



UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS DE CASTANHAL
FACULDADE DE COMPUTAÇÃO

**ANÁLISE SISTEMÁTICA DE APLICAÇÃO DE MÉTODOS ÁGEIS EM
AMBIENTES EMPRESARIAIS E INSTITUCIONAIS.**

ADRIELLE VERAS DE ALMEIDA

CASTANHAL – PARÁ – BRASIL
MARÇO / 2019

ADRIELLE VERAS DE ALMEIDA

ANÁLISE SISTEMÁTICA DE APLICAÇÃO DE MÉTODOS ÁGEIS EM AMBIENTES EMPRESARIAIS E INSTITUCIONAIS.

Trabalho de conclusão de curso submetido a análise da banca examinadora do curso de Bacharelado em Sistemas de Informação, como requisito para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Tássio Costa de Carvalho

Coorientador: Prof. Me. Fabrício Wickey da Silva Garcia

CASTANHAL – PARÁ – BRASIL

MARÇO / 2019

**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)**

D278a de Almeida, Adrielle Veras
Análise Sistemática de Aplicação de Métodos Ágeis em Ambientes Empresarias e Institucionais. / Adrielle Veras de Almeida. — 2019.
72 f. : il. color.

Orientador(a): Prof. Dr. Tássio Costa de Carvalho Coorientador(a): Prof. Me. Fabrício Wickey da Silva Garcia Trabalho de Conclusão de Curso (Graduação) - Faculdade de Sistemas de Informação, Campus Universitário de Castanhal, Universidade Federal do Pará, Castanhal, 2019.

1. Softwares. Metodologias de Desenvolvimento de Softwares. Metodologia Ágeis. Metodologias Tradicionais. Processo de Desenvolvimento. I. Título.

CDD 004.072

ADRIELLE VERAS DE ALMEIDA

ANÁLISE SISTEMÁTICA DE APLICAÇÃO DE MÉTODOS ÁGEIS EM AMBIENTES EMPRESARIAIS E INSTITUCIONAIS.

Trabalho de conclusão de curso submetido a análise da banca examinadora do curso de Bacharelado em Sistemas de Informação, como requisito para a obtenção do grau de Bacharelado em Sistemas de Informação.

APROVADA EM: 28 /03/2019

BANCA EXAMINADORA:

Prof. Dr. Tássio Costa de Carvalho
(Orientador – FACOMP/UFPA)

Prof. Me. Fabrício Wickey da Silva Garcia
(Coorientador – CCP/UFRA)

Prof. Dr. Igor Ruiz Gomes
(Avaliador Interno – FACOMP/UFPA)

Prof. Dr. José Jailton Henrique Ferreira Júnior
(Avaliador Interno – FACOMP/UFPA)

Profa. Dra. Yomara Pinheiro Pires
(Avaliador Interno – FACOMP/UFPA)

Dedico esse trabalho em especial a minha mãe e a minha vó (Alaide), que sempre me apoiaram em todos os meus momentos.

Não é o que eu sou por dentro e sim, o que eu faço é que me define.

Bruce Wayne / Batman

SUMÁRIO

LISTA DE ILUSTRAÇÕES	I
LISTA DE TABELAS	II
LISTA DE QUADROS	III
LISTA DE ABREVIATURAS E SIGLAS	IV
RESUMO	V
ABSTRACT	VI
1. INTRODUÇÃO	15
1.1 CONSIDERAÇÕES INICIAIS	15
1.2 CONTEXTUALIZAÇÃO	15
1.3 JUSTIFICATIVA E MOTIVAÇÃO	17
1.4 OBJETIVO GERAL	20
1.4.1 OBJETIVOS ESPECÍFICOS	20
1.5 CONSIDERAÇÕES FINAIS E ORGANIZAÇÃO	20
2. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	21
2.1 CONSIDERAÇÕES INICIAIS	21
2.2 ENGENHARIA DE SOFTWARE	21
2.3 METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE	22
2.3.1 METODOLOGIAS TRADICIONAIS DE SOFTWARE	24
2.3.2 METODOLOGIAS DE DESENVOLVIMENTO ÁGIL	24
2.4 CONSIDERAÇÕES FINAIS	30
3. METODOLOGIA DE PESQUISA	31
3.1 CONSIDERAÇÕES INICIAIS	32
3.2 PROCEDIMENTOS DE PESQUISA	32
3.3 PROCEDIMENTOS DE COLETA DE DADOS	34
3.3.1 ENGENHO DE BUSCA	33
3.3.2 ESTRATÉGIA DE BUSCA	34
3.3.3 EXPRESSÃO DE BUSCA	34
3.3.4 STRINGS DE BUSCA	35
3.4 VALIDAÇÃO	36
3.4.1 CRITÉRIOS DE SELEÇÃO	36
3.5 DOCUMENTAÇÃO E SELEÇÃO DE ESTUDOS	37

3.5.1 EXTRAÇÃO	38
3.5.2 SUMARIZAÇÃO DOS RESULTADOS	39
3.5.3 ANÁLISE DOS RESULTADOS	39
3.6 CONSIDERAÇÕES FINAIS	40
4. ANÁLISE E DISCUSSÃO DOS CONTRIBUTOS	40
4.1 CONSIDERAÇÕES INICIAIS	40
4.2 CORRELATOS SELECIONADOS	41
4.3 ÁREAS DE CONHECIMENTO	45
4.4 JUSTIFICATIVA DA METODOLOGIA ÁGIL	49
4.5 ANÁLISE E DADOS DA PESQUISA	53
4.6 COMPÊNDIO DOS CONTRIBUTOS	58
4.7 CONSIDERAÇÕES FINAIS	61
5. CONSIDERAÇÕES FINAIS	63
5.1 DIFICULDADES ENCONTRADAS	65
5.2 TRABALHOS FUTUROS	66
REFERÊNCIAS BIBLIOGRÁFICAS	67

LISTA DE ILUSTRAÇÕES

FIGURA 1. SOMA DAS RESPOSTAS DA PESQUISA EM METODOLOGIAS ÁGEIS.....	18
FIGURA 2. RELATÓRIO ANUAL DE METODOLOGIAS ÁGEIS NO MUNDO	19
FIGURA 3. MODELO CASCATA	25
FIGURA 4. PROCESSO DO SCRUM.....	27
FIGURA 5. PRÁTICAS XP	28
FIGURA 6. FAMÍLIA CRYSTAL	29
FIGURA 7. QUADRO KANBAN	30
FIGURA 8. ROTEIRO DO PROCESSO METODOLÓGICO.....	33
FIGURA 9. ETAPA DO PROCESSO DE SELEÇÃO DE ESTUDOS.....	38
FIGURA 10. RESUMO DAS CARACTERÍSTICAS DA METODOLOGIA DE PESQUISA UTILIZADA.....	41
FIGURA 11. RAZÕES E MOTIVOS PARA A ADOÇÃO DE UMA METODOLOGIA ÁGIL.	50
FIGURA 12. QUANTIDADE DE SEMANAS PARA UMA ITERAÇÃO.....	56
FIGURA 13. ADAPTAÇÕES DO SCRUM	58
FIGURA 14. PRINCIPAIS DIFICULDADES/DESAFIOS ENCONTRADOS NOS ARTIGOS SELECIONADOS.....	59

LISTA DE TABELAS

TABELA 1. TABELA DO CRITÉRIO INCLUSÃO (CI)	37
TABELA 2. TABELA DO CRITÉRIO DE EXCLUSÃO (CE).....	37
TABELA 3. RESULTADO DA COLETA DE DADOS	38
TABELA 4. QUANTITATIVO/ANO DAS PUBLICAÇÕES SELECIONADAS.....	41
TABELA 5. QUANTITATIVO DE ARTIGOS POR ÁREA DE CONHECIMENTO.....	45
TABELA 6. VISÃO GERAL DA SPRINT	57

LISTA DE QUADROS

QUADRO 1. PALAVRAS-CHAVE USADAS NAS STRINGS	35
QUADRO 2. STRINGS UTILIZADAS NA BASE DA SCOPUS	36

LISTA DE ABREVIATURAS E SIGLAS

APM	Agile Project Management
AR	Action Research
ASD	Agile Software Development
CE	Critério de Exclusão
CI	Critério de Inclusão
ERAU	Embry-Riddle Aeronautical University
FOSS	Free and Open Source Software
GAP	Gerenciamento Ágil Projetos
IOS	iPhone Operating System
JHU/APL	Johns Hopkins University Applied Physics Laboratory
KBS	Knowledge-Based System
NASA	National Aeronautics and Space Administration
OTAN	North Atlantic Treaty Organization
PMI	Project Management Institute
PMP	Project Management Professional
PMBOK	Project Management Body of Knowledge
RSL	Systematic Review of Literature
SES	Space Exploration Sector
SIS	Simulation Software Architecture
SPP	Solar Probe Plus
UFPA	Universidade Federal do Pará
UFSCAR	Universidade Federal de São Carlos
UFRGS	Universidade Federal do Rio Grande do Sul
XP	Extreme Programming
WIP	Work in Progress

RESUMO

Atualmente, os softwares são praticamente indispensáveis, uma vez que foram desenvolvidos para agilizar serviços que, outrora, eram realizados manualmente. Em decorrência disso surgiram também metodologias de desenvolvimento de software diversificadas, com a pretensão de tornar o processo de produção de sistemas organizado e padronizado e oferecer aos clientes um produto de maior qualidade. Esta monografia versa sobre a utilização de metodologias ágeis para o desenvolvimento de softwares e tem como objetivo realizar uma análise sobre os contributos dessa metodologia em organizações e instituições, por meio da revisão bibliográfica combinada a algumas práticas da revisão sistemática da literatura, aplicada a estudos produzidos sobre o tema, no período de 2010 a 2018. Para a realização desta monografia foram mapeados 570 trabalhos, disseminados em base de dados digitais que, posteriormente, foram filtrados a partir de critérios pré-definidos para a etapa de análise e discussão dos dados. A partir da pesquisa realizada, foi constatado a necessidade de conhecer as metodologias ágeis antes de adotá-las, a fim de verificar qual alternativa/opção disponível melhor se adequa a organização de determinado projeto, para assim atingir o objetivo desejado. Desse modo, os resultados obtidos nesta monografia validam das metodologias ágeis para acelerar o processo de desenvolvimento de softwares, por meio de aspectos como: planejamento direcionado, objetividade e produção em curto prazo de produtos que atendam as reais necessidades dos clientes.

Palavra-chave: Softwares. Metodologias de Desenvolvimento de Softwares. Metodologia Ágeis. Metodologias Tradicionais. Processo de Desenvolvimento.

ABSTRACT

Currently, the software is practically indispensable, since it was developed to speed up services that, once, were done manually. As a result of this, diverse software development methodologies have emerged, with the aim of making the system production process organized and standardized and offering customers a higher quality product. This monograph deals with the use of agile methodologies for the development of software and has the objective of analyzing the contributions of this methodology in organizations and institutions, through the bibliographic review combined with some practices of the systematic review of the literature, applied to studies produced on the subject, in the period from 2010 to 2018. For the accomplishment of this monograph 570 papers were mapped, disseminated in digital databases that were later filtered from predefined criteria for the analysis and discussion of the data. Based on the research, it was verified the need to know the agile methodologies before adopting them, in order to verify which alternative / option is best suited to the organization of a given project, in order to achieve the desired objective. Thus, the results obtained in this monograph validate the agile methodologies to accelerate the software development process, through aspects such as: targeted planning, objectivity and short-term production of products takes care the real customer needs.

Keywords: Softwares. Software Development Methodologies. Agile Methodology. Traditional Methodologies. Development.

1. INTRODUÇÃO

1.1 Considerações Iniciais

Esta seção apresenta a contextualização das metodologias de desenvolvimento de *softwares*, a problemática, justificativa, objetivos geral e específicos do desenvolvimento dessa monografia, assim como sua estrutura.

1.2 Contextualização

A demanda por informatização e melhorias no desenvolvimento tecnológico, é crescente na medida que *softwares* mostram-se muito mais eficientes devido a sua capacidade de gerir, organizar e disponibilizar informações em um menor espaço de tempo. Esta procura surgiu da necessidade de organizar e agilizar serviços que anteriormente realizados/executados utilizando métodos tradicionais.

O desenvolvimento de *softwares* pode ser considerado uma atividade complexa, pois envolve múltiplos fatores a serem observados durante o processo de criação, dentre os quais: mercado; usuários, que se caracterizam como um público exigente, posto que constantemente apresentam novas necessidades; linguagens de programação; equipamentos; planejamento; estimativa de riscos e prazos bem definidos. Estes aspectos geram a necessidade do uso de metodologias de desenvolvimento para acelerar o processo.

A maior parte dos projetos de desenvolvimento de *software* podem restringir-se apenas a prática de codificar e corrigir, sendo desenvolvidos sem planejamento ou precedidos por uma fase de organização de design do sistema, ocasionando problemas entre os quais destaca-se, uma grande quantidade de erros que só são descobertos quando o produto já está em fase de conclusão e disponível para testes, dentre outros aspectos, nos quais, geram implicações como a extrapolação dos prazos iniciais estabelecidos para a entrega do produto. Como um meio para corrigir este problema foi introduzido a noção de metodologia, na perspectiva de que o desenvolvimento de software pudesse ser mais eficiente e previsível (FOWLER, 2005).

Por meio da implementação de melhorias no processo de desenvolvimento de *softwares*, surgiram as mais conhecidas metodologias, a tradicional que, segundo Soares (2004), é bem aceita quando os acontecimentos no decorrer do projeto são esperados, isto porque, este aspecto favorece o processo de orientação e planejamento para seguir exatamente o que foi proposto, considerando que os requisitos iniciais dos sistemas são estáveis e os futuros são previsíveis, por esta razão essa metodologia é caracterizada como preditiva.

Em outra perspectiva, têm-se as metodologias ágeis que, de acordo com Ribeiro e Ribeiro (2015), costumam ser adotadas em projetos orientados a valor, em que geralmente a imprevisibilidade é presente, motivando assim um considerável número de mudanças no decorrer do projeto, em que o escopo não é totalmente definido desde o início, o que possibilita a inserção de novos recursos, sem grandes impasses.

A busca pelo diferencial em um mercado oscilante, faz com que as metodologias ágeis sejam uma alternativa para as organizações serem mais competitivas, pelo fato de produzir um software com qualidade, realizar entregas a curto prazo e pela forma como lidam com as mudanças. (ALBINO et al, 2014).

Como enfatiza Highsmith (2002), as organizações que adotam metodologias ágeis, veem as mudanças que oriundas durante processo como uma oportunidade de destaque no mercado, visto que os concorrentes que não acomodam essas mudanças ou não conseguem responder adequadamente, podem vir a ter um desempenho não satisfatório, perdendo clientes como consequência propiciando uma migração para uma organização considerada ágil.

As práticas apontadas por metodologias, sejam elas tradicionais ou ágeis, são eficientes para desenvolver sistemas, desde que usadas corretamente e direcionadas para o projeto que melhor atende as suas características, partindo do pressuposto que conhecer o que cada metodologia tem a oferecer ao tipo de projeto que será desenvolvido é um passo muito importante a ser considerado na escolha, para que se tenha um produto de qualidade.

1.3 Justificativa e Motivação

Os projetos têm a natureza de serem temporários, indicam ter início e fim e apresentam objetivos como, a criação de um produto, serviços ou resultados exclusivos, a serem alcançados dentro dos prazos previsto, o que requer planejamento e gerenciamento (PMI, 2013).

Durante a década de 40, o gerenciamento funcionava de uma forma pouco operacional e de baixa responsabilidade geral, onde cada gerente ao finalizar a tarefa da sua área organizacional, deixava a responsabilidade em aberto para que outra pessoa pudesse dar continuidade, mas essa forma de gerenciar apresentou problemas, principalmente, a falha na comunicação, posto que após a realização da tarefa, o responsável abandonava o projeto e o cliente tinha dificuldades no acesso à informação, havendo desperdício de tempo para filtrar informações relevantes.

Dessa forma, no início dos anos 60, as empresas e organizações começaram a perceber a necessidade de organizar o trabalho e entender a importância da comunicação entre os envolvidos, padronização do processo e melhores práticas. (KERZNER, 2017).

Posteriormente surgiram conceitos, técnicas, ferramentas, práticas, como meios para orientar e auxiliar o gerenciamento em projetos, entre eles, destacam-se “O PMI (Project Management Institute – Instituto de Gerenciamento de Projeto), que possui como objetivo, promover o desenvolvimento da gerência de projetos, divulgar as tecnologias e os métodos de treinamento para o gerenciamento” (CODAS, 1987).

