



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Aplicação de Gestão de Riscos no processo de Elicitação de Requisitos em uma empresa de Tecnologia de Informação

Fernanda Lopes Lima

Dissertação apresentada como requisito de Defesa para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador

Prof^ª. Dra. Simone Borges Simão Monteiro

Brasília
2026

Dedicatória

Gostaria de expressar minha profunda gratidão à Prof.a Dra. Simone Borges Simão Monteiro por sua liderança, apoio, paciência, compreensão e incentivo ao longo deste processo de pesquisa. Sua crença em mim foi fundamental, especialmente nos momentos em que eu mesma duvidei.

Minha sincera gratidão à Prof.a Dra. Viviane Vasconcellos Ferreira Grubisic pelo auxílio e parceria ao longo de todo o processo de evolução desta proposta de pesquisa.

Agradeço à Prof.a Dra. Edna Dias Canedo por sua valiosa contribuição para a evolução desta pesquisa, demonstrada por sua disponibilidade e pelos seus trabalhos publicados.

Ao Prof. Dr. Marcelo Ladeira, agradeço pelo incentivo e pela oportunidade de conhecer o Programa de Pós-graduação em Computação Aplicada. Sua compreensão e apoio nos momentos nos meus momentos de dificuldades foram essências para que eu chegasse até aqui.

À Universidade de Brasília, minha gratidão por proporcionar uma oportunidade tão valiosa de realizar um curso de altíssima qualidade e reconhecimento mundial.

Agradeço a todos os professores do PPCA por dedicarem seu tempo à disseminação do conhecimento e ao crescimento intelectual e profissional dos alunos. Todo incentivo e crença no meu sonho de me tornar mestre e na minha proposta de pesquisa foram fundamentais.

À minha família, especialmente ao meu filho e ao meu neto, por serem a minha maior motivação para não desistir de lutar é me impulsionam diariamente na realização dos meus sonhos.

Aos meus amigos, poucos mas preciosos, agradeço imensamente. Em especial, ao meu amigo Me. Rodrigo Gomes, por sempre me apoiar e compartilhar comigo os desafios na conclusão deste sonho.

Finalmente, à minha avó Santurina (in memoriam), que foi e sempre será meu maior exemplo de amor, compaixão, persistência e fé, minha eterna gratidão.

Agradecimentos

Primeiramente, agradeço a Deus por Sua infinita misericórdia e graça derramadas sobre minha vida.

Aos meus pais, Maria das Graças e José Oliveira, minha eterna admiração. Mesmo sem terem tido a oportunidade de estudar, educaram a mim e aos meus irmãos com sabedoria, valores sólidos e um amor extraordinário.

À minha irmã Fabiana, por ser minha fonte de serenidade e presença constante; e ao meu irmão Pedro, por sempre me impulsionar na busca pelos meus sonhos — minha sincera gratidão.

Ao meu noivo, Tayrone Oliveira, agradeço por sua paciência, apoio e compreensão diante das minhas ausências. Ao meu filho, Gustavo Lima, por ser meu maior incentivo a continuar, mesmo nos momentos mais difíceis. Ao meu neto, Ravi Lima, nosso presente divino, e ao meu querido sobrinho, Pedro Lopes, por ser parte da minha vida com todo seu carinho, gentileza e doçura me enchem de orgulho, deixo registrado meu amor e reconhecimento.

Resumo

A elicitação de requisitos é uma etapa crucial para o sucesso do desenvolvimento de um software. Com isso, este estudo tem por objetivo aplicar a Gestão de Riscos ao processo de Elicitação de Requisitos a partir da concepção de um protótipo de sistema para uma empresa de Tecnologia de Informação. A proposta da elaboração do protótipo de sistema automatizado destina-se a otimizar o procedimento de elicitação de requisitos utilizando métodos ágeis. Para alcançar este objetivo, a pesquisa visa realizar a modelagem do processo atual (AS-IS) de elicitação de requisitos de software, identificar os riscos associados a este processo, redesenhar um processo otimizado (TO-BE) como resposta às ameaças identificadas e desenvolver um protótipo de ferramenta computacional voltada para a gestão de riscos no âmbito do processo de elicitação de requisitos.

O caráter dessa pesquisa é aplicada, com uma abordagem qualitativa de natureza exploratória, utilizando a estratégia de estudo de caso de uma empresa de tecnologia denominada Liberty Tecnologia.

Esta pesquisa foi fundamental para a Liberty Tecnologia, oferecendo um suporte robusto por meio de um estudo aprofundado do processo de elicitação de requisitos da empresa. A análise detalhada identificou, examinou e avaliou os riscos inerentes, e criou uma proposta de ferramenta computacional especializada a partir da elaboração do protótipo de sistema. A validação evidenciou que o processo estruturado de elicitação de requisitos, apoiado pelo protótipo UPLearn, é viável e aderente ao contexto organizacional analisado, apresentando potencial para reduzir riscos recorrentes na definição de requisitos. O plano de ação executado confirmou a aplicabilidade da proposta e demonstrou que a integração entre práticas estruturadas, abordagem ágil e suporte ferramental contribui para aprimorar a comunicação, a organização das informações e a tomada de decisão. Os resultados indicam que o protótipo materializa a proposta teórica e constitui um instrumento promissor para conduzir a elicitação de requisitos de forma mais controlada, colaborativa e alinhada aos objetivos do negócio, ainda que limitado ao contexto do estudo e passível de evolução futura.

Palavras-chave: Gestão de riscos, Elicitação de requisitos, Métodos Ágeis, Desenvolvimento de Software

Abstract

Requirements elicitation is a crucial stage for the success of software development. In this context, this study aims to apply Risk Management to the Requirements Elicitation process through the design of a system prototype for an Information Technology company. The proposal to develop an automated system prototype is intended to optimize the requirements elicitation procedure using agile methods. To achieve this objective, the research seeks to model the current requirements elicitation process (AS-IS), identify the risks associated with this process, redesign an optimized process (TO-BE) in response to the identified threats, and develop a prototype of a computational tool focused on risk management within the scope of the requirements elicitation process. This research is applied in nature, with a qualitative and exploratory approach, using a case study strategy conducted in a technology company named Liberty Tecnologia. This study was fundamental to Liberty Tecnologia, providing robust support through an in-depth analysis of the company's requirements elicitation process. The detailed assessment identified, examined, and evaluated inherent risks and led to the proposal of a specialized computational tool through the development of a system prototype. The validation demonstrated that the structured requirements elicitation process, supported by the UPLearn prototype, is feasible and aligned with the analyzed organizational context, showing potential to reduce recurring risks in requirements definition. The executed action plan confirmed the applicability of the proposal and demonstrated that the integration of structured practices, an agile approach, and tool support contributes to improved communication, information organization, and decision-making. The results indicate that the prototype materializes the theoretical proposal and constitutes a promising instrument to support requirements elicitation in a more controlled, collaborative, and business-aligned manner, although limited to the study context and subject to future evolution.

Keywords: Risk Management, Requirements Elicitation, Agile Methods, Software development

Sumário

1	Introdução	2
1.1	Contextualização	2
1.2	Justificativa	5
1.3	Objetivos	7
1.3.1	Objetivo Geral	7
1.3.2	Objetivos Específicos	7
1.4	Estrutura do Trabalho	7
2	TEMAC	9
2.1	Revisão do Estado da Arte por Meio do Enfoque Meta Analítico Consolidado	9
2.2	Descrição do Método	9
2.3	Preparação da Pesquisa	10
2.3.1	Base de dados utilizada	10
2.3.2	Termos de pesquisa	10
2.3.3	Consolidação e tratamento dos dados	11
2.4	Apresentação dos dados inter-relacionais	12
2.4.1	Palavras-chave	12
2.4.2	Evolução do tema ano a ano	12
2.4.3	Autores e artigos mais citados	13
2.4.4	Países e linhas de pesquisa	14
2.5	Detalhamento, modelo integrador e validação por evidências	14
2.5.1	Co-ocurrence	14
2.5.2	Co-citação	15
2.5.3	Coupling	16
2.6	Determinação do Enfoque Teórico da Pesquisa	22
3	Referencial Teórico	24
3.1	Cadeia de Valor	24
3.2	Gestão de Processos	26

3.3	Gestão de Riscos na fase de elicitação de requisitos	29
3.4	Desenvolvimento de <i>Software</i>	38
3.5	Engenharia de Requisitos	40
3.6	Métodos de elicitação de Requisitos	83
3.6.1	Técnicas de elicitação de requisitos a partir da modelagem de proces- sos de negócio	84
3.6.2	Metodologia REMO	86
3.6.3	Heurísticas	87
3.6.4	Elicitação de requisitos utilizando método tradicional	90
3.6.5	Elicitação de requisitos no contexto das metodologias ágeis	95
3.6.6	Técnicas e Métodos de elicitação de requisitos existentes	99
3.6.7	Framework SCRUM	102
3.7	Ferramentas para gestão de requisitos	109
4	Método de Pesquisa	112
4.1	Método de Pesquisa	112
4.1.1	Local do estudo	119
4.1.2	Objeto de estudo	120
4.1.3	Instrumento de coleta de dados	120
5	Resultados e Análises	126
5.1	Estabelecimento do contexto	126
5.1.1	A organização	126
5.1.2	Análise dos projetos finalizados	130
5.1.3	Modelagem do processo de Elicitação de requisitos atual (AS-IS) . . .	130
5.2	Aplicação de gestão de riscos ao processo de elicitação de requisitos	133
5.2.1	Identificação dos riscos	134
5.2.2	Análise dos riscos	137
5.2.3	Avaliação dos riscos	141
5.3	Elaboração do processo (TO-BE) para a fase de elicitação de requisitos . . .	147
5.4	Elaboração de Plano de Ação	149
5.5	Desenvolvimento do Protótipo Computacional UpLean	151
5.5.1	Simulação de utilização do protótipo	153
5.5.2	Feedback, Revisão e Ajustes no Protótipo da Solução	156
5.5.3	Validação da Proposta de Solução	157
6	Conclusão	158
	Referências	160

Apêndice	167
A Pesquisa	168
A.1 TEMAC	168
A.2 Formulário	175
A.3 Levantamento dos Requisitos da Proposta de Solução	180
A.4 Telas do Protótipo - UPLearn	203

Lista de Figuras

3.1	Processo de Gestão de Riscos detalhado (ISO 31000:2018)	31
3.2	Decisões na Avaliação de Riscos (ISO 31000:2018)	34
3.3	Matriz de Risco (Elaborada pela autora)	35
3.4	Etapas de Identificação dos Riscos (ISO 31000:2018)	37
3.5	Engenharia dos Requisitos	47
3.6	Exemplo de descrições de Requisitos Funcionais	51
3.7	Categorias RNF	55
3.8	Tipos de requisitos de Domínio	57
3.9	Exemplos de Requisitos de Manutenibilidade	71
3.10	Exemplos de Requisitos de Suportabilidade	72
3.11	Exemplos de Requisitos de Leis e Regulamentos	74
3.12	Técnicas de Levantamento de Requisitos	101
3.13	Pilares do SCRUM	
	Fonte: Guia Scrum [1].	103
3.14	Processo SCRUM [1].	
	Fonte: Scrum.org	106
4.1	Método de pesquisa	113
4.2	Estrutura da pesquisa	116
4.3	Ferramenta 5W2H	124
5.1	Cadeia de Valor	127
5.2	Análise SWOT	128
5.3	Problemas	131
5.4	Principais Problemas	132
5.5	Identificação dos Riscos	135
5.6	Relação Probabilidade e Impacto na Matriz de Riscos	137
5.7	Matriz de Probabilidade e Impacto	139
5.8	Níveis de Probabilidade	139
5.9	Critérios de Impacto	140

5.10	CrITÉrios de Impacto	141
5.11	Risco - AlteraÇões frequentes nos requisitos	145
5.12	Risco - AusÊncia de EstruturaÇão no Processo de ElicitaÇão de Requisitos .	145
5.13	Risco - InexistÊncia de ferramentas adequadas	146
5.14	Risco - Metodologia ágil nÃo definida	146
5.15	Mapeamento TO-BE	147
5.16	Plano de AÇão	150
A.1	Etapas - TEMAC	168
A.2	Termos mais comuns	168
A.3	CombinaÇão dos termos de pesquisa utilizados no TEMAC	169
A.4	Áreas de Pesquisa	169
A.5	Tipo de Documento	170
A.6	Foco de pesquisa das principais áreas	170
A.7	AssociaÇão de palavras-chave	171
A.8	Quantidade de publicaÇões por ano	171
A.9	Quantidade de citaÇões por ano	172
A.10	Autores dos artigos mais citados	172
A.11	Artigos mais citados	173
A.12	Países com maior número de citaÇões	173
A.13	Rede Bibliométrica de termos mais recorrentes	174
A.14	Rede de Co-citation	174
A.15	Rede de Coupling	175
A.16	Parte I	175
A.17	Parte II	176
A.18	Parte III	176
A.19	Resultado Formulário	177
A.20	Resultado Formulário	178
A.21	Resultado Formulário	179
A.22	Resultado Formulário	180
A.23	Tela de Cadastrar Novo Usuário	203
A.24	Tela de Login do Sistema	203
A.25	Tela de AprovaÇão Pendente	204
A.26	Tela de Gerenciamento de Usuário Ativo	204
A.27	Tela de Criar Nova Iniciativa	204
A.28	Tela Listar Iniciativa	204
A.29	Tela Visualizar Detalhe da Iniciativa	205
A.30	Tela Convidar Participante	205

A.31	Tela Alterar Status da Iniciativa	205
A.32	Tela Visão do Produto	206
A.33	Tela Definir Visão do Produto	206
A.34	Tela Definir Metas e Objetivos	206
A.35	Tela Criar Personas	207
A.36	Tela Mapear Jornada	207
A.37	Tela Mapear Jornada	207
A.38	Tela Adicionar Etapa	208
A.39	Tela Brainstorm de Feature	208
A.40	Tela Avaliação 360	208
A.41	Tela Avaliação Salva	209
A.42	Tela Criar Épico	209
A.43	Tela Criar Épico	209
A.44	Tela Vincular Feature a Epico	210
A.45	Tela Criar User Stories	210
A.46	Tela Definir MVP	210
A.47	Tela Priorizar Feature para MVP	211
A.48	Tela Visualizar Dashboard Consolidado	211
A.49	Tela Exportar Discovery	211
A.50	Tela Exportar para o Jira	212
A.51	Tela Dashboard Administrador	212
A.52	Tela Indicador Progresso	212
A.53	Tela Consultar Auditoria de Acessos	212

Lista de Abreviaturas e Siglas

ANSI American National Standards Institute.

CMM Capability Maturity Model® Integration.

EAG Elicitação em Métodos Ágeis.

EPC Event-driven Process Chain.

IBM Business Machines Corporation.

ISO International Organization for Standardization.

MVP Minimum Viable Product.

OMG Object Management Group.

PBB Product Backlog Building.

RE Engenharia de requisitos.

SWOT Strengths, Weaknesses, Opportunities, Threats.

TEMAC Teoria do ENfoque Metanalítico COnsolidado.

TI Tecnologia da Informação.

UML Unified Modeling Language.

US User story.

WoS Web of Science.

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

La Lopes Lima, Fernanda
Aplicação de Gestão de Riscos no processo de Elicitação
de Requisitos em uma empresa de Tecnologia de Informação /
Fernanda Lopes Lima; orientador Simone Borges Simão
Monteiro. Brasília, 2026.
211 p.

Dissertação(Mestrado Profissional em Computação Aplicada)
Universidade de Brasília, 2026.

1. Gestão de riscos. 2. Elicitação de requisitos. 3.
Métodos Ágeis. 4. Desenvolvimento de Software. I. Borges
Simão Monteiro, Simone, orient. II. Título.

Capítulo 1

Introdução

Neste capítulo são apresentados a contextualização do tema central pesquisado, a definição do problema de pesquisa, a justificativa, o objetivo geral, os objetivos específicos, as contribuições esperadas e a estruturação do trabalho.

1.1 Contextualização

A tecnologia está profundamente integrada ao nosso cotidiano, com o software desempenhando um papel crucial em atividades pessoais e profissionais. Quando ocorrem falhas no software, os impactos podem ser diversos, como semáforos descoordenados, voos atrasados ou cancelados, dificuldades em realizar compras online ou acessar serviços bancários, pacientes sem atendimento clínicos e emergenciais, entre outros problemas. Em casos extremos, essas falhas podem até mesmo resultar em perdas de vidas. Uma causa comum desses problemas é a gestão inadequada de requisitos, incluindo a compreensão, especificação e gerenciamento inadequados, o que compromete o desempenho do software e é uma das principais razões para o fracasso de projetos de TI.

Para abordar essa questão, este estudo inicia a discussão sobre de levantamento de requisitos. Conforme descrito pelo Instituto de Engenheiros Eletricistas e Eletrônicos, os requisitos podem se definidos como [2] :

- Condições ou capacidades necessárias para que usuários solucionem problemas ou atinjam objetivos.
- Condições ou capacidades que sistemas ou componentes devem satisfazer para cumprir contratos, padrões, especificações ou outros documentos formais.
- Documentação dessas condições ou capacidades, conforme descrito nos itens 1 e 2.

As primeiras empresas de prestação de serviços na área de desenvolvimento de software surgiram nas primeiras décadas após a invenção do computador digital que foi por volta de 1910 [3]. No entanto, houve uma rápida evolução da indústria e ao surgimento simultâneo de várias empresas em diferentes partes do mundo. Algumas das principais empresas fundadas na época, foram: IBM e DEC.

A IBM (*International Business Machines*) - Fundada em 1911 a CRT - *Computing-Tabulating-Recording Co.* e renomeada para Business Machines Corporation (IBM) em 1924, foi uma das primeiras grandes empresas a oferecer soluções em software, embora inicialmente estivessem mais focada em *hardware*. Ao longo dos anos, a IBM expandiu seus serviços para incluir o desenvolvimento de software[3].

Na década de 1960, o *software* começou a ser reconhecido como um produto distinto, ou seja, um produto que poderia ser comercializado independente da aquisição do *hardware*. Antes disso, o *software* era frequentemente fornecido junto com o *hardware* sem custo adicional. No final dos anos 1960 e início dos anos 1970, começaram a surgir empresas especializadas em *software*, oferecendo serviços de desenvolvimento, consultoria e manutenção. Isso foi um desvio da norma anterior, onde empresas de *hardware*, como IBM e DEC, dominavam o cenário do *software*.

A DEC (*Digital Equipment Corporation*) é lembrada como uma das empresas mais influentes na história inicial da computação. Sua abordagem inovadora para a fabricação de mini-computadores e o desenvolvimento de software teve um impacto duradouro na indústria de tecnologia da informação, moldando o modo como o *software* é desenvolvido [4].

Na década de 1970 houve um aumento exponencial na indústria de *software* com o surgimento de muitas empresas independentes de *software*, à medida que o mercado reconheceu a utilidade e o valor do *software* como um produto e serviço separado [4].

No início dos anos 1990, o *Software Engineering Institute* introduziu o Capability Maturity Model® Integration (CMM) *Capability Maturity Model*, um padrão para medir a maturidade dos processos de desenvolvimento de *software* das empresas. Isso foi crucial para padronizar e melhorar a qualidade dos serviços de desenvolvimento de *software*. O que nos permite observar que a década de 1970 foi um período crítico de diversificação e especialização na indústria de *software* [3].

Nos últimos anos, a Tecnologia da Informação (TI) tornou-se imprescindível para empresas públicas e privadas. A sociedade encontra-se cada vez mais dependente da indústria de *software*, pois as empresas de Tecnologia da Informação desempenham um papel fundamental na manutenção, evolução, suporte de projetos e desenvolvimento de soluções destinadas a atender às necessidades de seus setores específicos de negócios. Para alcançar um nível de qualidade desejado em um prazo reduzido, esses projetos dependem

de processos, ferramentas e metodologias de desenvolvimento que orientam as equipes na execução de tarefas, com o objetivo de melhorar a entrega de serviços [5].

A maneira tradicional de desenvolver *softwares* segue o padrão genérico da engenharia de requisitos, projetos, construção e manutenção, também conhecida como metodologia cascata, *plan-driven ou Waterfall*. Por tratar-se de um método tradicional guiado por documentação robusta e com grande morosidade para que o *software* comece a ser desenvolvido, há inúmeras mudanças nos requisitos, seja por necessidade da área atendida, por leis, normativos e/ou regulamentos, além de mudanças arquiteturais, tecnologias e ambientes de negócio. Uma das principais limitações do modelo *Waterfall* é sua rigidez frente a mudanças, que frequentemente determinam o sucesso ou fracasso de um produto de *software*. Reconhecer essa lacuna fez com que o Manifesto Ágil surgisse por volta do ano 2000, trouxe consigo princípios e métodos que enfatizam a valorização das pessoas, entregas contínuas e antecipadas de *software*, além da flexibilidade para aceitar alterações nos requisitos. Assim, a agilidade passou a desempenhar um papel crucial na otimização da gestão de processos, organizando-os em etapas que maximizam a eficiência do tempo e dos recursos disponíveis. Os métodos ágeis surgiram em resposta às debilidades tangíveis e percebidas da engenharia de software tradicional", destacando-se por oferecer benefícios significativos [6].

Diante desse cenário, a engenharia de *software* evoluiu consideravelmente ao longo das últimas décadas, refletindo a crescente complexidade e escopo dos sistemas de *software* que permeiam quase todos os aspectos da vida moderna. O sucesso de qualquer projeto de *software* depende consideravelmente da fase de elicitação de requisitos, onde as necessidades e expectativas dos *stakeholders* são identificadas e documentadas.

Segundo pesquisa realizada pela *Standish Group*, cerca de 43% dos projetos de desenvolvimentos tecnológicos tendem a ter insucesso. Esses estudos têm demonstrado que a má definição dos requisitos está entre as principais fontes de problemas, constituindo 32,65%. Esse aspecto tem sido negligenciado na engenharia de requisitos e geralmente é considerado um risco potencial que pode afetar os projetos de forma negativa [7].

A fase de elicitação de requisitos é crítica em qualquer projeto de desenvolvimento de *software*. Erros ou ambiguidades nessa fase podem levar a problemas significativos tardio no ciclo de vida do projeto.

Com isso, a abordagem ágil de gerenciamento de projetos surgiu como alternativa para as fragilidades identificadas no método tradicional de desenvolvimento. Essa abordagem abrange um conjunto de modelos e técnicas com o objetivo de lidar com o risco inerente a ambientes dinâmicos por meio de atividades interativas e entregas constantes, tanto para atividades de desenvolvimento de software quanto para atividades relacionadas ao gerenciamento de projetos [6].

Diante desse cenário, nos últimos anos percebeu-se um aumento no interesse das empresas de Tecnologia da Informação (TI) na utilização de metodologias ágeis, tendo como principal objetivo minimizar os problemas encontrados constantemente no desenvolvimento de *software*, tais como: Identificação das necessidades do cliente, mudanças de requisitos que impactam diretamente no escopo e conseqüentemente no prazo e custo do projeto, além do não atendimento às expectativas do cliente. Sabe-se também que métodos ágeis são inovadores e iterativos que suportam mudanças e oferecem melhorias em vários aspectos dos projetos de desenvolvimento de sistemas e aplicações [8].

Para tanto, faz-se necessário realizar um estudo que compreenda as necessidades da empresa em relação à melhoria da organização na área de desenvolvimento de *software*. Isso inclui a utilização de ferramentas e processos adequados, bem como a gestão eficaz de riscos na fase de elicitação de requisitos, mitigando assim os problemas que resultam em maiores tempos de entrega, custos elevados e, conseqüentemente, insatisfação dos clientes e dos profissionais da empresa.

1.2 Justificativa

O desenvolvimento de *software* é uma disciplina complexa e dinâmica, envolvendo uma série de processos, desde a concepção da ideia até a entrega e manutenção do produto final. Para gerenciar efetivamente esses processos e garantir a qualidade e eficiência, as empresas necessitam realizar mapeamento, implantação de processos e adotar ferramentas para controlar as demandas, além de implantar metodologias de desenvolvimento de software [9].

A Liberty Tecnologia da Informação, que é o foco deste estudo, vem enfrentando uma série de desafios que comprometem a qualidade, eficiência e eficácia de suas entregas de *software*. A falta de processos estruturados na área de desenvolvimento, bem como a inexistência de ferramentas adequadas para o controle de demandas e a ausência de uma metodologia ágil bem definida, têm causado inconsistências no levantamento de requisitos. Essa carência em estabelecer uma abordagem ágil sólida e processos estruturados para gerenciar riscos durante a elicitação de requisitos tornou-se um ambiente propício para complicações. Tais desafios têm gerado reclamações, penalizações e multas por parte das empresa contratantes. Além das melhorias urgentes necessárias na fase de elicitação de requisitos, a empresa enfrenta desafios relacionados à falta de gestão da demanda e à organização adequada de seus profissionais em equipes.

A consequência imediata desses desafios é a entrega de sistemas que não satisfazem as necessidades e expectativas dos clientes. Além disso, observa-se um declínio no comprometimento e engajamento tanto dos profissionais quanto dos *stakeholders*, situação

que coloca em risco a continuidade e a expansão da empresa no competitivo setor de TI. A falta de processos bem definidos, gestão de demandas e formação de times *full time* para os projetos ampliam os riscos na fase de elicitação de requisitos. Sem uma abordagem estruturada para identificar, avaliar e mitigar riscos e os instrumentos necessários para gerenciar demandas e coordenar as equipes a empresa se posiciona em um cenário de vulnerabilidade, onde a incerteza e a instabilidade de engajamento dos profissionais e entregas dos projetos, torna-se comum, em vez de serem situações isoladas.

Uma das lacunas críticas na área de Tecnologia da empresa está na ausência de um processo coerente que atenda a fase de elicitação de requisitos. Atualmente, as solicitações de clientes são recebidas por e-mails, ferramentas de chat e, muitas vezes, são direcionadas especificamente para indivíduos ou departamentos específicos. Não há processo estruturado para a fase de elicitação de requisitos nem utilização de técnicas de mercado para realização do levantamento de requisitos. A área técnica, que é crucial para esse processo de refinamento dos requisitos, não está envolvida na fase de elicitação nem na validação dos requisitos. Isso resulta em *User story* - User story (US) também conhecidas popularmente como estória de usuário, incompletas e que são diretamente encaminhadas para desenvolvimento sem o devido refinamento técnico. Ainda na fase de elicitação de requisitos há uma ausência de análise do impacto, identificação de dependências, rastreabilidade das funcionalidades que precisam ser desenvolvidas além de técnicas para utilização de uma abordagem estruturada afim de responder às perguntas essenciais associadas a fase de elicitação de requisitos utilizando métodos ágeis:

- Como estabelecer um backlog de produto eficaz?
- Como garantir a construção de um produto valioso?
- Como identificar e atender as necessidades do cliente?
- Como determinar as prioridades do cliente desde o início?
- Qual é o processo para refinar itens do backlog?

Não existe definição de papéis e responsabilidades entre os profissionais da equipe. Existe sim, uma ambiguidade na definição de papéis dentro das equipes. Embora o analista seja responsável por manter o backlog de produto atualizado, não há uma clareza das responsabilidades entre esses profissionais.

O estudo contempla a análise detalhada das problemáticas identificadas e a proposição de soluções fundamentadas em práticas de mercado relacionadas à gestão de riscos, com ênfase na fase de elicitação de requisitos.

1.3 Objetivos

1.3.1 Objetivo Geral

Este projeto tem como objetivo propor melhoria ao processo de elicitação de requisitos na organização que atua no ramo de Tecnologia de Informação, a partir da abordagem de gestão de riscos.

1.3.2 Objetivos Específicos

Para atingimento do objetivo geral, destacam-se os seguintes objetivos específicos:

- OE01: Descrever o processo de elicitação de requisitos da empresa Liberty Tecnologia;
- OE02: Aplicar o processo de avaliação de riscos ao processo de elicitação de requisitos utilizando o caso selecionado;
- OE03: Elaborar um plano de ação para melhoria do processo de elicitação de requisitos baseado nas necessidades e nas boas práticas;
- OE04: Criar um protótipo de sistema automatizado para operacionalização do plano de ação de melhoria no processo estudado;
- OE05: Validar proposta sistêmica e melhoria do processo de elicitação de requisitos.

1.4 Estrutura do Trabalho

O presente trabalho é dividido em seis capítulos, conforme detalhado a seguir:

- No Capítulo 1, é contextualizado e justificado o problema de pesquisa, estabelecendo o objetivo principal e os objetivos específicos desejados, bem como as contribuições esperadas. Este capítulo também esboça a estrutura do estudo.
- O Capítulo 2 dedica-se a uma revisão sistemática da literatura, visando direcionar adequadamente a pesquisa e assegurar a inclusão de trabalhos relevantes. A Teoria do Enfoque Meta Analítico Consolidado (TEMAC) é empregada como base.
- No Capítulo 3, discorre-se sobre a fundamentação teórica relacionada ao foco da pesquisa.
- O Capítulo 4 detalha a metodologia adotada, descrevendo o ambiente e recursos empregados.

- O Capítulo 5 apresenta o desenvolvimento de um protótipo de sistema, desenhado para ilustrar o plano de ação proposto, que busca alinhar práticas recomendadas de gestão de riscos com a aplicação de métodos ágeis na etapa de elicitação de requisitos.
- No Capítulo 6, é realizada a validação da proposta sistêmica e ajustada conforme as melhorias sugeridas, aplicando-as a um projeto real.

Capítulo 2

TEMAC

2.1 Revisão do Estado da Arte por Meio do Enfoque Meta Analítico Consolidado

Todo trabalho científico, não sendo considerada a sua temática ou área de conhecimento, para ser aplicável e de alto impacto, deve-se levar em conta o conhecimento já existente sobre o assunto trabalhado. Antes de iniciar uma pesquisa específica, o pesquisador deve buscar nos meios confiáveis o que já se sabe sobre o fenômeno estudado para produzir um estudo que agregue conhecimentos extras à temática. Pesquisar é trabalhar com dados, saber onde obtê-los e como tratá-los para possuir informações novas. Para atender a este objetivo foi utilizando a Teoria do Enfoque Meta Analítico Consolidado – Teoria do ENfoque Metanalítico COnsolidado (TEMAC). O TEMAC é uma abordagem útil para pesquisadores que desejam realizar uma revisão de literatura de forma sistemática e abrangente. A abordagem integrativa adotada permite que os pesquisadores identifiquem os principais pontos de diferentes estudos, e que sintetizem esses resultados de forma clara e objetiva. O TEMAC possui método significativamente estruturado e robusto, ponderando os esforços de sua aplicação aos benefícios alcançados ao final do processo. Os itens a seguir descrevem de forma sintética as características do método, bem como os resultados específicos de sua aplicação para o contexto estudado [10].

2.2 Descrição do Método

o TEMAC atende aos princípios do Enfoque Meta Analítico, que utiliza os critérios de impacto de revistas, citações de autores e artigos e frequência de palavras-chaves, mas com o benefício de integrar ao processo ferramentas tecnológicas de acesso gratuito, que facilitam as análises e reduzem o trabalho manual [10].

A análise do TEMAC é organizada em etapas e possui o respaldo das teorias bibliométricas em seus princípios. Para atingir o resultado esperado, que é a definição das referências mais relevantes para o contexto da pesquisa, tendo em vista o conhecimento prévio do que já foi publicado sobre o tema, o TEMAC possui três principais etapas, conforme apresentado na figura A.1.

- (a) Preparação da pesquisa: definição de palavras-chaves que estão relacionadas ao tema pesquisado, o período (espaço-tempo) da análise, quais bases de dados utilizadas e as áreas de conhecimento que serão consideradas.
- (b) Apresentação e inter-relação dos dados: consiste em relacionar várias fontes de pesquisa, a critério do pesquisador, como a evolução do tema de forma anual, os autores mais citados, periódicos que mais publicam, entre outros.
- (c) Detalhamento, modelo integrador e validação por evidências: nesta etapa são identificados os principais autores, abordagens e linhas de pesquisa referentes ao tema, utilizando técnicas de co-citação e acoplamento [10].

2.3 Preparação da Pesquisa

2.3.1 Base de dados utilizada

Foram utilizadas as plataformas Web of Science (WoS) e Scopus como referência para este estudo em função da excelência operacional, existência de plataforma própria de análise, que facilita a consolidação e extração dos dados, e da disponibilidade temporal a partir de 1900, garantindo maior cobertura do tema pesquisado. Os dados foram coletados em 23 de agosto de 2025. Embora, as bases de dados dessas plataformas contenham apenas publicações em inglês, esse fato não foi considerado uma limitação para esta pesquisa, já que o processo natural é que trabalhos de elevada qualidade técnica sejam apresentados neste idioma, como forma de potencializar seu alcance a pesquisadores em abrangência mundial.

2.3.2 Termos de pesquisa

Tendo em vista as pesquisas realizadas no estudo em questão, onde o risco na fase de elicitação de requisitos pode ser abordado de diferentes formas, para determinar as palavras-chave da pesquisa foram retirados os termos mais relevantes relacionado ao tema que encontra-se reproduzido integralmente no Apêndice A.1.

Com o objetivo de identificar as principais referências compatíveis com o objetivo dessa pesquisa, foram realizadas diversas combinações desses termos, resultando nas palavras-chave utilizadas. As palavras-chave serão apresentadas tal qual foram utilizadas nas buscas, considerando facilitadores de pesquisa, como a utilização de aspas duplas para termos compostos, conectores *and* e *or*, conforme regra de utilização da WoS e o uso do asterisco para indicar que daquele ponto em diante a palavra pode apresentar qualquer letra e em qualquer quantidade, evitando a perda de variações da palavra causadas por prefixos ou sufixos. Nesta etapa, optou-se por delimitar apenas pelo período de análise (últimos 10 anos). Não foi levado em conta o tipo de documento ou áreas do conhecimento. Este refinamento foi realizado ao final do processo, quando houve a consolidação das bases com os resultados das pesquisas realizadas. Conforme apresentado na figura A.3.

Os 5 resultados encontrados foram acrescentados a listas marcadas pelas ferramentas das plataformas do Web of Science (WoS) e Scopus, que permitem salvar os dados da pesquisa em listas para extrair informações detalhadas posteriormente como autores, palavras chave, anos de publicação, entre outras. Essa lista foi nomeada de Elicitação em Métodos Ágeis (EAG) - Elicitação de Requisitos em Métodos Ágeis.

2.3.3 Consolidação e tratamento dos dados

Após consolidar a base de dados com os resultados das buscas individuais, procedeu-se a um refinamento dessas informações. A seguir estão os ajustes mais significativos.

a) Categoria Web of Science e Scopus

O WoS e o Scopus realizam a classificação dos documentos em categorias relacionadas às áreas de pesquisas. Ao analisar estas informações, foram identificados artigos de diferentes áreas. Os documentos dessas áreas foram excluídos, conforme apresentado na figura A.4.

Após a exclusão dos documentos das áreas de matemática aplicada, economia, robótica e engenharia elétrica, identificados na classificação feita pelo WoS e Scopus, o total de registros relevantes para o estudo reduziu-se para 364. Essa filtragem permitiu um foco mais preciso nos registros mais pertinentes à área de interesse da pesquisa.

b) Tipos de documentos

Foram mantidos na base apenas documentos provenientes de artigos de conferência e artigos em revisão. Artigos de Conferência: São trabalhos científicos apresentados e discutidos em conferências acadêmicas. Geralmente, os artigos de conferência são submetidos a um processo de revisão por pares antes de sua aceitação para apresentação. Eles podem ser publicados nos anais da conferência e posteriormente em revistas científicas, dependendo das políticas da conferência e da qualidade do trabalho. Artigos em Revisão: São artigos submetidos a uma revista científica para revisão por pares, mas que ainda não

passaram pelo processo de revisão e aceitação final. Esses artigos podem estar em diferentes estágios do processo editorial, desde a submissão inicial até a fase final de revisão antes da publicação. Não foram desconsiderados Proceeding, papers e demais tipos de documentos por não possuírem informações relevantes para o estudo em questão. Com este ajuste, o número de registros caiu para 218, sendo sua maioria composta por artigos de conferência conforme apresentado na Figura A.5.

A análise dos termos mais frequentes, entre os artigos pesquisados foram encontrados nas duas maiores áreas de artigos apresentados na pesquisa, são eles: Artigo de conferência e artigo de revisão, onde revelou os principais focos de pesquisa, conforme apresentado na Figura A.6.

Percebe-se que os temas Gestão de riscos, Engenharia de software e desenvolvimento de software são comuns às duas áreas, não havendo domínio específico de nenhuma delas.

2.4 Apresentação dos dados inter-relacionais

Para análise das inter-relações dos dados, além do Microsoft Excel, foi utilizado o software para mapeamento de redes VOSViewer [11].

2.4.1 Palavras-chave

A análise da associação entre as palavras-chaves utilizadas permite avaliar os principais termos que ocorrem juntos, auxiliando na identificação de abordagens diferentes, conforme apresentado na Figura A.7.

As palavras-chave encontradas na rede bibliométrica apresentada na figura A.7 relaciona o termo principal “*Requirements elicitation*”, seguido de dois termos não menos importantes: *Requirements Engineering* e *Software engineering*, assuntos centrais dessa pesquisa, cuja tradução é engenharia de requisitos, engenharia de requisitos e de software.

2.4.2 Evolução do tema ano a ano

A Figura A.8 corrobora a importância da pesquisa atual, demonstrando a tendência crescente do número de publicações relacionadas a elicitação de requisitos e análise de riscos na utilização de métodos ágeis no desenvolvimento de sistemas nos últimos anos. Nos anos de 2014 a 2024, foram realizadas 134 publicações, conforme Figura A.8.

Como podemos analisar os anos que mais tiveram publicações acerca do tema foram os anos: 2019 com 23 publicações, 2021 com 18 publicações e os anos de 2020 e 2024 com 15 publicações.

Com relação ao número de citações, a tendência de crescimento é ainda mais explícita, conforme demonstrado na Figura A.9.

O total de citações já realizadas no ano de 2021 corresponde a 31% do total de citações realizadas nos anos de 2016 a 2020. Importante mencionar que foram encontradas poucas citações acerca do tema no ano de 2023 e 2024, por isso houve uma queda significativa no gráfico apresentado.

2.4.3 Autores e artigos mais citados

Conforme apresentado na Figura A.10 apresenta os autores mais citados na literatura acerca do tema desse estudo.

Conforme apresentado na Figura A.10 os seis primeiros autores mais citados no período de 2014 a 2025 foram:

- Alessio Ferrari
- Paola Spoletini
- Stefania Gnezi
- Edna Dias Canedo
- Didar Zowghi
- Muneera Bano

Para uma melhor compreensão e clareza sobre a natureza dos estudos, os artigos mais citados foram sintetizados e apresentados conforme a Figura A.11.

O objetivo é elucidar as características essenciais de cada pesquisa, possibilitando sua classificação e agrupamento por temas. Após identificar as ocorrências mais relevantes relacionadas ao tema em questão, é essencial fundamentar essa revisão literária, concentrando-se nas principais fontes utilizadas para pesquisa. Em outras palavras, focar nas literaturas mais significativas sobre o tema central deste estudo [10].

Importante salientar que não há muitos trabalhos relacionados a gestão de riscos na fase de elicitação de requisitos disponibilizados na base de dados científica pesquisada. O maior acervo encontrado está vinculado a análise de riscos no processo de desenvolvimento de software. Contudo, as referências citadas na Figura A.11 - Artigos mais citados, possuem informações importantes e que contribuíram significativamente no desenvolvimento desse estudo.

2.4.4 Países e linhas de pesquisa

Na Figura A.12 estão apresentados os países que possuem mais trabalhos citados.

Conforme ilustrado na Figura A.12, os Estados Unidos, Brasil e Alemanha destacam-se como os três países que mais citam os autores mencionados na Figura A.11 – Artigos mais citados.

2.5 Detalhamento, modelo integrador e validação por evidências

A terceira etapa do método TEMAC consiste em analisar as principais contribuições e abordagens presentes na base consolidada, por meio de Co-citação e Coupling. Na co-citação, tem-se uma perspectiva das abordagens mais utilizadas por meio da análise de trabalhos que têm literaturas base semelhantes. Por meio do coupling, é possível inferir as principais fontes de pesquisa, isto é, os trabalhos que não podem faltar na revisão sistemática [10].

2.5.1 Co-occurrence

Nesta análise, o objetivo é estabelecer uma medida de similaridade com base no conteúdo dos documentos [27]. Após esta etapa de aprimoramento bibliográfico, as informações foram inseridas no software VOSviewer para criar visualizações de redes bibliométricas. A figura A.13 apresenta inicialmente os termos de maior ocorrência e relevância, permitindo identificar os termos mais frequentes e pertinentes nos artigos analisados.

As palavras-chave encontradas na rede bibliométrica apresentada na Figura 2.13 relaciona o termo principal *Requirements elicitation* (Elicitação de Requisitos), com cinco clusters relevantes ao tema, representados principalmente por *Risk management* (Gerenciamento de risco) em *Software development* (Desenvolvimento de software), *Risk management* (Gerenciamento de risco) utilizando Agile (Agilidade) e *Requirements elicitation* (Elicitação de requisitos) utilizando *Agile method* (Métodos ágeis). No cluster 1 (Laranja) termo que mais se destaca, cuja tradução é inovação na *Requirements elicitation* (Elicitação de Requisitos), consiste em apresentar os desafios de utilizar as práticas de elicitação usando *Agile method* (Métodos ágeis)[28]. No cluster 2 (azul) concentram-se os temas relacionados aos riscos no desenvolvimento de software, que possuem tarefas complexas e exaustivas. Destaca-se entre os fatores mais complexos no desenvolvimento, a fase de entendimento das necessidades do cliente e escrita das estórias de usuários [29]. No cluster 3 (roxo) são explorados temas acerca da qualidade dos requisitos levantados,

retrabalho e possíveis riscos na fase de elicitação de requisitos [30]. No cluster 4 (vermelho) destacam-se assuntos referentes ao gerenciamento do risco, utilização do método ágil SCRUM na qual é o método mais utilizado atualmente [12]. O cluster 5 (verde) especifica algumas facilitações, métodos de descoberta de necessidade do cliente, assim como o levantamento das dores e desconfortos que o cliente possui e que precisam serem sanadas com o desenvolvimento da solução. [31].

2.5.2 Co-citação

Após um levantamento mais apurado da bibliografia, foi realizada a análise de co-citation e bibliographic coupling, mostrado na Figura A.14. De acordo com Mariano e Rocha [20], a primeira análise verifica os artigos que tiveram suas citações regulares conjuntamente, podendo sugerir que haja semelhança entre os assuntos em estudo. A partir dos metadados dos 16 artigos encontrados nas bases WoS e Scopus possibilitou-se a construção da rede de cocitação por meio de visualização de rede, conforme mostrado na Figura A.14.

Verifica-se a partir da Figura A.14, a formação de três principais clusters nas cores vermelho, azul e verde. Por meio das citações geradas, foi possível analisar os principais artigos de cada cluster com o objetivo de auxiliar na pesquisa e obter melhores conclusões a respeito do estudo. O cluster 1 (vermelho) concentra-se em estudos sobre a crescente busca pelas empresas em obter processos e métodos ágeis para acompanhar o mercado tecnológico, que isso acarreta em riscos que precisam ser mitigados o quanto antes, pois entregar serviço/produto com qualidade e rápido é um dos maiores desafios de uma empresa de tecnologia. O estudo realizado por Boehm [31], testou uma metodologia chamada *Resilient Agile (RA)* que visa resolver problemas e, ao mesmo tempo, acelera o desenvolvimento. Em relação ao cluster 2 (azul) observa-se o foco no estudo no uso de métodos ágeis na mitigação dos riscos encontrados no desenvolvimento de Software. Segundo Dyba e Fillippetto [32], O Scrum é como indiscutivelmente o mais popular dos métodos ágeis neste momento, o Scrum é um foco razoável, mas é um dos vários métodos, seja caracterizado como ágeis ou não, baseados em um processo incremental e ciclo de vida iterativo. O cluster 3 (verde) é focado nos desafios encontrados na implantação de métodos ágeis em fabricas de software. Wang e Boehm [33] avalia o relacionamento entre a Engenharia de Requisitos (ER) e os Métodos Ágeis que tem atraído bastante atenção. A ER desempenha um papel fundamental em todos os tipos de processos de desenvolvimento de software. O desenvolvimento ágil de software vem crescendo em popularidade e, em contraste com a extensa pesquisa de ER focada no desenvolvimento de software tradicional, o papel da ER no desenvolvimento ágil ainda não foi estudado em profundidade. A ER é utilizada de forma diferente no desenvolvimento ágil quando comparada aos processos de desen-

volvimento mais tradicionais. Porém por se tratar da fase mais complexa e delicada do desenvolvimento de software, há vários aspectos importantes a serem considerados, tendo em vista seus riscos e impactos que essa fase dentro de um método ágil pode causar ao projeto e a empresa.

2.5.3 Coupling

Nesta análise, adota-se uma métrica de busca baseada na premissa de que artigos citando os mesmos trabalhos possuem similaridades temáticas. Embora essa abordagem e a análise de *coupling* tenham semelhanças conceituais, produzem resultados distintos. A análise de *coupling* visa identificar os principais e mais atuais campos de pesquisa [35], destacando temas em ascensão para futuros estudos. Com base nisso, os artigos foram catalogados no software VOSviewer, como ilustrado na Figura A.15.

Na Figura A.15, Ferrari [37] emerge como uma referência significativa no assunto abordado. Ele introduz um modelo sustentado tanto empiricamente quanto teoricamente voltado para avaliação e mitigação de riscos associados ao compartilhamento eficaz de conhecimento em contextos de desenvolvimento ágil. Este modelo é fruto de observações empíricas em quatro projetos ágeis de duas empresas de software, combinadas com estudos anteriores sobre estratégia de risco e compartilhamento de conhecimento em desenvolvimento de software. Situando-se na vasta tradição de modelos de gestão de riscos específicos para o desenvolvimento de software, sua eficácia prática foi confirmada em uma das empresas analisadas. O modelo proposto fornece conceitos e processos para avaliar o perfil de risco de um projeto no que tange ao compartilhamento de conhecimento, estabelecendo um plano estratégico para atenuar esses riscos. O estudo sublinha como diferentes abordagens no gerenciamento de risco podem influenciar os resultados finais de um projeto.

No artigo "Um estudo do impacto da volatilidade dos requisitos no desempenho de projetos de software" explora profundamente a ideia do custo irrecuperável (sunk cost, em inglês) e como ele influencia as decisões tomadas em projetos de TI. A noção de custo irrecuperável refere-se a um investimento que já foi feito e não pode ser recuperado, também chamado de retrabalho devido a especificação de requisitos incompletos e/ou falhos, e que, frequentemente, leva a decisões contínuas de investimento, mesmo quando as perspectivas de sucesso do projeto são questionáveis.

Os autores iniciam com observações sobre como a influência do custo irrecuperável em decisões pode levar a um comportamento de escalada, isto é, continuar investindo em um projeto mesmo quando há evidências claras de que ele pode não ser bem-sucedido. Este comportamento é amplamente observado em vários setores, mas o artigo foca es-

pecificamente nos riscos encontrados na fase de elicitação de requisitos em projetos de TI.

Há uma menção interessante sobre o "síndrome dos 90%", que sugere que os projetos geralmente parecem estar "90% completos" por muito tempo antes de realmente chegarem a 100%. Essa percepção pode ser uma das razões pelas quais os gestores continuam investindo tempo e recursos em um projeto, esperando que ele esteja quase completo, quando, na realidade, ainda há um longo caminho a percorrer.

Dentre os fatores que podem influenciar a decisão de continuar investindo em um projeto, o artigo destaca a presença ou ausência de uma alternativa viável. A presença de uma alternativa pode atenuar o efeito do custo irrecuperável na tomada de decisões. Em outras palavras, se houver uma opção alternativa viável disponível, os gestores podem ser menos propensos a continuar investindo em um projeto fadado ao fracasso devido ao custo irrecuperável gasto na elicitação de requisitos e conseqüentemente nas mudanças das funcionalidades desenvolvidas.

Os autores também observam que, em alguns casos, o comprometimento com um projeto pode ser tão forte que os tomadores de decisão ignoram ou subestimam o custo irrecuperável, não levando em consideração os riscos apresentados e concentrando-se em outros fatores, como o potencial de conclusão do projeto.

Uma parte vital do artigo aborda o papel da pesquisa experimental em entender esses fenômenos. Os autores conduziram experimentos para avaliar o impacto dos riscos na fase de elicitação e nas decisões dos participantes. Seus resultados indicam que os participantes são, de fato, influenciados por custos irrecuperáveis, mas essa influência pode ser atenuada quando apresentados com informações claras sobre alternativas viáveis.

Por fim, o artigo conclui enfatizando a necessidade de mais pesquisas no campo, especialmente em ambientes reais e não apenas em cenários experimentais. Também destaca a importância da conscientização sobre o efeito do custo na fase de elicitação para que os gestores possam tomar decisões mais informadas e, potencialmente, mais eficazes.

Em síntese, o artigo fornece uma análise abrangente sobre a influência do custo de retrabalho irrecuperável na tomada de decisões em projetos de TI. Destaca a propensão dos gestores a continuar investindo em projetos devido a investimentos anteriores, mesmo frente a indícios claros de possíveis falhas. Os autores sublinham a importância de reconhecer este viés e a necessidade de considerar alternativas viáveis quando se avalia o progresso e a viabilidade de um projeto. A pesquisa contribui significativamente para o campo da gestão de projetos, gestão de riscos na fase de elicitação, oferecendo insights valiosos sobre como melhorar a tomada de decisões e potencialmente evitar o desperdício de recursos em projetos fadados ao fracasso devido à influência de elicitações incorretas e incoerentes ao negócio.

No artigo "*A model of Requirements engineering in software startups*" de Melagali (2019), o artigo aborda a questão do compartilhamento eficaz de conhecimento em equipes de desenvolvimento ágil de software, reconhecendo que o sucesso de projetos ágeis depende em grande parte da comunicação e colaboração entre os membros da equipe. A falta de compartilhamento de conhecimento pode levar a riscos que afetam negativamente o desempenho do projeto. Para abordar essa questão, os autores desenvolvem um modelo de gerenciamento de riscos que visa identificar, avaliar e mitigar os riscos relacionados ao compartilhamento de conhecimento.

A revisão da literatura destaca a importância do compartilhamento de conhecimento em equipes de desenvolvimento ágil e os desafios que as equipes podem enfrentar nesse processo. Autores como Vidgen e Wang (2009) e Ramesh et al. (2010) enfatizam a necessidade de superar barreiras de comunicação e compartilhamento de conhecimento para o sucesso de projetos ágeis. Além disso, a literatura sobre gerenciamento de riscos, incluindo as abordagens de Davis (1982) e Iversen et al. (2004), fornece insights sobre a identificação e mitigação de riscos em projetos de software.

O estudo empírico baseia-se em dados coletados de quatro projetos de desenvolvimento de software em duas empresas de software. Os projetos foram divididos em dois grupos: projetos de alto desempenho (Alpha 1 e Beta 1) e projetos de baixo desempenho (Alpha 2 e Beta 2). A coleta de dados incluiu entrevistas com membros da equipe, análise documental e observação direta. Os resultados do estudo destacam a existência de riscos relacionados ao compartilhamento de conhecimento em todos os projetos, independentemente do desempenho geral do projeto. No entanto, os projetos de alto desempenho implementaram um maior número de ações de resolução em comparação com os riscos enfrentados, enquanto os projetos de baixo desempenho implementaram menos ações de resolução.

Com base nos resultados, os autores desenvolvem um modelo de gerenciamento de riscos que visa melhorar o compartilhamento de conhecimento em equipes de desenvolvimento ágil. O modelo consiste em cinco componentes principais: o perfil de risco do projeto, o processo passo a passo com heurísticas, o plano de estratégia de resolução, as áreas e itens de risco e as estratégias e ações de resolução. O modelo pode ser usado em práticas ágeis, como reuniões de retrospectiva, para avaliar riscos, priorizar ações de resolução e desenvolver um plano geral de estratégia de resolução. O envolvimento de diversas partes interessadas é essencial para garantir a eficácia do processo de gerenciamento de riscos. Além disso, a aplicação regular do modelo permite a aprendizagem organizacional e a melhoria contínua.

O artigo contribui teoricamente ao preencher uma lacuna na literatura sobre como equipes ágeis podem avaliar e mitigar riscos relacionados ao compartilhamento de conhe-

cimento. Além disso, o modelo oferece categorias conceituais e heurísticas que podem ser usadas em futuras pesquisas sobre comunicação e compartilhamento de conhecimento em contextos ágeis. Do ponto de vista prático, o modelo fornece uma ferramenta valiosa para equipes de desenvolvimento ágil melhorarem seu desempenho ao abordar riscos de conhecimento.

O estudo tem algumas limitações, incluindo o número limitado de projetos avaliados. Pesquisas futuras podem expandir o modelo por meio de estudos em diferentes empresas e projetos. Além disso, pesquisas adicionais podem explorar abordagens mais sistemáticas para avaliação de riscos, considerar a aplicação do modelo em diferentes fases de projetos e examinar os efeitos a longo prazo das ações de resolução. Em resumo, o artigo apresenta um modelo de gerenciamento de riscos para equipes de desenvolvimento ágil, visando melhorar o compartilhamento de conhecimento e mitigar os riscos associados. O modelo oferece insights teóricos e práticos no processo de desenvolvimento de software.

No estudo realizado por Shrivastava (2017), *Requirements Engineering in Agile Projects*, o artigo aborda a engenharia de requisitos desempenha um papel fundamental no desenvolvimento de software, influenciando diretamente o sucesso de projetos de TI. Em particular, projetos que adotam metodologias ágeis destacam-se por sua flexibilidade e abordagem iterativa. Este resumo abordará os principais pontos de um artigo de revisão sistemática que investiga a engenharia de requisitos em projetos ágeis.

