÃO</b>		
<p>O NORPLAN utiliza métricas de gerenciamento de projetos recém-propostas, métricas de qualidade de requisitos e um algoritmo orientado a riscos para calcular uma sequência de implementação de requisitos aprimorada que pode ajudar gerentes de projeto ágeis e equipes Scrum na priorização e planejamento de Requisitos Não Funcionais (NFRs).</p>		
<b>FINALIDADE</b>		
<p>O objetivo fundamental da Metodologia NORPLAN é desenvolver uma estrutura de modelagem de NFRs simples, porém sistemática, adaptada para processos ágeis. Além disso, os objetivos incluem propor novos requisitos de qualidade e métricas de gerenciamento de projetos, desenvolver uma ferramenta de simulação, computar e visualizar uma sequência de implementação de requisitos baseada em risco.</p>		
<b>COMO FUNCIONA</b>		
<p>A Metodologia NORPLAN propõe uma taxonomia de Requisitos Ágeis que classifica os requisitos como funcionais ou não funcionais. Um Caso de Uso também faz parte da taxonomia proposta e representa soluções potenciais de NFRs. Entre os conceitos fundamentais da Metodologia NORMAP estão as recém-propostas métricas de gerenciamento de projetos, métricas de qualidade de requisitos e o algoritmo orientado a riscos proposto para calcular uma sequência de implementação de requisitos potencialmente aprimorada. A sequência de implementação é orientada pelo cálculo de riscos técnicos e de gerenciamento de projetos para todos os requisitos, que podem ser usados para ajudar equipes ágeis e gerentes de projeto no planejamento de quais requisitos seriam implementados em quais releases/sprints. O plano resultante é gerado como um cronograma em forma de árvore que inclui ciclos de liberação e sprint e os requisitos específicos em cada sprint.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Metodologia de Ensino de Gestão de Projetos	Kamila Takayama Lyra, Marcela Lopes Alves, Fernando H. Carvalho Silva	<a href="https://dl.acm.org/doi/10.1145/3266237.3266248">https://dl.acm.org/doi/10.1145/3266237.3266248</a>
<b>DESCRIÇÃO</b>		
A metodologia abre a possibilidade de uma abordagem mais sistemática de apoio ao ensino da engenharia de software, primeiro com uma forma de definir um conjunto de práticas de equipa para as equipas seguirem e também na observação dos resultados arquivados para uma melhoria contínua.		
<b>FINALIDADE</b>		
Um dos principais desafios do ensino de engenharia de software (SE) é proporcionar aos alunos experiências e habilidades significativas que possam ser usadas em projetos do mundo real. Para superar esse desafio, a adoção de atividades que vão além das aulas teóricas tem apresentado bons resultados. A Metodologia de Ensino de Gestão de Projetos visa realizar projetos práticos colaborativos que exigem que os alunos apliquem amplamente os conceitos de engenharia de software que podem levar a um melhor desenvolvimento de habilidades.		
<b>COMO FUNCIONA</b>		
Em vez das tradicionais aulas teóricas e testes ao longo do semestre, os alunos desenvolvem um projeto que simula as fases/passos de um projeto real de desenvolvimento de software. Nesse método, os alunos de pós-graduação desempenham o papel de gerentes de uma equipe de desenvolvimento de software, enquanto os alunos de graduação compõem a equipe de desenvolvimento. A dinâmica abrange desde a fase de contratação de uma equipe até a fase final dos projetos, que é a entrega da documentação do projeto e a apresentação de um protótipo ao final do semestre. Visando utilizar o conhecimento prévio dos alunos de pós-graduação e fomentar a disseminação do conhecimento, o professor pode decidir em propor uma dinâmica de grupo para integrar os alunos. O projeto pode ser dividido em quatro fases distribuídas ao longo do semestre. As aulas continuam normalmente durante a execução do projeto, pois as aulas também podem ser organizadas com o objetivo de auxiliar os alunos ao longo das fases, bem como utilizar os conceitos no projeto. Cada é planejada para representar uma fase de desenvolvimento de software, portanto, ao final de cada fase, as equipes entregam um artefato de software.		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
uScrum	Joaquim Baptista	<a href="https://dl.acm.org/doi/abs/10.1145/1456536.1456596">https://dl.acm.org/doi/abs/10.1145/1456536.1456596</a>
<b>DESCRIÇÃO</b>		
<p>O uScrum gerencia a incerteza e o desconhecido, permitindo que os escritores reajam rapidamente às mudanças nas condições. uScrum usa ordens de ignorância para entender a dificuldade das tarefas, permitindo que a equipe priorize efetivamente o trabalho regular junto com o trabalho criativo difícil. O uScrum overbook escritores em ciclos iterativos chamados sprints, então permite que os escritores microgerenciem suas tarefas para superar obstáculos. Após cada sprint, a equipe decide o que publicar e se continuará com o trabalho inacabado.</p>		
<b>FINALIDADE</b>		
<p>Tarefas difíceis podem exigir que a equipe atrase o trabalho de manutenção de curto prazo em alguns sprints. As tarefas podem permanecer no backlog por anos sem nunca terem sido iniciadas. São boas ideias que são muito difíceis de colocar em prática, pequenas tarefas que simplesmente não são importantes o suficiente ou trabalho que não pode ser feito por falta de recursos.</p>		
<b>COMO FUNCIONA</b>		
<p>O uScrum prioriza as tarefas em iterações mensais chamadas sprints, com base em sua importância, urgência e oportunidade. Para conciliar tarefas mais longas e difíceis com pequenas tarefas importantes, a equipe deve desenvolver uma visão compartilhada de longo prazo do trabalho principal, bem como um sentimento para a importância e urgência do trabalho de curto prazo futuro. Sprints baseiam-se na ideia de reservar em excesso os escritores com mais trabalho do que se espera que eles concluam e, em seguida, permitir que os escritores gerenciem o trabalho exato que é feito. Em um mês com 22 dias úteis, por exemplo, um escritor pode ter um trabalho estimado em 18 a 30 dias. As iterações mensais permitem que a equipe reaja rapidamente às mudanças nas necessidades dos clientes internos e externos. Exigir que novos trabalhos “urgentes” fossem priorizados no final do mês manteve a maioria dos trabalhos “urgentes, mas não importantes” sob controle, evitando interrupções causadas por trabalhos que se revelaram inúteis.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
TCBPMM - Metodologia de Gerenciamento de Projetos Baseadas em Controle	BV Rajeev e Vinod Hejib	<a href="https://dl.acm.org/doi/10.1145/3196369.3196394">https://dl.acm.org/doi/10.1145/3196369.3196394</a>
<b>DESCRIÇÃO</b>		
A metodologia foca na excelência técnica e um objetivo de criar equipes em forma de T, envolvendo a transformação de uma cultura de trabalho baseada em controle para uma cultura de trabalho auto-organizada.		
<b>FINALIDADE</b>		
As Metodologias de Gerenciamento de Projetos Baseadas em Controle (TCBPMM) tradicionais evitam mudanças, por meio de um planejamento extensivo antes do sistema ser desenvolvido. A indústria sempre exige metodologias de gerenciamento de projetos com a capacidade de se adaptar às novas necessidades dos usuários.		
<b>COMO FUNCIONA</b>		
A equipe seguiu o princípio “Equipe feliz produz um ótimo trabalho”. O Gerente de Projeto de uma equipe trabalha como "líder servo", permitindo que a equipe atinja seu potencial máximo. O principal objetivo do gerente de projetos é o empoderamento da equipe, processo eficiente e voltado para a melhoria contínua que satisfaça tanto a equipe quanto o cliente. Um acordo de trabalho foi estabelecido com pontos centrais de trabalho de Orientação, Coaching, Colaboração e propriedade baseada em Equipe. Nosso objetivo foi criar um ambiente de fornecer e buscar feedback construtivo em uma base regular, o respeito por todos os indivíduos é a chave, Abraçar a transparência, liderança compartilhada em equipe e reconhecer o bom trabalho.		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
HASE	Jun Lin, Han Yu, Zhiqi Shen e Chunyan Miao	<a href="https://dl.acm.org/doi/10.1145/2642937.2642959">https://dl.acm.org/doi/10.1145/2642937.2642959</a>
<b>DESCRIÇÃO</b>		
<p>A metodologia oferece aos usuários uma nova fonte de dados de comportamento coletados discretamente que fornecem informações sobre as características e os efeitos da tomada de decisão que não podem ser extraídos dos métodos tradicionais de pesquisa e pesquisa baseados em entrevistas. Como os membros da equipe não são colocados em uma situação formal na qual sabem que estão sendo estudados, é mais provável que os padrões de comportamento capturados por esses dados reflitam seu comportamento real.</p>		
<b>FINALIDADE</b>		
<p>HASE é uma metodologia que fornece cinco recursos principais para apoiar o gerenciamento ágil de projetos que abrangem as fases de planejamento do sprint, sprint e retrospectiva do sprint.</p>		
<b>COMO FUNCIONA</b>		
<p>Para criar perfis de usuário, o HASE exige que os registrantes especifiquem seus níveis de competência autoavaliados em diferentes áreas de especialização, como familiaridade com linguagens de programação específicas, metodologias de design de sistema e ferramentas de design de interface de usuário (UI), etc. equipes e a atribuição de diferentes funções dentro de uma equipe (por exemplo, programadores e designers de interface do usuário). Os usuários do HASE podem se comportar como se estivessem usando qualquer ferramenta APM sem gastar esforço adicional. Assim, o processo de coleta de dados permanece discreto para os membros da equipe ASD. Com o tempo, as informações de perfil do usuário, informações de tarefas, decisões de alocação de tarefas e a pontualidade e qualidade das tarefas concluídas podem revelar informações sobre os padrões de comportamento das equipes de ASD que são úteis para a construção de sistemas automatizados de suporte à decisão de alocação de tarefas. Ele ajuda as equipes ASD a gerenciar seus projetos enquanto rastreiam discretamente os dados gerados durante o processo ASD para apoiar a pesquisa neste domínio.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Atividades Baseadas em Lego	Kiev Gama	<a href="https://dl.acm.org/doi/abs/10.1145/3350768.3353817">https://dl.acm.org/doi/abs/10.1145/3350768.3353817</a>
<b>DESCRIÇÃO</b>		
<p>Com base nas reações e reclamações dos alunos durante a dinâmica, que foram reforçadas nas respostas do questionário, no semestre seguinte, utilizamos uma sala de aula diferente para este curso, com mesas e cadeiras destacadas, ao invés de poltronas. As mesas podiam ser facilmente movidas (Figura 3) e facilitavam o trabalho em equipe. A super-simplificação mencionada na atividade de requisitos nos fez refletir se a atividade poderia ser mais detalhada e estendida para caber no tempo total de uma aula, assim como a atividade Scrum, em vez de dividir o mesmo slot de 100 minutos em uma atividade que é seguida por uma palestra tradicional.</p>		
<b>FINALIDADE</b>		
<p>A intenção era trazer uma experiência mais envolvente usando uma abordagem de aprendizagem ativa para aumentar sua motivação e engajamento na prática de conceitos abstratos de forma simples (ou seja, usando peças LEGO).</p>		
<b>COMO FUNCIONA</b>		

