



Universidade de Brasília – UnB  
Faculdade UnB Gama – FGA  
Engenharia de Software

## **Melhoria de um processo de engenharia de requisitos de software**

Autora: Giulia Lobo Barros  
Orientadora: MSc. Cristiane Soares Ramos  
Coorientador: MSc. Ricardo Ajax Dias Kosloski

Brasília, DF  
2023



Giulia Lobo Barros

# **Melhoria de um processo de engenharia de requisitos de software**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: MSc. Cristiane Soares Ramos

Coorientador: MSc. Ricardo Ajax Dias Kosloski

Brasília, DF

2023

---

Giulia Lobo Barros

Melhoria de um processo de engenharia de requisitos de software/ Giulia Lobo Barros. – Brasília, DF, 2023- 107 p. : il. (algumas color.) ; 30 cm.

Orientador: MSc. Cristiane Soares Ramos

Coorientador: MSc. Ricardo Ajax Dias Kosloski

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB

Faculdade UnB Gama – FGA , 2023.

1. Aplicativos de Governo. 2. Design da Experiência do Usuário. 3. Desenvolvimento Orientado ao Comportamento. 4. Transformação Digital. 5. Teste caixa-preta. I. MSc. Cristiane Soares Ramos. II. MSc. Ricardo Ajax Dias KosloskiIII. Universidade de Brasília. IV. Faculdade UnB Gama. V. Melhoria de um processo de engenharia de requisitos de software

CDU

---

Giulia Lobo Barros

## **Melhoria de um processo de engenharia de requisitos de software**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Brasília, DF  
2023

# Agradecimentos

Agradeço, primeiramente, a Deus, por ter me dado a vida e permitido que eu fizesse parte de uma família amorosa e incentivadora.

Também agradeço à Nossa Senhora, minha mãe do céu, que sempre intercede por mim e pela minha família todos os dias.

Agradeço aos meus pais, Gláucia e Gustavo, por serem amorosos, se importarem sempre com as minhas irmãs e comigo, por nos proporcionarem uma vida com boas oportunidades e muita fé, por estarem ao meu lado em todas as situações, me incentivarem a correr atrás dos meus objetivos e me permitirem ser frágil quando é necessário, assim como me motivar quando preciso ter força.

Também sou grata às minhas irmãs, Giovana por ser uma pessoa que está na minha vida para me ensinar a levá-la com mais leveza e menos preocupações e me fazer rir com o seu jeito bobo de ser e Gabriela por ser a pessoa que me ensina a ver como a vida passa rápido e devemos aproveitar cada segundo dela, ela é alguém que vi nascer e já era mais velha, portanto, tinha mais compreensão das coisas e acompanhei todo o seu crescimento até aqui e às vezes sinto falta de tempos mais simples, porém aproveito mais o presente por conta dessa lição que ela está sempre me ensinando.

Agradeço ao meu gatinho, Dengo, ele me mostra a forma mais pura do amor com seu jeitinho felino de ser. Deita do meu lado pedindo carinho e ajuda com que me sinta confortável, mesmo em momentos turbulentos.

Além também de agradecer à minha avó Maria Mará, ela me marcou muito, mesmo tendo passado poucos anos da minha vida em sua presença. Ouvir as histórias que minha mãe conta sobre ela é sempre leve, divertido e me dá vontade de tê-la conhecido melhor.

Sou muito grata também aos meus padrinhos, dinda Cláudia e dindo Zé Carlos, por serem como segundos pais para mim, estiveram e estão presentes nos momentos mais marcantes da minha vida, assim como este e sempre me incentivaram e me aconselharam como meus pais.

Agradeço à minha avó Áurea, uma pessoa autêntica e cheia de energia, apesar de sua idade. Além disso, sua fé em Deus e Nossa Senhora me inspira a ser uma pessoa cheia de esperança e confiante na vida.

Também agradeço à minha tia Grasiela, por ser alguém que se esforça muito e corre atrás dos sonhos. Com seu trabalho de enfermeira ela ajuda muitas pessoas e com sua contribuição a pesquisas científicas ela faz grandes descobertas que ainda vão ajudar o mundo. Essas características me lembram de ser persistente e ajudar as outras pessoas.

Meu primo Gabriel também é alguém que sou grata, nós somos muito próximos e contamos um com o outro. Nos momentos difíceis, assim como compartilhei muitas coisas com ele, ele também compartilhou comigo e estivemos lá um pelo outro para nos apoiarmos.

Agradeço também a oportunidade de ter convivido anos da minha vida e ter sido cuidada e amada pela minha bisa, Moema, meu biso, Waldemiro, minha tia avó Ana e meu tio avô Zé.

Sou grata aos meus demais tios: Glauco, Fernando, Beto, Didi e, em especial, meu tio Paulo, que partiu desta vida em 2022. Eles são pessoas que sempre me recebem na cidade natal dos meus pais, o Rio de Janeiro, são muito carinhosos e nos fazem sentir em casa.

O tio Fernando trabalha na mesma área que eu e, desde que iniciei essa jornada, ele me fala de suas experiências, me aconselha e me ajuda a ter mais direção do meu futuro.

Agradeço ao Ricardo e a sua filha Bárbara, que entraram de forma um pouco tardia em minha vida, me acolheram como família e me tratam como parte dela.

Uma pessoa muito especial que devo agradecer também é o meu namorado, Matheus, ele é uma pessoa incrível que está ao meu lado sempre e não me deixa desistir. Ele entende quando não estou bem e, além de me respeitar, me ajuda a me sentir melhor e a encontrar soluções para os problemas.

Sou grata também à família dele, minha sogra Alda, meu sogro Walderico e meu cunhado Samuel, por terem me recebido de braços abertos, não somente em sua casa, como também em sua vida e me tratam muito bem, com carinho e amor em todos os momentos.

Agradeço às minhas amigas mais antigas da vida, Isadora e Marina. Conheci as duas com 5 anos e nunca mais larguei, elas são pessoas que me mostram o que é a amizade para a vida toda e estão sempre me acompanhando.

À minha amiga Flávia também agradeço por estar ao meu lado e se importar comigo e com a nossa amizade.

Também sou muito grata à Maria Fernanda (Mafê), por, além de também já ser minha amiga há 12 anos, é uma inspiração na fé, é devota à Nossa Senhora e sempre me ajuda a perseverar.

Agradeço também aos meus amigos do Ensino Médio, Arielle, Cássia, Lucas (Pali-tão) e Vanessa, viveram e compartilharam muitos sonhos comigo, fizeram os meus anos de escola serem inesquecíveis e viram sonhos meus sendo realizados, assim como vi os deles.

Agradeço ao Palitão também por ser outra pessoa que me faz estar sempre presente, por ser sua madrinha de crisma e à Natália. Os dois são meus afilhados, os amo muito e desejo que Deus e Nossa Senhora sempre sejam os guias de suas vidas.

Também agradeço aos meus amigos da faculdade, que seguem nessa jornada comigo, entendem as minhas frustrações e me apoiam nas decisões, assim como vibram pelas minhas vitórias. Este agradecimento vai para a Nathália, o Kauan, os Gabrieis William e Menor, Pedro, Álvaro, Denniel e Carla.

De forma mais especial, agradeço à Gabriela, por ser minha companheira nesse caminho quase inteiro, uma das poucas mulheres que está no curso comigo e que se tornou uma grande amiga, que pretendo levar para a vida.

Agradeço ao Hugo, meu amigo guatemalteco, que sempre me alegra com seu jeitinho de ser, me trata com muito carinho e se importa muito comigo, é um amigo maravilhoso que também pretendo levar depois da faculdade.

Outra pessoa fundamental nos meus anos de faculdade é o Liverson (Livinho), que é um grande parceiro, uma pessoa que, apesar de não ser facilmente conquistada, abriu as portas para a nossa amizade sem pestanejar e assim permaneceu, ele é muito especial na minha vida e espero que continuemos próximos.

Também agradeço ao Murilo, que é o meu maior parceiro de faculdade e de trabalho, nós estamos sempre juntos e contamos muito uns com o outro, ele sempre me ajuda a ver o lado bom das situações e é um grande braço amigo.

Sou grata à empresa onde eu trabalho e às pessoas que me deram oportunidade de crescer lá, além das amizades que desenvolvi: Tiago, Isabella, Augusto e os demais.

Agradeço à minha psicóloga, Tatiane, por ser uma guia, especialmente nos momentos mais difíceis e angustiantes, apoiando no meu desenvolvimento pessoal.

Agradeço à faculdade Universidade de Brasília, por ter me proporcionado momentos inesquecíveis e pela empresa júnior Matriz Engenharia de Energia e todos que conheci lá.

Para finalizar, agradeço aos meus professores, em especial, ao meu professor de matemática, Cristiano Soares, que me orientou e me aconselhou na escolha do curso e aos meus orientadores: Cristiane Soares e Ricardo Ajax, por, além de serem guias, serem muito atenciosos, compreensivos e sinceros.

*"Mesmo que já tenhas feito uma longa caminhada, há sempre um novo caminho a fazer."  
(Santo Agostinho)*



# Resumo

As mudanças que estão ocorrendo nos ambientes de negócios motivam as empresas a melhorar suas estruturas organizacionais e seus processos produtivos. Os processos definem as fases e atividades responsáveis pelo desenvolvimento do produto. Seu controle e manutenção tem efeito direto no produto final e, neste contexto, torna-se relevante a melhoria de processo, que significa compreender os processos existentes e modificá-los para aumentar a qualidade do produto e/ou reduzir os custos e o tempo de desenvolvimento. Para apoiar a melhoria de processos é importante considerar, também, a maturidade e capacidade dos processos. A Engenharia de Requisitos é uma disciplina que influencia em todo o processo de desenvolvimento de software. Suas atividades são, principalmente, dedicadas a identificar, definir e gerenciar os requisitos do produto a ser desenvolvido e, dentro da melhoria de processos de software, muitos desafios estão relacionados aos requisitos. Para este trabalho, é realizado um estudo de caso, classificado como uma pesquisa descritiva, aplicada e qualitativa e tem o objetivo de mapear e identificar falhas no processo de engenharia de requisitos da empresa ATR, além de propor melhorias para esse processo. Para esse fim, foram seguidas etapas do modelo IDEAL e feitas coletas de dados que serviram para apoiar o diagnóstico e, posteriormente, a proposta de solução para a melhoria do processo.

**Palavras-chave:** Qualidade de software; Melhoria de processos de software; Engenharia de Requisitos; SRP-CMMI; processo ágil de software.

# Abstract

The changes happening in business environments, motivate companies to improve organizational structures and productive processes. Processes define phases and activities responsible for product development. Its control and maintenance has a direct effect on the final product and, considering that, software process improvement become relevant. Its definition means understanding existent processes and modify them to increase product quality and/or reduce cost and development time. To support process improvement, it is important to consider, capability and maturity as well. The Requirements Engineering is a discipline that influences the whole software development process. Its activities are, mainly, dedicated to identify, define and manage the requirements of the product to be developed and in software process improvement area, lots of challenges are related to requirements. For this paper, a case study was carried out, with the objective of mapping and identifying failures in the requirements engineering process of a company called ATR, in addition to proposing improvements to this process. Therefor, IDEAL model stages were followed and data were collected to support diagnosis stage and, subsequently, the proposed solution to improve the process.

**Key-words:** Software Quality; Software process improvement; Requirements Engineering; SRP-CMMI; Agile Methodology.

# Lista de ilustrações

Figura 1 – ETAPAS DA ENGENHARIA DE REQUISITOS. . . . .	21
Figura 2 – O PROCESSO DO EXTREME PROGRAMMING (XP). . . . .	24
Figura 3 – PROCESSO DO SCRUM. . . . .	25
Figura 4 – ENGENHARIA DE REQUISITOS NO SCRUM. . . . .	26
Figura 5 – CICLO DE VIDA (FDD). . . . .	28
Figura 6 – EXEMPLO DE UMA IMPLEMENTAÇÃO DO QUADRO KANBAN. . . . .	29
Figura 7 – MODELO DE QUALIDADE EM USO. . . . .	31
Figura 8 – MODELO DE QUALIDADE DO PRODUTO. . . . .	32
Figura 9 – QUALIDADE BASEADA EM PROCESSOS. . . . .	34
Figura 10 – ESTRUTURAS DAS REPRESENTAÇÕES CONTÍNUA E POR ESTÁGIOS. . . . .	36
Figura 11 – NÍVEIS DE MATURIDADE - CMMI-DEV. . . . .	37
Figura 12 – ESTRUTURA DO MR-MPS-SW. . . . .	39
Figura 13 – NÍVEIS DE MATURIDADE MPS.BR. . . . .	40
Figura 14 – OS CINCO NÍVEIS DO FRAMEWORK R-CMM. . . . .	43
Figura 15 – NÍVEL 2 R-CMM. . . . .	44
Figura 16 – COMPONENTES DO SRP-CMMI. . . . .	46
Figura 17 – NÍVEIS DE MATURIDADE DO SRP-CMMI. . . . .	47
Figura 18 – PRÁTICAS NÍVEL 1 SRP-CMMI. . . . .	47
Figura 19 – PRÁTICAS NÍVEL 2 SRP-CMMI. . . . .	48
Figura 20 – CICLO PDCA. . . . .	50
Figura 21 – O MODELO IDEAL. . . . .	52
Figura 22 – MEDIÇÕES DE PREVISÃO E CONTROLE. . . . .	53
Figura 23 – PLANO METODOLÓGICO. . . . .	59
Figura 24 – PROCESSO DE ENGENHARIA DE REQUISITOS DA EMPRESA ATR. . . . .	68
Figura 25 – QUADRANTE DO PICK CHART (EXEMPLO). . . . .	91
Figura 26 – MATRIZ DO PICK CHART (MIRO) . . . . .	92
Figura 27 – PLANEJAMENTO 1º CICLO DE MELHORIA - PARTE 1. . . . .	94
Figura 28 – PLANEJAMENTO 1º CICLO DE MELHORIA - PARTE 2. . . . .	94

# Lista de tabelas

Tabela 1 – TABELA DE COMPARAÇÃO R-CMM E SRP-CMMI. . . . .	48
Tabela 2 – O MODELO IDEAL DE PROGRAMAS DE MELHORIA. . . . .	51
Tabela 3 – CRONOGRAMA TCC1. . . . .	63
Tabela 4 – CRONOGRAMA TCC2. . . . .	64
Tabela 5 – ESCALA FILIPINY. . . . .	70
Tabela 6 – AVALIAÇÃO DE MATURIDADE - ELICITAÇÃO. . . . .	75
Tabela 7 – AVALIAÇÃO DE MATURIDADE - ANÁLISE. . . . .	77
Tabela 8 – AVALIAÇÃO DE MATURIDADE - ESPECIFICAÇÃO. . . . .	79
Tabela 9 – AVALIAÇÃO DE MATURIDADE - VALIDAÇÃO. . . . .	80
Tabela 10 – AVALIAÇÃO DE MATURIDADE - GESTÃO DE REQUISITOS. . . . .	81
Tabela 11 – AVALIAÇÃO DE MATURIDADE - SUPORTE ORGANIZACIONAL. . . . .	82
Tabela 12 – LEGENDA PARA PRIORIZAÇÃO DE AÇÕES. . . . .	91
Tabela 13 – PRIORIZAÇÃO DE AÇÕES POR ÁREA DE PROCESSO. . . . .	92
Tabela 14 – OKR 1. . . . .	97
Tabela 15 – OKR 2. . . . .	97
Tabela 16 – OKR 3. . . . .	97
Tabela 17 – OKR 4. . . . .	98
Tabela 18 – OKR 5. . . . .	98
Tabela 19 – OKR 6. . . . .	98
Tabela 20 – OKR 7. . . . .	98

# Lista de abreviaturas e siglas

UnB	Universidade de Brasília
TCC	Trabalho de Conclusão de Curso
MPS	Melhoria de Processo de Software
CMMI	Capability Maturity Model Integration
CMMI-DEV	Capability Maturity Model Integration Development
SRP-CMMI	Capability Maturity Model of Software Requirements Process and Integration
R-CMM	Requirements Capability Maturity Model
ISO	International Organization of Standardization
SEI	Software Engineering Institute
SPICE	Software Process Improvement and Capability Determination
MPS.BR	Melhoria de Processo do Software Brasileiro
MR-MPS-SW	Modelo de Referência MPS para Software
MA-MPS	Método de Avaliação
PDCA	Plan, Do, Check, Act
IDEAL	Initiating, Diagnosing, Establishing, Acting, Learning
XP	Extreme Programming
FDD	Feature Driven Development
WIP	Work In Progress
ER	Engenharia de Requisitos
BPMN	Business Process Model and Notation
OKR	Objectives and Key-Results

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
1.1	Contexto	16
1.2	Problema	17
1.3	Objetivo	17
1.4	Organização do Trabalho	18
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>19</b>
2.1	Considerações Iniciais do Capítulo	19
2.2	Engenharia de Requisitos	19
2.3	Processo Ágil em Desenvolvimento de Software na Engenharia de Requisitos	21
2.3.1	Métodos de Desenvolvimento Ágil	22
2.3.1.1	Extreme Programming (XP)	22
2.3.1.2	Scrum	24
2.3.1.3	Feature Driven Development (FDD)	26
2.3.1.4	Kanban	28
2.4	Melhoria de Processos em Engenharia de Requisitos	29
2.4.1	Qualidade de Software	29
2.4.1.1	Qualidade de Produtos de Software	30
2.4.2	Qualidade de Processos de Software	33
2.4.3	Melhoria de Processos de Software	34
2.4.4	Capacidade e Maturidade de Processos	34
2.4.4.1	CMMI	35
2.4.4.2	MPS-BR/MR-MPS-SW	38
2.4.5	Modelos específicos para engenharia de requisitos	41
2.4.5.1	R-CMM	41
2.4.5.2	SRP-CMMI	45
2.4.5.3	Comparação R-CMM e SRP-CMMI	48
2.5	Modelos de Gestão de Melhoria de Processos	49
2.5.1	Modelo PDCA	49
2.5.2	Modelo IDEAL	50
2.5.3	Métricas de Software	52
2.5.3.1	OKR	53
<b>3</b>	<b>METODOLOGIA DE PESQUISA</b>	<b>55</b>
3.1	Considerações Iniciais do Capítulo	55

<b>3.2</b>	<b>Classificação da Metodologia de Pesquisa</b>	<b>55</b>
<b>3.3</b>	<b>Plano Metodológico</b>	<b>56</b>
3.3.1	Planejamento	56
3.3.2	Coleta de Dados	56
3.3.2.1	Procedimentos de Pesquisa	56
3.3.2.2	Técnicas de Coletas de Dados	57
3.3.3	Análise de Dados	58
3.3.4	Documentação dos Resultados	58
<b>4</b>	<b>PROPOSTA DO TRABALHO</b>	<b>60</b>
<b>4.1</b>	<b>Considerações Iniciais</b>	<b>60</b>
<b>4.2</b>	<b>Contexto Organizacional</b>	<b>60</b>
<b>4.3</b>	<b>Proposta de Solução</b>	<b>61</b>
4.3.1	Inicialização	61
4.3.2	Diagnóstico	61
4.3.3	Estabelecimento	62
4.3.4	Ação	62
4.3.5	Lições	62
<b>4.4</b>	<b>Cronograma</b>	<b>63</b>
<b>5</b>	<b>APLICAÇÃO DA PROPOSTA</b>	<b>65</b>
<b>5.1</b>	<b>Considerações Iniciais</b>	<b>65</b>
<b>5.2</b>	<b>Inicialização</b>	<b>65</b>
5.2.1	Necessidades de negócio e Estímulo para mudança	65
5.2.2	Definição dos Agentes	66
5.2.3	Expectativa de Benefício	66
5.2.4	Fornecimento da Estrutura	67
<b>5.3</b>	<b>Diagnóstico</b>	<b>67</b>
5.3.1	Caracterização do Estado Atual e Desejado	67
5.3.1.1	Baseline de Processo	67
5.3.1.2	Baseline de Maturidade do Processo	70
5.3.1.3	Coleta de Dados	71
5.3.1.3.1	Documentos coletados e utilizados para análise	72
5.3.1.3.2	Roteiro aplicado para entrevistas	72
5.3.1.4	Caracterização das Práticas do Modelo SRP-CMMI	75
5.3.1.4.1	Área de Processo - Elicitação	75
5.3.1.4.2	Área de Processo - Análise	76
5.3.1.4.3	Área de Processo - Especificação	79
5.3.1.4.4	Área de Processo - Validação	80
5.3.1.4.5	Área de Processo - Gestão de Requisitos	80

5.3.1.4.6	Área de Processo - Suporte Organizacional . . . . .	81
5.3.2	Recomendações . . . . .	87
<b>5.4</b>	<b>Estabelecimento</b> . . . . .	<b>88</b>
5.4.1	Definir ações de melhoria . . . . .	88
5.4.1.1	Área de Processo - Elicitação . . . . .	88
5.4.1.2	Área de Processo - Análise . . . . .	88
5.4.1.3	Área de Processo - Especificação . . . . .	89
5.4.1.4	Área de Processo - Validação . . . . .	89
5.4.1.5	Área de Processo - Gestão de Requisitos . . . . .	89
5.4.1.6	Área de Processo - Suporte Organizacional . . . . .	90
5.4.2	Estabelecimento de prioridades . . . . .	91
5.4.3	Planejamento de ações . . . . .	93
<b>5.5</b>	<b>Ação</b> . . . . .	<b>95</b>
5.5.1	Criação da Solução Técnica . . . . .	95
5.5.2	Plano de Medição . . . . .	97
<b>5.6</b>	<b>Lições</b> . . . . .	<b>99</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>100</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>102</b>



# 1 Introdução

## 1.1 Contexto

As mudanças que estão ocorrendo nos ambientes de negócios motivam as empresas a modificar estruturas organizacionais e processos produtivos, saindo da visão tradicional baseada em áreas funcionais em direção a redes de processos centrados no cliente. A competitividade depende, cada vez mais, do estabelecimento de conexões nessas redes, criando elos essenciais nas cadeias produtivas. Alcançar competitividade pela qualidade, para as empresas de software, implica tanto na melhoria da qualidade dos produtos de software e serviços correlatos, como dos processos de produção e distribuição de software (SOFTEX, 2021).

A melhoria de processos significa compreender os processos existentes e modificá-los para aumentar a qualidade do produto e/ou reduzir os custos e o tempo de desenvolvimento (SOMMERVILLE, 2019). E a Melhoria de Processos de Software é o campo de pesquisa e prática, que surgiu da necessidade de resolver problemas de desenvolvimento de software, crescentemente complexos e onipresentes (KAUTZ et al., 2004).

De acordo com PAULA FILHO (2019), as ações de garantia da qualidade na produção de software incluem uma atitude preventiva para evitar problemas de qualidade do produto. Corrigir erros é dispendioso e, por isso, o objetivo é incorporar mecanismos preventivos aos processos para reduzir o número de erros ao longo do projeto e diminuir o número de erros que precisarão ser corrigidos no futuro, seja durante o desenvolvimento do projeto ou durante a vida útil do produto.

Paulk et al. (1993) dizem que, conforme uma organização amadurece, o processo de software torna-se melhor definido e mais consistentemente implementado por toda a organização. Nota-se, então, a necessidade e influência dos conceitos de maturidade e capacidade da empresa para a melhoria de processos.

O grau de maturidade é caracterizado como o grau de competência, técnica e gerência de uma organização. O grau de capacidade, no entanto, é caracterizado como a descrição do alcance dos resultados esperados que podem ser atingidos por um processo de software (PAULK et al., 1993). Conhecendo esses atributos é possível entender o nível no qual a organização pertence, identificar as características necessárias para ela atingir um próximo nível e, então, elaborar um plano de melhoria do processo.

Entender o status da prática da engenharia de requisitos na indústria de software é um passo importante para pesquisadores e praticantes, com o objetivo de melhorar o processo de software (NIKULA et al., 2004).

A ER é uma importante disciplina do processo de desenvolvimento de software, suas atividades são dedicadas a identificar, definir e gerenciar os requisitos do produto a ser desenvolvido (RAHMAN; SAHIBUDDIN; IBRAHIM, 2011).

É identificada como um dos fatores-chave para a melhoria da qualidade de software (SOMMERVILLE; RANSOM, 2005), pois os problemas de requisitos são caros e afligem quase todos os sistemas e organizações de desenvolvimento de software. Na maioria dos casos, o melhor que se pode esperar é detectar erros ou omissões nos requisitos a tempo de contê-los antes que o produto seja lançado (SAWYER; SOMMERVILLE; VILLER, 1999).

Muitos projetos de software falharam por conta de um conjunto de requisitos mal definidos (EMAM; MADHAVJI, 1995). Dificilmente um processo de software pode manter os prazos, custos e a qualidade do produto sobre controle se os requisitos estão mal definidos e gerenciados (SAWYER; SOMMERVILLE; VILLER, 1998). O processo de levantamento e a definição deles deve ser eficaz.

## 1.2 Problema

A execução do processo de Engenharia de Requisitos costuma ser um desafio para a maioria das organizações. Normalmente, ele é executado, contudo, não respeita boas práticas e por isso precisa passar por melhorias.

Na empresa ATR (nome fictício), além dos problemas vividos em qualquer modelo de trabalho para desenvolvimento de software, nota-se, também, dificuldades em adequar o processo de Engenharia de Requisitos à rapidez e dinâmica que a organização vive.

O uso de métodos ágeis auxilia para que a velocidade de desenvolvimento e entrega dos produtos seja respeitada, porém, o processo de Engenharia de Requisitos da empresa não acompanha o ritmo das atividades e gera insatisfação nos clientes e retrabalho para as equipes de desenvolvimento.

Ao considerar a realidade vivida pela empresa ATR e a necessidade de melhorar seu processo de Engenharia de Requisitos, a questão de pesquisa deste trabalho é: “Como identificar oportunidades de melhoria no processo de engenharia de requisitos da empresa a partir da análise do processo?”

## 1.3 Objetivo

O objetivo geral deste trabalho é identificar oportunidades de melhoria no processo de Engenharia de Requisitos da empresa ATR e, posteriormente, propor melhorias para ele.

Para alcançar o objetivo geral, foram definidos os objetivos específicos:

- Mapear o processo de requisitos atual;
- Identificar formas de avaliação do processo de requisitos na empresa (Métodos, técnicas, modelos);
- Construir uma linha de base de maturidade do processo;
- Fazer o diagnóstico do processo atual, identificando oportunidades de melhoria;
- Propor uma solução técnica para as oportunidades de melhoria do processo atual de requisitos.

## 1.4 Organização do Trabalho

Este trabalho de conclusão de curso está organizado em cinco capítulos, que são detalhados a seguir:

- **Capítulo 1 - Introdução:** Neste capítulo foram apresentados o contexto do trabalho, com uma síntese da metodologia planejada, o problema de pesquisa e os objetivos deste trabalho;
- **Capítulo 2 - Referencial teórico:** Descreve os conceitos que fundamentam este trabalho. O capítulo é subdividido nas seções de qualidade de software, melhoria de processos de software, processo ágil em desenvolvimento de software e engenharia de requisitos.
- **Capítulo 3 - Metodologia de Pesquisa:** Apresenta o plano metodológico adotado de forma mais detalhada e caracteriza o objeto de estudo;
- **Capítulo 4 - Proposta do Trabalho:** Apresenta a proposta deste trabalho, desde a abordagem até a proposta de um estudo de processo de uma empresa e como propor melhorias nesse processo.
- **Capítulo 5 - Aplicação da Proposta:** Apresenta a estratégia para definição de uma proposta para a melhoria do processo de engenharia de requisitos da empresa ATR.

## 2 Referencial Teórico

### 2.1 Considerações Iniciais do Capítulo

O Capítulo 2 aborda o conceito de engenharia de requisitos, as etapas que a definem e o relaciona com processo ágil de desenvolvimento de software, melhoria de processos, qualidade de software, capacidade e maturidade de processos e seus modelos (SRP-CMMI e R-CMM) e com alguns modelos de melhoria (PDCA e IDEAL).

### 2.2 Engenharia de Requisitos

Uma das principais fontes de complicações em produtos de software está associada aos requisitos. Requisitos mal interpretados, mal especificados e mal administrados podem afetar negativamente o desempenho de produtos de software e representam uma das maiores razões para o insucesso em projetos de tecnologia da informação, de acordo com [Reinehr \(2020\)](#).