A revisão sistemática abrangeu um total de 580 estudos, resultando em 24 artigos relevantes após a aplicação de critérios rigorosos de inclusão e exclusão. A maioria dos artigos (20) foi obtida por meio de buscas automáticas, com os Estados Unidos sendo o país mais proeminente nas pesquisas, representando sete dos artigos selecionados. Os autores identificaram sete estratégias para elicitar requisitos em projetos ágeis, com entrevistas com o cliente sendo a técnica mais amplamente utilizada. Além disso, foram citadas técnicas como JAD, Grupo Focal, *Brainstorm*, Formulários, *Trawling e Workshop*. Entretanto, apenas 10 dos 24 artigos mencionaram as técnicas utilizadas. A especificação de requisitos em projetos ágeis envolve uma variedade de técnicas, com *User Stories* e Protótipos sendo as mais comuns. Um total de 20 técnicas diferentes foram mapeadas nos estudos, com cinco artigos relatando o uso de técnicas tradicionais, como *Use Cases* e Cenários.

Os estudos identificaram diversos desafios e limitações da engenharia de requisitos em projetos ágeis. Problemas como pouca disponibilidade do cliente, documentação insuficiente, controle insuficiente de mudanças nos requisitos e dificuldade em estimar custo, prazo e produtividade foram destacados. Além disso, a categoria de Gestão de Requisitos apresentou uma série de desafios, incluindo questões relacionadas ao escopo e à mudança. Os resultados desta revisão sistemática apontam para a necessidade de ações tanto na academia quanto na indústria para abordar os problemas identificados na engenharia de

requisitos em projetos ágeis. A qualidade dos artigos selecionados foi variável, e a revisão destaca a necessidade de uma melhor atenção dos pesquisadores para reduzir o viés nos estudos. A academia deve investigar ajustes nas técnicas de especificação de requisitos em projetos ágeis e examinar como as práticas de engenharia de requisitos afetam a agilidade do processo. Esta revisão sistemática forneceu uma visão abrangente dos desafios e práticas de engenharia de requisitos em projetos ágeis. Os resultados destacam a importância de enfrentar os problemas identificados e sugerem que futuras pesquisas podem ajudar as empresas a melhorar seus processos de desenvolvimento de software em ambientes ágeis. No artigo de Hoda (2016) "*Multi-level agile project management challenges a self-organizing team perspective*", o artigo apresenta uma análise abrangente das práticas de Engenharia de Requisitos (RE) em projetos ágeis, com foco em resultados de uma pesquisa internacional conduzida pelo projeto NaPiRE (*Naming the Pain in Requirements Engineering*). O estudo investiga o estado atual da prática de RE ágil, bem como os problemas contemporâneos enfrentados por organizações que adotam abordagens ágeis.

O artigo destaca que as práticas de RE em ambientes ágeis são distintas das abordagens tradicionais de desenvolvimento de software. No entanto, até o momento, há pouco conhecimento empírico sobre o estado da prática e os desafios específicos enfrentados pela RE ágil. Portanto, a pesquisa NaPiRE se concentrou em preencher essa lacuna, coletando dados de representantes de 92 organizações em todo o mundo. As mudanças nos requisitos são comuns em projetos ágeis e a pesquisa mostrou que a maioria das organizações lida com essas mudanças por meio da atualização do *backlog* do produto. Os Impactos de mudanças nos requisitos são analisados principalmente entre os requisitos e, em menor medida, entre requisitos e código. A gestão explícita de traços entre requisitos e código e entre requisitos e documentos de design também é uma prática comum. A pesquisa indica que a melhoria contínua da RE é uma prática comum em projetos ágeis. Essa melhoria é motivada principalmente por fatores intrínsecos, como a busca por identificar pontos fortes e fracos individuais. As organizações geralmente adotam uma abordagem autônoma para melhorar seus processos de RE, com a equipe do projeto desempenhando um papel central nesse processo.

O artigo também identifica problemas comuns enfrentados pelas organizações na Engenharia de requisitos (RE) ágil. Alguns dos problemas principais incluem requisitos em constante mudança, requisitos incompletos e falhas na comunicação dentro das equipes de desenvolvimento e entre desenvolvedores e clientes. Esses problemas podem levar a retrabalho, insatisfação do cliente e riscos não gerenciados. Em resumo, o artigo fornece uma análise detalhada das práticas de RE em projetos ágeis e destaca que, apesar das diferenças entre abordagens ágeis e tradicionais, muitos aspectos da RE ágil são semelhantes. A pesquisa também aponta para problemas comuns que as organizações enfrentam

na RE ágil, enfatizando a importância e comunicação eficaz e da gestão de requisitos em ambientes ágeis.

Para pesquisas futuras, é importante aprofundar a compreensão das práticas ágeis de elicitação de requisitos e sua relação com os problemas identificados. Além disso, é crucial continuar refinando os instrumentos de pesquisa para capturar especificidades das práticas ágeis de RE. Essas descobertas fornecem insights valiosos para profissionais e pesquisadores interessados em aprimorar a Engenharia de Requisitos em projetos ágeis.

No artigo "Autenticidade na aprendizagem baseada em projetos para desenvolvimento de software: Uma Revisão Sistemática de Literatura" de Vinícius Gomes Ferreira e Edna Dias Canedo investiga a contribuição da Aprendizagem Baseada em Projetos (ABP) na autenticidade do desenvolvimento de software. A pesquisa destacou o uso do Scrum como o processo de desenvolvimento mais empregado em unidades de ABP que resultam em softwares autênticos, enfatizando a importância da divisão dos alunos em pequenos grupos e o papel vital dos monitores na qualidade do software.

A autenticidade, definida pela proximidade com situações reais e pela qualidade do artefato produzido, é vista como um elemento crucial na educação progressista, promovendo a autonomia do aluno e a relevância prática dos projetos desenvolvidos. O estudo revelou que, além da adoção do Scrum, a configuração de equipes pequenas e a supervisão ativa contribuem significativamente para a autenticidade e qualidade dos softwares desenvolvidos em ambientes educacionais.

A Revisão Sistemática da Literatura (RSL) baseou-se em questões específicas que abordam tanto aspectos técnicos quanto pedagógicos, destacando a importância do ensino teórico e técnico, a gestão do tempo nas unidades de ABP, modelos e técnicas de engenharia de software utilizados, e a formação de grupos de alunos. Os resultados indicam que, apesar dos desafios na definição e obtenção da autenticidade, a ABP, apoiada por processos ágeis como o Scrum e técnicas de engenharia de software, pode melhorar significativamente a experiência de aprendizagem e o desenvolvimento de softwares autênticos.

Este trabalho contribui para a educação em Ciência da Computação ao fornecer diretrizes para a formação de unidades de ABP que visam a autenticidade do software, beneficiando tanto os alunos, através de uma experiência educacional mais engajadora e prática, quanto a comunidade externa, pelo desenvolvimento de softwares úteis e de qualidade. Futuramente, sugere-se a elaboração de um modelo de ABP focado na autenticidade dos artefatos desenvolvidos, promovendo uma educação mais alinhada às demandas reais do mercado e da sociedade.

O artigo "Riscos de Engenharia de Requisitos com a utilização de Metodologias Ágeis" de Marcelo Ferreira de Almeida Araújo e Edna Dias Canedo, explora a relação entre os mé-

todos ágeis e a engenharia de requisitos, destacando os riscos inerentes e estratégias de mitigação. A pesquisa envolveu a análise de 21 artigos científicos e entrevistas com 6 gerentes de projetos, além de questionários respondidos por 53 desenvolvedores, visando identificar os riscos mais impactantes em projetos ágeis e as ações adotadas para tratá-los.

Os riscos identificados foram categorizados em cinco áreas principais: negócios e barreiras de comunicação, requisitos funcionais e não funcionais, mudanças de requisitos, questões relacionadas ao cliente e qualidade. Entre os riscos mais significativos estão o não entendimento dos requisitos pelo desenvolvedor, a falta de disponibilidade do cliente e falhas na gestão de mudanças. Curiosamente, a pesquisa também revelou riscos não citados previamente na literatura, como mudanças na equipe de desenvolvimento, equipes grandes com mais de 7 pessoas, e a falta de aderência à cultura ágil na organização.

Para a mitigação desses riscos, estratégias como melhoria da comunicação com o cliente, participação ativa do Product Owner (PO), uso de técnicas como Design Thinking e Brainstorming, e a promoção de uma cultura ágil na organização foram sugeridas. Além disso, foi proposto um framework para melhorar a qualidade das User Stories, incluindo critérios como clareza, atomicidade, orientação ao problema, entre outros.

Este estudo ressalta a importância do entendimento claro dos requisitos e da comunicação eficaz entre todos os envolvidos em um projeto ágil. Os achados enfatizam que, embora os métodos ágeis possam aumentar a eficiência e a satisfação do cliente, eles também trazem desafios específicos que necessitam ser abordados para minimizar os riscos e garantir o sucesso do projeto.

Futuras pesquisas são sugeridas para explorar as discrepâncias nas percepções entre desenvolvedores e líderes, bem como para investigar mais profundamente as causas subjacentes aos riscos identificados e suas soluções efetivas.

2.6 Determinação do Enfoque Teórico da Pesquisa

Este capítulo apresentou um levantamento do estado da arte, visando mostrar as principais análises envolvendo os temas mais relevantes, a fim de proporcionar um melhor entendimento sobre o contexto geral do trabalho.

Esta pesquisa destaca a importância crítica de uma abordagem equilibrada para a gestão de riscos na fase de elicitação de requisitos em desenvolvimento de software. Tradicionalmente, conforme apontado por Dalcher (2002), existia uma tendência de desvalorizar as abordagens de gestão de riscos formalizadas, priorizando a filosofia ágil de colocar pessoas acima de processos, o que poderia inadvertidamente suprimir os potenciais benefícios de comportamentos de risco calculados. No entanto, estudos mais recentes, como os de

Boehm e Turner (2013), ressaltam a relevância de harmonizar os pontos fortes da gestão de riscos com a elicitação de requisitos, sugerindo uma sinergia entre estes dois aspectos.

Os resultados encontrados nesta pesquisa reforçam a visão de que adotar práticas de gestão de riscos pode ser extremamente benéfico para avaliar e mitigar os riscos associados à elicitação de requisitos. Além disso, essa abordagem contribui para a criação de modelos mentais compartilhados entre as partes interessadas, facilitando a tomada de decisão coletiva e apoiando um processo de desenvolvimento mais eficiente e seguro. Portanto, a pesquisa oferece um novo entendimento sobre a integração da gestão de riscos na fase de elicitação de requisitos, enfatizando a sua importância estratégica no contexto do desenvolvimento ágil de software. (Lyytinen et al 1998) [7].

Capítulo 3

Referencial Teórico

O referencial teórico tem por objetivo elucidar as primeiras definições que serão necessárias durante a fase de apresentação como resultados.

3.1 Cadeia de Valor

O contexto de uma empresa refere-se ao conjunto de condições internas e externas que influenciam sua operação e estratégia. Internamente, inclui a cultura, recursos, capacidades e estrutura organizacional. Externamente, abrange fatores como o mercado, concorrência, tendências tecnológicas, regulamentações e condições econômicas. Identificar o contexto ajuda a empresa a entender suas forças e fraquezas, bem como as oportunidades e ameaças do ambiente, fundamentais para o planejamento estratégico e a tomada de decisões informadas[12]. Existem algumas ferramentas de mercado para análise de contexto empresarial, incluindo análise de cinco forças de Porter, cadeia de valor de Porter, análises SWOT (Forças, Fraquezas, Oportunidades, Ameaças) e PESTEL (Político, Econômico, Social, Tecnológico, Ambiental, Legal). Essas ferramentas ajudam a compreender o ambiente interno e externo da empresa, além de permitir um planejamento estratégico mais robusto e a identificação de oportunidades de melhoria e crescimento.

A cadeia de valor é uma metodologia usada para coordenar processos e mapear quais as vantagens competitivas da corporação. Essa ferramenta foi criada em 1985, por Michael Porter, um dos principais professores da *Harvard Business School*.

Sua principal função é revelar todas as atividades que a corporação produz para gerar valor ao público, e estabelecer quais os elos entre a empresa e sua audiência. Ao fortalecer as ligações, é possível desenvolver vantagens competitivas para a instituição e contribuir para o crescimento do negócio e seus lucros.

Cada empresa conta com uma proposta de valor, ou seja, vantagens exclusivas quanto aos produtos e serviços prestados. Isso é percebido pelo cliente por meio da criação

de soluções inovadoras, atendimento eficiente e desenvolvimento de produtos ou serviços capazes de atender à dor do público.

Desse modo, é possível estabelecer que a cadeia de valor é um conjunto de processos interligados — por eles — necessários para viabilizar uma percepção positiva dos clientes com relação às soluções desenvolvidas pela organização.

A análise SWOT foi resultado do trabalho de pesquisa realizado na década de 1960 por Albert Humphrey, consultor em gestão do instituto de pesquisas da Universidade de Stanford. Neste período, o planejamento empresarial pouco difundido necessitava um método para a criação de planejamentos de longo prazo, executáveis e plausíveis. Inicialmente propôs o acrônimo SOFT, formado por: *Satisfactory* (Satisfatório), *Opportunity* (Oportunidade), *Fault* (Falha), *Threat* (Ameaça) com o intuito de acrescentar maior responsabilidade e objetividade ao processo de planejamento. Até que em 1964 Urick e Orr alteraram o F para W de *Weaknesses* (Fraquezas) e o S se referiria a *Strengths* (Forças), além de apresentarem a SWOT como uma ferramenta autônoma, independente do processo produtivo.

A análise SWOT, também conhecida como FOFA em português, é uma metodologia de planejamento estratégico que reúne dados a fim de caracterizar o cenário em que uma empresa se encontra para auxiliar a tomada de decisões.

Por ser uma ferramenta simples e versátil, é capaz de se adequar a empresas de diversos tipos e tamanhos, que considera separadamente dois ambientes: externo e interno. A sigla SWOT provém das iniciais de quatro palavras em inglês, que são:

- *Strengths*, que significa Forças;
- *Weaknesses*, que significa Fraquezas;
- *Opportunities*, que significa Oportunidades;
- *Threats*, que significa Ameaças.

Através da cadeia de valor em conjunto a análise SWOT, a empresa estudada ganhará compreensão aprofundada sobre suas operações e como melhorar e alinhar atividades produtivas e estratégicas para uma visão integrada dos processos e gestão. Isso facilitará o desenvolvimento de uma proposta de valor ao cliente, refinar e aproveitar pontos fortes em estratégias de negócios, especialmente digital. Assim, o modelo não só melhorará o desempenho da empresa mas também fornecerá vantagens competitivas significativas no mercado.

Análise documental: A análise documental é um método de pesquisa que envolve a avaliação sistemática de documentos e registros para extrair informações, padrões e insights

relevantes. Este processo é fundamental em diversos campos, incluindo estudos históricos, pesquisas acadêmicas e projetos de desenvolvimento, ajudando a entender contextos, fundamentar teorias ou avaliar a implementação e resultados de políticas e programas. Ao analisar documentos, pesquisadores podem identificar tendências, confirmar hipóteses ou revelar novas questões a serem exploradas. Os documentos oferecem registros estáveis que podem ser revisitados e são sociais, o que reflete as condições de sua criação. A análise documental permite aos pesquisadores explorar como os documentos funcionam na comunicação e na constituição de realidades sociais. Essa técnica é valorizada por sua capacidade de fornecer dados históricos e contextuais, complementando outras formas de pesquisa [13].

3.2 Gestão de Processos

A análise de processo por meio da notação BPMN (*Business Process Model and Notation*) possibilita auxiliar na identificação e correção de problemas que se relacionam entre si. Para auxiliar no processo de elicitação de requisitos, é possível utilizar alguns métodos de mapeamento do processo da área ou até mesmo da organização que necessita de uma solução automatizada melhorar a operação da empresa. Muitas organizações usam o modelo BPMN para documentar operações de negócios e para formalizar esses requisitos em projetos de engenharia de software. O uso do BPMN (*Business Process Model and Notation*) tem por intuito conhecer e explorar os processos que são realizados numa organização, mensurá-los e dispor de ciclos de melhoria contínua para evolução da organização. Nessa linha, o mapeamento de processos consiste na análise ("AS-IS"), fornecendo uma visão detalhada do processo em questão, mostra a relação entre as entradas as saídas e as tarefas executadas. Existem várias formas de externalizar e documentar o processo intelectual para modelagens de processos de negócios, como por exemplo:

Business Process Model and Notation (BPMN): Padrão criado pelo *Object Management Group* Object Management Group (OMG) [7], útil para apresentar um modelo à públicos-alvo diferentes:

- Fluxograma: Inicialmente reconhecido como padrão pelo *American National Standards Institute* American National Standards Institute (ANSI), apresenta um conjunto conciso de símbolos.
- *Event-driven Process Chain* Event-driven Process Chain (EPC): Considera eventos como gatilhos para uma etapa do processo;
- *Unified Modeling Language* Unified Modeling Language (UML): mantido pela OMS® consiste em um conjunto padrão de notações técnicas de diagramação orientado à descrição de requisitos de sistemas de informação. Existe uma tendência de aumento da

importância da modelagem. Linguagens de especificação como UML e BPMN são aplicadas na maioria dos projetos de engenharia de software, atualmente. Todavia, cada uma possui aplicação em etapas específicas do processo de engenharia de software. Ao estudar sobre o uso do BPM em processos de elicitação de requisitos identificou-se que uma das notações mais utilizadas é o BPMN, quando o assunto está focado modelagem do processo de software com um todo [14].

Mapear processos é uma atividade crítica para muitas organizações que buscam melhorar sua eficiência, entender seus fluxos de trabalho ou implementar novos sistemas. Existem ferramentas disponíveis para essa finalidade. Segue a lista de ferramentas mais populares para mapeamento de processos e suas respectivas vantagens e desvantagens: [15]

Microsoft Visio:

A *Microsoft Visio* é uma aplicação de software usada para desenhar e criar diagramas, gráficos e ilustrações vetoriais. Faz parte do conjunto de produtos da Microsoft e é amplamente utilizado por profissionais de TI, gerentes de projeto, engenheiros e outros profissionais para visualizar e comunicar informações complexas de maneira clara e concisa através de diagramas.

- Vantagens: *Interface* intuitiva, integração com outros produtos da *Microsoft*, amplamente utilizado em muitas organizações.
- Desvantagens: Custo de licença, pode ser complexo para usuários iniciantes, menos funcionalidades de colaboração em comparação com algumas ferramentas modernas.

Lucidchart: O *Lucidchart* é uma aplicação web de diagramação e visualização gráfica que permite aos usuários criar, colaborar e compartilhar diagramas em tempo real. É uma alternativa popular a outras ferramentas de diagramação, como o Microsoft Visio. O Lucidchart é conhecido por sua interface intuitiva baseada em arrastar e soltar e por ser uma ferramenta versátil que atende a várias necessidades de diagramação

- Vantagens: Baseado na web, facilita a colaboração em tempo real, interface arrastar e soltar, integração com G Suite, *Microsoft Office* e outros.
- Desvantagens: Recursos avançados requerem pagamento, depende de conexão com a internet para funcionalidade completa.

BPMN.io: O BPMN.io é uma ferramenta open-source baseada na web para modelagem e visualização de processos usando a notação BPMN (*Business Process Model and Notation*). Ele oferece uma maneira intuitiva e fácil de criar, editar e visualizar diagramas BPMN diretamente em navegadores da web.

- Vantagens: Grátis, *open-source*, específico para BPMN (*Business Process Model and Notation*), leve e fácil de usar.
- Desvantagens: Menos recursos em comparação com ferramentas comerciais, pode não ser ideal para usuários não técnicos.

ARIS Express: O ARIS Express é uma versão mais simplificada e gratuita do ARIS, uma popular ferramenta de modelagem de processos de negócios. Enquanto o ARIS Express oferece muitos recursos úteis, ele também tem suas limitações, especialmente quando comparado com versões mais robustas e pagas da mesma família de produtos.

- Vantagens: Gratuito
- Desvantagens: Sendo uma versão gratuita, o ARIS Express não oferece todos os recursos avançados encontrados nas versões pagas do ARIS.

Bizagi

A Bizagi é uma plataforma de automação de processos empresariais (*Business Process Management*, BPM) que oferece soluções abrangentes para modelar, automatizar e otimizar processos de negócios. A Bizagi tem como objetivo transformar organizações tornando-as mais ágeis, eficientes e adaptáveis por meio da digitalização de processos.

- Vantagens: Interface Intuitiva, Padrão BPMN, versão gratuita, simulação de processos, integra com diversos outros sistemas, disponível em dispositivos móveis.
- Desvantagens: Curva de aprendizado, performance e custo.

Camunda

O Camunda é uma plataforma *open-source* de automação de processos de negócio baseada no padrão BPMN (*Business Process Model and Notation*). Ele permite modelar, executar e monitorar workflows e decisões empresariais. Além do BPMN, o Camunda também suporta DMN (*Decision Model and Notation*) para modelagem de regras de negócio e CMMN (*Case Management Model and Notation*) para gerenciamento de casos.

O Camunda é frequentemente utilizado em arquiteturas orientadas a microsserviços, sendo integrado a aplicações Java, Spring Boot e até soluções baseadas em APIs REST.

- Vantagens: *Open-source* e flexível, Padrão BPMN e DMN, Integração com Java e Spring Boot, Execução leve e escalável, Ferramentas de Monitoramento, APIs REST e gRPC.
- Desvantagens: Pouco intuitiva, Foco em desenvolvedores, Interface básica na versão *open-source*, Falta de suporte nativo a NoSQL.

Tendo em vista as necessidades da empresa na aquisição de uma solução para modelagem, simulação e automação de processos de negócios que atentam as necessidades específicas da empresa, a ferramenta escolhida neste estudo foi o Camunda, devido às vantagens que oferece, mas principalmente a possibilidade de realizar integrações com ferramentas de gestão e controle de demandas para otimização da fase de elicitação de requisitos parte fundamental no processo de desenvolvimento de software. Outro ponto é que por se tratar de uma ferramenta com o custo menor, foi uma decisão da empresa sua aquisição.

Com isso, a partir da definição da ferramenta, usou-se a metodologia de implantação da modelagem de processo de uma organização, dividindo-a em quatro fases: [15]

- Planejamento
- Mapeamento e Análise
- Monitoramento e Controle
- Avaliação do desempenho

3.3 Gestão de Riscos na fase de elicitação de requisitos

A ISO 31000:2018 oferece diretrizes para a gestão efetiva de riscos, propõe um processo estruturado e abrangente. O conceito de risco geralmente se refere a eventos adversos com uma probabilidade conhecida. A incerteza, elemento intrínseco à gestão de riscos, surge da falta de informação e está associada a probabilidades desconhecidas. O risco pode impactar de várias formas os objetivos de uma organização. Organizações de diversos tamanhos e setores estão sujeitas a influências e fatores tanto internos quanto externos. [16] O risco pode se materializar especialmente quando vulnerabilidades, ou seja, a ausência de controles específicos, são exploradas. Vulnerabilidades são conjuntos de falhas ou fatores internos que, se explorados por ameaças, podem levar a incidentes, eventos indesejados com consequências negativas. A gestão de riscos, de acordo com a norma ISO 31000:2018, deve ser proativa e focada na proteção dos ativos organizacionais, abrangendo elementos como pessoas, processos, ambiente, tecnologia, informação e comunicação, visando sempre apoiar e salvaguardar a organização.

De acordo com International Organization for Standardization (ISO) 31000:2018, ao realizar a gestão de riscos proativa é possível:

- Aumentar a probabilidade da organização de atingir os seus objetivos;

- Melhorar a identificação de oportunidades e ameaças;
- Atender às normas internacionais e requisitos legais e regulatórios pertinentes;
- Melhorar o reporte das informações financeiras e governança;
- Melhorar a confiança das partes interessadas;
- Estabelecer uma base confiável para a tomada de decisão e planejamento;
- Melhorar os controles, eficácia e a eficiência operacional;
- Melhorar a prevenção de perdas e a gestão de incidentes;
- Aumentar a resiliência da organização

Como uma das formas de resolução desses problemas, a ISO propõe um processo de gestão de riscos, conforme figura 3.1. [17]

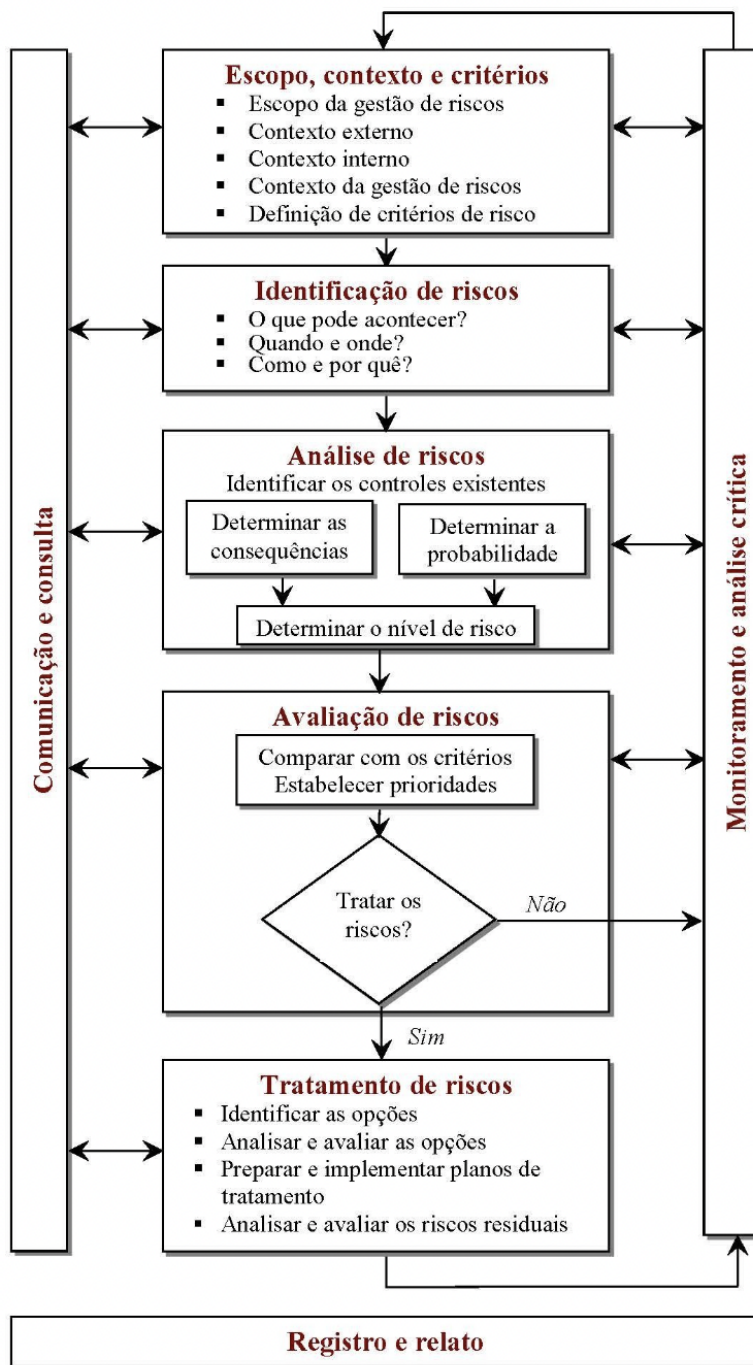


Figura 3.1: Processo de Gestão de Riscos detalhado (ISO 31000:2018)

Fonte: ISO 31000:2018

A primeira fase do processo é o **estabelecimento do contexto**, no qual utilizei a análise Strengths, Weaknesses, Opportunities, Threats (SWOT) para identificar tanto o cenário interno quanto o externo da organização.

A análise SWOT é uma metodologia estratégica empregada para avaliar os cenários interno e externo de uma organização, projeto ou circunstância específica. O termo "SWOT"origina-se do inglês, representando:

Strengths (Forças): Vantagens e qualidades internas que diferenciam uma organização de outras, englobando recursos especializados, competências distintas e outros elementos que proporcionem valor. *Weaknesses* (Fraquezas): Limitações e pontos vulneráveis internos que um projeto ou organização pode enfrentar ao ser comparado a competidores ou ao cenário no qual está inserido. *Opportunities* (Oportunidades): Cenários ou tendências externas benéficas, englobando mudanças mercadológicas, avanços tecnológicos ou alterações nas demandas dos consumidores. *Threats* (Ameaças): Desafios ou riscos oriundos de fatores externos, como alterações legislativas, entrada de novos concorrentes ou outros elementos que possam influenciar adversamente. O propósito central da análise SWOT é auxiliar organizações a reconhecerem suas forças e vulnerabilidades, assim como as chances e riscos presentes no ambiente externo. Esse discernimento possibilita a formulação de estratégias robustas e tomadas de decisão embasadas. Com a avaliação desses quatro pilares, as organizações conseguem vislumbrar suas capacidades, o panorama competitivo e, conseqüentemente, traçar um direcionamento estratégico apropriado [18].

Compreender os contextos interno e externo é primordial para identificar os objetivos e riscos organizacionais. Caso as metas não forem alcançadas, uma gestão de riscos efetiva torna-se indispensável. Internamente, avaliam-se aspectos como estrutura, responsabilidades e sistemas, proporcionando uma visão holística das competências e desafios da organização. Em contrapartida, o cenário externo abrange diversas dimensões do ambiente circundante, incluindo aspectos legais, sociais, financeiros, entre outros. É essencial considerar o relacionamento com stakeholders, pois suas percepções e valores podem influenciar decisivamente a organização. Integrando perspectivas dos cenários interno e externo, as entidades estão mais capacitadas para tomar decisões informadas e adaptar-se às vicissitudes do mundo corporativo.[13]

Identificação dos Riscos - Após o estabelecimento do contexto, se faz necessária a identificação dos riscos, ou seja, encontrar, reconhecer e registrar os riscos. Essa etapa deve ser baseada na identificação das fontes de riscos, áreas de impactos, eventos (incluindo mudanças nas circunstâncias) e suas causas e conseqüências potenciais. Esse processo busca responder perguntas como:

- O que pode atrapalhar o alcance dos objetivos ou resultados organizacionais?
- Quais são os eventos que podem afetar de alguma forma o sucesso de um determinado objetivo?
- Quais são as fontes dos riscos que permeiam os ativos organizacionais?

A saída do processo de identificação de riscos, é uma listagem abrangente e detalhada

dos potenciais riscos que possam criar, aumentar, evitar, reduzir, acelerar ou atrasar o cumprimento de determinado objetivo.

Entende-se que o processo de identificação de risco é crítico, tendo em vista que um risco não mapeado não poderá ser analisado posteriormente, o que pode impactar diretamente no insucesso da gestão de riscos proativa. Se faz necessária a identificação de todas as fontes de riscos, sendo estas sob o controle ou não da organização, mesmo que essas não sejam tão evidentes. Ao inventariar um risco identificado, deve-se documentar o impacto ou consequência dele, caso venham a se concretizar tornando-se um evento/incidente. Além de disso, é importante identificar e documentar o que pode acontecer, possíveis causas e cenários que mostram quais consequências esse incidente pode trazer para a organização. Para o processo de identificação dos riscos, se faz importante o envolvimento de pessoas que possuam domínio sobre o assunto de gestão proativa de riscos. Existem várias técnicas propostas para identificação de riscos como: Brainstorming, brainwriting, entrevistas, visitas técnicas, questionários, pesquisas, entre outras, e convém que elas sejam adequadas aos seus objetivos e capacidades e aos riscos enfrentados.

Uma vez que um risco é identificado, convém que a organização identifique quaisquer controles existentes, tais como funcionalidades projetadas, pessoas, processos e sistemas, que possam ser aplicados a esses riscos de forma a modificá-los de alguma forma. [19]

A Análise dos Riscos refere-se ao desenvolvimento da compreensão sobre o risco e à determinação do nível do risco, como também a definição da probabilidade do risco e a gravidade dos impactos associados aos riscos. Ela fornece uma entrada para o processo de avaliação de riscos e às decisões sobre se os riscos necessitam ser tratados. As consequências (impacto) e suas probabilidades são combinadas para determinar um nível de risco (escore de risco), sendo a probabilidade a chance de o evento ocorrer dentro do prazo previsto para se alcançar o objetivo e a consequência, os impactos que o risco trará para a organização caso aconteça. Devem ser identificados também os processos críticos de maior vulnerabilidade e as ameaças associadas a eles, os quais serão diminuídos a cada implementação de controle e formas de proteção ao ativo em cada momento. A análise dos riscos organizacionais é contínua e exige uma monitoração constante das ameaças que possam afetar os objetivos institucionais e deve considerar os resultados obtidos por meio da implementação dos controles em ciclos anteriores de forma que seja possível validar sua eficácia, melhorar ou substituir esses controles. Os impactos podem ter uma baixa consequência, porém alta probabilidade de ocorrência, ou uma alta consequência e baixa probabilidade, ou ainda algum resultado intermediário.[19]

Com a análise podemos acompanhar a evolução do risco, identificar se houve aumento no impacto ou na probabilidade, permitindo adotarmos uma estratégia mais assertiva.

A Avaliação dos Riscos consiste em comparar os níveis estimados de risco com critérios de risco definidos quando o contexto foi estabelecido [26], a fim de determinar a significância do nível e do tipo de riscos. O limite de exposição a riscos representa o nível de risco acima do qual é desejável o tratamento do risco [4]. Espera-se que, com os resultados do tratamento, o nível de risco residual fique abaixo do limite de exposição. A avaliação de riscos utiliza a compreensão do risco, obtida durante a análise de riscos, para tomar decisões sobre as ações futuras. Considerações éticas, legais, financeiras e outras, incluindo as percepções do risco, são também dados de entrada para a decisão [4]. As decisões na avaliação de um risco podem incluir, conforme figura 3.2.



Figura 3.2: Decisões na Avaliação de Riscos (ISO 31000:2018)

Fonte: ISO 31000:2018

A natureza das decisões que necessitam ser tomadas e os critérios que serão utilizados para tomar essas decisões foram decididos no estabelecimento do contexto, mas precisam ser revistos em mais detalhes nesta fase. A estrutura mais simples para a definição dos critérios de risco é um nível único que divide os riscos que necessitam de tratamento daqueles que não necessitam. Isso fornece resultados atrativamente simples, porém não reflete as incertezas envolvidas na estimativa de riscos e na definição da fronteira entre aqueles que necessitam de tratamento e aqueles que não necessitam [19].

A decisão de como tratar um risco frequentemente leva em consideração os custos e benefícios de assumi-lo, bem como os custos e benefícios de implementar controles mais aprimorados. Uma abordagem comum é categorizar os riscos em três faixas distintas, chamada de Matriz de Riscos.

A Matriz de Riscos, também conhecida como Matriz de Probabilidade e Impacto, é uma ferramenta crucial no gerenciamento de riscos, permitindo uma identificação visual dos principais pontos que exigem atenção. Essa ferramenta é fundamental na fase de avaliação de riscos, exigindo uma etapa prévia de identificação dos mesmos.

O diferencial significativo da Matriz de Riscos reside na sua capacidade de proporcionar uma visualização clara das informações relativas aos riscos em questão. Por sua natureza gráfica, torna-se simples discernir quais riscos terão maior ou menor impacto na organização, facilitando a tomada de decisões e a implementação de medidas preventivas para mitigar esses riscos. Além disso, sua facilidade de compreensão e a apresentação clara

e precisa dos dados contribuem para o envolvimento da equipe no processo de gestão de riscos.

Essa matriz é estruturada em duas dimensões: probabilidade e impacto. Através dessa estrutura, é possível calcular e visualizar a classificação do risco, que reflete a avaliação do impacto em relação à probabilidade.

A classificação do risco resulta em sua localização dentro da matriz. As células da matriz são diferenciadas por cores, indicando o nível de classificação do risco, ou seja, o quão crítico ele é.

As faixas coloridas - geralmente são verde, amarela e vermelha - por se tratar de uma abordagem comumente usada em muitos sistemas de gestão de riscos, independentemente da norma ou padrão seguido, conforme figura 3.3.

Probabilidade	Alta	Média	Alta	Alta
	Média	Baixa	Média	Alta
	Baixa	Baixa	Baixa	Média
		Insignificante	Moderado	Catastrófico
Impacto				

Figura 3.3: Matriz de Risco (Elaborada pela autora)

Fonte: Elaborada pela autora (2025)

No que tange a Matriz de Riscos pode-se dizer que:

- Faixa Verde (*Low Risk* / Baixo Risco): Indica um risco que é considerado aceitável e que geralmente não requer ações imediatas, a menos que a situação mude.
- Faixa Amarela (*Medium Risk* / Risco Médio): Indica um risco que pode necessitar de atenção ou ação, dependendo do contexto e dos critérios estabelecidos pela organização.
- Faixa Vermelha (*High Risk* / Alto Risco): Indica um risco que é considerado crítico e que requer ação imediata ou planejamento específico para mitigação.

A probabilidade, representada no eixo vertical na Figura 3.3 é a avaliação de quão provável é a ocorrência de um risco. Em outras palavras, trata-se de determinar a facilidade ou dificuldade com que um evento específico pode ocorrer. A probabilidade é geralmente categorizada em diferentes níveis, como muito baixo, baixo, moderado, alto e muito alto. Esses níveis podem ser quantificados em termos percentuais para facilitar a compreensão [16].

Essas categorias fornecem uma estrutura clara para avaliar a probabilidade de ocorrência de um determinado evento ou risco, permitindo uma análise mais precisa e eficaz no processo de gerenciamento de riscos.

O impacto, representado no eixo horizontal na figura 3.3, diz respeito às consequências decorrentes da ocorrência do risco. Em outras palavras, trata-se de avaliar quais serão os prejuízos ou danos causados caso o evento em questão se concretize. Esse impacto pode ser negativo, como prejuízos financeiros, perda de clientes, danos a equipamentos, entre outros; ou ainda positivo, incluindo novas oportunidades de negócio, adoção de tecnologias inovadoras, redução de taxas ou impostos, entre outros benefícios [16].

Assim como a probabilidade, o impacto também é categorizado em diferentes níveis, tais como muito baixo, baixo, moderado, alto e muito alto. Essa classificação em níveis permite uma análise mais precisa das possíveis consequências do risco, facilitando o processo de tomada de decisão e o desenvolvimento de estratégias de mitigação.

É importante mencionar que embora a norma ISO 31000:2018 não forneça uma definição específica de faixas, de probabilidade e impacto, ela reconhece a importância desses conceitos no processo de gestão de riscos e fornece orientações sobre como avaliar e abordar esses aspectos de maneira eficaz. Com isso, tanto os níveis de faixas, probabilidade e impacto são usadas como ferramentas visuais para facilitar a compreensão, comunicação e tomada de decisão em relação aos riscos identificados. A definição exata do que constitui os níveis desses itens dependerá da política, critérios e contexto específicos de cada organização [16].

Com isso, muitas organizações usam essa abordagem em conjunto com uma matriz de risco, onde a probabilidade de ocorrência e o impacto são avaliados para determinar a faixa de cor correspondente ao risco.

A avaliação dos riscos fornece subsídios para a tomada de decisão, não se constituindo em fator determinante para eventual tratamento do risco. Ou seja, cabe ao gestor, diante da lista de riscos ordenados por nível de risco, decidir quais merecem ações mitigadoras.

Tratamento dos riscos - Compreende o planejamento e a realização de ações para modificar o nível de risco. O nível de risco pode ser modificado por meio de medidas de resposta ao risco que mitiguem, transfiram ou evitem esses riscos. O tratamento dos riscos deve seguir os seguintes passos descritos na figura 3.3.

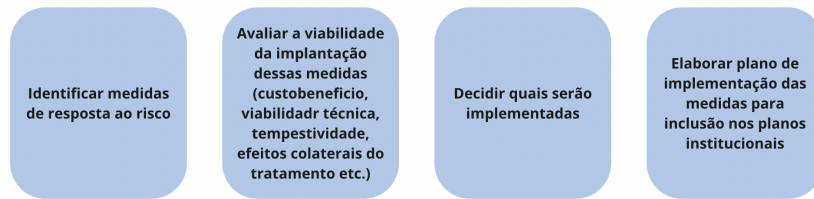


Figura 3.4: Etapas de Identificação dos Riscos (ISO 31000:2018)

Fonte: ISO 31000:2018 [20]

Importante mencionar que mesmo que não seja possível extinguir os riscos organizacionais por completo, existe a possibilidade de mitigá-los de alguma forma através da aplicação dos controles, conforme mostra na Figura 3.3. Dessa forma, entende-se que um controle não aplicado pode ocasionar brechas ou vulnerabilidades que poderão ser exploradas por determinada ameaça, causando então um impacto negativo ao negócio. O controle pode ser qualquer processo, política, dispositivo ou ação a ser tomada, com o objetivo de modificar o risco.

Monitoramento e Análise Crítica envolve o monitoramento e a avaliação contínua do desempenho e da situação dos elementos da gestão de riscos. Isso pode incluir política, atividades, riscos, planos de tratamento, controles, entre outros tópicos relevantes. A avaliação de riscos enfatiza o contexto e outros fatores que tendem a variar com o tempo, e que podem modificar ou até invalidar a própria avaliação. É vital identificar claramente esses fatores para garantir um monitoramento e uma revisão sistemática, permitindo atualizações na avaliação de riscos conforme necessário. Além disso, é essencial identificar e coletar os dados que serão monitorados para aprimorar o processo de avaliação. A efetividade dos controles deve ser constantemente observada e registrada, fornecendo informações valiosas para futuras análises de risco. Por fim, as responsabilidades pela elaboração e revisão das evidências e documentações devem ser claramente estabelecidas [16].

Melhoria Contínua compreende o aperfeiçoamento ou ajuste de aspectos da gestão de riscos avaliados no monitoramento. Ela pode ser entendida em duas dimensões: Uma relativa ao próprio Sistema de Gestão de Riscos da organização e outra relacionada aos resultados do monitoramento sobre a efetividade do tratamento do risco, a cargo dos gestores de riscos. Para que haja melhoria contínua é necessário primeiramente que se tenha instrumentos e/ou métodos que seja possível a identificação de falhas nos processos, que possa ter grande potencial em tornar-se risco para a empresa a organização [16].

Neste estudo será usada a técnica de *Brainstorming* e aplicação de formulário para identificação dos riscos nas atividades da fase de eliciação de requisitos no processos de desenvolvimento de software.

A integração da gestão de riscos no desenvolvimento de software, especialmente na fase de elicitação de requisitos, é uma estratégia essencial para garantir o sucesso e a sustentabilidade dos projetos de software. Esta abordagem abrangente combina práticas de gestão de riscos com metodologias de desenvolvimento de software para identificar, analisar e tratar eficazmente os riscos durante a fase inicial de definição dos requisitos do projeto.

3.4 Desenvolvimento de *Software*

O desenvolvimento de *software* constitui uma disciplina da engenharia que abrange todos os aspectos relacionados à produção de sistemas computacionais, desde as etapas iniciais de especificação até sua manutenção em operação. Trata-se de um campo que defende abordagens sistemáticas, organizadas e apoiadas por ferramentas e técnicas adequadas, com o propósito de assegurar a entrega de soluções de alta qualidade que atendam às necessidades dos usuários e aos objetivos do negócio.

Mais do que programação, o desenvolvimento de *software* compreende um conjunto estruturado de atividades que incluem a especificação de requisitos — frequentemente considerada a etapa mais complexa —, a implementação, os testes, os processos de validação e verificação, bem como a evolução contínua do sistema ao longo de seu ciclo de vida. Além dos aspectos técnicos, é fundamental considerar fatores organizacionais, estratégicos e de negócio, sobretudo na fase de elicitação de requisitos, por se tratar de um momento decisivo para o sucesso do projeto.

Ao longo das últimas décadas, o desenvolvimento de *software* passou por significativa evolução, marcada por transformações metodológicas, tecnológicas e conceituais.

Na Década de 1940 e 1950: Os Primórdios Com os primeiros computadores, como o ENIAC, o "*software*" encontrava-se essencialmente incorporado ao *hardware*. A programação era realizada diretamente em linguagem de máquina ou em linguagens de baixo nível, como o *Assembly*.

Na Década de 1960: Linguagens de Alto Nível e Modelo Cascata Surgiram as primeiras linguagens de alto nível, ampliando a produtividade e a abstração no desenvolvimento. Nesse período, consolidou-se o modelo Cascata (*Waterfall*), considerado o primeiro modelo formal de processo de desenvolvimento de *software*, caracterizado por uma abordagem linear e sequencial.

Na Década de 1970: Expansão das Linguagens e Metodologias Destacam-se a criação da linguagem C e o desenvolvimento do sistema operacional UNIX, demonstrando a flexibilidade e o poder dessas tecnologias. Surgiram também modelos estruturados de

ciclo de vida e o modelo relacional de bancos de dados, que revolucionou a forma de armazenamento e manipulação de dados.

Na Década de 1980: Era Pessoal e Orientação a Objetos Com a popularização dos computadores pessoais, houve uma expansão significativa do desenvolvimento de *software*. Linguagens orientadas a objetos, como C++ e *Smalltalk*, ganharam destaque, assim como metodologias voltadas à orientação a objetos, a exemplo do RUP (*Rational Unified Process*).

Na Década de 1990: Internet e Revolução dos *Softwares* A popularização da Internet impulsionou o desenvolvimento de aplicações web. A linguagem Java foi introduzida com a proposta de portabilidade (“escreva uma vez, execute em qualquer lugar”). No início dos anos 2000, com a publicação do Manifesto Ágil (2001), consolidou-se o Desenvolvimento Ágil, enfatizando colaboração com o cliente e adaptação a mudanças.

Na Década de 2000: Mobilidade, Nuvem e DevOps O surgimento dos *smartphones* impulsionou o desenvolvimento de aplicações móveis. A *cloud computing*, por meio de serviços como *AWS*, *Google Cloud* e *Azure*, permitiu a implementação e escalabilidade de sistemas em nuvem. Consolidou-se também a cultura DevOps, integrando desenvolvimento e operações com foco em integração e entrega contínuas.

Na Década de 2010: Inteligência Artificial e Microserviços A evolução da Inteligência Artificial e do *Machine Learning*, impulsionada por bibliotecas como TensorFlow e PyTorch, ampliou significativamente as possibilidades de aplicação de sistemas inteligentes. Arquiteturas baseadas em microserviços tornaram-se amplamente adotadas, favorecendo escalabilidade e manutenção modular.

Tradicionalmente, o desenvolvimento de *software* é estruturado em etapas distintas para facilitar o planejamento, o projeto, a implementação, os testes e a manutenção. Nesse contexto, a engenharia de requisitos destaca-se como uma das fases mais críticas e complexas, pois é nesse momento que os requisitos são identificados, elicitados, analisados e documentados, servindo de base para todas as etapas subsequentes. Falhas nessa fase tendem a gerar retrabalho, aumento de custos e atrasos no cronograma.

Diversos desafios são associados à elicitação de requisitos. A incompletude é recorrente, uma vez que stakeholders nem sempre conseguem expressar plenamente suas necessidades (Sommerville, 2011). A ambiguidade também representa risco significativo, pois descrições imprecisas podem gerar múltiplas interpretações e falhas na implementação (Pressman, 2014).

A mudança frequente de requisitos constitui outra característica comum, decorrente da evolução do contexto organizacional ou do surgimento de novos entendimentos ao longo do projeto (Wiegers, 2003). Conflitos entre stakeholders, motivados por interesses e prioridades divergentes, podem intensificar problemas de comunicação, especialmente

entre equipes técnicas e não técnicas (Pohl, 2010).

A priorização dos requisitos também apresenta elevada complexidade, sobretudo em ambientes com restrições de recursos ou múltiplos interesses envolvidos (Leffingwell, 2011). Soma-se a isso o desafio do conhecimento tácito, quando determinadas informações permanecem implícitas por serem consideradas “óbvias” pelos stakeholders, mas não necessariamente pela equipe de desenvolvimento (Dardenne et al., 1993) [21].

Diante desse cenário, a engenharia de requisitos consolida-se como uma área em constante evolução, com múltiplas técnicas e abordagens propostas para mitigar tais desafios [22]. Entretanto, devido ao seu forte componente humano, social e organizacional, é improvável que uma única técnica seja suficiente para todos os contextos. Assim, torna-se essencial a adoção de abordagens flexíveis e adequadas às particularidades de cada projeto, organização e conjunto de stakeholders envolvidos [22].

3.5 Engenharia de Requisitos

A Engenharia de Requisitos é um processo essencial no desenvolvimento de software, pois envolve a descoberta, análise, documentação e gerenciamento dos requisitos do sistema. De acordo com Sommerville, esse processo é fundamental para garantir que as necessidades dos *stakeholders* sejam corretamente compreendidas e atendidas ao longo do ciclo de vida do software [23].

É essencial garantir a comunicação eficaz entre todas as partes envolvidas. Uma vez concluída, as informações são documentadas e servem de base para as próximas etapas do projeto. Segundo Canedo, a elicitação de requisitos é a fase inicial onde os requisitos do *software* são coletados dos *stakeholders*. O objetivo é entender e documentar as necessidades e expectativas do usuário final e de outras partes interessadas [24]. Atividades como *workshops*, *brainstorming*, análise de documentos e observação são frequentemente utilizadas. A comunicação eficaz é crucial nesta fase para garantir que todas as necessidades e problemas sejam compreendidos corretamente [25].

A elicitação de requisitos pode ser impactada por diversos fatores, incluindo a complexidade dos sistemas e a falta de padronização no processo de coleta de informações. A modelagem de processos de negócio tem se mostrado uma abordagem eficiente para auxiliar na elicitação de requisitos em sistemas complexos, pois permite uma visão mais clara dos fluxos e das interações entre os usuários e o sistema [25]. Além disso, metodologias como o *Design Thinking*, *Lean Inception* têm sido empregadas para otimizar a elicitação de requisitos, permitindo uma maior interação entre os desenvolvedores e os usuários finais.[26].

O processo de engenharia de requisitos em quatro atividades principais [27]:

- Elicitação de Requisitos: Consiste na coleta de informações junto aos stakeholders para entender suas necessidades e expectativas. Pode envolver entrevistas, questionários e análise de documentos.
- Especificação de Requisitos: Transformação das informações coletadas em documentos formais, como Requisitos Funcionais e Não Funcionais.
- Validação de Requisitos: Verificação da consistência, completude e viabilidade dos requisitos para evitar falhas no projeto. Técnicas como revisões e prototipação podem ser usadas.
- Gerenciamento de Requisitos: Controle das mudanças e rastreamento dos requisitos ao longo do ciclo de vida do sistema.[28]

Uma Engenharia de Requisitos adequadamente aplicada contribui para a redução de custos, mitigação de retrabalho e assegura que os sistemas desenvolvidos atendam às necessidades dos usuários finais [20]. Ademais, constitui elemento central na construção de sistemas robustos e alinhados aos objetivos organizacionais, ao fornecer fundamentos consistentes para a definição, análise e validação das necessidades do negócio, demonstrando que:

- A influência da Engenharia de Requisitos na qualidade do software: Definir corretamente os requisitos reduz falhas e melhora a satisfação do usuário.
- O impacto da modelagem de requisitos: Técnicas como diagramas UML e casos de uso auxiliam na clareza e organização dos requisitos.
- A necessidade de ferramentas para gerenciar requisitos: Uso de *softwares como Jira, Redmine e IBM Rational RequisitePro* ajuda no controle e rastreabilidade dos requisitos [25].

Importante mencionar que uma boa comunicação entre equipes e o uso de metodologias ágeis contribuem para um processo eficiente [29].

Em uma Abordagem Profissional a Engenharia de Requisitos é vista como um processo composto pelas seguintes etapas:

- **Iniciação:** Identificação das necessidades do cliente.
- **Modelagem:** Representação dos requisitos por meio de diagramas e especificações.
- **Análise:** Identificação de conflitos, redundâncias e inconsistências nos requisitos.
- **Revisão:** Discussão com os *stakeholders* para refinamento dos requisitos.

Importante destacar ainda que os requisitos podem ser funcionais (requisitos que definem o que o sistema deve fazer) e não funcionais (requisitos que definem restrições e qualidades desejáveis, como desempenho e segurança). [30]

A fase de elicitação de requisitos além de ser realizado por profissionais totalmente qualificados, ou seja, com conhecimento na área de análise de sistemas, podem ser impactados em diversos fatores, incluindo a complexidade dos sistemas e a falta de padronização no processo de coleta de informações. A modelagem de processos de negócio tem se mostrado uma abordagem eficiente para auxiliar na elicitação de requisitos em sistemas complexos, pois permite uma visão mais clara dos fluxos e das interações entre os usuários e o sistema [25].

Além disso, metodologias como o *Lean Inception e Design Thinking* têm sido empregadas para otimizar a elicitação de requisitos, permitindo uma maior interação entre os desenvolvedores e os usuários finais. A imersão nos processos dos *stakeholders* auxilia na formulação de soluções mais alinhadas com as reais necessidades do usuário. Esses métodos tem sido utilizados como uma técnica de elicitação de requisitos e imersão nas áreas de processo, o que aproxima o cliente da equipe do projeto de software e possibilita a criação de projetos melhores com a análise dos requisitos mais assertiva [29].

A análise de requisitos é uma etapa crítica no desenvolvimento de software, realizada após a elicitação dos requisitos. Envolve revisão detalhada dos requisitos para identificar inconsistências, ambiguidades e lacunas. Os analistas trabalham com os *stakeholders* para garantir que os requisitos sejam completos, claros e verificáveis, utilizando ferramentas e métodos. Os requisitos funcionais e não funcionais são priorizados e documentados em um documento formal, orientando o restante do processo de desenvolvimento. A sabedoria coletiva dos membros da equipe pode desempenhar um papel fundamental na análise de requisitos. Há uma relação direta entre a colaboração dentro das equipes de desenvolvimento e o sucesso do projeto, pois a experiência e o conhecimento compartilhado entre os membros contribuem para a detecção precoce de falhas e inconsistências nos requisitos.