Em todas as atividades, os alunos foram apresentados aos conceitos de engenharia de software por meio de dinâmicas lúdicas antes de uma palestra tradicional sobre o tema. A intenção era trazer uma experiência mais envolvente usando uma abordagem de aprendizagem ativa para aumentar sua motivação e engajamento na prática de conceitos abstratos de forma simples (ou seja, usando peças LEGO). Estas foram todas as atividades a serem realizadas em grupos de 3 a 5 alunos. Materiais. Para essas atividades, foram utilizados dois conjuntos de 900 peças de LEGO 2 . No momento da redação deste artigo, o custo aproximado de varejo no Brasil era de cerca de R\$ 300,00 (reais brasileiros) cada conjunto, mas eles são mais acessíveis nos mercados dos EUA (U\$ 45,00) e europeu (EUR 45,00). No caso da atividade Scrum, também usamos notas adesivas de papel (por exemplo, post-its). Os preços variam de acordo com a marca, mas são muito mais acessíveis que os tijolos LEGO, embora não sejam reutilizáveis. Formato das aulas. As atividades aconteceram em aulas de 100 minutos (duas aulas consecutivas de 50 minutos). Uma palestra no formato passivo tradicional (ou seja, alunos ouvindo o professor) ocorreu após as atividades baseadas em LEGO, seja no mesmo horário, que foi o caso da atividade de requisitos, ou na próxima aula, que foi o que aconteceu no caso da atividade Lego4Scrum que ocupa todo o slot de 100 minutos.

## **MÉTODOS**

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Planning Poker	Mike Cohn	<a href="https://www.metodoagil.com/planning-poker/">https://www.metodoagil.com/planning-poker/</a>
<b>DESCRIÇÃO</b>		
<p>O Planning Poker é um método utilizado na reunião de planejamento da Sprint que tem como objetivo realizar a estimativa de esforço sobre as tarefas do backlog do produto. Dessa forma, a ferramenta possibilita pontuar e classificar as tarefas com um sistema de números utilizando o conceito de escala Fibonacci por meio de um baralho. A estimativa é realizada de forma consensual entre os membros do time de desenvolvimento.</p>		
<b>FINALIDADE</b>		
<p>O objetivo do Planning Poker é estimar os esforços para implementar as história do usuário e melhorar o desempenho da estimativa em comparação com a estimativa de grupo não estruturado.</p>		
<b>COMO FUNCIONA</b>		

O formato da reunião no Planning Poker permite que todos os envolvidos conheçam todas as informações a respeito do projeto; esse formato gamificado permite que as discussões sejam mais estruturadas; A conversa em grupo, que ocorre depois da revelação das estimativas iniciais, permite reunir mais ideais sobre as histórias do usuário que estão sendo discutidas e abordar os riscos de implementação das iniciativas.