[Sommerville \(2019\)](#) diz que os requisitos de um sistema são as descrições dos serviços que o ele deve fornecer e as limitações à sua operação. Esses requisitos refletem as necessidades dos clientes de um sistema que atende a um propósito específico, como gerenciar um dispositivo, realizar um pedido ou localizar informações. O processo de identificação, análise, documentação e verificação desses serviços e limitações é conhecido como engenharia de requisitos (ER).

Segundo [Kerr \(2015\)](#), a engenharia de requisitos, que antes era ignorada, tornou-se uma área cada vez mais importante no processo de produção de software, análise de negócios e gestão de processos. Isso se deve ao fato de sua importância para cumprimento de prazos e orçamentos e para garantir que o sistema entregue atenda às necessidades dos clientes ser cada vez mais reconhecida.

Do ponto de vista do processo de software, a engenharia de requisitos é uma atividade essencial na engenharia de software que começa durante a fase de comunicação e continua na fase de modelagem. A engenharia de requisitos estabelece uma base sólida para o design e desenvolvimento do software. Sem ela, há uma grande chance do software resultante não atender às necessidades do cliente. Ela deve ser adaptada às necessidades específicas do processo, projeto, produto e das pessoas envolvidas. É crucial compreender que cada uma dessas etapas é realizada de forma iterativa à medida que a equipe do projeto e os interessados continuam a compartilhar informações sobre suas respectivas preocupações, como dito por [Pressman e Maxim \(2021\)](#).

Segundo [Sommerville \(2019\)](#), alguns problemas que ocorrem durante a engenharia de requisitos são resultados da falta de clareza na separação entre diferentes níveis de descrição. É importante fazer uma distinção entre os requisitos de usuário, requisitos abstratos de alto nível, e os requisitos de sistema, descrições detalhadas do que o sistema deve fazer. Esses requisitos de usuário e requisitos de sistema podem ser definidos como:

1. **Requisitos de usuário** são declarações, em uma linguagem natural somada a diagramas, dos serviços que se espera que o sistema forneça para os usuários e das limitações sob as quais ele deve operar. Esses requisitos podem variar de declarações amplas das características necessárias do sistema até descrições precisas e detalhadas da sua funcionalidade;
2. **Requisitos de sistema** são descrições mais detalhadas das funções, dos serviços e das restrições operacionais do sistema de software. O documento de requisitos de sistema (chamado às vezes de especificação funcional) deve definir exatamente o que deve ser implementado. Pode fazer parte do contrato entre o adquirente do sistema e os desenvolvedores de software. Esses requisitos são classificados frequentemente como funcionais ou não funcionais.

[Reinehr \(2020\)](#) diz que um requisito funcional é, como o próprio nome diz, uma funcionalidade requerida pelos *stakeholders* para cumprir algum objetivo de negócio. Representa o que os desenvolvedores deverão implementar para que os usuários possam realizar as suas atividades. Geralmente, os requisitos funcionais são expressos em frases do tipo “o sistema deve”. Por exemplo: “O sistema deve permitir que o usuário pague com cartão de débito ou crédito”.

[Reinehr \(2020\)](#) também descreve os requisitos não funcionais, dizendo que se referem a como o software vai operar sob determinadas circunstâncias e, geralmente, estão relacionados a questões como desempenho, disponibilidade, usabilidade, portabilidade, escalabilidade, etc. Eles também são chamados de requisitos de qualidade.

Sabendo o que são os requisitos de sistema, pode-se aprofundar na maneira como eles são definidos e na sua importância para o desenvolvimento do projeto ([KERR, 2015](#)).

A seguir, são explicadas as fases da engenharia de requisitos por [Gallotti \(2017\)](#) e ilustradas na Figura 1:

1. **Estudo de viabilidade** - é uma análise sobre o que o cliente (*stakeholder*) deseja e o que é possível ser feito em termos de hardware e software. É plausível do ponto de vista orçamentário? É rentável para ambas as partes? Esse estudo é geralmente um processo rápido;

2. **Licitação e análise de requisitos** - este é o momento para acordar os requisitos com seus clientes e usuários, comparando o seu projeto atual a softwares já existentes. E a etapa de criação de protótipos (estudaremos sobre protótipos adiante);
3. **Especificação de requisitos** — aqui, deve-se especificar e organizar os requisitos do projeto em conjuntos específicos, de acordo com análises;
4. **Validação de requisitos** - deve-se verificar se os requisitos são consistentes. Geralmente, erros são encontrados nesta etapa. É o momento de corrigi-los e tornar os requisitos mais constantes e à prova de falhas.

Figura 1 – ETAPAS DA ENGENHARIA DE REQUISITOS.

Fonte: [Reinehr \(2020\)](#)

## 2.3 Processo Ágil em Desenvolvimento de Software na Engenharia de Requisitos

[Sommerville \(2019\)](#) diz que as empresas atualmente trabalham em um cenário global em constante mudança e precisam se adaptar às novas oportunidades de mercado, alterações econômicas e concorrentes. O software é uma parte importante em quase todos os negócios, então é necessário desenvolvê-lo rapidamente para aproveitar novas oportunidades e competir. A entrega, desenvolvimento e correção rápidos são, portanto, os requisitos mais importantes para a maioria dos sistemas de negócio.

Nesse contexto, [Sommerville \(2019\)](#) também diz que estabelecer um conjunto completo e estável de requisitos de software é um desafio e à medida que os requisitos mudam

ou que problemas de requisitos são descobertos, o projeto ou a implementação do sistema precisam ser retrabalhados e testados novamente o mais rápido possível.

Os dois processos de desenvolvimento (Tradicional e Ágil) consistem em cinco fases: Requisitos de Software, Design de Software, Implementação de Software, Testes de Software e Manutenção de Software, mas elas são executadas de formas diferentes (JAIN; SHARMA; AHUJA, 2018).

A abordagem ágil é um método interativo e incremental, em que cada fase definida é executada em vários pequenos ciclos iterativos do processo de desenvolvimento de software e permite a interação do cliente durante o processo, com adaptabilidade de mudanças requeridas (JAIN; SHARMA; AHUJA, 2018).

Pressman e Maxim (2021) dizem que para tornar o processo de software ágil, ele deve ser planejado de maneira a permitir flexibilidade para a equipe, ajustar suas tarefas, seguir uma metodologia de desenvolvimento ágil, ser eficiente e priorizar entregas incrementais, entregando o software em funcionamento o mais rápido possível para o produto e ambiente desejados.

Assim, as ideias fundamentais, que norteiam o desenvolvimento ágil, levaram ao desenvolvimento dos métodos ágeis, projetados para sanar fraquezas, supostas e reais, da engenharia de software convencional (PRESSMAN; MAXIM, 2021).

### 2.3.1 Métodos de Desenvolvimento Ágil

Segundo Oivo (2002), os métodos ágeis surgiram como uma alternativa aos métodos tradicionais de desenvolvimento de software, que eram rigorosos e desconsideravam a humanização do processo. Eles se caracterizam por serem leves e iterativos, enfatizando a utilização de testes para validar requisitos, envolvimento ativo dos usuários e confiança no conhecimento intuitivo da equipe.

Na metodologia ágil, diferentes modelos são o *Scrum*, *Extreme Programming*, *Feature Driven Development*, *Kanban*, entre outros (JAIN; SHARMA; AHUJA, 2018).

#### 2.3.1.1 Extreme Programming (XP)

O processo XP é voltado ao desenvolvimento de produtos com requisitos não totalmente definidos e em constante mudança (BECK, 2004). O processo é apoiado em valores, princípios e práticas, cujo foco é o desenvolvimento de um produto que venha a atender os objetivos do cliente. Alguns adeptos do XP o definem como sendo a prática e a perseguição da mais clara simplicidade, aplicado ao desenvolvimento de software (TELES, 2017).

Segundo [Beck \(2004\)](#), o XP tem em seu fundo uma filosofia de desenvolvimento de software baseada em valores, como comunicação, feedback, simplicidade, coragem e respeito. Além disso, possui um conjunto de práticas, comprovadamente eficientes para melhorar o desenvolvimento de software, juntamente com princípios e técnicas complementares que ajudam a aplicar esses valores nas práticas. Por fim, existe uma comunidade que compartilha desses princípios e muitas dessas práticas.

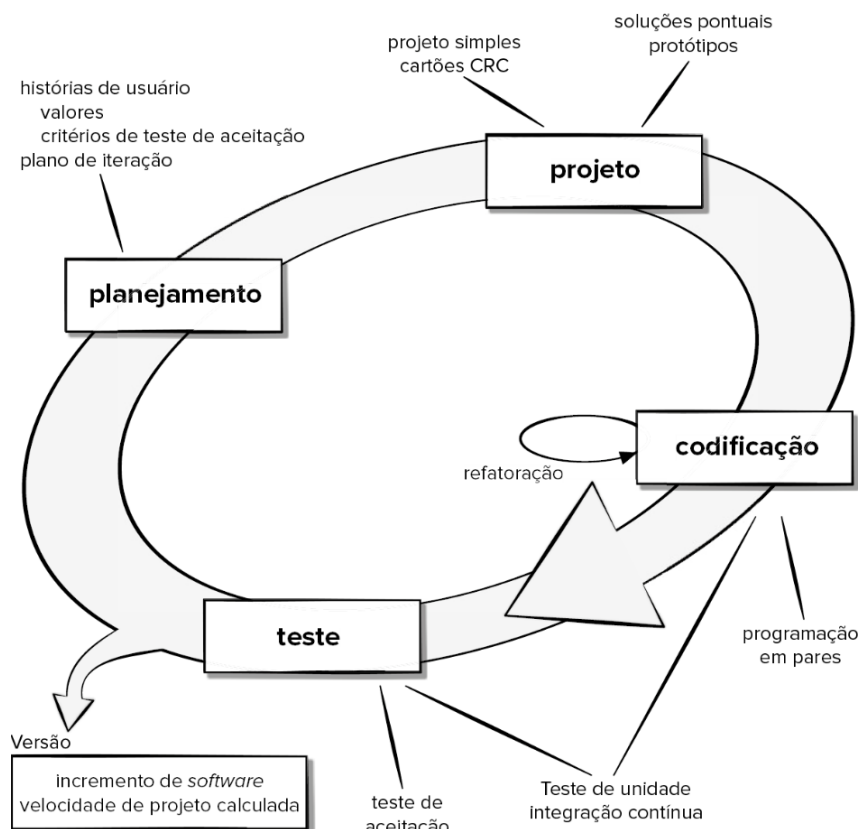
O *Extreme Programming* (Programação Extrema) envolve um conjunto de regras e práticas constantes no contexto de quatro atividades metodológicas: planejamento, projeto, codificação e testes ([PRESSMAN; MAXIM, 2021](#)).

De acordo com [Sommerville \(2019\)](#), no método XP, os requisitos são apresentados como histórias do usuário descritas em cenários. O desenvolvimento é feito diretamente a partir desses cenários, por uma série de tarefas. Os programadores trabalham em duplas e escrevem testes para cada tarefa antes de escrever o código. Antes de integrar o novo código ao sistema, todos os testes precisam ser executados com sucesso devido ao curto período entre lançamentos do sistema. O processo do XP é descrito pela Figura 2.

Os artefatos do XP são reduzidos para código e casos de teste. Não há reuniões de inspeção. O horizonte de planejamento é muito curto, então ferramentas avançadas de planejamento (como análise de ponto de função) são inúteis ([KONTIO; CONRADI, 2002](#)).



Figura 2 – O PROCESSO DO EXTREME PROGRAMMING (XP).



Fonte: [Pressman e Maxim \(2021\)](#)

### 2.3.1.2 Scrum

*Scrum* é um método de desenvolvimento ágil de software bastante popular concebido por Jeff Sutherland e sua equipe de desenvolvimento no início dos anos 1990 ([PRESSMAN; MAXIM, 2021](#)).

[Sommerville \(2019\)](#) diz que ele segue os princípios do manifesto ágil, no entanto, ele se concentra em proporcionar um arcabouço para a organização ágil do projeto e não impõe o uso de práticas de desenvolvimento específicas, como a programação em pares e o desenvolvimento com testes a priori (*test-first*).

Segundo [Hofmann et al. \(2018\)](#), o *scrum* é um *framework* leve com algumas regras rígidas. Existem três papéis em um time de *Scrum*: primeiro, o time de desenvolvimento multi-funcional, auto-empoderado e auto-organizado, composto por cinco a nove pessoas. Em seguida, o *Product Owner* (PO), responsável pelo conhecimento e visão do produto, é o único responsável pelo *product backlog*, pelo sucesso econômico e, conseqüentemente, pela priorização de funcionalidades e histórias de usuário. Finalmente, o *Scrum Master* é um líder de serviço para o time, garantindo a melhoria contínua, a comunicação e a remoção de obstáculos.

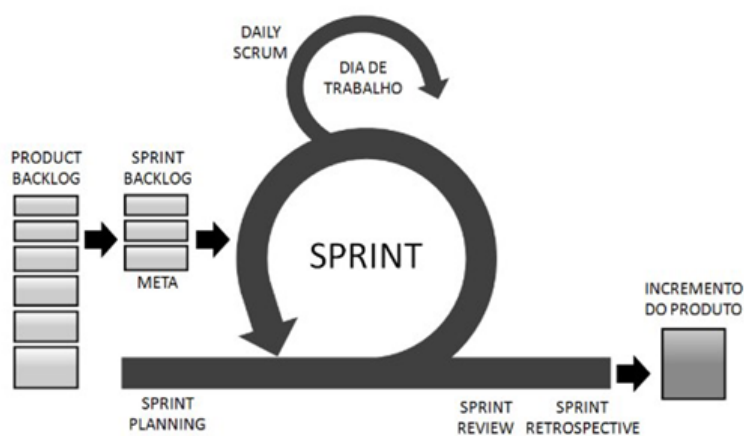
Hofmann et al. (2018) ainda diz que o *Scrum* segue uma abordagem iterativa com entregas rápidas e frequentes. Uma *sprint* (ciclo de desenvolvimento) é uma iteração onde um incremento funcional do produto é entregue. Antes de cada *sprint*, o time de desenvolvimento realiza uma sessão de planejamento chamada *Sprint Planning*, na qual se compromete com um conjunto selecionado de funcionalidades a serem implementadas na próxima *sprint*. Este conjunto é escolhido a partir do *backlog*, uma única lista ordenada que inclui todas as funcionalidades e requisitos desejados, baseados na visão do produto e priorizados pelo valor de negócio pelo *Product Owner*.

Segundo Pressman e Maxim (2021), a equipe de desenvolvimento cria um plano para fazer a entrega do incremento de software no tempo do *sprint*, com *time-box* de 2 a 4 semanas e negocia com o *product owner*.

Hofmann et al. (2018) diz que ao longo da *sprint*, há algumas cerimônias que são realizadas. Há uma reunião diária do *Scrum*, que é conduzida para garantir a sincronização do time e aumentar a transparência. Concluindo a *sprint*, há uma revisão das funcionalidades completas pelo *Product Owner*, seguida por uma sessão de retrospectiva para avaliar o trabalho do time, permitindo que eles expressem suas necessidades. É possível entender melhor sobre este ciclo na Figura 3.

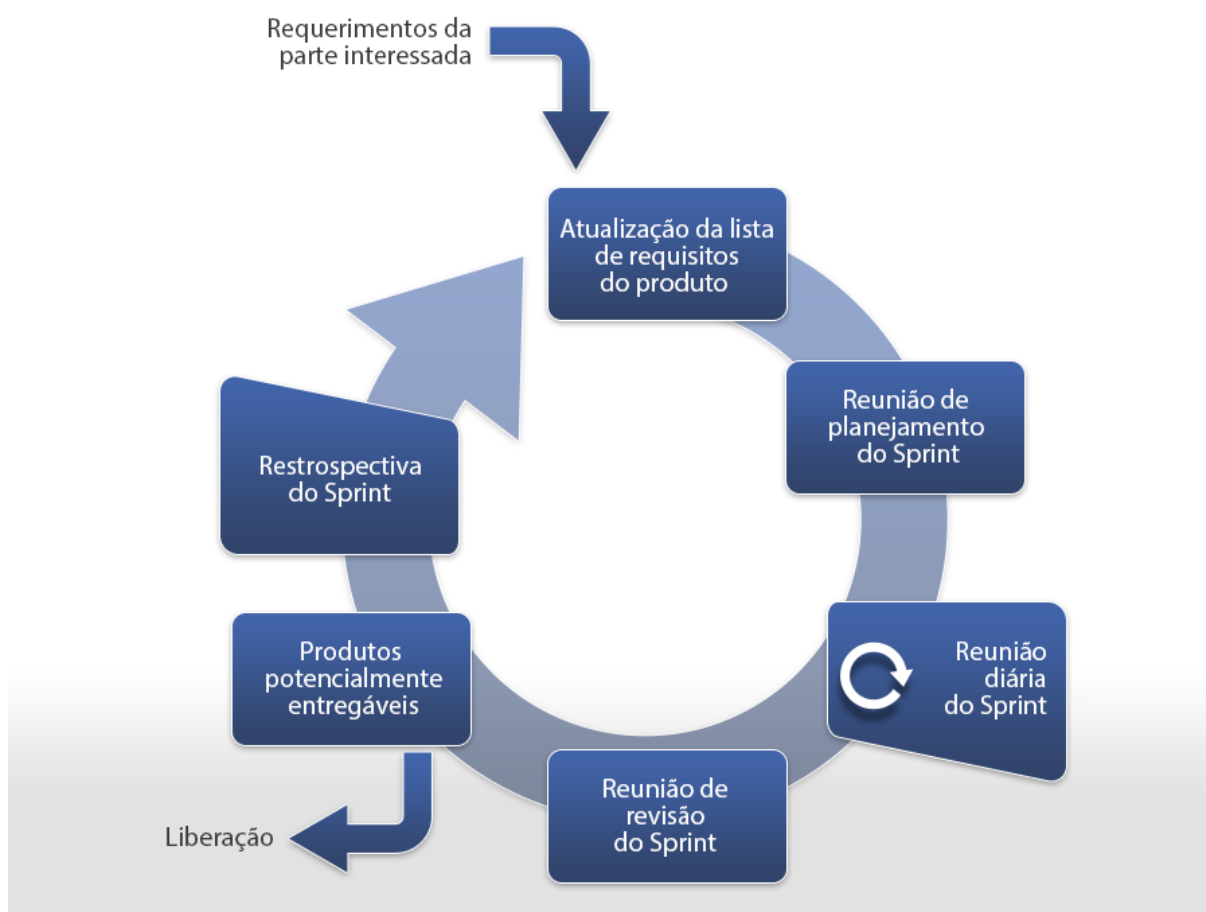
No *Scrum*, a engenharia de requisitos é especificada informalmente e depende muito do conhecimento e das habilidades dos indivíduos (ALHAZMI; HUANG, 2020). A Engenharia de Requisitos é performada iterativamente durante o processo de desenvolvimento, invés de ser uma fase inicial de duração fechada (SCHÖN, 2017), como é mostrado na Figura 4.

Figura 3 – PROCESSO DO SCRUM.



Fonte: Oliveira (2019)

Figura 4 – ENGENHARIA DE REQUISITOS NO SCRUM.



Fonte: [LERCHE-JENSEN \(2019\)](#)

### 2.3.1.3 Feature Driven Development (FDD)

O *Feature Driven Development* foi introduzido pela primeira vez em 1997. Jeff De Luca e Peter Coad trabalharam juntos em um time que foi formado para trabalhar em um projeto no banco United Overseas em Singapura ([CHOWDHURY; HUDA, 2011](#)).

FDD é um processo voltado para o cliente e orientado à modelagem, combinando algumas das melhores práticas de abordagem completa para Engenharia de Software orientada por objetos ([COAD; LUCA; LEFEBVRE, 1999](#)).

Ao seguir essa linha de pensamento, dentre as características comuns aos processos ágeis sempre existirão particularidades que os diferenciem ([LARMAN, 2003](#)).

Assim, [Palmer e Felsing \(2002\)](#) destaca as seguintes no FDD:

- Resultados úteis a cada duas semanas ou menos;
- Blocos bem pequenos de funcionalidade valorizados pelo cliente, chamados *features*;
- Planejamento detalhado e guia para medição;

- Rastreabilidade e relatórios com alta precisão;
- Monitoramento detalhado dentro do projeto, com resumos de alto nível para clientes e gerentes, tudo em termos de negócio; e
- Fornece uma forma de saber, dentro dos primeiros 10% de um projeto, se o plano e a estimativa são sólidos.

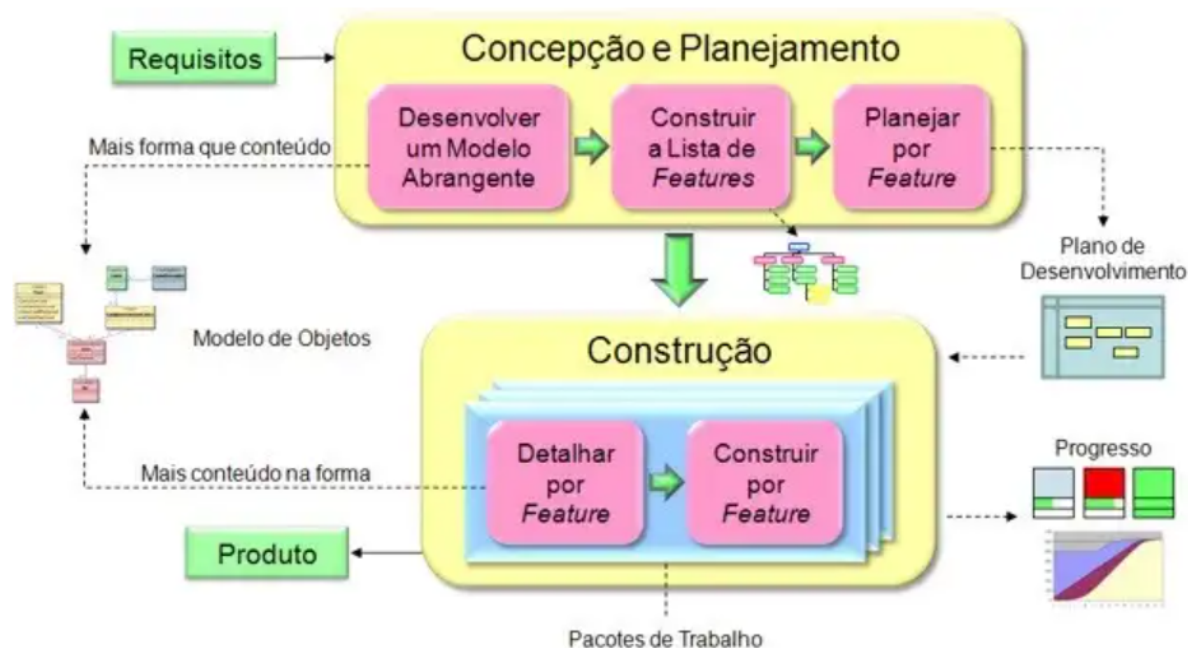
De acordo com [Martin \(2004\)](#), no *Feature Driven Development*, uma lista de características/funções é identificada, as relações funcionais entre as características são atribuídas e depois, em várias iterações de design e construção, a funcionalidade é desenvolvida em menos de duas semanas. O foco está no software funcional e a captura do conhecimento é baseada no conhecimento de uma pessoa, e não na documentação de uma organização.

A FDD é uma metodologia muito objetiva. Possui apenas duas fases; a Concepção e Planejamento, que se resume em pensar um pouco antes de fazer (tipicamente de 1 a 2 semanas) e a construção, que é fazer de forma iterativa (tipicamente em iterações de 2 semanas) ([ROBASKI, 2014](#)). As duas fases estão detalhadas na Figura 5.

O processo do FDD é composto por cinco etapas, sendo as três primeiras executadas uma única vez no início do processo e as duas últimas executadas a cada iteração ([ROBASKI, 2014](#)), como mostradas na Figura 5.

Para o FDD, a coleta e a análise de dados de requisitos funcionais e não funcionais deve ser executada em paralelo para assegurar o desenvolvimento de uma arquitetura apropriada, sem afetar a agilidade ([KANWAL; JUNAID; FAHIEM, 2010](#)). Este levantamento é feito na fase de concepção e planejamento.

Figura 5 – CICLO DE VIDA (FDD).



Fonte: Robaski (2014)

#### 2.3.1.4 Kanban

*Kanban* é uma técnica de produção enxuta originada pelo Sistema de Produção da Toyota e adaptado para a engenharia de software no final dos anos 2000 (POPPENDIECK; POPPENDIECK, 2003).

O método faz com que o gerenciamento de projetos de TI trabalhe eficientemente, implementando várias melhorias de fluxos de trabalho contínuos (AHMAD et al., 2016). O time de *Kanban* garante melhorias tanto fazendo pequenas mudanças, como implementando melhorias graduais no processo (N. Damij; T. Damij, 2021).

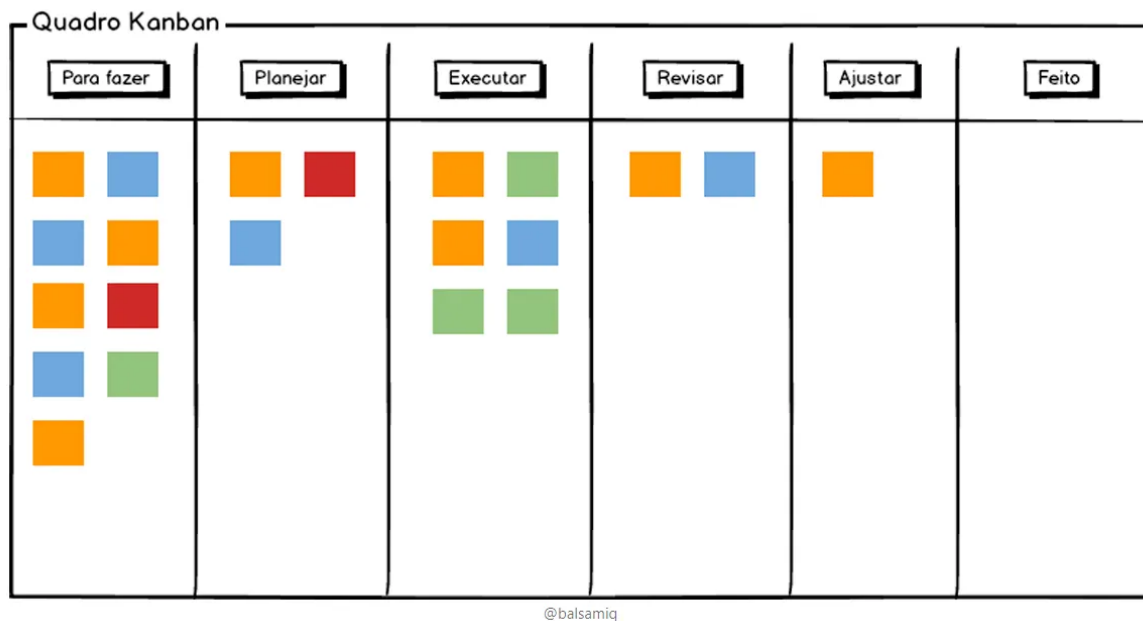
O *Kanban* tem cinco princípios fundamentais: visualizar o fluxo de trabalho, medir e gerir o fluxo, deixar as políticas do processo explícitas e usar modelos para reconhecer melhorias e oportunidades (YACOUB; MOSTAFA; FARID, 2016).

Os princípios do *Kanban* são aplicados com o uso de um quadro *Kanban*, que deixa o fluxo de atividade do processo visual em várias colunas. Cartões são utilizados para cada item de trabalho no quadro *Kanban* para mostrar seu estado atual (AHMAD et al., 2016).

Segundo Ahmad et al. (2016), o movimento dos itens de trabalho, ao longo do processo, é aprimorado, restringindo o WIP (*Work In Progress* - Trabalho em progresso) em cada etapa de atividade a um número máximo de itens que podem ser movidos para a etapa. Dessa forma, a equipe enxerga claramente seu fluxo de trabalho, controla a quantidade de itens WIP em cada fase e acompanha de maneira eficaz o tempo necessário para concluir o trabalho.