Em suma, a análise de requisitos é uma fase fundamental na Engenharia de Software, dedicada a compreender e refinar os requisitos coletados, assegurando que o sistema desenvolvido atenda às necessidades dos *stakeholders* de forma eficaz e eficiente. Sommerville destaca a importância dessa etapa e descrevem suas atividades principais:

Classificação e Organização dos Requisitos: Após a elicitação, os requisitos são categorizados em grupos coerentes, como funcionais e não funcionais. Essa organização facilita a análise subsequente e a identificação de áreas que necessitam de maior esclarecimento.

Prioritização dos Requisitos: Estabelecer uma hierarquia de importância entre os requisitos é essencial para orientar o planejamento e a alocação de recursos durante o desen-

volvimento. Isso garante que as funcionalidades críticas sejam implementadas primeiro, alinhando o produto às expectativas dos stakeholders.

Detecção e Resolução de Conflitos: Durante a análise, podem surgir conflitos entre requisitos, especialmente quando diferentes stakeholders possuem expectativas divergentes. Identificar e resolver esses conflitos por meio de negociações é crucial para alcançar um consenso que satisfaça todas as partes envolvidas.

Análise de Viabilidade: Avaliar a possibilidade de implementação dos requisitos considerando aspectos negociais, técnicos e financeiros o que assegura que os requisitos definidos são realistas e alcançáveis dentro das restrições do projeto.

Modelagem dos Requisitos: Utilizar representações gráficas ou textuais, como diagramas de casos de uso, diagramas de classes ou modelos de processos de negócio, ajuda a ilustrar e esclarecer os requisitos. Essa modelagem facilita a comunicação entre a equipe técnica e os stakeholders, promovendo um entendimento comum sobre o funcionamento esperado do sistema.

Verificação e Validação dos Requisitos: Antes de prosseguir para a especificação formal, os requisitos analisados são revisados em conjunto com os stakeholders para assegurar sua precisão, completude e alinhamento com as necessidades do negócio. Essa atividade previne equívocos e reduz a probabilidade de retrabalho em fases posteriores do desenvolvimento.

Essas atividades são iterativas e podem ser revisitadas conforme novas informações surgem ou mudanças ocorrem no contexto do projeto. Uma análise de requisitos bem conduzida estabelece uma base sólida para as etapas subsequentes do processo de desenvolvimento de software [31].

A engenharia de requisitos destaca a importância da análise de requisitos e suas atividades na construção de sistemas robustos e alinhados aos objetivos organizacionais. Entre os benefícios dessa prática, ela ressalta a melhoria da qualidade do software, a redução de falhas e o aumento da satisfação do usuário. Além disso, Canedo enfatiza que a adoção de metodologias ágeis, como Scrum e Kanban, aliada a uma comunicação eficiente entre as equipes o que contribui para um processo mais dinâmico e eficaz [32].

Finalizada a etapa de análise de requisitos, a próxima etapa no processo de Engenharia de Requisitos é a Especificação de Requisitos [33].

A especificação de requisitos é uma etapa fundamental no desenvolvimento de software, onde os requisitos do sistema são documentados de forma clara e precisa. Este processo envolve a coleta, análise e definição dos requisitos que o software deve atender para satisfazer as necessidades do cliente ou usuário final. O principal objetivo da especificação é transformar as necessidades e desejos dos *stakeholders* em um conjunto estruturado de requisitos que possa ser compreendido e utilizado por todos os envolvidos no projeto. Isso

inclui desenvolvedores, testadores, gerentes de projeto e os próprios *stakeholders*. Uma especificação bem elaborada facilita a comunicação entre as partes e reduz ambiguidades, servindo como referência durante todo o ciclo de vida do software.

A clareza na documentação dos requisitos é essencial para garantir a rastreabilidade e evitar ambiguidades. A falta de uma especificação precisa pode gerar atrasos e impactos negativos no desenvolvimento do software, comprometendo sua entrega no prazo estabelecido. Além disso, a adoção de práticas automatizadas na especificação de requisitos tem sido explorada em diferentes projetos. A implementação de modelos baseados em BPMN auxilia no refinamento da documentação, permitindo que a equipe de desenvolvimento compreenda melhor as interações do sistema [34].

Documentar os requisitos de forma clara e precisa é essencial para garantir que todas as partes interessadas tenham uma compreensão comum dos objetivos do projeto e do que será entregue. Isso ajuda a evitar mal-entendidos, ambiguidades e conflitos durante o desenvolvimento do software. Um Documento de Requisitos de Software é o resultado típico desse processo de especificação. Este documento descreve detalhadamente os requisitos funcionais e não funcionais do sistema, incluindo suas características, funcionalidades, restrições e interfaces com outros sistemas ou componentes. Ele serve como um guia para os desenvolvedores durante todas as fases do ciclo de vida do desenvolvimento de software, desde a concepção até a implementação e manutenção.

Além disso, a documentação clara e precisa dos requisitos também facilita a comunicação entre as equipes de desenvolvimento, teste e gerenciamento de projeto, garantindo que todos estejam alinhados com as expectativas do cliente e trabalhem na mesma direção para alcançar os objetivos do projeto.

Uma documentação bem elaborada não apenas reduz ambiguidades e retrabalho, mas também serve como base para a validação dos requisitos, permitindo que as partes interessadas revisem, confirmem e, se necessário, ajustem as especificações antes que o desenvolvimento avance. A próxima etapa a ser seguida é a etapa de validação dos requisitos que é um processo essencial para garantir que as necessidades do cliente foram corretamente interpretadas e traduzidas em funcionalidades que agreguem valor ao produto final [34].

A validação de requisitos é uma fase essencial do processo de engenharia de requisitos, cujo objetivo é garantir que os requisitos documentados refletem com precisão as necessidades dos stakeholders e são tecnicamente viáveis para implementação. Essa etapa funciona como uma verificação final antes de prosseguir para o design e desenvolvimento, assegurando que o sistema proposto atenda às expectativas dos usuários e demais partes interessadas.

A validação de requisitos envolve diferentes técnicas e atividades, são elas:

- Revisões de Requisitos – Consistem na análise detalhada dos documentos de requisitos pelos stakeholders, engenheiros de software e analistas de requisitos. Esse processo visa identificar inconsistências, ambiguidade e omissões que possam impactar a implementação do sistema.
- Prototipagem – A criação de protótipos funcionais ou de baixa fidelidade permite que os stakeholders visualizem e interajam com uma versão inicial do sistema, fornecendo *feedback* sobre a adequação dos requisitos antes do desenvolvimento. Essa abordagem reduz retrabalho e melhora a compreensão dos requisitos.
- Testes Iniciais de Viabilidade Técnica – Em alguns casos, são conduzidos testes preliminares para verificar a viabilidade de implementação dos requisitos. Essa atividade é crucial para antecipar desafios técnicos que possam surgir durante o desenvolvimento.
- Revisões com os Stakeholders – Encontros frequentes com as partes interessadas garantem que os requisitos documentados estejam alinhados com as necessidades reais do negócio, reduz riscos de falha na entrega e aumenta a satisfação do cliente.[35]

Há cinco pontos centrais sobre a validação de requisitos, são elas:

Importância da validação precoce: A validação deve ser feita o quanto antes no processo de desenvolvimento, para evitar problemas mais custosos nas fases seguintes (como implementação e testes). Ela destaca que requisitos mal definidos ou não validados corretamente são uma das principais causas de falhas de software.

Validação como parte crítica da qualidade dos requisitos: A validação ajuda a garantir que os requisitos são corretos, consistentes, compreensíveis e verificáveis. Ela relaciona fortemente a validação à qualidade do produto final, defendendo que sem validação adequada, a documentação de requisitos não cumpre seu papel.

Técnicas e práticas utilizadas: Em suas publicações, ela menciona revisões de requisitos (inspeções), prototipagem, simulações e reuniões com *stakeholders* como formas efetivas de validação. Também discute *checklists* e métodos mais formais de revisão para garantir que os critérios de qualidade sejam atendidos.

Participação de *stakeholders*: A validação eficaz envolve interação constante com os *stakeholders*, garantindo que os requisitos refletem de fato as necessidades do usuário.

Ensino e formação: há necessidade de ensinar práticas de validação desde cedo na formação de engenheiros de software, mostrando que isso não é apenas uma fase formal, mas parte do processo iterativo de entendimento do problema [36].

A validação dos requisitos constitui uma etapa essencial no processo de Engenharia de Requisitos, pois tem como objetivo assegurar que os requisitos especificados representem,

de forma fiel, as reais necessidades dos stakeholders e do contexto organizacional. Esse processo contribui diretamente para a qualidade do produto final, ao evitar a implementação de funcionalidades desnecessárias, reduzir erros de interpretação e prevenir falhas que possam comprometer a usabilidade e a aceitação do sistema (Wiegers; Beatty, 2013). Além disso, a validação adequada dos requisitos impacta positivamente o desempenho do projeto, uma vez que possibilita a identificação precoce de inconsistências e lacunas, reduzindo custos e prazos ao minimizar retrabalho e correções realizadas em fases tardias do desenvolvimento [33].

Nesse contexto, o Gerenciamento de Requisitos emerge como uma prática complementar e indispensável à validação, ao assegurar que os requisitos sejam continuamente acompanhados, controlados e atualizados ao longo de todo o ciclo de vida do projeto. O Gerenciamento de Requisitos é responsável por garantir que os requisitos do sistema sejam corretamente identificados, documentados, rastreados e gerenciados frente a mudanças, conflitos e novas demandas. Essa prática é fundamental para manter a coerência entre os objetivos do negócio, as expectativas dos stakeholders e as soluções técnicas implementadas, contribuindo de forma significativa para a qualidade do software e para o sucesso global do projeto.

As principais atividades envolvidas no Gerenciamento de Requisitos incluem:[37]

- Identificação de Requisitos: Consiste em coletar e documentar os requisitos do sistema, envolvendo a interação com *stakeholders* para compreender suas necessidades e expectativas.
- Análise de Requisitos: Refere-se à revisão e compreensão detalhada dos requisitos coletados para identificar inconsistências, ambiguidades e lacunas. Essa análise visa garantir que os requisitos sejam claros, completos e verificáveis.
- Rastreamento de Requisitos: Envolve a criação e manutenção de uma relação entre os requisitos e outros artefatos do projeto, como documentos de design, casos de teste e código-fonte. Isso permite acompanhar as mudanças nos requisitos ao longo do tempo e garantir que sejam implementados corretamente.
- Priorização de Requisitos: Consiste em atribuir prioridades aos requisitos com base em sua importância para o sucesso do projeto e para os objetivos do negócio. Isso ajuda a equipe a focar nos requisitos mais críticos e alocar recursos de forma eficiente.
- Controle de Mudanças: Refere-se ao processo de gerenciar e controlar mudanças nos requisitos ao longo do tempo. Isso inclui avaliar o impacto das mudanças, obter aprovação dos stakeholders e atualizar a documentação de requisitos conforme necessário.

- **Comunicação e Colaboração:** Envolve garantir uma comunicação eficaz entre todas as partes interessadas envolvidas no projeto, incluindo desenvolvedores, testadores, gerentes de projeto e stakeholders. A colaboração entre essas partes é essencial para garantir que os requisitos sejam compreendidos e atendidos adequadamente.

Em suma, o Gerenciamento de Requisitos desempenha um papel crucial no sucesso do projeto de software, garantindo que os requisitos do sistema sejam claramente definidos, compreendidos e implementados de forma eficaz ao longo de todo o ciclo de vida do projeto.

Conforme apresentado na figura 3.5 são apresentadas as fases da Engenharia de requisitos:



Figura 3.5: Engenharia dos Requisitos

Fonte: Elaborada pela autora (2025)

O Gerenciamento de Requisitos constitui uma prática fundamental no desenvolvimento de software, pois é responsável por assegurar que os requisitos do sistema sejam corretamente identificados, documentados, analisados, rastreados e controlados ao longo de todo o ciclo de vida do projeto. Ao integrar essas atividades de forma sistemática, o gerenciamento contribui para a consistência das informações, a rastreabilidade entre requisitos e artefatos do projeto e a correta implementação daquilo que foi solicitado, minimizando riscos, conflitos e desalinhamentos entre as expectativas dos stakeholders e as soluções desenvolvidas [38]. Nesse contexto, a elicitação de requisitos assume papel central, uma vez que representa o ponto de partida para todo o processo de gerenciamento,

sendo responsável por identificar e compreender as reais necessidades que o sistema deve atender.

A elicitação de requisitos pode ser compreendida como a atividade por meio da qual são mobilizados recursos, técnicas e métodos para compreender o problema a ser resolvido e captar as necessidades, expectativas e restrições dos stakeholders. Trata-se de uma atividade fortemente dependente da interação humana, do conhecimento do domínio e da capacidade de comunicação entre os envolvidos. Para sua execução, podem ser adotadas tanto abordagens tradicionais quanto métodos ágeis de desenvolvimento de software, de acordo com as características do projeto, do contexto organizacional e do grau de maturidade das equipes [39].

As metodologias tradicionais, como os modelos *Waterfall* e Espiral, caracterizam-se por uma abordagem sequencial e linear, com forte ênfase em planejamento inicial detalhado, documentação abrangente e processos rigorosos de controle. Essas abordagens são frequentemente classificadas como “metodologias de engenharia”, por se apoiarem em princípios consolidados das engenharias clássicas. Nesse contexto, a elicitação de requisitos ocorre predominantemente nas fases iniciais do projeto, por meio de técnicas como entrevistas, questionários, análise de documentos, estudos de caso, observação e revisão de modelos de negócio. Embora essas práticas favoreçam a formalização e a clareza dos requisitos, apresentam limitações significativas no tratamento de mudanças, uma vez que ajustes realizados após o planejamento inicial tendem a ser mais custosos e complexos [33].

Em resposta a essas limitações, as metodologias ágeis emergiram como uma alternativa às abordagens tradicionais, especialmente em cenários marcados por incerteza, dinamismo e evolução contínua dos requisitos. As abordagens ágeis caracterizam-se por estruturas mais leves, com menor número de artefatos formais, priorizando a colaboração entre os participantes, a comunicação contínua e a adaptação às mudanças. Nesses modelos, o gerenciamento de requisitos ocorre de forma incremental e iterativa, sendo fortemente integrado ao processo de elicitação, que passa a ser contínuo ao longo do desenvolvimento [40].

De modo geral, as metodologias ágeis adotam ciclos curtos de desenvolvimento, nos quais os requisitos são representados por meio de User Stories e critérios de aceitação, permitindo entregas frequentes e a obtenção de feedback constante dos stakeholders. Essa abordagem possibilita o refinamento progressivo dos requisitos, reduzindo riscos associados à incompreensão, à ambiguidade ou à volatilidade das necessidades identificadas [1]. Além disso, ao valorizar a interação humana e a simplificação dos processos, as metodologias ágeis contribuem para uma gestão de requisitos mais flexível e alinhada aos objetivos do negócio, especialmente em projetos com equipes reduzidas, altamente qualificadas ou

em contextos nos quais os requisitos são dinâmicos [41].

Adicionalmente, metodologias ágeis enfatizam a comunicação e são particularmente recomendadas para projetos onde os requisitos são dinâmicos e/ou a equipe é reduzida e altamente qualificada. Entre os exemplos mais conhecidos estão Scrum, Kanban e Extreme Programming (XP).

A Elicitação de requisitos pode ser significativamente aprimorada com a utilização de abordagens inovadoras, como *Lean Inception*, *Design Thinking*. Essa abordagem, amplamente discutida na literatura, se destaca por permitir uma maior imersão no contexto do usuário, facilitando a identificação de requisitos mais alinhados às necessidades reais. A *Lean Inception* também incentiva uma colaboração mais estreita entre os *stakeholders* e os desenvolvedores, possibilitando uma elicitação mais rica e detalhada dos requisitos[42].

Os requisitos em desenvolvimento de software são as descrições detalhadas do que um sistema deve fazer e como deve se comportar. Eles são as expectativas ou condições que um sistema ou aplicativo deve atender ou possuir para ser considerado bem-sucedido [38].

Os requisitos servem para:

- Definir Expectativas: Eles estabelecem as funcionalidades e condições que o sistema deve cumprir.
- Orientar o Desenvolvimento: Servem como um guia para desenvolvedores e designers na construção do software.
- Base para Testes: São usados como critérios para testar e validar o software final.
- Comunicação: Facilitam a comunicação clara entre os stakeholders, incluindo desenvolvedores, clientes e usuários finais.
- Características de Bons Requisitos:
 - Clareza: Devem ser claramente entendidos por todas as partes interessadas.
 - Consistência: Não devem entrar em conflito uns com os outros.
 - Completude: Deve cobrir todas as funcionalidades e condições do sistema.
 - Realismo: Devem ser realizáveis dentro das limitações de tempo, tecnologia e orçamento.
 - Verificabilidade: Devem ser possíveis de testar e validar.

A elicitação de requisitos pode ser potencializada por meio de técnicas como *Personas* e *Journey Maps*, que auxiliam no entendimento das necessidades e comportamentos dos

usuários durante a fase inicial do desenvolvimento de software. Tais técnicas não apenas proporcionam maior clareza no levantamento dos requisitos, mas também melhoram a colaboração entre equipes de desenvolvimento e os usuários, garantindo maior alinhamento com as expectativas dos *stakeholders* [25].

Além disso, um dos desafios enfrentados na engenharia de requisitos, especialmente em ambientes ágeis, é a priorização adequada dos requisitos coletados. Em muitos casos, as equipes de desenvolvimento acabam se concentrando nos requisitos funcionais e deixam de lado os requisitos não funcionais, como privacidade e segurança. Estudos demonstram que esse problema ocorre porque tais requisitos são percebidos como secundários durante o desenvolvimento, o que pode gerar consequências negativas na qualidade final do software [43].

Durante a fase de elicitação, é fundamental que os requisitos sejam identificados, classificados e documentados de forma clara e precisa. A correta compreensão dos diferentes tipos de requisitos contribui diretamente para a mitigação de riscos, evitando retrabalho, ambiguidades e falhas na comunicação entre as partes interessadas.

Os requisitos de um sistema podem ser divididos em duas categorias principais: requisitos funcionais e requisitos não funcionais. Além dessas, outras classificações complementares são adotadas na literatura e na prática da engenharia de software, como requisitos de domínio, de interface, operacionais, de transição e regulatórios[44].

Os requisitos funcionais especificam as funções, comportamentos e serviços que o sistema deve executar. Eles descrevem de maneira direta e objetiva o que o sistema deve fazer, considerando as interações com os usuários, outros sistemas ou dispositivos. São, portanto, a base do entendimento do escopo do sistema e orientam tanto o desenvolvimento quanto os testes.

Requisitos funcionais descrevem as interações entre o sistema e seu ambiente, independentemente de limitações de hardware ou software.” Eles indicam ações específicas que o sistema deve executar em resposta a entradas ou eventos, além de detalhar casos de uso, regras de negócio, fluxos de dados e respostas esperadas [27].

A norma ISO/IEC/IEEE 29148:2018 também define requisitos funcionais como “declarações do que o sistema deve fazer em termos de comportamento ou operação sob certas condições” [45].

São objetivo dos Requisitos Funcionais:

- Delimitar o escopo funcional do sistema.
- Fornecer uma base para especificação técnica e testes de validação.
- Garantir que as necessidades dos usuários sejam corretamente

- traduzidas em funcionalidades.
- Facilitar a comunicação entre desenvolvedores, analistas, usuários e stakeholders.
- Evitar ambiguidade e interpretações subjetivas sobre o que se espera do sistema.

Dessa forma, ao compreender os objetivos dos requisitos negociais, que envolvem alinhar o sistema às metas estratégicas da organização e garantir que o produto atenda aos interesses do negócio, torna-se evidente a necessidade de desdobrá-los em requisitos funcionais específicos. Esses requisitos funcionais atuam como respostas concretas às necessidades identificadas no nível de negócio, traduzindo-as em comportamentos observáveis do sistema, conforme exemplos na figura 3.6.

ID	Descrição do Requisito Funcional
RF01	O sistema deve permitir que o usuário se autentique por meio de login e senha.
RF02	O sistema deve permitir o cadastro de novos clientes com nome, CPF, endereço e telefone.
RF03	O sistema deve emitir relatórios mensais em formato PDF.
RF04	O sistema deve calcular o valor do imposto com base nas alíquotas configuradas por estado.
RF05	O sistema deve enviar notificações por e-mail em caso de falha no processamento de pedidos.

Figura 3.6: Exemplo de descrições de Requisitos Funcionais

Fonte: Elaborada pela autora (2025)

Tendo compreendido que os requisitos funcionais têm como principal objetivo delimitar os comportamentos esperados do sistema frente a determinadas condições e eventos, é essencial considerar suas fontes de origem para garantir uma elicitação completa e precisa.

São as origens da informação que servem de base para identificar e definir os requisitos do sistema. Em outras palavras, são os locais ou entidades de onde os requisitos “emergem”.

Os requisitos funcionais podem ser derivados de diversas fontes, tais como:

- *Stakeholders* (usuários, clientes, patrocinadores);
- Documentos existentes (normas, legislações, manuais);
- Sistemas legados;

- Processos organizacionais já existentes;
- Requisitos de negócio;
- Regras de negócio;
- Políticas de segurança e privacidade;
- Objetivos estratégicos da organização.

As fontes de origem são essenciais para garantir que os requisitos estejam alinhados com as necessidades reais do sistema. No entanto, o conteúdo dessas fontes não se revela de forma automática: é preciso aplicar técnicas apropriadas de elicitação para transformar esse conhecimento disperso em requisitos claros e verificáveis. Por exemplo:

- *Stakeholders* são melhor abordados por meio de entrevistas e *workshops*;
- Sistemas legados podem ser analisados por meio de observação direta e revisão técnica;
- Objetivos de negócio muitas vezes exigem a aplicação de técnicas de modelagem, como BPMN ou casos de uso, histórias de usuário (*User stories*).

Assim, a efetividade da engenharia de requisitos reside na sinergia entre as fontes e as técnicas utilizadas, garantindo uma coleta abrangente, precisa e alinhada com os objetivos estratégicos do projeto.

A identificação correta das fontes de requisitos funcionais é essencial para garantir a completude, consistência e viabilidade das especificações do sistema. No entanto, quando essas fontes não são devidamente consideradas ou exploradas por meio de técnicas apropriadas de elicitação, surgem riscos significativos ao projeto. Durante a fase de elicitação, a má definição de requisitos funcionais pode gerar riscos críticos para o projeto, como:

- Ambiguidade: Descrição vaga leva a interpretações diferentes por parte da equipe;
- Incompletude: Funcionalidades importantes podem ser esquecidas, afetando o produto final;
- Contradições: Conflitos entre requisitos de diferentes stakeholders;
- Escopo inflado (*scope creep*): Inserção de funcionalidades fora do escopo sem avaliação de impacto;
- Impacto na arquitetura e desempenho: Decisões erradas baseadas em funcionalidades mal compreendidas;

- A identificação precoce e precisa desses requisitos é fundamental para evitar retrabalho, aumento de custo e atrasos no cronograma.

A documentação de requisitos funcionais é essencial para o desenvolvimento de software, pois sua clareza, completude e rastreabilidade reduzem riscos comuns, como ambiguidades, interpretações equivocadas, retrabalho e falhas de implementação. Uma boa especificação atua como um contrato entre *stakeholders* e equipe técnica, assegurando entendimento uniforme sobre o comportamento esperado do sistema. O uso de práticas estruturadas — como modelos padronizados, versionamento e rastreabilidade — diminui riscos técnicos e gerenciais e facilita etapas posteriores, como testes, validação e manutenção. Esse entendimento converge com a ISO/IEC 25010:2011, que relaciona a qualidade do produto à precisão e consistência da especificação. Assim, a adequada documentação dos requisitos funcionais não apenas orienta o desenvolvimento, mas constitui um elemento indispensável para a avaliação sistemática da qualidade do software. [46].

Os requisitos funcionais constituem a espinha dorsal de um sistema de software, uma vez que descrevem seu comportamento esperado e suas funcionalidades essenciais. A elicitação adequada e a documentação estruturada desses requisitos são fatores críticos para a prevenção de riscos ao longo do desenvolvimento, além de desempenharem papel fundamental na satisfação do cliente e na aderência do produto às necessidades do negócio. Os requisitos funcionais devem ser descritos de forma precisa, clara e verificável, pois sua má definição é uma das principais causas de falhas em projetos de software. Quando bem especificados, esses requisitos orientam todas as fases do projeto, contribuindo diretamente para a qualidade, a rastreabilidade e a validação eficaz do sistema [27].

Contudo, garantir o sucesso de um sistema de software vai além de implementar suas funcionalidades básicas. É igualmente essencial atender a aspectos relacionados à qualidade, desempenho, segurança, usabilidade e conformidade, os quais são abordados por meio dos requisitos não funcionais. Esses requisitos não definem o que o sistema deve fazer, mas sim como ele deve se comportar diante de determinadas condições operacionais. A seguir, será apresentado um aprofundamento sobre os requisitos não funcionais, suas categorias, sua relação com a qualidade do produto e sua importância estratégica no ciclo de vida do software. Requisitos Não Funcionais (RNFs) são essenciais para garantir a qualidade de um sistema de software, abrangendo aspectos como desempenho, segurança, usabilidade e manutenibilidade [47].

Eles descrevem propriedades e restrições de qualidade que o sistema deve atender, sem estarem diretamente ligados às funcionalidades específicas que o sistema realiza. Eles determinam como o sistema deve se comportar, sob quais condições deve operar e quais padrões de qualidade deve seguir, sendo essenciais para garantir a eficácia, eficiência,

usabilidade, confiabilidade, segurança, desempenho e manutenção do software. Diante disso, podemos elencar como objetivos dos requisitos não funcionais:

- Estabelecer parâmetros de qualidade para o produto final;
- Definir limites de operação para as funcionalidades;
- Apoiar decisões de arquitetura de software e infraestrutura;
- Garantir a satisfação dos usuários e conformidade regulatória;
- Mitigar riscos técnicos e operacionais.

Esses requisitos impactam diretamente a experiência do usuário, a viabilidade técnica da solução e a capacidade de manutenção futura. O objetivos dos requisitos não funcionais se materializam por meio de suas categorias, que funcionam como dimensões práticas da qualidade e fornecem critérios objetivos para validação, rastreabilidade e melhoria contínua do software. De acordo com a ISO/IEC 25010:2011, os requisitos não funcionais podem ser classificados em oito categorias principais, que compõem o modelo de qualidade de produto [46].

- Funcionalidade (Adequação Funcional) - Capacidade do sistema de fornecer funções que satisfaçam as necessidades explícitas e implícitas dos usuários. Engloba correção funcional, completude e adequação;
- Eficiência de desempenho - Grau com que o sistema responde de forma adequada, em termos de tempo de resposta, uso de recursos e capacidade;
- Compatibilidade - Capacidade do sistema de funcionar em diferentes ambientes, compartilhando informações com outros sistemas ou componentes. Inclui interoperabilidade e coexistência;
- Usabilidade - Facilidade com que o sistema pode ser utilizado, aprendido, operado e atrativo para o usuário. Engloba acessibilidade, aprendibilidade e satisfação;
- Confiabilidade - Capacidade do sistema de manter seu nível de desempenho sob condições estabelecidas por um tempo determinado. Inclui disponibilidade, tolerância a falhas e maturidade;
- Segurança - Capacidade do sistema de proteger informações e dados contra acesso não autorizado. Inclui confidencialidade, integridade e autenticação;

- Manutenibilidade - Facilidade de modificar o sistema, incluindo correções, melhorias ou adaptação a novos ambientes. Engloba modularidade, reusabilidade e analisabilidade;
- Portabilidade - Facilidade com que o sistema pode ser transferido de um ambiente para outro. Inclui adaptabilidade, instalabilidade e substituibilidade.

Essas categorias representam áreas críticas que devem ser avaliadas e especificadas com clareza durante o processo de engenharia de requisitos. Assim, os objetivos dos requisitos não funcionais se materializam por meio de suas categorias, que funcionam como dimensões práticas da qualidade e fornecem critérios objetivos para validação, rastreabilidade e melhoria contínua do software. Para facilitar a compreensão e aplicação desses conceitos, a seguir são apresentadas na figura 3.7 as classificações mais comuns dos requisitos não funcionais, acompanhadas de exemplos práticos que ilustram sua importância no contexto do desenvolvimento de sistemas:

Categoria	Descrição	Exemplo
Funcionalidade (Adequação Funcional)	Capacidade do sistema de fornecer funções que satisfaçam as necessidades explícitas e implícitas dos usuários. Engloba correção funcional, completude e adequação.	O sistema deve permitir que os usuários emitam relatórios financeiros mensais com base nos dados cadastrados, sem perda de informações e com o agrupamento por centro de custo. Como esse requisito atende à "Adequação Funcional": <ul style="list-style-type: none"> • Correção funcional: o relatório apresenta os dados corretos. • Completude: todas as opções e filtros necessários estão disponíveis. • Adequação: a funcionalidade é útil e necessária para a tomada de decisões financeiras.
Eficiência de desempenho	Tempo de resposta, taxa de transferência, tempo de processamento.	O sistema deve responder em até 2 segundos para 95% das requisições.
Compatibilidade	Interoperabilidade com outros sistemas, versões e tecnologias.	O sistema deve integrar-se com a API da Receita Federal.
Usabilidade	Facilidade de aprendizado, uso, acessibilidade, satisfação do usuário.	Um novo usuário deve conseguir emitir um relatório em menos de 5 minutos.
Confiabilidade	Probabilidade de funcionamento correto sob condições definidas.	O sistema deve manter 99,9% de uptime mensal.
Segurança	Autenticação, controle de acesso, criptografia, integridade de dados.	O acesso ao sistema deve usar autenticação de dois fatores.
Manutenibilidade	Facilidade de modificar, corrigir e melhorar o sistema.	O código deve seguir padrão de nomenclatura e conter testes automatizados.
Portabilidade	Capacidade de adaptação a diferentes ambientes e plataformas.	O sistema deve funcionar em Windows, Linux e macOS.

Figura 3.7: Categorias RNF

Fonte: Elaborada pela autora (2025)

No entanto, a negligência na identificação e especificação adequadas desses requisitos segundo Canedo et al. (2016) pode acarretar diversos riscos, tais como [47].

- Ambiguidade: descrições vagas que levam a interpretações divergentes;
- Incompletude: omissão de requisitos críticos que comprometem a funcionalidade do sistema;

- Conflitos: incompatibilidades entre requisitos que dificultam a implementação;
- Dificuldade de verificação: requisitos não mensuráveis que impedem a validação eficaz.

Para mitigar esses riscos, diversas técnicas de levantamento e validação de requisitos não funcionais podem ser aplicadas de forma estratégica. Por exemplo:

- Entrevistas e workshops com *stakeholders*: para esclarecer e detalhar requisitos;
- Prototipação: para validar requisitos de usabilidade e desempenho;
- Modelagem de requisitos: utilizando padrões como UML para identificar e resolver conflitos;
- Definição de critérios de aceitação claros: para assegurar a verificabilidade dos requisitos.

A eficácia das técnicas de engenharia de requisitos depende de boas práticas de documentação que assegurem clareza, rastreabilidade e consistência ao longo do ciclo de vida do software. A falta dessa estrutura compromete a mitigação de riscos, dificulta a comunicação entre stakeholders e fragiliza a validação e verificação. Como destaca Sommerville (2011), a documentação dos requisitos não funcionais deve ser precisa, livre de ambiguidades, organizada por categorias e vinculada a critérios de aceitação mensuráveis[27]. Além disso, O uso de versionamento, templates padronizados e rastreamento entre requisitos e funcionalidades favorece o monitoramento e a validação contínua dos requisitos não funcionais. Assim, a mitigação de riscos associados aos RNFs depende tanto de boas técnicas de elicitação e validação quanto de uma documentação de alta qualidade, que sirva como referência formal e meio de comunicação entre as partes envolvidas[47].

Embora os requisitos não funcionais tratem de atributos de qualidade, como desempenho, segurança, usabilidade e manutenibilidade, muitos desses aspectos são diretamente influenciados por fatores próprios do domínio de aplicação, ou seja, pelos requisitos de domínio.

Os Requisitos de domínio são aqueles derivados diretamente do contexto de aplicação, refletindo regras de negócio, normas legais e conhecimentos específicos de setores como saúde, finanças ou educação. Eles garantem que o sistema esteja alinhado aos processos e restrições do domínio-alvo. Como observa Sommerville (2011), esses requisitos podem ser funcionais ou não funcionais e costumam ser difíceis de elicitar, pois exigem compreensão profunda do ambiente organizacional, regulatório e operacional em que o sistema será utilizado [27].

A complexidade dos requisitos de domínio decorre de sua dependência de conhecimento tácito, forte ligação com regras de negócio e influência direta na arquitetura, segurança, desempenho e conformidade do sistema. Envolve terminologias específicas, fluxos operacionais complexos e restrições particulares, demandando do analista compreensão contextual e comunicação interdisciplinar. Esses requisitos surgem do próprio domínio de aplicação e podem ser classificados em três tipos principais [27]:

- Restrições Não Funcionais de Domínio (*Domain Non-functional Requirements*);
- Requisitos de Dados de Domínio (*Domain Data Requirements*)
- Requisitos Funcionais de Domínio (*Domain Functional Requirements*).

As restrições não funcionais de domínio abrangem exigências de desempenho, segurança, interoperabilidade ou qualidade impostas pelo ambiente operacional, pela legislação ou por práticas do setor. Já os requisitos de dados de domínio tratam da estrutura e semântica de dados específicos, como vocabulário técnico, classificações e formatos obrigatórios. Os requisitos funcionais de domínio, por sua vez, descrevem comportamentos necessários para que o sistema opere segundo regras e condições particulares do negócio.

Os requisitos de domínio surgem de contextos específicos, como os setores bancário, jurídico, educacional ou de saúde, e refletem regras, processos e restrições próprias de cada área. Por dependerem de conhecimento especializado e influenciarem tanto requisitos funcionais quanto não funcionais — inclusive aspectos legais, normativos, de desempenho e decisões arquiteturais — são difíceis de identificar e exigem documentação precisa e análise contextualizada. Sua validação geralmente requer especialistas, pois impõem restrições críticas de segurança, desempenho e conformidade. Assim, compreender seus tipos e características permite ao engenheiro de requisitos antecipar desafios de elicitação, validação e priorização, especialmente em projetos de elevada complexidade técnica ou regulatória.

Na figura 3.8. seguem alguns tipos comuns de requisitos de domínio:

Tipo de Requisitos de Domínio	Descrição do Requisito
Regras de negócio específicas	Lógicas internas que regem processos decisórios (ex: cálculo de aposentadoria)
Restrições legais e normativas	Requisitos impostos por legislações ou normas técnicas (ex: LGPD, Anvisa, ISO)
Integração com sistemas externos	Exigência de interoperabilidade com sistemas de terceiros (ex: Receita Federal, SERPRO)
Protocolos de operação	Procedimentos padronizados do domínio (ex: triagem hospitalar, emissão de laudos técnicos)
Terminologia e estruturas de dados do domínio	Vocabulário técnico e formatos específicos usados (ex: CNAE, CID-10, NCM, etc.)

Figura 3.8: Tipos de requisitos de Domínio

Fonte: Elaborada pela autora (2025)

Compreender os tipos de requisitos de domínio é fundamental para antecipar e mitigar os riscos associados ao desenvolvimento de sistemas inseridos em contextos regulatórios, operacionais ou técnicos específicos. [37] [45].

Diante disso, torna-se imprescindível a adoção de estratégias eficazes de mitigação desde as etapas iniciais da engenharia de requisitos. Para enfrentar ambiguidade e incompletude, recomenda-se o uso de entrevistas estruturadas com especialistas de domínio, análise de documentos normativos e modelagem de processos de negócio (como BPMN), que permitem explicitar regras operacionais frequentemente implícitas. Já os riscos associados a restrições técnicas e interoperabilidade podem ser mitigados por meio da prototipação funcional, simulações operacionais e revisões técnicas colaborativas com equipes multidisciplinares. Quanto aos requisitos de dados, a adoção de glossários de termos, dicionários de dados padronizados e validação com *stakeholders* especializados garante maior consistência terminológica e semântica. Além disso, a aplicação de boas práticas de documentação, como versionamento, rastreabilidade e categorização estruturada dos requisitos, fortalece a comunicação entre as partes interessadas e facilita o monitoramento contínuo dos riscos. Assim, a mitigação efetiva dos riscos associados aos requisitos de domínio requer não apenas o uso de técnicas adequadas de elicitación e validação, mas também uma abordagem sistemática, iterativa e colaborativa, que una o conhecimento técnico da engenharia de requisitos à expertise contextual dos especialistas de domínio [46] [47] [48].

Encerrada essa etapa, o próximo ponto a ser tratado refere-se aos requisitos de interface, cuja importância reside na definição precisa das interações entre o sistema e seus elementos externos, como usuários, outros sistemas e dispositivos.

Os requisitos de interface são essenciais na definição de como o sistema de *software* se comunica com o mundo externo, abrangendo interações com usuários, outros sistemas e dispositivos físicos. Como destacado por Sommerville (2011), sua especificação precisa é crucial para garantir a integridade da operação e a experiência do usuário final.

A interface de usuário trata de como as pessoas interagem com o sistema. Para garantir que essa interação ocorra de forma eficiente e inclusiva, é necessário que os requisitos de usuário de interface contemplem aspectos como usabilidade, acessibilidade e navegabilidade. Por exemplo, a adoção de padrões como WCAG 2.1 (*Web Content Accessibility Guidelines* – Diretrizes de Acessibilidade para Conteúdo Web) que é um padrão internacional criado pelo W3C (*World Wide Web Consortium*) por meio da Iniciativa de Acessibilidade na Web (WAI). Ele fornece recomendações para tornar o conteúdo digital mais acessível a pessoas com deficiência, incluindo deficiências visuais, auditivas, motoras, cognitivas, entre outras. Esses requisitos de usuário de interface devem ser acompanhados de mockups, protótipos e fluxos de navegação, que auxiliam stakeholders a visualizar o

produto e validá-lo antecipadamente.

Requisitos de interface definem restrições e especificações de entrada/saída que asseguram a comunicação entre o sistema em desenvolvimento e seu ambiente externo. Eles são classificados como requisitos não funcionais, mas podem conter elementos funcionais quando especificam comportamentos necessários na interação [27].

Além da interface com o usuário, muitos sistemas dependem de integrações com serviços terceiros — como APIs de pagamento, bancos de dados externos ou sistemas legados. Nesses casos, os requisitos devem descrever protocolos de comunicação, formatos de dados e mecanismos de autenticação (e.g., *REST*, *JSON*, *OAuth2*), que garantem a interoperabilidade do sistema. Já os sistemas que interagem com dispositivos físicos, como sensores, leitores biométricos ou impressoras, exigem requisitos específicos que definam protocolos de comunicação e padrões de compatibilidade de hardware. A ausência desses requisitos pode comprometer funcionalidades essenciais do sistema.

A correta identificação e especificação dos requisitos de interface são fundamentais para assegurar que o sistema atue de forma eficiente dentro do ecossistema tecnológico e humano no qual será inserido. De acordo com a ISO/IEC 25010:2011, os requisitos de interface impactam diretamente atributos de qualidade como usabilidade, compatibilidade, portabilidade e segurança [46]. Além disso, a documentação detalhada dessas interfaces deve incluir não apenas os fluxos de interação, mas também os protocolos de comunicação, os padrões de dados (como JSON, XML), os requisitos de performance e as regras de autenticação/autorização. Tais especificações permitem que o sistema atinja níveis mais elevados de qualidade e conformidade normativa.

A norma ISO/IEC 25010:2011 estabelece também um modelo de qualidade que serve como referência para a avaliação de sistemas de software. Ela enfatiza que os requisitos de interface devem ser projetados com foco nos seguintes atributos:

- Usabilidade - Facilidade de uso e aprendizado pelo usuário;
- Compatibilidade - Capacidade do sistema de funcionar em conjunto com outros sistemas;
- Portabilidade - Facilidade com que o sistema pode ser transferido para outro ambiente;
- Segurança - Proteção contra acessos não autorizados nas interfaces.

Esses atributos reforçam a importância de tratar os requisitos de interface não apenas como uma questão técnica, mas como elementos centrais para garantir a qualidade, conformidade normativa e experiência do usuário [46]. A correta especificação dos requisitos de interface é determinante para o sucesso de um sistema de software. Quando mal

definidos ou omitidos, esses requisitos geram riscos significativos que impactam diretamente a qualidade, a funcionalidade e a sustentabilidade do sistema. Entre os principais problemas decorrentes dessa negligência, destacam-se:

- Falhas de integração entre sistemas ou módulos; item
- Retrabalho em fases avançadas do projeto;
- Incompatibilidades técnicas com dispositivos ou protocolos externos;
- Frustração do usuário final, caso a interface não atenda às suas expectativas;
- Altos custos e riscos de mudanças tardias em interfaces externas ou de hardware.

Conforme destacado na ISO/IEC 25010:2011, atributos como usabilidade, compatibilidade, segurança e portabilidade dependem fortemente de requisitos de interface bem elaborados. A ausência desses requisitos ou sua definição imprecisa compromete tanto a experiência do usuário quanto a viabilidade técnica do sistema.

Portanto, a negligência nessa etapa pode comprometer não apenas a funcionalidade do sistema, mas também sua aceitação e sustentabilidade a longo prazo.

A má definição de requisitos de interface está entre as causas mais recorrentes de falhas em projetos de software. Dessa forma, os requisitos de interface devem ser compreendidos também como elementos críticos na gestão de riscos. Com isso, no que tange a mitigação desses riscos temos [27]:

- Elicitação abrangente, com participação ativa de usuários, analistas de integração, engenheiros de software e especialistas de domínio;
- Modelos de documentação padronizados, que promovam consistência e clareza;
- Rastreabilidade, permitindo acompanhar a evolução dos requisitos ao longo do ciclo de vida;
- Versionamento e revisões periódicas, que garantem controle de mudanças e qualidade contínua.

Esse alinhamento entre qualidade, documentação e risco constitui a base para uma engenharia de requisitos mais robusta, segura e orientada ao sucesso do projeto. No entanto, para que o sistema atue de forma eficaz dentro de seu ambiente real de uso, é necessário considerar também os requisitos operacionais, que será tratado a seguir.

Os requisitos operacionais são um subconjunto dos requisitos não funcionais e que descrevem as condições práticas sob as quais o sistema deve funcionar. Esses requisitos

vão além da interface e tratam de aspectos como ambiente de operação, suporte técnico, políticas de backup, instalação, configuração e monitoramento são elementos cruciais para garantir a estabilidade e continuidade do sistema após a entrega. Eles orientam a configuração do ambiente de produção e garantem que o sistema será funcional dentro das limitações reais da organização [33]. Esses requisitos fornecem parâmetros para a instalação, operação e suporte contínuo do sistema e abrangem os seguintes aspectos:

- **Ambiente de Operação:** Refere-se às condições nas quais o sistema deverá funcionar corretamente. Inclui clima, umidade, altitude, rede, energia, e até restrições regulatórias.
- **Disponibilidade e Confiabilidade:** Define o grau com que o sistema deve estar disponível para uso e funcionar sem falhas.
- **Manutenção e Suporte Técnico:** Requisitos relacionados à facilidade de manutenção, tempo de resposta de suporte e atualização do sistema.
- **Procedimentos de Backup e Recuperação:** Envolve a definição de políticas para garantir a integridade e recuperação dos dados em caso de falha.
- **Segurança Operacional:** Define os mecanismos para proteger o sistema de riscos operacionais, ataques cibernéticos ou uso indevido.
- **Escalabilidade e Portabilidade Operacional:** Especifica a capacidade do sistema de operar em diferentes ambientes ou crescer conforme a demanda.

A adequada documentação dos requisitos operacionais é uma extensão direta da sua correta identificação e categorização. Como discutido anteriormente, esses requisitos abrangem aspectos críticos como ambiente de operação, disponibilidade, segurança, manutenção, entre outros — todos fundamentais para o sucesso do sistema em seu contexto real de uso.

Importante salientar que os requisitos operacionais mal documentados ou ausentes não apenas comprometem a qualidade da solução, mas também dificultam atividades posteriores como testes de aceitação, planejamento de contingência e operação assistida [49]. Assim, a documentação se torna o instrumento que materializa e formaliza esses requisitos, permitindo comunicação eficaz entre todas as partes interessadas (analistas, arquitetos, operadores e gestores).

A adoção de boas práticas documentais fortalece a consistência e a validade dos requisitos operacionais, assegurando que eles estejam acessíveis, compreensíveis e atualizados. Entre essas práticas, destacam-se:

- Uso de modelos estruturados e normatizados, como o IEEE 830 e a ISO/IEC/IEEE 29148:2018;
- Rastreabilidade bidirecional, permitindo mapear requisitos para seus casos de uso, testes e código;
- Controle de versão e histórico de alterações;
- Participação de especialistas operacionais na validação dos requisitos;
- Inserção de critérios de aceitação e indicadores operacionais claros.

A ausência de uma documentação clara e padronizada pode gerar dificuldades de entendimento entre os envolvidos, impactando diretamente na qualidade da solução e nos mecanismos de operação e manutenção [50].

A negligência na documentação, elicitacão incompleta ou equivocada dos requisitos operacionais dos requisitos operacionais pode gerar riscos severos, tais como:

- Incompatibilidades técnicas durante a integração com ambientes reais;
- Custos inesperados com aquisição de infraestrutura ou serviços externos;
- Ausência de requisitos críticos relacionados à disponibilidade ou desempenho;
- Ambiguidade, que compromete testes e verificações;
- Falta de plano de contingência e recuperação de falhas, gerando impactos na continuidade do serviço.
- Falhas operacionais por ausência de processos definidos de suporte e manutenção.
- Impactos na segurança da informação, caso o sistema seja operado em ambientes não controlados.
- Não conformidade com políticas internas da organização, como regras de rede ou políticas de backup.

A má especificação de requisitos operacionais é uma das principais causas de falhas pós-implantação em sistemas complexos, especialmente aqueles com dependências críticas de infraestrutura. Para mitigar esses riscos, algumas técnicas e abordagens recomendadas incluem [27].

- Workshops e reuniões com equipes de infraestrutura e operação, para elucidar requisitos implícitos;

- Modelagem de processos com BPMN, destacando pontos críticos do ciclo operacional;
- tem Prototipagem de cenários de operação, para testar a aderência dos requisitos;
- Revisões formais com checklists baseados em normas, como a ISO/IEC 25010;
- Testes de aceitação operacional, alinhando os critérios documentados com o ambiente real.

A engenharia de requisitos deve manter uma postura preventiva e colaborativa, garantindo que os aspectos operacionais sejam tratados desde a fase inicial [51].

Além disso, é fundamental destacar que os requisitos operacionais devem estar alinhados aos atributos de qualidade do software definidos pela norma ISO/IEC 25010:2011. Essa norma estabelece que características como instalabilidade, portabilidade e manutenibilidade estão intimamente relacionadas ao ambiente operacional no qual o sistema será implantado. Dessa forma, os requisitos operacionais exercem influência direta sobre importantes atributos de qualidade, tais como:

- **Compatibilidade ambiental:** Refere-se à capacidade de um sistema de software operar corretamente em diferentes ambientes técnicos, como sistemas operacionais, navegadores, dispositivos móveis, infraestrutura em nuvem ou servidores físicos.
- **Desempenho:** Refere-se ao tempo de resposta, utilização de recursos e capacidade do sistema sob determinadas condições.
- **Confiabilidade:** Diz respeito à capacidade do sistema de manter seu funcionamento sob condições específicas durante um período determinado.
- **Usabilidade:** Refere-se à facilidade com que os usuários conseguem aprender e utilizar o sistema.
- **Segurança:** Relacionada à proteção de informações e à resistência a acessos não autorizados.
- **Portabilidade:** Capacidade do sistema de ser transferido de um ambiente para outro (hardware, SO, navegador etc.).
- **Compatibilidade:** Capacidade de o sistema coexistir e operar com outros sistemas ou versões.

Os requisitos operacionais asseguram a estabilidade e o funcionamento contínuo do sistema em seu ambiente final. No entanto, para que ele atinja esse estágio, é necessária uma migração adequada do desenvolvimento para a produção, o que é definido pelos requisitos de transição. Esses requisitos estabelecem as condições, recursos e ações necessárias para implantação, configuração e adaptação do sistema, garantindo uma entrega segura e eficiente. Assim, enquanto os requisitos operacionais sustentam a operação contínua, os de transição viabilizam sua implantação inicial com mínimo risco e interrupção. Os requisitos de transição são fundamentais para garantir uma migração suave e segura entre o sistema atual e o novo, preservando a continuidade dos negócios. Por definirem as condições necessárias para substituição ou atualização de sistemas, constituem uma categoria crítica da engenharia de requisitos. Sua ausência ou má definição pode causar interrupções operacionais, perda de dados, falhas de integração e rejeição do sistema pelos usuários, comprometendo diretamente os objetivos organizacionais. [52].

Essa visão é compartilhada reforça que os requisitos de transição devem ser identificados desde o início do projeto, sobretudo quando há sistemas legados, infraestruturas distribuídas ou diversos stakeholders envolvidos. Por isso, a transição demanda um planejamento multidisciplinar que inclua desenvolvedores, analistas de negócios, especialistas em infraestrutura, usuários e gestores de mudança organizacional [27].

Os requisitos de transição podem ser classificados em subcategorias, conforme a natureza das ações envolvidas:

- Migração de dados: Inclui transformação de formatos, limpeza de inconsistências, importação segura e validação de integridade. Pode ser realizada por scripts automatizados ou ferramentas de ETL;
- Treinamento e capacitação: Abrange o desenvolvimento de materiais didáticos, realização de workshops, vídeos tutoriais e treinamentos presenciais ou a distância (Canedo et al., 2022) [50];
- Implantação progressiva (*rollout*): Define etapas de introdução do novo sistema, como ambientes paralelos, implantação por módulos ou regiões, e estratégias de corte (*“cut-over”*);
- Reversibilidade (*rollback*): trata da capacidade de retornar ao sistema anterior em caso de falha, garantindo continuidade sem perda de dados ou serviços.

Esses elementos garantem que a transição entre sistemas ocorra de forma segura, minimizando o impacto sobre os usuários e os processos organizacionais, a partir da elicitação de requisitos bem estruturada.

A elicitação de requisitos de transição exige a interação com múltiplos stakeholders, incluindo usuários finais, equipes de operação, segurança da informação e TI corporativa. A abordagem colaborativa e iterativa é fundamental para identificar riscos e dependências ocultas. Técnicas como entrevistas, workshops, análise de sistemas legados e prototipagem são especialmente úteis para captar requisitos temporários e operacionais [50].

A documentação desses requisitos deve adotar modelos padronizados com campos específicos para contexto de aplicação, tempo de validade, critérios de aceitação e responsáveis. O uso de versionamento e rastreabilidade, como previsto em boas práticas da ISO/IEC/IEEE 29148:2018, também é recomendado para garantir consistência e facilitar auditorias e testes [45].

Os requisitos de transição são indispensáveis para garantir uma passagem segura e controlada do sistema atual para o novo, constituindo condições temporárias necessárias ao período de mudança. Fortemente ligados à gestão de mudanças organizacionais e tecnológicas, eles são frequentemente negligenciados, o que pode gerar riscos como perda de dados, indisponibilidade de serviços e rejeição pelos usuários. Esses impactos ocorrem quando tais requisitos não são definidos de forma clara e antecipada. Por isso, é fundamental identificá-los e documentá-los adequadamente, incluindo exemplos que orientem as ações técnicas e organizacionais necessárias à substituição segura do sistema legado. [52].

Entre os exemplos mais comuns de requisitos de transição, pode-se destacar:

- Migração de dados históricos: Importar os dados do sistema legado, convertendo formatos conforme necessário, para garantir continuidade e integridade das informações após a implantação;
- Acesso simultâneo ao ambiente antigo e novo: Permitir, na primeira semana, o uso paralelo dos dois ambientes, reduzindo riscos e possibilitando validar a estabilidade do novo sistema;
- Treinamento de usuários: Disponibilizar um manual de operação em PDF para apoiar o treinamento durante a transição, facilitando a adoção e prevenindo erros;
- Migração de perfis e permissões: Migrar automaticamente perfis e permissões, testando-os antes da entrada em produção, a fim de manter fluxos de trabalho e níveis de acesso;
- Implantação controlada: Realizar instalação paralela ao sistema atual e só iniciar a produção após homologação, permitindo uma transição progressiva e segura.

Os requisitos de transição atuam como instrumentos práticos para operacionalizar o conceito de mudança controlada, servindo de ponte entre o sistema antigo e o novo. Cada

requisito reflete uma ação específica voltada à mitigação dos riscos inerentes ao processo de substituição, reforçando sua relevância dentro da engenharia de requisitos orientada à gestão de riscos e à qualidade de entrega.

A ausência ou o levantamento impreciso de requisitos de transição pode gerar riscos operacionais e técnicos, como:

- Perda de dados críticos durante a migração.
- Incompatibilidades entre sistemas legados e novos sistemas.
- Resistência dos usuários finais, por falta de preparo ou treinamento.
- Descontinuidade nos processos de negócio durante o corte da produção.
- Falta de testes em ambientes semelhantes ao real, levando a falhas inesperadas.

É crucial que os requisitos de transição sejam devidamente elicitados, sobretudo em projetos que envolvem sistemas legados, arquiteturas complexas ou múltiplos stakeholders. Quando mal definidos, esses requisitos tornam-se riscos latentes, manifestando-se apenas nas etapas finais do projeto, quando correções são mais custosas. A identificação antecipada permite preparar planos de contingência, testar migrações, treinar usuários e garantir disponibilidade de dados e funcionalidades essenciais. Essa postura preventiva evidencia a transição como elemento estratégico da engenharia de requisitos, especialmente em ambientes críticos.

Da mesma forma, os requisitos de desempenho são fundamentais para assegurar a qualidade e a continuidade operacional do sistema. Enquanto os de transição viabilizam uma implantação segura, os requisitos de desempenho garantem que, em produção, o sistema opere dentro de parâmetros adequados de tempo de resposta, escalabilidade, capacidade e eficiência. A seguir, são explorados seus conceitos, categorias, exemplos e riscos, à luz da literatura e da ISO/IEC 25010:2011.