A equipe vai se reunir na presença do Product Owner ao redor de uma mesa. Cada integrante vai ter suas cartas de Planning Poker, realmente como se fosse um baralho, com valores adequados de estimativa de pontos para cada história do usuário.

Não se preocupe, é fácil encontrar cartas de Planning Poker para imprimir, são diversos modelos que podem ser encontrados de Planning Poker Scrum online.

Antes de começar a rodada, um importante ponto para o gerente de projetos na condução dessa reunião é confiar na experiência da equipe e na opinião individual manifestada sobre cada história do usuário.

O Product Owner vai declarar o seu objetivo e o valor de uma história de usuário. Em silêncio, os membros da equipe vão selecionar uma estimativa e preparar um cartão correspondente, voltado para baixo.

Quando todos tiverem selecionado, as cartas serão voltadas para cima e as estimativas lidas em voz alta.

Se todos pegaram o mesmo valor, o consenso será mais fácil. Mas tanto quem deu uma estimativa alta, como quem deu uma estimativa baixa, precisa apresentar suas justificativas. Caso necessário, o Product Owner poderá esclarecer dúvidas dos membros da equipe. Além disso, o Scrum Master faz a mediação das diferenças.

Esses pontos serão discutidos e todos vão buscar um consenso no processo de planejamento, para isso, podem tentar mais uma ou duas rodadas do Planning Poker timer (cronômetro).

O interessante é que essa forma de estimar permite fornecer resultados com tempos bem mais curtos e com precisão maior, especialmente porque não está baseada na opinião de um único membro da equipe, mas sim na inteligência coletiva.