O *Kanban* não tem muitas restrições, ele é complementado por outros métodos, como *Scrum* e *XP* e se adequa a eles, assim como seu processo de engenharia de requisitos. É possível visualizar o fluxo do *Kanban* na Figura 6.

Figura 6 – EXEMPLO DE UMA IMPLEMENTAÇÃO DO QUADRO KANBAN.



Fonte: [Hernandes \(2020\)](#)

## 2.4 Melhoria de Processos em Engenharia de Requisitos

### 2.4.1 Qualidade de Software

O termo qualidade pode ser interpretado de formas diferentes e, para compreender efetivamente a Qualidade de Software, é crucial estabelecer um acordo quanto à sua definição ([ANDRADE, 2011](#)).

Segundo [Andrade \(2011\)](#), em geral, a Qualidade de Software pode ser descrita como o conjunto de características de um software que devem ser satisfatórias para que atenda às necessidades do usuário (sejam eles usuários finais, desenvolvedores ou organizações). A identificação dos atributos relevantes varia de acordo com o domínio da aplicação, as tecnologias utilizadas, as características específicas do projeto e as demandas do usuário e da organização.

[Vasconcelos e Oliveira \(2016\)](#) diz que, com o objetivo de se destacar no mercado de desenvolvimento de software, muitas empresas começaram a investir pesadamente para criar sistemas de alta qualidade. No entanto, com o passar do tempo, foi identificado que a qualidade dos produtos está diretamente ligada à qualidade dos processos usados para

sua criação. Por isso, houve um esforço para melhorar o processo de desenvolvimento de software e buscar a adoção de modelos e padrões de qualidade de processos de software com reconhecimento internacional, para garantir e avaliar que os sistemas desenvolvidos pela organização são sinônimo de excelência. Resumindo, pode-se afirmar que a qualidade de software, atualmente, pode ser medida com base em dois conceitos: qualidade de produtos e qualidade de processos.

Neste contexto, segundo [Vasconcelos e Oliveira \(2016\)](#), diversos modelos e normas internacionais de qualidade de produtos e de qualidade de processos de software foram definidos para atender as necessidades das empresas. No âmbito da qualidade de produtos, pode-se citar a norma ISO/IEC 25010:2014 e no de melhoria de processos de software, a ISO/IEC 12207:2017 ([ISO Central Secretary, 2017](#)), a ISO/IEC 33001:2015 ([ISO Central Secretary, 2015](#)) e o CMMI-DEV ([SEI, 2010](#)).

#### 2.4.1.1 Qualidade de Produtos de Software

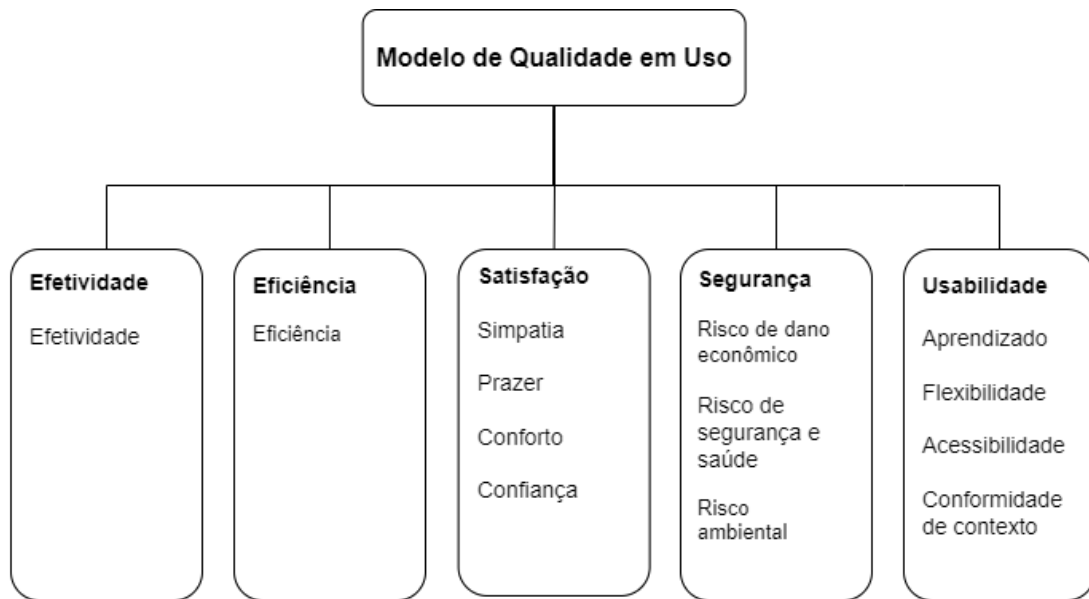
Da mesma forma como existem diversas interpretações para qualidade, de um modo geral, também existem para qualidade de um produto de software ([ANDRADE, 2011](#)).

O padrão ISO/IEC 25010:2011 ([ISO Central Secretary, 2011](#)) define:

a) Um modelo de qualidade em uso com cinco características (algumas das quais agrupadas em subcaracterísticas) que se relacionam com o resultado das interações quando o produto é usado em um contexto de uso específico. Este modelo de sistema pode ser aplicado a todo o sistema humano-computador, incluindo os sistemas de computador e produtos de software usados. Ele está representado na Figura 7.

b) Um modelo de qualidade do produto composto por oito características (divididas em subcaracterísticas) relacionadas a características estáticas de software e características dinâmicas de sistemas de computador. Este modelo é aplicável a sistemas de computador e produtos de software. Ele está representado na Figura 8.

Figura 7 – MODELO DE QUALIDADE EM USO.



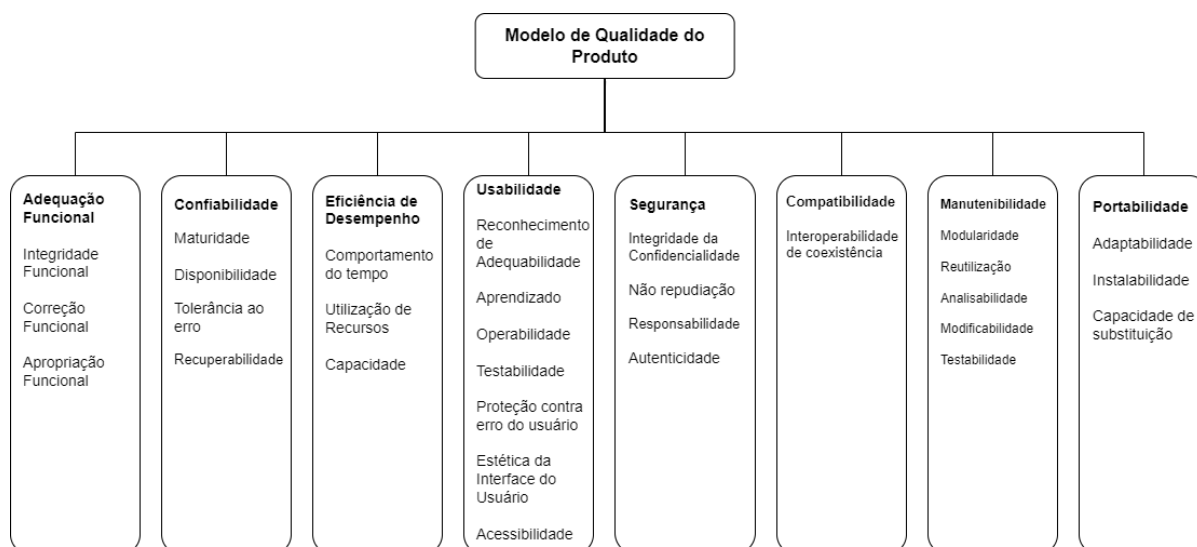
Fonte: Adaptado da ISO/IEC 25010:2011 (ISO Central Secretary, 2011)

A ISO/IEC 25010:2011 (ISO Central Secretary, 2011) descreve as características do modelo de qualidade em uso como:

- Efetividade refere-se à precisão e completude com que um usuário atinge um objetivo específico.
- Eficiência refere-se aos recursos gastos em relação à precisão e integridade com que os usuários atingem seus objetivos.
- Satisfação refere-se à satisfação das necessidades de um usuário ao usar um produto ou sistema em uma situação de uso particular.
- Segurança refere-se ao grau em que um produto ou sistema reduz os riscos potenciais à posição econômica, à vida humana, à saúde ou ao meio ambiente.
- Usabilidade é a medida em que um produto ou sistema pode ser usado de forma eficaz e eficiente, sem risco ou insatisfação, no contexto específico de uso e além das circunstâncias inicialmente explicitamente identificadas.



Figura 8 – MODELO DE QUALIDADE DO PRODUTO.



Fonte: Adaptado da ISO/IEC 25010:2011 (ISO Central Secretary, 2011)

A ISO/IEC 25010:2011 (ISO Central Secretary, 2011) também descreve as características do modelo de qualidade de produto como:

- Adequação funcional refere-se à medida que um produto ou sistema fornece funcionalidades que atendem aos requisitos declarados e implícitos quando usado sob condições especificadas;
- Confiabilidade refere-se ao grau em que um sistema, produto ou componente executa uma função especificada sob condições especificadas por um período de tempo especificado;
- Eficiência de Desempenho refere-se ao desempenho relativo à quantidade de recursos usados em condições específicas;
- Usabilidade refere-se à medida que um produto ou sistema pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso;
- Segurança refere-se à medida que um produto ou sistema protege os dados para fornecer aos indivíduos ou outros produtos ou sistemas um nível de acesso aos dados apropriado ao tipo e nível de autorização;
- Compatibilidade refere-se ao grau em que um produto, sistema ou componente pode fazer troca de informações com outros produtos, sistemas ou componentes e/ou exe-

cutar funções necessárias, enquanto compartilham o mesmo ambiente de hardware ou software;

- Manutenibilidade refere-se ao grau de eficácia e eficiência com o qual os mantenedores pretendidos podem modificar um produto ou sistema;
- Portabilidade refere-se ao grau de eficácia e eficiência com o qual um sistema, produto ou componente pode ser transferido de um hardware, software ou outro ambiente operacional, ou de uso para outro.

### 2.4.2 Qualidade de Processos de Software

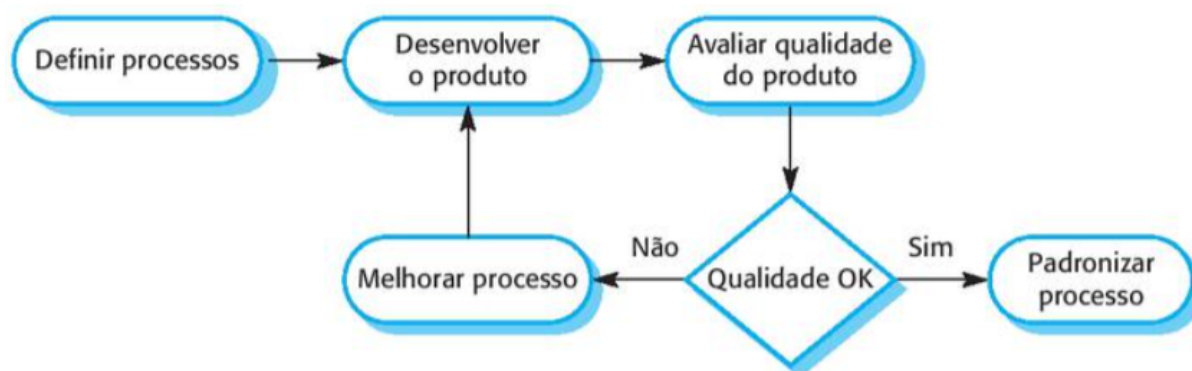
O processo de software pode ser definido como um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos necessários para conceber, desenvolver, implantar e manter um produto de software (FUGGETTA, 2000).

Pressman e Maxim (2021) dizem que, no contexto da engenharia de software, um processo não é uma prescrição rígida de como desenvolver um software. Ao contrário, é uma abordagem adaptável que possibilita às pessoas (a equipe de software) realizar o trabalho de selecionar e escolher o conjunto apropriado de ações e tarefas. A intenção é a de sempre entregar software dentro do prazo e com qualidade suficiente para satisfazer àqueles que patrocinaram sua criação e àqueles que vão utilizá-lo.

Segundo Sommerville (2019), embora a relação seja complexa, existe uma clara relação entre processo e qualidade do produto (figura 9). Como o design de software é um processo criativo, ele é altamente influenciado por habilidades e experiências individuais. Sem dúvida, o processo de desenvolvimento utilizado tem grande impacto na qualidade do software, e bons processos têm maior probabilidade de produzir software de alta qualidade.

A ISO 12207:2017 define uma estrutura comum para processos de ciclo de vida de software usando termos bem definidos aos quais a indústria de software pode se referir. Inclui processos, atividades e tarefas relacionadas à aquisição, fornecimento, desenvolvimento, operação, manutenção ou descarte de sistemas de software, produtos, serviços e partes de software de sistemas (internos ou externos a uma organização). Esses processos de ciclo de vida são executados por meio do engajamento das partes interessadas, com o objetivo final de alcançar a satisfação do cliente. Além disso, fornece processos para definir, controlar e melhorar os processos de ciclo de vida de software em uma organização ou projeto (ISO Central Secretary, 2017).

Figura 9 – QUALIDADE BASEADA EM PROCESSOS.



Fonte: Sommerville (2019)

### 2.4.3 Melhoria de Processos de Software

A definição da melhoria de processo é a criação de um software com alto índice de qualidade, que depende da qualidade do processo de produção, ou seja, quanto maior a qualidade do processo de produção, maior será a qualidade do programa final (GALLOTTI, 2017).

A melhoria do processo de software (MPS) preocupa-se com o fortalecimento das habilidades de desenvolvimento de software, o que significa que uma tarefa primária para a pesquisa de MPS é fortalecer as habilidades de gerenciamento de conhecimento para empresas de desenvolvimento de software (DINGSOYR, 2004).

Pressman e Maxim (2021) dizem que o termo melhoria de processo de software tem muitos significados. Primeiro, significa ser capaz de definir efetivamente os elementos de um processo de software eficaz. Em segundo lugar, a capacidade de avaliar as abordagens organizacionais existentes para o desenvolvimento de software em relação aos elementos. E em terceiro lugar, definir estratégias significativas de melhoria. A estratégia MPS transforma as abordagens existentes para o desenvolvimento de software em outras mais focadas, repetíveis e confiáveis (em termos de qualidade e entrega do produto).

### 2.4.4 Capacidade e Maturidade de Processos

Paulk et al. (1993) diz que um processo de software é composto de diferentes tarefas, estratégias, práticas e mudanças, que são utilizadas por pessoas para criar e manter o software e seus produtos complementares (como planos de projeto, documentos de design, código, casos de teste, manuais de usuários, etc). Conforme uma empresa se desenvolve, o processo de software se torna mais claramente definido e é consistentemente implementado por toda a organização.

O nível de maturidade dos processos de uma empresa de software é uma reflexão de sua gestão de processos, monitoramento e aderência às práticas recomendadas de engenharia de software. (SOMMERVILLE, 2019).

É a extensão para a qual um processo específico é explicitamente definido, gerenciado, medido, controlado, e eficaz. Maturidade implica um potencial de crescimento em capacidade e os dois indicam a riqueza do processo de software de uma organização e a consistência com que são aplicados em projetos por toda a instituição (PAULK et al., 1993).

A capacidade do processo reflete o nível de aperfeiçoamento e consolidação com que é executado. Em todas as etapas de maturidade, cada processo deve alcançar uma capacidade correspondente a um ou mais aspectos de processo, que resumem a eficiência na realização do processo (PAULA FILHO, 2019). Ela descreve o grau de resultados esperados, que podem ser alcançados seguindo o processo de software. A capacidade do processo de software de uma organização é um jeito de prever o resultado mais provável do próximo projeto de software que ela empreender (PAULK et al., 1993).

#### 2.4.4.1 CMMI

O modelos do CMMI (Capability Maturity Model Integration) são uma coleção de boas práticas que ajudam organizações a melhorar seus processos (SEI, 2010).

O modelo CMMI-DEV é um guia para aplicar boas práticas do CMMI em uma organização de desenvolvimento. As boas práticas do modelo focam em atividades para a qualidade do desenvolvimento de produtos e serviços para atender as necessidades dos clientes e usuários finais (SEI, 2010).

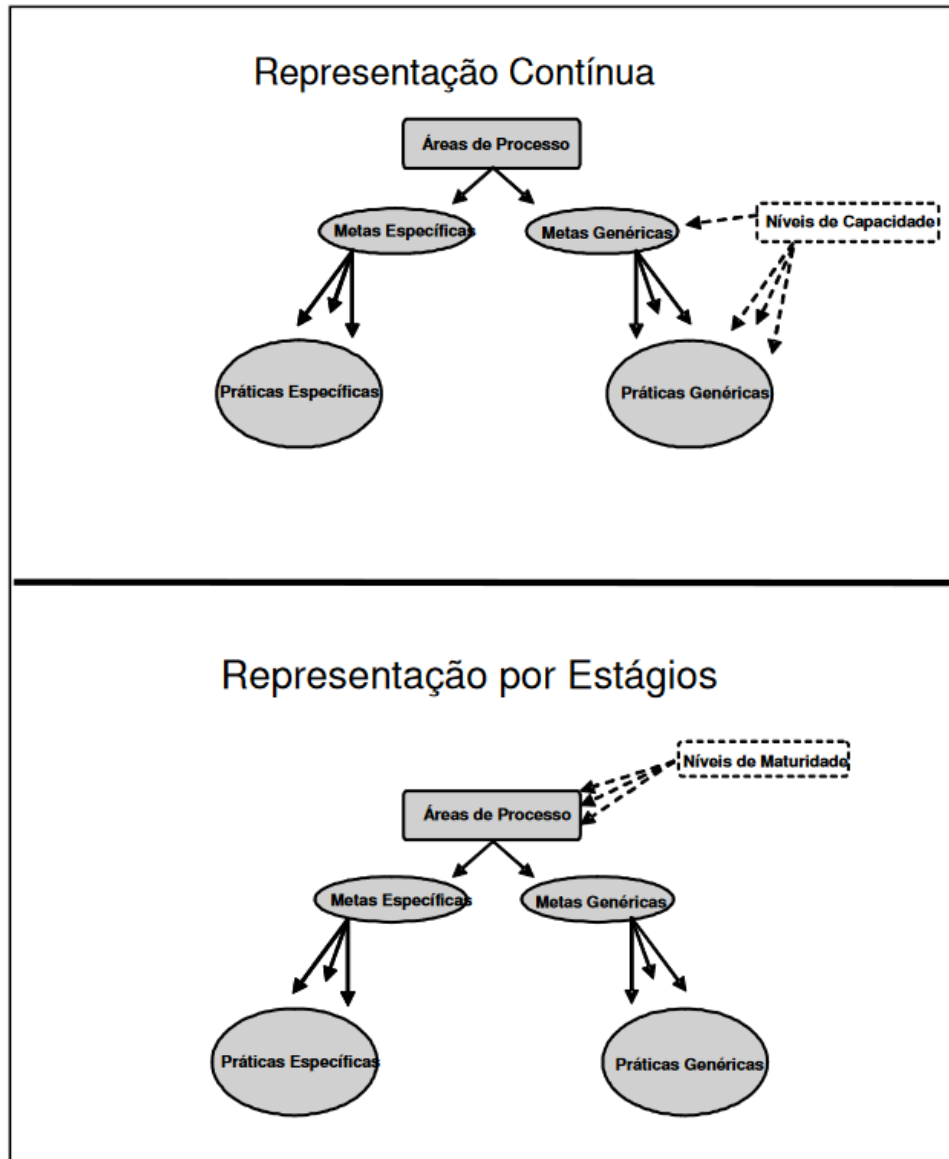
No CMMI-DEV 1.3, os níveis são usados para descrever caminhos de desenvolvimento recomendados para organizações que desejam melhorar os processos usados no desenvolvimento de seus produtos e serviços. Níveis podem, também, ser o resultado da atividade de classificação em avaliações. As avaliações podem ser aplicadas a toda a organização ou a grupos menores, como grupos de projetos (SEI, 2010).

De acordo com o SEI (2010), o CMMI oferece duas opções para a melhoria através do uso de níveis. A primeira opção permite que as organizações melhorem seus processos de software de forma incremental em uma área de processo individual selecionada pela organização. A segunda opção permite que as organizações melhorem um conjunto de processos relacionados, abordando incrementalmente conjuntos consecutivos de áreas de processo.

Ainda segundo o SEI (2010), estes dois caminhos de melhoria são associados a dois tipos de níveis: níveis de capacidade e de maturidade. Eles correspondem a duas abordagens de melhoria de processos, chamadas representações. Essas representações são

denominadas contínua e por estágios, representadas na Figura 10. A representação contínua possibilita que se alcancem os níveis de capacidade. A representação por estágios possibilita que se alcancem os níveis de maturidade.

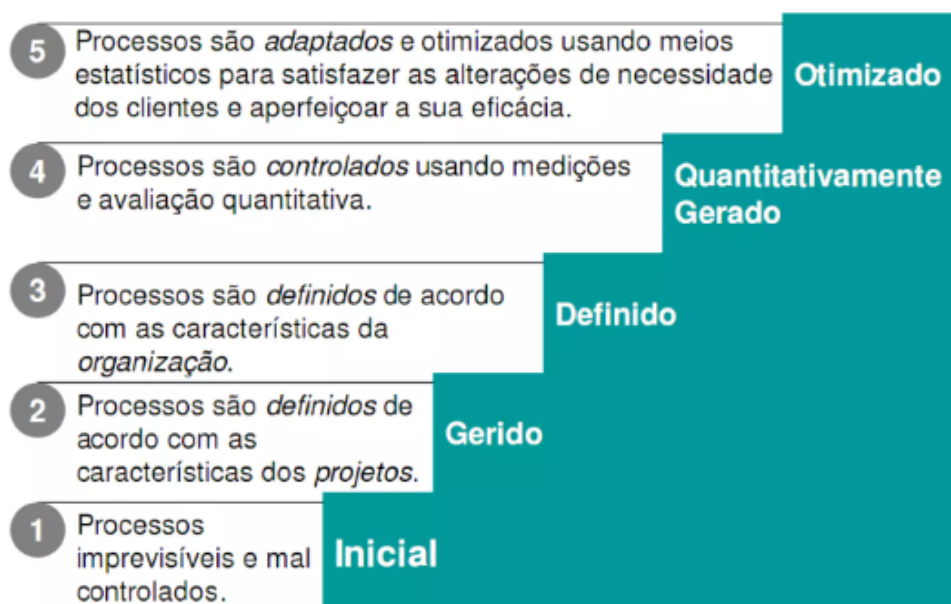
Figura 10 – ESTRUTURAS DAS REPRESENTAÇÕES CONTÍNUA E POR ESTÁGIOS.



Fonte: SEI (2006)

O CMMI-DEV estabelece cinco níveis de maturidade: Inicial, Gerenciado, Definido, Gerenciado Quantitativamente e Otimizado. Estes níveis são organizados de acordo com o que é mostrado na Figura 11.

Figura 11 – NÍVEIS DE MATURIDADE - CMMI-DEV.



Fonte: Castro (2012) adaptado do CMMI

O CMMI-DEV estabelece duas áreas de processo para requisitos: Requirements Development (RD) - Desenvolvimento de Requisitos e Requirements Management (REQM) - Gestão de requisitos.

Para o SEI (2010), a área de processo de desenvolvimento de requisitos busca compreender as necessidades do cliente e traduzi-las em exigências para o produto. Essas exigências são então analisadas para se chegar a uma solução conceitual geral. Em seguida, as exigências são divididas em subconjuntos para se estabelecer as necessidades iniciais para cada componente do produto.

Para esta área de processos, o SEI (2010) define metas específicas (SG) e práticas específicas (SP):

- SG 1 Requisitos de Desenvolvimento do Cliente
  - SP 1.1 Elicitar Necessidades
  - SP 1.2 Transformar as necessidades dos *stakeholders* em requisitos do cliente
- SG 2 Requisitos de Desenvolvimento do Produto
  - SP 2.1 Estabelecer requisitos do produto e dos componentes do produto
  - SP 2.2 Alocar os requisitos dos componentes do produto
  - SP 2.3 Identificar os requisitos de interface
- SG 3 Analisar e validar requisitos

- SP 3.1 Estabelecer conceitos operacionais e cenários
- SP 3.2 Estabelecer uma definição de funcionalidades requeridas e atributos de qualidade
- SP 3.3 Analisar requisitos
- SP 3.4 Analisar requisitos para atingir o equilíbrio
- SP 3.5 Validar Requisitos

A área de processo gestão de requisitos, segundo o [SEI \(2010\)](#), preserva os requisitos. Ela descreve os processos para identificar e controlar as alterações dos requisitos e garantir que outros planos relevantes e os prazos sejam atualizados. Além disso, permite o rastreamento dos requisitos do cliente até os requisitos do produto e, em seguida, até os requisitos dos componentes do produto.

Para esta área de processos, o [SEI \(2010\)](#) define uma meta específica e algumas práticas específicas:

- SG 1 Gerenciar Requisitos
  - SP 1.1 Entender os requisitos
  - SP 1.2 Obter comprometimento com os requisitos
  - SP 1.3 Gerenciar mudanças nos requisitos
  - SP 1.4 Manter uma rastreabilidade bidirecional dos requisitos
  - SP 1.5 Garantir o alinhamento entre o trabalho do projeto e os requisitos

#### 2.4.4.2 MPS-BR/MR-MPS-SW

O MPS.BR é um plano de longa duração, criado em 2003, pela Associação SOFTEX, com o apoio do Ministério MCTI, da Financiadora FINEP, do Serviço SEBRAE e do Banco Interamericano de Desenvolvimento (BID/FUMIN). Este programa tem como objetivo motivar e promover a excelência no software desenvolvido no Brasil ([SOFTEX, 2021](#)).

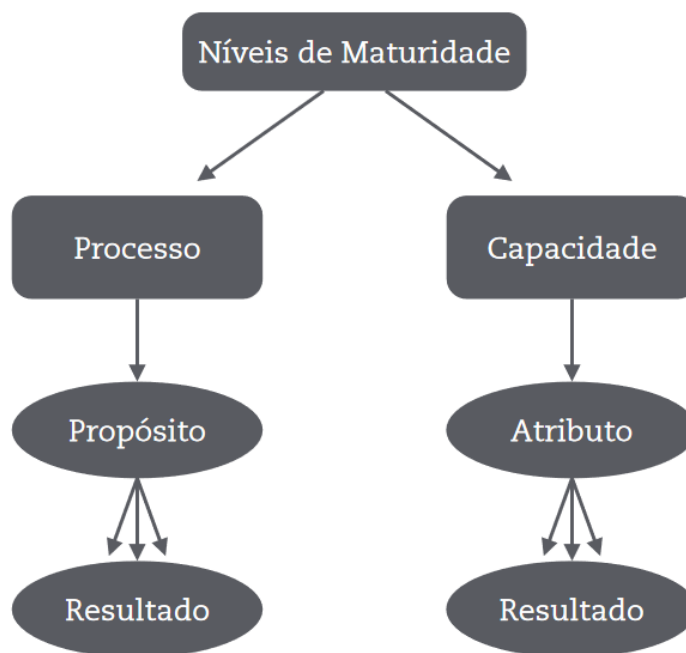
Conforme a [SOFTEX \(2021\)](#), o modelo MPS baseia-se no conceito de maturidade do processo e na capacidade de avaliar e melhorar a qualidade e eficiência do desenvolvimento de software e serviços relacionados e melhorar a qualidade e eficiência dos serviços prestados. Nesse contexto, o modelo MPS é composto por quatro componentes principais: o Modelo de Referência MPS para Software (MR-MPS-SW), o Modelo de Referência MPS para Serviços (MR-MPS-SV) e a Metodologia de Avaliação MPS (MA-MPS). Um modelo de negócios para melhorar os processos de software e serviços.

O Modelo de Referência MPS para Software define diferentes níveis de maturidade, que resultam da combinação entre os processos e sua capacidade. (SOFTEX, 2021).

A capacidade do processo é a caracterização da habilidade do processo para alcançar os objetivos de negócio, atuais e futuros; estando relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade (SOFTEX, 2021).