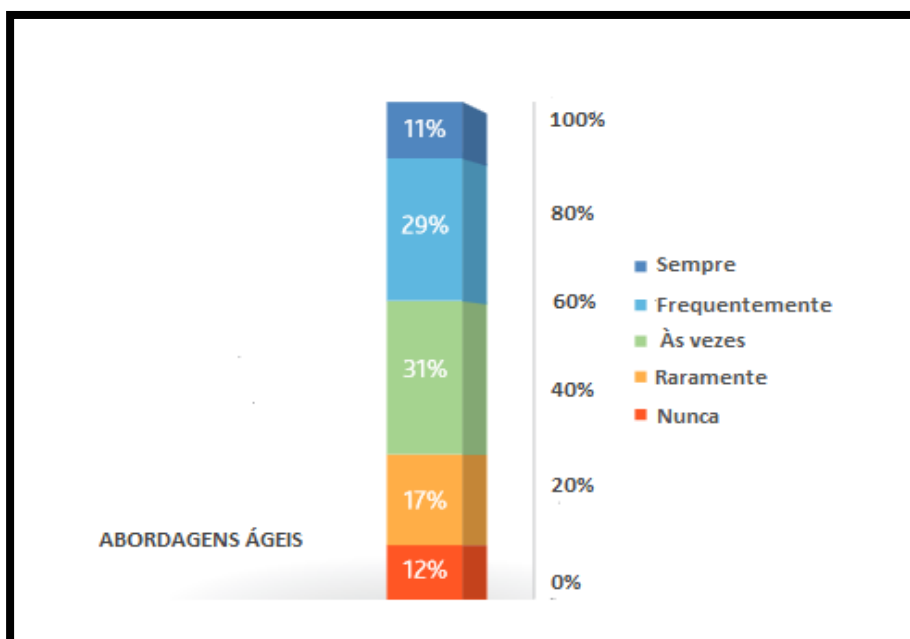
O PMBOK (Project Management Body of Knowledge – Corpo de Conhecimento em Gerência de Projeto), é um guia elaborado pelo PMI, que contém as melhores práticas para gerenciamento de projetos; O PMP (Project Management Professional – Projeto de Gerenciamento Profissional), é uma certificação reconhecida, que pode-se ser concedido mediante a aprovação em um exame realizado pelo PMI, no qual confere o título de gerente de projetos.

Surgiu também o APM/GAP (*Agile Project Management*/Gerenciamento Ágil de Projeto), que segundo Amaral et. al (2011), o objetivo é a entrega de um produto de valor em um ambiente oscilante, o que condiz que o processo de gerenciamento seja mais simples, flexível e iterativo, baseado em um conjunto de valores, princípios e

práticas das metodologias ágeis, de forma a obter resultados satisfatório tanto no custo, tempo e qualidade.

A prática de gerenciamento de projetos tem contribuído sobremaneira para que mais projetos tenham êxito, atingindo metas e intenções de negócios, bem como favorecido que estes sejam concluídos dentro do orçamento previsto. De acordo com uma pesquisa realizada no ano de 2017, pela Pulse of The Profession, com a participação de 3.234 profissionais ligados à área de gerenciamento de projetos, é cada vez maior o número de organizações que utilizam as metodologias ágeis como técnica de gerenciamento.

Figura 1. Soma das respostas da pesquisa em metodologias ágeis



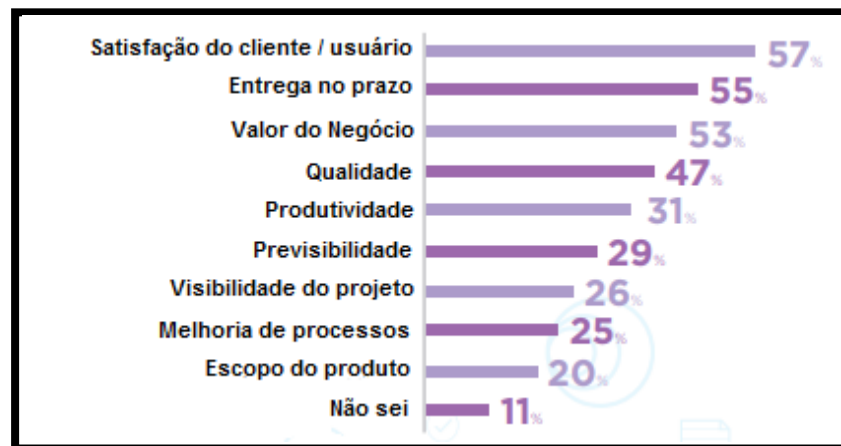
Fonte: Adaptado de (PULSE OF THE PROFESSION, 2018)

Desse modo, a pesquisa revelou que o uso de metodologias ágeis configura uma tendência crescente entre as organizações a que estes profissionais estão ligados. Este dado pode ser constatado por meio da soma das respostas em escala, às vezes, frequentemente ou sempre, que totaliza 71% das respostas, conforme ilustrou a figura 1. (PULSE OF THE PROFESSION, 2017).

A utilização de metodologias ágeis é algo recente na área de desenvolvedores e empresas de *software*. Entender o processo de adoção dessa metodologia é motivadora, por apresentar flexibilidade perante as mudanças e sem muitas dificuldades.

O valor comercial, a entrega pontual de projetos e a satisfação do cliente/usuário permaneceram como as três principais medidas do sucesso da iniciativa ágil. Os dados vêm comprovando que, assim como em anos anteriores, o grau de satisfação do cliente permanece em primeiro lugar e cresceu de 44% para 57%, no ano de 2017. Outro dado que aponta para o êxito na utilização de metodologias ágeis é que 74% das empresas pesquisadas em 2017, afirmaram que mais da metade de seus projetos ágeis obtiveram sucesso absoluto. A Figura 2 resume alguns destes resultados (AGILE REPORT, 2018).

Figura 2. Relatório anual de metodologias ágeis no mundo



Fonte: Adaptado do (AGILE REPORT, 2018)

“Apesar do interesse crescente no uso das metodologias ágeis, ainda faltam casos de sucesso de seu uso em projetos grandes e críticos. Quanto mais organizações usarem as metodologias ágeis, melhores serão os resultados empíricos em termos de vantagens, desvantagens, riscos e procedimentos para sua adoção nas organizações. Mesmo assim, os resultados iniciais em termos de qualidade, confiança, datas de entrega e custo são promissores” (SOARES, 2004, pag. 06).

Diversos profissionais, instituições e empresas de desenvolvimento de *software* possuem conflitos no momento de escolha das metodologias na produção de programas, sítios eletrônicos, aplicativos, entre outros serviços voltados à tecnologia da

informação e, sendo assim, é frequente a existência de dúvidas acerca da seleção entre métodos tradicionais e ágeis, sobretudo relacionadas a tempo, custos, documentação, o que implica conhecer sobre cada metodologia e constatar a melhor se adequa ao tipo de projeto.

Dessa forma, a presente pesquisa acadêmica pretende buscar, dentre a literatura, casos de organizações que utilizaram métodos ágeis, como contributos dessa metodologia de produção, a fim de esclarecer as especificidades e possibilidades promovidas por meio dos resultados relatados em produções científicas.

1.4 Objetivo Geral

O principal objetivo desta monografia é estudar por meio da análise bibliográfica combinada com RSL (Revisão Sistemática da Literatura), os principais contributos da utilização de engenharia de *software* no processo de melhoria do projeto final, especificamente através de metodologias ágeis, a fim de fomentar uma análise de diversos projetos que utilizaram métodos ágeis, em diferentes épocas e campos, observando como se comportam diante dessas especificidades.

1.4.1 Objetivos Específicos

- Apresentar uma revisão geral de engenharia de *software*;
- Apresentar um levantamento sobre processos de desenvolvimento de *software*;
- Apresentar as principais metodologias de desenvolvimento de *software*;
- Enfatizar a importância de trabalhos de pesquisa utilizando RSL;
- Utilizar base digital para a coleta dos resultados.
- Promover uma discussão documentada sobre metodologias tradicionais e ágeis;
- Analisar os métodos ágeis sobre trabalhos e estudos de caso;
- Analisar os contributos quanto prática de desenvolvimento de *software*.

1.5 Considerações Finais e Organização

O presente capítulo apresentou uma contextualização do projeto desta monografia, introduzindo o conteúdo para um melhor entendimento levando em consideração algumas referências importantes para a execução da mesma. Este capítulo apresentou também a motivação e justificativa para a execução desta monografia o, juntamente aos principais objetivos.

O Capítulo 2 apresentará o referencial teórico dos principais assuntos correlacionados a temática desta monografia.

O Capítulo 3 abordará os métodos de pesquisa aplicados para a consecução da coleta de dados e a geração dos resultados.

O Capítulo 4 descreverá uma análise aprofundada dos resultados pertinentes à pesquisa, e com isso, serão explicitadas as características de quatro métodos de acordo a natureza do projeto e tipo de organização na qual foi realizada o projeto.

No Capítulo 5 serão descritas as principais contribuições das análises dos resultados obtidos, as dificuldades encontradas para realização da pesquisa e recomendações para trabalhos futuros.

2. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

2.1 Considerações Iniciais

Esta seção descreve os principais conceitos que colaboram para um melhor entendimento, sobre o objeto de estudo desta monografia, trazendo como referências autores de renome na área estudada dentre eles: Sommerville e Pressman. Iniciando por um breve histórico sobre engenharia de *software* e abordando também tópicos como metodologias de desenvolvimento de *software* e alguns métodos, tanto tradicionais, quanto ágeis.

2.2 Engenharia de Software

O conceito de engenharia de *software* surgiu em 1960 em uma conferência da OTAN (*North Atlantic Treaty Organization* – Organização do Tratado do Atlântico Norte), considerada a primeira conferência da área (RANDELL, 1996), onde discutia-se sobre alguns problemas mais decorrentes no desenvolvimento de *software* e que afetavam diretamente na qualidade, sendo eles, atraso na entrega, extrapolação no orçamento, os produtos não eram confiáveis e geralmente não apresentavam as funcionalidades que o usuário precisava.

Este cenário corroborou para pensar e tornar o desenvolvimento de *software* em um processo estruturado, planejado e padronizado, que conseguisse mitigar esses problemas, de forma a entregar um software que atendesse a reais necessidades do cliente e que resultasse em um retorno financeiro compensador (MAINART E SANTOS, 2010).

De acordo com Pressman e Maxin (2016) os objetivos da prática da engenharia de *software* devem focar na entrega de um produto de qualidade dentro do prazo definido e contendo as funcionalidades requeridas pelo cliente. Para atingir esses objetivos é necessário adotar procedimentos como gerenciamento de projetos de softwares, desenvolvimentos de ferramentas, metodologias, práticas e processos técnicos de desenvolvimento, que por sua vez, auxiliam e repercutem em melhorias contínuas na produção de sistemas.

Sommerville (2011) acrescenta ainda, que os motivos mais pertinentes para a engenharia de *software* receber cada vez mais visibilidade, relacionam-se, sobretudo, a

necessidade de produzir sistemas confiáveis de forma rápida, ante a dependência das organizações na obtenção de softwares avançados.

Conforme Santos (2004), algumas organizações tendem a criar o seu próprio modelo de processo ou adaptar algum à sua realidade, no entanto, pode-se encontrar na literatura da engenharia de *software* vários processos, entre eles, as metodologias tradicionais, que são orientadas ao planejamento e as metodologias ágeis, que procuram responder as mudanças que surgem durante o desenvolvimento.

A engenharia de *software* oferece uma gama de artefatos, processos e métodos a serem adotados conforme as características específicas de cada projeto, cuja utilização dependerá, sobretudo, da capacidade de adequação destas a um determinado contexto. Esta flexibilidade em relação aos procedimentos adotados na execução dos projetos é interessante, na medida em que uma abordagem mais criativa e menos formal pode ser mais eficiente e adaptar-se melhor a natureza de alguns projetos, contudo, a depender de cada caso, para outros, esta mesma abordagem não apresentará as mesmas vantagens (SOMMERVILLE, 2011).

2.3 Metodologias de Desenvolvimento de Software

O desenvolvimento de *software* deixou de ser apenas o ato de programar, e sim envolver outras premissas. Atualmente os projetos demandam de uma metodologia para acelerar o processo de desenvolvimento de software e obter um produto de qualidade, o que torna um fator competitivo no mercado (SOUZA, 2014).

Alguns fatores são decorrentes da ausência de sistematização na produção de *software*, como a extrapolação no prazo, visto que sem um planejamento adequado é difícil ter uma visão geral do tempo necessário e quais processos serão utilizados para o desenvolvimento, seja a longo ou a curto prazo, impactando diretamente na qualidade do *software* e nos custos predefinidos (SOARES, 2004).

De acordo com Lima (2007), uma metodologia de desenvolvimento torna-se essencial pelo fato de ser uma abordagem estruturada, que promove um maior grau de organização, por meio do cumprimento de procedimentos preestabelecidos. A disposição dependendo da metodologia pode ser focada no processo ou em pessoas.

A partir desse cenário, surgiram diferentes metodologias e entre as mais conhecidas e usadas encontram-se a tradicional, com o enfoque em processos, criando uma vasta documentação e a ágil com o enfoque em pessoas, utilizando uma quantidade mínima de documentação e é iterativa, sendo que ambas metodologias mantêm o foco na qualidade (UTIDA, 2012).

2.3.1 Metodologias Tradicionais de Software

As chamadas metodologias tradicionais surgiram em um cenário totalmente diferente de hoje, o que justifica a planificação antes do *software* ser desenvolvido e uma documentação extensa, pois alguns fatores como: fazer alterações e correções era custoso, além de gerar atraso na entrega do produto, uma vez que o acesso aos computadores eram limitados e não existiam modernas ferramentas de apoio ao desenvolvimento do software (SOARES, 2004).

Conforme descrito em Tomás (2009), este modelo não é maleável quando se trata em aceitar mudanças nos requisitos nas fases posteriores ao planejamento devido à possibilidade de regressão na fase de planificação, o que pode gerar atraso e resultar uma mudança inesperada no desenvolvimento. Segundo Filho (2008), essas abordagens ainda prezam por uma documentação extensiva, detalhando tudo o que é produzido, sendo o seu foco direcionado para processos.

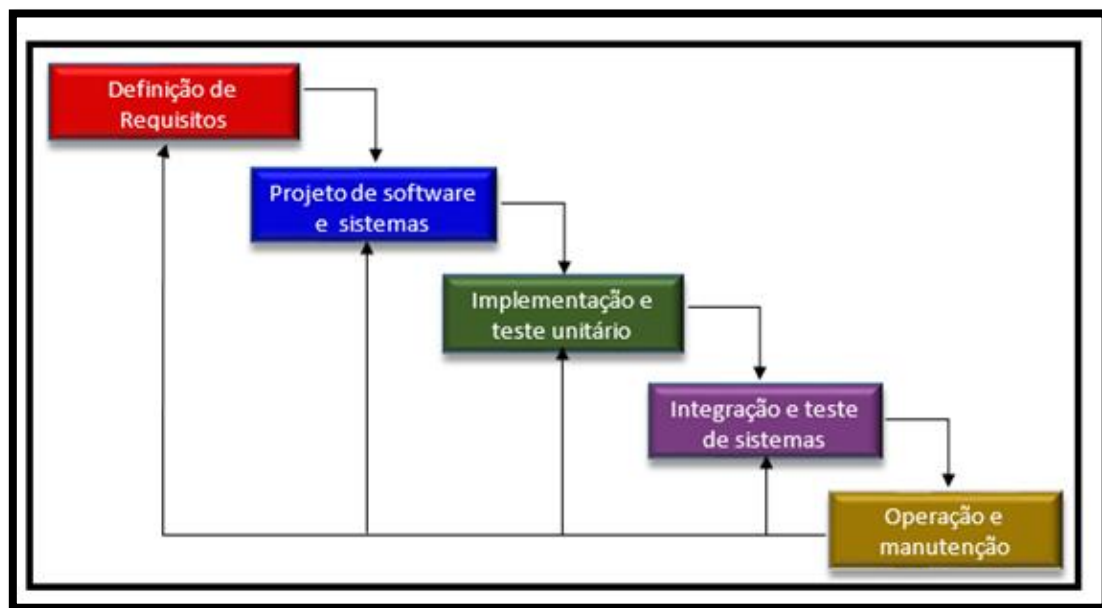
As metodologias tradicionais são completamente baseadas na predição, considerando que a forma a qual o projeto foi planejado inicialmente deve ser mantida até o final, pois as etapas dependem uma da outra, tendo por consequência a entrega do *software* somente após a finalização de todo o processo. A utilização desta metodologia restringe as possibilidades de alteração durante o desenvolvimento do projeto e são mais indicadas para casos que não demandam mudanças, isto é, em que há certa previsibilidade nos requisitos.

2.3.1.1 Modelo Cascata

Em meados dos anos 1970, surgiu um dos primeiros processos que ficou conhecido como cascata por ser considerado preditivo, é indicado para projetos onde os requisitos são bem compreendidos e nos quais a probabilidade de surgimento de mudanças seja mínima, dado que, trata-se de uma base sequencial, constituída por sete fases, onde as tarefas seguem um fluxo, no qual uma fase só pode iniciar após a

finalização da fase anterior. Se, durante esse processo, o cliente vier a solicitar alguma mudança no sistema, tais alterações acarretarão em alguns transtornos nesse modelo, na medida em que será necessário rever o planejamento do projeto, agregando mais custos, por se tratar de um método rígido (MARQUES, 2012). A Figura 3 a seguir esboça o modelo descrito.