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Simulação Baseada em Multiagente Integrada	Davy De Medeiros Baía	<a href="https://ieeexplore.ieee.org/document/7203112">https://ieeexplore.ieee.org/document/7203112</a>
<b>DESCRIÇÃO</b>		
Uma abordagem abrangente de simulação baseada em multiagentes integrada para apoiar o gerenciamento de projetos de software. Ao apoiar os gerentes de projeto no planejamento e execução do projeto, a abordagem leva em consideração as informações estáticas e dinâmicas.		
<b>FINALIDADE</b>		
O objetivo é ter impacto positivo nas áreas de conhecimento do gerenciamento de projetos de software. Ter influência positiva escopo do projeto. Algumas organizações se concentram na contratação de funcionários orientada para metas e outras não, portanto, também tem um grande impacto na gestão de RH.		
<b>COMO FUNCIONA</b>		
Para lidar com os dois tipos de dados, precisaremos definir e implementar um Filtro de Informações para que os dados possam ser solicitados e enviados de e para o Modelo de Simulação de Planejamento e Modelo de Simulação de Execução (MABS). Também será fornecido suporte para a criação de cenários de acordo com estratégias específicas, que podem ser pré-definidas pelo gerente de projeto ou geradas por Técnicas de Aprendizagem e descoberta de conhecimento. A instância de abordagem inicial será estendida para que o MABS também possa suportar dados de reconhecimento de situação em tempo real.		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
COSMIC	Jean Marc Desharnais, Luigi Buglione e Bugcaronra Kocaturk	<a href="https://dl.acm.org/doi/abs/10.1145/2181101.2181117">https://dl.acm.org/doi/abs/10.1145/2181101.2181117</a>
<b>DESCRIÇÃO</b>		
<p>O método COSMIC também foi aceito pela ISO / IEC JTC1 / SC7 como um padrão internacional (ISO / IEC 19761) [18] [19]. Quanto aos outros métodos de FSM, o COSMIC pode ser aplicado também desde as fases iniciais, necessitando, pelo menos, de requisitos de alto nível a serem refinados durante o ciclo de vida do projeto. No processo de medição, os movimentos de dados são contados, os quais podem ser definidos como “componentes funcionais básicos que movem um único tipo de grupo de dados”. Existem 4 subtipos de movimentos de dados: Entry (E), eXit (X), Read (R) e Write (W). A soma das movimentações de dados dentro do escopo de medição estabelecido retorna o número de Pontos de Função COSMIC (CFP).</p>		
<b>FINALIDADE</b>		
<p>O objetivo do método COSMIC é fornecer um método padronizado de medição do tamanho funcional do software, seja qual for o seu domínio de aplicação.</p>		
<b>COMO FUNCIONA</b>		
<p>Cada usuário descreve uma história do software. Cada história é formulada em uma ou duas frases na linguagem do cliente e normalmente escrita em um cartão de índice. Em vez de transmitir todos os detalhes sobre o recurso, detalhes que provavelmente serão alterados durante o projeto, os cartões serviram como lembretes para conversas sobre os recursos. Os detalhes são então desenvolvidos nas conversas e transmitidos e documentados na forma de testes. O primeiro passo foi contar apenas as histórias de alto nível. A discussão é orientada não sobre o esforço, mas sim sobre o trabalho que deve ser feito com as informações fornecidas para saber o tamanho funcional da história. Em primeiro lugar, nos Estados Unidos, pode-se ler ‘Os clientes podem alugar filmes no local’. Em vez de discutir quanto tempo leva para fazer este trabalho.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Antipadrões	Petr Picha e Přemek Brada	<a href="https://dl.acm.org/doi/abs/10.1145/3361149.3361169">https://dl.acm.org/doi/abs/10.1145/3361149.3361169</a>
<b>DESCRIÇÃO</b>		
<p>Os antipadrões (APs) são “abordagens comuns para resolver problemas recorrentes que se mostram ineficazes”. Ou, dito de outra forma: "Um antipadrão é como um padrão, exceto que, em vez de uma solução, ele dá algo que se parece superficialmente com uma solução, mas não é. Assim, um método de detecção de processos e anti-padrões de PM nos dados do projeto que podem ser usados para avisar as equipes de desenvolvimento de software sobre uma potencial ameaça ao projeto, ou para conduzir estudos mais gerais sobre o impacto da ocorrência de AP no sucesso do projeto e na qualidade do produto.</p>		
<b>FINALIDADE</b>		
<p>O método tem como objetivo prever os estados do projeto sem documentos; documentos de design, qualidade do produto e relatórios de progresso, o modelo pode ser adaptável a várias metodologias de desenvolvimento.</p>		
<b>COMO FUNCIONA</b>		
<p>Os alunos são divididos em equipes de (geralmente) quatro e atribuídos a um projeto com um cliente da vida real de dentro da universidade (grupo de pesquisa, departamento, corpo docente, administração, etc.) ou de outra organização ou parceiro industrial. Eles recebem acesso a um conjunto de ferramentas ALM - por padrão, Redmine para rastreamento de projeto e um repositório git - bem como um modelo de processo específico a seguir, que é, na maior parte, um amálgama de práticas RUP e Scrum, influenciadas também por Entrega Agile Disciplinada (DAD). Durante o projeto, que dura cerca de três meses, a equipe se reúne regularmente com um membro do corpo docente que avalia seu progresso e atua como um mentor. No entanto, a mineração e a análise dos projetos por meio de nosso conjunto de ferramentas SPADe foram feitas retroativamente após a conclusão dos projetos. Assim, os alunos não tinham conhecimento específico da intenção de detectar APs nos dados dos seus projetos, embora as consultas com um mentor os tenham alertado para alguns aspectos dos projetos aos quais os mentores prestam atenção como indícios de potenciais problemas no projeto. Isso inclui os burndowns de iteração, que também manifestam os sintomas do AP.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Escritório de Gerenciamento de Projetos	Volker Gruhn e Niklas Spitzzok Von Brisinski	<a href="https://dl.acm.org/doi/abs/10.1145/3377813.3381361">https://dl.acm.org/doi/abs/10.1145/3377813.3381361</a>
<b>DESCRIÇÃO</b>		
Uma área da organização que possui profissionais especializados em gerenciamento de projetos, que pode ser encontrada em níveis estratégicos, táticos e operacionais, com o objetivo de atuar na gestão de projetos, programas e portfolio, na abertura, priorização, desenvolvimento, implantação e encerramento dos projetos.		
<b>FINALIDADE</b>		
Por meio dessa abordagem, todas as partes interessadas gerenciam quaisquer incertezas que surjam, por exemplo, devido a mudanças na tecnologia ou quaisquer mudanças no ambiente.		
<b>COMO FUNCIONA</b>		
Consideramos um monte de dados de mais de 320 projetos realizados nos últimos anos. Identificamos uma série de riscos para projetos de preço fixo e explicamos como as situações de gastos excessivos não gerenciados podem ser perigosas para clientes, bem como para fornecedores de software. Mostramos uma correlação entre o uso do PMO e a redução do gasto excessivo. Além disso, mostramos que o uso sistemático de ferramentas de PMO veio junto com uma redução ainda maior do gasto excessivo.		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
ESSE - Early Software Size Estimation	Cheng Zhang, Shensi Tong, Wenkai Mo, Yang Zhou, Yong Xia e Beijun Shen	<a href="https://dl-acm-org./doi/10.1145/2993717.2993733">https://dl-acm-org./doi/10.1145/2993717.2993733</a>
<b>DESCRIÇÃO</b>		
<p>Um método inicial de estimativa de tamanho de software (ESSE), que pode extrair recursos semânticos de requisitos de linguagem natural automaticamente e construir modelos de estimativa de tamanho para o projeto. Em primeiro lugar, o ESSE faz uma análise semântica em dois níveis dos documentos de especificação de requisitos por extração de informações e disseminação de ativação. Então, características relacionadas à complexidade são extraídas dos resultados da análise semântica.</p>		
<b>FINALIDADE</b>		
<p>O Objetivo é ser um método unificado de estimativa de tamanho de software inicial que integra processamento de linguagem natural e técnicas de aprendizado de máquina e alcança resultados de estimativa superiores. Um algoritmo de análise semântica de requisitos de dois níveis para extrair recursos relacionados à complexidade dos requisitos.</p>		
<b>COMO FUNCIONA</b>		
<p>O ESSE processa cada requisito em um documento de especificação de requisitos (RSD) usando tecnologia de análise semântica em nível de bloco e extrai uma entidade de cinco tuplas. Além disso, o ESSE extrai recursos locais e recursos globais para fazer análise semântica em nível de palavra. E então, usando esses recursos, direcionadores de tamanho e tamanhos reais de dados históricos do projeto, o modelo de estimativa de tamanho pode ser estabelecido por algoritmos de regressão. Finalmente, a ESSE pode estimar o tamanho de um novo projeto usando este modelo de estimativa de tamanho.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Global Manager	Rubén Márquez, Aurora Vizcaíno, Félix Oscar García e Antonio Manjavacas	<a href="https://dl.acm.org/doi/abs/10.1145/3372787.3390443">https://dl.acm.org/doi/abs/10.1145/3372787.3390443</a>
<b>DESCRIÇÃO</b>		
<p>O Global Manager é um jogo onde os participantes podem receber formação e adquirir soft skills importantes através de situações simuladas, especialmente no que diz respeito às situações relacionadas à comunicação, coordenação e controle de um projeto Desenvolvimento Global de Software (<i>Global Software Development – GSD</i>). Além disso, os jogadores podem treinar a resolução de problemas, sendo submetidos a um determinado nível de estresse, como em um projeto real.</p>		
<b>FINALIDADE</b>		
<p>O jogo tenta desenvolver várias habilidades em seus jogadores, proporcionando-lhes, ao mesmo tempo, uma experiência envolvente, agradável e atraente. As habilidades desenvolvidas estão relacionadas à coordenação, comunicação e controle. O objetivo deste jogo é permitir que alunos e profissionais treinem em certas habilidades necessárias para a função de gerente de projeto em ambientes distribuídos. O jogador terá que enfrentar a gestão de um projeto GSD fictício desde o seu início até a entrega do produto e tentar resolver as várias dificuldades e obstáculos que possam surgir.</p>		
<b>COMO FUNCIONA</b>		

Os jogadores primeiro escolhem o tipo de projeto que desejam gerenciar e, em seguida, configuram uma série de parâmetros iniciais que irão influenciar a evolução do projeto GSD. Em primeiro lugar, os jogadores devem escolher o número de sites, o país em que o cliente está localizado e o idioma comum para a comunicação. Por exemplo, vamos imaginar que um jogador escolhe 3 sites, que o cliente estará no Reino Unido e o inglês será o idioma principal. Em seguida, o jogador deve definir a configuração de cada site, e 3 pequenas caixas irão aparecer abaixo desta configuração. Essas caixas mostram uma definição do país em que o projeto está localizado, o tamanho da equipe, o nível médio do idioma comum e, claro, o site principal. O jogador pode, por exemplo, estabelecer que o site principal está no Reino Unido, com alto nível de inglês, que outro site está na Coreia do Sul, com alto nível de inglês, e que o último site é na Espanha, onde o nível de inglês é médio, pois supõe-se que a equipe de trabalho não falará esse idioma com fluência. Na última etapa da configuração, o tipo de comunicação que existe entre cada um dos sites e com o cliente correspondente deve ser gerenciado.