De acordo com a SOFTEX (2021), o MR-MPS-SW estabelece sete níveis de maturidade, que são: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado), é possível visualizar esses níveis na Figura 13. A escala começa no nível G e progride até o nível A. Cada nível tem um perfil de processos que indica onde a organização deve melhorar. O progresso e a conquista de um determinado nível de maturidade são alcançados quando os objetivos e todos os resultados esperados dos processos correspondentes são atendidos e os resultados esperados dos atributos de processo estabelecidos para aquele nível são alcançados. A Figura 12 deixa esta explicação clara.

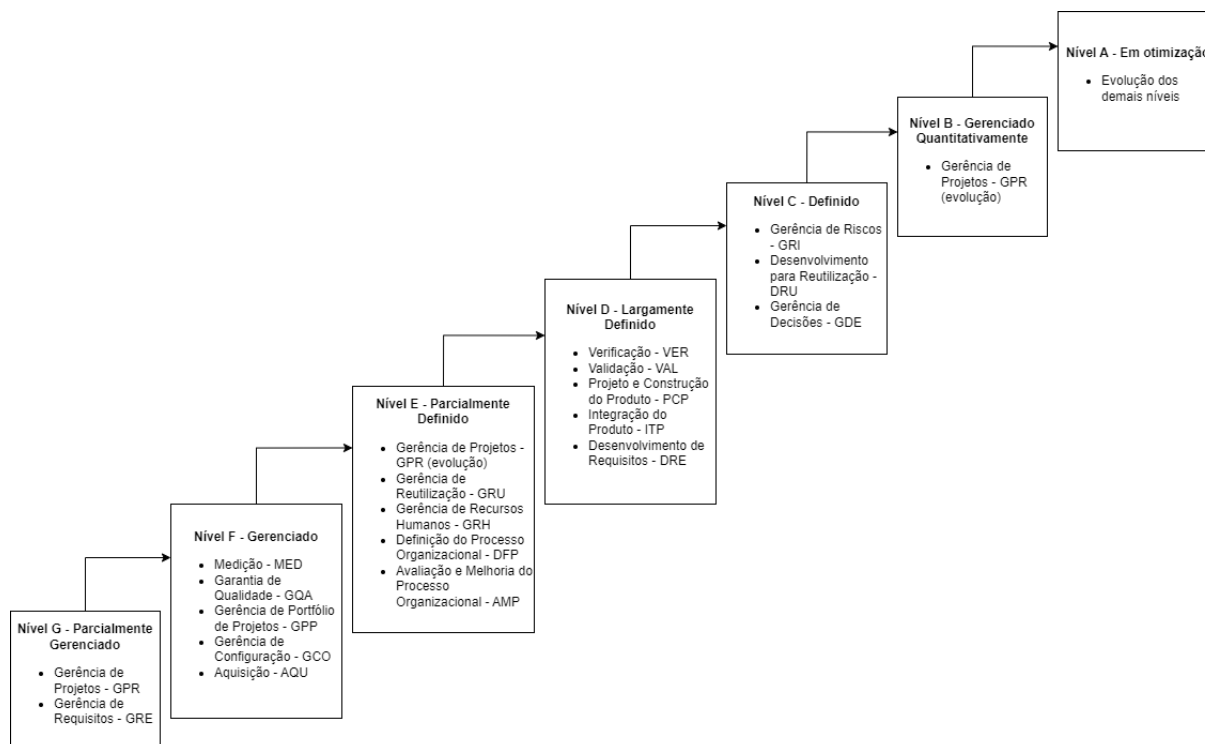
Figura 12 – ESTRUTURA DO MR-MPS-SW.



Fonte: MPS.BR SOFTEX (2021)



Figura 13 – NÍVEIS DE MATURIDADE MPS.BR.



Fonte: Adaptado do MPS.BR (SOFTEX, 2021)

O MPS.BR possui dois processos de requisitos definidos em seus níveis de maturidade. Os processos de Desenvolvimento de Requisitos (DRE) e Gerência de Requisitos (GRE) estão alocados nos níveis D e G, respectivamente.

O propósito do processo Gerência de Requisitos é gerenciar os requisitos do produto e dos componentes do produto do projeto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto (SOFTEX, 2021).

O MPS.BR (SOFTEX, 2021) define seus resultados esperados como:

- GRE 1. O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;
- GRE 2. Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido;
- GRE 3. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
- GRE 4. Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
- GRE 5. Mudanças nos requisitos são gerenciadas ao longo do projeto.

O propósito do processo Desenvolvimento de Requisitos é definir os requisitos do cliente, do produto e dos componentes do produto (SOFTEX, 2021).

O MPS.BR (SOFTEX, 2021) define seus resultados esperados como:

- DRE 1. As necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces, são identificadas;
- DRE 2. Um conjunto definido de requisitos do cliente é especificado e priorizado a partir das necessidades, expectativas e restrições identificadas;
- DRE 3. Um conjunto de requisitos funcionais e não-funcionais, do produto e dos componentes do produto que descrevem a solução do problema a ser resolvido, é definido e mantido a partir dos requisitos do cliente;
- DRE 4. Os requisitos funcionais e não-funcionais de cada componente do produto são refinados, elaborados e alocados;
- DRE 5. Interfaces internas e externas do produto e de cada componente do produto são definidas;
- DRE 6. Conceitos operacionais e cenários são desenvolvidos;
- DRE 7. Os requisitos são analisados, usando critérios definidos, para balancear as necessidades dos interessados com as restrições existentes;
- DRE 8. Os requisitos são validados.

### 2.4.5 Modelos específicos para engenharia de requisitos

A crescente importância do processo de engenharia de requisitos de software e da melhoria de processos no ciclo de vida de software originou vários modelos específicos de melhoria de processos de engenharia de requisitos, como o Requirements Capability Maturity Model (R-CMM) (BEECHAM; HALL; AUSTEN, 2005 apud NAJJAR; AL-SARAYREH, 2015) e o Capability Maturity Model of Software Requirements Process and Integration (SRP-CMMI) (NAJJAR; AL-SARAYREH, 2015).

#### 2.4.5.1 R-CMM

O R-CMM (Requirements Capability Maturity Model - Modelo de Capacidade e Maturidade de Requisitos) foi elaborado sobre a estrutura de melhoria estabelecida do Modelo de Maturidade de Capacidade de Software SEI (SW-CMM) (BEECHAM; HALL; AUSTEN, 2005).

De acordo com Beecham, Hall e Austen (2005), o processo de Engenharia de Requisitos pode ser combinado com o Modelo Integrado de Capacidade e Maturidade em

Software (SW-CMM) para melhorar as atividades de processo de desenvolvimento de software. Além disso, o Modelo de Capacidade e Maturidade de Engenharia de Requisitos (R-CMM) pode ser utilizado sozinho ou junto com o SW-CMM para avaliar a eficiência do processo de Engenharia de Requisitos.

O R-CMM inclui um método de avaliação que auxilia o usuário a entender seu atual processo de engenharia de Requisitos, visando priorizar a implementação do processo em relação às metas de maturidade (BEECHAM; HALL; AUSTEN, 2005).

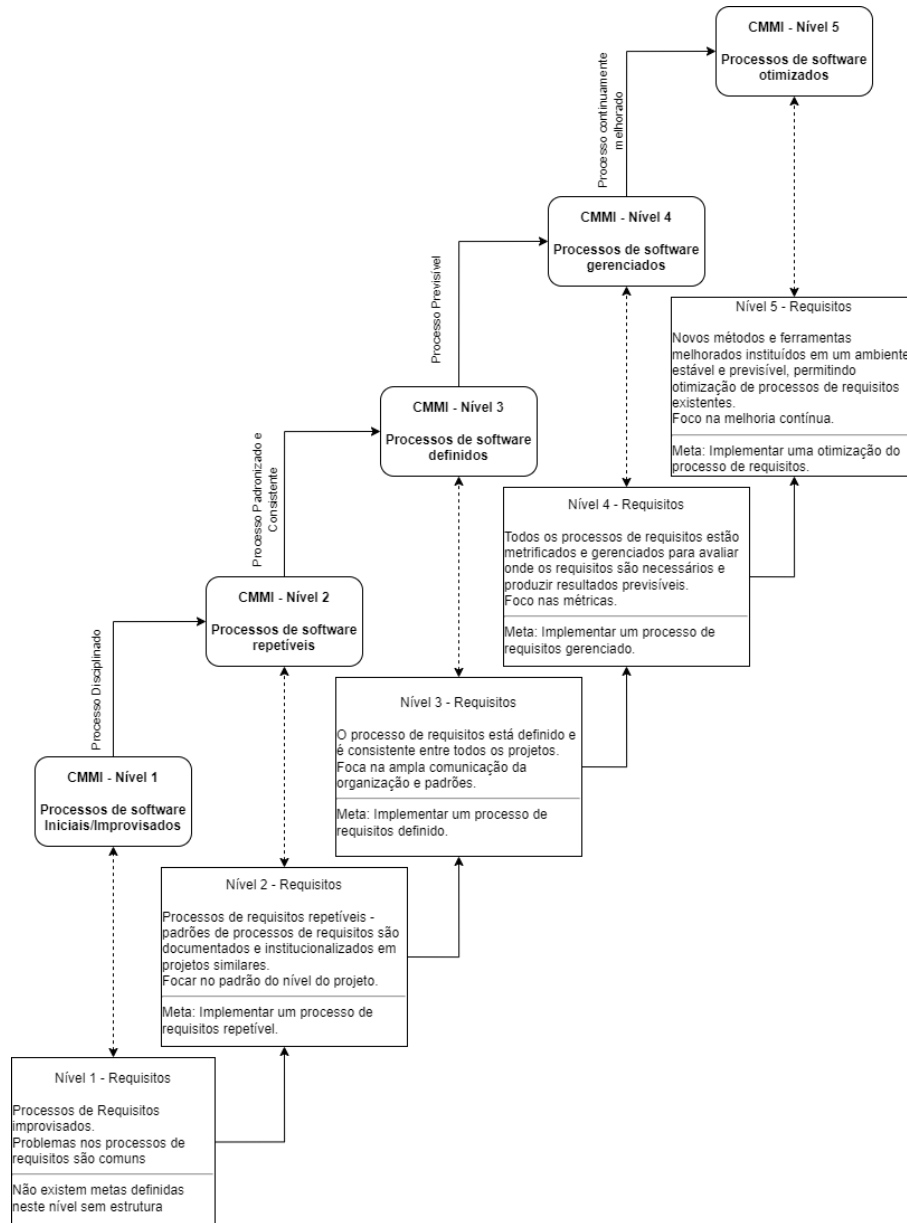
Conforme Beecham, Hall e Austen (2005), o R-CMM utiliza os pontos fortes do SW-CMM para formar um modelo especializado de melhores práticas que seja familiar, se integra com processos de software relacionados e tenha uma metodologia comprovada. Desta forma, o R-CMM ajuda os profissionais a definir suas prioridades e orientá-los na resolução de problemas de ER no contexto do desenvolvimento de software.

Para manter a apresentação limpa e utilizável, o R-CMM conecta o processo de ER aos níveis de maturidade, mas não é uma representação exhaustiva. Ele está dividido em subprocessos chave e cada subprocesso é definido e avaliado (BEECHAM; HALL; AUSTEN, 2005).

Beecham, Hall e Austen (2005) dizem que o R-CMM foi projetado para ajudar os profissionais a melhorar seu processo de Engenharia de Requisitos, implementando subprocessos (ou práticas recomendadas) de forma lógica. A estrutura de níveis do R-CMM ilustra como o processo de ER evolui de um nível não definido a um nível de melhoria contínua. Além disso, o modelo indica como cada nível do R-CMM tem um objetivo claro para auxiliar as empresas a se concentrarem em melhorar suas atividades. É possível visualizar estes níveis pela Figura 14.

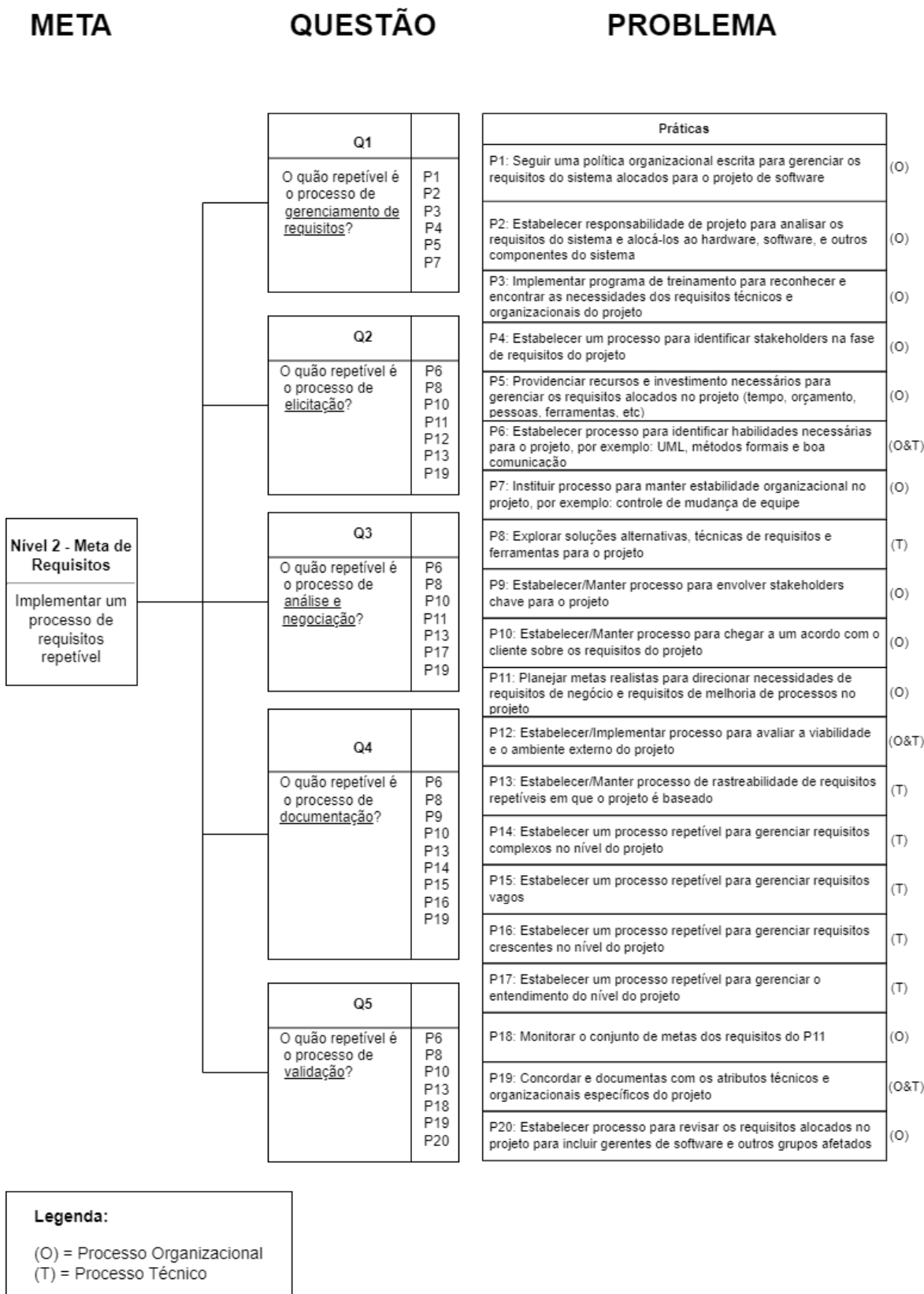
Para cada nível do R-CMM, Beecham, Hall e Austen (2005) fazem, ainda, uma adaptação do GQM (Goal-Question-Metric) e definem práticas para a melhoria do processo. É possível ver isso na Figura 15, que exemplifica essa adaptação para o nível 2.

Figura 14 – OS CINCO NÍVEIS DO FRAMEWORK R-CMM.



Fonte: Adaptada de Beecham, Hall e Austen (2005)

Figura 15 – NÍVEL 2 R-CMM.



Fonte: Adaptada de Beecham, Hall e Austen (2005)

#### 2.4.5.2 SRP-CMMI

Najjar e Al-Sarayreh (2015) dizem que o objetivo principal do SRP-CMMI é ajudar as organizações a melhorar e avaliar seus processos de engenharia de requisitos. Ele visa ajudar os praticantes a definir, compreender e aplicar de maneira eficiente suas atividades do processo de engenharia de requisitos, baseadas nas etapas do CMMI.

O SRP-CMMI é um padrão baseado no modelo de capacidade e maturidade CMMI-DEV, e basicamente consiste em três níveis de maturidade, que são adaptações dos níveis de capacidade do CMMI, numerados de 0 a 2: Incompleto, Executado, Gerenciado (NAJJAR; AL-SARAYREH, 2015).

Segundo Najjar e Al-Sarayreh (2015), como o modelo CMMI, o SRP-CMMI define um conjunto de áreas de processo distribuídas em níveis de maturidade. Cada área de processo possui objetivos de ER, que estão relacionados à prática de ER e são descrições das atividades necessárias para atingir os objetivos associados.

Cada prática de ER no SRP-CMMI consiste nos componentes a seguir:

- **Propósito:** Uma declaração que explica a meta que as práticas pretendem atingir;
- **Sub-práticas:** Uma declaração que providencia uma descrição detalhada de como interpretar e implementar uma prática específica ou genérica;
- **Técnica:** Uma lista de todas as técnicas que foram utilizadas para performar as práticas de ER;
- **Produto de Trabalho:** Uma lista de saídas que resultaram das práticas de ER;
- **Elaboração:** Detalhes extras de informação das práticas de ER.

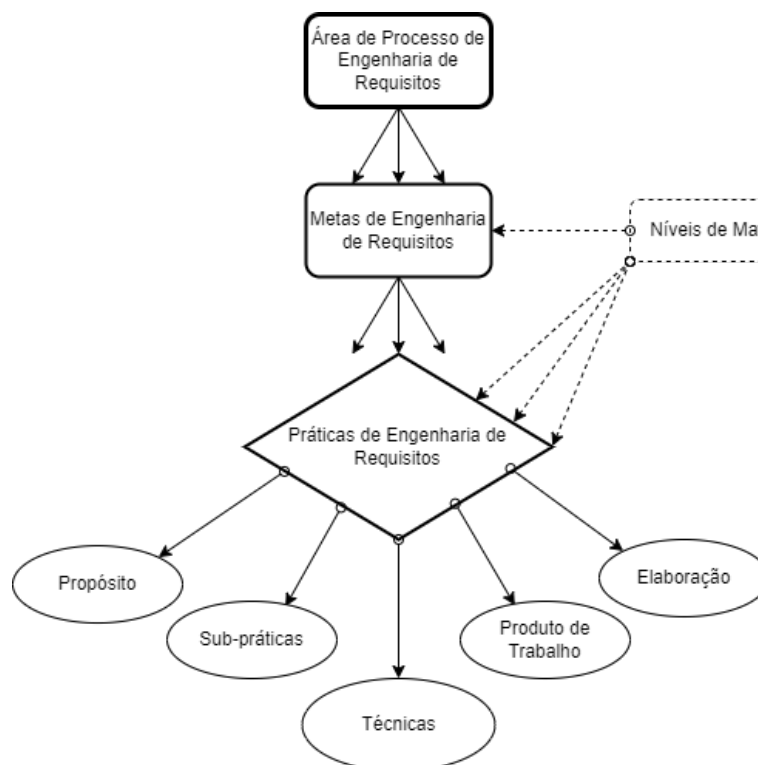
Najjar e Al-Sarayreh (2015) também definem que o SRP-CMMI possui diferentes níveis de maturidade em ER, cada um composto por uma área de processo de ER e suas metas e práticas correlatas. Alcançar um nível específico de maturidade em ER depende da implementação bem-sucedida de todas as práticas relacionadas a ER. Os três níveis de maturidade fornecidos pelo modelo oferecem uma descrição aprofundada de como a organização pode avaliar e melhorar seus processos de ER. Este processo é detalhado na Figura 16. Os três níveis de maturidade são definidos a seguir e podem ser visualizados na Figura 17:

- **Nível 0 - Processo incompleto de ER (Não há figura para ilustrar, porque assume-se que quase nenhuma ou nenhuma prática é implementada):** É similar ao nível 0 de capacidade do CMMI-DEV, onde não é executado e nem parcialmente executado, em outras palavras, existem uma ou mais práticas de ER não

implementadas. Considerando que não há razão para institucionalizar um processo de ER parcialmente executado, este nível não possui uma meta de ER;

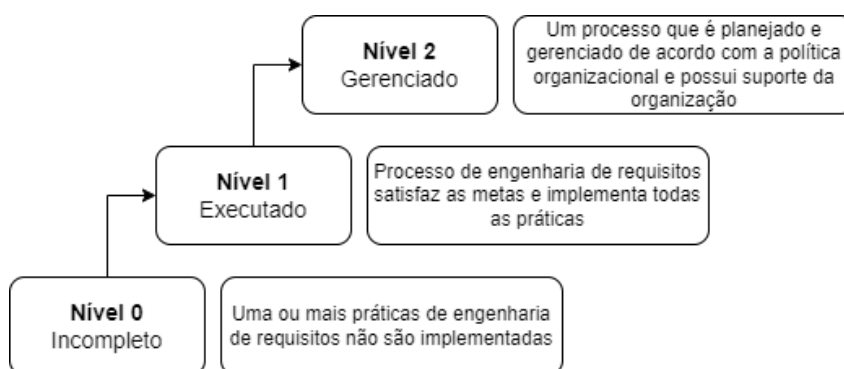
- **Nível 1 - Processo de ER executado (Práticas detalhadas na Figura 18):** Satisfaz as metas de ER e implementa todas as práticas de ER da área de processo. Neste nível de maturidade, o SRP-CMMI define cinco áreas de processo de ER: elicitação, análise, especificação, validação e gestão de requisitos, onde os requisitos são coletados, analisados, priorizados, documentados, validados e as mudanças e a rastreabilidade de requisitos são geridas. As práticas de ER neste nível são, em sua maioria, desenvolvidas baseadas no CMMI Gerenciado. Cada área de processo tem uma meta de ER e uma prática de ER relacionadas;
- **Nível 2 - Processo de ER gerenciado (Práticas detalhadas na Figura 19):** É um processo que é planejado e executado, conforme a política da organização. O SRP-CMMI define as áreas de processo de suporte organizacional neste nível de maturidade, onde a política de ER é estabelecida, *stakeholders* relevantes são envolvidos, recursos são alocados, pessoas são treinadas, responsabilidades são atribuídas e o processo de ER aderido é evoluído. As práticas de ER nesta área de processo são construídas em referência a quatro das áreas de processo do CMMI, chamadas de Planejamento do Projeto, Treinamento Organizacional, Gestão do Projeto e Controle e Garantia da Qualidade do Processo e do Produto.

Figura 16 – COMPONENTES DO SRP-CMMI.



Fonte: Adaptado de [Najjar e Al-Sarayreh \(2015\)](#)

Figura 17 – NÍVEIS DE MATURIDADE DO SRP-CMMI.



Fonte: Adaptado de Najjar e Al-Sarayreh (2015)

Figura 18 – PRÁTICAS NÍVEL 1 SRP-CMMI.

Nível	Área do Processo	Meta	Práticas
Nível 1 Executado	Elicitação	REG 1: Estabelecer um entendimento sobre o problema, soluções e stakeholders	REP 1.1: Identificar stakeholders e recursos de requisitos REP 1.2: Elicitar necessidades
		Análise	REG 1: Criar um modelo de análise que identifica dados, funções, recursos, restrições e requisitos comportamentais, usando as informações obtidas durante a elicitación
	REG 2: Resolver conflitos, priorizar requisitos e identificar riscos, visando obter um resultado positivo antes de prosseguir para as próximas atividades de engenharia de software		REP 2.1: Resolver conflitos REP 2.2: Priorizar requisitos REP 2.3: Obter comprometimento com os requisitos
	Especificação		REG 1: Formalizar os requisitos informativos, funcionais e comportamentais
	Validação	REG 1: Validar requisitos para garantir que o produto resultante vai performar como planejado, no ambiente final do usuário	REP 1.1: Confirmar que os requisitos estão corretos, completos, consistentes, testáveis e que satisfazem as necessidades do cliente
	Gestão de Requisitos	REG 1: Identificar, controlar e acompanhar todas as mudanças que ocorrerem com os requisitos	REP 1.1: Gerenciar as mudanças dos requisitos
		REG 2: Garantir que todos os requisitos iniciais estão devidamente rastreados	REP 2.1: Gerenciar a rastreabilidade dos requisitos

Fonte: Adaptado de Najjar e Al-Sarayreh (2015)



Figura 19 – PRÁTICAS NÍVEL 2 SRP-CMMI.

Nível	Área do Processo	Meta	Práticas
Nível 2 Gerenciado	Suporte Organizacional	REG 1: Avaliar quanto de suporte organizacional é dado para as práticas de engenharia de requisitos	REP 1.1: Estabelecer uma Política Organizacional de Engenharia de Requisitos
			REP 1.2: Monitorar o envolvimento do stakeholder
			REP 1.3: Identificar recursos do projeto
			REP 1.4: Atribuir responsabilidade
		REG 2: Desenvolver habilidades e conhecimento, e treinar pessoas para poderem performar seus papéis de forma eficaz e eficiente	REP 2.1: Definir programas de treinamento
			REP 2.2: Promover treinamento
			REP 2.3: Estabelecer registro de treinamentos
			REP 2.4: Avaliar a eficácia dos treinamentos
		REG 3: Avaliar objetivamente a adesão em relação às descrições de processos, padrões e procedimentos aplicáveis	REP 3.1: Avaliar processos objetivamente
REP 3.2: Avaliar produtos de trabalho objetivamente			
REP 3.3: Promover recursos adequados			
REP 3.4: Estabelecer um processo para revisar requisitos alocados no projeto			

Fonte: Adaptado de [Najjar e Al-Sarayreh \(2015\)](#)

### 2.4.5.3 Comparação R-CMM e SRP-CMMI

Tendo como base os modelos específicos para engenharia de requisitos apresentados, pode-se fazer uma comparação a partir de alguns critérios, como ilustrado na Tabela 1:

Tabela 1 – TABELA DE COMPARAÇÃO R-CMM E SRP-CMMI.

Critérios de Comparação (ER)	R-CMM	SRP-CMMI
Ano de Criação	2005	2015
Número de níveis de maturidade	5	3
Cobre todas as etapas da ER	Sim	Sim
Número de práticas (aproximadamente)	+/- 30 por nível de maturidade	23 no total
Nível de simplicidade (número de níveis e de práticas)	Complexo	Simple
Nível de completude	Completo e detalhado	Completo

Fonte: Autora

A partir da comparação entre os dois modelos, é possível identificar que, para o objetivo deste trabalho, o modelo SRP-CMMI se encaixa melhor, pois tem um escopo menor e é mais atual.

## 2.5 Modelos de Gestão de Melhoria de Processos

### 2.5.1 Modelo PDCA

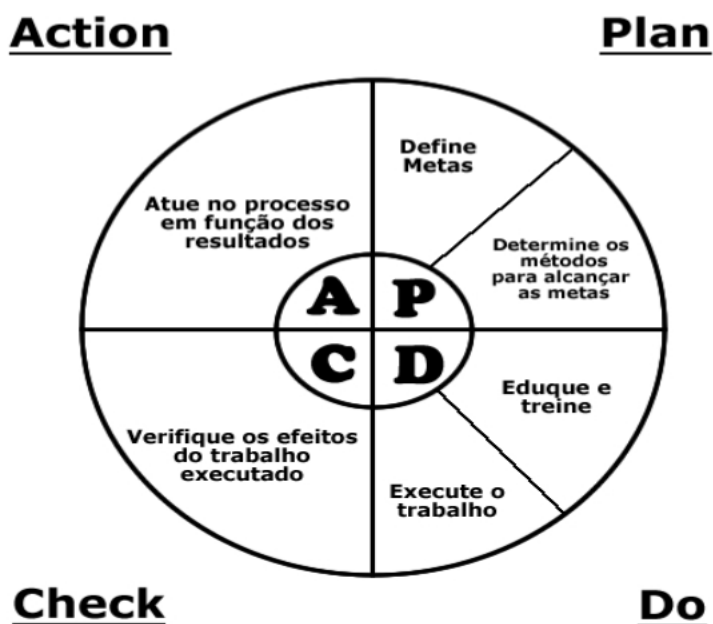
De acordo com [Andrade \(2011\)](#), os modelos de melhoria de processo de software utilizam estratégias para coordenar a melhoria contínua. Estas estratégias permitem compreender o estado atual do processo, desenvolver uma visão desejada do processo, estabelecer uma lista de ações prioritárias para melhoria e planejar a execução dessas ações, alocando recursos adequados.