Figura 3. Modelo cascata



Fonte: Adaptado do (SOMMERVILLE, 2011)

Em suma, se algum requisito ou processo não for bem definido na fase de planejamento, realizar alterações no decorrer do desenvolvimento, pois esse modelo não dispõe de um cronograma para revisões e nem prevê a flexibilidade para possíveis mudanças. No entanto, possibilita um controle gerencial, posto que cada fase é muito bem definida e fornece uma visão precisa do que será realizado (OLIVEIRA, 2018).

2.3.2 Metodologias de Desenvolvimento Ágil

A maioria dos projetos até o ano de 2001 eram baseados na metodologia tradicional de desenvolvimento de *software*, cujo enfoque era no planejamento que subsidia todo o desenvolvimento posterior (TOMAS, 2009).

Ainda no ano de 2001, ocorreu um grande movimento por parte de especialistas e analistas desta área o qual ficou conhecido como o manifesto ágil. Este manifesto teve como principal marco, uma declaração elaborada por 17 especialistas, dispendo de valores e princípios voltados a melhoria e agilidade do processo de desenvolvimento, onde se propôs uma metodologia alternativa à tradicional, conhecida como metodologia ágil. (OLIVEIRA, 2018). Durante o manifesto, ficou acordado que algumas áreas, princípios e campos deveriam ser melhor abordados e/ou valorizados. São eles:

- Indivíduos e interações ao invés de processos e ferramentas;
- *Software* executável ao invés de documentação;
- Colaboração do cliente ao invés de negociação de contratos;
- Respostas rápidas a mudanças ao invés de seguir planos.

Esse conjunto de princípios concentra-se na colaboração do cliente, no valor dos indivíduos e na adaptação às mudanças, o que corrobora para criação de produtos de qualidade e que melhor atenda à realidade do cliente.

Para Lima (2015), as metodologias ágeis prezam, prioritariamente, por um *software* funcional ao invés da produção de uma documentação extensa. O ponto chave é o excessivo gasto de tempo produzindo documentação que não agrega valor nenhum ao projeto e que esse esforço deve ser direcionado para o desenvolvimento do *software*, ou seja, deve-se evitar qualquer documentação que não será usada.

Oliveira (2018) enfatiza que o manifesto ágil não abandona processos, ferramentas, documentação e nem planejamento, mas mostra que eles têm uma importância secundária reforçando a ideia de que seja produzido e utilizado somente o necessário.

Nas abordagens ágeis o desenvolvimento de *software* acontece de forma incremental, ou seja, especificações vão sendo inseridas gradativamente, a partir de iterações de curta duração, que consiste em liberação de pequenas partes funcionais do sistema ao cliente, o que, conseqüentemente impõe o caráter adaptativo (ALVES, 2012).

Existem diversos métodos ágeis, entre eles: *Extreme Programming* (XP), *Scrum*, *Crystal*, *Feature Driven Development* (FDD), *Dynamic Systems Development Method* (DSDM), *Kanban*. Em relação a estes métodos, as pesquisas na literatura têm apontado para ganhos de produtividade nos mais diversos tipos de projetos, devido a forma como lidam com imprevisibilidades e têm sido usados tanto no gerenciamento, como no

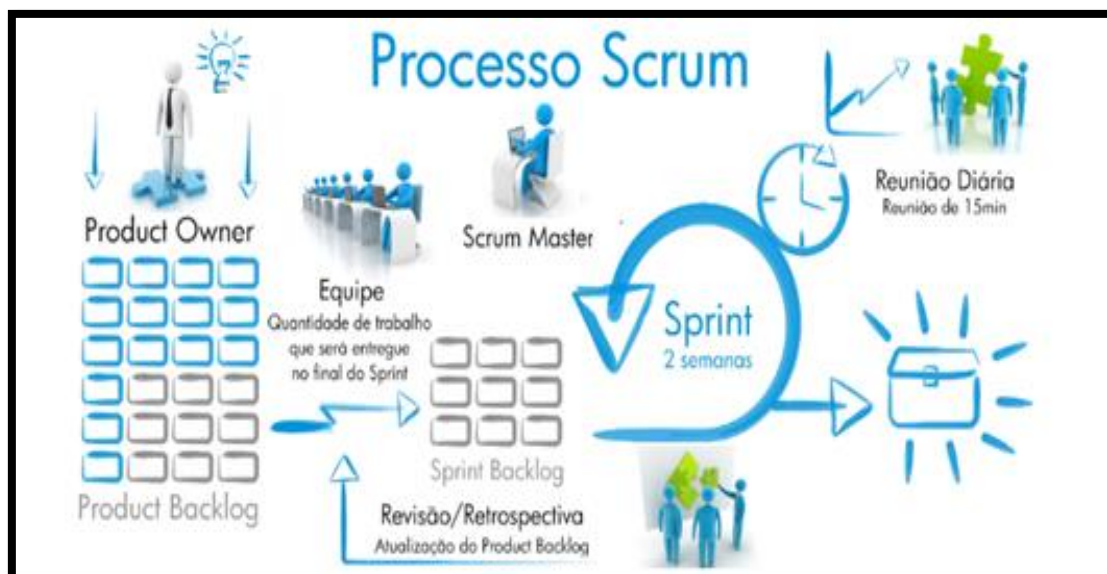
desenvolvimento de *software*. Por exemplo, o *Scrum* apresenta mais ênfase no gerenciamento de projetos; já o XP é mais focado no desenvolvimento.

Em resumo, no contexto de metodologias ágeis, ser ágil não está relacionado apenas a executar um projeto de forma rápida, mas sim de compreender que o envolvimento do cliente é necessário, o principal interessado em obter o produto; da percepção que os requisitos mudam e da necessidade de flexibilização para acomodar essas mudanças, onde uma comunicação frequente têm um impacto positivo, no sentido de troca de conhecimento, de deixar a equipe a par do que está sendo realizado, entre outros parâmetros.

2.3.2.1 Scrum

Tomás (2009), explica que o *Scrum* é um modelo ágil de processo foi desenvolvido por Jeff Sutherland e por sua equipe no início da década de 1990. Nele há um padrão de processo em que ocorrem as tarefas de trabalho, sendo esta, uma abordagem iterativa e incremental, a qual reage bem às mudanças e é indicado para projetos a serem desenvolvidos em curtíssimos prazos. Maiores informações conceituais podem ser encontradas em (RIBEIRO e RIBEIRO, 2015) e (SOMERVILLE, 2011). A Figura 4 esboça o processo *Scrum*.

Figura 4. Processo do Scrum



Fonte: (CATARINA GOMES, 2017)

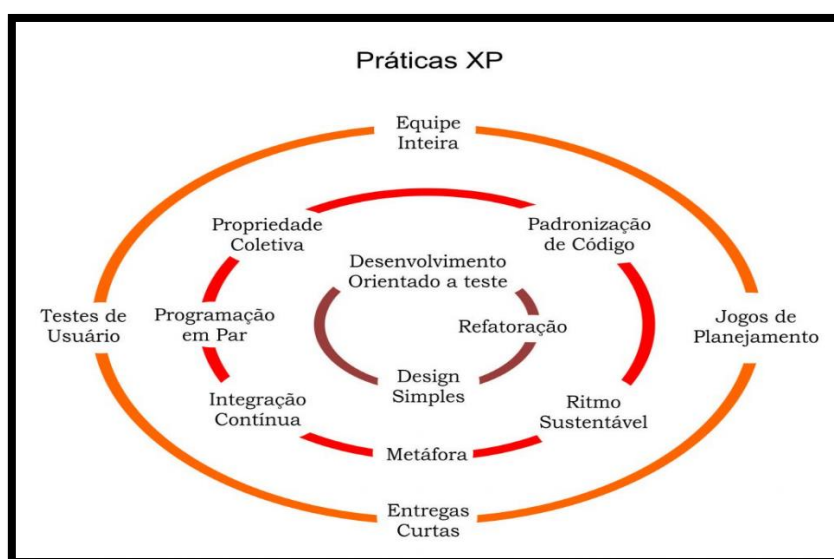
O ciclo de desenvolvimento do *Scrum*, inicia com a visão do produto que será desenvolvido, onde o proprietário (*Product Owner*) em conjunto com a equipe (*Team Member*) e o *Scrum* Master selecionam e priorizam as funcionalidades (*Product Backlog*) e as atribui as *Sprints* (*Sprint Backlog*). As *Sprints* são conhecidas como iterações e duram aproximadamente de 2 semanas a 4 semanas e nelas, as tarefas são executadas.

Durante a realização das *Sprints*, ocorrem reuniões diárias de aproximadamente de 15 minutos, em que o time se reúne para discutir as atividades a serem realizadas no próximo dia. No final de cada *Sprint* é realizada uma reunião mais incisiva para avaliar como está o andamento do processo (RIBEIRO e RIBEIRO, 2015).

2.3.2.2 Extreme Programming

Segundo Soares (2004), o XP é apropriado para projetos onde o cliente ainda não tem uma compreensão clara do que deseja, podendo sugerir alterações durante o desenvolvimento. Outras características do XP, que auxiliam entregar um *software* funcional, é o fato de que os programadores trabalham em duplas e desenvolvem testes a serem executados antes e após integração do código no sistema, na tentativa de encontrar erros que poderiam passar despercebidos pela equipe, que nesse ato contínuo procura corrigir os erros mais rápido possível. (SOMMERVILLE, 2011) (PRESSMAN, 2016). A Figura 5 esboça a ideia dessa metodologia.

Figura 5. Práticas XP



Fonte: (VIEIRA, 2018)

Esta nova metodologia foca na agilidade das equipes e na qualidade dos projetos baseando seu desenvolvimento em características como simplicidade, comunicação, feedback e força de vontade, ou seja, uma metodologia baseada em atitudes pessoais que fazem com que o projeto seja executado dentro do prazo preestabelecido e orçamento calculado, satisfazendo o cliente e fazendo com que a equipe mantenha o foco, qualidade e estabilidade funcional.

2.3.2.3 Crystal

Crystal é uma família de metodologias ágeis desenvolvidas por Alistair Cockburn em meados da década de 1990, destinadas para projetos que vão desde aqueles executados por pequenas equipes de desenvolvimento com baixa criticidade e provê abordagens até com grandes equipes capazes de implementar sistemas de alta criticidade. Eles focam na gestão de pessoas e nas comunicações (RIBEIRO E RIBEIRO, 2015). A Figura 6 esboça a ideia da família Crystal.

Figura 6. Família Crystal



Fonte: Adaptada pela autora

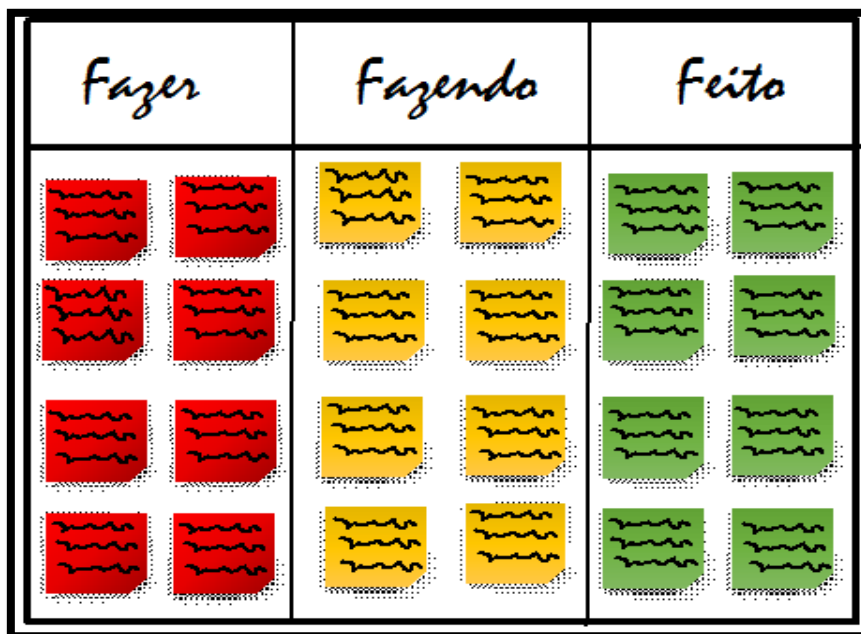
A família Crystal promove processos ágeis como, comunicação osmótica: onde os membros se reúne para melhorar a comunicação e apresentar os dados referentes ao andamento do projeto; no workshop de reflexão são identificados problemas e possíveis soluções, acontece no final de cada iteração com duração de 30 minutos;

entrega frequente por meio de interações de 2 a 4 semanas; preza pela segurança pessoal em que todos os envolvidos possam questionar; foca no que precisa ser realizado; um ambiente que permite testes automatizados, controle de configuração e uma integração frequente entre os membros das equipes (STANSBURY et. al., 2011).

2.3.2.4 Kanban

Nesse método, as equipes fixam um quadro em uma parede visível da sala de projetos, preenchendo-os com cartões que sinalizam os itens de trabalho selecionados para uma dada iteração, como forma de acompanhar o trabalho (CONCEIÇÃO E SILVEIRA, 2015). O quadro *Kanban* é dividido em fazer (“*To Do*”), onde são elencadas as tarefas a realizar; Fazendo (“*Doing*”), onde encontram-se as tarefas em andamento; e Feito (“*Done*”), constando as finalizadas. Assim, dispendo de uma visualização do status de cada tarefa. (RIBEIRO e RIBEIRO, 2015). A Figura 7 esboça esse funcionamento.

Figura 7. Quadro Kanban



Fonte: Elaborada pela autora

Conforme pode-se observar na figura acima e nas referências supracitadas, a utilização dos quadros *Kanban* apresentam inúmeros benefícios como:

- Visualizar o fluxo de trabalho: Permite visualizar os gargalos no fluxo de trabalho, posteriormente resolvê-los, melhorando o processo;
- Limitar o trabalho em progresso: Define limites como, só iniciar uma tarefa quando a outra estiver finalizada; quando houver algum problema; realizar apenas o que agrega valor;
- Tornar regras e processos explícitos: é discutir sobre o que precisa ser feito, o que está em andamento;
- Colaboração: é a percepção e contribuição sobre o processo.

2.4 Considerações Finais

Conforme foi apresentado, os pontos abordados nesse capítulo mostram que a engenharia de *software* surgiu com objetivo de minimizar ou até mesmo extinguir alguns problemas que afetam diretamente a qualidade do produto. Com isso, contribuiu-se para o surgimento e aperfeiçoamento de metodologias visando melhorar o processo de desenvolvimento de *software*, sendo as metodologias tradicionais e as metodologias ágeis que mais se destacam no mercado.

As metodologias ágeis surgiram como alternativa as tradicionais, passando a se tornar a melhor escolha da maioria dos projetos, pelo impacto positivo gerado na produção de sistemas, uma vez que traz elementos que atribui valor ao software, devido a sua flexibilidade em acomodar mudanças, atributo essencial diante do dinamismo do mercado.

3. METODOLOGIA DE PESQUISA

3.1 Considerações Iniciais

A presente seção define os métodos utilizados para a condução desta monografia, a fim de aumentar a confiabilidade e diminuir a possibilidade de eliminação de trabalhos relevantes para esta pesquisa.

Para cumprir com os objetivos propostos, esta monografia usou uma revisão bibliográfica combinada com alguns protocolos da RSL com o intuito de investigar os contributos de metodologias ágeis no mercado de desenvolvimento de *software*.

Para alcançar o objetivo, foi proposto um estudo aprofundado que buscou fazer uma análise e promover uma discussão a respeito dos trabalhos compilados que serão discutidos no capítulo 4.

3.2 Procedimentos de Pesquisa

Segundo Gerhardt & Silveira (2009, p12), “para fazer uma pesquisa científica, não basta o desejo do pesquisador em realizá-la; é fundamental ter o conhecimento do assunto a ser pesquisado.” A partir deste princípio esses autores reforçam a importância de ir em busca de estudos realizados por outros pesquisadores para melhor compreender uma dada realidade/contexto com base em diferentes e semelhantes visões, aspecto que possibilitará uma síntese apropriada para o que se pretende explorar.

Existem várias formas de sintetizar estudos presentes na literatura, a utilizada nesta monografia foi a revisão bibliográfica combinada com algumas boas práticas presentes na RSL. De acordo com Mafra e Travassos (2006), existe uma diferença na forma com é conduzido o processo de uma revisão sistemática para uma revisão bibliográfica, sendo uma informal e a outra formal.

Desde que a RSL foi introduzida na área de engenharia de *software* em 2004, essa metodologia proporcionou mais consistência e confiabilidade às pesquisas, onde contribuem constantemente com a qualidade do software, partindo do pressuposto que, o conhecimento resultante das buscas é pertinente na contribuição do seu desenvolvimento.