Os requisitos de desempenho são requisitos não funcionais que definem como o sistema deve se comportar quanto ao tempo de resposta, capacidade, uso de recursos e eficiência. Eles garantem rapidez, escalabilidade e estabilidade, alinhando o sistema às expectativas dos usuários. Esses requisitos descrevem características como a velocidade de resposta, o número de usuários simultâneos suportados e o volume de dados manipulados, influenciando diretamente a experiência do usuário e a viabilidade técnica da solução. [27].

Os requisitos de desempenho podem ser classificados em diversas categorias, conforme a ISO/IEC 25010:2011, seguem algumas:

- Tempo de Resposta: Estabelece o tempo máximo aceitável para que o sistema responda às solicitações do usuário.

- **Taxa de Processamento:** Define a quantidade de transações ou operações que o sistema deve processar dentro de um intervalo específico.
- **Utilização de Recursos:** Determina limites de uso de CPU, memória, rede e demais recursos para garantir eficiência.
- **Capacidade de Carga:** Especifica o número máximo de usuários ou processos simultâneos suportados sem degradação de desempenho.

A definição precisa dos requisitos de desempenho — como tempo de resposta, taxa de processamento, utilização de recursos e capacidade de carga — estabelece parâmetros objetivos para análise e validação da performance do sistema. Esses requisitos são fundamentais para assegurar qualidade e boa experiência do usuário, sobretudo em sistemas sujeitos a alta carga ou que demandam elevada disponibilidade.

As principais vantagens da correta definição desses requisitos incluem:

- Melhoria na experiência do usuário final, ao garantir respostas rápidas e previsíveis.
- Redução de falhas em ambientes de produção, uma vez que o sistema é projetado para suportar a carga esperada.
- Facilitação do planejamento de infraestrutura, pois os limites de desempenho guiam decisões sobre escalabilidade e recursos.
- Apoio à rastreabilidade e validação, pois requisitos bem definidos permitem testes de carga, stress e desempenho automatizados e mensuráveis

Entretanto, também existem desvantagens associadas à elicitación e especificación desses requisitos:

- Dificuldade na previsão precisa de carga futura, especialmente em sistemas inovadores ou com base de usuários volátil.
- Custos elevados de infraestrutura e testes, pois garantir requisitos de desempenho muitas vezes exige ambientes robustos para simulações realistas.
- Risco de superdimensionamento, caso os requisitos sejam exageradamente conservadores, gerando desperdício de recursos.
- Possível impacto em requisitos não funcionais conflitantes, como segurança e usabilidade, quando há priorização excessiva do desempenho.

A negligência ou inadequação na definição de requisitos de desempenho pode acarretar diversos riscos:

- Insatisfação do Usuário: Tempos de resposta lentos podem frustrar os usuários.
- Falhas Operacionais: Sobrecarga do sistema pode levar a falhas ou indisponibilidade.
- Perda de Oportunidades de Negócio: Sistemas lentos podem impactar negativamente a competitividade.

Algumas das estratégias de mitigação é envolver *stakeholders* para identificar expectativas realistas de desempenho, utilizar protótipos e testes de carga para validar requisitos, implementar ferramentas de monitoramento para detectar e corrigir problemas de desempenho em tempo real. Para que isso ocorra é importante que a documentação dos requisitos de desempenho deve ser clara, mensurável e verificável. A norma ISO/IEC 25010:2011 destaca a Eficiência de Desempenho como uma das características de qualidade do produto de software, subdividida em [46]:

- Comportamento em Relação ao Tempo: Capacidade do sistema de fornecer respostas dentro de um tempo adequado.
- Utilização de Recursos: Grau em que o sistema utiliza recursos de forma eficiente.
- Capacidade: Habilidade do sistema de lidar com a quantidade de trabalho especificada.

Enquanto o desempenho foca em como e com que rapidez o sistema responde sob carga, os requisitos de confiabilidade e disponibilidade dizem respeito à capacidade do sistema de operar corretamente e de estar acessível ao longo do tempo, mesmo diante de falhas ou condições adversas. Tais requisitos tornam-se particularmente críticos em sistemas que prestam serviços essenciais, onde interrupções ou falhas podem comprometer operações, segurança e confiança dos usuários.

A seguir, serão explorados os principais conceitos, categorias, exemplos práticos, riscos e formas de documentação associadas aos requisitos de confiabilidade e disponibilidade, com base em normas internacionais como a ISO/IEC 25010:2011 e alguns autores.

No contexto da engenharia de requisitos, os requisitos de confiabilidade e disponibilidade assumem papel fundamental para garantir a qualidade do sistema em ambientes críticos e de alta demanda. A confiabilidade de um sistema refere-se à sua capacidade de manter um desempenho correto sob condições operacionais específicas durante determinado intervalo de tempo. A disponibilidade, por sua vez, está relacionada à capacidade

do sistema de permanecer acessível e funcional sempre que necessário, sendo considerada uma subcaracterística da confiabilidade [27].

Esses requisitos são formalmente tratados na norma ISO/IEC 25010:2011, que estrutura a qualidade do produto de software em atributos mensuráveis. De acordo com essa norma, a confiabilidade compreende subatributos como maturidade, tolerância a falhas, disponibilidade e recuperabilidade. A maturidade expressa a probabilidade de o sistema operar sem falhas em condições normais; já a tolerância a falhas indica o grau em que o sistema é capaz de continuar operando diante de erros inesperados. A recuperabilidade, por fim, determina o esforço necessário para restaurar a operação normal após uma falha [46].

A definição clara e objetiva desses requisitos contribui significativamente para a redução de riscos técnicos e operacionais no ciclo de vida do software. A ausência ou má especificação de requisitos de confiabilidade pode levar a consequências severas, como:

- Falhas de serviço
- Interrupções operacionais;
- Perda de dados;
- Insatisfação do usuário final.

A documentação dos requisitos de confiabilidade deve seguir boas práticas, como as da ISO/IEC/IEEE 29148:2018, incluindo critérios de aceitação, contexto, prazo de validade e mecanismos de rastreabilidade. Esses cuidados facilitam auditorias, testes de robustez e avaliações de continuidade de serviço. Na prática, tais requisitos são definidos por metas mensuráveis — por exemplo, disponibilidade de 99,9% ou reinício automático após falha — complementadas por métricas como MTBF e MTTR, essenciais para monitorar o sistema em produção. Embora possam elevar a complexidade e os custos, especialmente devido à necessidade de redundância e monitoramento contínuo, esses requisitos oferecem benefícios significativos, como maior confiança do usuário, prevenção de prejuízos operacionais e redução de custos de suporte. Ainda assim, suas vantagens são evidentes:

- Promovem a confiança do usuário
- Evitam prejuízos operacionais
- Reduzem significativamente os custos com suporte corretivo.

Para mitigar os riscos relacionados à falha na definição desses requisitos, diversas estratégias podem ser aplicadas desde a fase de elicitação, como a realização de entrevistas

com especialistas técnicos, análise de SLAs pré-existentes, testes de estresse e simulações de falhas. O uso de ferramentas de monitoramento em tempo real, arquitetura distribuída e failover automático são exemplos concretos de práticas que reforçam a resiliência do sistema, não esquecendo da importância da confiabilidade como aspecto essencial da qualidade em sistemas críticos, especialmente em contextos regulados, onde falhas podem comprometer não apenas a operação, mas também a conformidade legal [24]. Importante mencionar acerca da necessidade de requisitos não funcionais, como confiabilidade e disponibilidade, sejam tratados com o mesmo nível de detalhamento e rigor que os requisitos funcionais, dada sua relevância na experiência do usuário e na viabilidade de negócios digitais [24].

A integração dos requisitos de confiabilidade e disponibilidade às práticas de documentação e gestão de riscos fortalece a construção de sistemas mais seguros, robustos e alinhados às expectativas dos stakeholders e às normas vigentes. Do mesmo modo, a atenção aos requisitos de manutenibilidade e suportabilidade vai além dos aspectos técnicos de atualização e suporte, envolvendo a necessidade de manter o software em conformidade com legislações e regulamentações em constante evolução.

Essa relação direta com o ambiente normativo reforça a importância dos requisitos legais e regulamentares, que asseguram a aderência do sistema a dispositivos como a LGPD, SOX, regulamentos da ANVISA e normas ISO. Tais requisitos impõem restrições que influenciam diretamente decisões de projeto, implementação e manutenção. Nesse contexto, a capacidade de adaptar o sistema a mudanças regulatórias torna-se essencial para mitigar riscos legais e garantir sustentabilidade operacional.

Assim, a robustez de um sistema depende não apenas de sua confiabilidade e disponibilidade, mas também da sua capacidade de ser mantido e evoluído com agilidade e baixo impacto. Os requisitos de manutenibilidade e suportabilidade, previstos na ISO/IEC 25010:2011, complementam essa visão e são especialmente críticos em projetos de médio e longo prazo, nos quais a sustentabilidade técnica e normativa é determinante para o sucesso.

Os requisitos de manutenibilidade e suportabilidade, pertencentes aos requisitos não funcionais, tratam da capacidade do sistema de ser modificado, corrigido e aprimorado de forma eficiente e segura ao longo do tempo. A manutenibilidade refere-se à facilidade de alterar o sistema com o mínimo de esforço, enquanto a suportabilidade diz respeito à facilidade de diagnóstico, monitoramento e resolução de problemas, sustentada por ferramentas, logs e documentação.

Esses requisitos devem ser identificados desde a elicitação, considerando infraestrutura, equipe de suporte e recursos organizacionais. Quando negligenciados, geram riscos como retrabalho, custos elevados de manutenção e falhas no suporte técnico. Além disso,

afetam atributos de qualidade definidos na ISO/IEC 25010:2011 — como modificabilidade, analisabilidade e estabilidade — comprometendo a sustentabilidade do sistema a longo prazo.

A partir da perspectiva da engenharia de requisitos, os objetivos de manutenibilidade e suportabilidade estão intimamente ligados à capacidade de o sistema permanecer sustentável, adaptável e operacional ao longo do tempo, mesmo diante de mudanças internas ou externas. Esses objetivos, ao serem corretamente elicitados e documentados, tornam-se instrumentos de mitigação de riscos técnicos, econômicos e operacionais no ciclo de vida do software [27] [46]. São os objetivos de manutenibilidade:

- Reduzir o tempo e o custo de manutenção do sistema
- Facilitar a evolução contínua do software em resposta a mudanças organizacionais ou tecnológicas.
- Aumentar a eficiência das equipes de suporte técnico.
- Melhorar a resiliência operacional, com respostas mais rápidas a falhas e erros.
- Evitar riscos técnicos e dívidas técnicas de difícil resolução no futuro.

Esses objetivos podem ser materializados por meio de exemplos práticos, conforme apresentado na figura 3.9, que ilustram como os requisitos se traduzem em ações concretas no sistema, com alguns exemplos extraídos e correlacionados diretamente com os objetivos estratégicos de cada tipo.

Código	Descrição
RM01	O código-fonte deve ser escrito seguindo o padrão de codificação da empresa.
RM02	O sistema deve possuir testes automatizados para no mínimo 80% do código.
RM03	Todas as alterações devem ser registradas em controle de versão (ex: Git).
RM04	O sistema deve possuir comentários explicativos em trechos críticos do código.
RM05	O sistema deve permitir substituição de componentes sem impactar outros módulos.

Figura 3.9: Exemplos de Requisitos de Manutenibilidade

Fonte: Elaborada pela autora (2025)

No que tange a norma ISO/IEC 25010 identifica a manutenibilidade como uma característica de qualidade do produto de software, subdividida em [46]:

- Modularidade: Capacidade de o sistema ser dividido em componentes independentes.
- Reusabilidade: Possibilidade de reutilizar partes do sistema em outros contextos.
- Analisabilidade: Facilidade de identificar causas de falhas.
- Modificabilidade: Capacidade de implementar alterações com risco mínimo.
- Testabilidade: Facilidade de verificar se alterações foram bem-sucedidas.

A Suportabilidade não é uma característica explícita na ISO 25010, mas é amplamente considerada como parte de operacionalidade, analisabilidade e confiabilidade em ambiente de produção

A figura 3.10 apresenta alguns exemplos de requisitos de suportabilidade.

Código	Descrição
RS01	O sistema deve registrar logs de erro com data, hora, usuário e stack trace.
RS02	O sistema deve enviar alertas por e-mail em caso de falhas críticas.
RS03	O sistema deve dispor de painel de monitoramento com métricas de saúde.
RS04	A documentação técnica deve estar atualizada e disponível para a equipe de suporte.
RS05	O sistema deve permitir ativar/desativar recursos de diagnóstico via configuração.

Figura 3.10: Exemplos de Requisitos de Suportabilidade

Fonte: Elaborada pela autora (2025)

Esses exemplos reforçam que a elicitacão de requisitos voltados à manutenção e suporte não deve ser negligenciada, principalmente em sistemas corporativos, públicos ou regulatórios, onde falhas na manutenção ou suporte representam riscos severos de indisponibilidade, descumprimento normativo ou aumento de custos com correções emergenciais [50] [33].

A consideracão dos requisitos de manutenibilidade e suportabilidade vai além dos aspectos técnicos de atualizacão e suporte, envolvendo também a necessidade de manter o software alinhado às exigências normativas ao longo de seu ciclo de vida. Diante da constante evoluçao de leis e regulamentacões, a capacidade de adaptar rapidamente o sistema torna-se estratégica para a sustentabilidade do produto e para a reduçao de riscos legais e operacionais.

Essa relação direta com o ambiente regulatório reforça a importância dos requisitos legais e regulamentares, responsáveis por garantir conformidade com normas e legislações como LGPD, SOX, regulamentos da ANVISA e padrões ISO. Esses requisitos impõem restrições que influenciam diretamente decisões de projeto, implementação, documentação e manutenção [53] [54].

Portanto, a capacidade de manter um sistema que suporte facilmente atualizações regulatórias — característica essencial da manutenibilidade e suportabilidade, torna-se um fator crítico de sucesso na gestão de riscos legais. Ao elicitar requisitos legais de forma estruturada e conectá-los com requisitos de manutenção, as equipes de desenvolvimento promovem maior conformidade normativa, reduzem a exposição a sanções e aumentam a confiabilidade institucional do software.

Os requisitos legais e regulamentares constituem uma categoria crítica dentro da engenharia de requisitos, especialmente em domínios altamente regulados, como saúde, finanças, governo e energia. Eles compreendem obrigações legais, normas técnicas, políticas internas e diretrizes regulatórias que o sistema deve obedecer. Esses requisitos “impõem restrições específicas que afetam diretamente decisões de projeto, implementação, documentação e manutenção”. Assim, sua identificação precoce e correta é essencial para assegurar a conformidade e evitar riscos jurídicos, financeiros e reputacionais [27].

Os requisitos legais e regulamentares devem ser tratados com o mesmo rigor técnico que os demais tipos de requisitos, devendo ser documentados, rastreados e validados ao longo do ciclo de vida do software. A norma recomenda que esses requisitos incluam a origem legal ou normativa, os critérios de conformidade, os responsáveis e o impacto esperado [45]. A elicitação desses requisitos deve envolver especialistas jurídicos, órgãos reguladores e representantes das áreas de compliance e governança, pois muitas vezes os desenvolvedores não possuem conhecimento profundo sobre as implicações legais. A ausência desses atores pode acarretar falhas sérias, como violações à LGPD (Lei Geral de Proteção de Dados), à SOX (Sarbanes-Oxley) ou ao Código de Defesa do Consumidor [24].

Esses requisitos podem ser aplicáveis a nível:

- Nacional (ex: leis brasileiras como LGPD, Código de Defesa do Consumidor)
- Internacional (ex: GDPR, ISO/IEC 27001, SOX, FDA, HIPAA)
- Setorial (ex: normas da Anvisa, Bacen, Aneel, CNJ, etc.)
- Corporativo (ex: políticas internas de compliance, segurança e governança)

Os requisitos legais também promovem transparência, rastreabilidade, proteção de dados sensíveis, integridade das informações e responsabilidade organizacional — pilares

essenciais em contextos como o da administração pública, da área da saúde e do setor financeiro.

No que tange os objetivos dos requisitos legais e regulamentares temos:

- Garantir que o sistema não infrinja leis ou normas vigentes.
- Evitar sanções legais, multas e ações judiciais.
- Demonstrar conformidade regulatória junto a auditorias e órgãos fiscalizadores.
- Assegurar a conformidade normativa
- Proteger os direitos de usuários, consumidores e instituições.
- Estabelecer regras claras e obrigatórias para o desenvolvimento e operação do sistema [24].

Diante dos objetivos, os exemplos de requisitos legais e regulamentares não são apenas instâncias descritivas, eles representam a aplicação concreta dos objetivos na engenharia de requisitos, servindo como ponte entre as demandas normativas e o desenvolvimento técnico do sistema, conforme apresentado na figura 3.11.

Código	Descrição do Requisito
RL01	O sistema deve garantir que os dados pessoais dos usuários sejam tratados conforme a LGPD.
RL02	Os relatórios financeiros devem ser exportados em conformidade com as normas da Receita Federal.
RL03	O sistema deve registrar o consentimento do usuário para coleta de dados sensíveis.
RL04	O sistema bancário deve seguir os padrões de segurança determinados pelo Bacen (Resolução nº 4.893).
RL05	O sistema deve atender aos requisitos de acessibilidade previstos no Decreto nº 5.296/2004.

Figura 3.11: Exemplos de Requisitos de Leis e Regulamentos

Fonte: Elaborada pela autora (2025)

Com base nos exemplos apresentados, torna-se evidente que os requisitos legais e regulamentares não são apenas obrigações formais, mas elementos estratégicos para garantir a sustentabilidade jurídica, reputacional e operacional do sistema. Cada requisito

traduz uma norma em uma especificação funcional ou não funcional que orienta o desenvolvimento do software conforme parâmetros legais específicos, além de antecipar riscos associados à não conformidade.

A não identificação ou violação de requisitos legais pode resultar em:

- Multas e sanções administrativas severas.
- Perda de certificações e credibilidade institucional.
- Ações civis, penais ou trabalhistas contra a organização
- Interrupção das atividades ou bloqueio do sistema por órgãos reguladores.
- Perda de confiança por parte dos clientes e usuários

Na fase de elicitação, é fundamental envolver especialistas jurídicos e de compliance, analisar contratos, legislações e regulamentos aplicáveis, além de levantar obrigações junto às áreas de jurídico, auditoria, LGPD e segurança da informação. Também se faz necessário identificar padrões exigidos por órgãos reguladores e validar os requisitos com base em jurisprudência, guias técnicos e decisões administrativas.

A documentação deve registrar não apenas a descrição do requisito, mas também seu vínculo com dispositivos legais ou normativos e as evidências de validação. A rastreabilidade entre esses requisitos e os artefatos de projeto — como código, testes e manuais — é essencial para fins de auditoria e certificação. A rastreabilidade entre esses requisitos e os artefatos de projeto (código, testes, manuais) é essencial para auditorias e certificações, como por exemplo:

- Elaborar uma documentação legal com referência à norma ou artigo correspondente.
- Manter uma planilha de rastreabilidade para facilitar auditorias;
- Acompanhar atualizações na legislação que possam afetar o sistema;
- Validar com especialistas e advogados antes da implementação
- Promover treinamento da equipe técnica quanto à importância desses requisitos.

Os requisitos legais e regulamentares são fundamentais para mitigar riscos jurídicos, operacionais e reputacionais, pois constituem exigências obrigatórias cuja ausência ou má definição pode comprometer o projeto e a própria legitimidade da organização perante seu ambiente regulatório. Na elicitação, sua identificação precisa é crucial, já que omissões podem gerar impactos severos em custos, cronograma, imagem institucional e até resultar em sanções legais.

Além de obrigatórios, esses requisitos possuem caráter estratégico, garantindo conformidade com legislações, normas técnicas e regulamentos, como LGPD, ISO/IEC 27001 e normas setoriais. Por isso, devem ser identificados desde o início, com apoio de especialistas jurídicos, compliance e auditoria, integrando-se à gestão contínua de requisitos e riscos.

No contexto da Engenharia de Requisitos, a elicitação constitui a etapa inicial responsável por compreender o problema a ser resolvido, identificar as necessidades dos stakeholders, analisar possíveis conflitos, definir limites do sistema e compreender o contexto operacional e normativo no qual a solução será inserida. Nesse processo, os requisitos legais e regulamentares assumem papel estruturante, uma vez que orientam decisões de projeto e estabelecem restrições fundamentais para a manutenção, a conformidade e a evolução segura do sistema ao longo de seu ciclo de vida [55]. Contudo, os requisitos obtidos durante a elicitação geralmente apresentam-se em um nível ainda elevado de abstração, demandando uma etapa subsequente capaz de organizá-los, refiná-los e torná-los tecnicamente viáveis.

É nesse cenário que se insere a análise de requisitos, etapa fundamental do processo de desenvolvimento de software, responsável por examinar de forma sistemática os requisitos elicitados, eliminando ambiguidades, identificando inconsistências, resolvendo conflitos e avaliando a viabilidade técnica e organizacional das necessidades levantadas. A análise de requisitos tem como objetivo transformar informações inicialmente expressas de maneira informal ou pouco estruturada em requisitos claros, completos, consistentes e verificáveis, aptos a orientar as etapas posteriores de projeto, implementação e validação do sistema.

Dessa forma, a análise de requisitos atua como um elo essencial entre a compreensão do problema — obtida na elicitação — e a definição precisa da solução a ser implementada, assegurando que as expectativas dos stakeholders sejam corretamente interpretadas e traduzidas em especificações que atendam aos objetivos do negócio, às restrições legais e às condições operacionais do sistema.

A análise de requisitos é um processo sistemático de examinar, identificar e entender os requisitos do sistema. Durante essa fase, os requisitos são revisados detalhadamente para identificar inconsistências, ambiguidades e lacunas, garantindo que sejam compreensíveis e completos para todas as partes interessadas [56].

Análise de requisitos também deve levar em consideração fatores externos que possam impactar o projeto, como regulamentações específicas, riscos técnicos e desafios operacionais.

A análise de requisitos desempenha um papel crucial na gestão de riscos em projetos de Tecnologia da Informação (TI). Em seu estudo, eles destacam que uma definição clara e precisa dos requisitos é essencial para identificar e mitigar riscos desde as fases iniciais

do projeto [57].

Além disso, os autores observaram que fatores como escopo, custo, prazo e qualidade têm um impacto significativo no sucesso dos projetos. Uma análise de requisitos bem conduzida contribui para o controle desses fatores, reduzindo a probabilidade de riscos e aumentando as chances de sucesso do projeto.

Contudo, a análise de requisitos não apenas define as necessidades e expectativas do projeto, mas também serve como uma ferramenta fundamental para a identificação e mitigação de riscos, sendo, portanto, indispensável para o sucesso dos projetos de TI [57].

A análise de requisitos não deve se limitar apenas a aspectos técnicos, mas também considerar a dinâmica organizacional e o impacto que o sistema pode ter nos processos internos. Uma abordagem estruturada e necessária na análise de riscos pois isso impacta na identificação de riscos financeiros e operacionais do projeto, uma análise de requisitos bem feita e detalhada auxilia na mitigação de problemas que podem comprometer a execução de um projeto. A aplicação de metodologias como FMEA (*Failure Modes and Effects Analysis*) e FTA (*Fault Tree Analysis*) permite uma avaliação sistemática das vulnerabilidades do sistema antes de sua implementação [58].

Uma análise de requisitos cuidadosa e abrangente garante que os requisitos do sistema atendam às necessidades e expectativas dos usuários finais. Isso envolve não apenas a identificação de requisitos funcionais (que descrevem as funcionalidades que o sistema deve oferecer), mas também requisitos não funcionais (como desempenho, usabilidade e segurança), que são igualmente importantes para o sucesso do sistema [38].

Para uma análise de requisitos detalhada e consistente, fase de análise de requisitos é dividida em três atividades essenciais [38].

- Classificação e Organização dos Requisitos.
- Priorização e Negociação dos Requisitos.
- Documentação dos Requisitos.

Essas atividades não apenas estruturam o processo de análise, mas também auxiliam na identificação de relações hierárquicas, dependências e prioridades entre os requisitos.

A etapa de priorização e negociação tem um papel central, pois é nela que os requisitos são avaliados quanto à sua importância, urgência e viabilidade. Conflitos entre stakeholders, restrições orçamentárias e limitações de tempo são comuns nesse momento, exigindo decisões sobre quais requisitos devem ser implementados primeiro, quais podem ser adiados e quais eventualmente serão descartados.

Para conduzir a elicitação de forma eficaz, podem ser empregadas técnicas como workshops colaborativos, prototipação e análise de impacto, que facilitam o alinhamento

entre stakeholders e a construção de um consenso realista sobre o escopo. Após essa etapa, os requisitos devem ser documentados de maneira clara, precisa e verificável. Essa documentação deve ser compreensível, livre de ambiguidades e servir como base para o desenvolvimento e testes.

A análise de requisitos envolve também a priorização, permitindo concentrar esforços nos itens mais críticos e otimizar o uso de recursos do projeto. Além disso, uma comunicação eficaz entre analistas e stakeholders é essencial para garantir entendimento compartilhado sobre funcionalidades e restrições do sistema, reforçando a importância do diálogo contínuo durante todo o processo [59]. A sabedoria coletiva da equipe de desenvolvimento tem um impacto significativo na performance do projeto, uma vez que o compartilhamento de conhecimento entre os membros pode minimizar erros e aumentar a precisão dos requisitos levantados [60]. No contexto de metodologias ágeis, a análise de requisitos também precisa ser adaptável e iterativa, assim como a modelagem de processos de negócios, que pode ser utilizada como um suporte à análise de requisitos, permitindo que a equipe compreenda melhor as interações entre os componentes do sistema e as necessidades do usuário. O uso de técnicas como BPMN (Business Process Model and Notation) auxilia na criação de representações visuais que facilitam a identificação de lacunas e pontos de melhoria no software [61].

Outro aspecto relevante para a análise de requisitos é a adequação às normas de proteção de dados, como a LGPD (Lei Geral de Proteção de Dados). Muitos times ágeis ainda enfrentam dificuldades na elicitação de requisitos de privacidade, o que reforça a necessidade de incluir essa categoria de análise no planejamento inicial do projeto. Sobre esse ponto, os autores destacam que "as equipes ágeis consideram que requisitos de software e construção de software são as áreas de conhecimento mais impactadas pela LGPD, e a maioria delas utiliza histórias de usuário na elicitação de requisitos de privacidade"[62].

Além disso, a implementação da LGPD exigiu que as organizações adotassem práticas e ferramentas específicas para assegurar conformidade regulatória. Contudo, as ferramentas existentes para apoiar a elicitação de requisitos de privacidade ainda são subutilizadas pelas equipes ágeis, o que reforça a importância de aprimorar essa fase do desenvolvimento [26].

Com os requisitos devidamente analisados, refinados e categorizados, a equipe de desenvolvimento encontra-se apta a avançar para a etapa subsequente do processo de Engenharia de Requisitos, voltada à sua formalização. Nesse contexto, a especificação de requisitos assume papel central ao transformar os resultados da análise em uma representação clara, estruturada e compreensível, capaz de consolidar todas as informações coletadas ao longo das fases anteriores. Essa formalização é fundamental para fornecer uma base consistente às atividades de projeto, desenvolvimento e validação do sistema,

além de atuar como um instrumento de alinhamento entre stakeholders e equipe técnica, assegurando uma compreensão comum e precisa das funcionalidades, restrições e critérios de qualidade do software a ser construído.

A especificação de requisitos tem como objetivo principal documentar os requisitos de forma organizada e não ambígua, reduzindo interpretações divergentes e promovendo a consistência ao longo do ciclo de vida do projeto. Essa etapa é essencial para garantir que todos os envolvidos compartilhem uma visão unificada do sistema, abrangendo tanto os aspectos funcionais quanto as restrições e atributos de qualidade. De modo complementar, a especificação de requisitos é compreendida como uma atividade fundamental da Engenharia de Software, uma vez que estabelece, de maneira clara e precisa, as funcionalidades e limitações do sistema, servindo como referência para as fases subsequentes de desenvolvimento e para os processos de verificação e validação [50].

A documentação dos requisitos deve ser compreensível para todas as partes interessadas e conter informações detalhadas sobre as funcionalidades e características do sistema a ser desenvolvido.[38].

Portanto, a fase de especificação de requisitos representa uma etapa crítica no desenvolvimento de software, na qual os requisitos previamente analisados são formalizados de maneira clara, estruturada e compreensível. Uma abordagem cuidadosa e sistemática nessa etapa é essencial para garantir a correta interpretação pelos desenvolvedores, o alinhamento com as expectativas dos stakeholders e, conseqüentemente, o sucesso do projeto e a satisfação do cliente. Além disso, a especificação de requisitos está intimamente relacionada à negociação dos requisitos, especialmente em contextos onde há conflitos entre as demandas de diferentes stakeholders ou quando os recursos disponíveis (como tempo, orçamento e tecnologia) são limitados em relação à quantidade e complexidade dos requisitos elicitados. Nesses casos, a negociação torna-se fundamental para estabelecer prioridades, alinhar expectativas e definir compromissos realistas, assegurando a viabilidade e a sustentabilidade do projeto.

O principal objetivo da elicitação é documentar os requisitos de maneira que eles possam ser:

- Interpretados corretamente pelos desenvolvedores,
- Validados pelos stakeholders,
- Utilizados como referência durante os testes

A especificação atua como um contrato técnico entre a equipe de desenvolvimento e os stakeholders, descrevendo o “o que” o sistema deve fazer, sem necessariamente indicar “como” isso será feito e ela pode assumir diferentes formas, de acordo com o contexto do projeto, como por exemplo:

Especificação natural (texto estruturado): Utiliza linguagem natural padronizada, com termos bem definidos, evitando ambiguidade e é recomendado para stakeholders não técnicos.

Especificação semi-formal: Emprega modelos gráficos, como diagramas UML (casos de uso, classes, atividades) ou BPMN e facilita a visualização dos fluxos do sistema.

Especificação formal: Utiliza linguagens matemáticas (ex: Z, VDM) e é indicada para sistemas críticos que exigem alta confiabilidade e ausência de ambiguidade.

A especificação de requisitos deve ser elaborada com rigor, de forma clara, precisa e sem margem para interpretações duplicadas, pois orienta todas as etapas do ciclo de vida do software. Quando mal definida, pode gerar retrabalho, falhas de comunicação entre equipe e stakeholders e funcionalidades que não atendem às necessidades reais do usuário. Assim, investir em uma especificação completa e bem estruturada é essencial para reduzir riscos, otimizar recursos e aumentar as chances de sucesso do projeto. [27].

Essa atenção à qualidade da especificação é ainda mais relevante quando se consideram aspectos legais e regulatórios envolvidos no desenvolvimento de sistemas. A ausência de requisitos de privacidade devidamente especificados pode comprometer a conformidade com legislações como a Lei Geral de Proteção de Dados (LGPD). Isso reforça a necessidade de incluir no documento de especificação categorias específicas, como requisitos legais, de segurança da informação e de privacidade, assegurando que o sistema esteja em conformidade com as exigências normativas e mitigue riscos associados à proteção de dados dos usuários [34].

A especificação de requisitos deve ser documentada de forma clara e precisa para servir como base para o desenvolvimento do software. A documentação dos requisitos deve ser compreensível para todas as partes interessadas e conter informações detalhadas sobre as funcionalidades e características do sistema a ser desenvolvido. O uso de ferramentas de modelagem de processos, como BPMN, pode contribuir para uma visualização mais clara dos requisitos e sua relação com os processos de negócio [43]. Portanto, a fase de elicitação de requisitos é uma etapa crítica no desenvolvimento de software, que envolve a coleta, análise e definição dos requisitos do sistema. Uma abordagem cuidadosa e sistemática durante essa fase é essencial para garantir o sucesso do projeto e a satisfação do cliente. A especificação de requisitos está diretamente ligada à negociação dos requisitos, o que se torna necessário quando diferentes stakeholders têm requisitos conflitantes ou quando há mais requisitos do que os recursos disponíveis (tempo, orçamento, tecnologia) podem suportar. A etapa de negociação de requisitos divide-se em três atividades essenciais [61]:

- Discussão de Prioridades: Determinar quais requisitos são essenciais e quais podem ser alterados ou adiados.

- Resolução de Conflitos: Encontrar soluções de compromisso quando stakeholders têm visões divergentes sobre certos requisitos.
- Ajuste de Expectativas: Garantir que as expectativas dos stakeholders sejam realistas em relação ao que pode ser entregue.

Além disso, os desafios enfrentados na elicitação de requisitos de privacidade, como falta de ferramentas apropriadas e desconhecimento dos princípios legais, podem comprometer a qualidade da negociação dos requisitos, tornando fundamental a adoção de diretrizes claras para lidar com esses aspectos desde o início do projeto. O resultado esperado é que haja um conjunto de requisitos acordado, balanceando as necessidades e limitações, e estabelecendo um entendimento claro do que será desenvolvido. Para isso, a precisão e eficácia do método de desenvolvimento adotado são fundamentais para o sucesso de qualquer projeto de software, especialmente quando se considera a importância crítica das quatro fases de elicitação, análise, validação e negociação de requisitos. Essas fases são interdependentes e cada uma contribui significativamente para garantir que o produto final não apenas seja viável, mas também atenda às expectativas dos stakeholders e seja concluído dentro dos prazos e orçamentos estipulados [62].

A utilização de metodologias ágeis associadas a práticas colaborativas como Lean Inception e modelagem de processos pode otimizar o alinhamento entre os requisitos do software e as necessidades dos usuários, garantindo um desenvolvimento mais eficiente e ajustado às demandas reais do projeto.

Assim, ao final da fase de especificação, é fundamental assegurar que os requisitos documentados realmente reflitam as necessidades e expectativas dos stakeholders, estejam em conformidade com normas e restrições aplicáveis, e sejam compreendidos de forma unívoca por todos os envolvidos no projeto. É nesse contexto que se inicia a validação dos requisitos, uma etapa essencial para verificar a consistência, completude e viabilidade dos requisitos especificados. A validação atua como um filtro de qualidade, garantindo que erros, ambiguidades ou omissões sejam identificados e corrigidos antes que o desenvolvimento avance para fases subsequentes, como o design e a implementação.

A fase de validação de requisitos é responsável por garantir que os requisitos especificados representem de forma fidedigna as necessidades dos stakeholders e estejam corretos, completos, consistentes e viáveis para implementação. Essa etapa verifica “se os requisitos realmente descrevem o sistema que os stakeholders desejam”, atuando como um controle de qualidade que antecede o design e o desenvolvimento do software.

Para garantir a validação adequada dos requisitos, é necessário envolver ativamente os stakeholders durante todo o processo. A comunicação eficaz entre os desenvolvedores e os stakeholders é crucial para garantir que os requisitos atendam às expectativas e necessidades do cliente.

A validação de requisitos é uma parte fundamental do processo de engenharia de requisitos, sendo uma verificação para confirmar se os requisitos capturados são corretos, completos e compreensíveis antes de avançar para as etapas de design e implementação do sistema [13].

A validação de requisitos envolve a realização de revisões e avaliações dos requisitos por parte das partes interessadas. Essas revisões podem incluir a realização de demonstrações, prototipagem rápida e outras técnicas para garantir que os requisitos estejam alinhados com as expectativas do cliente [39].

Além disso, a validação de requisitos envolve a realização de revisões e avaliações dos requisitos por parte das partes interessadas. Essas revisões podem incluir a realização de demonstrações, prototipagem rápida e outras técnicas para garantir que os requisitos estejam alinhados com as expectativas do cliente. No contexto da abordagem de Lean Inception, essa prática se torna ainda mais relevante, uma vez que promove um processo iterativo de refinamento dos requisitos. A Lean Inception se baseia na empatia, colaboração e experimentação para garantir que as necessidades do usuário sejam bem compreendidas e traduzidas em requisitos que atendam às expectativas do projeto".

A ausência de um processo eficaz de validação de requisitos pode acarretar diversos impactos negativos ao longo do ciclo de vida do software, comprometendo não apenas o sucesso do projeto, mas também a experiência dos usuários e a conformidade com exigências legais e técnicas. Entre as principais consequências de uma validação deficiente, destacam-se:

- Implementação de funcionalidades incorretas ou irrelevantes: Requisitos mal validados podem gerar funções que não atendem às necessidades reais, causando desperdício de tempo e recursos.
- Insatisfação dos usuários: Divergências entre requisitos implementados e expectativas dos stakeholders reduzem a aceitação e o valor prático do sistema.
- Aumento de retrabalho e custos: Erros ou omissões identificados tardiamente exigem correções mais caras em fases avançadas do projeto.
- Risco de não conformidade legal, como com a LGPD: Validação inadequada de requisitos de privacidade e segurança pode resultar em infrações e sanções.
- Comprometimento da qualidade final: A falta de validação afeta confiabilidade, usabilidade e segurança, gerando sistemas instáveis e difíceis de manter.

Diante disso, a validação deve ser encarada não apenas como uma etapa técnica, mas como um instrumento estratégico de garantia da qualidade e mitigação de riscos,

sendo fundamental para o alinhamento entre as expectativas dos stakeholders e a solução desenvolvida.

Essa atenção é ainda mais crítica quando se tratam de requisitos ligados à segurança e privacidade da informação. A validação de requisitos de privacidade é essencial para garantir a conformidade com legislações como a Lei Geral de Proteção de Dados (LGPD). Durante essa fase, requisitos relacionados à proteção de dados pessoais devem ser cuidadosamente revisados com o apoio de especialistas jurídicos e de segurança da informação, a fim de evitar falhas que possam comprometer a legalidade e a credibilidade do sistema [62].

Outro aspecto importante da validação de requisitos é a verificação de conformidade com padrões e regulamentos aplicáveis. É essencial garantir que os requisitos estejam em conformidade com normas técnicas, regulamentos governamentais e requisitos de segurança e privacidade, quando aplicável. Dessa forma, a validação deve incluir não apenas a análise dos requisitos funcionais, mas também a verificação de aspectos críticos como usabilidade, desempenho e proteção de dados, assegurando que o sistema esteja alinhado às exigências legais e às boas práticas de engenharia de software [62].

Em síntese, a validação de requisitos constitui uma etapa indispensável no desenvolvimento de software, assegurando que os requisitos identificados reflitam de maneira fiel e completa as necessidades e expectativas dos stakeholders. Uma condução rigorosa, estruturada e criteriosa dessa fase é determinante para o sucesso do projeto, reduzindo riscos, fortalecendo a qualidade do produto e promovendo a satisfação do cliente.

Nesse contexto, a fase de elicitação de requisitos ganha destaque, pois constitui o ponto de partida crítico para a definição do que será desenvolvido. A seguir, são abordados os métodos tradicionais e ágeis de elicitação, os quais compreendem técnicas estruturadas e adaptáveis utilizadas para identificar, coletar e registrar as necessidades dos stakeholders com clareza, precisão e alinhamento aos objetivos do projeto.

3.6 Métodos de elicitação de Requisitos

A elicitação de requisitos é uma etapa essencial da Engenharia de Requisitos e demanda técnicas e métodos específicos. Seu objetivo central é compreender claramente o problema a ser resolvido, reunindo informações que permitam identificar as necessidades que o sistema deve atender. Para isso, os requisitos podem ser obtidos por meio de abordagens tradicionais ou metodologias ágeis.

As metodologias tradicionais, como os modelos em cascata e espiral, adotam uma sequência rígida de fases, com forte ênfase no planejamento inicial e na produção detalhada de documentação. Essas abordagens são consideradas “metodologias de engenharia”,

pois seguem princípios clássicos em que cada etapa deve ser concluída antes do início da próxima.[63].

As abordagens tradicionais são adequadas para projetos de grande porte, equipes distribuídas e contextos com requisitos complexos. Nesses casos, utilizam-se técnicas como entrevistas, questionários, análise documental, modelagem de processos, casos de uso, observação e construção de cenários. Entretanto, seu principal limite é a baixa flexibilidade para incorporar mudanças, já que o planejamento ocorre antes da implementação.

Em contraste, as metodologias ágeis surgiram para superar a rigidez das abordagens tradicionais, adotando processos mais leves, maior colaboração e entregas incrementais. Elas valorizam comunicação direta, equipes autogerenciáveis e tomada de decisões técnicas pelos desenvolvedores. Métodos ágeis são iterativos e incrementais, utilizando histórias de usuário e critérios de aceitação para entregas frequentes e coleta contínua de feedback. Por isso, são recomendadas para projetos com requisitos mutáveis e equipes pequenas e experientes, com destaque para Scrum e Kanban.[40].

3.6.1 Técnicas de elicitação de requisitos a partir da modelagem de processos de negócio

Para se estabelecer as técnicas de elicitação de requisitos a partir da modelagem de processos de negócios, é imprescindível compreender a definição de requisitos, bem como a engenharia responsável pelo seu estudo. Entende-se como requisitos os problemas a serem resolvidos ou necessidades a serem atendidas por um sistema de software ou solução tecnológica, os quais se subdividem em requisitos de negócio, requisitos de stakeholders (ou de usuário) e requisitos de solução (ou do sistema). Esta pesquisa enfatiza este último — o terceiro nível — que, por sua vez, se classifica em requisitos funcionais e não funcionais, a depender do contexto e dos objetivos organizacionais [41]

A utilização da modelagem de processos de negócio como instrumento de apoio à elicitação de requisitos se justifica pela sua capacidade de representar de forma gráfica e padronizada os fluxos operacionais de uma organização. Essa abordagem proporciona uma visão sistêmica das atividades-chave, dos papéis envolvidos e das interações entre os componentes organizacionais, permitindo a identificação de lacunas, gargalos e oportunidades de automação que podem ser convertidas em requisitos de sistema. Ao modelar os processos de negócio antes ou durante a elicitação, os analistas conseguem compreender melhor as necessidades reais dos usuários, evitando a definição de requisitos com base apenas em percepções subjetivas ou insuficientemente contextualizadas. Essa prática fortalece a rastreabilidade entre os objetivos do negócio e as funcionalidades do sistema, promovendo maior alinhamento estratégico entre a solução e a organização [64].

A integração entre modelagem de processos e elicitação de requisitos contribui para o gerenciamento de riscos, tanto técnicos quanto gerenciais, ao permitir a análise de vulnerabilidades, dependências críticas e pontos de falha nos processos existentes. Com isso, torna-se possível antecipar falhas potenciais e desenhar soluções mais robustas, mitigando impactos futuros [65].

A modelagem de processos — especialmente com o uso da notação BPMN é destacada como uma prática recomendada em projetos que demandam padronização, clareza comunicativa e reutilização de modelos. O uso de diagramas padronizados facilita a comunicação entre analistas, desenvolvedores e stakeholders, aumentando a precisão na elicitação e na documentação dos requisitos [66].

Em projetos sujeitos a normas legais e regulatórias, como aqueles que devem atender à Lei Geral de Proteção de Dados (LGPD), a modelagem de processos torna-se ainda mais relevante. A identificação de requisitos de privacidade e segurança precisa ser integrada desde as fases iniciais do desenvolvimento, e a visualização clara dos fluxos de dados pessoais nos processos organizacionais é essencial para essa finalidade. A modelagem, portanto, contribui não apenas para a definição funcional do sistema, mas também para a conformidade normativa e a gestão de riscos legais [67].

Em síntese, a aplicação de técnicas de elicitação de requisitos apoiadas na modelagem de processos de negócio contribui para uma compreensão mais profunda das necessidades organizacionais, para a redução de incertezas no desenvolvimento e para a melhoria na qualidade da especificação de requisitos. Trata-se de uma abordagem cada vez mais valorizada na engenharia de software por sua capacidade de alinhar tecnologia, processos e pessoas de maneira eficaz.

A modelagem permite representar visualmente fluxos de trabalho, agentes envolvidos, regras de negócio e pontos críticos, o que facilita a identificação de requisitos implícitos, muitas vezes não verbalizados pelos stakeholders. Essa técnica é especialmente útil em ambientes organizacionais complexos, onde as relações entre áreas de negócio, sistemas legados e objetivos estratégicos precisam ser consideradas de maneira integrada.

Ao se utilizar a modelagem de processos de negócio como suporte à engenharia de requisitos, é possível obter ganhos expressivos em clareza, alinhamento e rastreabilidade. Além disso, essa abordagem contribui diretamente para a mitigação de riscos, pois proporciona uma visão holística dos processos organizacionais e suas possíveis fragilidades [64].

No entanto, apesar de sua eficácia, a modelagem de processos por si só pode não ser suficiente para garantir a abrangência e a precisão dos requisitos elicitados, principalmente em contextos onde as necessidades dos usuários são complexas, voláteis ou difíceis de articular. É nesse cenário que entram em cena metodologias específicas de elicitação,

desenvolvidas com o objetivo de fornecer diretrizes estruturadas e adaptáveis à realidade de organizações públicas ou privadas.

Uma dessas metodologias é a REMO (*Requirements Elicitation Methodology for Organizations*), proposta com o intuito de atender à necessidade de abordagens sistematizadas que combinem técnicas qualitativas de coleta de dados, participação ativa de stakeholders e alinhamento com os processos organizacionais.

3.6.2 Metodologia REMO

A técnica REMO (*Requirements Elicitation oriented to Business Process Modeling*) destaca-se entre as metodologias recentes de extração de requisitos a partir da modelagem de processos de negócio. Ela propõe um método sistemático para identificar requisitos com base na estrutura organizacional e nos fluxos de trabalho existentes, tomando a modelagem de processos como principal referência. A REMO utiliza um conjunto de heurísticas estruturadas para apoiar a extração de informações implícitas e explícitas sobre o funcionamento do sistema, atividade central da elicitación de requisitos.[68].

A REMO parte da premissa de que os processos de negócio são fontes diretas e valiosas para identificar o comportamento esperado do sistema, incluindo regras, interações e restrições. Por isso, assume que a modelagem de processos deve anteceder e orientar a elicitación, permitindo uma abordagem mais estruturada, alinhada ao domínio organizacional e com maior rastreabilidade.

A REMO reduz a distância entre o domínio da organização e o domínio do sistema ao oferecer heurísticas que ajudam o analista a interpretar elementos do modelo — como tarefas, eventos, decisões e fluxos — como potenciais requisitos. Essas heurísticas orientam a transformação dos componentes do processo de negócio em artefatos formais da engenharia de requisitos. [68].

A metodologia REMO também contribui significativamente para a qualidade da documentação de requisitos, uma vez que permite a extração sistemática, rastreável e validável de informações. Essa qualidade está relacionada à completude, à ausência de ambiguidade e à aderência às necessidades reais dos stakeholders — fatores essenciais apontados por Sommerville (2011) como indicadores da eficácia da engenharia de requisitos.

Além disso, a REMO dialoga com a ISO/IEC/IEEE 29148:2018, um dos principais instrumentos normativos internacionais utilizados em projetos de software e sistemas complexos. As diretrizes da ISO/IEC/IEEE 29148:2018 trata da elaboração e gestão de requisitos em projetos de engenharia de software e sistemas. A norma enfatiza a importância da consistência e da rastreabilidade dos requisitos, aspectos diretamente atendidos pela abordagem REMO ao vincular cada requisito identificado a elementos concretos do processo de negócio modelado [69].

A técnica REMO configura-se como uma evolução metodológica na elicitação de requisitos ao utilizar modelos organizacionais como base para a identificação sistemática de requisitos de software. Essa abordagem amplia a capacidade do analista de compreender o domínio, estruturar o processo de elicitação e produzir documentação mais precisa e alinhada aos objetivos institucionais.

Ao fundamentar-se na modelagem de processos de negócio, a REMO destaca-se por empregar heurísticas específicas que orientam a interpretação de elementos do modelo — como tarefas, fluxos, eventos e decisões — convertendo-os em requisitos funcionais e não funcionais de maneira clara e rastreável. Assim, o uso de heurísticas torna o processo de elicitação mais objetivo, consistente e menos dependente da subjetividade do analista.

Na sequência, serão detalhadas as heurísticas utilizadas na REMO, bem como seus propósitos, aplicações práticas e contribuições para a qualidade da engenharia de requisitos.

3.6.3 Heurísticas

No contexto da engenharia de requisitos orientada à modelagem de processos de negócio, heurísticas são definidas como diretrizes sistemáticas e práticas que auxiliam os analistas na identificação de requisitos a partir de elementos modelados nos processos organizacionais [70].

Em outras palavras, heurísticas funcionam como instrumentos cognitivos que orientam a interpretação de componentes dos modelos como tarefas, eventos, decisões e fluxos, permitindo traduzir esses elementos em requisitos funcionais, não funcionais ou regras de negócio.

Elas não são regras rígidas, mas sim pontos de partida organizados e validados empiricamente que ajudam a diminuir a ambiguidade e a subjetividade na elicitação. Ao aplicar heurísticas, o analista sistematiza sua observação sobre o processo, aumentando a consistência, rastreabilidade e completude dos requisitos extraídos.

Posteriormente à construção e análise dos modelos de processo, realiza-se o Processo de Elicitação de Requisitos (PER), a partir da extração inicial das informações contidas no diagrama de processos, como aqueles modelados em BPMN (*Business Process Model and Notation*). Essa etapa permite a identificação de dados gerais do processo, problemas recorrentes, necessidades dos stakeholders e regras de negócio. A partir dessa análise preliminar, passa-se então à aplicação das heurísticas específicas para o PER, as quais operam como guias sistemáticos para a extração de requisitos com base nos elementos estruturais do modelo de processo.

Essa abordagem visa tornar a elicitação mais objetiva, rastreável e completa, reduzindo a dependência da interpretação subjetiva por parte do analista de requisitos. [70]. A

aplicação das heurísticas no contexto da modelagem de processos consiste na análise minuciosa dos elementos do metamodelo BPMN, com o intuito de facilitar a identificação de Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF). As heurísticas, nesse sentido, atuam como ferramentas cognitivo-metodológicas que guiam o analista a partir de padrões recorrentes em estruturas de modelagem.

A seguir, são apresentadas as heurísticas propostas por Mota (2022), com seus respectivos objetivos:

Heurística de Elicitação 1: Raias (Swimlanes)

- As raias indicam os agentes responsáveis pelas atividades do processo, permitindo identificar atores, papéis, permissões e restrições de acesso.
- Exemplo: Se a raia corresponde ao setor financeiro, as atividades nela presentes devem refletir requisitos desse domínio, como “visualizar pagamentos” ou “emitir relatórios financeiros”.

Heurística de Elicitação 2: Atividades e/ou Tarefas

- Cada atividade ou tarefa descrita no modelo BPMN é potencialmente um requisito funcional. Esta heurística foca na conversão das ações modeladas em funcionalidades que o sistema deverá realizar.

Heurística de Elicitação 3: Gateways do Processo

- Gateways representam decisões e condições que afetam o fluxo do processo. Esta heurística busca identificar regras de negócio e lógicas condicionais que o sistema deve avaliar e aplicar.

Heurística de Elicitação 4: Eventos de Mensagem

- Eventos de mensagem descrevem comunicações entre participantes do processo. Esta heurística permite identificar requisitos de integração, como envio de e-mails, notificações, trocas entre sistemas (APIs) ou formulários eletrônicos.

Heurística de Elicitação 5: Evento Condicional

- Este evento ocorre quando uma condição externa se torna verdadeira. A heurística orienta a identificação de requisitos relacionados à monitoramento de estados ou eventos externos ao sistema, como mudanças em um banco de dados ou entrada de informações de outro sistema.

Heurística de Elicitação 6: Eventos de Temporizador

- Eventos de temporizador indicam restrições temporais, como atrasos, prazos ou execuções agendadas. A partir dessa heurística, extraem-se requisitos não funcionais relacionados a tempo de execução, agendamento e performance.

Heurística de Elicitação 7: Demais Eventos

- Engloba eventos menos comuns, como eventos de erro, término, escalonamento e cancelamento. Essa heurística identifica comportamentos alternativos que o sistema deve gerenciar.

Heurística de Elicitação 8: Objetos de Dados e/ou de Armazenamento

- Foca na análise de dados de entrada e saída utilizados ou gerados nas atividades. Essa heurística contribui para a definição de requisitos relacionados à manipulação, validação, persistência e segurança dos dados.

Heurística de Elicitação 9: Comentários Adicionais

- Elementos como anotações, observações ou comentários em diagramas também podem conter informações relevantes para a elicitação, mesmo que não estejam formalmente representadas como tarefas ou eventos.

Essas heurísticas, no contexto da metodologia ERAPRO (Engenharia de Requisitos Apoiada na Modelagem de Processos de Negócio), têm como objetivo fortalecer o processo de elicitação, promovendo uma abordagem mais sistemática, completa e rastreável. Ao aplicar essas heurísticas, o analista aumenta a assertividade na coleta de requisitos e reduz a ambiguidade e o retrabalho durante as fases subsequentes do desenvolvimento [71].

Diante do exposto, observa-se que a aplicação de heurísticas no contexto da modelagem de processos de negócio, representa uma evolução metodológica importante no processo de elicitação de requisitos, conferindo maior estrutura, rastreabilidade e alinhamento entre os objetivos organizacionais e as funcionalidades do sistema. No entanto, é igualmente relevante compreender como a elicitação de requisitos é conduzida em abordagens mais tradicionais de desenvolvimento de software, onde a ênfase recai sobre modelos sequenciais, planejamento detalhado e forte documentação. Nesse sentido, a próxima seção aborda os métodos tradicionais de desenvolvimento, como o modelo cascata (*Waterfall*) e o modelo espiral, evidenciando as principais técnicas utilizadas para elicitar requisitos nesses contextos e analisando suas contribuições e limitações no processo de definição de sistemas de software.