[Vasconcelos e Oliveira \(2016\)](#) dizem que o “ciclo PDCA – Plan, Do, Check and Action” (Figura 20), é uma metodologia de gerenciamento iterativa de quatro etapas usada para gerenciar e melhorar continuamente processos e produtos. Um modelo de liderança circular facilita a adaptação organizacional e possibilita o aprendizado consciente.

[Salgueiro e Arte \(2005\)](#) descreve as fases do PDCA da seguinte forma:

- **Plan/Planejar:** A primeira fase é estabelecer objetivos e processos para atender aos requisitos dos clientes e às políticas da organização;
- **Do/Fazer:** A segunda fase é a implementação dos processos;
- **Check/Checar:** A terceira fase é monitorar, medir e relatar o desempenho dos processos e produtos em relação às políticas, objetivos e requisitos para o produto;
- **Act/Agir:** A última fase é tomar ações para melhorar continuamente o desempenho dos processos.

Figura 20 – CICLO PDCA.



Fonte: Campos (2014)

### 2.5.2 Modelo IDEAL

Além do modelo PDCA, pode-se utilizar também o modelo IDEAL. Este é o modelo mais aplicado com enfoque na melhoria de processo (GALLOTTI, 2017).

De acordo com Gremba e Myers (1997), o modelo IDEAL foi desenvolvido pelo Software Engineering Institute (SEI). O nome IDEAL corresponde aos 5 estágios do ciclo de melhoria de processo de software, que incluem avaliar o processo atual, elaborar um plano de melhoria, implementar o plano, verificar sua eficácia por meio de uma avaliação do processo e, assim, reiniciar o ciclo.

O modelo implementa uma estratégia de aperfeiçoamento contínuo do processo de desenvolvimento de software, o que auxilia as empresas a otimizar seu método de produção de software (ANDRADE, 2011).

PAULA FILHO (2019) traz as 5 fases abaixo:

- **Initiating/Inicialização** – lançar as bases para um programa bem-sucedido
- **Diagnosing/Diagnóstico** – determinar onde se está e aonde se quer chegar
- **Establishing/Estabelecimento** – planejar os detalhes de como chegar ao destino
- **Acting/Ação** – realizar os planos
- **Learning/Lições** – aprender com a experiência

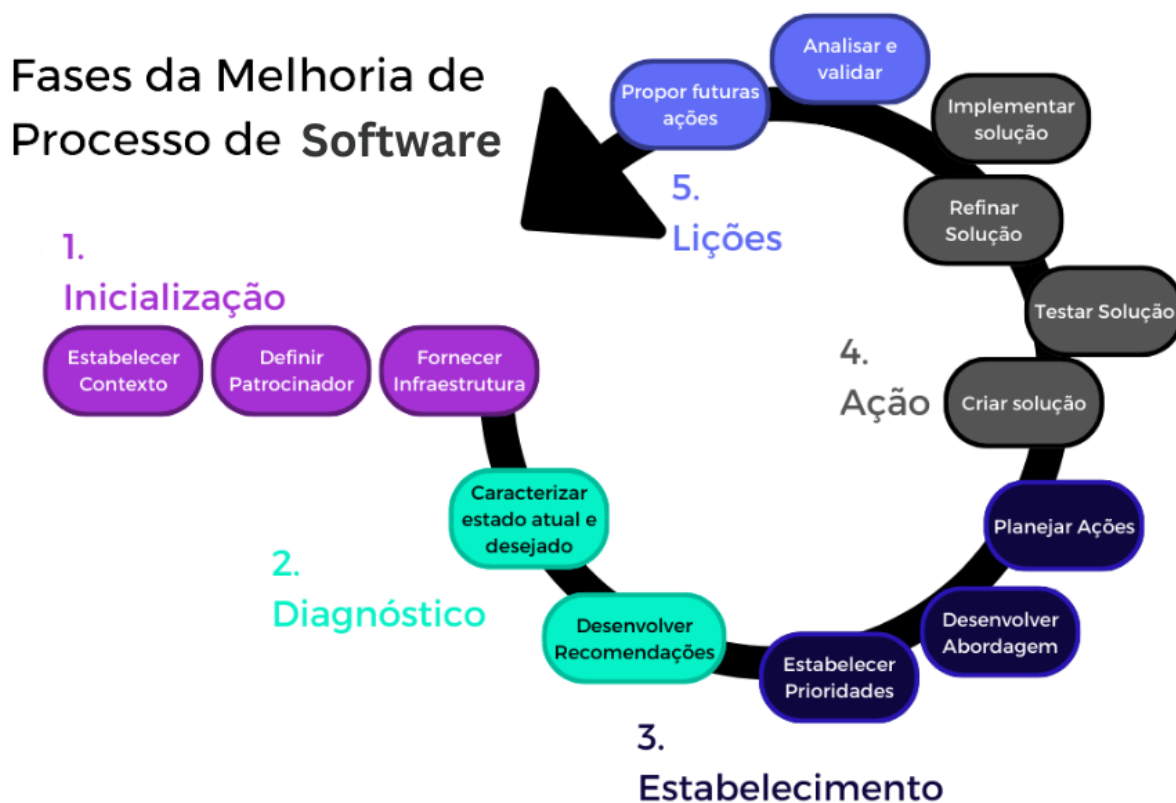
E elas podem ser detalhas de acordo com algumas etapas expressas na Tabela 2 e na Figura 21:

Tabela 2 – O MODELO IDEAL DE PROGRAMAS DE MELHORIA.

<b>Fase</b>	<b>Atividades</b>
<b>Inicialização</b>	Estabelecer Contexto (Motivação da organização para mudar o processo)
	Definir Patrocinador
	Fornecer Infraestrutura
<b>Diagnóstico</b>	Caracterizar os estados atual e desejado dos processos
	Desenvolver recomendações para resolver os problemas encontrados pela aferição
<b>Estabelecimento</b>	Estabelecer prioridades para os esforços de mudança, considerando a realidade da organização
	Definir a estratégia de mudança
	Elaborar plano detalhado de realização, considerando prazos, custos, riscos, pontos de controle, responsabilidades e outros elementos
<b>Ação</b>	Criar solução
	Testar solução
	Refinar solução
	Implementar solução
<b>Lições</b>	Analisar e validar
	Propor futuras ações

Fonte: Adaptado de PAULA FILHO (2019) e IDEAL.

Figura 21 – O MODELO IDEAL.



Fonte: Adaptado do IDEAL.

### 2.5.3 Métricas de Software

Sommerville (2019) diz que a medição de software se concentra na quantificação de aspectos de um sistema de software, tais como complexidade ou confiabilidade. Ao comparar esses valores com outros e com padrões estabelecidos pela organização, é possível avaliar a qualidade do software ou avaliar a efetividade de processos, ferramentas e métodos de software.

Uma métrica de software é uma característica de um sistema de software, da documentação do sistema ou do processo de desenvolvimento, que pode ser medida objetivamente (SOMMERVILLE, 2019).

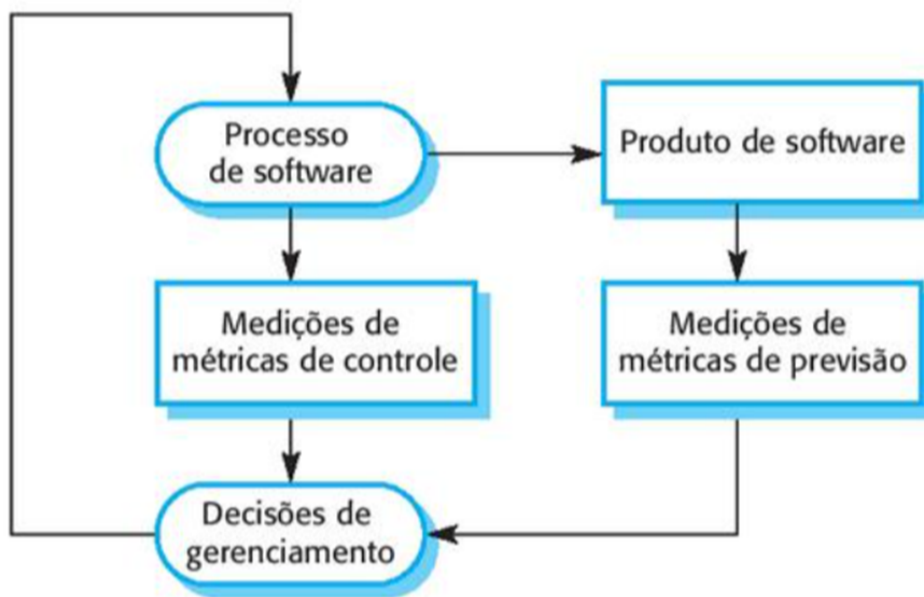
De acordo com Gallotti (2017), a métrica é o elemento mais empregado pela gerência de projetos de software. Isso ocorre porque, com as métricas certas, você consegue:

- Medir o nível de desempenho do processo de produção;
- Obter mais informações sobre a adequação ou não adequação do software em relação aos seus requisitos;
- Medir o esforço e o custo total investido - entre outros.

Desse modo, as métricas auxiliam a determinar a eficácia do alcance dos objetivos traçados ainda no começo do processo de desenvolvimento (GALLOTTI, 2017).

As métricas de software podem ser métricas de controle ou métricas de previsão (Figura 22). Como os nomes permitem deduzir, as métricas de controle apoiam o gerenciamento de processos, e as métricas de previsão ajudam a prever características do software (SOMMERVILLE, 2019) e estas métricas relacionam-se entre si.

Figura 22 – MEDIÇÕES DE PREVISÃO E CONTROLE.



Fonte: Sommerville (2019)

### 2.5.3.1 OKR

Uma das metodologias para definição de métricas que tem crescido, especialmente, no mercado de trabalho, é a de OKR, exemplificada na Figura ??.

Muniz et al. (2022) diz que o OKR consiste em dois elementos principais: objetivos e resultados-chave. O que se pretende alcançar é chamado de objetivos. Ele deve ser curto, qualitativo, inspirador e emocionante. Deve inspirar, desafiar e fornecer um senso de propósito à organização, ao grupo e às pessoas.

Já os Resultados-chave podem ser definidos como um pequeno conjunto de métricas que regulam o progresso em direção ao objetivo. Cada resultado-chave deve ser quantitativo e mensurável, ou seja, se não existe um número, não pode ser considerado um resultado-chave (MUNIZ et al., 2022).

A seguir um exemplo de OKR apresentado por [Muniz et al. \(2022\)](#):

**Objetivo:** Encantar os consumidores dos nossos produtos

**Resultados-chave:**

- *KR 1: Aumentar o NPS de 6,5 para 8*
- *KR 2: Aumentar a quantidade de acessos ao site de 5000 para 8000*
- *KR 3: Aumentar a taxa de recompra de 15% para 25%*

[Muniz et al. \(2022\)](#) também diz que o OKR é muito simples, ágil, flexível e descentralizado, o que o distingue de outros modelos de definição de metas. Ele pode ser usado para estabelecer metas organizacionais em qualquer nível (operacional, tático e estratégico).

## 3 Metodologia de Pesquisa

### 3.1 Considerações Iniciais do Capítulo

Considerando que o objetivo do trabalho é a proposição de melhorias de processo de software a partir da análise do processo de Engenharia de Requisitos da empresa ATR, neste capítulo apresenta-se o plano metodológico e a classificação da metodologia de pesquisa.

### 3.2 Classificação da Metodologia de Pesquisa

A pesquisa é classificada quanto ao objetivo, à natureza dos dados, ao ambiente de pesquisa e quanto ao tipo de coleta e análise de dados.

A classificação quanto ao objetivo é descritiva, pois visa a descrição das características de determinada população ou fenômeno (GIL, 2022). Descrição do processo de engenharia de requisitos da empresa ATR.

Quanto à natureza, esta pesquisa é classificada como aplicada, por ser voltada à aquisição de conhecimentos com vistas à aplicação numa situação específica, como diz (GIL, 2022).

A pesquisa pode ser classificada como estudo de caso, pois seu objetivo é a descrição ou reconstrução precisa dos casos. Segundo Flick (2013), o termo “caso” é compreendido como algo amplo, onde pode-se incluir pessoas, comunidades sociais (por exemplo, famílias), organizações e instituições (por exemplo, uma casa de repouso) como objeto de análise. Para este trabalho, a descrição do processo de engenharia de requisitos da empresa ATR.

Na abordagem técnica, a coleta de dados da pesquisa é classificada como qualitativa. De acordo com Flick (2013), esta abordagem não parte necessariamente de um modelo teórico do objeto de estudo e evita hipóteses e operacionalizações. Um pesquisador qualitativo seleciona os participantes de acordo com sua relevância e agrupa um pequeno número de casos. A coleta de dados é pensada para ser mais aberta, visando a visão abrangente possibilitada pela reconstrução do caso sob investigação.

Quanto à classificação dos procedimentos de pesquisa, os utilizados para o trabalho são: pesquisa bibliográfica e pesquisa documental.

Classifica-se a pesquisa, ainda, com base nas técnicas de coleta e análise de dados, descritas por (FLICK, 2013). As técnicas adotadas são:



- *Entrevistas semiestruturadas* - Preparar várias perguntas que cobrem o escopo pretendido da entrevista (Entender, na visão de papéis estratégicos, o processo atual e os gargalos existentes nele);
- *Observação participante* - Participação durante um período estendido no caso que é estudado, como um instrumento essencial da coleta de dados (Participação de sessões de levantamento de necessidades com o cliente, sessões de transformação das necessidades em requisitos e sessões de refinamento de atividades);
- *Análise Qualitativa de Documentos* - Utilização dos documentos produzidos para o estudo ou documentos já existentes e gerados pelo processo (Análise de documentos gerados pelo processo atual de engenharia de requisitos);
- *Análise Qualitativa de Conteúdo* - Seleção de entrevistas ou partes relevantes de entrevistas, que respondam à questão de pesquisa e análise da situação da coleta de dados (Análise de entrevistas semiestruturadas).

### 3.3 Plano Metodológico

Após a classificação da metodologia de pesquisa, é possível definir também o plano metodológico, que compõe as etapas para seguimento da metodologia escolhida.

O plano metodológico adotado para este trabalho é organizado em 4 fases: planejamento de pesquisa, coleta de dados, análise de dados e documentação de resultados. É possível ver uma representação deste plano na Figura 23.

#### 3.3.1 Planejamento

Na fase de planejamento, o esforço compreende à definição do tema de pesquisa, a questão de pesquisa, o objetivo geral e os objetivos específicos a ser atingidos e a classificação metodológica, já apresentados em tópicos anteriores do trabalho.

#### 3.3.2 Coleta de Dados

Para a fase de coleta de dados, os procedimentos e técnicas de coleta adotados são: Pesquisa Bibliográfica, Pesquisa Documental, Pesquisa Observacional, Entrevista Semiestruturada e Observação Participante.

##### 3.3.2.1 Procedimentos de Pesquisa

Segundo M. Lakatos e E. Lakatos (2021), a *Pesquisa Bibliográfica* inclui todas as fontes públicas escritas, orais e audiovisuais relacionadas ao assunto de estudo, incluindo

revistas, livros, teses, jornais, pesquisas, programas de rádio, filmes, gravações, transcrições verbais, material cartográfico, entre outras. O objetivo da pesquisa bibliográfica é fornecer ao pesquisador um contato direto com tudo o que foi escrito, dito ou gravado sobre o tema, incluindo conferências registradas.

Neste trabalho, a pesquisa bibliográfica gerou insumos necessários sobre Engenharia de Requisitos, Processo Ágil em Desenvolvimento de Software na Engenharia de Requisitos, Melhoria de Processos em Engenharia de Requisitos e Modelos de Gestão de Melhoria de Processos.

A característica da *Pesquisa Documental* é que a fonte de coleta de dados está restrita a documentos, escritos ou não, constituindo o que se denomina de fontes primárias. Estas podem ser recolhidas no momento em que o fato ou fenômeno ocorre, ou depois (M. Lakatos; E. Lakatos, 2021).

Para este trabalho, foram coletados dados gerados pelo processo como: atas de reunião, evidências de modelagem de requisitos, tickets de atividades, gravações de reuniões, etc. O uso destes dados apoiou toda a análise de maturidade do processo.

Na *Pesquisa Observacional* o investigador atua meramente como expectador de fenômenos ou fatos, sem, no entanto, realizar qualquer intervenção que possa interferir no curso natural e/ou no desfecho dos mesmos, embora possa, neste meio tempo, realizar medições, análises e outros procedimentos para coleta de dados (FONTELLES et al., 2009).

A pesquisa observacional, no contexto deste trabalho, foi feita a partir de participações em reuniões da equipe sobre requisitos e com clientes e também ao observar o processo de fora.

### 3.3.2.2 Técnicas de Coletas de Dados

As *Entrevistas Semiestruturadas* baseiam-se em um roteiro de assuntos ou perguntas e o entrevistador tem a liberdade de fazer outras perguntas para precisar conceitos ou obter mais informação sobre os temas desejados (isto é, nem todas as perguntas estão predeterminadas) (SAMPIERI; COLLADO; LUCIO, 2013).

Para este propósito, é preciso desenvolver um guia da entrevista como uma forma de orientação para os entrevistadores (FLICK, 2013).

Para este trabalho, foi criado um roteiro de entrevista baseado em conhecimentos básicos sobre o processo e em referências externas. A entrevista foi realizada por meio da ferramenta Microsoft Teams, disponibilizada pela universidade e foi feito o registro das perguntas e respostas por meio de gravações e transcrições.

Uma forma de coleta de dados contrastante é proporcionada pela *Observação Participante*. Aqui, a distância do pesquisador da situação observada é reduzida. Sua participação, durante um período estendido, no campo que é estudado, torna-se um instrumento essencial da coleta de dados (FLICK, 2013).

A observação participante consiste na participação real do pesquisador na vida da comunidade, da organização ou do grupo em que é realizada a pesquisa. O observador assume, pelo menos até certo ponto, o papel de membro do grupo (GIL, 2022).

A observação participante contribuiu para a pesquisa observacional. Assim como citado acima: a forma que foi feita no trabalho foi por meio de participação em reuniões e a observação por fora do processo.

### 3.3.3 Análise de Dados

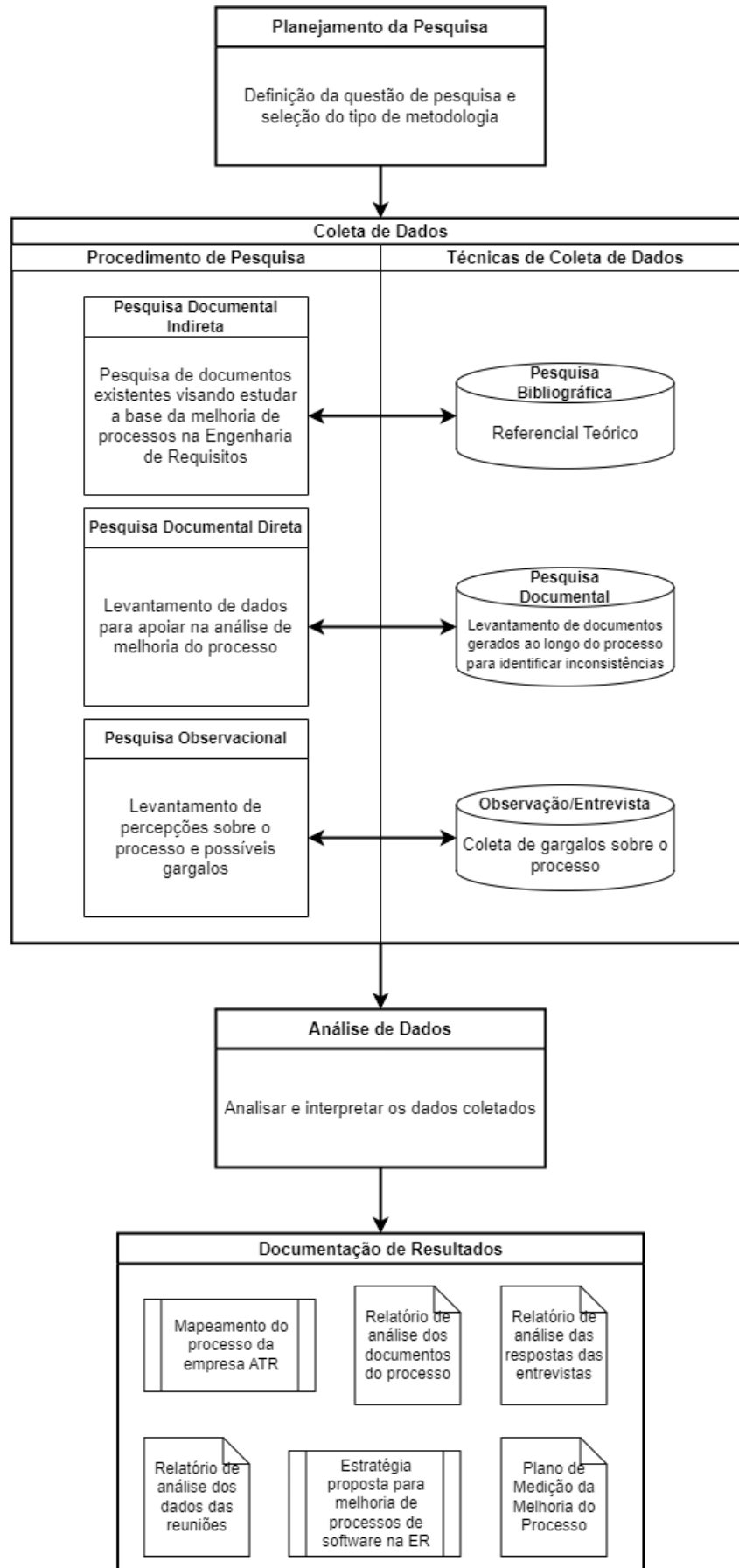
Nesta fase é feita a análise dos resultados observados nas fases anteriores, nesse caso, os resultados são os documentos gerados ao longo do processo de engenharia de requisitos, as respostas do questionário desenvolvido, as falas da entrevista gravada e a documentação construída após a observação participante em diversas reuniões. Para a análise deles, neste trabalho utiliza-se a *Análise Qualitativa de Documentos* e a *Análise Qualitativa de Conteúdo*.

### 3.3.4 Documentação dos Resultados

Os documentos de resultados apresentados para esta pesquisa são:

1. *Mapeamento do processo de Engenharia de Requisitos da empresa ATR*: Diagrama BPMN do fluxo atual e do fluxo ideal;
2. *Relatório de análise dos documentos do processo*: Análise dos documentos gerados ao longo do processo;
3. *Relatório de análise das respostas da entrevista*: Identificação do processo atual e de gargalos do processo;
4. *Relatório de análise dos dados das reuniões*: Complemento aos demais relatórios de análise;
5. *Estratégia proposta para melhoria de processos de software na engenharia de requisitos*: Proposta de melhorias a partir dos relatórios de análise e fluxo BPMN do processo considerando as melhorias propostas;
6. *Plano de Medição da Melhoria do Processo*: Plano de medição para avaliar melhoria do processo de engenharia de requisitos, com o uso da metodologia OKR.

Figura 23 – PLANO METODOLÓGICO.



Fonte: Autora

## 4 Proposta do Trabalho

### 4.1 Considerações Iniciais

A qualidade de produtos e a qualidade de processos estão fortemente ligadas, especialmente na produção de um software, e todas as etapas do processo definem quanto valor aquele produto vai agregar para seu usuário final.

Antes de iniciar o desenvolvimento de projetos, é necessário fazer um bom planejamento para evitar inconsistências. O processo de desenvolvimento de software, normalmente, é composto pelas fases de Comunicação, Planejamento, Modelagem, construção e Entrega.

Esses cinco passos acontecem repetidamente após o fim de ciclos de desenvolvimento, também conhecidos como *sprints*, e precisam ser constantemente aprimorados.

Considerando isso, este trabalho foca no entendimento e na melhoria contínua da etapa de comunicação, mais especificamente, no início do projeto, com o levantamento das necessidades dos clientes e a transformação delas em requisitos para o sistema na empresa ATR.

Apresenta-se, a seguir, o contexto da empresa ATR e a proposta de solução a ser aplicada no Trabalho de Conclusão 2, com o auxílio de partes do Modelo IDEAL.

### 4.2 Contexto Organizacional

A empresa ATR é uma organização privada que tem como principal atividade o desenvolvimento de uma solução de software para a gestão de frotas florestais e seu serviço atende grandes empresas nacionais.

Ela conta com uma cultura ágil e colaborativa, na qual as equipes estão sempre trabalhando com o intuito de resolver os problemas da forma mais rápida e com a maior qualidade possível.

Por ainda ser uma organização relativamente nova, passa por desafios organizacionais, onde existem áreas com uma carga de trabalho elevada e, por esse motivo, torna-se difícil a elaboração de planos de melhoria contínua para processos que necessitam esta atenção, mesmo que esta melhoria seja sempre encorajada e hajam tentativas constantes para que ela aconteça.

Na equipe destinada ao desenvolvimento do produto, que compreende desenvolvedores, testadores, analistas de *business intelligence*, gerente de projetos, analistas de

negócio e lideranças serventes, utiliza-se o *framework Scrum* como modelo ágil para gestão do trabalho. Além dele, também existem conceitos do XP, do FDD e do *Kanban* que auxiliam um processo mais sólido para o desenvolvimento do software.

Como o *scrum* possui ciclos de desenvolvimento, conhecidos como *sprints*, é importante destacar que cada uma delas compreende uma quantidade de atividades retiradas do *product backlog* e, quando elas são comprometidas, no momento da *Sprint Planning*, já devem estar refinadas e a equipe já deve ter pleno entendimento delas e com o mínimo de dúvidas de negócio.

Considerando isso, os papéis de gerente de projetos, analista de negócio e a estagiária em análise funcional trabalham no processo de início do projeto e levantamento de requisitos, conhecendo as necessidades dos clientes, apresentando a proposta de solução acordada, transformando as necessidades em requisitos de software, priorizando as demandas para serem repassadas e construindo um cronograma com prazos para a entrega dos produtos comprometidos.

Para auxiliar nessa organização, a equipe utiliza as ferramentas Microsoft Teams, para reuniões com clientes, Clickup, para gestão de atividades, documentação de reuniões e o Miro, para apoiar no refinamento.

## 4.3 Proposta de Solução

Para este trabalho, serão utilizadas partes do modelo IDEAL para auxiliar no conhecimento e mapeamento de inconsistências e possíveis melhorias do processo de engenharia de requisitos. A proposta de solução é apresentada com base em seus conceitos.

### 4.3.1 Inicialização

Nesta etapa do modelo, inicia-se com o estabelecimento do contexto, onde serão apresentadas e detalhadas as necessidades de negócio e o estímulo para a mudança do processo, além da expectativa de benefício da empresa com relação à melhoria do processo. Depois, será feita a definição do patrocinador, em que se apresentam os agentes envolvidos no processo, ou seja, os papéis que fazem parte e são afetados por ele. E por fim, será apresentado o fornecimento da estrutura, em que é estabelecida a infraestrutura, recursos em geral, para a execução da melhoria do processo.

### 4.3.2 Diagnóstico

Na etapa de diagnóstico, primeiramente, será feita a caracterização do estado atual e desejado, em que são apresentados os fluxos do processo atual, descrito e executado, assim como a diferença entre eles e do fluxo desejado. Posteriormente, serão desenvolvidas

recomendações, feita a avaliação do processo, conforme a linha de base de maturidade definida, determinado o nível de maturidade em que ele se encontra e recomendações para melhoria.

Nesta etapa serão utilizadas as técnicas de coleta de dados: questionário, entrevista semiestruturada e observação participante e feita a análise das documentações geradas pela coleta.

Também será empregado o modelo SRP-CMMI para apoiar o diagnóstico do processo de engenharia de requisitos da empresa ATR. Este modelo foi escolhido com base na Tabela 1, apresentada no Capítulo 2, por ser mais apropriado no contexto da empresa, que trabalha com um processo ágil. É um modelo de implementação simples e de fácil entendimento.