**MODELOS**

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Escritório de Gerenciamento de Projetos	Volker Gruhn e Niklas Spitzzok Von Brisinski	<a href="https://dl.acm.org/doi/abs/10.1145/3377813.3381361">https://dl.acm.org/doi/abs/10.1145/3377813.3381361</a>
<b>DESCRIÇÃO</b>		
Uma área da organização que possui profissionais especializados em gerenciamento de projetos, que pode ser encontrada em níveis estratégicos, táticos e operacionais, com o objetivo de atuar na gestão de projetos, programas e portfolio, na abertura, priorização, desenvolvimento, implantação e encerramento dos projetos.		
<b>FINALIDADE</b>		
Por meio dessa abordagem, todas as partes interessadas gerenciam quaisquer incertezas que surjam, por exemplo, devido a mudanças na tecnologia ou quaisquer mudanças no ambiente.		
<b>COMO FUNCIONA</b>		
Consideramos um monte de dados de mais de 320 projetos realizados nos últimos anos. Identificamos uma série de riscos para projetos de preço fixo e explicamos como as situações de gastos excessivos não gerenciados podem ser perigosas para clientes, bem como para fornecedores de software. Mostramos uma correlação entre o uso do PMO e a redução do gasto excessivo. Além disso, mostramos que o uso sistemático de ferramentas de PMO veio junto com uma redução ainda maior do gasto excessivo.		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Antipadrões	Petr Picha e Přemek Brada	<a href="https://dl.acm.org/doi/abs/10.1145/3361149.3361169">https://dl.acm.org/doi/abs/10.1145/3361149.3361169</a>
<b>DESCRIÇÃO</b>		
<p>Os antipadrões (APs) são “abordagens comuns para resolver problemas recorrentes que se mostram ineficazes”. Ou, dito de outra forma: "Um antipadrão é como um padrão, exceto que, em vez de uma solução, ele dá algo que se parece superficialmente com uma solução, mas não é. Assim, um método de detecção de processos e anti-padrões de PM nos dados do projeto que podem ser usados para avisar as equipes de desenvolvimento de software sobre uma potencial ameaça ao projeto, ou para conduzir estudos mais gerais sobre o impacto da ocorrência de AP no sucesso do projeto e na qualidade do produto.</p>		
<b>FINALIDADE</b>		
<p>O método tem como objetivo prever os estados do projeto sem documentos; documentos de design, qualidade do produto e relatórios de progresso, o modelo pode ser adaptável a várias metodologias de desenvolvimento.</p>		
<b>COMO FUNCIONA</b>		
<p>Os alunos são divididos em equipes de (geralmente) quatro e atribuídos a um projeto com um cliente da vida real de dentro da universidade (grupo de pesquisa, departamento, corpo docente, administração, etc.) ou de outra organização ou parceiro industrial. Eles recebem acesso a um conjunto de ferramentas ALM - por padrão, Redmine para rastreamento de projeto e um repositório git - bem como um modelo de processo específico a seguir, que é, na maior parte, um amálgama de práticas RUP e Scrum, influenciadas também por Entrega Agile Disciplinada (DAD).</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
NORPLAN	Weam M. Farid e Frank J. Mitropoulos	<a href="https://ieeexplore.ieee.org/document/6567463">https://ieeexplore.ieee.org/document/6567463</a>
<b>DESCRIÇÃO</b>		
<p>O NORPLAN utiliza métricas de gerenciamento de projetos recém-propostas, métricas de qualidade de requisitos e um algoritmo orientado a riscos para calcular uma sequência de implementação de requisitos aprimorada que pode ajudar gerentes de projeto ágeis e equipes Scrum na priorização e planejamento de Requisitos Não Funcionais (NFRs).</p>		
<b>FINALIDADE</b>		
<p>O objetivo fundamental da Metodologia NORPLAN é desenvolver uma estrutura de modelagem de NFRs simples, porém sistemática, adaptada para processos ágeis. Além disso, os objetivos incluem propor novos requisitos de qualidade e métricas de gerenciamento de projetos, desenvolver uma ferramenta de simulação, computar e visualizar uma sequência de implementação de requisitos baseada em risco.</p>		
<b>COMO FUNCIONA</b>		
<p>O método NORPLAN propõe uma taxonomia de Requisitos Ágeis que classifica os requisitos como funcionais ou não funcionais. Um Caso de Uso também faz parte da taxonomia proposta e representa soluções potenciais de NFRs. Entre os conceitos fundamentais da Metodologia NORMAP estão as recém-propostas métricas de gerenciamento de projetos, métricas de qualidade de requisitos e o algoritmo orientado a riscos proposto para calcular uma sequência de implementação de requisitos potencialmente aprimorada. A sequência de implementação é orientada pelo cálculo de riscos técnicos e de gerenciamento de projetos para todos os requisitos, que podem ser usados para ajudar equipes ágeis e gerentes de projeto no planejamento de quais requisitos seriam implementados em quais releases/sprints. O plano resultante é gerado como um cronograma em forma de árvore que inclui ciclos de liberação e sprint e os requisitos específicos em cada sprint.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Jogo Legado para Gerenciamento de Projetos	Jefferson Seide Molléri, Javier Gonzalez-Huerta e Kennet Henningsson	<a href="https://dl.acm.org/doi/abs/10.1145/3209087.3209094">https://dl.acm.org/doi/abs/10.1145/3209087.3209094</a>
<b>DESCRIÇÃO</b>		
Este modelo trata-se um projeto de desenvolvimento de software e aos diferentes modelos de processo de software. O jogo representa alguns dos desafios a serem enfrentados ao lidar com o planejamento e gestão de um projeto de software da vida real sem exigir que os alunos desenvolvam o sistema de software.		
<b>FINALIDADE</b>		
Estimular os alunos a adquirir conhecimentos (teóricos e práticos) que lhes permitam resolver desafios semelhantes aos que enfrentarão em projetos de software do mundo real.		
<b>COMO FUNCIONA</b>		
A equipe deve manter um documento descrevendo um plano de projeto de acordo com os objetivos de aprendizagem do curso. Durante a primeira rodada, a equipe apresenta uma versão inicial deste plano. Em cada rodada subsequente, uma versão atualizada é enviada. Se os alunos demonstrarem um plano de projeto sólido, que adere às boas práticas e corresponda às necessidades do projeto, eles recebem bônus para o próximo turno do jogo. Os alunos também podem ser penalizados se o plano de projeto omitir certas atividades de desenvolvimento necessárias (por exemplo, testes) ou práticas (por exemplo, retrospectiva de sprint).		