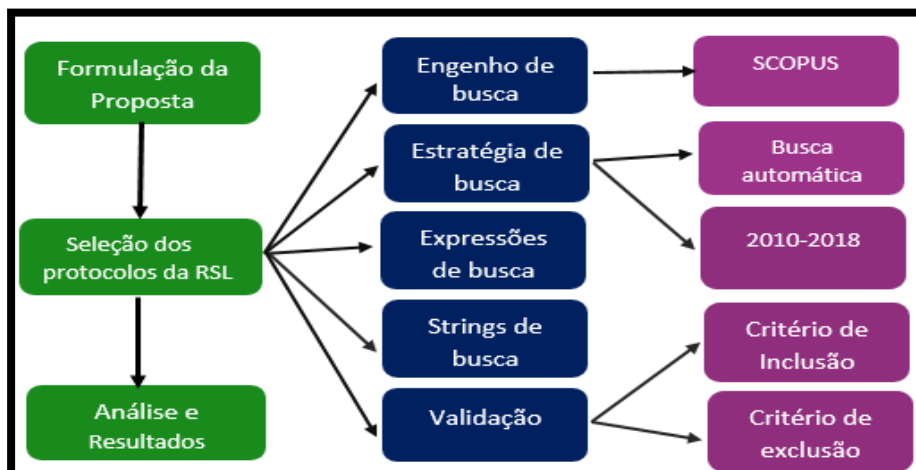
Uma RSL tem o propósito de mapear de forma bem definida estudos publicados em um determinado período, considerando que estes serão analisados e interpretados criteriosamente, baseando-se em protocolos/técnicas propostos pela RSL e relevando-se uma questão científica. (KITCHENHAM et al., 2007). Desta forma, possibilita uma ampla busca e uma compreensão aprofundada de um determinado tema. (BRERETON et. al., 2007).

Esta monografia compõe de um estudo quanti-qualitativo, em que apresenta a mensuração e estatísticas no que diz respeito a coleta e extração dos dados. Estes foram analisados, correlacionados e discutidos, a partir do entendimento sobre o objeto de estudo.

Em relação aos procedimentos técnicos adotados na presente monografia, a análise e resultados foi desenvolvida por meio da revisão bibliográfica, que consiste em discutir e relacionar os trabalhos de forma tradicional, não utilizando-se, na maioria dos casos, formular questões previstas em uma RSL. Contudo, é pertinente ressaltar que todo o processo referente a coleta e extração de dados foi realizado conforme algumas práticas propostas na RSL, posto que estes fornecem mais confiabilidade e maior retorno de trabalhos, além de ser muito utilizado na pesquisa sobre engenharia de *software*.

Um roteiro fora planejado, descrevendo o passo a passo das tarefas que foram executadas, facilitando a organização e melhorando o processo de pesquisa desta monografia. As tarefas aconteceram conforme a figura 10 a seguir:

Figura 8. Roteiro do processo metodológico.



Fonte: Elaborada pela autora

3.3 Procedimentos de Coleta de Dados

Os dados foram coletados a partir de refinamentos realizados por meio de um engenho de busca, análise e leitura dos trabalhos selecionados. As práticas da RSL utilizadas nesta monografia foram selecionadas de acordo com as orientações propostas por Kitchenham (2007) e são apresentadas a seguir.

3.3.1 Engenho de Busca

O engenho de busca escolhido para o levantamento dos trabalhos que contribuíram para esta pesquisa foi a Scopus, considerada a maior base de dados no que diz respeito a resumos e citações da literatura, dispondo de informações científicas globais de todos os campos de pesquisa, sendo, comumente, usado por mais de 3.000 instituições acadêmicas, governamentais e corporativas. A Scopus contém, atualmente mais de 20.000 títulos de mais de 5.000 editoras, incluindo a cobertura de mais de 16.500 revistas em diversas áreas do conhecimento. (SCOPUS, 2016)

3.3.2 Estratégia de Busca

Quanto abordagem da pesquisa, optou-se pela busca automática em um engenho de buscas online, realizada através de *strings* de busca, executadas a partir de um recorte temporal, delimitado no período de 2010 a 2018.

Quanto a este aspecto é pertinente mencionar que, como a busca iniciou no ano de 2018, foram compilados o máximo de trabalhos durante esse período. O principal critério para a escolha desse recorte temporal, delimitando a análise de trabalhos publicados nos últimos 08 anos, reside no fato de que o desenvolvimento e estudos na área de metodologias ágeis é um campo relativamente novo, considerando que o mercado está em constante mudança e, por esta razão, optou-se por trabalhar com conteúdos mais atuais, a fim de identificar como essa metodologia está sendo usada e qual sua contribuição para um mercado oscilante.

3.3.3 Expressão de Busca

A expressão de busca é conhecida através de palavras chaves essenciais, pois representam, em síntese, qual o foco principal de um contexto e quando são bem definidas apresentam resultados satisfatórios. As palavras chaves usadas na busca são apresentadas no Quadro 1 a seguir:

Quadro 1. Palavras-chave usadas nas strings

Palavras chave / Expressões de busca
Metodologias ágeis
Metodologia de desenvolvimento de software ágil
Metodologia de desenvolvimento de software
Práticas ágeis
Princípios ágeis
Ágil

Fonte: Elaborada pela autora

As palavras chave foram definidas com base na proposta principal da monografia, visando alcançar, como retorno, trabalhos que fossem compatíveis a proposta.

3.3.4 Strings de Busca

Este protocolo foi executado em um repositório de buscas acadêmicas/engenharia de busca, a Scopus. Neste processo foram usadas duas expressões de buscas, inseridas nos idiomas português e inglês. Essas *strings* ajudaram na sistematização da procura de trabalhos, delimitando assim o tema a ser encontrado. As *strings* utilizadas nesta pesquisa foram formadas por palavras chave e unidas por operadores lógicos, além de chaves, como ilustra o Quadro 2 a seguir:

Quadro 2. Strings utilizadas na base da Scopus

(ALL ("AGILE METHODOLOGI*") OR ALL ("METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE*") OR ALL ("DESENVOLVIMENTO DE SOFTWARE AG*") OR ALL ("METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE*")) AND (ALL (agile AND practices) OR ALL (agile AND principles)) (ALL (agile) OR ALL (agil))

Fonte: Elaborada pela autora

Para a definição das *strings* citadas no quadro acima, foram realizadas várias combinações e simulações até obter um resultado satisfatório. O ALL teve um papel importante, permitindo a consulta no título, no resumo, nas palavras chaves e em qualquer parte dos textos, abrangendo, desta forma, todo o conteúdo.

Para uma limitação dos operadores utilizados, foram usadas a conjunção AND (E, em português), onde a busca pode ser feita por mais de dois termos e o a disjunção OR (OU, em português), que oferece possibilidades na busca, podendo ser um por um termo ou por outro. Para ampliar as buscas, asteriscos foram empregados em alguns termos indicando possíveis variações.

3.4 Validação

A validação ocorreu por filtros baseados em critérios de seleção definidos e foram executados através de leituras dos trabalhos compilados e que atendiam com certo grau de similaridade a proposta do presente estudo.

A presente monografia focou em pesquisar diversas organizações e instituições que desenvolvem *software* utilizando métodos ágeis, para se chegar a um melhor entendimento sobre sua adoção e contribuição.

3.4.1 Critérios de Seleção

Os critérios de seleção consistem em requisitos definidos para determinar o que é relevante para a pesquisa, e assim alcançar o seu objetivo.

3.4.1.1 Critérios de Inclusão

Corresponde aos critérios de inclusão adotados para a escolha dos 9 principais artigos utilizados nesta monografia de conclusão. A Tabela 1 mostra esses critérios.

Tabela 1. Tabela do critério inclusão (CI)

CI	Requisitos
1	Apresentar o texto completo do estudo em formato eletrônico;
2	Ser um estudo empírico sobre metodologias ágeis em organizações;
3	Estar disponível na internet para download sem custo;
4	Estudo escrito em inglês ou em português;

Fonte: Elaborado pela autora

O Segundo critério tem um maior peso para determinar qual trabalho a ser selecionado, pois ele apresenta elementos que são primordiais para atender a proposta desta monografia.

3.4.1.2 Critérios de Exclusão

Corresponde aos critérios de exclusão adotados para o descarte dos demais trabalhos que não são relevantes para a pesquisa da presente monografia. A Tabela 2 a seguir mostra esses critérios.

Tabela 2. Tabela do critério de exclusão (CE)

CE	Requisitos
1	Não apresentar no mínimo 4 folhas;
2	Estudos sem resultados empíricos de metodologias ágeis;
3	Não tem caráter de revisão de estudos;
4	Estar em outros idiomas que não seja o Inglês ou Português;

Fonte: Elaborado pela autora

O Primeiro critério de retirada, é justificado pelo fato de que os estudos a serem selecionados para atender a proposta devem dispor de uma maior documentação, ou seja, mais 4 de páginas, relatando o máximo de detalhes possíveis sobre o uso de metodologias ágeis em organizações e instituições.

3.5 Documentação e Seleção de Estudos

A seleção de estudos é um processo indispensável, delicado e trabalhoso. A mesma deve ser executada de forma precisa e cautelosa, já que a partir dos trabalhos

retornados, se obteve os resultados pertinentes a proposta desta pesquisa. O processo de seleção foi desenvolvido por meio das 3 etapas apresentadas na Figura 9:

Figura 9. Etapa do processo de seleção de estudos



Fonte: Elaborada pela autora

Conforme a figura, a primeira etapa foi a seleção, onde ocorreu o retorno dos trabalhos, após a pesquisa executada na Scopus, por meio das *strings* de buscas.

Na segunda etapa foi realizada uma leitura preliminar (análise do título, leitura do resumo e palavras-chave) de todos os trabalhos encontrados, a fim de categorizá-los de acordo com os critérios de inclusão e exclusão descritos anteriormente.

Na terceira etapa, após o primeiro filtro, foram lidos os itens: resumo, palavras-chaves, introdução, principais resultados e conclusão dos trabalhos restantes, a fim de identificar aqueles que estavam relacionados diretamente com a proposta, aplicando novamente os critérios de inclusão e exclusão. Para facilitar a análise dos dados, será utilizada uma tabela no capítulo 4 que relaciona características dos trabalhos selecionados por essa pesquisa.

3.5.1 Extração

A pesquisa na base da Scopus retornou o total de 570 trabalhos, publicados entre os anos de 2010 e 2018, e foi realizada em janeiro de 2018 na Universidade Federal do Pará – UFPA, Campus Castanhal. Os trabalhos estão disponíveis em vários formatos, entre eles, capítulos de livros, livros, artigos e revisões. A Tabela 3 a seguir resume esses resultados.

Tabela 3. Resultado da coleta de dados

Base da dados	Total	Baixados	Repetidos	1º Filtro	2º Filtro	Selecionados
Scopus	570	446	12	434	92	9

Fonte: Elaborado pela autora

A fase de extração teve início com o download de 446 trabalhos, dos quais 124 trabalhos não estavam disponíveis para downloads de forma gratuita, mesmo o processo de coleta de dados e extração serem realizados na UFPA. Durante este processo foram identificados 12 trabalhos repetidos. Todos os estudos encontrados foram identificados através de um código e gravados em tabela.

Após o primeiro filtro foram aceitos 92 trabalhos e 342 não eram relevantes para a pesquisa, de acordo com os critérios de exclusão listados anteriormente e da leitura preliminar. Todos os 92 trabalhos foram lidos e com o resultado desse segundo filtro foram selecionados 9 trabalhos que atendem a todos os critérios de inclusão e respondem as propostas de pesquisa.

A proposta tem como intuito investigar os contributos, desafios de metodologias ágeis no mercado de desenvolvimento de *software*, por meio da análise de estudos que relatam na prática a adoção da referida metodologia em organizações.

3.5.2 Sumarização dos Resultados

Durante a pesquisa, os 9 trabalhos selecionados foram categorizados com base no ano, descrição dos projetos, objetivo, campo de conhecimento, principais razões para a adoção ágil, método ágil, iterações e dificuldades.

3.5.3 Análise dos Resultados

A fase de análise consistiu em relacionar e explicar os diferentes trabalhos selecionados, pontuando suas respectivas características, as quais são apresentadas em uma tabela presente no capítulo 4 desta monografia. A tabela foi elaborada, visando organizar estas características de modo mais didático, a fim de facilitar a compreensão dos dados apresentados e de obter subsídios para promover uma discussão com as pessoas/organizações mencionadas nos estudos aqui analisados, os quais discorreram sobre experiências com a adoção de metodologias ágeis, para, deste modo, alcançar os objetivos do estudo. Estes resultados respondem de maneira pontual ao que foi proposto, viabilizando assim a pesquisa.

3.6 Considerações Finais

Nesta seção, pode-se verificar o processo para a realização da coleta de dados, através de uma RSL, que permitiu mapear trabalhos que respondessem de maneira pontual ao que foi proposto, viabilizando assim a pesquisa.

4. ANÁLISE E DISCUSSÃO DOS CONTRIBUTOS

4.1 Considerações Iniciais

Nesta seção são apresentados os nove trabalhos selecionados em todos os filtros utilizados, cujo temas, tratam sobre como métodos ágeis vêm sendo adotados no mercado de desenvolvimento de *softwares*. Seu objetivo é analisar, estabelecer relações e promover uma discussão entres os mesmos, para reconhecer a maneira que cada trabalho pode contribuir para uma melhor escolha e utilização dessas metodologias.

A pesquisa realizada por esta monografia de conclusão de curso, o amplo levantamento bibliográfico da metodologia de pesquisa utilizada e a ampla análise de mais de 500 publicações e aprofundada dos 9 artigos selecionados, sendo estes comentados nas páginas seguintes, pode-se resumir conforme a tabela ilustrada na Figura 10 a seguir.

Figura 10. Resumo das características da metodologia de pesquisa utilizada.

Artigos	Ano	Descrição do Projeto	Objeto de Estudo	Campo de Conhecimento	Justificativa para a adoção de Metodologia Ágil	Método ágil adotado	Iterações (Semanas)	Desafios/Dificuldades
ART-1	2017	Descreve uma missão espacial da NASA.	Software de solo (Activity_planner e CLCC).	Indústria Aeroespacial	A satisfação do cliente; Capacidade de adicionar novas funcionalidades.	SCRUM	2-4	Mudança cultural; Conformidade com os processos.
ART-2	2017	Estudo de caso real de uma organização governamental.	Main C2.	Segurança Pública	A velocidade na entrega de lançamentos; Reduzir de custo.	SCRUM	5	A ambiguidade relativa às funções;
ART-3	2017	Combinação FOSS com um método ágil	Projeto Guarda Chuva.	Educação	Adaptativo; participativo.	KANBAN	—	Superar problemas;
ART-4	2016	Relatar a mudança de rotina da empresa Ericsson.	Empresa de telefonia móvel e fixo (Ericsson).	Tecnologia	Atividades de testes; Entregas em curto prazo.	SCRUM	2-4	O uso de estórias de Usuário.
ART-5	2015	Adoção e adaptações de uma metodologia ágil.	Projetos Capstone.	Educação	Entregas em curto prazo; Colaborativo;	SCRUM	6-12	Utilização do Scrum com algumas adaptações.
ART-6	2014	Combinação de uma SiS com um método ágil.	Sistemas de Mísseis Raytheon.	Indústria Aeroespacial	Atividades de testes; Foco nas pessoas.	SCRUM	2	Software em paralelo com o desenvolvimento de hardware;
ART-7	2011	Examinar o uso real de práticas ágeis.	InsightMed.	Saúde	Adaptativo; Colaborativo.	XP	1-2	Priorização de requisitos;
ART-8	2011	Implementação de um método ágil.	Projeto Capstone.	Educação	Iterativo; Flexível.	CRYSTAL	2	Documentação;
ART-9	2011	Entender aplicação do Scrum em ambientes PDCA e simulação.	Projeto de Simulação Industrial.	Tecnologia	Adaptativo; Entregas em curto prazo.	SCRUM	1-2	Mudança Cultural;

Fonte: Elaborada pela autora.

A tabela exhibe dados de cada projeto, sendo eles, identificação dos artigos, ano, tipo de projeto, campo de conhecimento, objetivos, método ágil adotado, iterações,

principais motivos/razões para adoção e desafios/dificuldades, dispondo de uma visualização geral, que ajudará em uma maior compreensão do que será abordado nesta discussão.

Os 9 artigos selecionados foram filtrados de 2011 a 2017. A Tabela 4 mostra a equivalência dos artigos por ano de publicação.

Tabela 4. Quantitativo/Ano das publicações selecionadas

Ano	2017	2016	2015	2014	2011
Quantidade	3	1	1	1	3

Fonte: Elaborado pela autora

Em termos quantitativos os anos de 2011 e 2017 foram os que mais se destacaram em número de produções, ambos apresentaram três artigos, e durante o intervalo desses anos, foi possível observar que as abordagens ágeis se adequaram melhor ao mercado, e foram capazes de satisfazer as necessidades de seus clientes. Além disso, no âmbito tecnológico ocorrem constante mudança, oportunizando novas tentativas de melhorias de processo.