### 4.3.3 Estabelecimento

Na etapa de estabelecimento o objetivo será, primeiramente, definir ações de melhoria para o processo, conforme as recomendações da fase de diagnóstico e, posteriormente, estabelecer as prioridades dessas ações. Para fechar essa etapa, será construída uma estratégia de utilização das ações recomendadas para o alcance do objetivo, definindo o planejamento de ações detalhado para o primeiro ciclo de melhoria do processo de engenharia de requisitos da empresa ATR.

### 4.3.4 Ação

Esta etapa compreende a criação, o teste, o refinamento e a implementação da solução. Nela será proposta a solução técnica a ser adotada para as ações de melhoria que serão priorizadas na etapa estabelecimento. Também nesta etapa, serão definidas as métricas, baseadas na metodologia OKR, que serão coletadas para apoiar a análise do alcance dos objetivos de melhoria (ou benefícios esperados). A parte de aplicação e testagem dessa fase ficará sob total responsabilidade da empresa ATR, se for de seu interesse.

### 4.3.5 Lições

Esta etapa compreende a análise e validação da solução e a proposição de ações futuras. Ela não é abordada neste trabalho e será realizada independentemente pela empresa ATR, se for de seu interesse.

## 4.4 Cronograma

Tabela 3 – CRONOGRAMA TCC1.

<b>Mês</b>	<b>Atividades</b>
Novembro	Elaboração do problema, justificativa, questão de pesquisa e objetivos geral e específico; Definição de estratégia para construção do referencial teórico; Leitura dos materiais a serem utilizados no referencial teórico.
Dezembro	Escrita um resumo preliminar do TCC; Criação da string de busca; Escrita da introdução preliminar do TCC; Definição dos tópicos do referencial teórico; Escrita do referencial teórico.
Janeiro	Finalização do referencial teórico; Escrita da metodologia de pesquisa; Escrita da proposta do trabalho; Escrita das considerações finais; Revisão e ajustes da introdução e resumo; Revisão final do trabalho com os orientadores; Ajustes finais do TCC1.
Fevereiro	Entrega da versão final do TCC1; Preparação da defesa; Defesa; Ajustes pedidos pela banca.

Fonte: Autora



Tabela 4 – CRONOGRAMA TCC2.

<b>Mês</b>	<b>Atividades</b>
Abril	Criação e aplicação do questionário; Criação de perguntas para a entrevista semiestruturada; Aplicação das entrevistas semiestruturadas; Observação Participante em reuniões.
Maio	Análise e documentação dos dados coletados; Escrita da etapa inicialização do IDEAL; Avaliação do processo e mapeamento dos fluxos BPMN atual e desejado do processo; Determinação do nível de capacidade e maturidade do processo e fazer a definição de recomendações.
Junho	Escrita da etapa diagnóstico do IDEAL; Priorização das ações definidas no diagnóstico; Planejamento de ações; Definição de um plano de medições.
Julho	Revisão final do TCC2; Entrega da versão final do TCC2; Preparação para a defesa; Defesa; Ajustes finais pedidos pela banca.

Fonte: Autora

# 5 Aplicação da Proposta

## 5.1 Considerações Iniciais

O objetivo deste capítulo é apresentar como foi realizada a pesquisa proposta, mostrando a realidade da empresa ATR, especialmente no que diz respeito ao processo de engenharia de requisitos da organização. Nesta parte do trabalho, é registrado o diagnóstico do processo, feito a partir de coleta e análise de dados e também o resultado obtido, que, no caso, é a proposta de solução para melhoria do processo.

Os tópicos abordados seguem a ordem em que a pesquisa foi executada, baseando-se no modelo IDEAL. Pode-se ver que, neste trabalho, o modelo não foi realizado de forma completa, contemplando até o início da fase de ação.

## 5.2 Inicialização

Como já introduzido no Capítulo 4, essa primeira fase tem como foco a apresentação e o entendimento do contexto no qual será trabalhado, conhecendo as necessidades e o estímulo para mudança da empresa em questão, além das expectativas de benefício com relação às mudanças que serão propostas para o processo de engenharia de requisitos e os recursos disponibilizados para análise.

### 5.2.1 Necessidades de negócio e Estímulo para mudança

A empresa ATR está crescendo rapidamente no mercado de trabalho, porém, por ainda ser considerada uma empresa pequena para a quantidade de demandas que precisa cumprir, sofre com processos não tão bem definidos, documentados e executados.

Um dos processos onde essa realidade fica mais evidente é o processo de engenharia de requisitos da empresa, que, por sua vez, gera um grande impacto para todo o desenvolvimento de software.

Este processo, atualmente, enfrenta problemas como:

- Falta de constância para entendimento de necessidades e levantamento de requisitos junto às *sprints*;
- Excesso de reuniões com o cliente;
- Poucos atores no processo;

- Falta de documentação formal do processo e dos resultados do processo.

E gera problemas como:

- Possível insatisfação do cliente com o produto final;
- Retrabalho para as equipes de desenvolvimento;
- Atraso de funcionalidades;
- Falta de um backlog priorizado.

### 5.2.2 Definição dos Agentes

Para que as mudanças no processo sejam significativas, faz-se necessária a seleção de agentes que possam opinar e contribuir para que a melhoria seja a mais eficaz possível.

No contexto da empresa ATR, os envolvidos com o processo de engenharia de requisitos que vão contribuir ativamente para essa melhoria são:

- Analista de Negócio/*Product Owner* - Responsável por ter reuniões com o cliente, conhecer suas necessidades e fazer a definição de requisitos, considerando requisitos de usuário e de sistema;
- Estagiária em Análise Funcional - Este papel é responsável por apoiar o *Product Owner* em reuniões com clientes, documentando o que é dito e criando propostas de solução para apresentar;
- Desenvolvedor - É o responsável por desenvolver as funcionalidades do sistema, conforme os requisitos definidos.

### 5.2.3 Expectativa de Benefício

Ao finalizar os ciclos de melhoria contínua, espera-se um processo bem definido, documentado e executado, além de um processo conhecido, não somente pelos envolvidos diretamente com ele, mas, também, pelos demais influenciados por ele.

Também é esperado que o trabalho dos desenvolvedores ocorra de forma mais fluida, conhecendo a prioridade dos requisitos e das atividades com antecedência. Dessa forma garantindo a satisfação dos clientes com o resultado final da entrega e que ela seja feita dentro do tempo esperado e com os requisitos corretos.

## 5.2.4 Fornecimento da Estrutura

Para a realização da coleta e análise de dados necessários para a pesquisa, foram fornecidas pela Universidade de Brasília, as ferramentas do Pacote Office, sendo uma delas o Word Online da Microsoft, que foi utilizado para a criação de um roteiro de perguntas para as entrevistas e também a ferramenta Teams para gravação dessas entrevistas.

A empresa também colaborou com o compartilhamento de informações: foi disponibilizado por eles o acesso a gravações de reuniões feitas pelo Teams e também a documentações e registros de atividades no Clickup.

## 5.3 Diagnóstico

Nesta importante fase do IDEAL, será apresentada, primeiramente, a caracterização do estado atual e do estado desejado do processo de engenharia de requisitos da empresa, para comparação entre a realidade presente e as expectativas para o futuro.

Posteriormente, será feita uma análise de dados, baseada nas informações apresentadas na etapa de caracterização e, por fim, serão realizadas recomendações para melhoria do processo, como conclusão da análise dos dados.

### 5.3.1 Caracterização do Estado Atual e Desejado

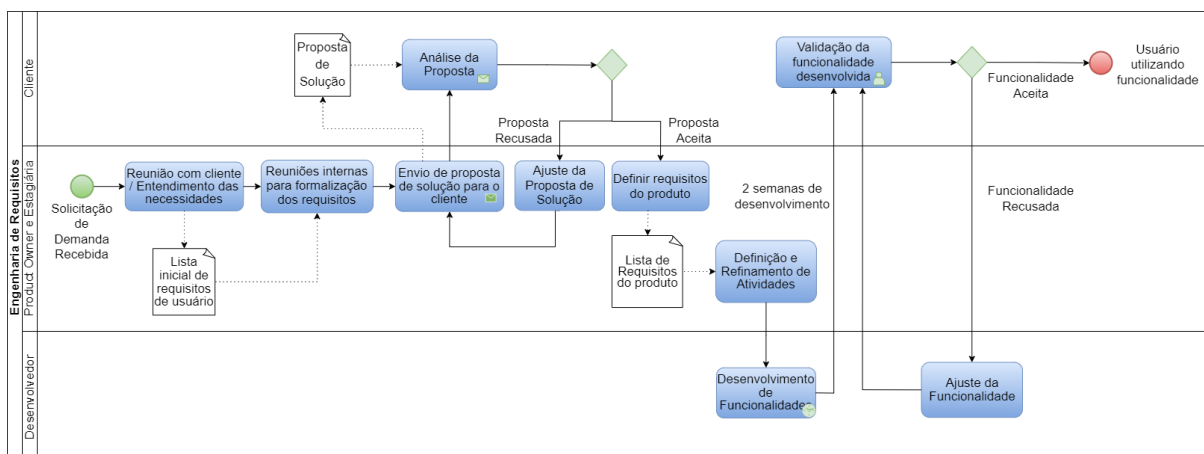
#### 5.3.1.1 Baseline de Processo

Ao fazer a caracterização, inicialmente, é necessário conhecer o processo com o qual será trabalhado, por meio de uma baseline de processo.

Para definir a baseline de processo, são apresentados o processo definido e o processo executado da empresa e é feita uma comparação entre os dois.

Contudo, a organização ATR não possui nenhum tipo de processo formalmente definido/documentado, logo, neste tópico será apresentado o fluxo BPMN, ilustrado na Figura 24, do processo executado para a baseline.

Figura 24 – PROCESSO DE ENGENHARIA DE REQUISITOS DA EMPRESA ATR.



Fonte: Autora

A partir do BPMN, é possível identificar sete tarefas no processo e fazer sua descrição da seguinte forma:

- **Tarefa 1:** Reunião com cliente / Entendimento de Necessidades

*Descrição da tarefa:* Momento de reunião com o cliente, onde há a exposição e o entendimento de necessidades.

*Produto de Entrada:* Solicitação de Demanda Recebida.

*Produto de Saída:* Lista Inicial de Requisitos do usuário.

*Ferramentas:* Power Point para apresentação, Teams para reunião com clientes, Word para anotação dos pontos relevantes da reunião e criação da proposta de solução.

- **Tarefa 2:** Reuniões internas para formalização dos requisitos

*Descrição da tarefa:* Após o entendimento inicial das necessidades do cliente, são feitas reuniões internas para formalizar o escopo técnico.

*Produto de Entrada:* Lista Inicial de Requisitos do usuário.

*Produto de Saída:* Proposta de Solução.

*Ferramentas:* Word para criação do documento de proposta da solução e Outlook para envio da proposta e recebimento da devolutiva.

- **Tarefa 3:** Envio da Proposta de Solução

*Descrição da tarefa:* Após o entendimento inicial das necessidades do cliente e reuniões internas para fechamento do escopo técnico, a empresa faz o envio de uma proposta de solução.

*Produto de Entrada:* Proposta de Solução.

*Produto de Saída:* Devolutiva do cliente sobre a proposta de solução.

*Ferramentas:* Word para criação do documento de proposta da solução e Outlook para envio da proposta e recebimento da devolutiva.

- **Tarefa 4:** Análise da Proposta de Solução

*Descrição da tarefa:* Após enviada a proposta de solução, o cliente vai avaliar se ela está cumprindo com as expectativas passadas.

*Produto de Entrada:* Proposta de Solução.

*Produto de Saída:* Devolutiva do cliente sobre a proposta de solução.

*Ferramentas:* Outlook para recebimento do e-mail de devolutiva do cliente.

- **Tarefa 5:** Definição de Requisitos do Produto

*Descrição da tarefa:* A partir do entendimento das necessidades e aprovação da proposta de solução, com os devidos ajustes, se necessário, é feita a definição de requisitos do produto.

*Produto de Entrada:* Devolutiva do cliente sobre a proposta de solução.

*Produto de Saída:* Lista de Requisitos do Produto.

*Ferramentas:* Outlook para recebimento do e-mail de devolutiva do cliente, Miro para definição dos requisitos e Word para criação da lista de requisitos.

- **Tarefa 6:** Definição e refinamento de atividades

*Descrição da tarefa:* As funcionalidades/histórias são definidas e refinadas a partir dos requisitos que foram acordados.

*Produto de Entrada:* Lista de Requisitos do Produto.

*Produto de Saída:* Definição das histórias/funcionalidades a serem desenvolvidas.

*Ferramentas:* Miro para organização de pensamentos e refinamento e clickup para criação e gestão das histórias.

- **Tarefa 7:** Desenvolvimento das Funcionalidades

*Descrição da tarefa:* Momento do desenvolvimento das funcionalidades a partir de sua definição.

*Produto de Entrada:* Atividades refinadas no clickup.

*Produto de Saída:* Funcionalidades com desenvolvimento pronto.

*Ferramentas:* Clickup para gestão das funcionalidades sendo desenvolvidas, ferramentas de edição de código, github para organização do código e Jenkins como gerenciador de versões.

- **Tarefa 8:** Validação da funcionalidade desenvolvida

*Descrição da tarefa:* A partir das funcionalidades desenvolvidas, é necessário validar se os requisitos foram devidamente respeitados.

*Produto de Entrada:* Funcionalidades com desenvolvimento pronto.

*Produto de Saída:* Funcionalidades prontas e validadas.

*Ferramentas:* Clickup para gestão do desenvolvimento das funcionalidades, ferramentas de edição de código, github para organização do código, jenkins como gerenciador de versões e lista de requisitos para validação das funcionalidades.

### 5.3.1.2 Baseline de Maturidade do Processo

Para a definição da baseline de maturidade do processo, é necessário analisar quais são as práticas de engenharia de requisitos que estão sendo implementadas e qual é o grau de implementação delas.

Para esta pesquisa, foi selecionado o modelo SRP-CMMI para análise das práticas, que foram introduzidas na subseção [SRP-CMMI](#) do trabalho e a escala FILIPINY, proveniente do método SCAMPI ([CARNEGIE MELLON UNIVERSITY, 2011](#)), do modelo CMMI, para selecionar do grau de implementação. A escala está ilustrada na Tabela 5.

Tabela 5 – ESCALA FILIPINY.

Rótulo	Sigla	Significado
Totalmente implementado (Fully Implemented)	FI	- Artefatos suficientes e/ou afirmações estão presentes e são considerados adequados para demonstrar a implementação da prática; - Não são observadas fraquezas na aplicação prática.
Largamente implementado (Largely Implemented)	LI	- Artefatos suficientes e/ou afirmações estão presentes e são considerados adequados para demonstrar a implementação da prática; - Uma ou mais fraquezas são observadas na aplicação prática.

Parcialmente implementado (Partially Implemented)	PI	<ul style="list-style-type: none"> <li>- Alguns ou todos os dados requeridos estão ausentes ou considerados inadequados;</li> <li>- Alguns dados estão presentes para indicar que alguns aspectos da prática são implementados;</li> <li>- Uma ou mais fraquezas são observadas na aplicação prática.</li> </ul> <p>OU</p> <ul style="list-style-type: none"> <li>- Os dados fornecidos à equipe (artefatos e/ou afirmações) conflitantes – alguns dados indicam que a prática é implementada e alguns indicam que a prática não é implementada;</li> <li>- Uma ou mais fraquezas são observadas na aplicação prática.</li> </ul>
Não implementado (Not Implemented)	NI	<ul style="list-style-type: none"> <li>- Alguns ou todos os dados requeridos estão ausentes ou considerados inadequados;</li> <li>- Os dados fornecidos não indicam a conclusão de que a prática é implementada;</li> <li>- Uma ou mais fraquezas são observadas na aplicação prática.</li> </ul>
Ainda não implementado (Not Yet)	NY	<ul style="list-style-type: none"> <li>- A organização ainda não atingiu o estágio esperado do trabalho para implementar a prática.</li> </ul>

Fonte: Adaptada do método SCAMPI ([CARNEGIE MELLON UNIVERSITY, 2011](#))

Com a introdução da tabela FILIPINY, o próximo passo é coletar os dados necessários para apoiar a análise e classificação da maturidade do processo.

### 5.3.1.3 Coleta de Dados

Para a coleta e análise de dados, como introduzido no capítulo 3, foram feitas entrevistas semiestruturadas e levantados documentos gerados pelo processo.



### 5.3.1.3.1 Documentos coletados e utilizados para análise

Uma das técnicas que apoiou a análise do processo de engenharia de requisitos da empresa ATR, foi a análise de documentos gerados pelo processo.

As documentações disponibilizadas pela organização foram:

- Atas de Reunião com clientes diversos;
- Gravações de reuniões com clientes e de reuniões internas da equipe;
- Quadros diversos da ferramenta Miro;
- Propostas de Solução enviadas para clientes;
- Tickets de histórias já refinadas e histórico delas;
- Documentos gerados por clientes diversos para divulgação de dados e interesses com a empresa.

### 5.3.1.3.2 Roteiro aplicado para entrevistas

Outra técnica utilizada para apoiar a análise foi a de entrevistas semiestruturadas, que teve maior foco nos membros envolvidos no processo.

Foram entrevistados três perfis:

- A estagiária em análise funcional:
  - Superior Incompleto em Engenharia de Software;
  - 2 anos de experiência no mercado;
  - Está há 1 ano na empresa;
  - Atua como analista de requisitos.
- O *Product Owner* e Gerente do Produto da empresa:
  - Formado em Ciência da Computação;
  - 5 anos de experiência no mercado;
  - Está há 3 anos na empresa;
  - Atua como gerente e analista de requisitos.
- Um dos desenvolvedores que trabalham na empresa:
  - Superior Incompleto em Engenharia de Software;

- 2 anos de experiência no mercado;
- Está há 2 anos na empresa;
- Atua como programador.

Também foi elaborado um roteiro de entrevistas visando esclarecer como o processo é enxergado e ajudar a identificar as oportunidades de melhoria nele.

As perguntas aplicadas foram:

- P1.** Qual é o cargo que você ocupa?
- P2.** Quais são as suas responsabilidades, em geral?
- P3.** O que você sabe sobre o processo engenharia de requisitos?
- P4.** Como funciona o processo de engenharia de requisitos da empresa? - Descreva
- P5.** O que, do processo de engenharia de requisitos, é realizado pela empresa?
- P6.** Quais são as documentações geradas pelo processo?
- P7.** Em qual padrão genérico o processo de engenharia de requisitos da empresa é baseado? Qual foi a base para a criação desse processo?
- P8.** Quais são os recursos necessários para a execução deste processo?
- P9.** Quais são os indicadores de medição da qualidade deste processo?
- P10.** Quais são os indicadores de medição da qualidade dos resultados deste processo?
- P11.** Caso não haja indicadores ainda, quais você considera que devem ser adotados para medir a eficácia do processo?
- P12.** O quanto o processo de engenharia de requisitos é transparente para toda a equipe? E o quão fácil é de entender a definição do processo?
- P13.** O quão aceitável é o processo para os envolvidos e o quanto ele é seguido?
- P14.** Quais são as ações tomadas pela equipe para identificar erros no processo e em seus resultados e realizar mudanças?
- P15.** Como funciona a introdução e o treinamento para novos integrantes do processo?
- P16.** Como este processo afeta no processo de desenvolvimento?
- P17.** O quão satisfeito você está com o processo atual?

- P18.** Considerando tudo o que já foi falado sobre o processo atual, quais são as barreiras que você vê hoje?
- P19.** O que você gostaria que fosse diferente nesse processo? Tem alguma sugestão de melhoria?
- P20.** Quais são as iniciativas atuais da empresa para executar melhorias neste processo?

As perguntas foram divididas em categorias e foi possível tirar algumas conclusões a partir das respostas dos entrevistados:

- Direcionadas às funções do entrevistado (P1 e P2)
  - Como os três entrevistados tinham perfis diferentes, suas respostas foram diversas. A estagiária em análise funcional levantou algumas responsabilidades como apoio em construção de atas de reunião e propostas de solução e também a gestão de bugs da empresa. O *Product Owner* relatou, em linhas gerais, sobre o levantamento e modelagem de requisitos e a manutenção e gestão do backlog da equipe de desenvolvimento. E, por último, o desenvolvedor comentou sobre o desenvolvimento de novas funcionalidades para o produto da empresa e a resolução de defeitos.
- Conhecimento geral sobre Engenharia de Requisitos (P3)
  - Todos relataram ter pouco conhecimento teórico do processo de Engenharia de Requisitos, se limitando a aulas que tiveram na faculdade e tendo pouca ou nenhuma literatura de base real que sirva de inspiração para procurarem um processo mais condizente com o ideal.
- Descrição do processo na empresa (P4, P5, P6, P7, P8, P16, P17 e P18)
  - De forma geral, as respostas indicaram um processo executado de forma muito prática e corrida, sem muita base. Foi relatado que não há definições, documentações ou padrões do processo que ajudem em uma melhor execução dele.
- Qualidade e medição do processo (P9, P10, P11, P14, P19 e P20)
  - Como o processo não tem uma definição exata, também não há um sistema de medição de sua qualidade. Atualmente fica muito restrito à opinião do cliente, que não é formalizada. Porém, pelas respostas dadas, este ponto foi reconhecido como importante e eles acreditam que é necessário fazer essa implementação e incluir indicadores como, por exemplo, dois formulários, um interno e um externo, onde seja dada uma nota de 0 a 10 para o processo e para o produto.
- Transparência do processo com os envolvidos (P12, P13 e P15)

- Ficou muito claro que, para a equipe que trabalha diretamente com os requisitos, o processo é conhecido, mas não é bem definido e precisa ser reestruturado. Já para os afetados pelo processo, mas não necessariamente envolvidos nele, há pouca ou nenhuma transparência. Foi relatada também a necessidade que a equipe de desenvolvimento sente em ser mais incluída nesse processo.

#### 5.3.1.4 Caracterização das Práticas do Modelo SRP-CMMI

Com todas as informações coletadas, é possível classificar cada prática do modelo SRP-CMMI por área de processo: *Elicitação*, classificada na Tabela 6, *Análise*, Tabela 7, *Especificação*, Tabela 8, *Validação*, Tabela 9, *Gestão de Requisitos*, Tabela 10 e *Suporte Organizacional*, Tabela 11. As práticas são detalhadas por seu código (CP), descrição, grau de implementação (GI), evidência objetiva (EO) e justificativa.

##### 5.3.1.4.1 Área de Processo - Elicitação

**Meta:** *Estabelecer um entendimento sobre o problema, soluções e stakeholders.*

Tabela 6 – AVALIAÇÃO DE MATURIDADE - ELICITAÇÃO.

CP	Descrição	GI	EO	Justificativa
REP 1.1	Identificar <i>stakeholders</i> e recursos de requisitos	PI	Ata de Reunião Inicial do Projeto	Os <i>stakeholders</i> e recursos são evidenciados na ata de reunião, porém não existe um documento onde o perfil desses <i>stakeholders</i> possa ser detalhado ou mesmo os recursos fornecidos por eles. Além disso, os dados são conflitantes, porque existem 2 pessoas que fazem a ata em formatos diferentes e podem ter interpretações diferentes dos pontos abordados na reunião. Todos esses pontos apontam fraquezas no processo.

REP 1.2	Elicitar necessidades	LI	Ata de Reunião Inicial do Projeto e Quadro no Miro	As necessidades dos clientes são elicitadas ou em uma ata de reunião ou em um quadro visual. Essas evidências são suficientes para indicar que a prática ocorre, porém, ainda é um processo com fraquezas, visto que não há uma documentação ou um espaço adequado para o armazenamento desses dados.
------------	-----------------------	----	---	---

Fonte: Autora

#### 5.3.1.4.2 Área de Processo - Análise

##### **Metas:**

1. *Criar um modelo de análise que identifica dados, funções, características, restrições e requisitos comportamentais, usando a informação obtida durante a elicitação.*
2. *Resolver conflitos, priorizar requisitos, e identificar riscos, visando um resultado positivo antes de prosseguir para as atividades subsequentes de engenharia de software.*

Tabela 7 – AVALIAÇÃO DE MATURIDADE - ANÁLISE.

CP	Descrição	GI	EO	Justificativa
REP 1.1	Estabelecer conceitos operacionais e cenários	LI	Quadro no Miro e Proposta de Solução	Existem conceitos bem definidos na prática e os cenários criados no quadro do Miro e na proposta de solução evidenciam bem isso. Porém, não existe nenhuma documentação única e específica com todos esses conceitos evidenciados, o que enfraquece o processo de forma geral.
REP 1.2	Modelar Requisitos	PI	Quadro no Miro	A modelagem dos requisitos é feita por meio de quadros do Miro. Lá são criadas visualizações de como o produto deve ficar após o desenvolvimento, porém não são evidenciados de forma clara/escrita, demonstrando fraquezas no processo e dificuldades na consulta deles.

REP 2.1	Resolver conflitos	PI	Histórico do Ticket	<p>A prática da resolução de conflitos dos requisitos não é bem evidenciada. Normalmente, essa prática vem com alguma confusão ou dificuldade do desenvolvedor de uma nova funcionalidade. Não é correto dizer que a prática não ocorre, mas também não existe uma evidência muito adequada dessa resolução, que não seja o histórico da atividade na qual está sendo trabalhada.</p>
REP 2.2	Priorizar requisitos	PI	Quadro no Miro e Proposta de Solução	<p>A priorização também não é evidenciada de forma clara. É fato que os requisitos são priorizados, pois em reuniões com os <i>stakeholders</i>, existe uma comunicação sobre isso. Contudo, a única evidência que apoia no entendimento dessas prioridades, é a comparação entre os requisitos triados no quadro do Miro e a proposta de solução enviada para o cliente.</p>

REP 2.3	Obter comprometimento com os requisitos	PI	Proposta de Solução	Não existe uma evidência específica que deixe clara essa prática. A documentação na qual é feito o compromisso com os requisitos é a enviada para o cliente como proposta. Porém, não deixa claro o comprometimento da equipe como um todo internamente, o que demonstra fraquezas no processo.
------------	---	----	---------------------	--

Fonte: Autora

#### 5.3.1.4.3 Área de Processo - Especificação

**Meta:** *Formalizar os requisitos informacionais, funcionais e comportamentais.*

Tabela 8 – AVALIAÇÃO DE MATURIDADE - ESPECIFICAÇÃO.

CP	Descrição	GI	EO	Justificativa
REP 1.1	Produzir um documento que pode ser revisado, avaliado e aprovado sistematicamente	PI	Proposta de Solução	O único documento que formaliza os requisitos é o de proposta de solução, que é enviado para o cliente. Contudo, não é possível identificar os requisitos de forma detalhada.

Fonte: Autora



## 5.3.1.4.4 Área de Processo - Validação

**Meta:** *Validar requisitos para garantir que o produto resultante irá performar como pretendido no ambiente do usuário final.*

Tabela 9 – AVALIAÇÃO DE MATURIDADE - VALIDAÇÃO.

CP	Descrição	GI	EO	Justificativa
REP 1.1	Confirmar que os requisitos estão corretos, completos, consistentes, testáveis e que satisfazem as necessidades do cliente	PI	Ticket de História com requisitos e critérios de aceitação	A confirmação acontece a partir do momento em que os tickets são passados para a equipe de desenvolvimento e, com os feedbacks e as dúvidas que surgem. A prática não é implementada de forma adequada e que incentiva o aprendizado. Portanto, torna o processo muito fraco.

Fonte: Autora

## 5.3.1.4.5 Área de Processo - Gestão de Requisitos

**Metas:**

1. *Identificar, controlar e rastrear todas as mudanças que ocorrem com os requisitos.*
2. *Garantir que todos os requisitos de origem estão completamente rastreados.*

Tabela 10 – AVALIAÇÃO DE MATURIDADE - GESTÃO DE REQUISITOS.

CP	Descrição	GI	EO	Justificativa
REP 1.1	Gerenciar as mudanças dos requisitos	PI	Histórico de modificação de Ticket de História	Assim como a resolução de conflitos, a gestão das mudanças dos requisitos também não é muito bem evidenciada. A única evidência e não a mais adequada é o histórico de modificações de ticket.
REP 2.1	Gerenciar a rastreabilidade dos requisitos	NI	Não há	Não há nenhuma documentação de rastreabilidade dos requisitos. Eles não têm códigos que os identificam e linkáveis, tornando seu acesso dificultado. Portanto, a prática é considerada não implementada.