**FERRAMENTAS**

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Bluejay	ISA Group , Universidade de Sevilla	<a href="https://www.governify.io/quickstart/auditing-agile">https://www.governify.io/quickstart/auditing-agile</a>
<b>DESCRIÇÃO</b>		
<p>Bluejay é uma ferramenta composta por um conjunto de componentes que permite modelar, monitorar e auditar práticas de equipes (TP). Ele é software de código aberto que usa as APIs de várias ferramentas para coletar dados brutos, sintetizá-los em medições de práticas de equipes e apresentar painéis para auditar as práticas de equipes (TP). Ele fornece um ecossistema de ferramentas para os instrutores definirem suas próprias práticas recomendadas para seguir e rastrear a aderência das equipes a fim de aprender sobre os riscos e melhorar com o tempo. Especificamente, o framework fornece uma arquitetura de microsserviços baseada no ecossistema Governify.</p>		
<b>FINALIDADE</b>		
<p>O Bluejay fornece um ecossistema de ferramentas para os instrutores definirem suas próprias práticas e rastrear a adesão das equipes a elas, permitindo que elas possam ser melhoradas ao longo do tempo.</p>		
<b>COMO FUNCIONA</b>		
<p>Primeiro, é realizada a modelagem do processo pelo qual os instrutores definem um conjunto de TP, criando um Contrato de Prática de Equipe (TPA) para as equipes cumprirem.</p> <p>Em segundo lugar, é iniciado um processo de monitoramento para cada TPA; como parte desse processo, a Bluejay calcula o cumprimento do TPA com base nas métricas calculadas com as informações coletadas das diferentes APIs REST; as informações reais de atendimento e os valores das métricas são posteriormente renderizados em um painel gráfico disponível para instrutores e equipes.</p> <p>Terceiro, os instrutores podem usar o painel para auditar as equipes e determinar se elas aderem às práticas.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Sistema de Apoio à Decisão de Planejamento de Sprint (SPESS)	Alhejab Alhazmi e Shihong Huang	<a href="https://ieeexplore.ieee.org/document/8479063">https://ieeexplore.ieee.org/document/8479063</a>
<b>DESCRIÇÃO</b>		
O SPESS auxilia os gestores no planejamento de Sprint. O SPESS é baseado principalmente em três fatores: a competência do desenvolvedor, a antiguidade do desenvolvedor e a dependência da tarefa.		
<b>FINALIDADE</b>		
O objetivo atribuir as tarefas de cada Sprint aos desenvolvedores garantindo que cada membro da equipe contribua com o máximo de seu potencial, e o planejamento do projeto seja otimizado para o menor tempo possível.		
<b>COMO FUNCIONA</b>		
O SPESS reduz do tempo da sprint em até um dia. Isso se dá pelo fato de o SPESS ser aplicado considerando que todos os desenvolvedores têm a competência necessária para trabalhar em qualquer tarefa. A razão da diferença neste caso particular é devido ao seu método que depende da igualdade do número de tarefas atribuídas a cada desenvolvedor. Além da antiguidade dos desenvolvedores, o SPESS também leva em consideração dois fatores adicionais de planejamento de software - os níveis de competência dos desenvolvedores e a dependência das tarefas. A competência dos desenvolvedores é um dos principais fatores para determinar a quais desenvolvedores adequados são atribuídos as tarefas que correspondem às suas habilidades. O fator de dependência da tarefa é usado para determinar a ordem em que as tarefas são desenvolvidas. Entre cada iteração no Sprint, o SPESS verifica o status de trabalho dos desenvolvedores e prepara o número de desenvolvedores com os quais trabalhar para a próxima iteração. Como resultado, a ferramenta SPESS atinge seu objetivo principal de que os desenvolvedores trabalhem nas tarefas de acordo com seu nível de habilidade com o tempo de acabamento ideal.		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
MasePlanner	Roberto Morgan e Frank Maurer	<a href="https://ieeexplore.ieee.org/document/4031753">https://ieeexplore.ieee.org/document/4031753</a>
<b>DESCRIÇÃO</b>		
<p>MasePlanner é um software de planejamento ágil que oferece suporte para o planejamento baseado em cartão, facilitando a comunicação não verbal. O MasePlanner foi desenvolvido para melhorar o processo de planejamento, fornecendo às equipes um ambiente digital que oferece suporte ao gerenciamento de informações.</p>		
<b>FINALIDADE</b>		
<p>O MasePlanner tem como finalidade oferecer suporte a reuniões de planejamento, fornecendo um espaço de trabalho comum semelhante a uma mesa para criar e organizar cartões de história.</p>		
<b>COMO FUNCIONA</b>		
<p>O MasePlanner suporta a criação de cartões de história e iterações permitindo sua organização. Ele permite o planejamento baseado em cartão que fornece conhecimento espacial da localização dos cartões. O MasePlanner foi projetado como uma ferramenta de suporte e, como tal, requer que exista um canal de voz de alta qualidade para que ocorra uma verdadeira colaboração. O MasePlanner permite que os artefatos de planejamento sejam criados, editados e organizados de maneira semelhante às reuniões de planejamento em papel.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
cscw	Tobias Hildenbrand, Michael Geisser, Thomas Kude, Denis Bruch, Thomas Acker	<a href="https://ieeexplore.ieee.org/document/4606732">https://ieeexplore.ieee.org/document/4606732</a>
<b>DESCRIÇÃO</b>		
Os cscw referem-se a grupos de software que trabalham simultaneamente em um objetivo comum. A aplicação dos CSWS pode compensar a falta de proximidade física para promover a aplicabilidade do XP e do ASD em geral em ambientes distribuídos. Além disso, a abordagem auxilia na comunicação da equipe e na integração com o cliente.		
<b>FINALIDADE</b>		
Os cscw têm como finalidade apoiar a comunicação do projeto, em especial aqueles distribuídos. Os sistemas cscw visam a transferência de informações mais rápida, melhor utilização do conhecimento tácito, agilização dos processos de desenvolvimento e redução da sobrecarga administrativa.		
<b>COMO FUNCIONA</b>		
Os cscw equipam os membros do grupo com meios de comunicação e interfaces para um ambiente de trabalho compartilhado, fomentando assim a consciência do grupo, ou seja, o conhecimento sobre as atividades atuais de outros membros do grupo. As equipes são treinadas para coordenar seus processos de trabalho com mais eficiência.		