Para a discussão neste capítulo, foi usado a abreviação ART seguido da numeração correspondente ao mesmo diante dos nove artigos selecionados.

4.2 Correlatos Selecionados

O ART-1 intitulado “Agile Methodology for Spacecraft Ground Software Development: A Cultural Shift”, publicado no ano de 2017 (Wortman *et.al*, 2017) faz uma descrição sobre uma missão espacial da NASA (*National Aeronautics and Space Administration* - Administração Nacional Aeronáutica e Espacial), a SPP (*Solar Probe Plus* – Sonda Solar Plus), é projetada para voar até a atmosfera com a finalidade de estudar o aquecimento coronal, aceleração do vento solar, a produção, evolução e transporte de partículas energéticas solares, com o nível máximo de detalhamento. A sonda voou significativamente mais perto do Sol do que qualquer outra nave espacial foi capaz de fazer no passado, seu lançamento foi realizado em agosto de 2018.

Para apoiar as atividades dessa missão, a Johns Hopkins University Applied Physics Laboratory (JHU/APL), localizada nos Estados Unidos, tem como função

desenvolver softwares de solo confiáveis para enviar comandos para a espaçonave (SPP), receber respostas e avaliar os dados retornados da mesma, como parte de melhoria de processos e a fim de fornecer um produto de qualidade. Para tanto, um método ágil, no caso o Scrum, foi adotado para os processos de gerenciamento de qualidade.

O ART-2 intitulado “Agile Knowledge Engineering for Mission Critical Software Requirements”, publicado em 2017 (Ciancarini *et. al*, 2018) apresenta um estudo de caso real de uma organização governamental, onde em uma missão crítica é essencial que o software seja criado para o exercício de autoridade e direção, posto que isto possibilita coordenar e implementar processos, como coleta de informações, gestão pessoal e de forças, inteligência, logística, comunicação, entre outras funções, para efetivamente apoiar as ações de comando e controle. O *software* em questão foi desenvolvido utilizando metodologias ágeis, chamado de Main C2 um sistema de informação militar.

No ART-3 intitulado “Can FOSS projects benefit from integrating Kanban: a case study”, foi publicado no ano de 2017 (Annemarie Harzl, 2017), o autor descreve uma situação do mundo real, onde uma combinação de FOSS (*Free and Open Source Software* – Software Livre e de Código Aberto) e o uso de um método ASD (*Agile Software Development* – Desenvolvimento Ágil de Projetos), são aplicadas em um projeto guarda-chuva ou projeto integrado, que segundo a definição da comissão de pesquisa da Faculdade de Educação da UFRGS (Universidade Federal do Rio Grande do Sul) (2018), “são aqueles projetos que se articulam e/ou se desdobram em outros subprojetos que são desenvolvidos em parceria: (1) com profissionais de outras instituições e/ou centros de pesquisa ou (2) com estudantes em formação”.

O ART-4 intitulado como “Replacing old routines: how Ericsson software developers and managers learned to become agile” publicado em 2016 (Lindkvi, 2017) explica como a introdução de um modo de desenvolvimento ágil e práticas ágeis implica na mudança de rotina.

Na empresa Ericsson (ERICSSON, 2019) os funcionários tiveram que desaprender as virtudes previamente aprendidas, advindo com o uso de metodologias tradicionais, reconhecendo que a mudança seria necessária. A combinação dos processos de aprendizagem de tentativa e erro foi considerada um novo ponto de partida e uma

direção geral na qual os colaboradores devem adaptar seus esforços, com objetivo de desenvolver novos testes, novas práticas e mudanças progressivas. Este cenário tornou-se interessante e apropriado para investigar mudanças de rotina.

O ART-5 intitulado “Evaluating capstone project through flexible and collaborative use of *Scrum* framework” publicado em 2015 (Souza *et. al*, 2015) avalia a adoção e adaptações de uma metodologia ágil para gerenciar projetos Capstone, que são propostas de projetos finais e que podem ser considerados trabalhos de conclusão de cursos, no curso de especialização em engenharia de *software* para desenvolvimento *web*, oferecido pela Universidade Federal de São Carlos (UFSCAR).

O ART-6 intitulado “Simulation-Based Embedded Agile Development”, publicado em 2014 (Ard, *et. al.*, 2014) trata de uma experiência de combinação de uma SiS *Simulation Software Architecture* – Arquitetura de Simulação de Software), subsidiado por um conjunto de práticas relativamente conhecidas como desenvolvimento de software ágil e o uso de uma metodologia ágil. A simulação fornece hardware virtual para desenvolver e testar o *software* de mísseis Raytheon.

O ART-7 intitulado “the impact of Absorptive Capacity on the Ex-Post Adoption of AgileMethods: The Case of Extreme Programming Model”, publicado em 2011 (Bahli, 2011) relata duas implementações ex-post, que é uma análise realizada após o uso de metodologias e está aplicada em dois projetos de uma empresa Canadense, desenvolvidos separadamente e em tempos diferentes. O primeiro foi desenvolvido por meio de uma metodologia tradicional e o segundo utilizando uma metodologia ágil.

O ART-8 intitulado “Agile Methodologies for Hardware/Software Teams for a Capstone Design Course: Lessons Learned”, publicado em 2011 (Stansbury, 2011) disserta sobre a implementação de um processo ágil no curso de projeto básico para engenharia de computação e engenharia de *software* da Universidade Aeronáutica Embry-Riddle (ERAU), que foi introduzido inicialmente em 2008.

O ART-9 intitulado “Simulation Projects Management Using *Scrum*”, publicado em 2011 (Quaglia e Tocantins, 2011), propõe o uso de uma metodologia ágil em projetos de simulação industrial em uma empresa de eletrônicos. A simulação consistiu de partes distintas do processo, onde cada peça a ser simulada se tornava um subprojeto. E esses subprojetos de simulação foram relacionados à engenharia de produção, incluindo

aspectos de melhoria do processo de produção, fluxo de materiais e análise de fluxo de recursos dentro da fábrica.

4.3 Áreas de Conhecimento

Os projetos estão distribuídos em diversos ramos de conhecimento, onde possibilitam uma melhor percepção dos cenários em que estão inseridas as metodologias ágeis. Dessa forma, eles auxiliam no entendimento de como e porque estes métodos tornam-se cada vez mais conhecidos e usados, independente da área. Neste contexto, deve-se analisar diversos fatores, dentre os quais podem-se destacar: mercado de trabalho, possibilidade de inovação e público alvo.

Os campos de conhecimentos apresentados na Tabela 5 a seguir demonstra que além da área da tecnologia, com 2 artigos, constata-se, também, a utilização de metodologias ágeis em outras áreas, como a Educação, com 3 artigos; a Indústria Aeroespacial, com 2 artigos; a Saúde, com 1 artigo e a Segurança Pública, com 1 artigo.

Tabela 5. Quantitativo de artigos por área de conhecimento

Áreas de Conhecimento	Quantidade de Artigos
Educação	3
Industria Aeroespacial	2
Tecnologia da Informação	2
Saúde	1
Segurança Pública	1

Fonte: Elaborado pela autora

Conforme distribuído na tabela acima, a aplicação de práticas baseadas em métodos ágeis é utilizada em vários segmentos, devido a sua agilidade e flexibilidade que são peças fundamentais para conseguir responder às mudanças do mercado. Esta aplicação também pode ser usada em duas vertentes, sendo uma exclusivamente voltada ao desenvolvimento de softwares e a outra com o foco na formação de profissionais com o conhecimento teórico e prático desejado sobre o assunto.

Os artigos ART-3, ART-5 e ART-8 tratam de estudos que utilizaram as metodologias ágeis na área educacional.

No ART-3, a codificação na maior parte do tempo foi feita na Universidade e por se tratar de um projeto guarda-chuva recebe contribuição de todo o mundo. Além dos estudantes, há muitos outros participantes que não são discentes, inclusive, ligados a instituições internacionais.

Como em todo projeto, há um acompanhamento de um gerente/supervisor, neste caso, um professor da Universidade, que foi o principal responsável pela orientação geral e o autor da ideia do *software* destinado para adolescentes.

Na UFSCAR, como destaca o ART-5, era importante apoiar o ensino de engenharia de *software*, utilizando metodologias ágeis, por meio do curso de especialização em engenharia de *software*. O propósito do curso foi proporcionar aos discentes um aprendizado gradual e a liberação de uma parte compatível do projeto *Capstone* com base nos assuntos estudados durante as aulas.

As turmas possuíam entre 30 a 40 alunos, estes agrupados em equipes de 6 a 7 pessoas. Cada equipe recebeu um problema e ficou responsável por um projeto, que se torna uma solução de *software*. Esses sistemas são destinados a solucionar e agilizar processos da universidade e visam, dentre outras questões, simular um ambiente de aprendizagem mais próximo à realidade.

O Projeto *Capstone*, estudado no ART-8, foi financiado pela Rockwell Collins, Inc, que é uma empresa que oferece soluções, produtos e serviços destinados a aviação e aeroportos e foi executado na ERAU. Nele os estudantes foram desafiados a implementar um sistema para detectar, identificar e afugentar animais selvagens do entorno dos aeroportos.

No curso de projeto básico foram inscritos 18 alunos (10 de Engenharia da Computação e 8 de Engenharia de Software) e a turma foi dividida em quatro equipes, com uma mesclagem de estudantes dos dois cursos em cada equipe. Eles receberam durante o processo, tarefas a serem realizadas em cada iteração, onde as funcionalidades do sistema foram desenvolvidas por etapas.

Notadamente, esses artigos apontam parâmetros que confirmam a importância desse ensino, utilizando metodologias ágeis de desenvolvimento de *softwares*, com aulas teóricas e práticas que priorizam, por exemplo: interações trabalhadas em

equipes; comunicação frequente; entregas realizadas no prazo; apresentação dos métodos e a instrução dos alunos, nestes casos apresentados, por professores e coordenadores do curso. A integração destas práticas contribuiu para tornar o ambiente acadêmico mais próximo da realidade.

Tendo em vista que as organizações operam em um mercado altamente dinâmico, preparar e melhorar as habilidades dos alunos se torna essencial, ou seja, qualificá-los para que sejam profissionais flexíveis e ágeis para lidar com essa instabilidade e aptos a desenvolverem e entregarem um produto funcional, de qualidade e de valor em um curto período.

Uma das áreas promissoras é a da indústria aeroespacial, que compreende a fabricação de aeronaves e espaçonaves, setor que se encontra empenhado em desenvolver novas tecnologias para o uso em exploração espacial, sistemas de defesa e aviação, envolvendo conhecimento científico e tecnológico, exemplos desta área são citados nos artigos ART-1 e ART-6.

O SES (Space Exploration Sector - Setor de Exploração Espacial) do JHU/APL (Johns Hopkins University Applied Physics Laboratory - Laboratório de Física Aplicada da Universidade Johns Hopkins), referenciado no ART-1, oferece suporte às Operações da Missão SPP da NASA e também projetam e constroem os subsistemas de *hardware* e *software* da nave SPP. Para isso a metodologia ágil está sendo introduzida para o desenvolvimento do *software*, sendo aplicado especificamente em dois produtos: a ferramenta de geração de carga de comando SPP (*activity_planner*) e o simulador de software (verificador de restrição de carga de comando, CLCC).

O desenvolvimento de *software* em simulação, mencionado no ART-6, ajuda na detecção de lançamento e rastreamento de mísseis e desempenha uma função fundamental, pois fornece informações precisas sobre o tipo, a trajetória e o possível alvo do míssil, com intuito de impedir ataques, e para este projeto foi realizada uma transição para um desenvolvimento ágil de *software*.

Conforme é mencionado no ART-2, em contexto de missão crítica, a segurança pública tende a tornar-se mais efetiva, pois nestes casos, essas missões, atualmente, são mais complexas e dinâmicas. Por exemplo, no campo militar, os comandantes planejam e executam longas e complicadas operações, necessitando assim de *softwares* que auxiliem no processo decisório, com possibilidade de produzir informações apuradas e

precisas, para poder executar da melhor forma as decisões tomadas e assim cumprirem a sua missão com êxito.

Nesse sentido a agência começou a desenvolver *softwares* de missão crítica, em que desenvolvedores da indústria de defesa e funcionários do governo fazem parte da equipe, estes alocados na sede da agência em Roma. Tanto a ontologia, quanto às principais funcionalidades do sistema de comando e controle foram desenvolvidas usando metodologias ágeis.

A área da saúde ainda apresenta certa resistência a introdução de recursos tecnológicos que agilizem o processo de atendimento ao paciente, gestão hospitalar, em diagnósticos e tratamentos de doenças, sobretudo, quando estes serviços se destinam aos pacientes. Em casos onde eles precisam lidar diretamente com dispositivos e *softwares*, é comum que alguns apresentem dificuldades em acessar. Para evitar questões dessa ordem, deve-se atentar para a oferta de um produto que supra as necessidades, mas que seja fácil de ser utilizado e ágil em responder as ações solicitadas.

O InsightMed, conforme citado no ART-7, é um produto que foi desenvolvido por uma empresa Canadense e para manter o anonimato ela identificada como (ABC), ela fornece informações médicas à comunidade de assistência médica. Este produto é baseado na *web* e permite que os clientes ligados a empresa interajam com os módulos de *business intelligence*, onde foram adotados os métodos ágeis de desenvolvimento de *software*.

Pelo potencial que hoje se tem de produzir e disponibilizar informação é complicado imaginar a resolução de problemas, oferta de serviços e produtos sem a utilização dos recursos disponibilizados pela tecnologia, seja por necessidade ou pelo simples prazer do entretenimento, para se adaptar a esta realidade, aproveitar as oportunidades e se manter no mercado, as empresas estão sempre buscando inovar, oferecendo novos produtos e serviços em qualquer área.

Com a demanda crescente por dispositivos de comunicação cada vez mais avançados, as empresas são compelidas a intensificarem a produção e investirem em inovação, para que assim possam sobreviver e prosperar. É o que a empresa Ericsson que opera um Centro de Desenvolvimento de Software localizado em Karlskrona na Suécia, estudada no ART-4, tem experimentado. E como uma forma para atender a

demanda por rápido desenvolvimento de novas soluções, o uso de metodologias ágeis foi adotado.

A Ericsson tinha cerca de 840 funcionários distribuídos em vários escritórios. Por se tratar de uma empresa de telecomunicações, operando internacionalmente, a organização de desenvolvimento era composta por 60 equipes ágeis, formadas por 6 a 8 pessoas, até o ano de 2011, sendo que maioria dos seus clientes é do segmento de telefonia móvel, notadamente, um público ávido por novos recursos em curto período de tempo.

O projeto de que trata o ART-9, foi aplicado em uma fábrica de dispositivos eletrônicos. Para o seu desenvolvimento foi usado uma metodologia ágil, com uma equipe composta por 7 pessoas, com participação efetiva do cliente. Cada subprojeto teve a participação de 2 a 4 pessoas, dentre elas, gerentes e especialistas técnicos, responsáveis por fornecerem respostas técnicas e instruções para a equipe e oferecendo suporte necessário.

4.4 Justificativa da Metodologia Ágil

Nos artigos elencados para este estudo, a seleção de uma metodologia de desenvolvimento de sistemas tem sido uma etapa determinante e importante para o êxito dos projetos em execução, pois uma escolha que não se adequa bem ao processo/situação, ou que seja usada indevidamente, pode ter reflexos negativos.