Fonte: Autora

## 5.3.1.4.6 Área de Processo - Suporte Organizacional

**Metas:**

1. *Avaliar a quantidade de suporte organizacional dado para as práticas de engenharia de requisitos.*
2. *Desenvolver habilidades e conhecimento, e treinar pessoas para que performem seus papéis efetivamente e eficientemente.*
3. *Avaliar objetivamente a adesão em relação às descrições do processo, padrões e procedimentos aplicáveis.*

Tabela 11 – AVALIAÇÃO DE MATURIDADE - SUPORTE ORGANIZACIONAL.

CP	Descrição	GI	EO	Justificativa
REP 1.1	Estabelecer uma Política Organizacional de Engenharia de Requisitos	NY	Não há	Não existe nenhuma iniciativa de estabelecimento dessa política organizacional e o processo não está maduro o suficiente para este passo ainda.
REP 1.2	Monitorar o envolvimento do <i>stakeholder</i>	NI	Não há	Não é possível afirmar que o monitoramento do envolvimento do <i>stakeholder</i> é feito, apesar dos responsáveis pelo processo terem noção desse envolvimento. Contudo, não existe nada para evidenciar a prática.

REP 1.3	Identificar recursos do projeto	PI	Documento de informações do cliente e Ata de Reunião	A identificação de recursos existe e é evidenciada, inicialmente, na ata de reunião com os <i>stakeholders</i> , onde são feitos acordos, e também em documentações fornecidas pelo cliente, que indicam os recursos que serão disponibilizados por eles. No entanto, ainda não é uma documentação formal, interna e adequada para a demonstração da prática e indica fraquezas no processo.
------------	---------------------------------	----	--	--

REP 1.4	Atribuir responsabilidade	PI	Ata de Reunião	<p>Na prática, os membros da empresa sabem quem são os principais responsáveis pelo processo, porém não há nenhum tipo de organograma ou qualquer outra documentação que evidencie quem é responsável por cada etapa do processo.</p> <p>A ata de reunião possui os nomes dos envolvidos, mas não é suficiente para evidenciar com clareza os responsáveis e suas atribuições. Além dos dados poderem ser conflitantes, por essas atribuições ficarem misturadas e confusas nos documentos.</p>
REP 2.1	Definir programas de treinamento	NY	Não há	<p>Não existe nenhuma iniciativa de definição de treinamentos e o processo não está maduro o suficiente para este passo ainda.</p>
REP 2.2	Promover treinamento	NY	Não há	<p>Não existe nenhuma iniciativa de preparação de treinamentos e o processo não está maduro o suficiente para este passo ainda.</p>

REP 2.3	Estabelecer registro de treinamentos	NY	Não há	Não existe nenhuma iniciativa de registro de treinamentos e o processo não está maduro o suficiente para este passo ainda.
REP 2.4	Avaliar a eficácia dos treinamentos	NY	Não há	Não existe nenhuma iniciativa de avaliação de treinamentos e o processo não está maduro o suficiente para este passo ainda.
REP 3.1	Avaliar processos objetivamente	NI	Não há	A prática de avaliar processos objetivamente não ocorre. Apesar de não haver um processo bem definido na empresa atualmente, esse passo poderia ser feito. Contudo, também é uma prática que ainda não é executada pela organização.
REP 3.2	Avaliar produtos de trabalho objetivamente	NI	Não há	Para avaliar um produto objetivamente, é necessário ter um documento com critérios para essa avaliação. Porém, não há nenhuma evidência que comprove que essa prática ocorre.

REP 3.3	Promover recursos adequados	PI	Quadro do Miro	Essa prática é evidenciada pelo planejamento executado na ferramenta Miro, porém não é um documento oficial e que indica de forma adequada e específica os recursos necessários para o processo.
REP 3.4	Estabelecer um processo para revisar requisitos alocados no projeto	NI	Não há	Não existe nenhuma iniciativa de revisão de requisitos e o processo não está maduro o suficiente para este passo ainda.

Fonte: Autora

Após analisadas e classificadas todas as áreas de processo e suas práticas, os pontos fracos que ficaram evidentes foram:

- PF1.** Pouca ou nenhuma definição clara do que é o processo;
- PF2.** Pouco ou nenhum conhecimento da equipe sobre um bom processo de engenharia de requisitos;
- PF3.** Pouca ou nenhuma transparência do processo internamente e externamente;
- PF4.** Pouca ou nenhuma documentação referente ao processo de engenharia de requisitos;
- PF5.** Nenhuma padronização para as documentações que existem e são geradas pelo processo;
- PF6.** Perda de informações relevantes do processo, incluindo a revisitação de requisitos;
- PF7.** Nenhuma rastreabilidade dos requisitos;
- PF8.** Pouco ou nenhum processo de garantia da qualidade do processo ou dos produtos gerados pelo processo (validação);
- PF9.** Pouco controle sobre as mudanças e/ou dificuldades que ocorrem ao longo do processo e após ele (processo de desenvolvimento);

**PF10.** Dificuldade de inserção de novos membros ao processo.

Esses pontos deixam claro a necessidade que a empresa tem de melhorar o processo. Nas condições atuais, as expectativas de um processo bem definido, documentado, executado e conhecido, além de um desenvolvimento mais fluido de funcionalidades e uma maior satisfação do cliente com o produto final, torna-se impossível, não somente pela falta de conhecimento e organização geral, mas também pela quantidade de envolvidos.

### 5.3.2 Recomendações

Ao ser feito o levantamento dos pontos fracos e, portanto, das dificuldades enfrentadas pelos envolvidos no processo de Engenharia de Requisitos, torna-se necessária a definição de recomendações para a melhoria desse processo:

- REC1.** Definição/Documentação (fluxograma) do processo, de forma geral e dos papéis e responsabilidades dele;
- REC2.** Levantamento de recursos necessários para o processo de requisitos;
- REC3.** Documentação padronizada para levantamento de informações em reuniões;
- REC4.** Documentação padronizada para registro de informações de reuniões (Ata de Reunião);
- REC5.** Documentação padronizada evidenciando conceitos operacionais e cenários;
- REC6.** Definição/Documentação (fluxograma) da modelagem de requisitos;
- REC7.** Documentação padronizada evidenciando a rastreabilidade, a resolução de conflitos, a priorização e as mudanças nos requisitos e justificativas;
- REC8.** Revisão, avaliação e aprovação dos requisitos produzidos (Corretos, completos, consistentes, testáveis e conforme as necessidades do cliente);
- REC9.** Definição/Documentação de uma política organizacional de Engenharia de Requisitos;
- REC10.** Definição/Documentação (fluxograma) de monitoramento do envolvimento do *stakeholder*;
- REC11.** Definição/Documentação, execução e avaliação de treinamentos do processo;
- REC12.** Definição/Documentação (fluxograma) de avaliação do processo e produtos do processo (melhoria contínua);
- REC13.** Definição/Documentação (fluxograma) de um processo para revisão de requisitos alocados no projeto.



## 5.4 Estabelecimento

Para a fase de estabelecimento, registra-se neste trabalho a primeira atividade de estabelecimento de prioridade, onde há a definição das ações de melhoria para o processo de engenharia de requisitos e a priorização das determinadas ações. Logo depois, apresenta-se o planejamento dessas ações a serem executadas no primeiro ciclo de melhoria do processo.

### 5.4.1 Definir ações de melhoria

Após feitas as recomendações na fase do diagnóstico, é possível definir as ações de melhoria que serão executadas no plano de melhoria do processo.

#### 5.4.1.1 Área de Processo - Elicitação

##### A1. Planejar a reunião de levantamento de necessidades

- Preparar perguntas para guiar a reunião;
- Criar documento para registrar os pontos discutidos (Ata de Reunião padrão).
  - \* Informações relevantes de *stakeholders* (nome, função, interesses, etc);
  - \* Perguntas feitas e respostas dadas;
  - \* Necessidades e interesses expressados;
  - \* Dúvidas, dificuldades e interesses conflitantes levantados entre *stakeholders*;
  - \* Recursos pedidos para o cliente apoiar na análise, especificação e validação dos requisitos.

##### A2. Identificar e classificar *stakeholders*

- Fazer uma lista de quem são os *stakeholders*;
- Entender os interesses e expectativas de cada um;
- Classificar *stakeholders* por ordem de influência;
- Planejar como interagir com cada stakeholder;
- Levantar os riscos que podem ser gerados por cada stakeholder.

#### 5.4.1.2 Área de Processo - Análise

##### A3. Criar documentação de estabelecimento de conceitos operacionais e cenários do produto

##### A4. Planejar e executar a modelagem de requisitos

- Selecionar e criar modelos (ex.: diagramas) para apoiar na interpretação do fluxo do produto/projeto;
- Resolver conflitos de requisitos;
- Documentar requisitos, categorizando em funcionais e não-funcionais.

**A5.** Fazer a priorização dos requisitos

**A6.** Fazer alinhamento sobre os requisitos com a equipe interna

- Documentar os impactos, compromissos e as mudanças dos requisitos;
- Criar matriz de rastreabilidade dos requisitos.

#### 5.4.1.3 Área de Processo - Especificação

**A7.** Gerar um documento interno que deixe claro o processo de definição dos requisitos e os requisitos em si

- Indicar rastreabilidade dos requisitos e anexar matriz de rastreabilidade.

**A8.** Gerar uma proposta de solução, baseada nos requisitos definidos

#### 5.4.1.4 Área de Processo - Validação

**A9.** Fazer validação interna dos requisitos com a equipe técnica e recolher pontos levantados

**A10.** Fazer validação de requisitos com os *stakeholders*

- Enviar proposta de solução;
- Enviar documento de especificação de requisitos;
- Fazer apresentação de proposta de solução e recolher pontos levantados.

**A11.** Fazer alterações necessárias (retorno equipe técnica e *stakeholders*)

#### 5.4.1.5 Área de Processo - Gestão de Requisitos

**A12.** Gerenciar mudanças nos requisitos

- Atualizar documentação de requisitos, evidenciando o histórico de mudanças;
- Avaliar impacto das mudanças.

**A13.** Gerenciar a rastreabilidade dos requisitos

- Gerenciar matriz de rastreabilidade.

#### 5.4.1.6 Área de Processo - Suporte Organizacional

**A14.** Criar e estabelecer uma Política Organizacional de Engenharia de Requisitos

**A15.** Monitorar o envolvimento dos *stakeholders*

- Definir momentos específicos de interação com os *stakeholders*;
- Registrar interações fora das definidas inicialmente.

**A16.** Identificar recursos necessários para execução do projeto

- Promover recursos adequados;
- Documentar recursos necessários.

**A17.** Atribuir responsabilidades

- Definir os responsáveis pelo desenvolvimento do projeto

**A18.** Definir programas de treinamento

- Promover treinamento;
- Estabelecer registro de treinamentos;
- Avaliar a eficácia dos treinamentos.

**A19.** Avaliar processos objetivamente

- Criar modelo de avaliação de processos;
- Executar avaliação de processos.

**A20.** Avaliar produtos de trabalho objetivamente

- Criar modelo de avaliação de produtos;
- Executar avaliação de produtos interna;
- Executar avaliação de produtos junto aos *stakeholders*.

**A21.** Revisar requisitos alocados no projeto

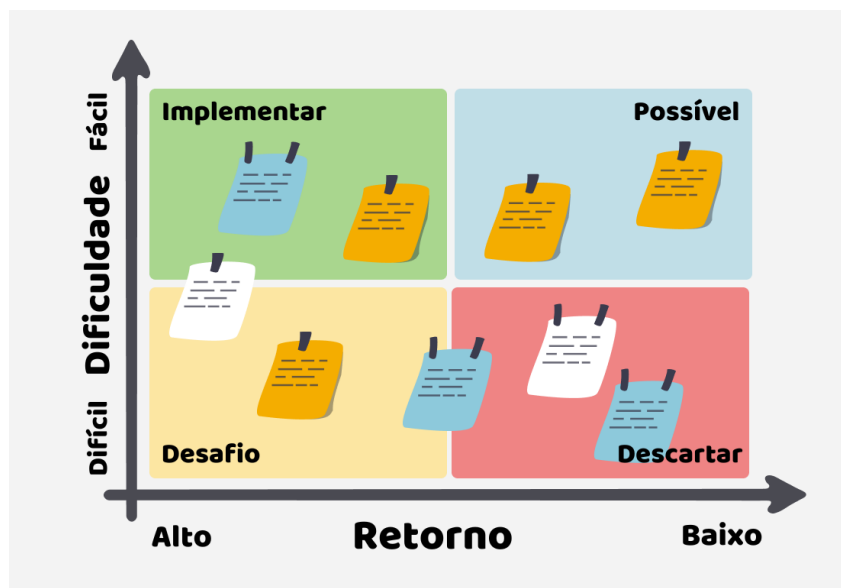
- Estabelecer processo de revisão dos requisitos alocados;
- Executar processo de revisão.

### 5.4.2 Estabelecimento de prioridades

Com a definição das ações, o próximo e um importante passo da melhoria do processo, é fazer a priorização de ações para definição do 1º ciclo de melhoria.

Para essa priorização, foi utilizado como base o quadrante do PICK Chart, exemplificado na Figura 25 e detalhado na Tabela 12.

Figura 25 – QUADRANTE DO PICK CHART (EXEMPLO).



Fonte: Adaptado de [Krumina \(2022\)](#)

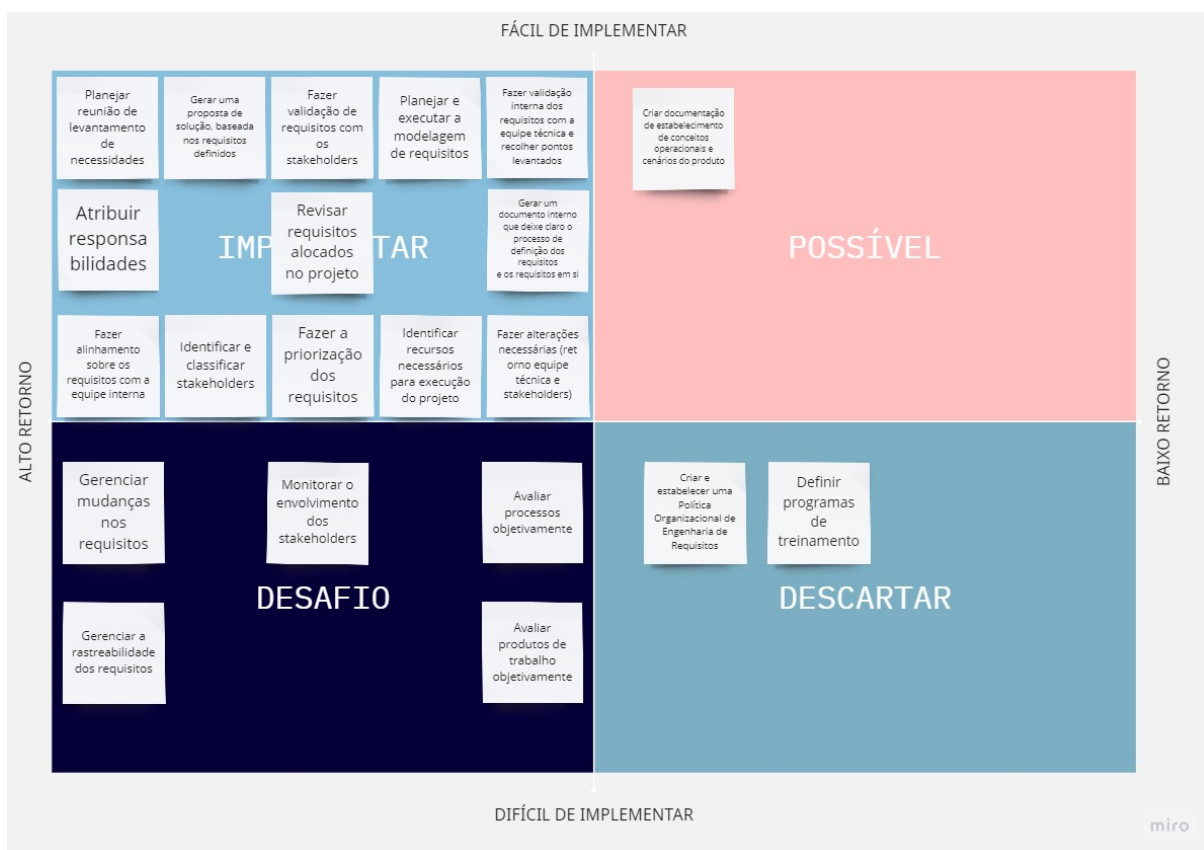
Tabela 12 – LEGENDA PARA PRIORIZAÇÃO DE AÇÕES.

Quadrante	Implementação	Retorno	Impacto no planejamento
Possível (Possible)	Fácil	Baixo	As ações não são implementadas, pois têm baixo retorno.
Implementar (Implement)	Fácil	Alto	As ações são obrigatoriamente implementadas, pois têm alto retorno e são fáceis de incluir no planejamento. Devem ser alocadas no primeiro ciclo de melhoria.
Desafio (Challenge)	Difícil	Alto	A implementação dessas ações é definida com base em uma discussão com a empresa para analisar a viabilidade de inclusão delas no processo. Apesar de apresentarem alto retorno, possuem difícil implementação.
Descartar (Kill)	Difícil	Baixo	As ações não são implementadas, pois possuem baixo retorno e alta dificuldade de implementação.

Fonte: Adaptado do PICK Chart

A partir das definições do PICK Chart, foi construída uma matriz na ferramenta Miro e lá foi feita a priorização das ações, seguindo o modelo. É possível ver o resultado dessa priorização na Figura 26:

Figura 26 – MATRIZ DO PICK CHART (MIRO)



Fonte: Autora

Por fim, então, a seguir a priorização das ações é apresentada por área de processo:

Tabela 13 – PRIORIZAÇÃO DE AÇÕES POR ÁREA DE PROCESSO.

Área de Processo	Ação	Priorização
Elicitação	A1	Implementar
	A2	Implementar
Análise	A3	Possível
	A4	Implementar
	A5	Implementar
	A6	Implementar
Especificação	A7	Implementar
	A8	Implementar

Validação	A9	Implementar
	A10	Implementar
	A11	Implementar
Gestão de Requisitos	A12	Desafio
	A13	Desafio
Suporte Organizacional	A14	Descartar
	A15	Desafio
	A16	Implementar
	A17	Implementar
	A18	Descartar
	A19	Desafio
	A20	Desafio
	A21	Implementar

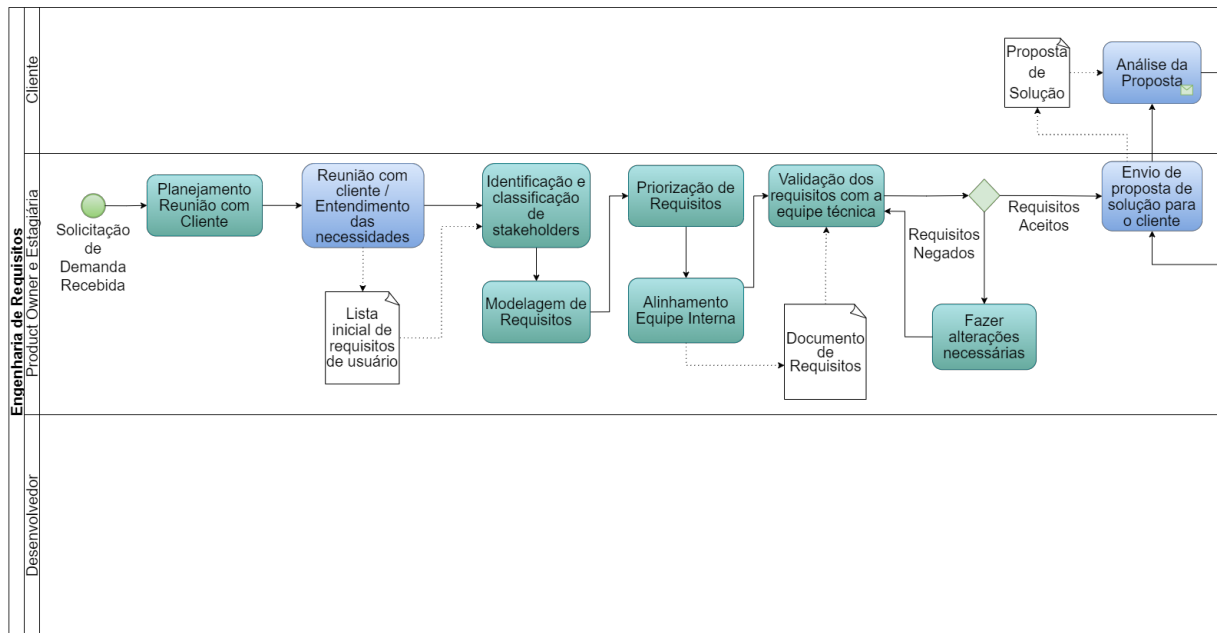
Fonte: Autora

### 5.4.3 Planejamento de ações

Após definida a priorização, foi feito o planejamento de ações do 1º ciclo de melhoria, representado por um BPMN atualizado, ilustrado nas Figuras 27 e 28. Para este ciclo, foram selecionadas as ações priorizadas como "Implementar".

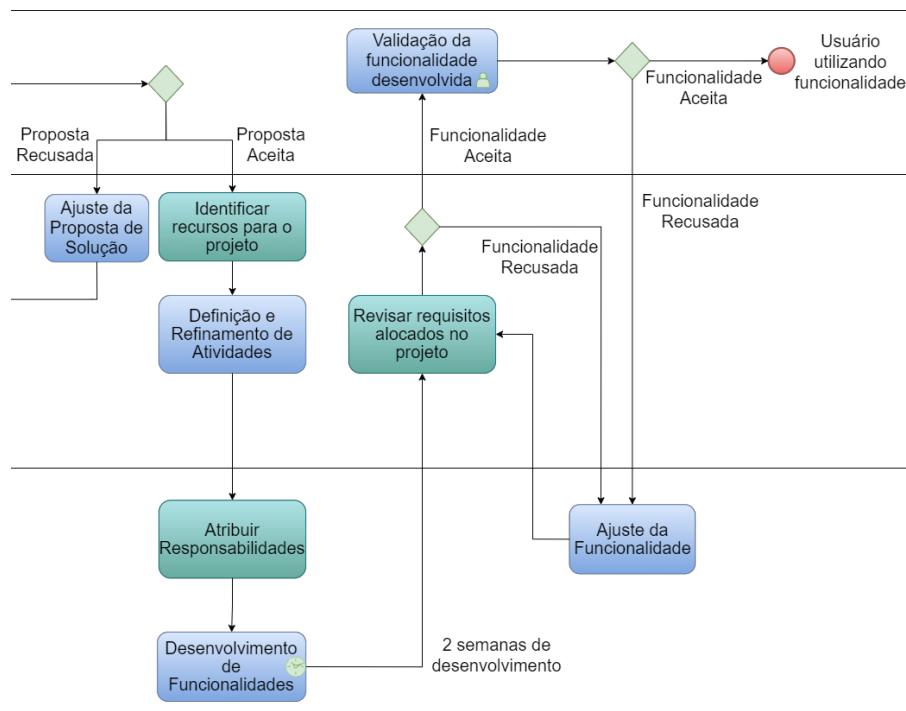
Neste BPMN foram acrescentadas as novas ações que passarão a fazer parte do processo da empresa ATR e elas estão representadas pela cor verde, com mais destaque comparado às ações que já existiam antes no processo, indicadas na cor azul e já ilustradas no processo inicial da Figura 24.

Figura 27 – PLANEJAMENTO 1º CICLO DE MELHORIA - PARTE 1.



Fonte: Autora

Figura 28 – PLANEJAMENTO 1º CICLO DE MELHORIA - PARTE 2.



Fonte: Autora

## 5.5 Ação

Na última fase abordada neste trabalho é apresentada a criação da solução técnica e o plano de medição do ciclo de melhoria. A parte de aplicação e testagem dessa fase ficará sob total responsabilidade da empresa ATR, se for de seu interesse.

### 5.5.1 Criação da Solução Técnica

O objetivo da criação da solução técnica, neste trabalho, é detalhar como será executado o 1º ciclo de melhoria na empresa ATR e quais são as adaptações que precisarão ser feitas para que esse ciclo ocorra conforme o planejado.

Antes do início do 1º ciclo será necessário que a equipe de requisitos, que já estará ciente deste estudo, prepare um treinamento, visando disseminar o conhecimento para os demais envolvidos e explicar como será executado o processo daquele momento em diante.

Para a primeira ação incluída no processo: *Preparação para a reunião com o cliente*, deverá ser organizada uma nova sessão interna entre a equipe de requisitos, onde serão definidas e revisadas as perguntas que servirão de roteiro para o entendimento completo das necessidades do cliente na reunião. Essas perguntas serão documentadas na ferramenta Clickup e deverão ser baseadas, não somente em experiências anteriores, mas também deverão ter um embasamento mais teórico. Logo, será necessário procurar modelos de roteiros de entrevistas sobre levantamento de requisitos.

Com relação à segunda ação incluída: *Identificação e classificação de stakeholders*, a sugestão é, também, fazer uma sessão interna, onde o intuito será identificar os *stakeholders* e classificá-los. Para isso, é indicado o uso da ferramenta Miro, na qual serão levantados critérios para a classificação, considerando a influência deles para o projeto em questão, minimizando conflitos nas decisões de escopo.

Para as ações de *Modelagem e Priorização de Requisitos*, atualmente executadas com o auxílio da ferramenta Miro, a sugestão de mudança é com relação à base teórica para a execução dessa etapa, pois, atualmente, não há referências tão bem infundadas que garantam a eficiência dela. Logo, será necessário buscar técnicas utilizadas para uma melhor modelagem e priorização de requisitos como, por exemplo: Matriz de rastreabilidade de requisitos, Casos de uso, Histórias de usuários, Diagramas de fluxo de processo, Diagramas de atividades e Diagramas de sequência.

A ação de *Alinhamento da Equipe Interna*, que foi incluída, visa a formalização dos requisitos e a criação de uma documentação no Clickup sobre o seu detalhamento. A sugestão é que eles tenham uma mínima rastreabilidade para serem consultados por todos, sempre que possível, de forma prática e que, posteriormente, possam ser linkados à atividade, criada no Clickup, na qual estão sendo desenvolvidos.



Quanto à ação de *Geração de um documento interno sobre o processo de requisitos e requisitos*, a sugestão é a criação de um documento que esteja armazenado em uma pasta do clickup, de preferência, com o nome do cliente para o qual aqueles requisitos estão direcionados e que o título do documento seja referente ao nome do projeto. O ideal seria que este documento apresentasse, inicialmente, os requisitos de usuário e sua evolução até se tornarem requisitos de sistema, ou seja, com evidências de sua modelagem e priorização. Outra sugestão é que os requisitos finais sejam rastreáveis, tenham um código de identificação (Ex.: REQ1).

Com relação à ação de *Validação de Requisitos com a Equipe Técnica*, será necessário organizar uma nova sessão entre a equipe de requisitos e as peças-chave da equipe técnica para decidirem sobre os requisitos, ver se são viáveis para a etapa de desenvolvimento, se eles estão claros e se não há mais nenhuma dúvida com relação a eles.

Sobre a ação de *Geração de Proposta de solução baseada nos requisitos definidos*, a sugestão é aproveitar o modelo de proposta já existente feito pela empresa, porém ajustá-lo para ser mais detalhado e melhor construído, considerando que os requisitos estarão, também, melhor modelados.

Na ação de *Validação de requisitos com os stakeholders* será necessário que já tenha sido feita a validação com a equipe técnica antes, pois, dessa forma, os requisitos já estarão melhor pensados e os stakeholders vão poder fazer uma validação mais precisa em cima do que for apresentado a eles.

Para a ação *Fazer alterações necessárias (retorno equipe técnica e stakeholders)*, a sugestão é fazer a coleta de todos os pontos de observação e ajuste que forem levantados, tanto pela equipe técnica, quanto pelos *stakeholders* e fazer as modificações no documento de requisitos e na proposta de solução.