**TÉCNICAS**

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Mapping FLOW	Douglas Sanches da Cunha e Luiz Carlos	<a href="https://dl.acm.org/doi/abs/10.1145/3357419.3357426">https://dl.acm.org/doi/abs/10.1145/3357419.3357426</a>
<b>DESCRIÇÃO</b>		
<p>Mapping FLOW é uma técnica do Método FLOW para planejar e orientar a comunicação em projetos de desenvolvimento distribuído. Para atingir esses objetivos, a técnica é centrada na visualização de um mapa FLOW. Um mapa FLOW é um modelo FLOW especial (ou seja, visualização dos participantes do projeto, documentos e fluxos de informações) estendido por recursos para melhorar a conscientização em equipes distribuídas.</p>		
<b>FINALIDADE</b>		
<p>Ao usar a abordagem FLOW Mapping, a comunicação de um projeto distribuído pode ser planejada em um dia de trabalho. Usando o FLOW Mapping para planejar a comunicação para o estudo de caso, podemos mostrar que o conceito é viável. Além disso, a comunicação obteve níveis de conformidade entre 79% e 88% com relação ao que foi planejado, o que mostra que o plano em si também é viável.</p>		
<b>COMO FUNCIONA</b>		
<p>A abordagem FLOW Mapping em um projeto de programação ágil distribuída que faz parte do currículo de ciência da computação da Leibniz Universität Hannover (LUH). Este curso foi realizado em um ambiente não distribuído anteriormente, com uma forte ênfase no ensino das práticas de Programação eXtreme (XP) em um ambiente realista. Desta vez, integramos alunos da Technische Universität Clausthal (TUC) no curso para formar uma equipe distribuída. O núcleo do curso é uma semana de bloco que ajuda a implementar a prática de 40 horas semanais de XP e a criar um ambiente de projeto realista. Em preparação para esta semana, conduzimos dois workshops de 4 horas para ensinar os fundamentos do XP. Os alunos do TUC participaram da segunda oficina. Assim, os dois grupos de alunos se conheceram pessoalmente antes do início da semana de 40h XP. Também tivemos um workshop de 4h para o cliente para aprender os requisitos (distribuídos) e mais dois slots de 4h para picos técnicos (distribuídos). Havia outros quatro slots de 4h após a semana XP para finalização, análise post-mortem e discussão.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
OPENTCQ	Alhejab Alhazmi e Shihong Huang	<a href="https://ieeexplore.ieee.org/document/8479063">https://ieeexplore.ieee.org/document/8479063</a>
<b>DESCRIÇÃO</b>		
<p>A maioria dos modelos anteriores para o gerenciamento de mudanças compartilham uma estratégia comum, e o processo ou análise de cada iteração é ignorado, por isso encontrou a falha para incorporar a mudança. O formulário de solicitação de mudança (CRF) é documentado para cada vez que o requisito de mudança surge. Uma estratégia sólida de estimativa de requisitos está faltando para lidar com a mudança. No modelo proposto, o Modelo OPENTCQ, processamos cada iteração antes de sua integração ao produto como um todo. Nosso modelo é, na verdade, customizado a partir do Unified Process (UP) e do Agile Model.</p>		
<b>FINALIDADE</b>		
<p>A OPENTCQ se propõe a dimensionar o modelo tradicional para atingir o nível de agilidade. O modelo UP é fundido com o modelo em cascata para uma mudança de endereço eficiente. Cada vez que surge a iteração, o modelo proposto classificará o quantum de mudança em qualquer estágio do modelo em cascata, que é o requisito avaliado em todo o ciclo de vida.</p>		
<b>COMO FUNCIONA</b>		
<p>Um projeto denominado Automated Invoice Generation é criado na ferramenta de código aberto ‘Tiaga’ para a experimentação OPENTCQ. Coletamos dados dos três cenários, melhor, média e pior caso. Também tomamos todos os benchmarks de métricas para serem comparados. Em nossa estrutura, propusemos dois casos limítrofes e um caso médio. Melhor caso: menos tempo, baixo custo e alta qualidade. Caso Médio: High Time, Low Cost e High Quality. Pior caso: maior tempo, maior custo e baixa qualidade. Tomamos o tempo real como referência para ser comparado com o índice de tempo após o cálculo. O índice de tempo da iteração 1 é comparado com o benchmark e dá o pior resultado, portanto, em nosso framework, não precederemos tais iterações, o que ajudará a evitar o fracasso do projeto. A iteração 2 dá resultados muito satisfatórios quanto ao seu tempo em comparação com o benchmark. É o melhor case em todo o cenário. A análise de tempo da iteração 3 fornece o resultado médio para que possamos prosseguir com a iteração da solicitação de mudança também nesse caso.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Mineração de Repositórios	Valerie Trapa e Santhpur Rao	<a href="https://ieeexplore.ieee.org/document/1667585">https://ieeexplore.ieee.org/document/1667585</a>
<b>DESCRIÇÃO</b>		
A mitigação de risco leve com base na mineração de repositório e suporte de remediação harmonizada oferece a esperança de que seja possível justificar economicamente e neutralizar com eficiência os efeitos da deterioração da base de código na prática comercial.		
<b>FINALIDADE</b>		
A mineração de repositórios os custos aceitáveis, a mineração do repositório provavelmente desempenhará um papel central na mitigação de riscos leves.		
<b>COMO FUNCIONA</b>		
Para manter o custo de utilização do sistema de apoio à decisão baixo, a redução de dados deve ser mantida fácil de usar por não especialistas. Um ponto menos óbvio surge quando o gerente do projeto está tentando encontrar causas atribuíveis [7] para medidas de saúde do projeto fora de controle: as medidas originais das quais as respostas são derivadas precisam ser prontamente acessíveis. Isso implica que o sistema de suporte à decisão deve permitir o detalhamento dos dados originais ou o mais próximo possível deles.		