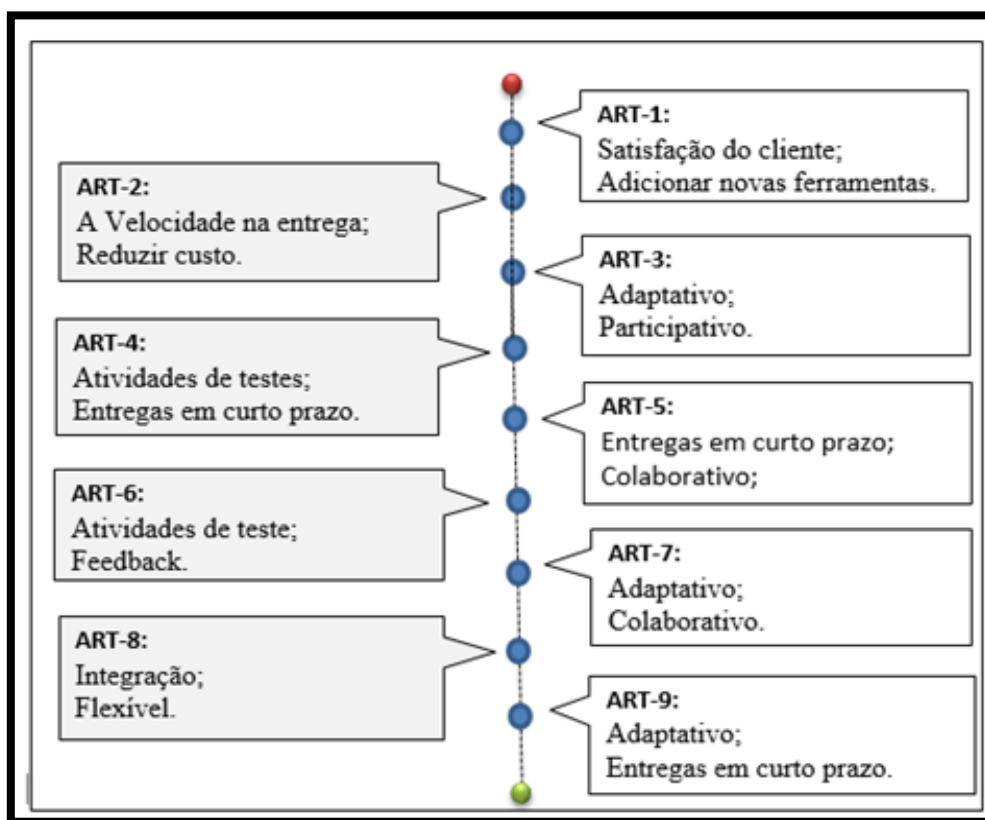
Há várias metodologias, as mais conhecidas são: a tradicional e a ágil, ambas com intuito de melhorar o ciclo de desenvolvimento, no qual cada uma possui características, vantagens e desvantagens específicas, cabendo assim conhecer o projeto, as pessoas envolvidas e o atual cerne, como aspectos essenciais que orientarão a escolha certa de uma metodologia que se adequa ao meio e avaliar a sua utilização, considerando a pré-disposição dos indivíduos a mudança cultural, já que substituir métodos tradicionais por métodos ágeis implica em abandonar padrões rígidos de desenvolvimento e ter postura receptiva às inovações.

A maior parte dos trabalhos selecionados para este estudo apresentou uma experiência com metodologias tradicionais em seus projetos anteriores e, em alguns casos, estas metodologias foram mantidas na fase inicial de seus atuais projetos,

estratégia que foi repensada a partir da compreensão de que a adoção destas metodologias já não atendia mais as expectativas e necessidades de seu público, considerando inconvenientes como o alto consumo de tempo que, em geral, supera as estimativas e cronograma de entrega, a falta de comunicação, clientes insatisfeitos, entre outros motivos que, conjuntamente, são determinantes para a adoção de uma metodologia ágil.

A Figura 11 destaca as principais razões que levaram cada projeto a adotar a metodologia de desenvolvimento ágil. Pode-se notar que os motivos que as conduziram a esta escolha foi, principalmente, sua característica adaptativa, o que pressupõe entender que as coisas se encontram em constante mudança, que nada é permanente; além disso, por apresentar um menor custo em relação a outra metodologia, na medida em que só produz a documentação necessária, por envolver o cliente em todo o processo e realizar entregas em curto prazo.

Figura 11. Razões e motivos para a adoção de uma metodologia ágil.



Fonte: Elaborado pela autora

Para entender a decisão da escolha relatada no ART-1, que retrata a adoção de um método ágil para o desenvolvimento de duas ferramentas de solo da SPP, cabe analisar que o principal critério que a justificou foi justamente a necessidade de manter seu potencial competitivo na indústria aeroespacial, posto que a aplicação de um método ágil oferece a possibilidade de construir esses produtos de forma confiável, bem como de manter um padrão de qualidade, o que resulta na satisfação do cliente.

Outra razão foi a que cada compilação incremental foi possível adicionar mais funcionalidades de acordo com a priorização de requisitos, o que possibilitou inserir novas funções, demandadas ao longo do processo e consideradas importantes e essenciais para o projeto. Estas modificações puderam ser realizadas até que a compilação completa do Ground Software fosse entregue às Operações da Missão na iteração final.

Para o ART-2, que trata de missão crítica, um dos ativos mais valiosos para aderir aos métodos ágeis decorre da velocidade de entrega, o que condiz com o objetivo principal do projeto, que era o de entregar um *software* de maneira rápida, e que a cada lançamento, funcione de forma útil para seus usuários, o que é possível devido às interações contínuas.

Em tempos de instabilidade econômica e oscilação de mercado, a redução de custo é um outro ativo que foi levado em conta na hora de optar pelo desenvolvimento de *software* ágil, pois os idealizadores do projeto almejavam o menor custo possível. A contenção nos gastos resultou, principalmente, do curto ciclo de entrega, da compreensão que os requisitos são voláteis, da produção mínima de documentação, feedbacks dos usuários, aspectos que levaram a uma importante economia nos custos de desenvolvimento, manutenção e minimização do retrabalho.

Um dos principais motivos que influenciou o projeto mencionado no ART-3 para a adoção de uma metodologia ágil, foi por ser adaptativo, pois permite mudanças, estas a serem analisadas como evolutivas, o que não requer um poder posicional, fundamentado na premissa de que algo que começou dentro de um determinado padrão deve continuar da mesma forma que foi iniciada e atende muito bem a característica de projeto de *software* livre.

O outro foi, por ser apreciado como participativo, em que há uma participação ativa de todas as pessoas envolvidas nesse processo, inclusive o usuário final, que agrega

um maior feedback, pois elas são incentivadas a contribuir também com suas opiniões, o que gera uma troca de conhecimento e experiências.

As justificativas apresentadas no ART-4, para a introdução do método *Streamline*, envolvem atividades de testes que acontecem a cada lançamento entregue, em que são testadas as funcionalidades para a averiguação e constatação de que as mesmas estejam de acordo os requisitos e necessidades do cliente, o que ocorre sem a necessidade de esperar pela versão final) que eles não precisam esperar pela versão final. Deste modo, conclui-se que os testes acontecem frequentemente, como forma de tentar encontrar erros e corrigi-los, proporcionando uma melhoria contínua.

O aumento na velocidade de entrega também foi outro quesito necessário para a escolha, a fim de manter-se no mercado, na medida em que seu público alvo é constituído por operadores no segmento de telefonia móvel, em que a necessidade de reduzir pela metade o tempo de desenvolvimento para atender às exigências do cliente em curto prazo é um fator bastante atrativo.

Diante da necessidade que o curso de especialização em engenharia de *software* para desenvolvimento tem para oportunizar aos pós-graduandos um ambiente mais próximo da realidade, considerando as constantes mudanças, foi o que o levou o curso mencionado no ART-5 a escolher uma abordagem ágil. E que uma característica forte dessa abordagem é a liberação de produtos funcionais de forma mais rápida, garantindo uma vantagem competitiva no mercado, logo os alunos se sentiriam mais preparados em atuar no mercado.

O trabalho em equipe foi outro ponto importantíssimo, no sentido que estimula a participação de todos, aumenta a interação, melhora a comunicação, deixa a equipe mais motivada, na medida em que seus integrantes aprendem enquanto colaboram.

O que levou o ART-6 a transição para um desenvolvimento ágil de *software* foi a necessidade de realizar testes frequentes no *software* de míssil, para identificar problemas de integração e implementação até o final do ciclo de desenvolvimento, e no momento em que os dados de testes forem obtidos, estes ajudem a avaliar se a simulação é suficientemente precisa para o uso pretendido.

O feedback entre os envolvidos possibilita melhorar, adicionar novos caminhos de simulação, e é um princípio utilizado em todo o processo de desenvolvimento e teste, o que ajuda a fechar o ciclo entre as fases iterativas, e entregar um produto de valor.

A experiência relatada no ART-7, revela que a decisão pela adoção de métodos ágeis, justifica-se em razão de que é um modelo que permite a adaptação às mudanças e facilita as atualizações, em situações cada vez mais frequentes nas quais os requisitos mudam como uma taxa muito maior, devido ao instabilidade do mercado e dos usuários que por muitas vezes no início não sabem definir exatamente como deseja o seu produto.

Através da comunicação frequente, contatos mútuos, o trabalho em equipe entre os usuários e equipe do projeto, melhora consideravelmente a aquisição e exploração de conhecimento entre os envolvidos, pois essa metodologia não foca somente em processos, mas sim em pessoas.

Metodologias ágeis, de acordo o ART-8 oferecem uma boa alternativa para processos de projeto de engenharia, por proporcionar um desenvolvimento iterativo, o que orienta as equipes através do desenvolvimento de um produto desde o início até a entrega, com base nas necessidades do cliente, por meio de iterações, que acontece quando o projeto é dividido em partes e facilmente implementadas, influenciando assim em entregas funcionais.

A flexibilidade, sendo essa mais uma das razões pela escolha de metodologia ágil, implica no poder que as equipes têm para definir as especificidades de seu próprio processo de design, pois mudanças de requisitos e mudanças de design, acontecem com frequência, ainda mais em um ambiente educacional, em que os alunos estão aprendendo.

A decisão do objeto de estudo do ART-9 aderir metodologia ágil surgiu das incertezas e comportamentos imprevisíveis que ocorrem na elaboração de casos de simulação, à medida que o trabalho avança, é comum notar mudanças de requisitos, em que métodos ágeis como sendo uma alternativa plausível, por este se adaptar a essas inconsistências e definir o melhor caminho para lidar com os imprevisto.

4.5 Análise e Dados da Pesquisa

Considerada umas das características principais dos métodos ágeis, as iterações são divisões de procedimentos/funcionalidades que constituem um software, a serem desenvolvidos de acordo com um planejamento e baseados em requisitos.

No ART-1, o método adotado foi o *scrum*, onde cada ciclo interativo teve a duração de 10 a 30 dias, referente a 2 até 4 semanas. A cada *sprint* é obtida uma lista crescente da funcionalidade priorizada e de novos requisitos a serem adicionados conforme o necessário, o que é conhecido com *Backlog* do Produto.

A participação efetiva do usuário no processo, foi de suma importância em que ele avaliava continuamente cada Sprint do sistema que era entregue, como forma de descobrir erros, que pudessem ser corrigidos em um tempo hábil, ajudava na escolha de um formato de relatório, entre outras tarefas.

A documentação estava sendo descrita à medida que novas funcionalidades eram adicionadas ao Backlog do Produto, e também fase de testes e design, sendo esta evolutiva ao invés de ser completa desde o início, o que dificultaria alterações no projeto até o lançamento final do *software*.

Enquanto que no ART-2, O ciclo de produção foi de 5 semanas para o desenvolvimento do Main C2, onde papéis foram atribuídas as pessoas, por exemplo, o *product owner* é um funcionário governamental que com sua equipe discute e analisa funcionalidades que serão priorizadas, estas a serem implementadas durante as *Sprints*. Após cada Sprint uma versão funcional do sistema era disponibilizada, o que possibilitou vários testes antes da implantação na rede de missão crítica. Os *stakeholders* tiveram conhecimento da metodologia ágil, no caso o *Scrum* e se envolveram no processo, principalmente dentro das iterações, e assim foram alinhando as expectativas, o que gerou um grau de satisfação.

A justificativa para o uso do *Kanban* no projeto FOSS do ART-3, foi que muitos contribuintes são desenvolvedores de softwares inexperientes, uma vez que o projeto não atende só a esta categoria, mas também ao ensino. Por este motivo o *Kanban* se mostrou uma ferramenta útil, leve e apropriada para o desenvolvimento e fins de ensino.

No entanto, sobre as iterações realizadas, apenas comenta-se no artigo, mas não relata sobre quantidade e nem a duração da mesma, apenas que a duração de uma iteração foi derivada somente da quantidade de total de pontos de histórias de usuários dividida pela velocidade da equipe. Entende-se que a duração vai depender de como equipe reage em um determinado tempo diante dos problemas impostos nas tarefas.

A empresa Ericsson, objeto de estudo do ART-4, havia em sua lacuna a demora na entrega do produto, onde a adoção do *Scrum* culminou em entregas de partes funcionais do sistema em intervalos curtos, cada entrega é chamada de *sprints* que poderiam ser concluídas dentro de 2 a 4 semanas e garantir que mudanças ou novas solicitações de clientes pudessem ser adicionadas regularmente.

A utilização do *Scrum* durante o projeto *Capstone*, relatado no ART-5, facilitou o professor a enxergar as dificuldades enfrentadas pelos alunos durante o desenvolvimento, bem como a demora na entrega do produto, inexperiência porcentagem com métodos ágeis, entres outros, e orientá-los da melhor forma possível e prestar assistência a equipes.

O aspecto iterativo do ciclo de produção para uma entrega em cada Sprint foi de 6-12 semanas, com base nas funcionalidades que foram apresentadas à equipe, onde deveriam fazer uma avaliação e o processo de priorização (*Backlog* do Produto).

Para ter uma melhor organização, um *product owner* foi nomeado para ajudar todas as equipes e um outro *product owner* foi escolhido para cada equipe para criar uma realidade, o que chamamos definição de papéis, e ter um impacto no ambiente de aprendizagem. Assim, no final de cada *sprint*, uma revisão (*sprint review*) é realizada na presença do professor responsável que mostra o que foi realizado e todos os desafios enfrentados pelas equipes durante o processo.

De acordo com o ART-6, o uso do *Scrum* na simulação facilitou o amadurecimento dos requisitos, pelo fato de que ocorria durante o planejamento e revisões das *sprints*, da forma que permite atualizar os requisitos iterativamente, alimentando assim o *backlog* de produto, para definir funcionalidades que terão prioridades a serem desenvolvidas. A integração contínua em cada *sprint* foi de 2 semanas, o que facilitou descobertas de falhas, propiciando respostas mais ágeis às mudanças necessárias nos requisitos e no design, fechando assim o ciclo entre as fases iterativas.

O estudo de caso realizado pelo ART-7, escolheu o *XP* como método ágil para o desenvolvimento do InsightMed, em que a fase de construção passou por três iterações, de acordo com a literatura leva-se de 1-4 semanas para cada iteração, onde as compilações, ou seja, a execução do código, este desenvolvidos por 3 programadores,

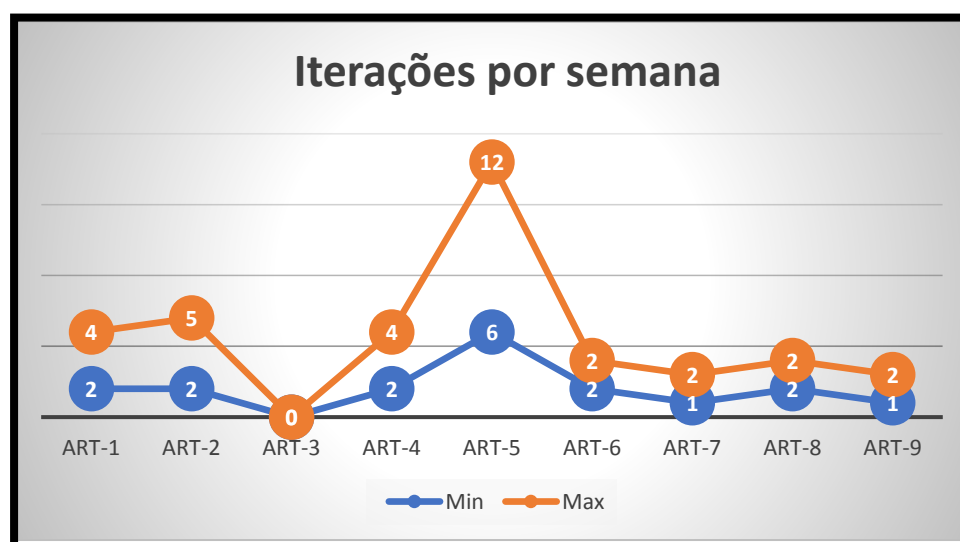
foram feitas pelo menos uma vez por semana e ao final da 3ª iteração, a aplicação ficou razoavelmente estável.

Características do *Crystal* foram estudadas pela Universidade Aeronáutica Embry-Riddle, objeto de estudo do ART-8, no qual concerniu a sua adoção. Para o curso foi determinado que as iterações seriam de 2 semanas, no caso foram três iterações, concerniu que neste período os alunos teriam tempo suficiente para realizar novos recursos e integrá-los no esqueleto ambulante, sendo que no final de cada iteração as entregas são examinadas pelo cliente, para fornecer um feedback sobre o produto.

A adoção do método *Scrum*, relatado no ART-9, foi com o intuito de melhorar o desempenho do gerenciamento de projetos de simulação, como medida de controle de mudanças durante as *sprints*, que ocasionalmente ocorre por parte do cliente em pedidos de melhorias, conforme o grau de prioridade e que podem ser adicionados ou não. O projeto começou com *sprints* de 2 semanas, mas mudou para 1 semana de duração de *sprint*. Em que a participação do cliente é maior em modelagem de simulação do que produção de *software*, o que é mais fácil de ter alterações, como posto, houve a necessidade de dividir as histórias de usuários em menores, reduzindo o escopo de cada uma, sendo possível reduzir a duração de *sprints* para 1 semana.

Conforme é apresentado na Figura 12, a maioria dos exemplos citados nos artigos seguiram o período de realização para uma iteração como é descrito na literatura, posto que cada iteração deve ser de 2 a 4 semanas.