Com relação à *Identificação de Recursos do Projeto*, será necessário fazer uma revisão de todos os esforços comprometidos e um levantamento dos recursos necessários e, posteriormente, a garantia desses recursos para o projeto. A sugestão é a criação de uma lista, documentada na ferramenta Clickup, baseada nos requisitos fundados, classificando recursos humanos, ferramentas, recursos de capacitação da equipe, etc.

Considerando que, neste novo processo, a equipe de desenvolvimento têm mais conhecimento dos requisitos e, conseqüentemente, das atividades que serão desenvolvidas na *sprint*, além de conhecer também os recursos necessários para a execução do desenvolvimento, a ação de *Atribuir Responsabilidades* torna-se mais eficiente e eficaz. As atribuições serão registradas pela ferramenta Clickup.

A última ação incluída: *Revisar requisitos alocados no projeto*, será executada pela equipe de requisitos, por meio da atualização do documento de requisitos, alocado no Clickup. O intuito é analisar o requisito, ver se ele foi implementado corretamente e

incluir na documentação uma evidência de sua implementação, podendo ser *prints*, *gifs* ou vídeos.

### 5.5.2 Plano de Medição

Para finalizar a fase ação, foi feito um plano de medição para avaliar a melhoria contínua do processo de forma quantitativa. O plano foi feito com a utilização da metodologia OKR, apresentada no Capítulo 2 e foi baseado nas expectativas de benefício, introduzidas na seção 5.2.3.

Os objetivos definidos para o plano de medição, com seus respectivos *key-results* (resultados-chave) foram:

Tabela 14 – OKR 1.

<b>O1</b>	<b>Processo de Engenharia de Requisitos Definido e Documentado</b>
<b>KR 1</b>	Criar 1 documentação do processo de engenharia de requisitos da empresa
<b>KR 2</b>	Organizar 1 sessão de validação do processo com a equipe
<b>KR 3</b>	Criar versão final do processo

Fonte: Autora

Tabela 15 – OKR 2.

<b>O2</b>	<b>Processo de Engenharia de Requisitos Executado</b>
<b>KR 1</b>	Escolher 1 requisito para avaliar se o processo está sendo executado
<b>KR 2</b>	Criar 1 documento com as evidências da ocorrência de cada etapa do processo
<b>KR 3</b>	100% das evidências documentadas

Fonte: Autora

Tabela 16 – OKR 3.

<b>O3</b>	<b>Requisitos Documentados</b>
<b>KR 1</b>	Criar 1 pasta de documentos de requisitos no clickup
<b>KR 2</b>	Criar 1 template padrão da estrutura de documentação de requisitos
<b>KR 3</b>	Aplicar nova forma de documentação a 1 projeto
<b>KR 4</b>	Ter 100% dos requisitos do projeto descritos no documento

Fonte: Autora

Tabela 17 – OKR 4.

<b>O4</b>	<b>Trabalho mais fluido dos desenvolvedores e atividade feitas dentro do tempo esperado</b>
<b>KR 1</b>	Mapear as principais dúvidas sobre os requisitos ao longo de 1 sprint
<b>KR 2</b>	Mapear atrasos de atividades ao longo de 1 sprint
<b>KR 3</b>	Gerar 1 documento de registro das dúvidas
<b>KR 4</b>	Organizar e executar 1 sessão de refinamento com foco nas principais dúvidas, antes da próxima sprint

Fonte: Autora

Tabela 18 – OKR 5.

<b>O5</b>	<b>Conhecimento da equipe quanto às prioridades dos requisitos e das atividades com antecedência</b>
<b>KR 1</b>	Compartilhar documentação do processo de engenharia de requisitos com 100% dos colaboradores
<b>KR 2</b>	Compartilhar documentação dos requisitos definidos com 100% dos colaboradores
<b>KR 3</b>	Organizar 1 sessão de validação de requisitos com a equipe técnica antes de cada sprint
<b>KR 4</b>	Promover 2 sessões de refinamento antes da sprint

Fonte: Autora

Tabela 19 – OKR 6.

<b>O6</b>	<b>Garantir a satisfação dos clientes</b>
<b>KR 1</b>	Criar 1 modelo de avaliação de funcionalidades para enviar aos clientes
<b>KR 2</b>	Enviar o modelo os clientes de 1 projeto por e-mail
<b>KR 3</b>	Criar 1 documento de análise da avaliação
<b>KR 4</b>	Fazer revisão e ajustes de 100% dos requisitos do projeto

Fonte: Autora

Tabela 20 – OKR 7.

<b>O7</b>	<b>Garantir que os requisitos estão corretos</b>
<b>KR 1</b>	Organizar 1 sessão de validação com a equipe interna após cada sprint
<b>KR 2</b>	Incluir 1 evidência da validação para cada requisito no documento de requisitos

Fonte: Autora

Considerando que os objetivos de medição foram criados para, futuramente, alcançarem as expectativas de benefício, foram relacionados esses objetivos a cada expectativa de benefício do início do capítulo.

O O1 e o O3, apresentados nas Tabelas 14 e 16, contempla as expectativas de um processo definido e documentado e o O2, apresentado na Tabela 15, a expectativa de um processo executado.

O objetivo O4, apresentado na Tabela 17, visa a expectativa de um trabalho mais fluido entre os desenvolvedores, além de atividades feitas dentro do tempo/prazo esperado.

O objetivo O5, apresentado na Tabela 18, tem o objetivo de alcançar a expectativa de um processo conhecido pela equipe, onde são conhecidas as prioridades dos requisitos e as atividades com antecedência.

O objetivo O6, apresentado na Tabela 19, visa alcançar a expectativa de garantir a satisfação dos clientes com o resultado final do produto.

Por último, o O7, apresentado na Tabela 20, contempla a expectativa de requisitos corretos, onde eles serão validados, comparando o seu princípio e o seu resultado final.

## 5.6 Lições

Esta etapa compreende a análise e validação da solução e a proposição de ações futuras. Ela não é abordada neste trabalho e será realizada independentemente pela empresa ATR, se for de seu interesse.

## 6 Considerações Finais

Neste trabalho foi possível notar a influência que a qualidade de processos tem no resultado que chega para os clientes. Por esse motivo, empresas precisam ter muita organização para enfrentar os desafios e a competitividade do mercado.

É importante destacar, também, que a melhoria de processos e produtos não deixa de existir, é um processo contínuo: é analisado e aprimorado a cada ciclo de desenvolvimento e, para estarmos sempre atentos a estas oportunidades de melhorias, usamos métodos e técnicas que nos auxiliam nesta constante missão.

Neste trabalho, ainda, ficou clara a importância da fase de engenharia de requisitos em um processo de desenvolvimento de software e o quanto ela impacta na opinião final do cliente sobre o produto desenvolvido.

Ao levar tudo isso em consideração, foi definida a seguinte questão de pesquisa (Seção 1.2): *"Como identificar oportunidades de melhoria no processo de engenharia de requisitos da empresa a partir da análise do processo?"*

E, para iniciar o processo de pensamento que corresponde à resposta para esta pergunta, também foram determinados os objetivos geral e específicos do trabalho (Seção 1.3).

Para contemplar o objetivo geral de *"Propor melhorias de processo de Engenharia de Requisitos na empresa ATR"*, ainda foram executadas algumas ações que concluem os objetivos específicos:

Para concluir o objetivo específico *"Mapear do processo de requisitos atual"* foi esquematizado e apresentado o processo atual de Engenharia de Requisitos da empresa, levando em consideração os relatos dos entrevistados para o trabalho (Subseção 5.3.1.1).

Para o objetivo específico *"Identificar fragilidades do processo de requisitos atual"*, foram analisadas gravações de reuniões, documentos gerados pelo processo e as respostas das entrevistas feitas. Além disso, também foi possível entender melhor sobre essas fragilidades na fase do diagnóstico do modelo IDEAL e, ao final dela, os pontos foram apontados (Subseção 5.3.1.3).

Com relação ao objetivo específico *"Construir uma linha de base de maturidade atual do processo"* foi definida uma forma de análise da maturidade do processo, baseada na escala FILIPINY, do método SCAMPI (Subseção 5.3.1.2).

A partir do objetivo anterior, foi possível concluir, também, o objetivo específico de *"Fazer o diagnóstico do processo atual, identificando oportunidades de melhoria"*, pois

foi feita a análise e, além de terem sido apontados os gargalos do processo, também foi possível encontrar as oportunidades de melhoria (Subseções 5.3.1.4 e 5.3.2).

Quanto ao objetivo específico de "*Identificar formas de avaliação do processo de requisitos na empresa (Métodos, técnicas, modelos)*", foram identificados e utilizados a escala FILIPINY, do método SCAMPI e o modelo SRP-CMMI, para a avaliação inicial do processo (Subseções 2.4.5.2 e 5.3.1.2), o PICK Chart para priorização de ações do processo (Subseção 5.4.2) e também a metodologia OKR para a medição do processo (Subseções 2.5.3.1 e 5.5.2).

Sobre o último objetivo específico: "*Propor soluções técnicas para as oportunidades de melhoria do processo atual de requisitos*", foram definidas e priorizadas ações de melhoria (Subseção 5.4.2), a partir disso foi criado e esquematizado o 1º ciclo de melhoria do processo de Engenharia de Requisitos da empresa ATR (Subseção 5.4.3) e, após isso, foi possível criar propostas de soluções técnicas para a realização desse 1º ciclo de melhoria (Subseção 5.5.1), além da criação também de um plano de medição para avaliar a melhoria contínua do processo (Subseção 5.5.2).

Com todos esses objetivos específicos concluídos e, por consequência, com o objetivo geral concluído, foi possível identificar e propor diversas melhorias em um processo que parecia confuso e sem solução.

Em conclusão, cabe aqui dizer que a melhoria de processos não é um trabalho fácil, mas é muito compensatório e merece mais atenção no mundo corporativo atual. Processos não melhoram se não os analisarmos bem e produtos não podem ser bem feitos se seus processos de execução não estão bem definidos e implementados.

## Referências

- AHMAD, M. O. et al. Transition of software maintenance teams from scrum to kanban. In: *2016 49th Hawaii International Conference on System Sciences (HICSS)*. [S.l.: s.n.], 2016. p. 5427–5436. ISSN: 1530-1605. Citado na página 28.
- ALHAZMI, A.; HUANG, S. Survey on differences of requirements engineering for traditional and agile development processes. In: IEEE. *2020 SoutheastCon*. [S.l.], 2020. p. 1–9. Citado na página 25.
- ANDRADE, M. *Qualidade De Software*. 1. ed. Rio de Janeiro: ESTACIO EDITORA, 2011. ISBN 978-85-5548-089-8. Disponível em: <<https://books.google.com.br/books?id=HYtyzgEACAAJ>>. Acesso em: 2023-01-02. Citado 4 vezes nas páginas 29, 30, 49 e 50.
- BECK, K. *Programacao Extrema (Xp) Explicada*. 1. ed. [S.l.]: Bookman, 2004. ISBN 978-85-363-0387-1. Acesso em: 2023-02-05. Citado 2 vezes nas páginas 22 e 23.
- BEECHAM, S.; HALL, T.; AUSTEN, R. Defining a Requirements Process Improvement Model. *Software Quality Journal*, v. 13, n. 3, p. 247–279, set. 2005. ISSN 09639314. Num Pages: 247-279 Place: Dordrecht, Netherlands Publisher: Springer Nature B.V. Disponível em: <<https://www.proquest.com/docview/235100481/abstract/509F3D224F1B4E44PQ/1>>. Acesso em: 2023-01-11. Citado 4 vezes nas páginas 41, 42, 43 e 44.
- CAMPOS, V. F. *TQC- Controle da Qualidade Total no estilo japonês*. 9. ed. [S.l.]: Falconi Editora, 2014. ISBN 978-85-98254-84-5. Citado na página 50.
- CARNEGIE MELLON UNIVERSITY. *SCAMPI: Standard cmmi appraisal method for process improvement (scampi) a, version 1.3: Method definition document*. Pittsburgh, PA, 2011. Disponível em: <<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9703>>. Acesso em: 2023-01-13. Citado 2 vezes nas páginas 70 e 71.
- CASTRO, R. M. d. *Apresentação sobre CMM e CMMI*. 2012. Apresentação do Power Point. Disponível em: <<https://pt.slideshare.net/RonneyMoreiradeCastro/cmm-e-cmmi>>. Acesso em: 2023-02-04. Citado na página 37.
- CHOWDHURY, A. F.; HUDA, M. N. Comparison between adaptive software development and feature driven development. In: *Proceedings of 2011 International Conference on Computer Science and Network Technology*. [S.l.: s.n.], 2011. v. 1, p. 363–367. Acesso em: 2023-01-15. Citado na página 26.
- COAD, P.; LUCA, J. d.; LEFEBVRE, E. *Java modeling color with UML: Enterprise components and process with Cdrom*. [S.l.]: Prentice Hall PTR, 1999. Citado na página 26.
- DINGSOYR, T. (Ed.). *Software Process Improvement*. Springer, 2004. (Lecture Notes in Computer Science, v. 3281). Disponível em: <<http://link.springer.com/10.1007/b102170>>. Acesso em: 2022-12-03. Citado na página 34.

EMAM, K. E.; MADHAVJI, N. H. A field study of requirements engineering practices in information systems development. In: IEEE. *Proceedings of 1995 IEEE International Symposium on Requirements Engineering (RE'95)*. [S.l.], 1995. p. 68–80. Citado na página 17.

FLICK, U. *Introdução à Metodologia de Pesquisa*. Penso Editora LTDA, 2013. ISBN 978-85-65848-13-8. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788565848138/pageid/22>>. Acesso em: 2023-01-17. Citado 3 vezes nas páginas 55, 57 e 58.

FONTELLES, M. J. et al. Metodologia da pesquisa científica: diretrizes para a elaboração de um protocolo de pesquisa. *Revista paraense de medicina*, Amazônia, v. 23, n. 3, p. 1–8, 2009. Citado na página 57.

FUGGETTA, A. Software process: a roadmap. In: *Proceedings of the Conference on The Future of Software Engineering*. New York, NY, USA: ACM, 2000. p. 25–34. ISBN 978-1-58113-253-3. Disponível em: <<https://dl.acm.org/doi/10.1145/336512.336521>>. Acesso em: 2023-08-11. Citado na página 33.

GALLOTTI, G. M. A. *Qualidade de Software*. Pearson Education do Brasil LTDA, 2017. ISBN 978-85-430-2035-8. Disponível em: <<https://www.bibliotecavirtual.com.br/>>. Acesso em: 2023-01-02. Citado 5 vezes nas páginas 20, 34, 50, 52 e 53.

GIL, A. C. *Como Elaborar Projetos de Pesquisa*. 7. ed. Editora Atlas Ltda., 2022. ISBN 9786559771653. Disponível em: <[https://integrada.minhabiblioteca.com.br/reader/books/9786559771653/epubcfi/6/2\[%3Bvnd.vst.idref%3Dhtml1!\]/4/2/2%4051:2](https://integrada.minhabiblioteca.com.br/reader/books/9786559771653/epubcfi/6/2[%3Bvnd.vst.idref%3Dhtml1!]/4/2/2%4051:2)>. Acesso em: 2022-11-27. Citado 2 vezes nas páginas 55 e 58.

GREMBA, J.; MYERS, C. The ideal model: A practical guide for improvement. *Software Engineering Institute Bridge*, 1997. Citado na página 50.

HERNANDES, P. *O que é Kanban? A melhor forma de organizar tarefas e projetos!* 2020. Disponível em: <<https://hbfs.com.br/o-que-e-kanban/>>. Acesso em: 2023-01-22. Citado na página 29.

HOFMANN, C. et al. Development of an agile development method based on kanban for distributed part-time teams and an introduction framework. v. 23, p. 45–50, 2018. ISSN 2351-9789. Publisher: Elsevier. Disponível em: <<https://www.sciencedirect.ez54.periodicos.capes.gov.br/science/article/pii/S2351978918304633>>. Acesso em: 2023-01-14. Citado 2 vezes nas páginas 24 e 25.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 25010:2011(en)*: Systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models. Geneva, CH, 2011. Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>>. Acesso em: 2022-12-22. Citado 3 vezes nas páginas 30, 31 e 32.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Standard, *ISO/IEC 33001:2015*: Information technology — process assessment — concepts and terminology. Geneva, CH, 2015. Disponível em: <<https://www.iso.org/standard/54175.html>>. Acesso em: 2023-01-06. Citado na página 30.



- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 12207:2017(en)*: Systems and software engineering — software life cycle processes. Geneva, CH, 2017. Disponível em: <<https://www.iso.org/standard/63712.html>>. Acesso em: 2023-01-06. Citado 2 vezes nas páginas 30 e 33.
- JAIN, P.; SHARMA, A.; AHUJA, L. The Impact of Agile Software Development Process on the Quality of Software Product. In: *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. [S.l.: s.n.], 2018. Citado na página 22.
- KANWAL, F.; JUNAID, K.; FAHIEM, M. A. A hybrid software architecture evaluation method for FDD - an agile process model. In: *2010 International Conference on Computational Intelligence and Software Engineering*. [S.l.: s.n.], 2010. p. 1–5. Acesso em: 2023-01-29. Citado na página 27.
- KAUTZ, K. et al. Networked technologies — the role of networks in the diffusion and adoption of software process improvement (SPI) approaches. In: DAMSGAARD, J.; HENRIKSEN, H. Z. (Ed.). *Networked Information Technologies: Diffusion and Adoption*. Boston, MA: Springer US, 2004, (IFIP International Federation for Information Processing). p. 203–211. ISBN 978-1-4020-7862-0. Disponível em: <[https://doi.org/10.1007/1-4020-7862-5\\_13](https://doi.org/10.1007/1-4020-7862-5_13)>. Acesso em: 2023-08-22. Citado na página 16.
- KERR, E. S. (Ed.). *Gerenciamento de Requisitos*. Pearson Education do Brasil LTDA, 2015. Disponível em: <<https://plataforma.bvirtual.com.br/Acervo/Publicacao/22130>>. Acesso em: 2023-01-15. Citado 2 vezes nas páginas 19 e 20.
- KONTIO, J.; CONRADI, R. *Software Quality - ECSQ 2002*. Springer, 2002. v. 2349. ISBN 3-540-43749-5. Disponível em: <<https://link-springer-com.ez54.periodicos.capes.gov.br/book/10.1007/3-540-47984-8>>. Acesso em: 2022-11-29. Citado na página 23.
- KRUMINA, K. *What is a PICK chart in project management?* 2022. Disponível em: <<https://deskttime.com/blog/pick-chart>>. Acesso em: 2023-14-12. Citado na página 91.
- LARMAN, C. Agile and iterative development: A manager's guide. In: . [S.l.: s.n.], 2003. Citado na página 26.
- LERCHE-JENSEN, S. *Fundamentos Internacionais do Scrum Master - 3.7 Passo a passo de uma Iteração*. 2019. Disponível em: <<https://www.scrum.as/academy.php?show=5&chapter=12>>. Acesso em: 2023-01-29. Citado na página 26.
- M. Lakatos; E. Lakatos. *Técnicas de Pesquisa*. 9. ed. Atlas LTDA, 2021. ISBN 978-85-970266-0-3. Disponível em: <[https://integrada.minhabiblioteca.com.br/reader/books/9788597026610/epubcfi/6/24\[%3Bvnd.vst.idref%3Dhtml10\]!/4/4/2](https://integrada.minhabiblioteca.com.br/reader/books/9788597026610/epubcfi/6/24[%3Bvnd.vst.idref%3Dhtml10]!/4/4/2)>. Acesso em: 2023-01-19. Citado 2 vezes nas páginas 56 e 57.
- MARTIN, C. D. Lessons from chernobyl for it. *SIGCSE Bull.*, Association for Computing Machinery, New York, NY, USA, v. 36, n. 2, p. 11, jun 2004. ISSN 0097-8418. Disponível em: <<https://doi.org/10.1145/1024338.1024342>>. Acesso em: 2023-08-16. Citado na página 27.

MUNIZ, A. et al. *Jornada OKR na Prática: unindo práticas e experiências que potencializam resultados*. Rio de Janeiro: Brasport, 2022. ISBN 9786588431467. Disponível em: <[https://books.google.com.br/books?hl=pt-BR&lr=&id=-PZVEAAAQBAJ&oi=fnd&pg=PA2&dq=okr&ots=OHHEK-LYFL&sig=FCW6c-EJbiwkdnNUYpevvi9YEcA&redir\\_esc=y#v=onepage&q=okr&f=false](https://books.google.com.br/books?hl=pt-BR&lr=&id=-PZVEAAAQBAJ&oi=fnd&pg=PA2&dq=okr&ots=OHHEK-LYFL&sig=FCW6c-EJbiwkdnNUYpevvi9YEcA&redir_esc=y#v=onepage&q=okr&f=false)>. Citado 2 vezes nas páginas 53 e 54.

N. Damij; T. Damij. An approach to optimizing kanban board workflow and shortening the project management plan. p. 1–8, 2021. ISSN 1558-0040. Conference Name: IEEE Transactions on Engineering Management. Acesso em: 2023-01-14. Citado na página 28.

NAJJAR, S. K.; AL-SARAYREH, K. T. Capability maturity model of software requirements process and integration (SRPCMMI). In: *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*. Association for Computing Machinery, 2015. (IPAC '15), p. 1–5. ISBN 978-1-4503-3458-7. Disponível em: <<http://doi.org/10.1145/2816839.2816856>>. Acesso em: 2022-11-27. Citado 5 vezes nas páginas 41, 45, 46, 47 e 48.

NIKULA, U. et al. Introducing basic systematic requirements engineering practices in small organizations with an easy to adopt method. 2004. Citado na página 16.

OIVO, M. *Product Focused Software Process Improvement: 4th International Conference, PROFES 2002 Rovaniemi, Finland, December 9-11, 2002, Proceedings*. [S.l.]: Springer Science & Business Media, 2002. v. 4. Citado na página 22.

OLIVEIRA, W. *O que é scrum? Conceito, definições e etapas*. 2019. Disponível em: <<https://evolvemvp.com/o-que-e-scrum-conceito-definicoes-e-etapas/>>. Acesso em: 2023-01-22. Citado na página 25.

PALMER, S.; FELSING, J. A practical guide to feature-driven development. prentice hall. *Upper Saddle Hill*, 2002. Citado na página 26.

PAULA FILHO, W. d. P. *Engenharia de Software - Projetos e Processos - Vol. 2*. 4. ed. LTC — Livros Técnicos e Científicos Editora Ltda., 2019. v. 2. ISBN 978-85-216-3673-1. Disponível em: <[https://integrada.minhabiblioteca.com.br/reader/books/9788521636748/epubcfi/6/26\[%3Bvnd.vst.idref%3Dchapter1\]/4/78/4%4054:100](https://integrada.minhabiblioteca.com.br/reader/books/9788521636748/epubcfi/6/26[%3Bvnd.vst.idref%3Dchapter1]/4/78/4%4054:100)>. Acesso em: 2022-12-03. Citado 4 vezes nas páginas 16, 35, 50 e 51.

PAULK, M. et al. Capability maturity model, version 1.1. *IEEE Software*, v. 10, n. 4, p. 18–27, jul. 1993. ISSN 1937-4194. Conference Name: IEEE Software. Citado 3 vezes nas páginas 16, 34 e 35.

POPPENDIECK, M.; POPPENDIECK, T. *Lean software development: an agile toolkit*. [S.l.]: Addison-Wesley, 2003. Citado na página 28.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software*. 9. ed. AMGH Editora Ltda., 2021. ISBN 978-1-259-87297-6. Disponível em: <[https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/epubcfi/6/52\[%3Bvnd.vst.idref%3DC15.xhtml\]/4\[PRESSMAN\\_Completo-22\]/2/40\[sigil\\_toc\\_id\\_283\]/2/1:6\[twa%2Cre\]](https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/epubcfi/6/52[%3Bvnd.vst.idref%3DC15.xhtml]/4[PRESSMAN_Completo-22]/2/40[sigil_toc_id_283]/2/1:6[twa%2Cre])>. Acesso em: 2022-12-03. Citado 7 vezes nas páginas 19, 22, 23, 24, 25, 33 e 34.

RAHMAN, A. A.; SAHIBUDDIN, S.; IBRAHIM, S. A study of process improvement best practices. In: *ICIMU 2011 : Proceedings of the 5th international Conference on Information Technology & Multimedia*. [S.l.: s.n.], 2011. p. 1–5. Citado na página 17.

- REINEHR, S. *Engenharia de Requisitos*. GRUPO A EDUCAÇÃO S.A., 2020. ISBN 978-65-5690-067-4. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/pageid/12>>. Acesso em: 2023-01-15. Citado 3 vezes nas páginas 19, 20 e 21.
- ROBASKI, J. R. *FDD (Feature Driven Development)*. 2014. Disponível em: <<https://medium.com/@jrobaski/fdd-feature-driven-development-7d08c5c24c8f>>. Acesso em: 2023-01-15. Citado 2 vezes nas páginas 27 e 28.
- SALGUEIRO, L. F.; ARTE, M. V. D. *MODELOS DE QUALIDADE DE SOFTWARE*. TCC — Escola de Informática Aplicada - Curso de bacharelado em sistemas de informação - Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2005. Acesso em: 2023-01-11. Citado na página 49.
- SAMPIERI, R. H.; COLLADO, C. F.; LUCIO, M. d. P. B. *Metodologia de Pesquisa*. Penso Editora LTDA, 2013. ISBN 978-85-65848-36-7. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788565848367/pageid/430>>. Acesso em: 2023-01-18. Citado na página 57.
- SAWYER, P.; SOMMERVILLE, I.; VILLER, S. Improving the requirements process. In: CITESEER. *REFSQ*. [S.l.], 1998. p. 71–84. Citado na página 17.
- SAWYER, P.; SOMMERVILLE, I.; VILLER, S. Capturing the benefits of requirements engineering. v. 16, n. 2, p. 78–85, 1999. ISSN 1937-4194. Conference Name: IEEE Software. Citado na página 17.
- SCHÖN, T. Escalona, 2017 schön em, thomaschewski j., escalona mj. *Agile requirements engineering: a systematic literature review*, *Comput. Stand. Interf.*, v. 49, p. 79–91, 2017. Citado na página 25.
- SEI. *Portuguese language translation of CMMI for Development, V1.2*. SEI, 2006. Disponível em: <<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=28943>>. Acesso em: 2023-01-05. Citado na página 36.
- SEI. *CMMI for Development, Version 1.3*. 2010. Disponível em: <[https://resources.sei.cmu.edu/asset\\_files/technicalreport/2010\\_005\\_001\\_15287.pdf](https://resources.sei.cmu.edu/asset_files/technicalreport/2010_005_001_15287.pdf)>. Acesso em: 2023-01-29. Citado 4 vezes nas páginas 30, 35, 37 e 38.
- SOFTEX. *Guia Geral MPS de Software (2021) – Softex*. Sociedade SOFTEX, 2021. ISBN 978-85-99334-48-5. Disponível em: <<https://softex.br/download/guia-geral-de-software-2021/>>. Acesso em: 2022-12-03. Citado 5 vezes nas páginas 16, 38, 39, 40 e 41.
- SOMMERVILLE, I. *Engenharia de Software, 10ª edição*. 10. ed. Pearson Education do Brasil LTDA, 2019. ISBN 978-85-430-2497-4. Disponível em: <<https://www.bibliotecavirtual.com.br/>>. Acesso em: 2023-01-03. Citado 11 vezes nas páginas 16, 19, 20, 21, 23, 24, 33, 34, 35, 52 e 53.
- SOMMERVILLE, I.; RANSOM, J. An empirical study of industrial requirements engineering process assessment and improvement. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, ACM New York, NY, USA, v. 14, n. 1, p. 85–117, 2005. Citado na página 17.

TELES, V. M. *Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade*. [S.l.]: Novatec Editora, 2017. Citado na página 22.

VASCONCELOS, A. M. L. d.; OLIVEIRA, S. R. B. *Qualidade, Gestão e Processos de Software*. [S.l.]: UFPE, 2016. ISBN 978-85-415-0733-2 (online). Acesso em: 2023-01-02. Citado 3 vezes nas páginas 29, 30 e 49.

YACOUB, M. K.; MOSTAFA, M. A. A.; FARID, A. B. A new approach for distributed software engineering teams based on kanban method for reducing dependency. *J. Softw.*, v. 11, n. 12, p. 1231–1241, 2016. Citado na página 28.