3.6.4 Elicitação de requisitos utilizando método tradicional

A elicitação de requisitos, embora reconhecidamente uma atividade crítica em qualquer abordagem de engenharia de software, assume contornos específicos conforme o modelo de desenvolvimento adotado. Entre os modelos mais representativos das abordagens tradicionais, destacam-se o modelo em cascata (Waterfall) e o modelo espiral, ambos amplamente discutidos na literatura clássica da área. Essas metodologias são caracterizadas por uma sequência de fases bem definidas — como levantamento de requisitos, análise, projeto, implementação, testes e manutenção — com forte ênfase em documentação extensiva, planejamento detalhado e controle rigoroso das etapas [59]

O Modelo cascata de Royce foi o primeiro modelo de software a ser desenvolvido em 1970. Também conhecida como metodologia cascata (*waterfall*) ou modelo sequencial, é uma abordagem de desenvolvimento de software linear e sequencial. Ela é uma das mais antigas metodologias, caracterizada por uma série de fases que fluem rigidamente uma após a outra, como uma cascata [72].

No modelo em cascata, a fase de elicitação e análise de requisitos é considerada a fundação sobre a qual todas as fases seguintes se baseiam. Uma vez concluída, essa fase não deve ser revista, a menos que haja replanejamento formal — o que implica um desafio para projetos que envolvem requisitos voláteis. Esse modelo pressupõe que os requisitos sejam claramente definidos desde o início do projeto, o que raramente ocorre em sistemas complexos ou inovadores.

Já o modelo espiral, introduz uma perspectiva iterativa, na qual os riscos técnicos e gerenciais são avaliados continuamente ao longo do projeto. Nesse modelo, a elicitação ocorre em ciclos, permitindo a revisão e o refinamento dos requisitos ao longo do tempo, embora ainda mantenha a estrutura rígida típica dos métodos tradicionais [73].

As etapas tradicionais, frequentemente baseadas no modelo de cascata, incluem:

- Definição de Requisitos: Entendimento das necessidades e expectativas do usuário. Documentação dos requisitos funcionais e não funcionais do software. Coleta de requisitos por meio de entrevistas, questionários, observações e workshops.
- Análise de Requisitos: Avaliação aprofundada dos requisitos coletados. Transformação desses requisitos em especificações claras e detalhadas. Identificação de possíveis desafios ou ambiguidades nos requisitos. Etapa mais complexa no desenvolvimento de software.
- Projeto (Design): Definição da arquitetura do software. Criação de diagramas de design, modelos de dados e interfaces. Escolha das tecnologias e ferramentas a serem utilizadas.

- Implementação (Codificação): Escrita do código-fonte do software. Tradução das especificações de design em código executável. Uso de linguagens de programação e ferramentas adequadas ao projeto.
- Testes: Verificação da funcionalidade, confiabilidade e performance do software. Identificação e correção de erros ou bugs. Realização de testes unitários, de integração, de sistema, entre outros.
- Integração: Combinação de diferentes partes ou módulos do software em um sistema coeso. Resolução de conflitos ou inconsistências entre os módulos.
- Implantação (Deployment): Instalação e configuração do software no ambiente do usuário final. Treinamento dos usuários e suporte inicial após a implantação.
- Manutenção e Suporte: Correção de bugs ou erros identificados após a implantação. Atualizações para atender a novos requisitos ou melhorar a performance. Suporte contínuo aos usuários.
- Desativação (caso aplicável): Encerramento do uso do software. Migração de dados e transição para novos sistemas, se necessário.

No modelo tradicional, a fase de levantamento de requisitos é decisiva e ocorre logo no início do processo. Nesse momento, os requisitos são definidos de forma detalhada, amplamente documentados e essa documentação serve de base para todas as etapas seguintes. A coleta ocorre por meio de reuniões com stakeholders, análise de documentos existentes e outras técnicas.

Uma característica central desse método é sua natureza sequencial: cada fase deve ser concluída antes do início da próxima. Além disso, ele é fortemente orientado à documentação, produzindo diversos artefatos mesmo antes do início do desenvolvimento. Por não ser iterativo, raramente permite retornar a fases anteriores.

Nesse contexto, o método tradicional utiliza diferentes técnicas de elicitação para compreender o que o sistema deve fazer e como deve se comportar. A seguir, apresentam-se algumas das técnicas mais comuns [59].

1. Entrevistas:

- Descrição: Conversas diretas com stakeholders para identificar necessidades e expectativas.
- Aplicação: Conduzir entrevistas estruturadas (com perguntas definidas) ou não estruturadas (mais abertas) com usuários, clientes ou demais envolvidos.

2. Questionários e Pesquisas:

- Descrição: Uso de questionários para obter informações de muitos participantes.
- Aplicação: Elaborar questionários direcionados e enviá-los aos stakeholders, especialmente quando o contato direto é difícil ou o público é amplo.

3. Revisão de Documentação:

- Descrição: Análise de documentação existente para entender os requisitos atuais e futuros.
- Aplicação: Inclui a revisão de manuais de operação, relatórios de erros, requisitos de sistemas antigos etc.

4. Workshops e Reuniões de Grupo:

- Descrição: Sessões colaborativas com vários stakeholders para discutir requisitos.
- Aplicação: Organizar sessões de brainstorming ou workshops de requisitos para coletar ideias e requisitos de vários participantes simultaneamente.

5. Análise de Modelos de Negócios:

- Descrição: Compreender e analisar o modelo de negócios para identificar requisitos do sistema.
- Aplicação: Analisar como o negócio opera e como o sistema proposto se encaixará nas operações existentes.

6. Prototipagem:

- Descrição: Criação de protótipos iniciais do sistema para coletar feedback e refinar requisitos.
- Aplicação: Desenvolver protótipos simples para ajudar os stakeholders a visualizar o produto final e coletar feedback.

7. Análise de Cenários e Casos de Uso:

- Descrição: Desenvolvimento de cenários ou casos de uso para entender como o sistema será usado.
- Aplicação: Criar narrativas detalhadas descrevendo como os usuários interagem com o sistema em diferentes situações.

8. Observação e Shadowing:

- Descrição: Observar os usuários em seu ambiente natural para entender como eles trabalham e quais são suas necessidades.
- Aplicação: Passar tempo com os usuários enquanto eles realizam suas tarefas diárias para identificar requisitos não expressos.

Essas técnicas são aplicadas com o objetivo de produzir um documento de requisitos de software (Software Requirements Specification – SRS), geralmente estruturado conforme normas como a IEEE 830 ou a ISO/IEC/IEEE 29148:2018, que visam garantir completude, clareza, consistência e rastreabilidade.

As principais vantagens e desvantagens do método tradicional cascata no que se refere a fase de elicitação de requisitos:

- Vantagens do Método Cascata:
 - Simplicidade e Facilidade de Entendimento: Seu modelo linear é fácil de entender e implementar, especialmente para equipes menos experientes.
 - Documentação Detalhada: Enfatiza a documentação completa em cada fase, o que pode ser útil para rastreabilidade e referência futura.
 - Definição Clara de Etapas: Cada fase tem objetivos e resultados bem definidos, o que facilita o planejamento e a gestão.
 - Facilidade de Gerenciamento: Como as fases são realizadas uma após a outra, a gestão do projeto torna-se mais direta.
 - Previsibilidade: Permite estimativas de custo e tempo relativamente estáveis desde o início do projeto.
 - Especialização de Tarefas: Cada fase requer habilidades específicas, permitindo que os membros da equipe se especializem em diferentes áreas do desenvolvimento de software.
- Desvantagens do Método Cascata:
 - Inflexibilidade a Mudanças: Uma vez que uma fase é concluída, voltar atrás para fazer mudanças é difícil e muitas vezes custoso.
 - Atraso no Feedback do Usuário: O software só é testado após a sua completa construção, o que retarda o feedback do usuário e a detecção de problemas.

- Risco de Não Atender às Necessidades do Usuário: Os requisitos do usuário podem mudar ao longo do projeto, mas o modelo Cascata não se adapta bem a essas mudanças.
- Alto Risco e Incerteza: Problemas e erros podem não ser descobertos até as fases finais, aumentando os riscos do projeto.
- Inadequado para Projetos Grandes ou Complexos: Em projetos grandes, os requisitos podem ser difíceis de definir completamente no início, tornando o método Cascata inadequado.
- Longo Tempo de Entrega: O produto final só é entregue após a conclusão de todas as fases, o que pode levar muito tempo.
- Desperdício de Recursos: Se os requisitos estiverem mal definidos inicialmente, o trabalho feito nas fases posteriores pode se tornar obsoleto, resultando em desperdício de tempo e recursos.

Embora os métodos tradicionais tenham desempenhado um papel fundamental na consolidação das boas práticas de engenharia de software, seus limites quanto à flexibilidade, adaptabilidade e resposta a mudanças tornaram-se evidentes ao longo do tempo, principalmente em projetos com alto grau de incerteza, ciclos de entrega longos e envolvimento limitado dos usuários durante o desenvolvimento. A dificuldade em adaptar os requisitos às mudanças do mercado, aliada ao elevado custo de retrabalho e à insatisfação frequente dos clientes com os produtos entregues, motivou profissionais da área a repensarem o modelo tradicional de desenvolvimento.

Foi nesse contexto que surgiu, em 2001, o Manifesto Ágil, resultado da reunião de dezessete especialistas em desenvolvimento de software, que buscavam alternativas mais flexíveis, colaborativas e voltadas à entrega contínua de valor. O manifesto propôs quatro valores fundamentais e doze princípios que passaram a orientar uma nova forma de desenvolver software, baseada em interações humanas, feedback contínuo e adaptação constante [74].

Como consequência, a elicitação de requisitos também passou a ser tratada de forma diferente, abandonando a rigidez e o foco em documentação pesada em favor de técnicas iterativas e incrementais, como User Stories, Product Backlog, entrevistas informais e colaboração contínua com o cliente. Assim, a próxima seção trata da Elicitação de Requisitos no contexto das Metodologias Ágeis, explorando como os valores ágeis influenciam a forma de identificar e priorizar as necessidades dos stakeholders em ciclos curtos de desenvolvimento.

3.6.5 Elicitação de requisitos no contexto das metodologias ágeis

Com a consolidação das metodologias ágeis após o Manifesto Ágil o desenvolvimento de software passou a priorizar interação contínua com stakeholders, entregas incrementais e rápida adaptação a mudanças. Esse novo paradigma transformou profundamente o tratamento dos requisitos, especialmente quanto à elicitação, documentação e validação.

Diferentemente dos métodos tradicionais, em que a elicitação ocorre de forma inicial e altamente documentada, nas metodologias ágeis os requisitos são definidos de maneira iterativa, evolutiva e centrada no cliente. Isso ocorre porque, na prática, muitos stakeholders não têm plena clareza de suas necessidades no início do projeto, além de essas necessidades mudarem ao longo do desenvolvimento [27].

O software tornou-se essencial em praticamente todos os setores, e cada organização possui particularidades quanto ao desenvolvimento e à entrega de soluções tecnológicas. Em muitas empresas, especialmente no setor de tecnologia, é comum o desenvolvimento de sistemas de grande escala que precisam ser entregues totalmente finalizados, com alta qualidade e atendendo plenamente às expectativas dos clientes [75].

Entre 1999 e 2001, começaram a surgir os primeiros conceitos de métodos ágeis de desenvolvimento. Um exemplo é o Framework SCRUM que propunha uma abordagem mais simplificada para a documentação de requisitos funcionais, como histórias de usuário, que se baseia em entregas incrementais de software em períodos curtos de até quinze dias.[76] As metodologias ágeis priorizam a comunicação constante com o cliente, a colaboração entre os membros da equipe e a valorização da entrega de software funcional acima da documentação extensa. A engenharia de requisitos em ambientes ágeis é orientada por uma abordagem *just enough*, ou seja, captura-se apenas o necessário para iniciar o trabalho e os requisitos vão sendo refinados à medida que o produto evolui [27].

No Scrum, por exemplo, a elicitação ocorre principalmente por meio da manutenção e refinamento contínuo do Product Backlog, onde os requisitos são representados como *User Stories* que são pequenas descrições de funcionalidades, ou seja, a menor granularidade da funcionalidade sob a perspectiva do usuário. Essas histórias uma vez validadas com os clientes, são discutidas entre o *Product Owner* e o time de desenvolvimento durante eventos como a *Sprint Planning* e reuniões diárias, possibilitando ajustes constantes com base no feedback recebido [1].

Dentro do contexto das metodologias ágeis, especialmente em frameworks como *Scrum*, observa-se uma distinção importante entre os processos de desenvolvimento propriamente ditos (*Downstream*) e os momentos de descoberta, análise e alinhamento inicial de produto, conhecidos como *Upstream*. A fase de *upstream* antecede a execução formal das sprints e tem como objetivo garantir que os itens priorizados no backlog estejam bem

compreendidos, alinhados ao valor de negócio e suficientemente refinados para que a equipe possa desenvolvê-los com clareza e confiança [77].

O trabalho de upstream envolve atividades como:

- Compreensão do problema ou oportunidade de negócio;
- Descoberta e mapeamento de hipóteses de valor;
- Análise de stakeholders e definição de personas;
- Estruturação inicial do Product Backlog com epics e features;
- Priorização baseada em valor e viabilidade técnica.

Esse tipo de abordagem ajuda a mitigar os riscos associados a requisitos mal compreendidos, comuns em projetos que operam com sprints mal preparados ou excesso de retrabalho por falta de clareza nos objetivos.

Além disso, o upstream reforça a importância da colaboração entre stakeholders de negócio, product owners, desenvolvedores e usuários finais antes do início da codificação.

Algumas das principais técnicas de descoberta incluem Lean Inception, Design Thinking, PBB (Product Backlog Building) dentre outras. No que tange as principais técnicas, apresenta-se a visão geral:

- **Lean Inception:** Lean Inception é uma técnica colaborativa para alinhar rapidamente a equipe em torno de uma visão compartilhada do produto e definir os primeiros passos do projeto. Durante uma Lean Inception, os participantes trabalham juntos para identificar os objetivos do produto, definir as personas de usuários, elaborar o MVP (Minimum Viable Product) e priorizar os primeiros itens do backlog do produto. Essa técnica ajuda a garantir um alinhamento claro entre a equipe e os stakeholders desde o início do projeto.
- **Design Thinking:** Design Thinking é uma abordagem centrada no ser humano que visa resolver problemas complexos, priorizando a compreensão das necessidades dos usuários finais. É um processo iterativo e colaborativo que envolve empatia, definição, ideação, prototipagem e teste. No contexto do desenvolvimento de software, o Design Thinking ajuda a identificar e compreender profundamente os problemas e necessidades dos usuários, facilitando a criação de soluções inovadoras e centradas no usuário.
- **Product Backlog Building (PBB):** PBB é uma técnica ágil para criar o backlog do produto em colaboração com a equipe de desenvolvimento e os stakeholders.

Durante as sessões de PBB, os participantes identificam e priorizam os requisitos do sistema, descrevendo as funcionalidades necessárias para o sucesso do projeto. Isso ajuda a garantir que o backlog do produto seja compreensivo, priorizado e alinhado com a visão do produto [78].

Algumas das técnicas mais utilizadas na elicitação de requisitos em ambientes ágeis incluem:

- *User Stories*: Estruturadas geralmente na forma “Como [persona], eu quero [ação] para que [benefício]”, essas histórias ajudam a capturar de forma simples e centrada no valor funcional esperado pelo usuário.
- Critérios de aceitação: Associados às User Stories, definem claramente o que deve ser cumprido para que uma funcionalidade seja considerada “pronta”, atuando como elemento de validação.
- Workshops colaborativos: Sessões de levantamento e refinamento de backlog com participação ativa do cliente e da equipe.
- Personas e cenários de uso: Utilizados para representar usuários típicos e suas interações com o sistema, auxiliando na compreensão de contextos e expectativas [59].
- Prototipagem rápida: Aplicada para obter feedback visual e antecipado sobre requisitos relacionados à interface e à usabilidade.
- Conversas e feedback contínuo: A comunicação frequente com stakeholders substitui a documentação detalhada como principal forma de captura e esclarecimento dos requisitos.

Além das técnicas mencionadas acima, existem várias outras técnicas na fase de descoberta que podem ser utilizadas no *upstream* do desenvolvimento de software, como: Entrevistas com os *stakeholders*, análise de documentos existentes, observação do usuário em seu ambiente natural, estudo de concorrência e análise de tendências de mercado. Essas técnicas ajudam a obter uma compreensão abrangente das necessidades do cliente, dos usuários finais e do contexto do mercado, informando assim o processo de desenvolvimento de *software*. Essas abordagens colaborativas e centradas no usuário são fundamentais para o desenvolvimento de produtos de software que realmente agregam valor aos usuários finais. Por se tratar de método ágil, seu processo é:

- Iterativo e Incremental: Em vez de definir todos os requisitos de uma vez, a elicitação é realizada em ciclos (sprints ou iterações).

- Colaborativa: Envolve uma estreita colaboração entre a equipe de desenvolvimento, o cliente e os usuários finais.
- Flexível: Os requisitos podem ser ajustados e refinados ao longo do projeto à medida que novas informações são descobertas e as necessidades dos usuários evoluem.

A fase de levantamento de requisitos é frequentemente iniciada com um processo de descoberta ou "*discovery*", que tem como principal objetivo compreender em profundidade o problema e as necessidades dos usuários. Esta fase envolve diversas atividades, tais como sessões de brainstorming, entrevistas com os usuários, workshops, além de ferramentas e técnicas como: *Lean Inception*, *Product Backlog Building - PBB* entre outras, focadas na descoberta e a observação do ambiente onde o usuário está inserido [79].

Durante o processo de *discovery*, a ênfase é colocada na identificação do valor que o software precisa oferecer e na definição do escopo mínimo viável (MVP). Além disso, busca-se um entendimento abrangente das necessidades, das funcionalidades planejadas, das regras de negócio, e realiza-se o mapeamento para o refinamento técnico e comercial. Este processo é fundamental para mitigar problemas potenciais, bem como para identificar e resolver impedimentos e dependências.

Para o levantamento de requisitos em métodos ágeis, uma ferramenta central é a "*User Story*" (ou história de usuário, em português). Esta técnica envolve a descrição de funcionalidades sob a perspectiva do usuário, facilitando tanto o entendimento quanto a comunicação eficaz dos requisitos.

Além da fase de *discovery*, a priorização de requisitos é realizada por meio de uma lista conhecida como "*Product Backlog*". Nesta, os requisitos e funcionalidades considerados mais importantes pelo cliente são listados no topo. Essa lista priorizada é essencial para planejar quais requisitos serão abordados em cada sprint.

Outro aspecto relevante na agilidade é o uso de personas - representações fictícias dos usuários finais. Elas são extremamente úteis para compreender e se concentrar nas necessidades dos usuários. A partir disso, cria-se o "Mapa de Jornada do Usuário", que oferece uma visão das etapas percorridas pelo usuário ao interagir com o produto, além de ajudar na identificação de pontos críticos de interação e requisitos [55].

Para apoiar visualmente o desenvolvimento, são criados protótipos e *wireframes* dos modelos iniciais do sistema, que permitem coletar feedbacks precoces. Esta prática é fundamental para validar ideias e requisitos de maneira ágil. Para a organização das user stories (estórias de usuários) é usado o *Story mapping* Story mapping, um mapa que representa o fluxo de trabalho do usuário. Ajuda a visualizar o sistema como um todo e a definir prioridades. Para realizar a mensuração de esforço/tempo de desenvolvimento das user story é utilizada técnicas de estimativa de esforço necessário para implementar cada requisito, promovendo o entendimento comum e a colaboração da equipe.

- Vantagens da utilização de métodos ágeis na fase de elicitação de requisitos [74]
- Alta adaptabilidade e rápida resposta a mudanças
- Entregas contínuas e incremento de valor
- Forte colaboração com o cliente
- Feedback rápido e refinamento contínuo
- Comunicação eficaz e foco nas pessoas
- Simplicidade e redução de desperdícios
- Desvantagens da utilização de métodos ágeis na fase de elicitação de requisitos:
- Documentação reduzida ou insuficiente
- Dependência de stakeholders ativos
- Requisitos voláteis e difíceis de estabilizar
- Pressão por entregas frequentes
- Necessidade de equipes maduras e disciplinadas
- Baixa adequação a grandes projetos sem frameworks complementares

Portanto, a fase de upstream, com suporte de técnicas como a Lean Inception, representa uma ponte estratégica entre a elicitação de requisitos e o desenvolvimento ágil de software, pois possibilita transformar hipóteses iniciais em requisitos compreensíveis, priorizados e com valor validado. Sua adoção reduz a lacuna entre o que os stakeholders desejam e o que a equipe efetivamente constrói, favorecendo entregas mais assertivas e ciclos de feedback mais eficazes.

3.6.6 Técnicas e Métodos de elicitação de requisitos existentes

No contexto da engenharia de requisitos, uma técnica ou modelo de elicitação refere-se a um conjunto de práticas, procedimentos ou abordagens sistemáticas utilizadas para identificar, compreender e documentar os requisitos de um sistema de software. Essas técnicas podem envolver entrevistas, observação, modelagem de processos, uso de artefatos visuais, coleta de feedbacks ou colaboração direta com os stakeholders.

Técnicas de elicitação são fundamentais para permitir que o engenheiro de requisitos obtenha uma visão clara das necessidades do usuário e compreenda o domínio do problema antes da construção do sistema. A escolha da técnica adequada está diretamente relacionada ao contexto do projeto, à natureza da organização e à maturidade da equipe.

- Atividades Exploratórias: Visam descobrir novas informações, entender o problema ou levantar hipóteses sobre o sistema.
- Atividades Colaborativas: Envolvem a interação entre diversos participantes para consolidar, validar ou priorizar requisitos.
- Atividades Observacionais: Consistem em observar o comportamento de usuários no ambiente real para inferir necessidades e problemas.
- Atividades Analíticas: Caseiam-se na interpretação de dados existentes, documentos, sistemas legados ou modelos já definidos.
- Atividades de Modelagem: Constroem representações visuais dos requisitos, processos, funcionalidades ou interações com o usuário.

Essas atividades são essenciais para garantir que os requisitos elicitados sejam completos, consistentes, relevantes e compreensíveis, reduzindo os riscos de retrabalho e falhas no produto final.

A elicitação de requisitos pode ser vista como um processo de aprendizado, que envolve um conjunto estruturado de atividades para descobrir o que os stakeholders realmente precisam do sistema [27]

Com base em tudo o que discutimos até agora, e considerando a literatura clássica, atualmente é possível identificar pelo menos 26 métodos e técnicas de elicitação de requisitos amplamente reconhecidos e aplicados na prática.

Essas técnicas podem ser classificadas conforme o contexto (tradicional, ágil ou híbrido) e conforme o tipo de atividade, sendo: Exploratória, colaborativa, observacional, analítica, modelagem etc, conforme apresentado na Figura 3.11.

Métodos e Técnicas de Elicitação de Requisitos			
Método/Técnica	Contexto de uso	Descrição	Tipo de Atividade
Análise de documentos	• Tradicional	• Revisa documentos existentes para levantar requisitos e restrições.	• Analítica
Questionários/Pesquisas	• Tradicional	• Obtém dados quantitativos ou qualitativos em larga escala.	• Exploratória
Caso de Uso (Use Case)	• Tradicional	• Descreve interações entre usuário e sistemas em fluxos de funcionalidade.	• Modelagem
Revisão de sistemas existentes	• Tradicional	• Avaliação de funcionalidades em sistemas legados ou concorrentes	• Analítica
Delphi	• Tradicional	• Técnica de consulta estruturada a especialistas para consenso em requisitos.	• Exploratória
Análise de Tarefas	• Tradicional	• Estudo detalhado de tarefas para compreender ações e requisitos dos usuários.	• Observacional / Analítica
Análise SWOT	• Tradicional	• Avaliação de forças, fraquezas, oportunidades e ameaças relacionadas ao projeto	• Analítica
Benchmarking	• Tradicional	• Comparação com produtos similares para identificar competitivos.	• Analítica
Lean Inception	• Ágil	• Técnica estruturada para definição colaborativa do MVP com foco em alinhamento e valor.	• Colaborativa / Modelagem
User Stories	• Ágil	• Narrativas curtas que representam funcionalidades do ponto de vista do usuário.	• Colaborativa / Modelagem
Backlog Refinement	• Ágil	• Refinamento contínuo dos itens do backlog com apoio do time e do Product Owner.	• Colaborativa / Analítica
Personas e Cenários	• Ágil	• Perfis e histórias que representam usuários e seus contextos de uso.	• Modelagem / Exploratória
Critérios de Aceitação	• Ágil	• Condições objetivas que definem quando uma funcionalidade está pronta.	• Colaborativa / Analítica
Reunião de Planejamento de Sprint	• Ágil	• Define, em equipe, o escopo da sprint com base no backlog priorizado.	• Colaborativa
Story Mapping	• Ágil	• Organiza visualmente funcionalidades em jornadas para priorização incremental.	• Modelagem / Colaborativa
Shadowing	• Ágil	• Acompanhamento direto do usuário enquanto ele interage com o sistema.	• Observacional
Reuniões de Visão de Produto	• Ágil	• Sessões para alinhamento estratégico do produto com stakeholders.	• Colaborativa / Exploratória
Mapa de Jornada do Usuário	• Ágil	• Representa visualmente a experiência do usuário em etapas, revelando requisitos.	• Modelagem / Observacional
Heurísticas com BPMN (REMO)	• Híbrido	• Aplicação de heurísticas sobre diagramas de processos para extrair requisitos.	• Analítica / Modelagem
Entrevista	• Híbrido	• Coleta de requisitos por meio de conversas estruturadas com stakeholders.	• Exploratória
Observação Direta (Etnografia)	• Híbrido	• Observa usuários no ambiente real para identificar necessidades implícitas.	• Observacional
Prototipagem	• Híbrido	• Cria versões preliminares da interface ou funcionalidades para validação.	• Modelagem / Colaborativa
Workshops colaborativos	• Híbrido	• Reuniões facilitadas para levantamento e validação de requisitos em grupo.	• Colaborativa
Modelagem de Processos de Negócio	• Híbrido	• Mapeia visualmente os processos da organização, identificando requisitos associados.	• Modelagem
Análise de Stakeholders	• Híbrido	• Identifica as partes interessadas e analisa sua influência no projeto.	• Analítica / Exploratória
Brainstorming	• Híbrido	• Técnica de geração de ideias rápidas para levantamento inicial de requisitos.	• Exploratória / Colaborativa

Figura 3.12: Técnicas de Levantamento de Requisitos

Fonte: Elaborada pela autora (2025)

Diante da diversidade de técnicas e modelos de elicitação apresentados, torna-se evidente que a escolha da abordagem mais adequada depende do contexto organizacional, do nível de incerteza do projeto e do grau de maturidade da equipe. Nesse cenário, a

crescente adoção de metodologias ágeis em ambientes de desenvolvimento de software tem impulsionado o uso de práticas mais iterativas, colaborativas e orientadas à entrega contínua de valor. Entre essas abordagens, o método ágil SCRUM se destaca como um dos frameworks mais consolidados e amplamente utilizados no mercado que será abordado na seção 3.6.7.

3.6.7 Framework SCRUM

O Scrum é um framework ágil utilizado para desenvolver produtos complexos, especialmente software, por meio de ciclos iterativos e incrementais. Criado por Ken Schwaber e Jeff Sutherland no início dos anos 1990 e formalizado no Scrum Guide em 2010, ele promove colaboração, transparência, inspeção contínua e adaptação rápida ao longo do desenvolvimento [1].

O Scrum é um Framework que foi associado ao desenvolvimento pela primeira vez por Hirotaka Takeuchi e Ikujiro Nonaka no livro *“The New Product Development Game”* (1986). Os autores enaltecem a importância de se adotar uma forma de desenvolvimento onde toda a equipe trabalhe como uma unidade para atingir um objetivo comum. Em 1995, Ken Schwaber formalizou o Scrum para projetos de desenvolvimento de software baseado no sistema Lean da Toyota, na conferência Oopsla em Austin no Texas.

O Scrum é um framework ágil para desenvolver e sustentar produtos complexos. Baseia-se no empirismo, que defende que o conhecimento é adquirido por meio da experiência e da tomada de decisões informadas, e na abordagem de desenvolvimento incremental, que promove ciclos curtos de aprendizado e entrega contínua de valor.

O Scrum se fundamenta em cinco valores essenciais que orientam o comportamento e a interação das pessoas dentro da equipe. São eles:

- **Comprometimento** - As pessoas no Scrum se comprometem com as metas da equipe, assumindo responsabilidade coletiva pelos resultados e trabalhando de forma dedicada para alcançar os objetivos do Sprint e do produto.
- **Coragem** - Os membros da equipe têm coragem para fazer o que é certo, enfrentar desafios, assumir riscos conscientes, lidar com problemas difíceis e trabalhar de forma transparente, mesmo diante de incertezas.
- **Foco** - Todos concentram seus esforços nas metas do Sprint e do Product Goal. O Scrum cria condições para que a equipe mantenha foco nas entregas, limitando distrações e permitindo um trabalho consistente e de qualidade.

- Abertura e respeito - A equipe e os stakeholders concordam em estar abertos sobre o trabalho, os desafios e o progresso. Esse valor incentiva a transparência de informações, a troca sincera de perspectivas e a aceitação de novas ideias.

O Scrum possui valores, mas também possui 3 pilares, conforme apresentado na figura 3.13:

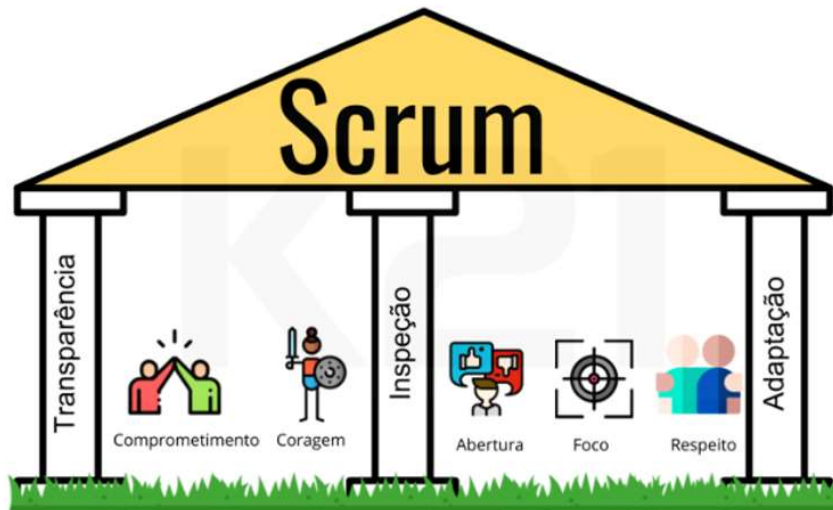


Figura 3.13: Pilares do SCRUM Fonte: Guia Scrum [1].

- Transparência: todos os aspectos significativos do processo devem ser visíveis e compreendidos por todos.
- Inspeção: artefatos e progresso devem ser verificados regularmente para identificar desvios.
- Adaptação: ajustes devem ser feitos rapidamente quando a inspeção identifica problemas.

Esses pilares sustentam todo o framework e orientam a forma como a equipe trabalha, garantindo visibilidade do progresso, identificação rápida de desvios e ajustes contínuos ao longo do desenvolvimento. É justamente para colocar esses pilares em prática que o Scrum organiza o trabalho em ciclos iterativos e incrementais chamados Sprints. A Sprint é um período de tempo fixo — geralmente de duas a quatro semanas — durante o qual a equipe trabalha para atingir um conjunto definido de objetivos. Ao final desse ciclo, deve ser produzido um incremento potencialmente entregável, ou seja, uma parte do produto que esteja realmente pronta: testada, funcional, integrada e com qualidade suficiente para, em teoria, ser entregue ao usuário final ou ao cliente. [80].

São alguns pontos-chave sobre a Sprint no Scrum:

- **Duração Fixa:** A Sprint possui duração fixa, definida antecipadamente e mantida igual em todas as Sprints do projeto — por exemplo, se a equipe escolhe três semanas, todas as Sprints seguirão esse mesmo período.
- **Objetivos Claros:** No início de cada Sprint, durante o Planejamento da Sprint, a equipe define um objetivo para a Sprint, que descreve o que a equipe espera alcançar até o final dela.
- **Sem Mudanças Drásticas:** Uma vez que uma Sprint começa, o seu escopo (os itens selecionados do Product Backlog) idealmente não deve ser alterado. Isso permite que a equipe tenha estabilidade para se concentrar no trabalho comprometido.
- **Incremento Potencialmente Entregável:** É uma parte essencial do Scrum. É o resultado concreto ao final de cada Sprint e representa um subconjunto do produto final que é funcional, testado e de alta qualidade. Vamos detalhar mais sobre isso:

Ainda no Scrum temos os ritos (ou cerimônias) que são eventos prescritos que têm a finalidade de criar regularidade e minimizar a necessidade de reuniões não especificadas. Os principais ritos do Scrum são:

- **Planejamento da Sprint (Sprint Planning):** No início de cada Sprint, a equipe se reúne para planejar o trabalho a ser realizado na próxima Sprint.
- **Daily Scrum (Reunião Diária):** É uma reunião diária de curta duração (geralmente 15 minutos) em que a equipe de desenvolvimento discute o progresso em relação à Sprint e planeja o trabalho das próximas 24 horas.
- **Revisão da Sprint (Sprint Review):** No final de cada Sprint, a equipe de desenvolvimento e o Product Owner se reúnem para revisar o Incremento de Produto resultante. É uma oportunidade para inspecionar o que foi feito, demonstrar o trabalho realizado e obter feedback.
- **Retrospectiva da Sprint (Sprint Retrospective):** Após a Revisão da Sprint e antes do próximo Planejamento da Sprint, a equipe se reúne para discutir o que correu bem, o que pode ser melhorado e como implementar essas melhorias na próxima Sprint.
- **Refinamento do Backlog (Backlog Refinement):** Embora não seja uma cerimônia oficial no Guia Scrum, muitas equipes adotam a prática de refinar regularmente seu Product Backlog. Isso envolve analisar, detalhar e estimar itens do backlog para que estejam prontos para futuras Sprints.

O objetivo desses ritos é garantir transparência, inspeção e adaptação ao longo do processo de desenvolvimento. Ao seguir essas cerimônias, as equipes são capazes de manter um ritmo sustentável, adaptar-se às mudanças e entregar valor de maneira contínua.

O Scrum implementa um esqueleto interativo e incremental por meio de papéis e responsabilidades descritas a seguir:

No Scrum, o Product Owner representa os stakeholders, entende profundamente o negócio e garante o ROI, sendo responsável por priorizar e gerenciar o Product Backlog. O Scrum Master atua como facilitador, assegurando que o framework seja aplicado corretamente, removendo impedimentos e protegendo a equipe de interferências para manter a produtividade. Já a Equipe de Desenvolvimento é multifuncional e colaborativa, responsável por transformar itens do backlog em incrementos prontos, sem depender de papéis rígidos, garantindo continuidade e qualidade no desenvolvimento.

Comumente, uma equipe Scrum é composta por 5 a 9 integrantes, embora existam relatos de projetos bem-sucedidos com grupos maiores. A equipe mantém-se comprometida e focada nas atividades designadas. Se surgirem obstáculos que comprometam o progresso, eles são identificados e comunicados ao Scrum Master para resolução.

No Scrum, os Stakeholders são todos os interessados no software que está em desenvolvimento a começar pelo cliente (contratante), usuários finais, equipe de marketing e vendas, entre outros e são representados pelo Product Owner. O Scrum apela a todos os interessados em fornecer um feedback frequente em toda a sprint [1].

Isto produz um produto que adere mais estreitamente com as necessidades do cliente, ou seja, todos possuem um senso de propriedade. A responsabilidade pela entrega do produto é de toda a equipe, independente de papéis. Caso ocorra alguma falha durante o Sprint, todos são responsáveis pelo seu fracasso.

Fluxo de Processo Segundo Fonseca (2009), o Scrum divide um projeto em interações (duas a quatro semanas no máximo), chamadas de Sprints, onde cada perfil possui seu papel e suas responsabilidades, conforme figura 3.13 [81].

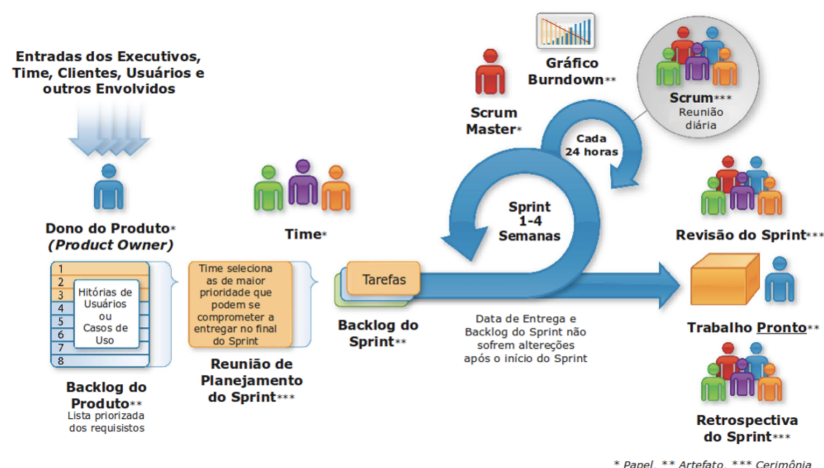


Figura 3.14: Processo SCRUM [1].Fonte: Scrum.org

No início de cada Sprint, o time define, junto ao Product Owner, quais funcionalidades do Product Backlog serão desenvolvidas, considerando sua prioridade. As Sprints duram entre 1 e 4 semanas, e incluem reuniões diárias (Daily Meetings) para acompanhar o progresso. O Scrum promove uma transformação cultural ao incentivar novas formas de pensar e agir, com foco em colaboração, adaptação e entrega contínua de valor.

Nesse contexto, a elicitação de requisitos se beneficia de ciclos curtos e colaboração constante. Para garantir alinhamento e foco desde o início, é fundamental definir o Produto Mínimo Viável (MVP). A técnica Lean Inception se destaca como abordagem estratégica para estruturar o backlog inicial, promovendo o alinhamento entre negócio, tecnologia e usuários.

A *Lean Inception* é um método utilizado para a obtenção MVP's (Produtos Mínimos Viáveis.), que foi desenvolvida por Paulo Caroli em meados de 2010, um especialista em agilidade e fundador da consultoria Agilebrazil. Esse método tem sido amplamente adotado em projetos ágeis para auxiliar na fase inicial de um projeto, especialmente na definição e elicitação de requisitos [82].

A história da Lean Inception remonta ao desejo de Paulo Caroli de criar uma abordagem ágil que permitisse iniciar um projeto com um entendimento compartilhado e alinhado entre todas as partes envolvidas, incluindo a equipe de desenvolvimento, stakeholders, e outras partes interessadas. A técnica foi desenvolvida para superar desafios comuns encontrados em projetos de software, como a falta de clareza nos requisitos, a falta de engajamento dos stakeholders e o risco de entregar um produto que não atenda às necessidades do negócio.

A Lean Inception é projetada para ser uma atividade colaborativa e envolvente que ocorre em um curto período de tempo, geralmente entre uma semana e dez dias. Ela

envolve a participação de uma equipe multifuncional que inclui desenvolvedores, designers, testadores, gerentes de produto, entre outros. Durante esse período, a equipe trabalha intensamente para alcançar os seguintes objetivos:

- Definir a Visão do Produto: A equipe colabora para criar uma visão clara e compartilhada do produto que estão prestes a desenvolver. Isso inclui a identificação dos principais objetivos, metas e benefícios esperados.
- Priorizar Features: As funcionalidades (features) são identificadas e priorizadas com base no valor que agregam ao negócio. Essa priorização ajuda a equipe a focar nos elementos mais importantes do produto.
- Definir Personas e Jornadas: Personas (representações fictícias de usuários) são criadas para ajudar a equipe a entender as necessidades e comportamentos dos usuários. Jornadas do usuário são mapeadas para compreender como os usuários interagirão com o produto.
- Estruturar o MVP: O *Minimum Viable Product* (MVP) é definido. O MVP é a versão mais enxuta do produto que oferece valor suficiente para ser lançada. Ele serve como um guia para o desenvolvimento inicial.
- Identificar Riscos e Desafios: Possíveis obstáculos e riscos são identificados e discutidos, permitindo que a equipe se prepare para enfrentá-los.
- Estabelecer Métricas de Sucesso: Métricas para medir o sucesso do produto são definidas, ajudando a equipe a acompanhar o progresso e a eficácia do produto ao longo do tempo.

A *Lean Inception* é valiosa na etapa de elicitación de requisitos porque ajuda a criar um ambiente colaborativo e alinhado desde o início do projeto. Ela promove a compreensão mútua entre a equipe de desenvolvimento e os *stakeholders*, reduzindo a ambiguidade e o risco de mal-entendidos. Além disso, a técnica enfatiza a entrega de valor precoce por meio da priorização de recursos essenciais, garantindo que o desenvolvimento se concentre nas funcionalidades mais importantes para o negócio [82].

A *Lean Inception*, combinada com a técnica de *Product Backlog Building* (PBB), revela-se extremamente eficaz para a elicitación de requisitos em projetos ágeis. Enquanto a *Lean Inception* foca em definir uma visão compartilhada, priorizar funcionalidades, mitigar riscos e estabelecer uma base sólida para o desenvolvimento de software, o PBB complementa esse processo ao ajudar na construção estruturada do *Product Backlog*.

O PBB envolve a identificação detalhada e a organização de itens do backlog do produto de uma maneira que alinha as expectativas dos *stakeholders* com os objetivos do

projeto. Essa abordagem garante que todos os requisitos críticos sejam considerados e priorizados de acordo com sua importância e impacto no sucesso do projeto.

Juntas, a Lean Inception e o PBB contribuem significativamente para a entrega bem-sucedida de produtos que atendam às necessidades dos clientes e do negócio. Essa combinação oferece um caminho claro para a definição de metas, facilitando a comunicação entre a equipe e os *stakeholders* e garantindo que o desenvolvimento esteja alinhado com as expectativas e requisitos do mercado. Ao integrar essas metodologias, as equipes ágeis podem desenvolver uma compreensão mais profunda das necessidades do projeto e criar um plano de ação eficaz para sua execução.

A técnica Product Backlog Building (PBB) - *Product Backlog Building* é uma abordagem que funciona principalmente no contexto de refino de backlog em métodos ágeis, como o Scrum. Ela se concentra em melhorar e manter o Product Backlog de forma eficiente, garantindo que as histórias de usuário e os itens de backlog estejam bem definidos, priorizados e prontos para serem implementados nas próximas iterações [83].

O *Product Backlog Building* (PBB) ocorre da seguinte maneira:

1. Preparação: A equipe de desenvolvimento, incluindo o Product Owner, Scrum Master e membros da equipe, se reúne para a sessão de refinamento de backlog.
2. Revisão do Backlog Existente: Inicialmente, a equipe revisa o Product Backlog existente, que contém histórias de usuário e itens a serem entregues no projeto.
3. Identificação de Histórias de Usuário: Durante a reunião, a equipe identifica histórias de usuário adicionais ou requisitos que podem ser incluídos no Product Backlog. Isso pode envolver feedback de partes interessadas, mudanças nos requisitos do projeto ou novas ideias que surgiram desde a última sessão de refinamento.
4. Descrição Detalhada: O refinamento do backlog consiste em detalhar cada nova história de usuário para garantir sua clareza e completude, incluindo critérios de aceitação, exemplos e eventuais aspectos técnicos. A equipe, junto ao Product Owner, revisita e ajusta as prioridades do backlog conforme a importância atual do projeto. Quando necessário, são realizadas estimativas de esforço ou complexidade para apoiar o planejamento das próximas iterações. Também podem ser discutidos detalhes técnicos e, caso a história seja grande, ela pode ser dividida em tarefas menores. Todas as decisões são registradas no Product Backlog, mantendo-o atualizado e confiável. Ao final do processo, as histórias bem definidas e priorizadas tornam-se prontas para implementação na próxima Sprint.

A técnica PBB é valiosa porque ajuda a equipe a manter um backlog saudável e atualizado, garantindo que o trabalho mais importante seja priorizado e bem definido. Isso

contribui para uma execução mais suave das iterações, com menos incertezas e retrabalho. Além disso, a técnica promove uma colaboração eficaz entre o Product Owner, a equipe de desenvolvimento e outras partes interessadas, resultando em um alinhamento claro em relação aos objetivos do projeto e às necessidades dos usuários.[83]

Para tanto se faz necessária adoção de ferramenta para que esse controle e acompanhamento seja realizado de forma mais assertiva e eficaz.

O Design Thinking (DT) é uma abordagem amplamente reconhecida para a inovação centrada no usuário, que busca criar soluções que atendam às necessidades dos usuários e agreguem valor ao negócio. Ele é considerado uma metodologia para a concepção de produtos, serviços e experiências, e é frequentemente empregado em diversas áreas, desde design de produtos até estratégias de negócios [84].

O Design Thinking é baseado na ideia de que soluções mais eficazes e inovadoras surgem quando se entende profundamente os usuários e se aborda os problemas de forma criativa e colaborativa. Segundo Tim Brown, CEO da IDEO e um dos principais defensores do Design Thinking, essa abordagem envolve empatia, colaboração e experimentação para resolver problemas complexos [85].

O Design Thinking é composto por três fases principais — Imersão, Ideação e Prototipagem — que orientam o desenvolvimento de soluções criativas e centradas no usuário.

Na Imersão, busca-se compreender profundamente o problema e as necessidades dos usuários. Essa fase envolve uma etapa preliminar, em que se define o foco, limitações e stakeholders, e uma etapa aprofundada, baseada em pesquisas, entrevistas e observação do contexto.

A fase de Ideação concentra-se na geração de ideias e possibilidades para resolver o problema identificado. Técnicas como brainstorming e ferramentas criativas são utilizadas para explorar alternativas, que são continuamente discutidas e refinadas com o cliente.

Por fim, a Prototipagem permite transformar ideias em representações tangíveis para testar, validar e melhorar as soluções. Essa etapa pode ocorrer em paralelo às anteriores, facilitando ciclos rápidos de aprendizado e ajustes.

Por sua abordagem centrada no usuário e foco em inovação, o Design Thinking tornou-se amplamente utilizado em diferentes setores para criar soluções mais eficazes e relevantes [85].

3.7 Ferramentas para gestão de requisitos

Existem várias ferramentas disponíveis no mercado para auxiliar na gestão de requisitos de software. Essas ferramentas são projetadas para ajudar as equipes de desenvol-

vimento a capturar, rastrear, gerenciar e priorizar os requisitos do projeto de maneira eficiente. Abaixo estão algumas das ferramentas mais populares atualmente:

Jira - É uma das ferramentas de gerenciamento de projetos mais populares, amplamente utilizada por equipes ágeis de desenvolvimento de software. Ele oferece recursos abrangentes para gerenciar requisitos, incluindo a criação e rastreamento de user stories, epics e tarefas. Recursos Principais: Rastreamento de problemas, quadros Kanban e Scrum, relatórios personalizáveis, integração com outras ferramentas de desenvolvimento.

Confluence - Também desenvolvido pela Atlassian, o Confluence é uma ferramenta de colaboração e documentação que é frequentemente usada em conjunto com o Jira. Ele permite criar documentação detalhada de requisitos, especificações e outros artefatos de projeto. Recursos Principais: Edição colaborativa em tempo real, organização de conteúdo em páginas e espaços, integração com o Jira.

Trello - É uma ferramenta de gerenciamento de projetos baseada em quadros Kanban, que é adequada para equipes de desenvolvimento menores ou iniciantes. Ele permite criar listas de tarefas, cartões e atribuir responsáveis. Recursos Principais: Quadros Kanban personalizáveis, integração com outras ferramentas populares, como o Slack e o Google Drive.

Azure DevOps - Anteriormente conhecido como Visual Studio Team Services, o Azure DevOps é uma plataforma de colaboração e gerenciamento de projetos da Microsoft. Ele oferece recursos abrangentes para planejamento, desenvolvimento, teste e entrega de software. Recursos Principais: Rastreamento de problemas, planejamento de sprints, pipelines de integração contínua e entrega contínua, integração com o Visual Studio e o GitHub.

IBM Engineering Requirements Management DOORS - Esta ferramenta é amplamente utilizada por grandes empresas para gerenciar requisitos complexos em projetos de engenharia de sistemas. Ela oferece recursos avançados para captura, análise e rastreamento de requisitos. Recursos Principais: Gestão de requisitos hierárquicos, rastreamento de alterações, análise de impacto, integração com outras ferramentas de desenvolvimento. Esses recursos avançados oferecidos pela IBM Engineering Requirements Management DOORS são fundamentais para ajudar as empresas a lidar com requisitos complexos em projetos de engenharia de sistemas, proporcionando uma plataforma robusta e abrangente para captura, análise e rastreamento de requisitos.

ReqSuite® RM - É uma solução de gerenciamento de requisitos que se destaca por sua abordagem intuitiva e fácil de usar. Ele oferece recursos abrangentes para captura, análise, rastreamento e documentação de requisitos. Recursos Principais: Interface intuitiva, personalização flexível, análise de rastreabilidade, colaboração em equipe. Essas são apenas algumas das muitas ferramentas disponíveis para a gestão de requisitos de software.

A escolha da ferramenta certa dependerá das necessidades específicas da equipe, do tamanho do projeto, do orçamento disponível e de outros fatores contextuais. É importante realizar uma avaliação cuidadosa das opções disponíveis antes de tomar uma decisão.

No que refere-se a Automação e Integração Contínua:

- Jenkins e CircleCI: Estas são ferramentas dedicadas à integração contínua. Elas auxiliam as equipes no processo automático de construção, teste e implantação de software, garantindo que qualquer mudança no código seja adequadamente testada e integrada.
- Docker: Esta plataforma moderna permite que os desenvolvedores "containerizem" suas aplicações. Ao usar contêineres, as equipes podem garantir que o software seja executado de maneira consistente em diferentes ambientes, facilitando a implantação e a escalabilidade [86].

No que refere-se o controle de versão de software, tem-se:

Git - É um sistema de controle de versão distribuído que permite que as equipes rastreiem e colaborem nas mudanças do código. GitHub é uma plataforma baseada em nuvem que utiliza Git para versionamento, fornecendo uma interface amigável e recursos adicionais para colaboração em projetos de software.

Este estudo tem como objetivo não apenas realizar uma análise aprofundada das problemáticas identificadas, mas também oferecer soluções baseadas em práticas comprovadas de gestão de riscos aplicadas à elicitação de requisitos. Adicionalmente, visa propor estratégias específicas para o desenvolvimento de software que visam aprimorar a qualidade do produto final. A implementação de processos robustos, adoção de metodologias apropriadas e a utilização de ferramentas adequadas são elementos cruciais para assegurar uma elicitação de requisitos eficiente e eficaz que assegure a qualidade da entrega e a satisfação do cliente.

Capítulo 4

Método de Pesquisa

A metodologia de pesquisa desempenha um papel crucial em qualquer estudo acadêmico, fornecendo o roteiro e as diretrizes necessárias para a condução de uma investigação rigorosa e sistemática. Ao longo deste capítulo, são apresentados os procedimentos adotados para a coleta, análise e interpretação das informações, bem como as estratégias utilizadas para garantir a validade e a confiabilidade do estudo. Além disso, são discutidas as escolhas metodológicas fundamentais, incluindo a abordagem de pesquisa adotada, os métodos de coleta das informações selecionadas e as técnicas de análise empregadas.

4.1 Método de Pesquisa

A metodologia representa um conjunto de procedimentos sistemáticos, fundamentados em raciocínio lógico, com o propósito de encontrar soluções para os problemas propostos, empregando métodos científicos [87].

Por outro lado, o termo "método" denota o próprio caminho em direção a um objetivo, enquanto "metodologia" representa o estudo detalhado desse método. Em outras palavras, a metodologia é o conjunto de regras e procedimentos meticulosamente estabelecidos para conduzir uma pesquisa, abrangendo um corpo de conhecimento preciso e metodicamente organizado relacionado a um domínio específico do saber [87].

Esta abordagem possibilita uma compreensão clara e estruturada do método de pesquisa adotado, o que favorece a comunicação e a replicação do processo por outros pesquisadores. O método de pesquisa corresponde à sequência de etapas necessárias para demonstrar de que maneira os objetivos do estudo são alcançados. Os tópicos a seguir apresentam as características desta pesquisa e o modo de obtenção dos resultados. [88].

A estruturação da pesquisa deve manter alinhamento com sua modalidade, conforme o objetivo proposto. Dessa forma, o método de pesquisa adotado neste trabalho foi definido de acordo com a figura 4.1.

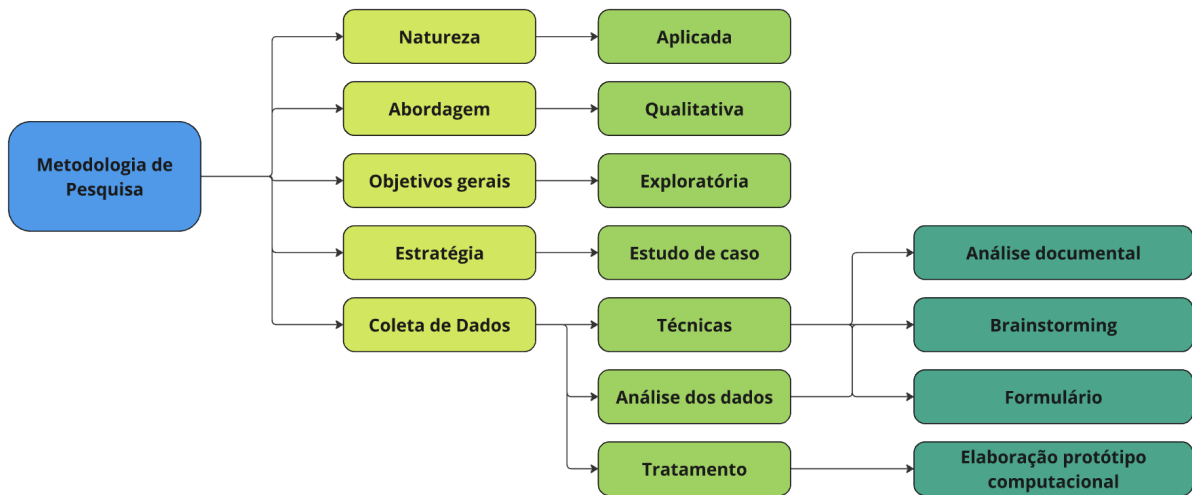


Figura 4.1: Método de pesquisa
 Fonte: Elaborada pela autora (2025)

O tipo de pesquisa adotado visa à compreensão do fenômeno estudado por meio de uma análise aprofundada e interpretativa dos dados, o que possibilita uma apreensão rica e detalhada dos aspectos subjetivos e contextuais envolvidos. A pesquisa qualitativa caracteriza-se pela ênfase na compreensão holística e na interpretação dos significados, valores e crenças dos participantes, com o objetivo de captar a complexidade e a profundidade das experiências humanas. Nesse contexto, a pesquisa qualitativa destaca-se pela flexibilidade metodológica e pela capacidade de explorar as múltiplas dimensões do fenômeno investigado, o que contribui para o aprofundamento da compreensão e para a teorização do tema em estudo [35].