Figura 12. Quantidade de semanas para uma iteração.



Fonte: Elaborada pela autora

E no artigo ART-5 a disparidade na quantidade de dias para uma iteração em relação aos demais se atribui ao fato de ser em um ambiente acadêmico, onde o curso tem a duração de 2 anos, em que o foco é direcionado exclusivamente para o aprendizado e não no tempo de entrega, como geralmente é visto no mercado, vindo desta de forma a preparar os alunos para o mesmo, o que implica em deixá-los propícios a trabalhar e melhorar suas habilidades e adquirir conhecimento e experiência.

Durante a execução das *sprints*, houveram algumas regressões, principalmente na *sprint 4*, como mostra a Tabela X, ocasionado por excesso de trabalho o que gerou uma falta de comprometimento com as práticas do *scrum*, causando atrasos nas tarefas e que algumas delas não fossem realizadas. No entanto, na *Sprint 7* todas as atividades foram executadas, exceto uma, por falta de concordância quanto ao seu fluxo. Este resultado evidenciou que a equipe mostrou uma maturidade no processo de desenvolvimento e ao lidar com gargalos.

Tabela 6. Visão geral da sprint

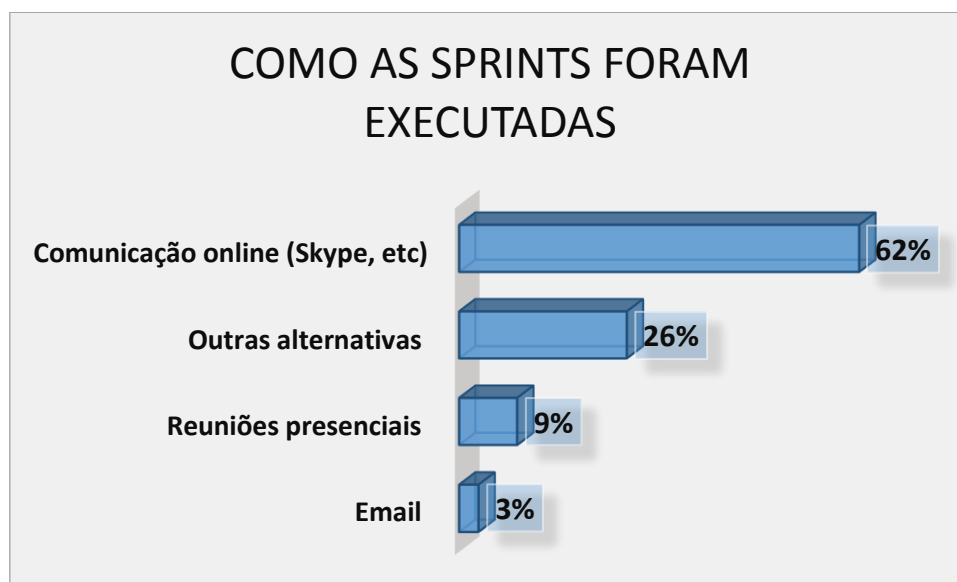
<i>Sprint</i>	<i>Dias</i>	<i>Obstáculos</i>
1	41	Escopo do Projeto
2	71	Falta de planejamento nas reuniões
3	56	Complicações no autogerenciamento
4	84	Problemas técnicos
5	63	Documentação desatualizada
6	63	Falta de dedicação dos membros
7	55	Sobrecarga de atividades

Adaptado de (Leite e Lucrédio, 2014)

Outra adaptação que as equipes tiveram que realizar e que também se difere das equipes dos outros artigos são relacionadas as reuniões diárias que na versão do *Scrum* sejam realizadas pessoalmente, como recomenda/preza a literatura. Como o curso de especialização é oferecido duas por semana, onde a maioria dos integrantes das equipes não residem na mesma cidade e têm empregos em tempo integral, restando assim, aos finais semanas e as horas livres durante a semana para dedicar-se ao projeto.

Diante desse cenário, para manter a comunicação e garantir um bom projeto, os alunos tiveram que recorrer a comunicadores online, onde 65% das vezes como mostra a Figura 13 (dados gerados através de um questionário preenchidos pelos alunos), o uso de comunicadores instantâneos e e-mails foi mais frequente do que as reuniões presenciais, como apresenta a figura tal. As reuniões presenciais eram possíveis no final das aulas para discutir a evolução das *sprint*, tirar dúvidas e ao final de cada *sprint*, uma revisão era realizada na presença do professor responsável. Os artefatos necessários para gerenciar o projeto *Capstone* também migraram para uma versão online.

Figura 13. Adaptações do Scrum



Fonte: Adaptado de (Souza et. al., 2015)

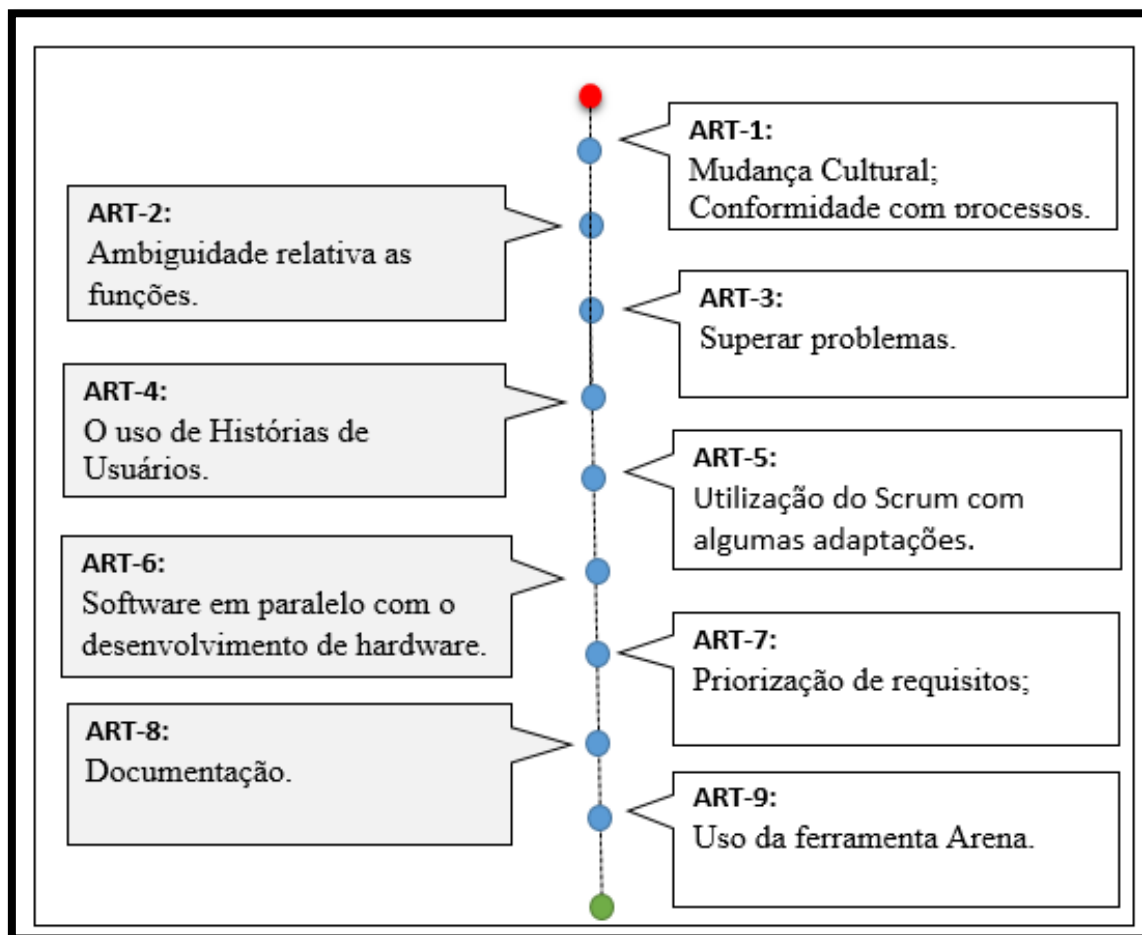
4.6 Compêndio dos contributos

À exemplo do que ocorre em outras áreas, a implementação de um processo inovador requer um período de transição e adaptação, cujos impactos podem ser positivos ou negativos, na medida em que as mudanças ocasionadas por esta nova realidade mexe com as pessoas envolvidas no processo, podendo mesmo tirá-las de sua zona de conforto, isto é, do que é habitual e, de certa forma, encontrava-se automatizado. Mudanças estruturais e procedimentais tendem a gerar dúvidas e incertezas quanto à adequação das pessoas a novas rotinas de trabalho e novos contextos de produção.

No caso da adoção de metodologias ágeis, as empresas e/ou instituições podem sentir algumas dificuldades, encontrar alguns desafios durante a fase inicial de sua implantação. Nas experiências relatadas pelos artigos analisados neste estudo, foi observado que a mudança de metodologia ocorreu, sobretudo, com o objetivo de solucionar problemas identificados durante a execução de projetos anteriores.

Exemplos são citados na imagem e percorrido ao longo desta seção de acordo com cada relato dos 9 trabalhos, para um melhor entendimento. A Figura 14 descreve as principais dificuldades e desafios da pesquisa.

Figura 14. Principais dificuldades/desafios encontrados nos artigos selecionados.



Fonte: Elaborado pela autora

Na imagem são elencados os principais problemas que foram relatados nos artigos, estes que tiveram um impacto maior, resultando em uma análise após serem

identificados os motivos e o que poderia ser feito para amenizar ou solucionar essas questões.

Um dos desafios do ART-1 ao introduzir uma abordagem ágil ao desenvolvimento, foi a mudança cultural, em que os componentes do sistema de solo da missão foram desenvolvidos seguindo outra metodologia, exceto os componentes selecionados o CLCC e *activity_planner*, estes por metodologia ágil. Logo percebe-se que esse histórico e a introdução de um modo ágil exigiu uma mentalidade diferente de toda a equipe, em que as duas abordagens têm dinâmicas de trabalho que diferem uma para a outra.

Por exemplo, em métodos ágeis a documentação pode ser menos formal, menos extensa, sendo produzido somente o necessário em relação ao que se produz nos casos em que se utiliza da metodologia tradicional. Há um envolvimento maior do cliente, o processo é flexível, conseguem lidar com os imprevistos, é interativo, adaptativo. O que precisou de um forte compromisso da gerência e desenvolvedores para que esta mudança cultural pudesse acontecer e seguir o conjunto definido de regras do *Scrum*.

Da compreensão que a fase mais crítica do sistema, encontrada no ART-2 é aquela relacionada a captura e análise dos requisitos, onde os usuários têm que explicar como imaginam e como querem o sistema. A partir dessa perspectiva surgem dificuldades quanto a implementar funções, devido à volatilidade das necessidades e desejos do cliente, que surgem durante o processo de desenvolvimento, podendo levar a uma funcionalidade que a princípio não seria desenvolvida que não era de fato a ser desenvolvida. A fim de superar essa ambiguidade, uma abordagem ágil, foi escolhida.

O ART-3 apresentou uma certa dificuldade em superar problemas, estes que são impostos e tornam-se visíveis muito cedo, no qual os alunos por um tempo ignoravam e esperavam que alguém os resolvessem para eles. Então, para tornar impossível ignorar problemas e gargalos, foi explicado para eles que teriam que resolver, ou o limite de Work in Progress (WIP) iria interromper todo o processo, logo viriam as consequências.

O desafio para empresa Ericsson, estudada pelo ART- 4, foi atribuir as Histórias de Usuários para as equipes de desenvolvimento, que em vez de anotar os trabalhos detalhadamente, ou seja, documentar tudo de forma extensiva, essas histórias de usuários se concentrariam em como os clientes usariam os recursos desejados.

Ao analisar o ART-5, percebe-se que a Universidade Federal de São Carlos no curso de especialização em Engenharia de Software, precisou realizar algumas adaptações adequadas na metodologia ágil e no Framework *Scrum*, entre elas, a rotatividade de funções do *Scrum* Master, que geralmente costumam ficar centradas em uma só pessoa considerada gerenciador central. Essa rotatividade de funções permitiu que cada aluno entendesse e pudesse vivenciar o papel de gerenciador “maior”, tendo à ideia por trás do gerenciamento diário de reuniões, removesse obstáculos, protegesse a equipe de problemas externos e facilitasse as coisas.

Para o ART-6, combinar desenvolvimento de software e desenvolvimento de hardware em paralelo é uma tarefa difícil, pois quando os recursos de hardware se tornam disponíveis na maioria das vezes são insuficientes para suportar as necessidades exigidas pelos desenvolvedores do Sistema. Para lidar com esse desafio, uma SiS com o desenvolvimento de software ágil foi empregada para subsidiar o projeto e garantir que ele executasse de forma precisa.

O principal objetivo identificado no ART-7, sobre o uso de método ágil, foi o fato que no início o usuário mostrou-se relutante quanto a tarefa de priorização de requisitos, sendo que o mesmo atribuiu um grau de obrigatoriedade para quase todos requisitos, não havendo uma priorização adequada neste primeiro momento. O que gerou uma demanda de tempo maior, pois foram meses tentando entrar em acordo com que o cliente realmente queria e o que a equipe de desenvolvimento podia oferecer dentro do tempo estimado, para então definir quais requisitos a serem definidos com prioridade alta e trabalhar com eles nas primeiras interações.

A experiência relatada no ART-8, comenta uma certa dificuldade em relação a decrescente documentação, considerando que a metodologia ágil preza por uma mínima documentação, ficando de fato só o que se intitula como necessário.

A justificativa apresentada foi que um documento de projeto deve ser bem detalhado, relatando tudo o que envolve e acontece no projeto, o que gera uma documentação bem extensa. E como o processo é iterativo, houve a necessidade dos alunos escreverem e reescreverem a documentação, o que foi considerado um trabalho oneroso.

A maior dificuldade do ART-9, foi quanto ao uso da ferramenta Arena, usada para simulação do projeto, por esta não permitir que várias pessoas trabalhem no mesmo

modelo. O ideal para uma boa execução de trabalho em equipe é que a ferramenta propicia flexibilidade na distribuição de tarefas entre os membros da equipe *Scrum*, o que afetou na produtividade e no desempenho.

4.7 Considerações Finais

No decorrer deste estudo, pôde-se constatar que a implementação de metodologias ágeis representa uma evolução na área de desenvolvimento de softwares que tem cada vez mais inserção na execução de projetos. Isto porque elas têm como premissa uma série de mecanismos que simplificam e dinamizam o processo de desenvolvimento, priorizando aspectos que favorecem a rápida adaptação em um contexto de permanentes mudanças.

As metodologias ágeis reúnem elementos que se adequam tanto as necessidades de mercado, no tocante aos interesses dos clientes, na execução do processo em curto espaço de tempo, na redução de custos, na minimização ou eliminação de etapas dispensáveis, a exemplo do excedente de documentação, muitas vezes injustificada; quanto corrobora ao aprimoramento dos processos de desenvolvimento de softwares, fortalecendo as equipes e dinâmicas de trabalho, aspectos que, de modo articulado, são fundamentais para a qualificação, produção e ampliação dos conhecimentos relativos a esse ramo da tecnologia.

5. CONSIDERAÇÕES FINAIS

Esta seção apresenta as principais conclusões desta monografia e as contribuições das mesmo para a área de engenharia de software. Apresenta também as dificuldades encontradas no processo de desenvolvimento e propostas para possíveis trabalhos.

Para um projeto de desenvolvimento de software ter sucesso é necessário que seja traçado objetivos/metapas, a serem alcançados por meio de escolhas de ferramentas, metodologias e boas práticas capazes de auxiliarem no processo, além de uma predisposição dos indivíduos envolvidos na realização das tarefas propostas.

Os trabalhos selecionados para a presente monografia foram elaborados no período de 2011 a 2017. Dentre esses anos foi possível analisar um avanço, no que se refere ao conhecimento sobre metodologias ágeis, devido à maior publicização e disponibilização de trabalhos que fazem referência ao tipo de metodologia tratada no decorrer desta monografia, aspecto que proporciona mais informações e relatos de experiências.

No decorrer desta monografia, pôde-se observar que os resultados dos 9 trabalhos que utilizaram metodologias ágeis em todo o projeto ou em uma parte dele, obtiveram contributos. Estes vêm sendo agregados como melhorias no desenvolvimento de software e contribuindo para a área de engenharia de software.

O Sucesso da adoção da metodologia ágil no ART-1 deu-se através da capacidade de se adequar ao sistema interno de gerenciamento de qualidade já estabelecido, o NPR 7150.2B, sendo contratualmente necessário para o desenvolvimento dos produtos de software de missão espacial.

Atribuiu-se o sucesso da utilização das metodologias ágeis no ART-2, ao fato de atender as reais necessidades do cliente, por meio de histórias de usuários elaboradas para o desenvolvimento de funcionalidades, permitindo alinhar os diferentes modelos mentais de usuários e desenvolvedores. Além disso, a metodologia levou a uma importante economia nos custos de desenvolvimento e manutenção.