## FRAMEWORKS

NOME DA ABORDAGEM	FORNECEDOR/DESENVOLVEDOR	SITE DE ACESSO
Eagle	Jarno Vähäniitty e Kristian Rautiainen	<a href="http://www.agilefant.org/">http://www.agilefant.org/</a>
<b>DESCRIÇÃO</b>		
<p>Eagle é um framework de gerenciamento de backlog de código aberto aprimorada com recursos de manipulação de requisitos hierárquicos. Eagle fornece três níveis de planejamento diferentes para gerenciamento de backlog: backlogs de produto, projeto e iteração. Além dos backlogs tradicionais de lista simples, os requisitos podem ser expressos em forma de árvore, a fim de manter a rastreabilidade e a transparência.</p> <p>Além do backlog e do gerenciamento de requisitos, o Eagle contém diferentes visualizações de composição para os itens em diferentes backlogs, o que pode ser útil para organizações de desenvolvimento de pequeno e médio porte. Essas visualizações de composição incluem recursos de controle de tempo, gerenciamento de trabalho pessoal e gerenciamento de portfólio.</p>		
<b>FINALIDADE</b>		
<p>O Eagle tem como finalidade permitir que os usuários façam um planejamento mais alto, mantendo o controle de detalhes e atualizações diárias.</p>		
<b>COMO FUNCIONA</b>		
<p>Os gerentes de projeto podem acompanhar os principais marcos, bem como pequenos detalhes com a visualização dinâmica em árvore da história.</p> <p>Os usuários têm a flexibilidade de trabalhar com Scrum, Kanban, SAFe, Waterfall ou GTD. Eagle oferece uma solução escalável.</p> <p>Eagle suporta integração com JIRA e Trello.</p> <p>Os usuários podem priorizar projetos em andamento de forma eficaz usando arrastar e soltar.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Framework de Gerenciamento de Projetos para o Desenvolvimento Global de Software	Jun Lin, Han Yu, Zhiqi Shen e Chunyan Miao	<a href="https://dl.acm.org/doi/10.1145/2642937.2642959">https://dl.acm.org/doi/10.1145/2642937.2642959</a>
<b>DESCRIÇÃO</b>		
<p>O framework oferece aos usuários uma nova fonte de dados de comportamento coletados discretamente que fornecem informações sobre as características e os efeitos da tomada de decisão que não podem ser extraídos dos métodos tradicionais de pesquisa e pesquisa baseados em entrevistas. Como os membros da equipe não são colocados em uma situação formal na qual sabem que estão sendo estudados, é mais provável que os padrões de comportamento capturados por esses dados reflitam seu comportamento real.</p>		
<b>FINALIDADE</b>		
<p>O framework fornece cinco recursos principais para apoiar o gerenciamento ágil de projetos que abrangem as fases de planejamento do sprint, sprint e retrospectiva do sprint.</p>		
<b>COMO FUNCIONA</b>		
<p>Para criar perfis de usuário, o framework exige que os registrantes especifiquem seus níveis de competência autoavaliados em diferentes áreas de especialização, como familiaridade com linguagens de programação específicas, metodologias de design de sistema e ferramentas de design de interface de usuário (UI), etc. equipes e a atribuição de diferentes funções dentro de uma equipe (por exemplo, programadores e designers de interface do usuário). Os usuários do framework podem se comportar como se estivessem usando qualquer ferramenta APM sem gastar esforço adicional.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Framework de Medição de Agilidade para Gerenciamento de Projetos de Software	Valerie Trapa e Santhpur Rao	<a href="https://ieeexplore.ieee.org/document/1667585">https://ieeexplore.ieee.org/document/1667585</a>
<b>DESCRIÇÃO</b>		
O framework fornece visibilidade do andamento do projeto para dar suporte ao desenvolvimento rápido. Usando visualizações centradas na história, as equipes ágeis podem avaliar rapidamente o status, determinar dependências e identificar pontos problemáticos. O framework para relata o status realista do projeto e melhora a comunicação entre os membros da equipe distribuídos geograficamente.		
<b>FINALIDADE</b>		
O framework tem como finalidade abordar questões sistemáticas em uma ampla gama de projetos. Além disso, ele busca auxiliar na fácil identificação da complexidade funcional do projeto e a experiência dos desenvolvedores.		
<b>COMO FUNCIONA</b>		
O framework calcula a porcentagem de métricas completas do projeto. Além disso, fornece uma visão geral, por meio de relatórios, sobre o tempo de implementação de cada uma das histórias de usuários e do tempo total de implementação de todo o projeto, com base nessas histórias de usuários.		

**PRÁTICAS**

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
MEDIATION	Marc Schreiber, Bodo Kraft e Albert Zündorf	<a href="https://ieeexplore.ieee.org/document/7964364">https://ieeexplore.ieee.org/document/7964364</a>
<b>DESCRIÇÃO</b>		
<p>A abordagem é chamada Metrics Driven Research Collaboration (MEDIATION). De acordo com o nome, a definição de métricas é a tarefa mais importante em uma colaboração de pesquisa; a definição de métricas traz definições de objetivos comuns em foco. Cada métrica está vinculado aos requisitos globais do projeto. A colaboração de pesquisa deve elaborar um consenso sobre um pequeno número de métricas de alto nível que colocam os requisitos mais importantes em foco. A princípio, todos os participantes devem escolher um requisito e definir uma métrica correspondente por completo. Esta métrica deve ter as seguintes propriedades.</p>		
<b>FINALIDADE</b>		
<p>A abordagem tem como finalidade abordar questões sistemáticas em uma ampla gama de projetos. Além disso, ele busca auxiliar na fácil identificação da complexidade funcional do projeto e a experiência dos desenvolvedores.</p>		
<b>COMO FUNCIONA</b>		
<p>A abordagem garante continuamente que todos os participantes do projeto tenham um objetivo comum: o sucesso do projeto. A abordagem permite esse esclarecimento ao tornar o status, os desafios e o progresso do projeto transparentes para todos os participantes a qualquer momento. Depois de aplicar o MEDIATION o desenvolvimento e o progresso tornaram-se mais eficazes. Recursos podem ser atribuídos aos problemas que causam estagnação na conformidade métrica. Todos os participantes foram convencidos de que o produto de software desenvolvido aumenta o valor do negócio. Além disso, os pesquisadores se concentraram em desafios relevantes, resolvendo o caso de negócios. O foco nos valores de negócios por meio de métricas reduziu essencialmente o risco de falha. Por meio de nossa abordagem MEDIATION.</p>		

<b>NOME DA ABORDAGEM</b>	<b>FORNECEDOR/DESENVOLVEDOR</b>	<b>SITE DE ACESSO</b>
Fireteam	He Zhang, Huang Huang, Dong Shao e Xin Huang	<a href="https://dl.acm.org/doi/10.1145/3368089.3417068">https://dl.acm.org/doi/10.1145/3368089.3417068</a>
<b>DESCRIÇÃO</b>		
<p>A fim de reduzir as despesas gerais de gerenciamento de projetos e melhorar a produtividade, uma empresa global de tecnologia da informação e comunicação institucionaliza uma prática de equipe pequena em toda a organização, chamados fireteams, para resolver os problemas decorrentes de aspectos humanos e sociais, como amizade, talento, habilidade e comunicações.</p>		
<b>FINALIDADE</b>		
<p>O objetivo do fireteam é reduzir a sobrecarga de comunicação e gerenciamento, maior agilidade e simultaneidade e melhor habilidade pessoal</p>		
<b>COMO FUNCIONA</b>		
<p>Seguindo as diretrizes para o projeto de pesquisa [6], uma abordagem de métodos mistos foi adotada neste estudo, que combina análise de arquivo, entrevista e levantamento, para alcançar as triangulações de método de pesquisa e fonte de dados. Nossas descobertas vêm de três fontes de dados: os documentos arquivados (por exemplo, guia e dados dos funcionários), duas rodadas de entrevistas com as equipes selecionadas e uma pesquisa em seis institutos da empresa. Os dados qualitativos para análise vêm de entrevistas, da pesquisa e do guia do Fireteam, enquanto alguns dados quantitativos da pesquisa e do banco de dados de funcionários no arquivo da empresa. No momento do estudo, todas as atividades de pesquisa eram realizadas com a aprovação e apoio da empresa. Aprendemos sobre a Fireteam pela leitura minuciosa do guia da Fireteam que foi desenvolvido pelos consultores sobre métodos ágeis na empresa, bem como analisamos seu banco de dados de funcionários. As duas rodadas de entrevistas exploraram ainda mais a implementação de Fireteams com detalhes específicos e experiências de membros da equipe. Eles foram seguidos por uma pesquisa confirmatória sobre as descobertas primárias com feedbacks das equipes de fogo de toda a empresa.</p>		