Dessa forma, a adoção de uma abordagem qualitativa justifica-se pela complexidade e pela natureza subjetiva dos riscos envolvidos na elicitação de requisitos de *software*. Esse tipo de abordagem possibilita uma compreensão aprofundada dos fatores contextuais, sociais e organizacionais que influenciam a ocorrência desses riscos, bem como das percepções, experiências e interações dos stakeholders envolvidos no processo.

Quanto aos objetivos gerais, a pesquisa caracteriza-se como exploratória, uma vez que se baseia na análise documental detalhada de projetos anteriores, relatórios de falhas, registros de lições aprendidas e outras fontes relevantes de informação. Essa análise contribui para a identificação de padrões, recorrências e tendências associadas aos principais riscos enfrentados na elicitação de requisitos, além de subsidiar sua análise, avaliação e tratamento no contexto organizacional estudado. A natureza exploratória da pesquisa

visa, portanto, proporcionar maior familiaridade com o problema investigado, tornando-o mais explícito e favorecendo a construção de hipóteses e proposições analíticas [89].

A estratégia de pesquisa adotada é o "estudo de caso", uma abordagem que visa conhecer em profundidade o como e o porquê de uma determinada situação que se supõe ser única em muitos aspectos, procurando descobrir o que há nela de mais essencial e característico [39]. Além disso, permite analisar criticamente a aplicação de um protótipo computacional como suporte à gestão de riscos na elicitação de requisitos. Essa escolha metodológica nos permitirá investigar detalhadamente o contexto específico em que o protótipo será implementado, testando sua viabilidade e eficácia.

Um estudo de caso pode ser caracterizado como um estudo de uma entidade bem definida como um programa, uma instituição, um sistema educativo, uma pessoa, ou uma unidade social [39].

Esse tipo de pesquisa objetiva conhecer a variável a ser estudada em sua literalidade, do mesmo modo como se apresenta, acrescido do seu significado e o contexto no qual ela é inserida. Para realizar coleta dos dados as técnicas utilizadas são:

- Identificação do contexto: inicialmente, o estudo em questão será analisado o contexto da empresa a partir da sua cadeia de valor para entendimento de processos e negócio empresarial. A ferramenta cadeia de valor destaca as operações que agregam valor ao cliente, promovendo uma conexão mais forte entre a empresa e seu público.
- Análise SWOT: Como a ferramenta cadeia de valor não tem por finalidade proporcionar uma análise mais detalhada dos problemas, ameaças, oportunidades e pontos fortes da empresa. Será proposta a empresa uma reformulação da sua análise SWOT (Forças, fraquezas, oportunidades, Ameaças) por se tratar de uma ferramenta de gestão estratégica amplamente aceita que visa identificar e avaliar fatores internos e externos que afetam a organização.
- Análise Documental: Após a reestruturação da análise SWOT da empresa, este estudo prossegue com uma análise minuciosa de documentos, que abrange projetos anteriores, relatórios de falhas e registros de lições aprendidas. Essa abordagem permite identificar padrões e tendências relacionados aos principais desafios encontrados ao longo do processo de elicitação de requisitos de software.

Após a análise documental, estabelece-se a distinção entre problemas e riscos. Os problemas correspondem a questões atuais que demandam solução, enquanto os riscos referem-se a eventos futuros capazes de impactar negativamente o processo de elicitação de requisitos, caso não sejam adequadamente gerenciados. Com base nos dados coletados,

elabora-se o mapeamento do processo atual da fase de elicitação de requisitos da empresa (AS-IS).

- Modelagem do processo de Elicitação de requisitos atual (AS-IS)

Após a elaboração da modelagem do processo atual da empresa na fase de elicitação de requisitos, elabora-se a matriz de riscos, a qual considera a probabilidade de ocorrência e o impacto de cada risco identificado. A matriz de riscos possibilita uma visualização clara e a categorização dos riscos, com destaque para aqueles que apresentam maior potencial de impacto negativo sobre o processo de elicitação de requisitos.

Na sequência, desenvolve-se um formulário detalhado, destinado à coleta de percepções e informações sobre os riscos identificados no processo de elicitação de requisitos, com o objetivo de elencar os principais riscos a serem tratados neste estudo. O formulário é estruturado de forma a explorar de maneira sistemática os riscos levantados, o que permite à organização reconhecer e priorizar os riscos críticos a serem abordados.

Com base nos resultados do formulário, que indicam os riscos prioritários que demandam atenção imediata, elabora-se um plano de ação composto por propostas de estratégias de tratamento dos riscos, as quais podem incluir mitigação, transferência, aceitação ou evitação.

Por fim, desenvolve-se um protótipo computacional que integra as informações coletadas, a matriz de riscos e as estratégias de tratamento em um sistema de suporte à decisão, voltado à mitigação de riscos na fase de elicitação de requisitos de software.

A Figura 4.2 apresenta de forma resumida o estudo estruturado, contendo o objetivo geral, os objetivos específicos, as etapas de pesquisa e técnicas e procedimentos.

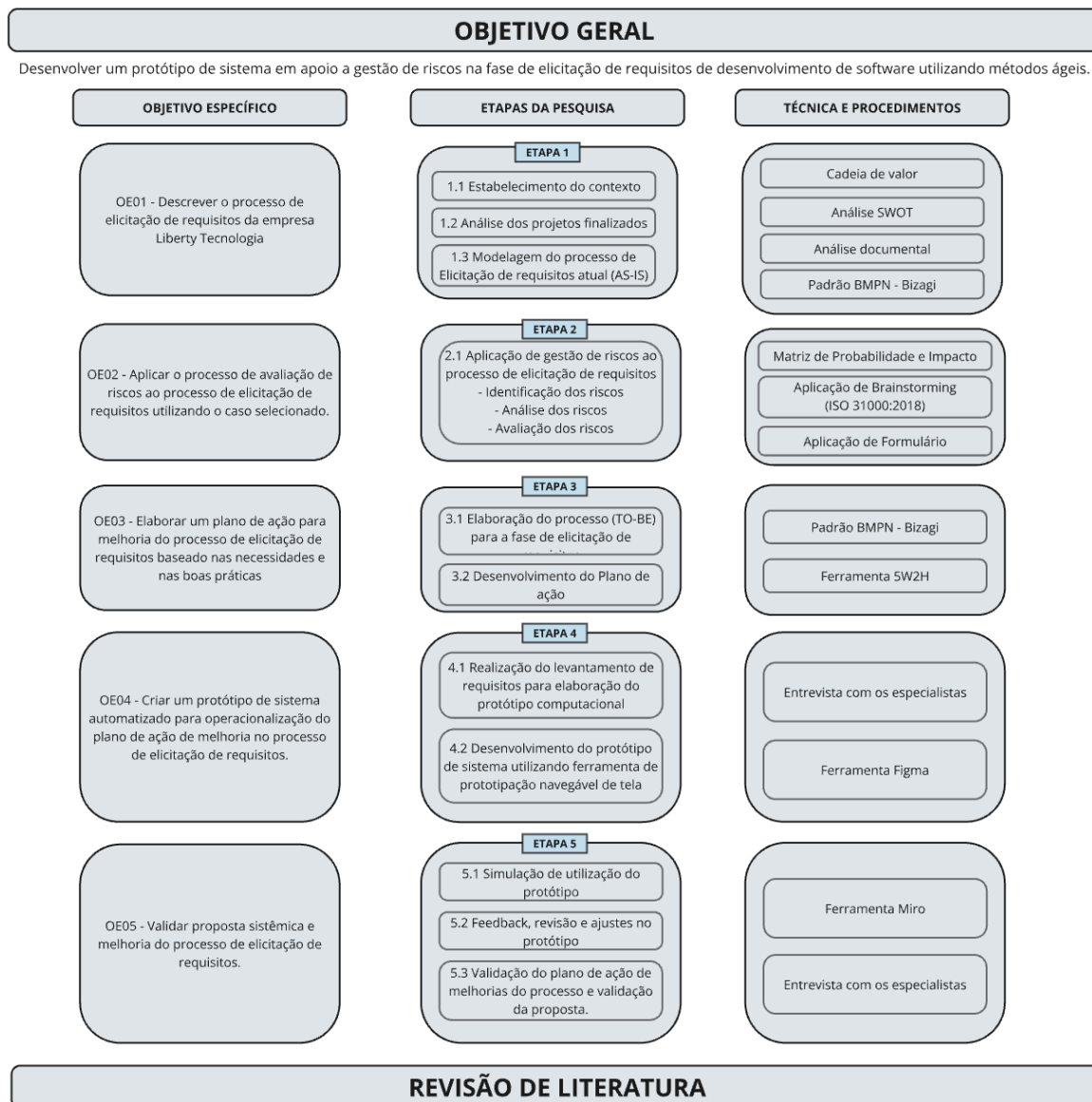


Figura 4.2: Estrutura da pesquisa
 Fonte: Elaborada pela autora (2025)

Este trabalho está organizado em cinco etapas de pesquisa distintas:

- Primeira Etapa - Identificação do Contexto

Esta fase inicial divide-se em três partes. A primeira parte tem como propósito o estabelecimento do contexto organizacional, o qual se fundamenta na análise SWOT, com a identificação dos pontos fortes, pontos fracos, ameaças e oportunidades da empresa.

A segunda parte consiste na análise minuciosa de documentos, tais como projetos anteriores, relatórios de falhas e registros de lições aprendidas. Essa análise visa à identi-

ificação de padrões e tendências relacionados aos principais riscos enfrentados na elicitação de requisitos de software, além da análise de documentos oriundos de pesquisas bibliográficas, com o objetivo de selecionar métodos, ferramentas e técnicas adequadas para a identificação de riscos no processo de desenvolvimento de software, especificamente na fase de elicitação de requisitos.

A terceira parte contempla o mapeamento do processo atual (AS-IS), no qual se utiliza a notação BPMN (Business Process Model and Notation) para representar de forma detalhada o processo de desenvolvimento de software durante a fase de elicitação de requisitos, o que proporciona uma visão clara das operações e das potenciais áreas de risco.

- Segunda Etapa - Aplicação do processo de Gestão de Riscos

Nesta etapa trataremos do processo de gestão de riscos, onde a primeira fase trata-se da:

- Identificação dos Riscos: Esta etapa tem início com a análise minuciosa de documentos de projetos anteriores e de fontes bibliográficas relevantes, o que proporciona subsídios para a identificação de potenciais riscos associados à fase de elicitação de requisitos. Ademais, elabora-se o mapeamento do processo AS-IS da fase de elicitação de requisitos, com o objetivo de identificar áreas de vulnerabilidade e oportunidades de melhoria. A partir dessas informações, constrói-se a matriz de probabilidade e impacto, destinada à identificação dos riscos e de suas respectivas ocorrências.
- Criação da Matriz de Probabilidade e Impacto: Com base nos riscos identificados, elabora-se a matriz de probabilidade e impacto. Essa matriz possibilita uma compreensão mais precisa da gravidade e da probabilidade de ocorrência de cada risco, o que favorece a priorização e o planejamento das ações de mitigação.
- Análise dos Riscos: Nesta etapa, realiza-se uma análise detalhada dos riscos previamente identificados. Para a coleta de informações sobre os principais riscos percebidos pelos stakeholders envolvidos no processo de elicitação de requisitos, utiliza-se um formulário estruturado com escalas do tipo Likert. Esse instrumento contempla diferentes áreas, tais como comunicação, compreensão dos requisitos, mudanças de escopo e conflitos de interesse, fundamentando-se nos riscos previamente identificados e registrados na matriz de probabilidade e impacto.
- Avaliação dos Riscos: Nesta fase, com base nos resultados do formulário e da matriz de probabilidade e impacto, os riscos são classificados em diferentes categorias, tais

como baixo, médio ou alto, conforme sua posição na matriz. Essa classificação possibilita a priorização dos riscos de acordo com sua importância e urgência, orientando a definição de estratégias de tratamento adequadas.

Portanto, a avaliação dos riscos proporciona uma base sólida para a tomada de decisões informadas e a implementação de medidas de controle e mitigação eficazes, visando reduzir a probabilidade de ocorrência e minimizar o impacto dos riscos identificados na fase de elicitación de requisitos de software.

O tratamento dos riscos serão tratados na próxima etapa.

- Terceira Etapa - Elaboração do processo 'TO BE' para a fase elicitación de requisitos

Nesta etapa, inicia-se o tratamento dos riscos a partir da elaboração do processo "TO BE" de elicitación de requisitos, incorporando as melhorias necessárias a fim de assegurar que o novo processo aborde de forma específica os riscos e desafios previamente identificados. O objetivo consiste no desenvolvimento de um processo de elicitación de requisitos mais robusto e eficiente, capaz de minimizar riscos e aprimorar a qualidade geral do desenvolvimento de software. Adicionalmente, realiza-se a construção de um Plano de Ação estruturado para a implementação do processo "TO BE". Esse plano contempla etapas definidas, atribuição de responsabilidades, prazos e os recursos necessários à sua execução. Essa abordagem contribui para o aumento da eficiência e da eficácia do processo de elicitación de requisitos, bem como para a redução dos riscos associados ao desenvolvimento de software, assegurando que os requisitos levantados sejam mais precisos, completos e alinhados às necessidades do projeto.

- Quarta Etapa - Realização do levantamento de requisitos para elaboração do protótipo computacional

Nesta etapa o objetivo é realizar o tratamento dos riscos a partir da criação de protótipo navegável de uma ferramenta computacional. Essa etapa será realizada em 2 fases:

- Realizar o levantamento de requisitos para elaboração do protótipo computacional

Nesta fase, realiza-se o levantamento dos requisitos, os quais são frequentemente expressos na forma de histórias de usuário, caracterizadas como descrições breves de funcionalidades ou características do sistema, elaboradas sob a perspectiva do usuário final. O objetivo consiste em capturar as necessidades e expectativas dos usuários em relação ao software a ser desenvolvido. A ferramenta selecionada é projetada especificamente para facilitar a gestão de riscos no processo de elicitación de requisitos. Para o gerenciamento desses requisitos, utiliza-se a ferramenta Jira.

- Desenvolver o protótipo de sistema utilizando ferramenta de prototipação navegável de tela

Nesta fase, realiza-se a elaboração do design do protótipo, utilizando-se a ferramenta Figma, reconhecida por sua eficácia na prototipagem de interfaces e na experiência do usuário (UI/UX). A proposta desse protótipo consiste na introdução de uma solução inovadora e prática para o desafio de gerenciar riscos de forma eficiente durante as fases iniciais do desenvolvimento de software.

- Quinta Etapa - Simulação de utilização do protótipo

Nesta etapa o objetivo é realizar a implantação e elencar melhorias no plano de ação. Essa etapa será realizada em 3 fases:

- Simulação de utilização do protótipo: Nesta etapa o protótipo navegável será implementado e testado em um ambiente real. Esse teste prático visa validar a funcionalidade, a usabilidade e a eficiência da ferramenta na gestão de riscos durante a elicitação de requisitos.
- Nesta etapa, realizam-se a coleta de feedback, a revisão e os ajustes no protótipo. Esse processo possibilita a obtenção de contribuições relevantes, fundamentais para o aprimoramento contínuo do protótipo, assegurando que o artefato atenda às necessidades reais dos usuários e contribua para a melhoria dos processos de elicitação de requisitos.
- Nesta etapa, realiza-se a validação do plano de ação de melhorias do processo, bem como a validação da proposta desenvolvida. Essa etapa mostra-se fundamental para a transformação de teorias e conceitos em uma solução aplicável e tangível, evidenciando o potencial das inovações tecnológicas na melhoria de processos críticos na fase de elicitação de requisitos no desenvolvimento de software.

Cada etapa deste estudo contribui para o desenvolvimento de um instrumento prático e eficaz voltado à gestão de riscos na fase de elicitação de requisitos. Essa abordagem estruturada possibilita não apenas a identificação, mas também a mitigação dos riscos, resultando em processos mais eficientes e com maior probabilidade de sucesso.

4.1.1 Local do estudo

A empresa objeto deste estudo caracteriza-se como uma sociedade anônima brasileira de capital fechado, fundada em 2010. Sua atuação no mercado de Tecnologia da Informação é marcada pela especialização no desenvolvimento de soluções sob medida. Com a

missão de consolidar-se como um grupo empresarial de referência no setor de TI, a organização orienta suas atividades para o sucesso de clientes, colaboradores e acionistas. Tal objetivo é perseguido por meio de uma estratégia inovadora, que prioriza a capacitação de seus recursos humanos, a adoção de ferramentas avançadas de gestão e a manutenção de uma postura ágil voltada ao alcance de resultados e à geração de valor para o negócio. Ademais, a empresa compromete-se com a oferta de serviços de elevada qualidade, buscando continuamente a excelência no relacionamento com clientes e parceiros.

4.1.2 Objeto de estudo

Este estudo concentrou-se na compreensão e gestão dos riscos associados à fase de elicitação de requisitos na empresa Liberty Tecnologia. O objetivo primordial foi identificar, analisar e propor estratégias para lidar com esses riscos específicos, que desempenham um papel crucial no desenvolvimento de software.

Ao direcionar nossa atenção para os desafios inerentes à etapa de elicitação de requisitos, buscamos não apenas reconhecer os potenciais obstáculos, mas também oferecer soluções práticas e eficazes para mitigá-los. Dessa forma, nosso intuito foi otimizar o processo de desenvolvimento de software na empresa, garantindo uma abordagem mais robusta e orientada para resultados.

Ao longo deste estudo, exploraremos minuciosamente os diversos cenários de risco que podem surgir durante a elicitação de requisitos, compreendendo sua natureza e impacto no projeto. Com base nessa análise profunda, formulamos estratégias de tratamento adaptadas às necessidades específicas da Liberty Tecnologia, visando promover a eficiência, a qualidade e o sucesso contínuo de seus projetos na fase de elicitação de requisitos.

4.1.3 Instrumento de coleta de dados

o instrumento de coleta de dados do meu estudo envolve diversas técnicas e abordagens para obter informações relevantes sobre os riscos na fase de elicitação de requisitos de software. Segue uma descrição detalhada do instrumento de coleta de dados:

Identificação de Contexto:

A identificação de contexto da empresa estudada foi realizada a partir da utilização da análise SWOT (*Strengths, Weaknesses, Opportunities, Threats*) que em português significa: Forças, Fraquezas, Oportunidades e Ameaças. A SWOT envolve uma análise abrangente dos diversos aspectos internos e externos que afetam o desempenho e a posição estratégica da empresa. Importante mencionar que para identificar o contexto da empresa foi essencial envolver várias partes interessadas, incluindo membros da alta administração, funcionários de diferentes departamentos e clientes-chave. Isso garantiu uma

avaliação abrangente e holística do ambiente interno e externo da empresa, fornecendo *insights* valiosos para o desenvolvimento de estratégias eficazes na tomada de decisões.

Análise Documental:

Esta técnica envolveu a revisão detalhada dos documentos dos projetos finalizados da empresa relacionados à elicitação de requisitos. Os documentos examinados incluíram especificações de requisitos anteriores, relatórios de projetos, registros de reuniões e outros artefatos relevantes. O objetivo foi identificar padrões, tendências e lições aprendidas relacionadas aos riscos enfrentados na fase de elicitação de requisitos.

Referências Bibliográficas: Além da análise documental interna, também realizamos consultas as referências bibliográficas sobre gestão de risco na fase de elicitação de requisitos. além de pesquisas de métodos ágeis e desenvolvimento de software. Essas referências nos apoiaram teoricamente ao estudo e forneceram insights sobre práticas recomendadas e abordagens eficazes para lidar com os riscos na elicitação de requisitos.

Modelagem do processo de elicitação de requisitos - AS-IS

Com base nas informações coletadas por meio da análise documental realizada e nos estudos conduzidos a partir das referências bibliográficas, tornou-se possível avaliar o processo atual da empresa no que se refere à fase de elicitação de requisitos, bem como identificar problemas e potenciais riscos. A partir dessa análise, elaborou-se a modelagem do processo de elicitação de requisitos no estado atual (AS-IS).

Reunião de Brainstorming: De posse das informações coletadas e da modelagem do processo previamente elaborada, realizou-se uma sessão de brainstorming com os profissionais indicados pela empresa, com o objetivo de identificar riscos específicos na fase de elicitação de requisitos. Durante esse encontro, os participantes compartilharam percepções, experiências e preocupações relacionadas aos riscos, o que resultou na elaboração de uma lista inicial de potenciais ameaças ao sucesso do processo de elicitação de requisitos.

Matriz de Probabilidade e Impacto:

A Matriz de Probabilidade e Impacto foi desenvolvida inicialmente em meados da década de 1950, com o objetivo de auxiliar os militares dos Estados Unidos no gerenciamento de projetos complexos. Desde então, essa matriz passou a ser amplamente adotada em diversas áreas, incluindo desenvolvimento de software, engenharia, finanças, saúde, entre outras. O método constitui uma das ferramentas mais difundidas no gerenciamento de riscos e é amplamente abordado em cursos de gerenciamento de projetos. Com base na lista de riscos identificados, elaborou-se uma matriz de probabilidade e impacto com a finalidade de analisar e classificar os riscos de acordo com sua probabilidade de ocorrência e seu impacto potencial na fase de elicitação de requisitos. Essa matriz possibilita a priorização dos riscos e orienta os esforços de gerenciamento para as áreas consideradas mais críticas.

Aplicação de Formulário

Para a avaliação dos riscos, elaborou-se um formulário a ser respondido pelos profissionais da empresa, utilizando-se uma escala do tipo Likert, a partir dos riscos previamente analisados. Na etapa de avaliação de riscos, realizou-se uma análise detalhada dos riscos identificados. Para a coleta de informações acerca dos principais riscos percebidos pelos stakeholders envolvidos no processo de elicitação de requisitos, utilizou-se um formulário com escalas de Likert, com a finalidade de avaliar a gravidade dos riscos percebidos. Essa abordagem quantitativa possibilita a mensuração das percepções dos participantes e fornece uma base objetiva para a priorização dos riscos.

Avaliação dos Riscos:

Com base nos resultados do formulário e da matriz de probabilidade e impacto, os riscos foram classificados em diferentes categorias, tais como baixo, médio ou alto, conforme sua posição na matriz. Essa classificação possibilitou a priorização dos riscos de acordo com sua importância e urgência, orientando a definição de estratégias de tratamento adequadas. Dessa forma, a avaliação dos riscos forneceu uma base consistente para a tomada de decisões informadas e para a implementação de medidas de controle e mitigação eficazes, com vistas à redução da probabilidade de ocorrência e à minimização do impacto dos riscos identificados na fase de elicitação de requisitos de software. O tratamento dos riscos foi conduzido na etapa subsequente.

Tratamento dos riscos - Modelagem do processo de elicitação de requisitos - TO-BE

Nesta etapa, elaboraram-se o mapeamento do processo TO-BE para a fase de elicitação de requisitos e o plano de ação voltado à mitigação dos riscos identificados.

A elaboração do processo TO-BE de elicitação de requisitos teve como finalidade incorporar as melhorias necessárias para assegurar que o novo processo abordasse, de forma específica, os riscos e desafios previamente identificados. O objetivo consistiu no desenvolvimento de um processo de elicitação de requisitos mais robusto e eficiente, capaz de reduzir os riscos e aprimorar a qualidade geral do desenvolvimento de software. Adicionalmente, elaborou-se um plano de ação estruturado para a implementação do processo TO-BE, contemplando etapas definidas, atribuição de responsabilidades, prazos e recursos necessários. Essa abordagem contribuiu para o aprimoramento da eficiência e da eficácia do processo de elicitação de requisitos, bem como para a redução dos riscos associados ao desenvolvimento de software, assegurando que os requisitos capturados fossem mais precisos, completos e alinhados às necessidades do projeto.

Elaboração do Plano de ação:

Um plano de ação é uma ferramenta essencial no gerenciamento de projetos, processos e tarefas em diversas áreas, como negócios, educação, saúde e governança. Ele serve para:

1. Organizar e Estruturar: Ajuda a organizar as atividades necessárias para atingir um objetivo específico, estruturando as etapas e os recursos necessários para alcançá-lo.
2. Definir Responsabilidades: Atribui responsabilidades claras a cada membro da equipe, estabelecendo quem é responsável por cada tarefa e atividade.
3. Estabelecer Prazos: Define prazos e cronogramas para cada etapa do plano, garante que as atividades sejam concluídas dentro do tempo estipulado.
4. Monitorar Progresso: Permite acompanhar o progresso do projeto ou da tarefa, identifica eventuais desvios e possibilita ajustes para manter o plano no caminho certo.
5. Tomar Decisões: Facilita a tomada de decisões ao fornecer informações claras sobre o que foi feito, o que está em andamento e o que ainda precisa ser feito.
6. Alinhar Objetivos: Garante que todos os membros da equipe estejam alinhados com os objetivos e metas do projeto, evitando mal-entendidos e conflitos.
7. Melhorar a Comunicação: Promove uma comunicação eficaz ao fornecer um documento claro e compartilhado que descreve o que precisa ser feito e como será feito.
8. Aumentar a Eficiência: Contribui para a eficiência operacional, evitando retrabalho, minimizando desperdícios de tempo e recursos e otimizando o uso dos recursos disponíveis.

Para a elaboração do plano de ação deste estudo, utiliza-se a ferramenta 5W2H, a qual consiste em uma técnica de planejamento e gestão voltada à garantia da compreensão e da execução eficiente de projetos, tarefas ou atividades. A denominação da ferramenta deriva das sete questões fundamentais que orientam sua aplicação, conforme descrito a seguir:

What (O quê?): refere-se à definição clara e objetiva do que será realizado. Why (Por quê?): busca identificar os motivos e os objetivos associados à execução da atividade. Where (Onde?): determina o local ou o ambiente no qual a atividade será desenvolvida. When (Quando?): estabelece o cronograma e os prazos para a realização da atividade. Who (Quem?): identifica os responsáveis pela execução da atividade. How (Como?): descreve os métodos, os processos e os recursos necessários para a execução da atividade. How much (Quanto?): define os custos envolvidos na realização da atividade.

A aplicação dessa técnica contribui para o alinhamento e a clareza entre os envolvidos no projeto, além de facilitar o planejamento, a execução e o controle das atividades, conforme a figura 4.3.



Figura 4.3: Ferramenta 5W2H
 Fonte: Rockontent [90] (2025)

A origem da ferramenta 5W2H não está associada a um autor específico, sendo caracterizada como uma abordagem tradicionalmente empregada no gerenciamento de projetos e na gestão empresarial. Ainda assim, a ferramenta é amplamente reconhecida e aplicada em diferentes áreas e contextos organizacionais, em razão de sua simplicidade e eficácia na organização e na comunicação de informações.

Diversos estudos e práticas consolidadas na área de gestão destacam a importância da utilização de técnicas claras e estruturadas, como a 5W2H, para o sucesso organizacional. Essas abordagens enfatizam a relevância do planejamento detalhado, da definição de objetivos claros e da atribuição de responsabilidades como elementos essenciais para a obtenção de resultados eficientes e eficazes.

Dessa forma, elabora-se um plano de ação a partir da adoção da ferramenta 5W2H, com o propósito de assegurar que as atividades sejam executadas de maneira eficiente, eficaz e organizada, contribuindo para o alcance dos objetivos estabelecidos

Realizar o levantamento de requisitos para elaboração do protótipo computacional

O processo de construção de uma ferramenta computacional voltada à elicitação de requisitos na empresa Liberty iniciou-se com o levantamento das necessidades do negócio

e a avaliação de sua viabilidade. Identificou-se a ausência de uma metodologia ágil e de ferramentas adequadas, evidenciando a necessidade de uma solução que auxiliasse na identificação assertiva das necessidades dos clientes. A ferramenta proposta caracteriza-se também por sua natureza intuitiva, permitindo uma visão compartilhada em tempo real entre os participantes do processo.

Para a compreensão das necessidades da área de tecnologia, realizaram-se entrevistas iniciais, considerando-se apenas informações pertinentes ao estudo, com a exclusão de dados de caráter sigiloso da empresa.

Desenvolvimento do protótipo de sistema

A elaboração do protótipo computacional realiza-se a partir dos riscos prioritários, do plano de ação e do mapeamento do processo TO-BE proposto à empresa. Durante a fase de elicitação de requisitos, procede-se à identificação das personas, das necessidades e dos demais requisitos de negócio da área de desenvolvimento de sistemas, com o objetivo de tornar o processo de elicitação mais assertivo.

Na sequência, elaboram-se as histórias de usuário, as quais representam os caminhos que os usuários percorrem para alcançar os resultados de negócio esperados. Busca-se, com isso, capturar as necessidades e expectativas dos usuários em relação ao software a ser desenvolvido, de modo a facilitar a gestão de riscos no processo de elicitação de requisitos. Para o gerenciamento desses requisitos, utiliza-se o software Jira.

Posteriormente, desenvolve-se o protótipo do sistema por meio de uma ferramenta de prototipagem de telas navegáveis. Nessa etapa, o design do protótipo é elaborado utilizando a ferramenta Figma. A proposta do protótipo consiste em apresentar uma solução inovadora e prática para o desafio de gerenciar riscos de forma eficiente durante a fase de elicitação de requisitos.

Por fim, na fase de simulação de utilização do protótipo, realiza-se a implementação e o teste do protótipo navegável em um ambiente real, com a finalidade de validar sua funcionalidade, usabilidade e eficácia na gestão de riscos durante a elicitação de requisitos.

Em seguida, procede-se à coleta de feedback para a revisão e o ajuste do protótipo, com base nas contribuições recebidas. Esse processo permite o refinamento do artefato, assegurando que atenda às necessidades reais dos usuários e contribua para a melhoria dos processos de elicitação de requisitos. Dessa forma, efetua-se a validação do plano de ação voltado às melhorias do processo, bem como da proposta desenvolvida. Essa etapa mostra-se fundamental para a transformação de conceitos teóricos em uma solução tangível e aplicável, evidenciando o potencial das inovações tecnológicas na melhoria de processos críticos na fase de elicitação de requisitos no desenvolvimento de software.

Capítulo 5

Resultados e Análises

5.1 Estabelecimento do contexto

Neste capítulo, será abordado o estabelecimento do contexto organizacional e a estrutura da fase de elicitação de requisitos de software ao qual esta pesquisa está relacionada.

5.1.1 A organização

A Liberty Tecnologia da Informação foco desse estudo é uma sociedade anônima de capital fechado, brasileira, que teve sua fundação em 2010. Ela se destaca no mercado de Tecnologia da Informação por sua abordagem exclusiva no desenvolvimento de soluções personalizadas. A missão da organização é clara e ambiciosa: tornar-se um grupo empresarial de destaque no setor de TI, proporcionando sucesso a clientes, colaboradores e acionistas.

No que tange a sua cadeia de valor, a empresa apresenta-se em uma estrutura, conforme Figura 5.1.

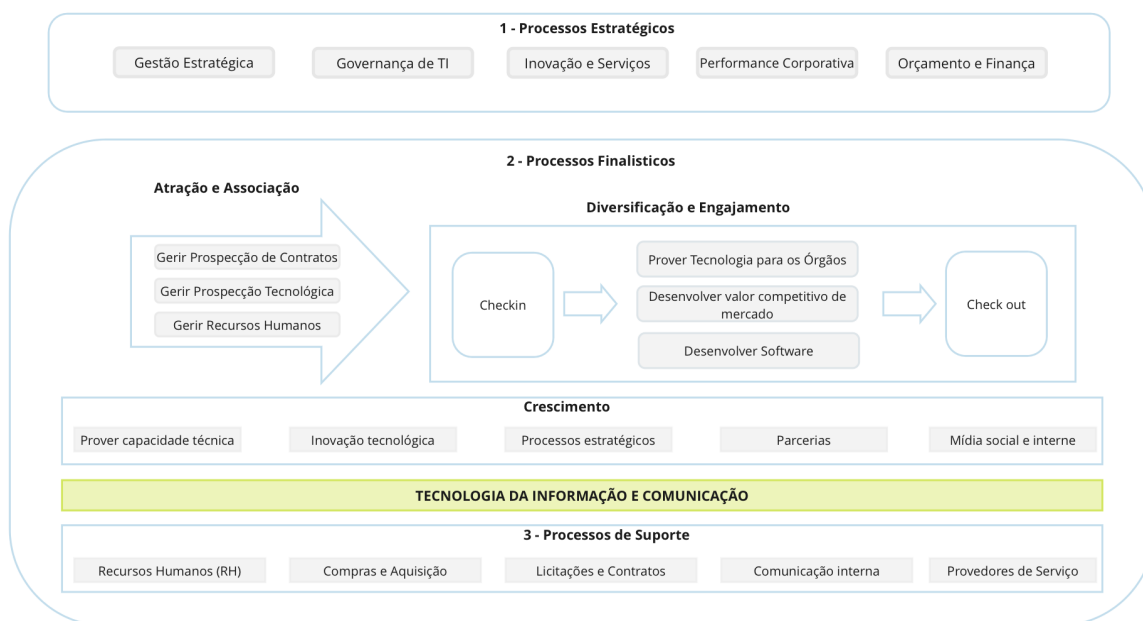


Figura 5.1: Cadeia de Valor

Fonte: Elaborada pela autora (2025)

A cadeia de valor da empresa está dividida em três categorias principais:

Processos Estratégicos: Estes processos são fundamentais para a definição e alcance dos objetivos estratégicos da empresa. Eles englobam atividades relacionadas ao planejamento, à visão de longo prazo e às estratégias que impulsionam o sucesso organizacional.

Processos Finalísticos: Esses processos estão no cerne das operações da empresa e estão diretamente relacionados à entrega de soluções personalizadas para os clientes. Eles envolvem atividades de desenvolvimento, implementação e entrega de produtos ou serviços que atendem às necessidades dos clientes.

Processos de Suporte: Os processos de suporte são essenciais para garantir a eficiência e a eficácia das operações da empresa, incluem atividades de suporte administrativo, gerenciamento de recursos e infraestrutura, que sustentam os processos estratégicos e finalísticos.

Contudo, a estrutura utilizada para a cadeia de valor não proporciona uma análise mais detalhada dos problemas, ameaças, oportunidades e pontos fortes da empresa. Consequentemente, dados os desafios que a Liberty Tecnologia da informação enfrenta, é imperativo que a empresa realize uma análise SWOT abrangente. A análise SWOT (Forças, fraquezas, oportunidades, Ameaças) é uma ferramenta de gestão estratégica amplamente aceita. É usado para identificar e avaliar fatores internos e externos que afetam a organização. Ao

mesmo tempo Executamos uma análise SWOT da empresa, conforme apresentado na Figura 5.2.

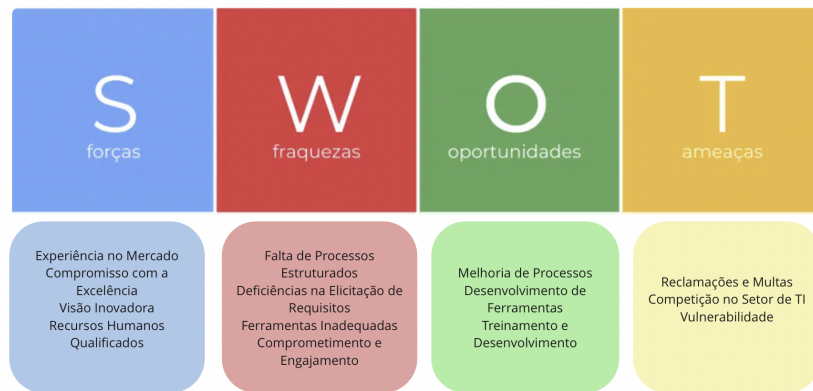


Figura 5.2: Análise SWOT

Fonte: Elaborada pela autora (2025)

Após a realização da análise SWOT (Strengths, Weaknesses, Opportunities, Threats) da empresa foram apresentados os seguintes cenários:

Forças (Strengths):

- Experiência no Mercado: A empresa possui experiência no mercado de Tecnologia da Informação.
- Compromisso com a Excelência: A empresa demonstra um compromisso firme em fornecer serviços de excelência aos seus clientes e parceiros.
- Visão Inovadora: A visão inovadora da organização pode ser um trunfo importante para a busca contínua de agilidade e sucesso.
- Recursos Humanos Qualificados: A capacitação de recursos humanos altamente qualificados é um ativo valioso para a empresa.

Fraquezas (Weaknesses):

- Falta de Processos Estruturados: A ausência de processos bem definidos na área de desenvolvimento é uma fraqueza significativa.
- Deficiências na fase de Elicitação de Requisitos: A falta de uma metodologia ágil bem definida e processos robustos para gerenciar riscos durante a elicitação de requisitos compromete a qualidade das entregas.

- Ferramentas Inadequadas: A inexistência de ferramentas apropriadas para o controle de demandas e a gestão de projetos é uma fraqueza que impacta a eficiência operacional.
- Comprometimento e Engajamento: O declínio no comprometimento e engajamento dos profissionais e stakeholders representa outra fraqueza que afeta a produtividade.

Oportunidades (Opportunities):

- Melhoria de Processos: A empresa tem a oportunidade de implementar processos estruturados, incluindo uma metodologia ágil sólida, para melhorar a qualidade das entregas.
- Desenvolvimento de Ferramentas: Pode explorar oportunidades para desenvolver ou adquirir ferramentas eficientes para a gestão de demandas.
- Treinamento e Desenvolvimento: Investir em treinamento e desenvolvimento de profissionais pode melhorar o engajamento da equipe.

Ameaças (Threats):

- Reclamações e Multas: As reclamações e multas por parte das empresas contratantes representam uma ameaça à reputação e às finanças da empresa.
- Competição no Setor de TI: A competição no setor de Tecnologia da Informação é alta, o que torna essencial a melhoria da qualidade e eficiência das entregas.
- Vulnerabilidade: A falta de processos bem definidos e ferramentas adequadas expõe a empresa a riscos operacionais e financeiros.

Para superar as fraquezas identificadas e aproveitar as oportunidades existentes, a empresa Libert Tecnologia necessita concentrar esforços na implementação de processos estruturados, na definição de uma metodologia ágil consistente e na adoção de ferramentas eficazes para a gestão de demandas. O investimento no treinamento e no desenvolvimento da equipe mostra-se igualmente relevante, uma vez que contribui para o aumento do comprometimento e do engajamento dos profissionais. Ademais, a manutenção de elevados padrões de qualidade nas entregas revela-se fundamental para enfrentar a concorrência no mercado de Tecnologia da Informação, bem como para mitigar riscos relacionados a reclamações e eventuais penalidades.

5.1.2 Análise dos projetos finalizados

Para a realização da análise documental deste estudo, selecionaram-se cinco projetos críticos da empresa, a saber: Projeto UGH, Projeto IMH – Implantação Médico-Hospitalar, Projeto Medicamentos, Projeto Contrato e Licitação e Projeto Pagamento de Fornecedores. Todos os projetos analisados encontravam-se finalizados e apresentavam escopo, prazo e custo previamente definidos. A escolha desses projetos fundamentou-se em uma análise conduzida no ambiente organizacional, a qual identificou elevado nível de criticidade na fase de elicitação de requisitos e no desenvolvimento dos projetos examinados. Verificou-se que todos enfrentaram desafios significativos, resultando em elevado índice de retrabalho para a empresa. As informações obtidas foram utilizadas como referência para a elaboração de uma lista de problemas e riscos potenciais, bem como serviram de insumo para a modelagem do processo da fase de elicitação de requisitos. Para a condução do estudo, contou-se com a colaboração de dez profissionais da área de Tecnologia da Informação disponibilizados pela empresa, com formações acadêmicas em Ciência da Computação (seis profissionais), Sistemas de Informação (dois profissionais) e Análise e Desenvolvimento de Sistemas (três profissionais). A maioria dos participantes ocupa o cargo de analista de requisitos, sendo quatro deles gerentes de projetos. As experiências profissionais variam entre três e oito anos, sendo que parte dos participantes possui até dois anos de atuação na empresa. Nesse contexto, tornou-se possível a modelagem do cenário atual por meio da notação BPMN, com o apoio desses especialistas.

Além da análise interna da documentação, utilizaram-se referências bibliográficas relacionadas à gestão de riscos na etapa de elicitação de requisitos, bem como estudos sobre métodos ágeis e desenvolvimento de software. Essas fontes forneceram suporte teórico à pesquisa e contribuíram com reflexões sobre práticas recomendadas e abordagens eficazes para o tratamento de riscos durante a elicitação de requisitos.

Como resultado, o estudo possibilitou o desenho do processo atual (AS-IS) da fase de elicitação de requisitos, abrangendo desde o surgimento da necessidade até a etapa de início do processo de desenvolvimento do software.

5.1.3 Modelagem do processo de Elicitação de requisitos atual (AS-IS)

Com base nos estudos realizados nas documentações dos sistemas supracitados, foi adotada a notação de modelagem de processos BPMN (*Business Process Model and Notation*), Em conjunto com o software Bizagi, ferramenta escolhida pela empresa em razão de sua robustez e de suas possibilidades de integração com diversas plataformas de gestão

empresarial, viabilizou-se o desenho do processo AS-IS. Esse processo foi modelado de acordo com as informações coletadas, conforme apresentado na figura 5.3.

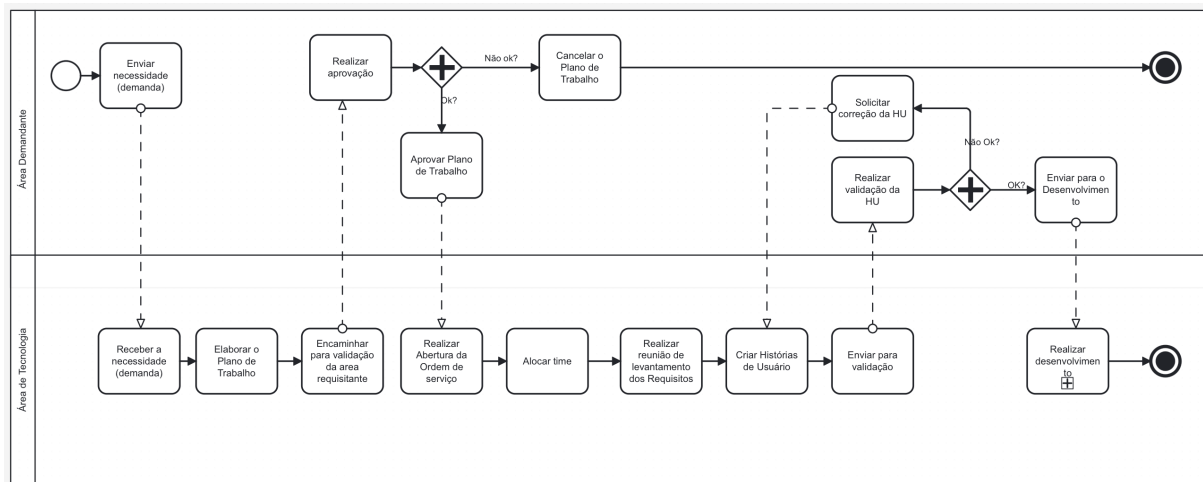


Figura 5.3: Problemas

Fonte: Elaborada pela autora (2025)

Para a identificação do processo atual (AS-IS), utilizou-se o resultado da análise documental e bibliográfica, da qual se obteve como produto a lista de problemas identificados na fase de elicitação de requisitos.

Considera-se problema todo evento ou condição que já ocorreu e que impacta negativamente o projeto ou a operação. Diferentemente do risco, o problema representa uma certeza e, portanto, demanda uma solução imediata. Os problemas apresentam as seguintes características:

- Baseado em eventos que já aconteceram.
- Precisa ser resolvido para evitar impactos negativos adicionais.
- Geralmente requer ações corretivas ou de remediação.
- Identificado durante a execução do projeto ou operação [19]

Com isso, foi possível identificar os problemas relacionados à fase de elicitação de requisitos, conforme apresentado na Figura 5.4.

PRINCIPAIS PROBLEMAS
Dificuldade em alinhar expectativas entre as partes interessadas.
Mal-entendidos frequentes que podem levar a retrabalho.
Dificuldade em medir o progresso e o sucesso do projeto.
Conflitos entre equipe de desenvolvimento e stakeholders.
Requisitos mal interpretados, resultando em funcionalidades incorretas.
Aumento de custos e prazos devido a correções tardias.
Omissão de funcionalidades críticas no produto final.
Falhas na entrega das expectativas do cliente ou do mercado.
Necessidade de revisões significativas do projeto.
Gestão de mudanças constante, aumentando a carga de trabalho.
Desvio dos objetivos originais do projeto, afetando o escopo.
Risco de não cumprimento dos prazos e orçamento.
Incapacidade de adaptar-se a mudanças de forma eficiente.
Perda de rastreabilidade e controle sobre o projeto.
Reputação da empresa e confiança do cliente prejudicadas.
Alto turnover de funcionários e desmotivação da equipe.
Comprometimento do sucesso global do projeto.
Dificuldades em escalar operações ou gerenciar múltiplos projetos.
Falhas na comunicação e na colaboração entre equipes.
Ineficiência nos processos de desenvolvimento e entrega.
Dificuldades em manter a competitividade no mercado.
Danos à imagem e credibilidade da empresa.
Impacto financeiro negativo devido a glosas, multas ou penalidades.

Figura 5.4: Principais Problemas
Fonte: Elaborada pela autora (2025)

O processo AS-IS de elicitação de requisitos, representado na Figura 5.4, foi identificado a partir da análise documental e bibliográfica, permitindo o mapeamento do fluxo atual desde o envio da necessidade pela área demandante até o encaminhamento para

desenvolvimento pela área de tecnologia. O fluxo tem início com o recebimento da demanda e a elaboração do plano de trabalho, o qual é submetido à validação e à aprovação, podendo resultar no cancelamento ou na continuidade do processo. Uma vez aprovado, realizam-se a abertura da ordem de serviço, a alocação da equipe e o levantamento dos requisitos, culminando na elaboração das histórias de usuário. Essas histórias passam por ciclos de validação, nos quais podem ser solicitados ajustes até que estejam adequadas para o encaminhamento à fase de desenvolvimento. A análise desse mapeamento evidenciou a ocorrência de problemas na fase de elicitação de requisitos, caracterizados como eventos certos e com impacto negativo sobre os projetos, demandando ações corretivas imediatas. A identificação desses problemas reforça a necessidade de uma abordagem estruturada, pautada em processos bem definidos, papéis claramente estabelecidos, uso de ferramentas adequadas, adoção consistente de metodologias ágeis e aplicação de práticas de gestão de riscos, com vistas à melhoria do desempenho organizacional, à satisfação dos clientes e à sustentabilidade das operações no mercado de Tecnologia da Informação. Nesse contexto, torna-se necessária a adoção de uma abordagem abrangente e estruturada para o atendimento às demandas do mercado de TI, promovendo a inovação, a excelência na prestação de serviços e a busca contínua pela satisfação de clientes e acionistas. O investimento na capacitação de recursos humanos, aliado à utilização de ferramentas eficazes de gestão, evidencia a orientação da organização para a sustentabilidade e o sucesso organizacional no longo prazo.

5.2 Aplicação de gestão de riscos ao processo de elicitação de requisitos

Nesta seção, foi estruturada para a identificação, análise, avaliação e tratamento do risco, a partir da elaboração do mapeamento de processo da fase de elicitação de requisitos da empresa estudada.

- **Identificação de Riscos:** O objetivo desta etapa foi identificar os possíveis riscos, ameaças e controles associados ao processo atual AS-IS de elicitação de requisitos.
- **Análise de Riscos:** Nesta etapa, buscou estabelecer as probabilidades e os impactos de cada risco identificado.
- **Avaliação dos Riscos:** Para esta etapa, foram definidos os níveis de cada risco, bem como sua priorização.
- **Tratamento de Riscos:** No tratamento dos riscos, realizamos a modelagem de processo TO-BE com a mitigação dos riscos identificados no processo AS-IS.

5.2.1 Identificação dos riscos

Para a identificação dos riscos, tornou-se necessária a análise da lista de problemas e do desenho do processo AS-IS (estado atual), examinando-se as informações existentes a fim de antecipar possíveis dificuldades futuras no processo de elicitação de requisitos. Para isso, foram seguidas as seguintes etapas:

- Revisão da lista de problemas: Examinou-se a lista de problemas existentes. Estes problemas eram sintomas de riscos mais profundos do processo de elicitação de requisitos, pois a partir da análise de cada problema foi possível entender suas causas e efeitos, o que nos ajudou a prever riscos potenciais.
- O mapeamento do processo AS-IS representa o estado atual do processo de elicitação de requisitos, possibilitando a identificação de lacunas existentes. A partir da análise desse desenho, tornam-se evidentes os pontos nos quais ocorrem eventos e problemas recorrentes, bem como as etapas em que o processo se apresenta excessivamente complexo ou pouco definido. Esses pontos mostram-se mais suscetíveis à ocorrência de riscos, evidenciando a necessidade de intervenções e melhorias no processo.
- Identificação de padrões e tendências: A identificação de padrões e tendências nos problemas observados e no desenho do processo possibilitou evidenciar a existência de riscos sistêmicos ou recorrentes. A repetição de determinados tipos de problemas em etapas específicas do processo indica a presença de riscos que demandam tratamento adequado. A partir dessa análise, tornou-se possível a identificação dos riscos associados ao processo de elicitação de requisitos.

Ressalta-se que, nesta seção, são apresentados os riscos identificados no processo de elicitação de requisitos que se encontram direta ou indiretamente relacionados a esse processo. A condução deste trabalho contou com a colaboração de profissionais disponibilizados pela empresa. Para a realização do estudo, adotou-se o modelo de gestão de riscos estabelecido pela norma ISO 31000:2018, em conjunto com as ferramentas e técnicas recomendadas pela ISO 31010:2012, aplicadas ao contexto da gestão de riscos no processo de elicitação de requisitos.

De acordo com a ISO 31000, risco é definido como o “efeito da incerteza nos objetivos”, sendo o efeito compreendido como uma consequência, positiva ou negativa, resultante de ações ou eventos. Essa definição evidencia que o risco pode influenciar a capacidade da organização de alcançar seus objetivos, gerando impactos tanto favoráveis quanto adversos. A norma enfatiza a forma como a incerteza afeta os objetivos organizacionais e abrange atividades como identificação, análise, avaliação, tratamento, monitoramento e comunicação dos riscos no contexto da gestão de riscos[48].

- Estará sempre associado à incerteza.
- Tem uma probabilidade de ocorrência e um impacto associado.
- Pode ser mitigado ou gerenciado através de estratégias de antecipação.
- Identificado na fase de planejamento e monitorado ao longo do do processo ou projeto.

Com isso, tornou-se possível identificar os riscos relacionados à fase de elicitação de requisitos, conforme apresentado na figura 5.5.

RISCOS
R01 - Definições imprecisas de negócio, escopo e objetivos
R02 - Desentendimentos sobre os requisitos
R03 - Insuficiência na definição dos requisitos funcionais e não funcionais
R04 - Alterações frequentes nos requisitos
R05 - Ineficácia na gestão de mudanças dos requisitos
R06 - Entregas insatisfatórias
R07 - Declínio no comprometimento
R08 - Riscos não gerenciados
R09 - Ausência de processos estruturados
R10 - Inexistência de ferramentas adequadas
R11 - Metodologia ágil não definida
R12 - Riscos financeiros e reclamações

Figura 5.5: Identificação dos Riscos
 Fonte: Elaborada pela autora (2025)

Em síntese, o risco consiste na possibilidade de ocorrência de eventos futuros capazes de afetar a fase de elicitação de requisitos. No contexto deste estudo, consideram-se exclusivamente os riscos de natureza negativa.

- Riscos relacionados a negócios e às barreiras de comunicação: definições imprecisas de negócio, escopo e objetivos podem gerar mal-entendidos e atrasos no andamento do projeto.

- Desentendimentos quanto aos requisitos, tanto por parte do cliente quanto da equipe de desenvolvimento, decorrentes de falhas no processo de comunicação.
- Insuficiência ou ausência na definição dos requisitos funcionais e não funcionais, o que pode ocasionar lacunas no entendimento e comprometer a execução do projeto.
- Alterações frequentes nos requisitos, sejam eles funcionais, não funcionais ou relacionados a aspectos de negócio, mercado ou tecnologia.
- Ineficácia na gestão de mudanças de requisitos, resultando em impactos significativos no desenvolvimento e na estabilidade do projeto.

Esses desafios têm levado a consequências diretas, incluindo:

- Entregas insatisfatórias: A falta de estrutura e processos adequados resulta na entrega de sistemas que não atendem às necessidades e expectativas dos clientes.
- Declínio no comprometimento: A empresa observa um declínio no comprometimento e engajamento tanto de seus profissionais quanto de seus clientes, o que impacta negativamente o desempenho e a satisfação.
- Riscos não gerenciados: A falta de processos robustos para gerenciar riscos durante a elicitação de requisitos aumenta a probabilidade de problemas e complicações durante o projeto.
- Ausência de processos estruturados: A empresa não possui processos bem definidos na área de desenvolvimento de software, o que resulta em inconsistências no levantamento de requisitos e em um ambiente propício para complicações durante o projeto.
- Inexistência de ferramentas adequadas: A ausência de ferramentas eficazes para o controle de demandas e a gestão de projetos dificulta o acompanhamento e a coordenação das atividades, aumentando a complexidade das entregas.
- Metodologia ágil não definida: Não há um método clássico nem tão pouco uma metodologia ágil sólida na fase de elicitação de requisitos e no desenvolvimento de software, tornando assim a empresa muito distante de uma postura competitiva, inovadora e de excelência no nicho de desenvolvimento de software. Empresas principalmente de tecnologia e inovação precisam ter processos ágeis para uma melhor adaptação às mudanças no mercado competitivo[91].
- Riscos financeiros e reclamações: A falta de processos bem definidos e ferramentas apropriadas aumenta os riscos operacionais e financeiros, levando a reclamações e penalizações por parte das empresas contratantes.