Apesar de algumas limitações apresentadas no ART-3, a utilização de uma metodologia ágil, teve sucesso diante dos benefícios da integração de práticas Kanban no desenvolvimento de software livre. Foi observado uma mudança positiva na

interação e comunicação entre os membros das equipes, fator essencial para identificação e resolução de problemas. Constatou-se também que o tempo gasto com práticas Kanban reduziu o tempo disponível para escrever código.

No ART-4, a empresa Ericsson conseguiu atingir o objetivo de entregar um produto a curto prazo, devido a adoção de metodologias ágeis, ao alterar a rotina de trabalho, gerando equipes ágeis, e assim acelerando o processo de desenvolvimento de software.

Segundo os relatos dos alunos, sujeitos da pesquisa apresentada no ART-5, o uso da metodologia ágil foi positivo, pois foi uma abordagem que permitindo melhorar significativamente o desenvolvimento do projeto *capstone* e contribuiu para aspectos como: a flexibilidade em acomodar as mudanças, a comunicação entre os membros da equipe e a liberação de partes funcionais do sistema, de modo a possibilitar a criação de pontos de ação para tentar reduzir os problemas relacionados à entrega.

No ART-6, o sucesso foi atribuído pelo fato do Scrum ter facilitado o amadurecimento dos requisitos com mínima ou nenhuma crise, ou seja, acomodar as mudanças no decorrer do processo de desenvolvimento.

No ART-7, após obter insights da experiência no uso de metodologias no *InsightMed*, a empresa se dispôs a usar o modelo XP em seus projetos futuros. Usando esse método ágil, os desenvolvedores de sistemas de informação demonstraram uma capacidade de absorver, assimilar e explorar tanto o XP como uma inovação quanto aos requisitos dos usuários ao longo do processo de desenvolvimento de um sistema.

A partir das experiências relatadas no ART-8, com metodologias ágeis, conclui-se que estas são viáveis para cursos de design básico que incluem alunos de outros cursos. Dentre os fatores responsáveis pelo sucesso obtido constam: o nível do processo de integração frequente, maior habilidade em solucionar problemas e uma interação mais frequente.

No ART-9, os resultados da aplicação ágil no desenvolvimento de software foram considerados bons, melhorando a eficácia do projeto. Essa abordagem demonstrou-se eficiente para o controle de mudanças no escopo, o que acontece com frequência na simulação.

As limitações e dificuldades que surgiram durante a realização dos projetos apresentados pelos nove artigos não são suficientes para invalidar os resultados quanto

às contribuições, sendo as principais, planejamento direcionado, objetividade, produção em curto prazo de produtos que atendam as reais necessidades dos clientes e maior interação entre os membros envolvidos, uma vez que a maioria dos problemas foram solucionados no decorrer do desenvolvimento.

O processo de construção desta monografia forneceu a autora um entendimento mais abrangente sobre metodologias ágeis no contexto de projetos de software, além de conhecimentos sobre outros assuntos afins, dentre os quais estão, engenharia de software, metodologias tradicionais e revisão sistemática da literatura.

O processo de revisão foi uma etapa determinante para o crescimento intelectual e profissional da autora, que mesmo com as dificuldades encontradas, manteve-se obstinada a continuar com a proposta. Da mesma forma contribuíram para este processo as reflexões e diálogos oportunizados pelos momentos de orientação, compartilhados com os orientadores sobre o objeto de estudo, bem como os momentos de estudo individual, aspectos fundamentais que a instigaram a um aprendizado contínuo, a produzir e a se questionar mais sobre os resultados, a elaboração e a construção da monografia como um todo.

5.1 Dificuldades Encontradas

Apesar das metodologias ágeis se caracterizarem como um modelo relativamente novo na área de planejamento e desenvolvimento de softwares, os estudos acerca deste assunto encontram-se em constante crescimento, sendo disseminados em bases de dados, tais como, ACM, IEEE, Scopus, entre outras, que constituem um meio facilitador e uma fonte confiável para mapear trabalhos dessa natureza.

Em meio as diversas bases de dados, houve uma certa dificuldade para selecionar a base ou as bases de dados a serem utilizadas para obtenção dos resultados necessários a realização dessa pesquisa. Inicialmente foram feitos testes de *strings* de busca na *IEEE Xplore Digital Library* e na *ACM Digital Library*, contudo o resultado não foi satisfatório.

Atribui-se a dificuldade encontrada na utilização da última base mencionada, ao fato de ela não apresentar uma interface interativa. Assim, a base de dados que melhor

retornou um resultado satisfatório foi a Scopus e que apresentou uma interface bastante intuitiva e interativa.

Vale a ressalva que a união de palavras chaves é diferente para cada base de dados, o que requereu um esforço maior para a combinação dessas. No caso da Scopus várias tentativas de combinação e teste foram realizadas o que consumiu um tempo expressivo.

Como usualmente uma RSL é realizada por mais de uma pessoa, este foi um fator que também implicou na escolha de apenas uma base de dados, evitando-se assim abranger trabalhos provenientes de outras bases de dados, uma vez que nestes termos, chegar-se-ia a centenas de resultados, demandando o auxílio de mais pessoas na execução dessa etapa da pesquisa.

O idioma dos trabalhos mapeados por meio da pesquisa na Scopus, em sua maioria, está em inglês, aspecto que exigiu um conhecimento sobre o idioma para a compreensão dos trabalhos. Sendo que a autora tem um domínio Intermediário, fato que também tornou o processo um pouco mais difícil, ante a frequente necessidade de recorrer ao uso de ferramentas de tradução.

5.2 Trabalhos Futuros

Pretende-se aprofundar as pesquisas voltadas para escalabilidade em grandes projetos com uma equipe maior. Para isso, a pesquisa deverá ser realizada em duas ou mais bases de dados, ampliando-se o período de busca de 10 a 15 anos. Como procedimento metodológico será adotado exclusivamente a RSL, com questionamentos pertinentes acerca do assunto, a fim de investigar a viabilidade de adoção de métodos ágeis, neste contexto.

Pretende-se também aplicar os métodos ágeis em empresas/instituições com o objetivo de fazer uma avaliação dos reais contributos destes em locais que não o utilizem ou o utilizem de forma não adequada.

REFERÊNCIAS BIBLIOGRÁFICAS

AGILE REPORT, VersionOne 12th Annual State of Agile Report. 2018. Disponível em: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>. Acesso em: 14. Jan. 2019.

ALBINO, R. D., Souza, C. A. de, & Prado, E. P. V. “Benefícios Alcançados por Meio de um Modelo de Gestão Ágil de Projetos em uma Empresa de Jogos”. 2014. Revista de Gestão e Projetos - GeP, V.5, 15–27. Disponível em: <https://doi.org/10.5585/gep.v5i1.197>. Acesso: 12 jan. 2019.

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: 07 de nov. 2018

AMARAL, Daniel Capaldo. et al. Gerenciamento ágil de projetos: aplicação em produtos inovadores. São Paulo: Saraiva, 2011. 240p.

ARD, J., Davidsen, K., & Hurst, T. Simulation-based embedded agile development. 2014. IEEE Software, 31(2), 97–101. Disponível em: <https://doi.org/10.1109/MS.2014.42>. Acesso em: 07 fev. 2018.

BAHLI, B., Benslimanne, Y., & Yang, Z. The impact of absorptive capacity on the ex-post adoption of agile methods: The case of Extreme Programming model. In IEEE International Conference on Industrial Engineering and Engineering Management (pp. 1660–1664). Disponível em: <https://doi.org/10.1109/IEEM.2011.611819>. Acesso em: 07 fev. 2018.

BRERETON, P. et al. 2007. Lessons from applying the systematic literature review process within the software engineering domain, *Journal of Systems and Software*, vol. 80–4.

CIANCARINI, Paulo, et. al. Agile knowledge engineering for mission critical software requirements. 2018. In *Advances in Intelligent Systems and Computing* (Vol. 626, pp. 151–171). Springer Verlag. Disponível em: https://doi.org/10.1007/978-3-319-64161-4_8. Acesso em: 07 fev. 2018.

CODAS, Manuel, *Gerência de projetos: Uma Reflexão Histórica*, 1987. Disponível em: <http://www.scielo.br/pdf/rae/v27n1/v27n1a04.pdf>. Acesso em: 02 jan. 2019.

DE SOUZA, R. T., Zorzo, S. D., & Da Silva, D. A.. Evaluating capstone project through flexible and collaborative use of *Scrum* framework. 2015 In *Proceedings - Frontiers in Education Conference, FIE* (Vol. 2014). Institute of Electrical and Electronics Engineers Inc. Disponível em: <https://doi.org/10.1109/FIE.2015.7344249>. Acesso em: 07 fev.2018.

ERICSSON. Empresa de Tecnologia. 2019. Disponível em: <https://www.ericsson.com/en>. Acesso em: 27 jan. 2019.

FILHO, Dairton Luiz Bassi. *Experiências com desenvolvimento ágil*. 2008. Disponível em: <https://www.ime.usp.br/~dairton/files/Dissertacao-DairtonBassi.pdf>. Acesso em: 30 dez. 2018.

FOWLER, Martin. *The New Methodology*. 2005. Disponível em: <https://www.martinfowler.com/articles/newMethodology.html>. Acesso em 12 jun. 2018.

GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo. *Métodos de pesquisa*. 1ª Edição, Porto Alegre: UFRGS, 2009.

GOMES, Catarina. Scrum: A Metodologia ágil simplificada. 2017. Disponível em: <http://blog.europneumaq.com/scrum-metodologia-agil-simplificada>. Acesso em: 14 dez. 2018.

HARZL, Annemarie. Can FOSS projects benefit from integrating Kanban: a case study. 2017. *Journal of Internet Services and Applications*, 8(1). Disponível em: <https://doi.org/10.1186/s13174-017-0058-z>. Acesso em: 07 fev.2018.

HIGHSMITH, Jim. *Agile Software Development Ecosystems*. Addison-Wesley, 2002. Disponível em: <https://epdf.tips/queue/agile-software-development-ecosystems.html>. Acesso em: 15 jan. 2019.

KERZNER, Harold. *Gestão de Projetos: As melhores Práticas*. 3. ed. Porto Alegre: Bookman, 2017. Disponível em: http://srvd.grupoa.com.br/uploads/imagensExtra/legado/K/KERZNER_Harold/Gestao_de_Projetos_3Ed/Lib/Amostra.pdf. Acesso em: 02 jan. 2019.

KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Keele University and Durham University Joint Report, Tech. Rep. EBSE 2007-001, 2007.

LIMA, Greick Roger de Carvalho. *Benefícios das metodologias ágeis no gerenciamento de projetos de Tecnologia da Informação (TI)*. 2015. Disponível em: <https://www.ipog.edu.br/revista-especialize-online/edicao-n10-2015/>. Acesso em: 03 jul. 2018.

LIMA, Roberto Ricardo. *Metodologia de Desenvolvimento de Sistemas de Informação baseados em OO*. 2007. Disponível em: http://www.unibratec.edu.br/tecnologus/wp-content/uploads/2007/08/n2_lima_rr.pdf. Acesso em: 13 dez. 2018.

LINDKVIST, Lars., et al. Replacing old routines: How Ericsson software developers and managers learned to become agile. 2017. *Industrial and Corporate Change*, 26(4), 571–591. <https://doi.org/10.1093/icc/dtw038>. Acesso em: 07 fev.2018.

MAFRA, S. N.; TRAVASSOS, G. H. Estudos Primários e Secundários Apoiando a Busca por Evidência em Engenharia de Software. COOP/UFRJ. Rio de Janeiro, p. 32. 2006. (RT -ES 687/06).

MAINART, Domingos de A; SANTOS, Ciro M. Desenvolvimento de Software: Processos Ágeis ou Tradicionais? Uma visão crítica. 2010. Disponível em: http://www.enacomp.com.br/2010/anais/artigos/completos/enacomp2010_4.pdf. Acesso em: 20 jun. 2018.

MARQUES, André Nobrega. Metodologias ágeis de desenvolvimento: Processos e Comparações. 2012. Disponível em: Acesso em: 17.out. 2018.

OLIVEIRA, Bruno Carazato. Gaia Protótipo: Um Modelo De Prototipação Para Processos de Desenvolvimento De Software. 2018. Disponível: http://www.uel.br/cce/dc/wp-content/uploads/TCC-BRUNO_CARAZATO_DE_OLIVEIRA-BCC-UEL-2017.pdf Acesso: 17.out. 2018.

PRESSMAN, Roger S. Engenharia de software: Uma Abordagem Profissional. 7. ed. São Paulo: McGraw-Hill, 2011. 780p.

PRESSMAN, Roger S; MAXIM, Bruce R. Bruce. Engenharia de software: Uma Abordagem Profissional. 8. ed. São Paulo: McGraw-Hill, 2016. 968p.

PROJECT MANAGEMENT INSTITUTE – PMI, A Guide to the Project Management Body of Knowledge. 5. ed. Porto Alegre: Bookman, 2013. Disponível em: http://dinus.ac.id/repository/docs/ajar/PMBOKGuide_5th_Ed.pdf. Acesso em: 02 jan. 2019.

QUAGLIA, E. J., & Tocantins, C. A. Simulation projects management using scrum. In Proceedings - Winter Simulation Conference (pp. 3421–3430). Disponível em: <https://doi.org/10.1109/WSC.2011.6148038>. Acesso em: 07 fev. 2018.

RANDELL, Brian. A OTAN de 1968/1969: Relatórios de Engenharia de Software. 1996. Disponível em: <https://www.homeyou.com/~edu/otan-de-1968-1969>. Acesso em: 20 jan. 2019.

RIBEIRO, Rafael Dias; RIBEIRO, Horácio da Cunha e Souza. Métodos Ágeis em Gerenciamento de Projetos. 1ª Edição, Rio de Janeiro, 2015. Disponível em: <http://www.faeterj-rio.edu.br/downloads/bbv/0059.pdf>. Acesso em: 17 out. 2018.

SCOPUS. Empresa de Tecnologia. 2016. Disponível em: https://www.periodicos.capes.gov.br/images/documents/Scopus_Guia%20de%20refer%C3%Aancia%20r%C3%A1pida_10.08.2016.pdf. Acesso em: 27 jan. 2018.

SILVEIRA, Juliano; CONCEIÇÃO, Sidney. Aplicação de Metodologias Ágeis para Desenvolvimento de Software: Um Estudo de Caso na Empresa Alliance Software. 2015. Disponível em: <https://repositorio.ufsm.br/handle/1/12846?show=full>. Acesso em: 17 out. 2018.

SOARES, Michel do Santos. Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. 2004. Revista Eletrônica de Sistemas de Informação, 3 (1). Disponível em: <https://doi.org/10.21529/RESI.2004.0301006>. Acesso em: 27 nov. 2018.

SOARES, Michel dos Santos. Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. 2004. Disponível em: <http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/68>. Acesso em: 25 nov. 2018.

SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Prentice, 2011. 529p.

SOUZA, Diogo Rodrigues. Implantação Da Metodologia Ágil Scrum Em Um Ambiente De Desenvolvimento. 2014. Disponível: <https://repositorio.ufsc.br/>. Acesso em: 03 jul. 2018.

STANSBURY, Richard. Agile methodologies for hardware / software teams for a capstone design course: Lessons learned. 2011. In ASEE Annual Conference and Exposition, Conference Proceedings. Disponível em: [//doi.org/10.1007/s003820050230](https://doi.org/10.1007/s003820050230). Acesso em: 07 fev.2018.

TOMÁS, Mario Rui Sampaio. Métodos ágeis: Características, Pontos Fortes e Fracos e Possibilidades de Aplicação. 2009. Disponível em: <https://run.unl.pt/handle/10362/2003>. Acesso: 17.out. 2018.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, Faculdade de Educação. Roteiro Aberto Para Elaboração De Projeto De Pesquisa Integrado/“Guarda-Chuva”. 2018. Disponível em: <https://www.ufrgs.br/compesqedu/wp-content/uploads/2018/09.pdf>. Acesso em: 28 jan. 2019.

UTIDA, Kleber Hiroki. Metodologias Tradicionais E Metodologias Ágeis: Análise Comparativa Entre Rational Unified Process E Extreme Programming. 2012. Disponível em: <http://www.fatecsp.br/dti/tcc/tcc00055.pdf>. Acesso em: 03 jul. 2018.

VIERA, Agatha. Metodologia Agile- O que é e como aplicar. 2018. Disponível em: <https://www.lecom.com.br/blog/valores-da-metodologia-agile/>. Acesso em: 19 jan. 2019.

WORTMAN, K., Duncan, B., & Melin, E. Agile methodology for spacecraft ground software development: A cultural shift. 2017. In IEEE Aerospace Conference

Proceedings. IEEE Computer Society. Disponível em:
<https://doi.org/10.1109/AERO.2017.7943886>. Acesso: 07 fev. 2018.