5.2.2 Análise dos riscos

Nesta seção, estruturou-se o processo de elaboração da matriz de Probabilidade e Impacto a partir da realização de uma sessão de brainstorming. Ressalta-se que são apresentados os riscos identificados no processo de elicitação de requisitos, vinculados direta ou indiretamente a esse processo. A condução do estudo contou com a colaboração de profissionais disponibilizados pela empresa. Para a realização do trabalho, adotou-se o modelo de gestão de riscos descrito na ISO 31000:2018, em conjunto com as ferramentas e técnicas recomendadas pela ISO 31010:2012, aplicadas ao contexto da gestão de riscos no processo de elicitação de requisitos.

A matriz apresentada a seguir fornece uma visão geral da avaliação de cada risco em termos de probabilidade e impacto. Destaca-se que essa avaliação fundamentou-se em análises detalhadas e em discussões entre os participantes do estudo, de modo a refletir as condições reais e o ambiente de negócios, conforme apresentado na figura 5.6.

RISCO	PROBABILIDADE	IMPACTO
R01 - Definições imprecisas de negócio, escopo e objetivos	Alta	Alto
R02 - Desentendimentos sobre os requisitos	Alta	Muito alto
R03 - Insuficiência na definição dos requisitos funcionais e não funcionais	Média	Moderado
R04 - Alterações frequentes nos requisitos	Alta	Alto
R05 - Ineficácia na gestão de mudanças dos requisitos	Alta	Muito Alto
R06 - Entregas insatisfatórias	Média	Moderado
R07 - Declínio no comprometimento	Média	Moderado
R08 - Riscos não gerenciados	Média	Moderado
R09 - Ausência de processos estruturados	Alta	Muito Alto
R10 - Inexistência de ferramentas adequadas	Alta	Muito Alto
R11 - Metodologia ágil não definida	Alta	Alto
R12 - Riscos financeiros e reclamações	Média	Alto

Figura 5.6: Relação Probabilidade e Impacto na Matriz de Riscos

Fonte: Elaborada pela autora (2025)

A probabilidade, por padrão, é representada no eixo vertical da matriz, indicando o quão provável é a ocorrência de um risco. Essa medida reflete a frequência esperada de materialização de determinado risco, ou seja, avalia-se a possibilidade de ocorrência do evento. A probabilidade é, geralmente, categorizada em níveis como muito baixo, baixo,

moderado, alto e muito alto, os quais podem ser quantificados em percentuais para uma compreensão mais clara, conforme apresentado a seguir [92].

- Muito baixo: 1 a 10
- Baixo: 11 a 30
- Moderado: 31 a 50
- Alto: 51 a 70
- Muito alto: 71 a 90

Essa escala percentual auxilia na avaliação quantitativa dos riscos, contribuindo para a análise e para a tomada de decisão.

O impacto, por padrão, é representado no eixo horizontal da matriz e refere-se às consequências decorrentes da ocorrência de um risco, abrangendo possíveis prejuízos ou benefícios. As consequências podem ser negativas, como prejuízos financeiros, perda de clientes ou danos a equipamentos, ou positivas, como o surgimento de novas oportunidades de negócio, a adoção de novas tecnologias ou a redução de custos. Para fins de mensuração, o impacto é categorizado em níveis como muito baixo, baixo, moderado, alto e muito alto [92].

Na Matriz de Riscos, a relação entre a probabilidade, representada no eixo vertical, e o impacto, representado no eixo horizontal, é fundamental para a avaliação dos riscos. A matriz é, geralmente, estruturada com cinco níveis tanto para a probabilidade quanto para o impacto, permitindo uma visualização clara da gravidade potencial dos riscos. Ressalta-se, entretanto, que o número de níveis pode ser ajustado de acordo com as necessidades da análise. No presente estudo, adotaram-se cinco níveis de impacto, conforme apresentado na figura 5.7.

PROBABILIDADE	90%	Média	Média	Alta	Alta	Alta
	70%	Baixa	Média	Média	Alta	Alta
	50%	Baixa	Baixa	Média	Alta	Alta
	30%	Baixa	Baixa	Média	Média	Alta
	10%	Baixa	Baixa	Baixa	Baixa	Média
		Muito Baixo	Baixo	Moderado	Alto	Muito Alto
		IMPACTO				

Figura 5.7: Matriz de Probabilidade e Impacto

Fonte: Elaborada pela autora (2025)

Os critérios e as respectivas descrições adotados para a avaliação da probabilidade foram definidos no processo de gestão de riscos, conforme apresentado na figura 5.8.

Probabilidade		Descrição dos critérios de probabilidade
Numérica	Descritiva	
1% a 10%	Muito baixa	Não é provável que aconteça
11% a 30%	Baixa	Pode ser que ocorra uma vez dentro de um ano
31% a 50%	Moderada	Pode ser que ocorra mais de uma vez dentro de um ano
51% a 70%	Alta	Pode ser que ocorra mensalmente
71% a 90%	Muito alta	Pode ser que ocorra semanalmente

Figura 5.8: Níveis de Probabilidade

Fonte: Elaborada pela autora (2025)

Os critérios e as respectivas descrições adotados para a avaliação do impacto foram definidos no processo de gestão de riscos, conforme apresentado na figura 5.9.

Impacto	Descrição dos critérios de impacto
Muito baixo	Os riscos possuem consequências pouco significativas
Baixo	Os riscos possuem consequências reversíveis em curto e médio prazo com custos pouco significativos
Moderado	Os riscos possuem consequências reversíveis em curto e médio prazo com custos baixos
Alto	Os riscos possuem consequências reversíveis em curto e médio prazo com custos altos
Muito alto	Os riscos possuem consequências irreversíveis ou com custos inviáveis

Figura 5.9: Critérios de Impacto
Fonte: Elaborada pela autora (2025)

Ressalta-se que o processo de análise de riscos foi conduzido de forma colaborativa, com a participação de todos os envolvidos no estudo. Por meio da utilização da matriz de probabilidade e impacto e da realização de sessões de brainstorming, possibilitou-se o engajamento de gerentes e analistas no compartilhamento de percepções acerca dos principais riscos associados à fase de elicitação de requisitos. Essa abordagem favoreceu uma análise abrangente e aprofundada dos riscos identificados. As discussões realizadas mostraram-se relevantes, contribuindo de maneira significativa para a compreensão do nível real de impacto e da probabilidade dos riscos avaliados, conforme apresentado na figura 5.10.

PROBABILIDADE	90%	Média	Média	R08 - Riscos não gerenciados	R11 - Metodologia ágil não definida	R02 - Desentendimentos sobre os requisitos
	70%	Baixa	Média	R07 - Declínio no comprometimento	R04 - Alterações frequentes nos requisitos	R09 - Ausência de processos estruturados
	50%	Baixa	Baixa	R03 - Insuficiência na definição dos requisitos funcionais e não funcionais	R01 - Definições imprecisas de negócio, escopo e objetivos	R10 - Inexistência de ferramentas adequadas
	30%	Baixa	Baixa	R06 - Entregas insatisfatórias	R12 - Riscos financeiros e reclamações	R05 - Ineficácia na gestão de mudanças dos requisitos
	10%	Baixa	Baixa	Baixa	Baixa	Média
		Muito Baixo	Baixo	Moderado	Alto	Muito Alto
IMPACTO						

Figura 5.10: Critérios de Impacto
 Fonte: Elaborada pela autora (2025)

Após a análise dos riscos, procedeu-se à etapa de avaliação, utilizando-se um formulário com o objetivo de identificar os principais riscos a serem abordados neste estudo. Ressalta-se que os riscos elencados no formulário foram categorizados de acordo com os níveis de probabilidade (moderada, alta e muito alta) e de impacto (moderado, alto e muito alto).

5.2.3 Avaliação dos riscos

Com base na análise dos riscos e considerando a disponibilidade da equipe mencionada no tópico 5.1.7, em conjunto com a orientação de um especialista em gestão de riscos, optou-se por adotar a técnica de formulário para elencar os principais riscos que serão tratados neste estudo. A escolha do formulário como instrumento metodológico desta pesquisa fundamenta-se na necessidade de coletar percepções, atitudes, julgamentos e experiências dos participantes envolvidos no processo de elicitação de requisitos e na gestão de riscos da organização analisada. Diferentemente de formulários, que se destinam majoritariamente à coleta de dados objetivos e cadastrais, o formulário mostra-se mais adequado a investigações que demandam a compreensão de fatores subjetivos, como a percepção de analistas, gestores e stakeholders acerca de eventos de risco, bem como de seus níveis de probabilidade e impacto.

Instrumentos de pesquisa estruturados assumem papel relevante no contexto da Engenharia de Requisitos, uma vez que possibilitam captar a visão dos atores envolvidos sobre problemas, dificuldades e riscos associados ao processo de desenvolvimento. A coleta sistemática da percepção dos participantes é fundamental para identificar fragilidades organizacionais e compreender como o processo é vivenciado no contexto real, o que reforça a adequação do questionário como meio para a obtenção de dados confiáveis sobre a maturidade do processo e a exposição a riscos durante a elicitação de requisitos. Ademais, a utilização da Escala Likert no questionário permite que percepções qualitativas sejam convertidas em dados quantitativos passíveis de análise, fortalecendo a confiabilidade dos resultados e possibilitando comparações entre diferentes grupos ou etapas do processo [93].

Com o objetivo de identificar, sob a perspectiva dos colaboradores da área de Tecnologia da Informação da empresa Libert Tecnologia, quais dos riscos previamente mapeados são percebidos como mais críticos, elaborou-se um formulário composto por oito afirmações, correspondentes aos riscos R01, R02, R04, R05, R08, R09, R10 e R11. O instrumento foi estruturado de forma a permitir que cada participante indicasse seu grau de concordância por meio de uma escala do tipo Likert, possibilitando a manifestação das percepções individuais a partir das seguintes opções de resposta:

- 1 - Concordo totalmente
- 2 - Concordo parcialmente
- 3 - Nenhuma das opções
- 4 - Discordo parcialmente
- 5 - Discordo totalmente

Cada uma das opções expressa diferentes níveis de concordância ou discordância com uma declaração ou pergunta, conforme listado:

Concordo totalmente: Quando o entrevistado está totalmente de acordo com a declaração ou pergunta apresentada. Isso significa que o respondente apoia completamente a afirmação feita ou concorda integralmente com a questão em análise, sem qualquer reserva ou dúvida. Essa resposta indica um alto nível de concordância e concordância total com o conteúdo apresentado no formulário, demonstrando uma adesão completa à perspectiva ou afirmação em questão.

Concordo parcialmente: Quando o entrevistado seleciona esta opção, significa que ele está em algum grau de acordo com a declaração ou pergunta, mas não completamente.

Há alguma concordância, mas também alguma reserva ou dúvida em relação ao conteúdo apresentado.

Nenhuma das opções: Essa opção é escolhida quando o entrevistado não se identifica com nenhuma das alternativas fornecidas. Isso indica que as opções apresentadas não refletem adequadamente a opinião ou perspectiva do entrevistado sobre a declaração ou pergunta.

Discordo parcialmente: Quando o entrevistado opta por esta alternativa, significa que ele está em algum grau de desacordo com a declaração ou pergunta, mas também pode haver algumas áreas em que ele concorda ou tem dúvidas. É uma forma de expressar um desacordo parcial com o conteúdo apresentado.

Discordo totalmente: Essa opção é selecionada quando o entrevistado está em completo desacordo com a declaração ou pergunta. Não há concordância em nenhuma medida, e o entrevistado acredita que a afirmação está errada, inadequada ou não se aplica à sua perspectiva.

Essas opções permitem que os entrevistados forneçam uma avaliação mais precisa de suas opiniões e sentimentos em relação às questões apresentadas no formulário, criando um espectro de respostas que vai desde a concordância total até o completo desacordo, com nuances no meio.

Para a criação do formulário, utilizamos a plataforma "Google Forms."

A fim de disponibilizar o formulário, foi criado um link utilizando o Google Forms e, em seguida, esse link foi enviado individualmente via e-mail aos profissionais selecionados, esse instrumento utilizado para essa coleta de dados com o enunciado das questões e as orientações de preenchimento encontra-se reproduzido integralmente no Apêndice A.2.

As categorias dos riscos no formulário foram definidas conforme a probabilidade (moderada, alta e muito alta) e o impacto (moderado, alto e muito alto). Assim, o formulário abordou as seguintes perguntas:

- R01. Entendo que as definições de negócios, escopo e objetivos do projeto são imprecisas.
- R02. Há desentendimentos frequentes sobre os requisitos do projeto.
- R04. Ocorrem alterações frequentes nos requisitos.
- R05. A gestão de mudanças dos requisitos é ineficaz.
- R08. Os riscos não são gerenciados adequadamente.
- R09. Há uma ausência de processo estruturado na fase de elicitação de requisitos.
- R10. Inexistência de ferramentas adequadas.

- R11. Metodologia ágil não definida.

O formulário foi distribuído aos profissionais da área de Tecnologia da Informação da empresa no período de 18 a 22 de março de 2025. Atualmente, o setor de tecnologia conta com 56 profissionais, dos quais 35 estão envolvidos no desenvolvimento de sistemas.

Obteve-se a participação de 32 profissionais da área de desenvolvimento de software, os quais responderam ao formulário dentro do prazo estabelecido. Com isso foi possível obter os seguintes resultados, conforme apresentado nas figuras A.19, A.20, A.21 e A.22.

Como resultado do formulário utilizou-se a resposta: "Concordo totalmente" como fator de critério de priorização ao risco. Com isso, conforme apresentado nos resultados do formulário, os riscos elencados como mais prioritários foram:

- **R04 - Alterações frequentes nos requisitos - 37,5%**
- **R09 - Ausência de processos estruturados - 59,4%**
- **R10 - Inexistência de ferramentas adequadas - 50%**
- **R11 - Metodologia ágil não definida - 56,3%**

Neste cenário, definiu-se o objetivo central da organização: "Impulsionar uma transformação ágil na fase de obtenção de requisitos."

Com essas diretrizes estabelecidas, foi possível desenvolver o tratamento dos riscos, incluindo a identificação dos fatores de risco, o impacto desses riscos nos negócios da empresa e nos clientes, um plano de ação para mitigação desses riscos e os controles necessários para gerenciá-los. Essas informações estão detalhadas nas figuras a seguir: Figura 5.11, figura 5.12, figura 5.13 e figura 5.14.

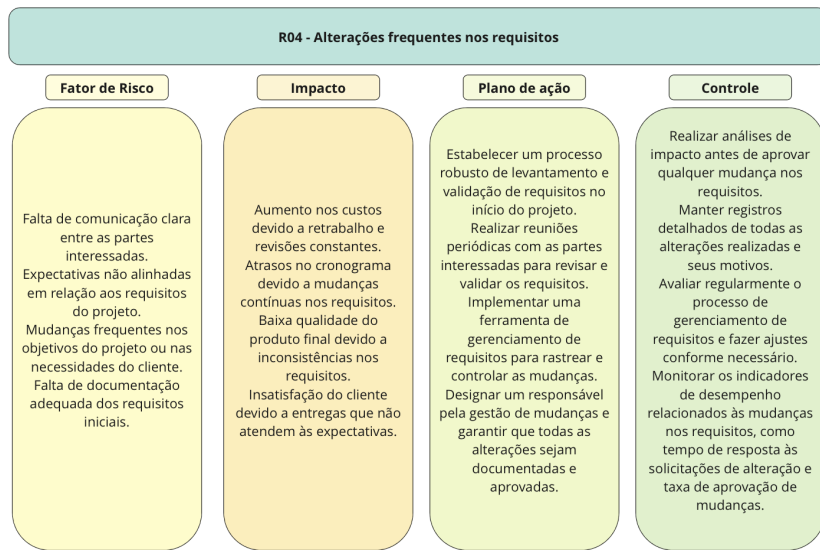


Figura 5.11: Risco - Alterações frequentes nos requisitos

Fonte: Elaborada pela autora (2025)

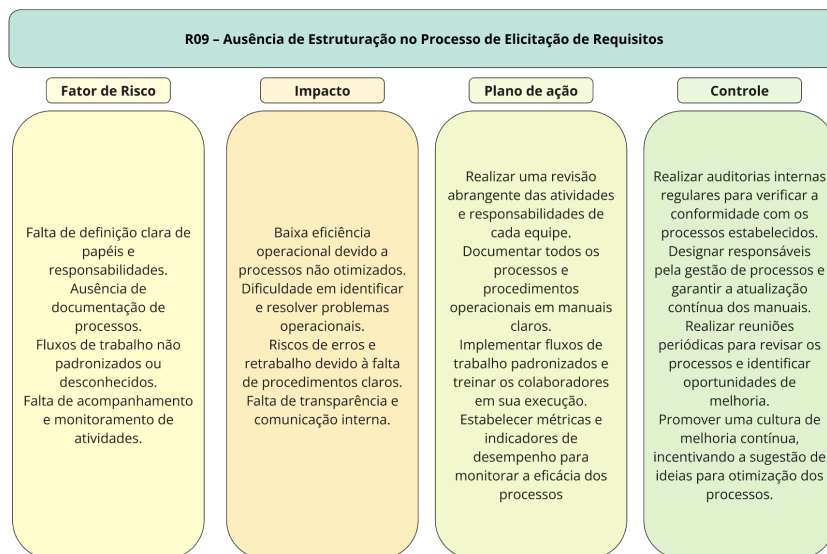


Figura 5.12: Risco - Ausência de Estruturação no Processo de Elicitação de Requisitos

Fonte: Elaborada pela autora (2025)

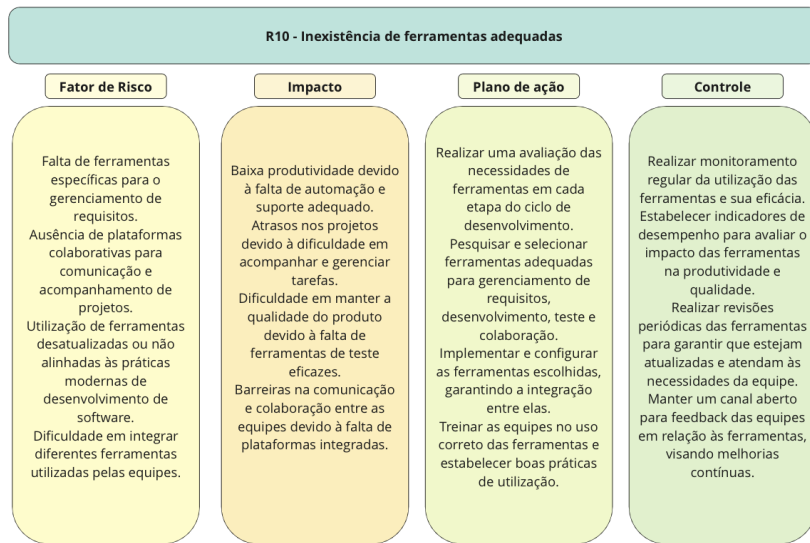


Figura 5.13: Risco - Inexistência de ferramentas adequadas

Fonte: Elaborada pela autora (2025)

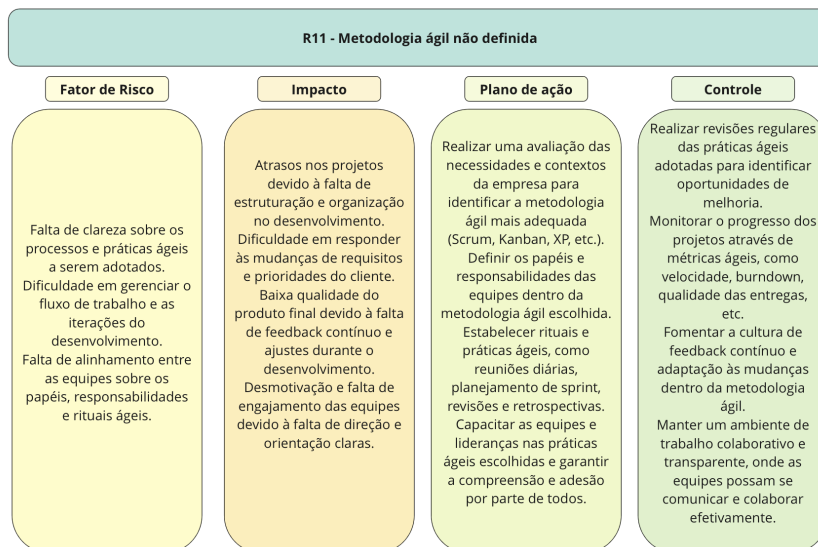


Figura 5.14: Risco - Metodologia ágil não definida

Fonte: Elaborada pela autora (2025)

Conforme priorização dos riscos avaliados processo de elicitação de requisitos, foi proposto para a área gestora uma modelagem de processo TO-BE com a mitigação dos riscos identificados no processo AS-IS e priorizados a partir da avaliação e aplicação do formulário aos profissionais da área de desenvolvimento de software da empresa Liberty tecnologia. Na próxima sessão será apresentado o tratamento dos riscos através da modelagem do processo TO-BE.

5.3 Elaboração do processo (TO-BE) para a fase de elicitação de requisitos

Como forma de mitigar/diminuir os riscos identificados no processo AS-IS na fase de elicitação de requisitos, e priorizados após a aplicação do formulário, foi proposta a modelagem do processo TO-BE de maneira a melhorar o processo anterior e mitigar os riscos selecionados, conforme figura 5.15.

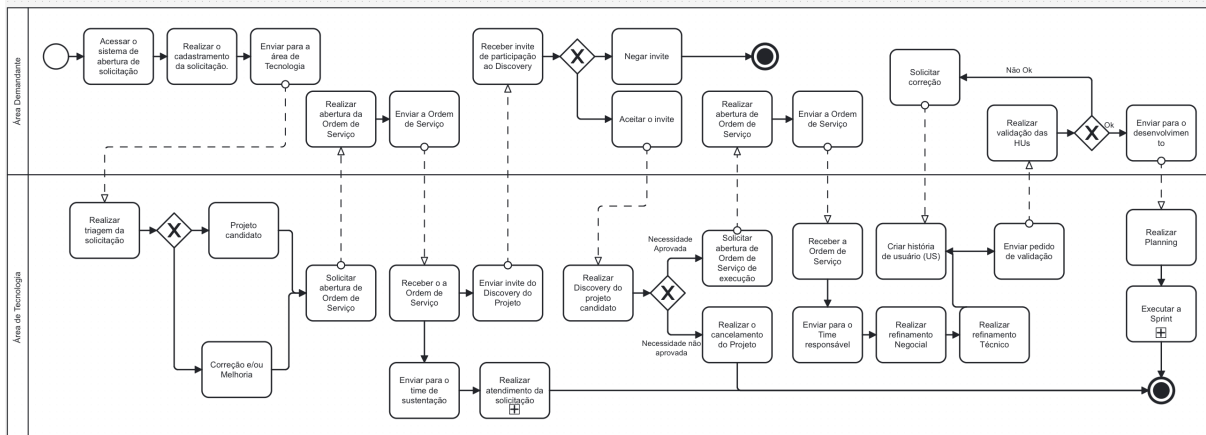


Figura 5.15: Mapeamento TO-BE
 Fonte: Elaborada pela autora (2025)

Com base em análise comparativa entre os processos AS-IS e TO-BE o mapeamento TO-BE evidencia uma evolução significativa na estruturação da elicitação de requisitos. Enquanto o processo AS-IS apresenta um fluxo mais linear, com menor formalização na entrada da demanda e na condução da elicitação, o processo TO-BE introduz mecanismos de controle e governança ao longo de todo o ciclo. Dentre as principais alterações, destacam-se a formalização da abertura da demanda por meio de sistema, a realização de triagem, a inclusão de uma etapa de Discovery com decisão de continuidade, a separação entre ordens de serviço de iniciação e execução, a realização de refinamentos negociais e técnicos, a validação estruturada das histórias de usuário e a adoção explícita de práticas ágeis, como planning e sprint.

Essas mudanças contribuem diretamente para o tratamento dos riscos identificados no estudo. O risco R04 (alterações frequentes nos requisitos) é mitigado pelo Discovery, pelos refinamentos e pelo ciclo formal de validação das histórias de usuário; o risco R09 (ausência de processos estruturados) é tratado pela padronização das atividades, papéis e pontos de decisão ao longo do fluxo; o risco R10 (inexistência de ferramentas adequadas) é enfrentado pela institucionalização do uso de sistemas para abertura, rastreamento e

gestão das demandas; e o risco R11 (metodologia ágil não definida) é abordado por meio da incorporação explícita de cerimônias e práticas ágeis no processo de execução. Dessa forma, o mapeamento TO-BE estabelece um fluxo mais previsível, colaborativo e alinhado às boas práticas de Engenharia de Requisitos e gestão de riscos.

A proposta visa otimizar a elicitação de requisitos na empresa Liberty Tecnologia, por meio da revisão e aprimoramento dos processos, adoção de ferramentas adequadas, definição clara de papéis e uso de metodologias alinhadas às práticas ágeis. Essa melhoria será conduzida com base em um plano de ação estruturado, com foco na qualidade do processo e na busca pela melhoria contínua, reduzindo os seguintes fatores de risco identificados:

- F04 – Falta de comunicação clara entre as partes interessadas;
- F04 – Mudanças frequentes nos objetivos do projeto ou nas necessidades do cliente;
- F09 – Falta de definição clara de papéis e responsabilidades;
- F09 – Ausência de documentação de processos;
- F10 – Falta de ferramentas específicas para o gerenciamento de requisitos;
- F10 – Ausência de plataformas colaborativas para comunicação e acompanhamento de projetos;
- F11 – Falta de clareza sobre os processos e práticas ágeis a serem adotados;
- F11 – Dificuldade em gerenciar o fluxo de trabalho e as iterações do desenvolvimento.

Esses fatores contribuem diretamente para riscos críticos que afetam os objetivos organizacionais, tais como:

- R04 – Alterações frequentes nos requisitos;
- R09 – Ausência de processos estruturados;
- R10 – Inexistência de ferramentas adequadas;
- R11 – Metodologia ágil não definida.

Dessa forma, a solução proposta busca estabelecer um novo modelo de elicitação de requisitos mais robusto e eficiente, capaz de minimizar os riscos priorizados pela equipe de desenvolvimento da Liberty Tecnologia, contribuindo para a melhoria da qualidade do produto final, o aumento da previsibilidade dos projetos e a redução de retrabalho ao longo do ciclo de vida do software. Nesse contexto, torna-se necessária a definição de um Plano de Ação, no qual são detalhadas as iniciativas, responsabilidades e medidas a serem adotadas para operacionalizar as melhorias propostas e assegurar o tratamento efetivo dos riscos identificados.

5.4 Elaboração de Plano de Ação

O plano de ação é uma ferramenta essencial para organizar atividades, definir responsabilidades, estabelecer prazos, monitorar o progresso, apoiar a tomada de decisões, alinhar objetivos, melhorar a comunicação e aumentar a eficiência operacional. Para este estudo, a elaboração do plano de ação utilizará a técnica 5W2H, que estrutura o planejamento a partir de sete perguntas fundamentais: o quê, por quê, onde, quando, quem, como e quanto. Amplamente aplicada em gestão de projetos por sua simplicidade e clareza, a 5W2H facilita o entendimento e a execução das atividades, garantindo alinhamento entre os envolvidos.

Com base nos riscos mapeados — R04 (Alterações frequentes nos requisitos), R09 (Ausência de processos estruturados), R10 (Inexistência de ferramentas adequadas) e R11 (Metodologia ágil não definida) foi elaborado um Plano de Ação estruturado para orientar a implementação das melhorias necessárias. As ações propostas visam aprimorar a engenharia de requisitos, fortalecer a maturidade dos processos internos e criar um ambiente mais eficiente, transparente e alinhado às boas práticas de gestão e desenvolvimento de software. A partir dessa abordagem, será construído o plano de ação que orientará as etapas necessárias para alcançar os objetivos definidos no estudo, conforme figura 5.16:

O QUÊ?	POR QUÊ?	ONDE?	QUANDO?	QUEM?	COMO?	QUANTO CUSTA?
Estabelecer processo de levantamento de requisitos	Melhorar processo atual Mitigar riscos de alterações repetida de requisitos	Área de desenvolvimento de sistemas	25/04 a 29/04	Especialistas participantes do estudo em questão	Reunião com os especialistas Levantamento e validação de requisitos	Tempo e esforço
Realizar reuniões periódicas	Validar requisitos	Área de desenvolvimento de sistemas	25/04 /a 25/06	Especialistas participantes do estudo em questão	Revisar e validar requisitos	Tempo e esforço
Implementar ferramenta de gestão	Controlar demandas e gerir mudanças	Área de desenvolvimento de sistemas	01/06 a 25/08	Especialistas participantes do estudo em questão	Configurar ferramenta e treinar equipes	Tempo e esforço
Implantar metodologia ágil	Resposta mais rápida às mudanças Melhoria na comunicação Redução de custos e prazos	Área de desenvolvimento de sistemas	01/05 a 25/09	Especialistas participantes do estudo em questão	Realização de Workshop Estruturar equipes Capacitar equipes no método SCRUM	Tempo e esforço
Criar Proposta de elicitação de requisitos	Alinhar visões Identificar requisitos Promover a colaboração Reduzir desperdícios	Área de desenvolvimento de sistemas	25/04 a 10/07	Fernanda Lima	Apresentar aos envolvidos pelo estudo Realizar uma POC - Prova de Conceito	Tempo e esforço
Feedback, revisão e ajustes no protótipo	Melhorar a proposta apresentada	Área de desenvolvimento de sistemas	13/08 a 17/10	Fernanda Lima	Especialistas participantes do estudo em questão	Tempo e esforço
Validação do plano de ação de melhorias do processo e validação da proposta.	Finalização do estudo	Área de desenvolvimento de sistemas	20/10 a 24/11	Fernanda Lima	Especialistas participantes do estudo em questão	Tempo e esforço

Figura 5.16: Plano de Ação

Fonte: Elaborada pela autora (2025)

A figura apresentada sintetiza o plano de ação adotado para implementar e validar a proposta de melhoria no processo de elicitação de requisitos, estruturado com base na lógica 5W2H (O quê, Por quê, Onde, Quando, Quem, Como e Quanto custa). Esse instrumento permitiu organizar as atividades, definir responsabilidades, estabelecer prazos e garantir maior controle e rastreabilidade das ações realizadas.

As etapas contemplaram a estruturação do processo de levantamento de requisitos, a realização de reuniões periódicas de validação, a implementação de ferramenta de gestão, a adoção de metodologia ágil e o desenvolvimento de uma proposta formal apoiada por prova de conceito. Posteriormente, foram conduzidas atividades de feedback, ajustes no protótipo e validação final do plano de melhorias.

Executadas na área de desenvolvimento de sistemas, com participação dos especialistas envolvidos e da pesquisadora, as ações foram dimensionadas em termos de tempo e

esforço. O board, portanto, funcionou como instrumento de organização e governança da intervenção, demonstrando a viabilidade prática da solução proposta.

5.5 Desenvolvimento do Protótipo Computacional UpLean

A proposta em análise consiste na concepção de um protótipo navegável denominado *UpLean*.

O *UpLean* configura-se como uma plataforma digital projetada para estruturar, organizar e conduzir de forma integrada a fase de *Upstream*, correspondente ao processo de *discovery* no contexto da Lean Inception. A ferramenta permite que um facilitador, no papel de organizador, crie uma iniciativa, inclua ou convide participantes e conduza a equipe ao longo das atividades metodologicamente estabelecidas. Ao final do processo, o sistema realiza a consolidação automática dos resultados em um documento unificado, além de possibilitar a integração de épicos, *features* e histórias de usuário com o Jira, favorecendo a transição estruturada do *Discovery* para o *Delivery*.

Considerando a relevância desses artefatos para a modelagem de soluções ágeis orientadas ao usuário e alinhadas aos objetivos de negócio, a seção seguinte apresenta a definição do épico, das *features* e das *user stories* elaboradas a partir da Lean Inception conduzida no âmbito desta proposta.

No contexto do desenvolvimento ágil de *software*, especialmente em *frameworks* como Scrum e Kanban, a organização dos requisitos em diferentes níveis de granularidade mostra-se essencial para assegurar clareza, priorização adequada e alinhamento estratégico. Entre esses níveis, destacam-se três categorias amplamente adotadas:

- Épico
- Feature
- User Story

Épico - Pode ser definido como uma narrativa de alto nível que representa uma necessidade ampla e estratégica do negócio. Devido à sua complexidade e escopo, um épico normalmente não pode ser implementado em uma única iteração e precisa ser dividido em funcionalidades menores para sua execução [76].

Feature São subconjuntos funcionais derivados dos épicos. Representam funcionalidades específicas, de valor claro para o usuário ou negócio, que podem ser entregues em um prazo mais curto, geralmente em uma ou poucas iterações. São mais detalhadas que os épicos, mas ainda não chegam ao nível de granularidade necessário para o desenvolvimento direto [49].

User story - Constituem a unidade básica de trabalho no desenvolvimento ágil. Elas descrevem, sob a perspectiva do usuário, uma funcionalidade ou necessidade que o sistema deve atender. Segundo Cohn (2004), histórias de usuários são uma técnica eficaz para capturar requisitos funcionais de forma enxuta, mantendo o foco no valor entregue ao usuário. Elas são essenciais para a comunicação entre as partes interessadas e para orientar o desenvolvimento incremental, promovendo alinhamento com os objetivos de negócio. Uma boa história de usuário segue o critério INVEST (Independent, Negotiable, Valuable, Estimable, Small e Testable), proposto por Wake (2003), garantindo que cada item seja funcionalmente útil, tecnicamente viável e passível de validação [22].

Para a elaboração da proposta de solução apresentado neste estudo, foi realizada a etapa de descoberta utilizando a técnica Lean Inception na ferramenta Miro, com doze profissionais, onde foi possível percorrer todas as etapas essenciais do Discovery, incluindo o alinhamento do problema, a construção da visão do produto, o mapeamento de personas e jornadas, a exploração colaborativa de funcionalidades e a estruturação do backlog. O processo metodológico resultou em um conjunto significativo de artefatos estratégicos para a concepção do UPLean e fundamentou a transição para as etapas de prototipação e desenvolvimento.

Como resultado direto do trabalho desenvolvido, foram identificados 8 épicos, 36 *features* e 37 histórias de usuário, devidamente organizados no backlog estruturado do sistema.

As especificações detalhadas que incluem o levantamento de requisitos e a redação completa das histórias de usuário encontram-se apresentadas no Apêndice desta dissertação, com o propósito de assegurar rastreabilidade, transparência metodológica e consistência documental.

A consolidação desses artefatos possibilitou a definição das interfaces e a modelagem das telas do sistema, o que resultou na construção do protótipo navegável desenvolvido na ferramenta Figma.

Diante do conjunto de artefatos apresentados no Apêndice A.3 deste documento, o desenvolvimento do protótipo navegável constituiu a etapa seguinte e indispensável no processo de concepção da solução. A prototipação possibilitou o teste do fluxo das interações, a antecipação de possíveis dúvidas dos usuários, a redução de riscos relacionados à interpretação dos requisitos e o aumento da assertividade na transição para o desenvolvimento. Adicionalmente, o protótipo assumiu o papel de instrumento de validação incremental, ao assegurar que as funcionalidades priorizadas mantivessem alinhamento com as necessidades de negócio e com a experiência esperada pelos usuários.

O protótipo navegável do UPLean construído com base nas telas documentadas e descritas no material técnico do sistema espelha de forma fiel os comportamentos, fluxos e elementos visuais definidos durante a Lean Inception. Essa correspondência entre o

backlog estruturado e a prototipação assegura rastreabilidade entre as etapas do Discovery e a posterior implementação, atendendo às recomendações das boas práticas de engenharia de requisitos preconizadas por Sommerville (2011) e pela ISO/IEC 29148 [27] [45]. As telas prototipadas podem ser visualizados no apêndice A.4 desse documento.

Após essa etapa de validação visual e funcional, iniciou-se o processo de implementação da solução. A construção do UPLean foi orientada pelos princípios ágeis de iteração, colaboração constante e entrega contínua de valor, permitindo que ajustes fossem incorporados de maneira eficiente a partir do feedback das partes interessadas. O desenvolvimento inicial priorizou os fluxos essenciais identificados no conjunto de épicos e features definidos, estabelecendo, assim, uma espinha dorsal sólida para evolução incremental da plataforma.

Essa abordagem iterativa proporcionou não apenas o amadurecimento progressivo do sistema, mas também a oportunidade de avaliar continuamente a adequação das funcionalidades propostas. Dessa forma, o desenvolvimento do UPLean tornou-se um exercício prático de integração entre engenharia de requisitos, metodologias ágeis e princípios de Lean Inception, reforçando a aderência metodológica e a viabilidade da solução como ferramenta de apoio ao processo de Upstream.

Na seção seguinte, são apresentados os principais resultados obtidos durante a validação da proposta e o desenvolvimento da solução, incluindo demonstração inicial do sistema, funcionalidades implementadas e uma análise crítica sobre o atendimento aos requisitos definidos no Discovery. Esses elementos permitirão avaliar a eficácia do UPLean como plataforma digital voltada à estruturação de iniciativas, colaboração entre equipes e consolidação automatizada de artefatos de produto.

5.5.1 Simulação de utilização do protótipo

Objetivo da simulação:

A simulação de utilização do protótipo teve como objetivo verificar, em um contexto próximo do real, se o UPLean atende às necessidades do processo de elicitação de requisitos com gestão de riscos, avaliando principalmente:

- Funcionalidade percebida: se o protótipo contempla as etapas esperadas do processo e permite registrar/organizar as informações necessárias.
- Usabilidade: Se usuários conseguem executar as tarefas com clareza, pouco esforço e sem ambiguidades.
- Aderência ao fluxo de trabalho: Se o protótipo “encaixa” no modo como o time realmente executa elicitação, refinamento e decisões.

- Compreensão dos artefatos: Se as fases e subetapas são entendidas pelos participantes (ex.: Visão do Produto, definição do produto, metas, personas etc.).
- Potencial de apoio à gestão de riscos: Se a ferramenta, do ponto de vista dos usuários, contribui para reduzir falhas típicas da elicitação (omissões, ruídos, retrabalho, conflitos e mudanças tardias).

Ambiente e recursos utilizados:

A simulação foi executada utilizando um protótipo navegável, apresentado em ambiente controlado, com os seguintes recursos:

- Link do protótipo navegável;
- Roteiro de tarefas (cenários);
- Instrumentos de coleta (Checklist com as etapas solicitadas pela empresa);
- Registro de observações (via planilha).

Com isso, ficou mais simples atuar nas possíveis melhorias proposta pelos colaboradores desse estudo.

Participantes e papéis: A simulação foi planejada para envolver participantes com perfis distintos, aproximando-se do uso real do sistema:

- Administrador: Responsável por organizar iniciativas, aprovar usuários e acompanhar o andamento das fases.
- Stakeholder/Negócio/Participante: Contribui com visão, metas, validações de escopo e decisões de priorização.
- Técnico: contribui com restrições técnicas, riscos de implementação e viabilidade.

Importante mencionar que a escolha desses perfis se deve ao próprio modelo do UPLearn, que prevê tipos de usuário e dashboards por papel, além de colaboração ao longo das fases.

Cenário de simulação (caso guiado)

Para garantir consistência entre sessões e comparabilidade dos resultados, a simulação utilizou um caso guiado, partimos da ideia de realizarmos o discovery de uma iniciativa de gestão de processos e pagamentos. Com isso, foram levados em consideração:

- Contexto do produto (problema, público-alvo e objetivo);
- Expectativas de entregas (backlog, MVP, prioridades);

- Pontos de risco típicos (mudanças frequentes, conflitos de interesse, incerteza técnica, prazo curto, dependências externas etc.).

Em suma, seguiu-se para as etapas da Lean Inception a fim de conseguir realizar o levantamento de todas as necessidades do projeto estudado nesta etapa de simulação, conforme disponível no Apêndice A.3.

Roteiro de tarefas:

A simulação foi baseada em tarefas observáveis, seguimos o seguinte roteiro:

Tarefa 1 — Acesso e onboarding:

- Usuário acessa a tela de Login/Cadastro.
- Realiza cadastro e seleciona perfil (Stakeholder ou Técnico).

Simular aprovação do administrador e login:

O perfil Facilitador acessa o sistema e avalia o cadastro do solicitante e vincula ele a um determinado perfil para acessar as iniciativas na qual ele será vinculado.

Tarefa 2 — Navegação inicial e entendimento do dashboard:

- Usuário acessa o Dashboard e identifica os principais atalhos (Iniciativas, Backlog Estruturado etc.).
- Verifica como o dashboard muda conforme o papel.

Tarefa 3 — Criação e gestão de iniciativa:

- Administrador cria uma Nova Iniciativa.
- Define título, descrição e status.
- Inclui participantes (stakeholders e técnicos).
- Acessa o Detalhe da Iniciativa e visualiza o progresso das fases.

Tarefa 4 — Execução das fases do processo:

- Usuários acessam a Fase 1 – Alinhamento e Contexto.
- Usuários acessam a Fase 2 – Exploração das funcionalidades.
- Usuários acessam a Fase 3 - Definição do Backlog do Produto.
- Usuários acessam a Fase 4 - Definição do MVP
- tem Usuários acessam a Fase 5 - Detalhamento e Encerramento

Após a simulação de utilização do protótipo, seguimos para os feedbacks e melhorias.

5.5.2 Feedback, Revisão e Ajustes no Protótipo da Solução

Na etapa de consolidação, o usuário acessou o Backlog Estruturado gerado pelo sistema a partir das informações registradas ao longo das fases anteriores do processo de elicitação. Nessa atividade, os participantes foram orientados a analisar a estrutura final do backlog, verificando se os itens apresentados refletem de maneira coerente a visão do produto, as metas estabelecidas, as personas definidas e os entendimentos construídos durante o processo. Essa tarefa tem como objetivo avaliar se o protótipo é capaz de transformar os insumos obtidos nas etapas iniciais em um artefato final organizado, compreensível e passível de utilização como saída do processo, contribuindo para a continuidade do projeto e para a redução de ambiguidades, desalinhamentos e retrabalho, aspectos diretamente relacionados à gestão de riscos na fase de elicitação de requisitos.

Durante a execução do roteiro de tarefas, foi realizada a observação sistemática do comportamento dos participantes, registrando aspectos relevantes relacionados à usabilidade, compreensão do fluxo e aderência do protótipo ao processo de trabalho proposto. Foram documentadas eventuais dúvidas e interrupções manifestadas pelos usuários, como questionamentos sobre a localização de funcionalidades, a sequência das etapas ou o significado de termos e conceitos apresentados no sistema.

Também foram identificados erros de fluxo, tais como caminhos sem saída, dificuldades de retorno entre telas ou sequências de navegação que gerem confusão ou hesitação por parte dos usuários. Quando aplicável, o tempo necessário para a realização das tarefas é considerado como um indicativo complementar do esforço cognitivo exigido pelo sistema. Adicionalmente, foram registrados elementos da interface que provoquem incerteza ou dificuldade de interpretação, bem como sugestões espontâneas de melhorias, ajustes ou funcionalidades não contempladas no protótipo.

Por fim, foram coletadas evidências relacionadas ao valor percebido pelos participantes, caracterizadas por comentários que indiquem que o uso do sistema poderia contribuir para a melhoria do processo de elicitação, auxiliar na organização das informações, evitar problemas recorrentes ou apoiar a tomada de decisão, especialmente no que se refere à mitigação de riscos associados à definição e ao entendimento dos requisitos.

Ao final da simulação de utilização do protótipo, foram gerados diferentes entregáveis que subsidiam a análise dos resultados e a validação da proposta apresentada nesta pesquisa. Entre os principais resultados obtidos, destaca-se a elaboração de uma lista consolidada de achados relacionados à usabilidade e ao entendimento do sistema, evidenciando pontos de dificuldade, ambiguidades e fragilidades identificadas durante a interação dos usuários com o protótipo.

Adicionalmente, foi construída uma lista de funcionalidades desejadas e sugestões de melhoria, a partir das percepções e contribuições dos participantes, indicando oportuna-

des de aprimoramento do sistema e do processo representado. Também foram identificados ajustes necessários no fluxo de navegação e na organização das etapas, com o objetivo de tornar o processo mais fluido, intuitivo e alinhado à prática dos usuários.

Por fim, a simulação forneceu uma percepção qualitativa sobre a aderência do protótipo ao processo de elicitação de requisitos e sobre seu potencial de apoio à gestão de riscos, servindo como base para a revisão do plano de ação, a melhoria do processo proposto e a validação da solução apresentada, conforme os objetivos definidos para esta etapa da pesquisa.

5.5.3 Validação da Proposta de Solução

A validação da proposta de solução demonstrou que o processo de elicitação de requisitos estruturado e apoiado pelo protótipo UPLearn apresenta aderência ao contexto organizacional analisado e potencial para reduzir riscos recorrentes associados à definição de requisitos. O plano de ação executado permitiu comprovar a viabilidade da proposta, evidenciando que a combinação entre práticas estruturadas, abordagem ágil e suporte ferramental contribui para a melhoria da comunicação, da organização das informações e da tomada de decisão ao longo do processo.

Os resultados obtidos indicam que o protótipo não apenas materializa a proposta teórica apresentada nesta pesquisa, mas também se configura como um instrumento promissor para apoiar equipes de desenvolvimento na condução da elicitação de requisitos de forma mais controlada, colaborativa e alinhada aos objetivos do negócio. Apesar das limitações inerentes ao uso de um protótipo e ao contexto específico do estudo, a validação realizada reforça a relevância da solução proposta e aponta caminhos para sua evolução futura, incluindo a implementação completa do sistema e sua aplicação em cenários organizacionais mais amplos.

Capítulo 6

Conclusão

Esta dissertação buscou aprimorar o processo de elicitação de requisitos na empresa Liberty Tecnologia, integrando de forma estratégica a Gestão de Riscos às práticas de desenvolvimento de software. A fase de elicitação foi identificada como o ponto de maior vulnerabilidade, onde falhas de compreensão e especificações imprecisas comprometiam o desempenho técnico e eram as principais razões para o insucesso de projetos de TI na organização. A análise do cenário inicial (AS-IS) revelou que a Liberty enfrentava desafios críticos, como a ausência de processos estruturados, a falta de uma metodologia ágil definida e a inexistência de ferramentas adequadas para o controle de demandas. Através da aplicação de técnicas como a análise SWOT e o brainstorming com especialistas, foram priorizados riscos que geravam retrabalho e insatisfação, destacando-se as alterações frequentes nos requisitos (R04) e a falta de estruturação no processo (R09). Para mitigar essas ameaças, foi proposto um novo modelo de processo (TO-BE) e elaborado um Plano de Ação detalhado sob a metodologia 5W2H. A principal contribuição prática deste trabalho foi o desenvolvimento do protótipo UPLean, uma plataforma digital voltada para organizar a fase de Upstream (discovery) utilizando os princípios da Lean Inception. O UPLean foi projetado para alinhar as visões de negócio e tecnologia, permitindo a criação colaborativa de personas, jornadas e backlogs estruturados com integração ao Jira. A validação da proposta, realizada por meio de simulações com profissionais de TI, confirmou a viabilidade e eficácia da solução. Os participantes observaram que o suporte ferramental e o fluxo estruturado auxiliam na organização das informações e na tomada de decisão fundamentada, reduzindo ruídos de comunicação e ambiguidades. Conclui-se que a harmonização entre a gestão de riscos e a engenharia de requisitos é um diferencial estratégico no desenvolvimento ágil, garantindo entregas mais assertivas. No que tange as limitações da Pesquisa embora os resultados tenham sido satisfatórios, a pesquisa apresenta limitações importantes. O estudo concentrou-se em um único estudo de caso na empresa Liberty Tecnologia, o que pode restringir a generalização dos resultados para outros con-

textos organizacionais. Além disso, a validação foi baseada na utilização de um protótipo navegável e não em um sistema completamente implementado em ambiente de produção contínua, o que limitou a coleta de dados sobre o desempenho técnico real em larga escala. Outro ponto limitante foi o número de projetos analisados durante a fase de análise documental e o tamanho da amostra de participantes na simulação. Como sugestão de trabalhos futuros para a continuidade a este estudo, sugere-se a implementação completa do sistema UPLearn, transformando o protótipo em uma ferramenta funcional integrada aos ecossistemas de desenvolvimento da empresa. Recomenda-se também a aplicação do modelo em diferentes cenários organizacionais e com equipes de variados níveis de maturidade ágil para validar sua escalabilidade e adaptabilidade. Estudos futuros poderiam explorar abordagens mais sistemáticas para a avaliação quantitativa de riscos residuais e examinar os efeitos a longo prazo das ações de mitigação propostas pelo UPLearn no desempenho global dos projetos. Realizar o desenvolvimento de software sem uma elicitación de requisitos baseada em riscos é comparável a construir um edifício em um terreno instável sem uma sondagem prévia do solo. O protótipo UPLearn atua como o estudo de engenharia que identifica fragilidades e propõe a base correta para que o software final seja resiliente, seguro e não desmorone após o investimento de grandes recursos.

Referências

- [1] Schwaber, Ken e Jeff Sutherland: *Le guide scrum*. 2020. xi, 48, 95, 102, 103, 105, 106
- [2] Bourque, Pierre, Robert Dupuis, Alain Abran, James W Moore, Leonard Tripp, Karen Shyne, Bryan Pflug, Marcela Maya e Guy Tremblay: *Guide to the software engineering body of knowledge-a straw man version*. 1998. 2
- [3] Saur, Ricardo Adolfo de Campos: *A tecnologia da informação na reforma do estado: Considerações sobre a prestação de serviços de informática na área pública*. *Ciência da Informação*, 26, 1997. 3
- [4] Ceruzzi, Paul E.: *A History of Modern Computing*. MIT Press, Cambridge, MA, 2ª edição, 2003. 3
- [5] Junior, Ivo Pedro Gonzalez, Leidiane Moreira Penha e Claudiene Maria Silva: *A importância da tecnologia da informação como ferramenta para o processo da gestão hospitalar no setor privado: Um estudo de caso em uma organização hospitalar em feira de santana (ba)*. *Revista de Gestão em Sistemas de Saúde*, 2(1):91–115, 2013. 4
- [6] Vacari, Isaque: *Um estudo empírico sobre a adoção de métodos ágeis para desenvolvimento de software em organizações públicas*. Tese de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, 2015. 4
- [7] Gaikema, Maurice, Mark Donkersloot, Jim Johnson e Hans Mulder: *Increase the success of governmental it-projects*. *Systemics, Cybernetics and Informatics*, 17(1):97–105, 2019. 4, 23
- [8] Vidal, André: *Agile Think Canvas*. Brasport, 2017. 5
- [9] Manuel, Roseneide Francisco *et al.*: *Mudanças tecnológicas e o impacto nas organizações: o processo de comunicação no estudo de caso da empresa angolana 'teleservice, sa'*. Tese de Mestrado, 2017. 5
- [10] Mariano, Ari Melo e Maíra Santos Rocha: *Revisão da literatura: apresentação de uma abordagem integradora*. Em *AEDEM International Conference*, volume 18, páginas 427–442, 2017. 9, 10, 13, 14
- [11] Caneppele, Nairana Radtke, Helena Belintani Shigaki, Heidy Rodriguez Ramos e Ivano Ribeiro: *A utilização do software vosviewer em pesquisas científicas*. *Revista Ibero-Americana de Estratégia*, 22(1):e24970–e24970, 2023. 12

- [12] Grotto, Silvio Fratti: *A utilização do mapeamento da cadeia de valor como ferramenta para identificar o alinhamento da área de tecnologia da informação com uma área de negócio em uma instituição de ensino superior*. 2012. 24
- [13] Cruz, Diogenes Marco de Brito, Lucas Magno Neris, Lucas Gomes Vilas Boas e Jeane Denise de Souza Menezes: *Aplicação do planejamento estratégico a partir da análise swot: Um estudo numa empresa de tecnologia da informação*. Anais do IX SIMPROD, 2017. 26, 32, 82
- [14] Moraes, Guilherme Mendonça de: *Análise e melhoria de processo de reengenharia de software: proposta de instrumento para gerenciamento de riscos do processo*. 2021. 27
- [15] Villela, Cristiane da Silva Santos *et al.*: *Mapeamento de processos como ferramenta de reestruturação e aprendizado organizacional*. 2000. 27, 29
- [16] Santos, Thiago de Jesus dos: *Gestão de riscos e a norma iso 31000: uma abordagem literária*. Management Journal, 3(1):1–14, 2021. 29, 36, 37
- [17] Hutchins, Gregory: *ISO 31000: 2018 enterprise risk management*. Greg Hutchins, 2018. 30
- [18] Hofrichter, Markus: *Análise SWOT: Quando usar e como fazer*. Simplíssimo, 2017. 32
- [19] Mello, Marcos Sanmartin de, João Souza Neto e Tomás Roberto Cotta Orlandi: *Risk management in software engineering projects employing agile scrum methodology gerenciamento de riscos em projetos de engenharia de software que empregam a metodologia ágil scrum*. 33, 34, 131
- [20] Medeiros, Juliana Dantas Ribeiro Viana de, Daniela CP Alves, Alexandre Marcos Lins de Vasconcelos, Carla Taciana Lima Lourenço Silva Schuenemann e Eduardo Wanderley: *Engenharia de requisitos em projetos ágeis: uma revisão sistemática da literatura*. 37, 41
- [21] Leffingwell, Dean: *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, Boston, 2011. 40
- [22] Cohn, Mike: *Agile Estimating and Planning*. Prentice Hall, Upper Saddle River, NJ, 2004. Explora práticas colaborativas, foco em valor e abordagem iterativa no planejamento ágil. 40, 152
- [23] Oliveira, Leonardo Rocha de, Geraldo Santoro Gomes e Fernando Policarpo de Lima: *Análise de riscos pelo uso de métodos ágeis na gestão de projetos de desenvolvimento de software*. Revista de Gestão e Projetos, 5(2):90–101, 2014. 40
- [24] Canedo, Edna Dias, Viviane Vasconcellos e Simone Borges Simão Monteiro: *Práticas de engenharia de requisitos para conformidade legal em ambientes Ágeis*. Em *Anais do Congresso Brasileiro de Software: Teoria e Prática (CBSoft)*. SBC, 2022. Available at: <https://sol.sbc.org.br/index.php/cbsoft/article/view/20400>. 40, 70, 73, 74

- [25] Canedo, Edna Dias, Angelica Toffano Seidel Calazans, Ian Nery Bandeira, Pedro Henrique Teixeira Costa e Eloisa Toffano Seidel Masson: *Guidelines adopted by agile teams in privacy requirements elicitation after the brazilian general data protection law (lgpd) implementation*. Requirements Engineering, 27(4):545–567, 2022. 40, 41, 42, 50
- [26] Alves, Carina e Moisés Neves: *Especificação de requisitos de privacidade em conformidade com a lgpd: Resultados de um estudo de caso*. Em *WER*, 2021. 40, 78
- [27] Sommerville, Ian: *Engenharia de Software*. Pearson Prentice Hall, São Paulo, 9ª edição, 2011, ISBN 9788576058634. 40, 50, 53, 56, 57, 59, 60, 62, 64, 66, 69, 71, 73, 80, 95, 100, 153
- [28] DA CUNHA, RAYANA GONÇALVES e ANDREA RAYMUNDO BALLE: *Comparação de indicadores de projetos com uso de metodologia ágil ou híbrida*. 41
- [29] Calazans, Angelica Toffano Seidel, Roberto Ávila Paldês, Edna Dias Canedo, Eloisa Toffano Seidel Masson, Fernando de Albuquerque Guimarães, Kiane Mabel Fialho Rezende, Fábio de Souza Gonçalves e Ari Melo Mariano: *Quality requirements and the requirements quality: The indications from requirements smells in a financial institution systems*. Em *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, páginas 472–480, 2019. 41, 42
- [30] Oliveira, Cléber Ivo de: *Reengenharia e engenharia reversa de software*. 42
- [31] Page, Cersei: *Software engineering*. Larsen and Keller Education, 2017. 43
- [32] Silva, Dyego Alves da, Edna Dias Canedo e Edgard Costa de Oliveira: *Proposta para análise de riscos no processo de planejamento da contratação de ti: um estudo exploratório para órgãos governamentais*. *iSys-Brazilian Journal of Information Systems*, 9(1):168–186, 2016. 43
- [33] Pressman, Roger S e Bruce R Maxim: *Engenharia de software-9*. McGraw Hill Brasil, 2021. 43, 46, 48, 61, 72
- [34] Canedo, Edna Dias, Angelica Toffano Seidel Calazans, Anderson Jefferson Cerqueira, Pedro Henrique Teixeira Costa e Eloisa Toffano Seidel Masson: *Agile teams’ perception in privacy requirements elicitation: Lgpd’s compliance in brazil*. Em *2021 IEEE 29th International Requirements Engineering Conference (RE)*, páginas 58–69. IEEE, 2021. 44, 80
- [35] Denzin, Norman K e Yvonna S Lincoln: *The Sage handbook of qualitative research*. sage, 2011. 45, 113
- [36] Silva Souza, Lucas da: *Processos da Engenharia de Requisitos no Contexto de Internet das Coisas (IoT) e Técnicas de Validação de Requisitos*. Tese de Doutorado, Universidade de Brasília, 2021. 45
- [37] Sommerville, Ian: *Engineering software products*, volume 355. Pearson London, UK, 2020. 46, 58

- [38] Semedo, Maria João Moreno *et al.*: *Ganhos de produtividade e de sucesso de metodologias ágeis vs metodologias em cascata no desenvolvimento de projectos de software*. 2012. 47, 49, 77, 79
- [39] Engel, Guido Irineu: *Pesquisa-ação*. *Educar em Revista*, páginas 181–191, 2000. 48, 82, 114
- [40] Bourque, Pierre e Richard E. Fairley (editores): *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. IEEE Computer Society, Los Alamitos, CA, version 3.0 edição, 2014. <https://www.computer.org/education/bodies-of-knowledge/software-engineering>. 48, 84
- [41] Oliveira, João, Mariana Silva e Pedro Costa: *Uma análise sobre metodologias ágeis: Xp, scrum e kanban na prática de desenvolvimento de software*. Em *Anais do Congresso Brasileiro de Engenharia de Software (CBSoft)*, São Paulo, Brasil, 2020. SBC. Exemplo representativo; substituir conforme fonte exata. 49, 84
- [42] Deepa, T *et al.*: *Exploring the ideology of agile framework in a versalite field through state of agile report – a survey*. *NeuroQuantology*, 20(7):1, 2022. 49
- [43] Canedo, Edna Dias, Ian Nery Bandeira, Larissa Pereira Gonçalves, Alessandra de Vasconcelos Sales, Fábio Mendonça, Cláudio Azevedo Costa e Rafael T de Sousa Jr: *Business process modeling supporting the requirements elicitation of an audit system: An experience report*. *International Journal of Electronic Government Research (IJEGR)*, 19(1):1–20, 2023. 50, 80
- [44] Medeiros, Juliana Dantas Ribeiro Viana de, Daniela CP Alves, Alexandre Marcos Lins de Vasconcelos, Carla Taciana Lima Lourenço Silva Schuenemann e Eduardo Wanderley: *Engenharia de requisitos em projetos ágeis: uma revisão sistemática da literatura*. 2015. 50
- [45] International Organization for Standardization, International Electrotechnical Commission e Institute of Electrical and Electronics Engineers: *Iso/iec/ieee 29148:2018 - systems and software engineering – life cycle processes – requirements engineering*, 2018. <https://www.iso.org/standard/72089.html>. 50, 58, 65, 73, 153
- [46] International Organization for Standardization e International Electrotechnical Commission: *Iso/iec 25010:2011 - systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models*, 2011. <https://www.iso.org/standard/35733.html>. 53, 54, 58, 59, 68, 69, 71
- [47] Silva, Dyego Alves da, Edna Dias Canedo, Edgard Costa de Oliveira e Simone Borges Simão Monteiro: *Roteiro para aferição de requisitos não funcionais de software*. Em *Anais do 13º Congresso Internacional de Gestão de Tecnologia e Sistemas de Informação (CONTECSI)*, páginas 1–10, São Paulo, Brasil, 2016. TECSI - FEA/USP. <https://www.tecsi.org/contecsi/index.php/contecsi/13CONTECSI/schedConf/presentations>. 53, 55, 56, 58

- [48] Almeida Araújo, Marcelo Ferreira de, Angélica Toffano Seidel Calazans, Roberto Ávila Paldês, Kiane Fialho Rezende, Fábio de Souza Gonçalves, Fernando de Albuquerque Guimarães, Eloisa Toffano Seidel Calazans, Emeli Braosi e Edna Dias Canedo: *Riscos de engenharia de requisitos com a utilização de metodologias ágeis*. 58, 134
- [49] Mendonça, Ricardo Augusto Ribeiro de: *Levantamento de requisitos no desenvolvimento ágil de software*. Semana da Ciência e Tecnologia da PUC Goiás, página 12, 2014. 61, 151
- [50] Canedo, Edna Dias, Simone Borges Simão Monteiro e Viviane Vasconcellos: *Engenharia de requisitos: Uma abordagem para sistemas críticos*. Em *Anais do Congresso Brasileiro de Software*, 2022. 62, 64, 65, 72, 79
- [51] Monteiro, Simone Borges Simão e Edna Dias Canedo: *Requisitos operacionais em sistemas governamentais: desafios e estratégias de mitigação*. Revista de Engenharia de Software Aplicada, 9(1):33–49, 2021. 63
- [52] Wiegers, Karl e Joy Beatty: *Software Requirements*. Microsoft Press, Redmond, WA, 3rd edição, 2013, ISBN 978-0735679665. 64, 65
- [53] *Lei geral de proteção de dados pessoais (lcpd) – lei nº 13.709, de 14 de agosto de 2018*. Presidência da República Federativa do Brasil, 2018. https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm. 73
- [54] *Sarbanes-oxley act of 2002*. Public Law 107-204, U.S. Congress, 2002. <https://www.congress.gov/bill/107th-congress/house-bill/3763>. 73
- [55] Nunes, Rodrigo Dantas: *A implantação das metodologias ágeis de desenvolvimento de software scrum e extreme programming (xp): uma alternativa para pequenas empresas do setor de tecnologia da informação*. ForScience, 4(2), 2016. 76, 98
- [56] Pfleeger, Shari Lawrence e Joanne M. Atlee: *Software Engineering: Theory and Practice*. Prentice Hall, Upper Saddle River, NJ, 4th edição, 2012, ISBN 9780136061694. 76
- [57] Alves, Priscilla Souza Ramos, Simone Borges Simão Monteiro e Viviane Vasconcellos Ferreira Grubisic: *Gestão de riscos e fatores de sucesso em projetos de ti em uma instituição pública: teoria e prática*. Revista de Gestão e Secretariado, 14(5):7626–7646, 2023. 77
- [58] Lima, Rogério Gabriel Nogalha de, Simone Borges Simão Monteiro, Viviane Vasconcellos Ferreira Grubisic e Ari Melo Mariano: *Análise de riscos do processo de execução financeira na área de ti e propostas de melhoria: aplicação de fmea e fta*. Revista de Gestão e Secretariado, 14(5):7082–7103, 2023. 77
- [59] Pressman, Roger S. e Bruce R. Maxim: *Software Engineering: A Practitioner’s Approach*. McGraw-Hill Education, 8ª edição, 2016, ISBN 9780078022128. 78, 90, 91, 97

- [60] Farias, Eduardo, Lucas Moreira e Tatiane Gomes: *A sabedoria coletiva da equipe como fator de sucesso na engenharia de requisitos*. Journal of Software Engineering Practices, 11(2):45–61, 2023. 78
- [61] Canedo, Edna Dias e Simone Borges Simão Monteiro: *Governança de Tecnologia da Informação: Modelagem de Processos de Negócio Automatizados e a Automatizar*. Editora Universitária, 2023. 78, 80
- [62] Canedo, Edna Dias, Vítor Paes, Simone Borges e Tiago Monteiro: *Elicitação de requisitos de privacidade em tempos de lgpd: Uma investigação em equipes Ágeis brasileiras*. Em *Anais do XXI Simpósio Brasileiro de Qualidade de Software (SBQS)*, páginas 541–550. SBC, 2022. <https://sol.sbc.org.br/index.php/sbqs/article/view/222146>. 78, 81, 83
- [63] Fowler, Martin: *Patterns of Enterprise Application Architecture*. Addison-Wesley, Boston, MA, 2003. Discussão sobre metodologias de engenharia no desenvolvimento de software. 84
- [64] Monteiro, Ana Paula, João Carlos Silva e Fernanda Andrade: *A importância da modelagem de processos de negócio na elicitação de requisitos de software*. Revista Brasileira de Engenharia de Software, 15(2):45–60, 2023. Exemplo representativo; adaptar conforme a fonte original. 84, 85
- [65] Grubišić, Ivica: *The contribution of business process modeling to risk management*. International Journal of Computers, 4(3):152–159, 2010. <https://www.naun.org/main/NAUN/computers/19-516.pdf>. 85
- [66] Canedo, Edna Dias e colaboradores: *Boas práticas de modelagem de processos de negócio com bpmn*. Em *Anais do Workshop de Engenharia de Requisitos*, 2018. 85
- [67] Canedo, Edna Dias, Talita Ribeiro Silva, Vanessa C. Oliveira e José Renato Sousa: *Identificação de requisitos de privacidade com apoio da modelagem de processos de negócio*. Em *Anais do XVII Simpósio Brasileiro de Sistemas de Informação (SBSI)*, páginas 1–8, Niterói, RJ, 2022. SBC. <https://sol.sbc.org.br/index.php/sbsi/article/view/21779>, Discussão sobre LGPD, modelagem e requisitos de privacidade. 85
- [68] Mendonça, Nome Completo: *Remo: Extração de requisitos orientada à modelagem de processos de negócio*. Revista Brasileira de Engenharia de Requisitos, 10(2):123–140, 2017. Entrada exemplificativa para a técnica REMO, conforme discutido no texto. 86
- [69] *Iso/iec/ieee 29148:2018 — systems and software engineering — life cycle processes — requirements engineering*, 2018. <https://www.iso.org/standard/72089.html>, Norma internacional sobre engenharia de requisitos: elaboração, análise, rastreabilidade e gestão. 86
- [70] Mota, José Augusto: *Engenharia de Requisitos Aplicada à Modelagem de Processos de Negócio*, páginas 145–168. Editora Ciência Moderna, Rio de Janeiro, Brasil, 2022.

Discussão sobre o uso de heurísticas na elicitação de requisitos a partir de processos organizacionais. 87

- [71] Mota, Isaac Borges: *Engenharia de requisitos apoiada pela modelagem de processos de negócio: Estudo de caso do projeto map*, 2022. https://bdm.unb.br/bitstream/10483/33942/1/2022_IsaacBorgesMota.pdf. 89
- [72] Royce, Winston W.: *Managing the development of large software systems*. Em *Proceedings of IEEE WESCON*, páginas 1–9, Los Angeles, CA, 1970. IEEE. <https://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>, Primeira descrição do modelo waterfall no contexto de engenharia de software. 90
- [73] Boehm, Barry W.: *A spiral model of software development and enhancement*. ACM SIGSOFT Software Engineering Notes, 11(4):14–24, 1986. <https://doi.org/10.1145/12944.12948>, Artigo original que propõe o modelo espiral de desenvolvimento de software. 90
- [74] Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas: *Manifesto for agile software development*. <https://agilemanifesto.org/>, 2001. Acordo original entre os 17 signatários do Manifesto Ágil. 94, 99
- [75] Pressman, Roger S: *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005. 95
- [76] Caroli, Paulo: *Direto ao ponto: criando produtos de forma enxuta*. Editora Casa do Código, 2015. 95, 151
- [77] Ponniah, Paulraj: *Data Warehousing: Fundamentals for IT Professionals*. Wiley, Hoboken, NJ, 2014. Conceitua atividades upstream como parte do alinhamento inicial de projetos de TI. 96
- [78] Leopold, Klaus: *Practical Kanban: From Team Focus to Creating Value*. Leanability Press, Vienna, Austria, 2017. Discute atividades upstream como compreensão de problemas, descoberta de valor e estruturação inicial de backlog. 97
- [79] Cao, Lan e Balasubramaniam Ramesh: *Agile requirements engineering practices: An empirical study*. IEEE software, 25(1):60–67, 2008. 98
- [80] Massari, Vitor: *Agile Scrum Mster no Gerenciamento Avançado de Projetos*. Brasport, 2016. 103
- [81] Fonseca, Fernando: *Scrum e Agile em Projetos: Gestão Ágil de Projetos de Software com Scrum*. Novatec Editora, São Paulo, Brasil, 2009. Explica o fluxo de processo Scrum, papéis, eventos e responsabilidades de cada perfil. 105
- [82] NICHI, LUCIANA, BRUNO D'LUCCA SILVA CARVALHO e MARCELLA CANEDO TRISTÃO: *Aplicação das técnicas de lean inception e mvp no processo de uma fábrica de software acadêmica*. 2021. 106, 107

- [83] Gomes Filho, AVELINO FERREIRA: *Modelo de ensino baseado nos métodos ágeis de desenvolvimento de software*. Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 40, 2016. 108, 109
- [84] Oliveira Junior, Antônio Carvalho de e Hugo Ferreira Martins: *Uso de design thinking na elicitação de requisitos em projeto ágil de software*. 2018. 109
- [85] Souza, Cynara Lira de Carvalho: *Uso do design thinking na elicitação de requisitos de ambientes virtuais de aprendizagem móvel*. Tese de Mestrado, Universidade Federal de Pernambuco, 2014. 109
- [86] GUEDES, MELANY PRISCILA e FRANCIANE PEREIRA REIS: *Atuação do escritório de gerenciamento de projetos na implantação da metodologia de gp na fundação parque tecnológico itaipu*. 2015. 111
- [87] Rodrigues, William Costa *et al.*: *Metodologia científica*. Faetec/IST. Paracambi, páginas 2–20, 2007. 112
- [88] Silva, Gisele Cristina Resende Fernandes da: *O método científico na psicologia: abordagem qualitativa e quantitativa*, 2010. 112
- [89] Gerhardt, Tatiana Engel e Denise Tolfo Silveira: *Métodos de pesquisa*. Plageder, 2009. 114
- [90] autor sem: *A utilização do mapeamento da cadeia de valor como ferramenta para identificar o alinhamento da área de tecnologia da informação com uma área de negócio em uma instituição de ensino superior*. 2023. 124
- [91] Buchele, Gustavo Tomaz, Pierry Teza, João Artur de Souza e Gertrudes Aparecida Dandolini: *Métodos, técnicas e ferramentas para inovação: o uso do brainstorming no processo de design contribuindo para a inovação*. Pensamento & Realidade, 32(1):61–61, 2017. 136
- [92] DE ALMEIDA, PAULA SOARES: *Gestão de riscos*. 2016. 138
- [93] Canedo, Edna Dias e Geise Kelly Silva: *Análise da percepção de stakeholders no processo de engenharia de requisitos*. Em *Anais do Congresso Brasileiro de Software*. SBC, 2017. 142

Apêndice A

Pesquisa

A.1 TEMAC

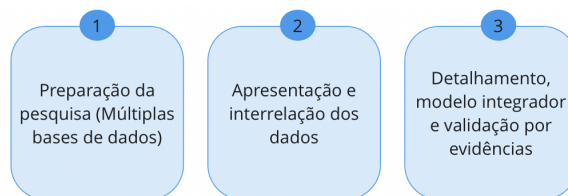


Figura A.1: Etapas - TEMAC
Fonte: Elaborada pela autora (2025)

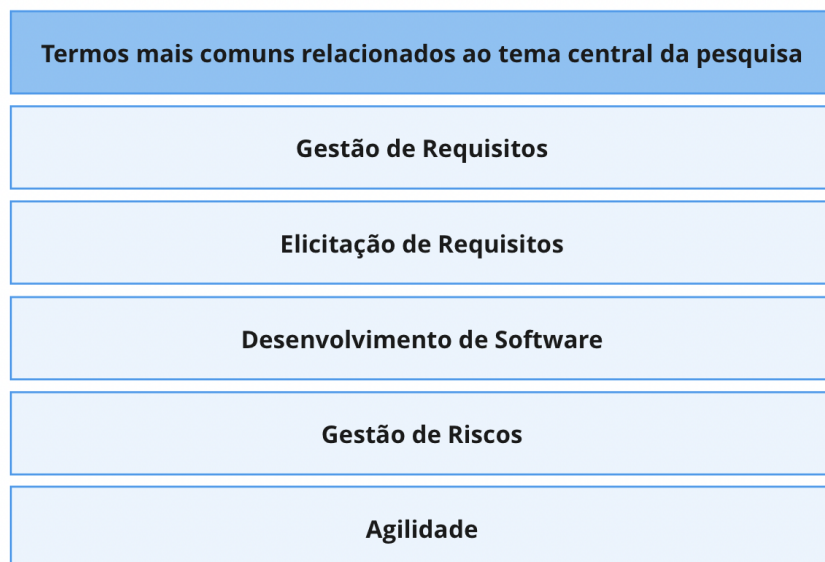


Figura A.2: Termos mais comuns
Fonte: Elaborada pela autora (2025)

Palavra - chave	Buscadas no Título	WoS	Scopus
1 - Requirements elicitation <i>and</i> Risk management	Requirements, Risk management	76	134
2 - Requirements elicitation <i>and</i> Risk management <i>and</i> . Software development	Requirements, Software development	12	32
3 - Requirements elicitation <i>and</i> Risk management <i>and</i> . Agile	Requirements Elicitation, Agile	6	108
4 - Requirements elicitation <i>and</i> Software development <i>and</i> Agile	Requirements Elicitation, Software	67	118
5 - Risk management <i>and</i> Agile	Risk, Agile	159	198
Total		320	590

Figura A.3: Combinação dos termos de pesquisa utilizados no TEMAC

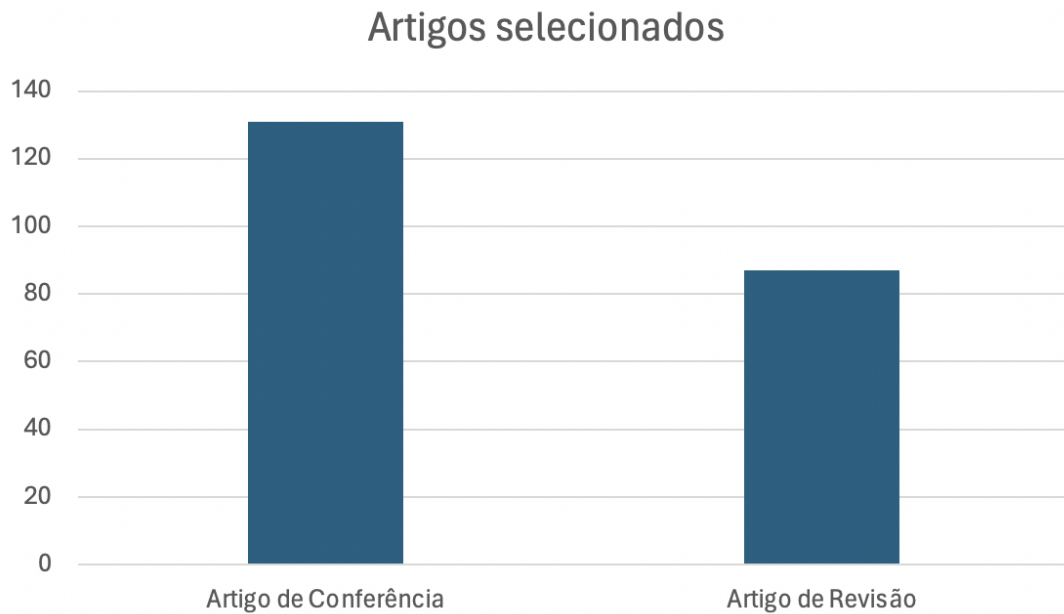
Fonte: Elaborada pela autora (2025)

Figura A.4: Áreas de Pesquisa



Fonte: Elaborada pela autora (2025)

Figura A.5: Tipo de Documento



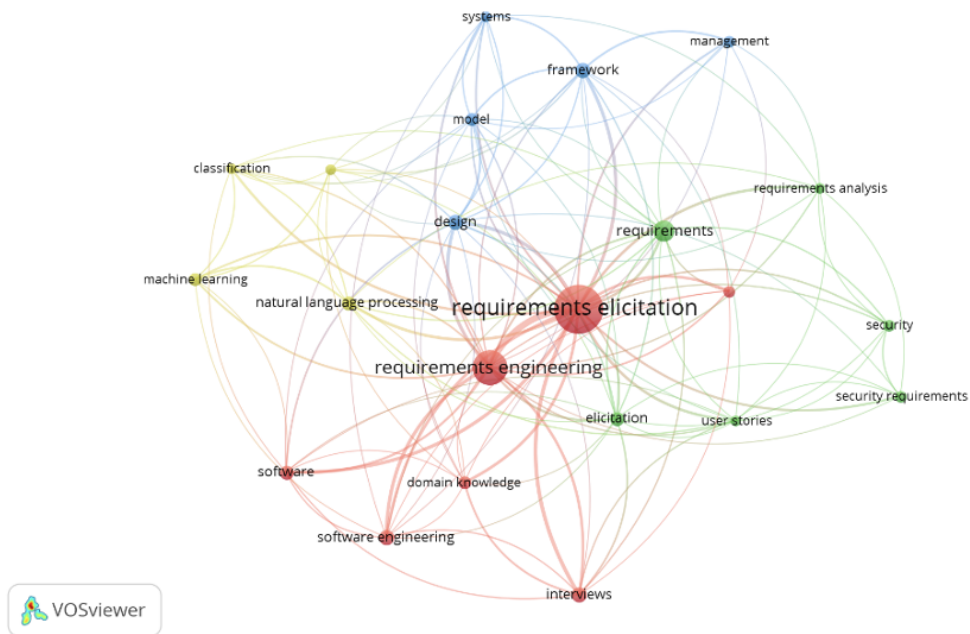
Fonte: Elaborada pela autora (2025)

Figura A.6: Foco de pesquisa das principais áreas

Área de Pesquisa	Principais Temas
Computer Science Software Engineering	<ul style="list-style-type: none">• Engenharia de Software• Desenvolvimento de Software• Engenharia de Requisitos• Gestão de Riscos• Métodos Ágeis
Computer Science Information Systems	<ul style="list-style-type: none">• Engenharia de Software• Gestão de Projetos• Desenvolvimento de Software• Gestão de Riscos

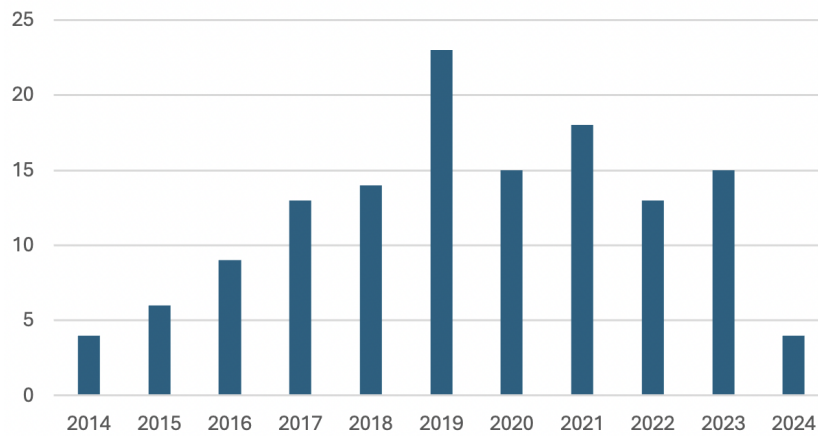
Fonte: Elaborada pela autora (2025)

Figura A.7: Associação de palavras-chave



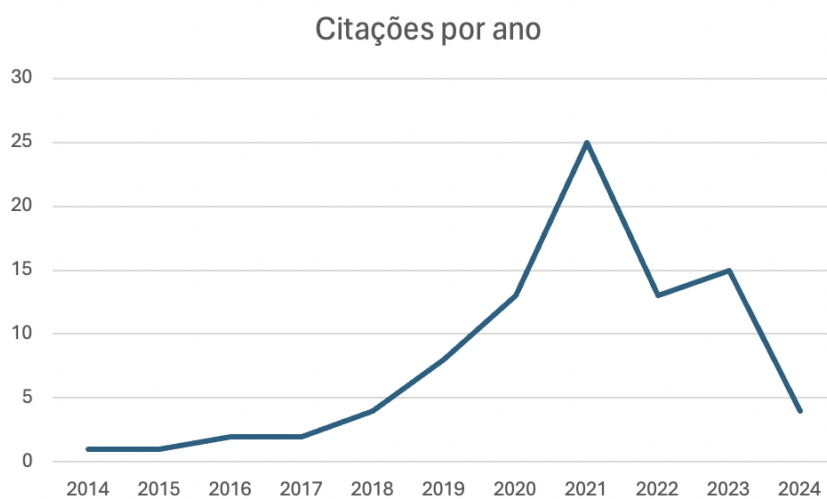
Fonte: Elaborada pela autora (2025)

Figura A.8: Quantidade de publicações por ano
Quantidade de Publicações por ano



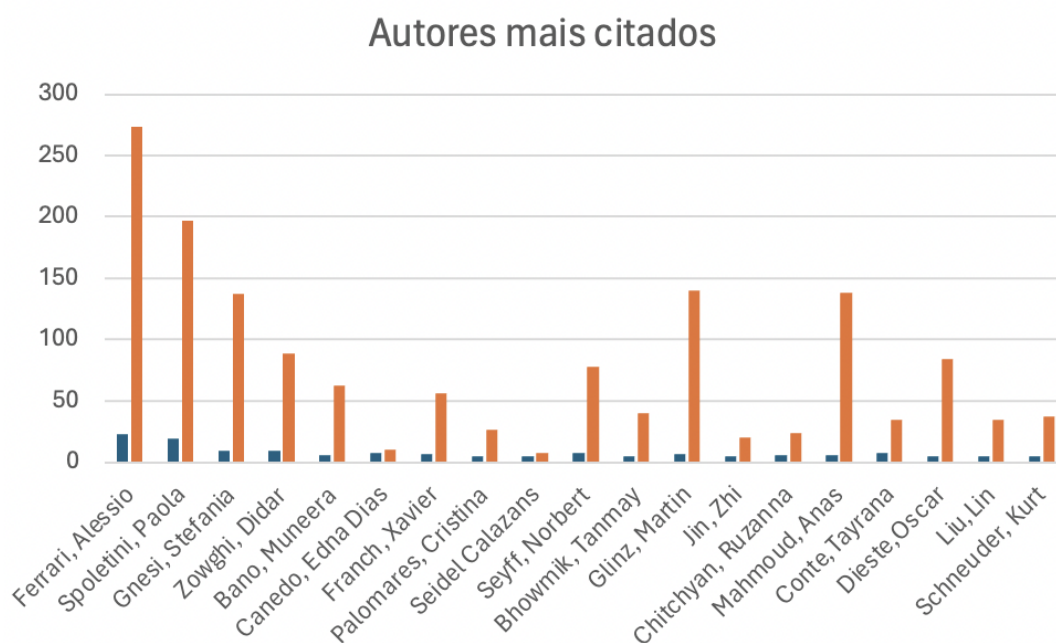
Fonte: Elaborada pela autora (2025)

Figura A.9: Quantidade de citações por ano



Fonte: Elaborada pela autora (2025)

Figura A.10: Autores dos artigos mais citados



Fonte: Elaborada pela autora (2025)

Figura A.11: Artigos mais citados

TÍTULO	AUTORES	ANO PUBLICAÇÃO	TOTAL DE CITAÇÕES	MÉDIA POR ANO
Natural Language Processing (NLP) for Requirements Engineering (RE): A Systematic Mapping Study	Liping Zhao, Waad Alhoshan, Alessio Ferrari, Keletso J. Letsholo	2019	210	52
SaPeer and ReverseSaPeer: teaching requirements elicitation interviews with role-playing and role reversal	Alessio Ferrari, Paola Spoletini, Muneera Bano e Didar Zowghi	2019	27	7
A Refinement Calculus for Requirements Engineering Based on Argumentation Theory	Yehia Elrakaiby, Alexander Borgida, Alessio Ferrari e John Mylopoulos	2020	12	4
Interview Review: An Empirical Study on Detecting Ambiguities in Requirements Elicitation Interviews	Paola Spoletini, Muneera Bano, Didar Zowghi e Stefania Gnesi	2018	18	3
Using Argumentation to Explain Ambiguity in Requirements Elicitation Interviews	Yehia Elrakaiby, Alessio Ferrari, Stefania Gnesi e Paola Spoletini	2017	27	3
Refinement of Behavioural Models for Variability Description	Alessandro Fantechi e Stefania Gnes	2016	3	1
"Human Value Requirements in AI Systems: Empirical Analysis of Amazon Alexa"	Didar Zowghi, Rifat Ara Shams, Muneera Bano, Qinghua Lu e Jon Whittle	2023	2	1
"Large Language Models for Qualitative Research in Software Engineering: Exploring Opportunities and Challenges"	Rashina Hoda, Didar Zowghi e Christoph Treude	2023	6	3
Design Thinking: Challenges for Software Requirements Elicitation	Ferreira Martins, Hugo; Carvalho de Oliveira Junior, Antônio; Dias Canedo, Edna	2020	61	16
Riscos de Engenharia de Requisitos com a utilização de Metodologias Ágeis	Marcelo Ferreira de Almeida Araújo, Angélica Toffano Seidel Calazans, Edna Dias Canedo	2019	1	1
Autenticidade na aprendizagem baseada em projetos para desenvolvimento de software	Vinicius Gomes Ferreira e Edna Dias Canedo	2019	5	1

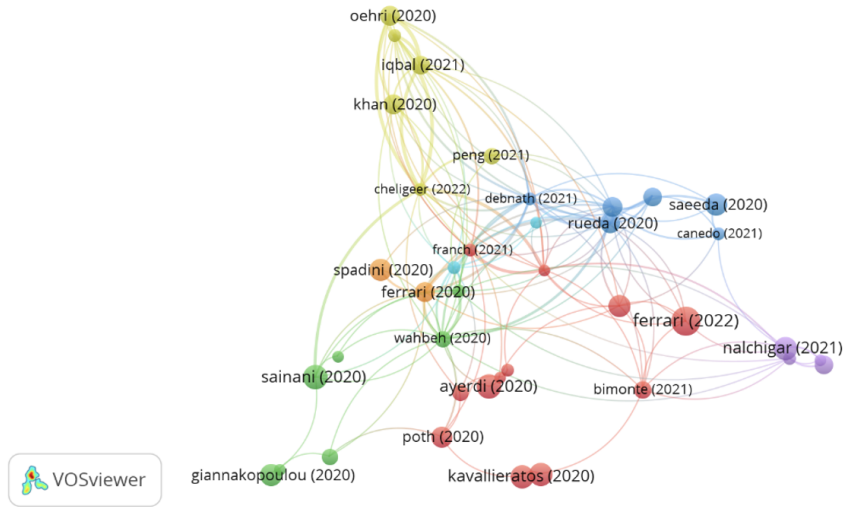
Fonte: Elaborada pela autora (2025)

Figura A.12: Países com maior número de citações



Fonte: Elaborada pela autora (2025)

Figura A.15: Rede de Coupling



Fonte:

Web Of Science

A.2 Formulário

Figura A.16: Parte I

Pesquisa - Liberty Tecnologia - Elicitação de Requisitos - Principais Riscos

B *I* U ↻ ✕

Olá!

A Liberty Tecnologia está conduzindo um estudo sobre os principais riscos encontrados na empresa. O objetivo é, com sua valiosa colaboração, priorizar e mitigar esses riscos para aprimorar continuamente nossos processos, especialmente durante a fase crucial de Elicitação de Requisitos no desenvolvimento de soluções para nossos clientes.

Instruções: Avalie cada afirmação a seguir com base em seu nível de concordância. Utilize a seguinte escala, onde 1 significa "Discordo Totalmente" e 5 significa "Concordo Totalmente".

Sua avaliação é extremamente importante para nós. Contamos com a sua participação ativa para fazer a diferença!

Fonte: Elaborada pela autora (2025)

Figura A.17: Parte II

R01. Entendo que as definições de negócios, escopo e objetivos do produto são imprecisas. *

- 1 - Discordo totalmente
- 2 - Discordo parcialmente
- 3 - Neutro
- 4 - Concordo parcialmente
- 5 - Concordo totalmente

Fonte: Elaborada pela autora (2025)

Figura A.18: Parte III

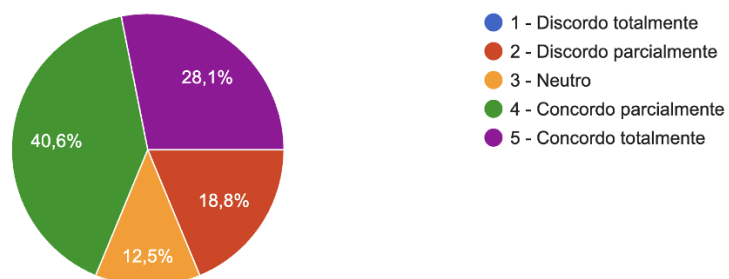
R02. Há desentendimentos frequentes sobre os requisitos do produto. *

- 1 - Discordo totalmente
- 2 - Discordo parcialmente
- 3 - Neutro
- 4 - Concordo parcialmente
- 5 - Concordo totalmente

Fonte: Elaborada pela autora (2025)

R01. Entendo que as definições de negócios, escopo e objetivos do projeto são imprecisas.

32 respostas



R02. Há desentendimentos frequentes sobre os requisitos do projeto.

32 respostas

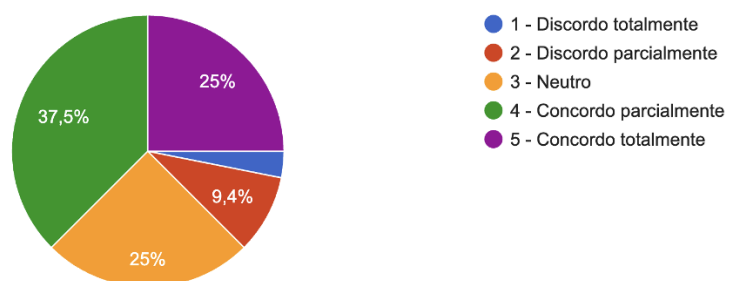
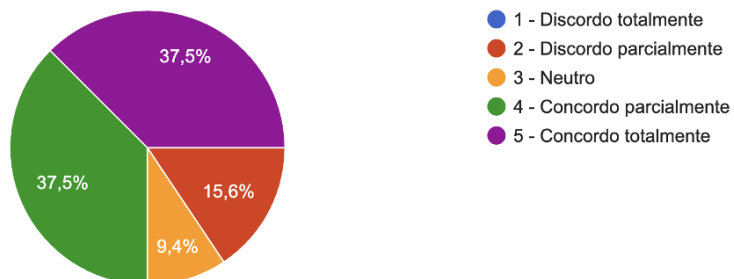


Figura A.19: Resultado Formulário
Fonte: Elaborada pela autora (2025)

Figura A.20: Resultado Formulário

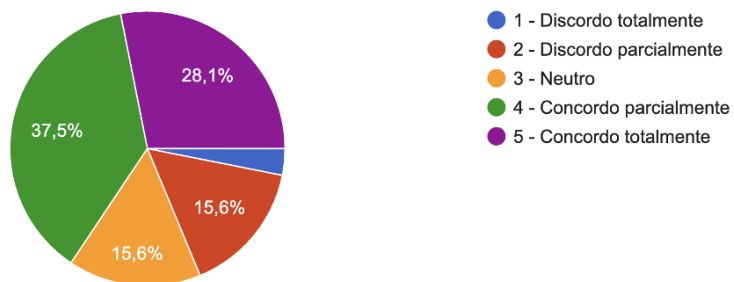
R04. Ocorrem alterações frequentes nos requisitos.

32 respostas



R05. A gestão de mudanças dos requisitos é ineficaz.

32 respostas

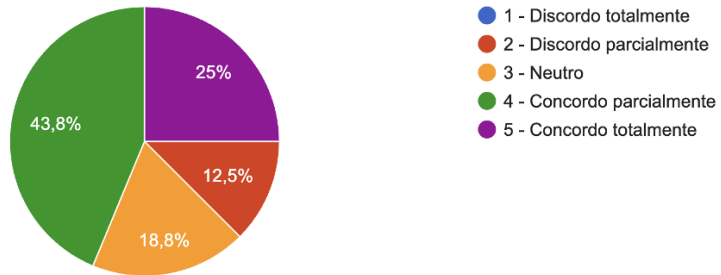


Fonte: Elaborada pela autora (2025)

Figura A.21: Resultado Formulário

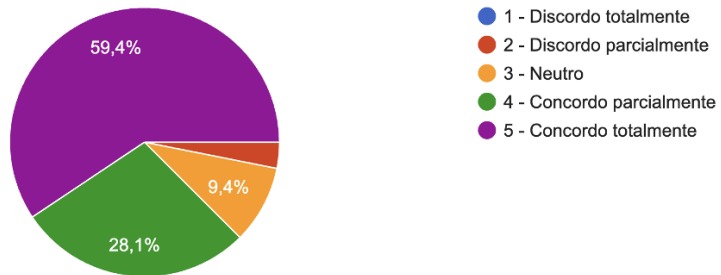
R08. Os riscos não são gerenciados adequadamente.

32 respostas



R09. Há uma ausência de processo estruturado na fase de elicitação de requisitos.

32 respostas

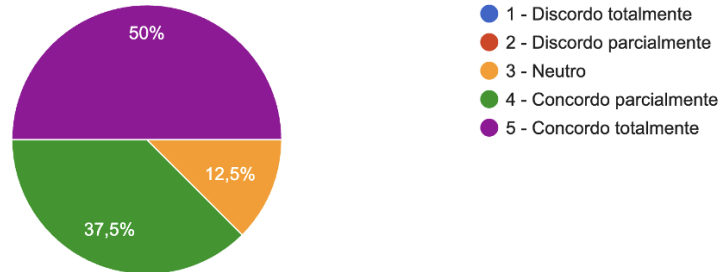


Fonte: Elaborada pela autora (2025)

Figura A.22: Resultado Formulário

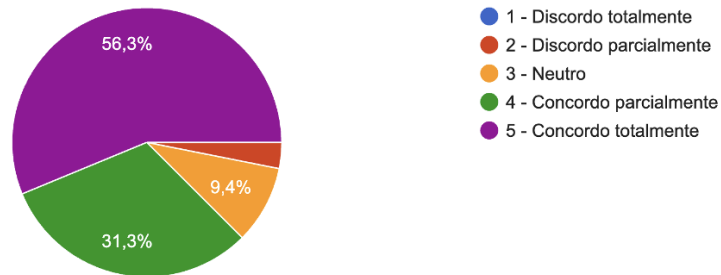
R10. Inexistência de ferramentas adequadas.

32 respostas



R11. Metodologia ágil não definida.

32 respostas



Fonte: Elaborada pela autora (2025)

A.3 Levantamento dos Requisitos da Proposta de Solução

O UPLean será uma plataforma colaborativa para construção de produto que seguirá metodologias ágeis. O sistema será organizado em 5 fases principais:

• Fase 1: Alinhamento e Contexto • Fase 2: Exploração de Funcionalidades • Fase 3: Definição do Backlog do Produto • Fase 4: Definição do MVP • Fase 5: Detalhamento e Encerramento

Tipos de Usuário: • Administrador: Pode criar iniciativas, gerenciar usuários e participar de todas as fases • Stakeholder: Participa focando na visão de negócio e requisitos • Técnico: Participa focando em aspectos técnicos e implementação

Seguem abaixo os épicos, features e histórias de usuários da proposta de solução:

Épico 01: Gestão de Autenticação e Usuários:

Descrição: Permitir que usuários se cadastrem, autentiquem e sejam gerenciados no sistema com diferentes níveis de acesso (Admin e Participante). Valor de Negócio: Garantir segurança e controle de acesso à plataforma, diferenciando permissões entre administradores e participantes.

Épico 02: Gestão de Iniciativas:

Descrição: Permitir a criação, visualização, edição e exclusão de iniciativas de produto, bem como o gerenciamento de participantes em cada iniciativa. Valor de Negócio: Organizar e estruturar os projetos de descoberta de produto, facilitando a colaboração entre equipes.

Épico 03: Fase 1 - Alinhamento e Contexto:

Descrição: Coletar e documentar informações fundamentais sobre o produto através de visão, definição, metas, personas e jornadas de usuário. Valor de Negócio: Estabelecer uma base sólida de entendimento sobre o produto, seu público-alvo e objetivos estratégicos.

Épico 04: Fase 2 - Exploração de Funcionalidades:

Descrição: Identificar, listar e avaliar funcionalidades potenciais do produto através de brainstorming e avaliação 360°. Valor de Negócio: Mapear todas as possibilidades de funcionalidades antes de priorizar, garantindo visão holística do produto.

Épico 05: Fase 3 - Definição do Backlog:

Descrição: Estruturar o backlog do produto organizando features em épicos e detalhando user stories. Valor de Negócio: Criar uma estrutura de trabalho clara e organizada para o desenvolvimento do produto.

Épico 06: Fase 4 - Definição do MVP:

Descrição: Definir o contexto do MVP e priorizar as funcionalidades essenciais para a primeira versão do produto. Valor de Negócio: Focar esforços nas funcionalidades mais importantes, acelerando o time-to-market.

Épico 07: Fase 5 - Detalhamento e Fechamento:

Descrição: Consolidar toda a documentação gerada nas fases anteriores e permitir exportação em formato profissional. Valor de Negócio: Facilitar a comunicação dos resultados do discovery para stakeholders e equipes de desenvolvimento.

Épico 08: Dashboards e Navegação:

Descrição: Fornecer interfaces intuitivas e específicas para cada tipo de usuário visualizar e acessar suas iniciativas. Valor de Negócio: Melhorar a experiência do usuário e facilitar o acesso rápido às informações relevantes.

Features:

Épico 01: Gestão de Autenticação e Usuários:

- FT01.01 - Cadastro de Usuário: Permitir que novos usuários se cadastrem na plataforma informando CPF, nome completo e tipo de usuário.

- FT01.02 - Login e Autenticação: Permitir que usuários autenticados acessem o sistema de forma segura.
- FT01.03 - Aprovação de Usuários: Permitir que administradores aprovem ou rejeitem novos cadastros de usuários.
- FT01.04 - Gerenciamento de Perfis: Permitir que administradores visualizem e gerenciem perfis de usuários cadastrados.
- FT01.05 - Recuperação de Acesso: Permitir recuperação de acesso por e-mail/CPF com código temporário.
- FT01.06 - Auditoria de Acessos: Registrar logins, logouts e tentativas falhas para auditoria.

Épico 02: Gestão de Iniciativas:

- FT02.01 - Criação de Iniciativa: Permitir que administradores criem novas iniciativas com título, descrição e status.
- FT02.02 - Listagem de Iniciativas: Exibir todas as iniciativas disponíveis com filtros e busca.
- FT02.03 - Visualização de Detalhes da Iniciativa: Permitir visualizar informações detalhadas de uma iniciativa, incluindo fases e participantes.
- FT02.04 - Convite de Participantes: Permitir que administradores convidem usuários para participar de iniciativas.
- FT02.05 - Gerenciamento de Status: Permitir alteração do status da iniciativa (planejamento, em progresso, concluída, arquivada).
- FT02.06 - Papéis na Iniciativa: Definir papéis (Product Owner, Facilitador, Convidado) para participantes.
- FT02.07 - Anexos da Iniciativa: Upload e gestão de arquivos relacionados à iniciativa.

Épico 03: Fase 1 - Alinhamento e Contexto:

- FT03.01 - Definição da Visão do Produto: Capturar a visão do produto usando template estruturado.

- FT03.02 - Definição do Produto (Is/Is Not, Does/Does Not): Documentar o que o produto é e não é, faz e não faz.
- FT03.03 - Definição de Metas e Objetivos: Registrar metas de negócio e objetivos mensuráveis do produto.
- FT03.04 - Criação de Personas: Documentar personas com nome, idade, profissão, objetivos, desafios e comportamentos.
- FT03.05 - Mapeamento de Jornadas do Usuário: Mapear jornadas completas do usuário identificando etapas, ações, sentimentos e oportunidades.
- FT03.06 - Versionamento das Definições: Manter histórico e versões das visões, definições e metas.

Épico 04: Fase 2 - Exploração de Funcionalidades:

- FT04.01 - Brainstorming de Features: Permitir cadastro de ideias de funcionalidades com título, descrição e categoria.
- FT04.02 - Avaliação 360º de Features: Avaliar cada feature em dimensões: valor para usuário, complexidade técnica, alinhamento estratégico, viabilidade e risco.
- FT04.03 - Tags de Features: Classificar features por tags e filtrar por elas.

Épico 05: Fase 3 - Definição do Backlog:

- FT05.01 - Criação de Épicos: Criar épicos com título, descrição e objetivo de negócio.
- FT05.02 - Vinculação de Features a Épicos: Associar features a épicos específicos.
- FT05.03 - Criação de User Stories: Criar histórias de usuário seguindo formato "Como [papel], quero[ação], para [benefício]".
- FT05.04 - Definição de Critérios de Aceite: Documentar critérios de aceite detalhados para cada user story.

Épico 06: Fase 4 - Definição do MVP:

- FT06.01 - Definição do Contexto do MVP: Documentar premissas, restrições e objetivos específicos do MVP.

- FT06.02 - Priorização de Features para MVP: Selecionar e priorizar features essenciais que farão parte do MVP.

Épico 07: Fase 5 - Detalhamento e Fechamento:

- FT07.01 - Visualização Consolidada: Exibir resumo executivo com contadores de todos os artefatos criados.
- FT07.02 - Validação de Completude: Validar se todas as fases obrigatórias foram preenchidas.
- FT07.03 - Exportação para PDF: Gerar documento PDF profissional com todo o conteúdo do discovery.
- FT07.04 - Navegação Entre Artefatos: Permitir navegação rápida entre diferentes artefatos através dos contadores.
- FT07.05 - Template de Exportação: Permitir personalização de capa, sumário e seções do PDF.

Épico 08: Dashboards e Navegação:

- FT08.01 - Dashboard Administrativo: Fornecer dashboard com acesso a gestão de usuários, iniciativas e configurações.
- FT08.02 - Dashboard do Participante: Fornecer dashboard personalizado mostrando iniciativas do participante.
- FT08.03 - Indicador de Progresso: Exibir barra de progresso mostrando fases completadas da iniciativa.

Histórias de Usuários:

- **US01.01 - Cadastro de Novo Usuário:**

Como novo usuário técnico Quero me cadastrar na plataforma informando meu CPF, nome completo e tipo de perfil (stakeholder ou Técnico). Para poder acessar o sistema após aprovação

Critérios de Aceite:

- CA01: O CPF deve ser validado quanto ao formato (XXX.XXX.XXX-XX).
- CA02: O CPF não pode ser duplicado no sistema.

- CA03: O nome completo é obrigatório e deve ter no mínimo 3 caracteres.
- CA04: O usuário deve selecionar obrigatoriamente entre "Stakeholder" ou "Técnico".
- CA05: Após cadastro bem-sucedido, uma mensagem de confirmação deve ser exibida.
- CA06: O usuário fica com status "Pendente de Aprovação" até revisão por admin.
- CA07: Todos os campos são obrigatórios.

- **US01.02 - Login no Sistema**

Como usuário cadastrado e aprovado Quero fazer login usando meu CPF Para acessar as funcionalidades do sistema conforme minhas permissões.

Critérios de Aceite:

- CA01: O campo CPF deve aceitar entrada com ou sem máscara
- CA02: Sistema deve verificar se o usuário está aprovado
- CA03: Se não aprovado, exibir mensagem "Seu cadastro está pendente de aprovação"
- CA04: Se aprovado, redirecionar para dashboard apropriado (admin ou participante)
- CA05: Sessão deve persistir durante navegação no sistema
- CA06: Mensagem de erro clara para CPF não cadastrado
- CA07: Interface deve indicar loading durante autenticação

- **US01.03 - Aprovação de Cadastros Pendentes**

Como administrador Quero visualizar lista de usuários pendentes de aprovação e aprovar ou rejeitar cada um Para controlar o acesso à plataforma e garantir que apenas usuários autorizados entrem.

Critérios de Aceite:

- CA01: Lista deve exibir: nome completo, CPF, tipo de usuário e data de cadastro
- CA02: Deve haver botão "Aprovar" verde para cada usuário pendente
- CA03: Deve haver botão "Rejeitar" vermelho para cada usuário pendente
- CA04: Ao aprovar, status do usuário muda para "Aprovado" imediatamente

- CA05: Ao rejeitar, registro do usuário é removido permanentemente
- CA06: Confirmação deve ser solicitada antes de rejeitar
- CA07: Mensagem de sucesso/erro deve ser exibida após cada ação
- CA08: Lista deve atualizar automaticamente após aprovação/rejeição
- CA09: Contador de pendências deve ser exibido.

- **US01.04 - Gerenciamento de Usuários Ativos**

Como administrador Quero visualizar todos os usuários ativos no sistema com suas informações Para ter visibilidade sobre quem tem acesso à plataforma.

Critérios de Aceite:

- CA01: Lista deve exibir: nome, CPF, tipo (stakeholder/técnico), role (admin/participante)
- CA02: Lista deve mostrar apenas usuários com status "Aprovado"
- CA03: Deve haver indicador visual diferenciando admins de participantes
- CA04: Lista deve ser ordenável por nome ou data de cadastro
- CA05: Campo de busca deve filtrar por nome ou CPF
- CA06: Informação de último acesso deve ser exibida
- CA07: Paginação deve ser implementada para mais de 20 usuários

- **US02.01 - Criar Nova Iniciativa**

Como administrador Quero criar uma nova iniciativa informando título, descrição e status inicial Para organizar um novo projeto de discovery de produto.

Critérios de Aceite:

- CA01: Título é obrigatório e deve ter entre 5 e 100 caracteres
- CA02: Descrição é opcional mas tem limite de 500 caracteres
- CA03: Status inicial deve ser "Planejamento" por padrão
- CA04: Data de criação deve ser registrada automaticamente
- CA05: Criador da iniciativa é registrado automaticamente
- CA06: Após criação, usuário é redirecionado para detalhes da iniciativa

- CA07: 5 fases padrão são criadas automaticamente para a iniciativa
- CA08: Mensagem de sucesso deve ser exibida
- CA09: Modal de criação deve fechar após sucesso

- **US02.02 - Listar Iniciativas**

Como usuário (admin ou participante) Quero visualizar lista de iniciativas disponíveis com informações resumidas Para acessar rapidamente os projetos dos quais participo.

Critérios de Aceite:

- CA01: Lista deve exibir: título, descrição resumida, status, data de criação
- CA02: Badge colorido deve indicar status (planejamento=azul, em progresso=amarelo, concluída=verde, arquivada=cinza)
- CA03: Admin vê todas as iniciativas do sistema
- CA04: Participante vê apenas iniciativas em que foi convidado
- CA05: Campo de busca deve filtrar por título
- CA06: Filtro por status deve estar disponível
- CA07: Ao clicar em uma iniciativa, usuário é direcionado aos detalhes
- CA08: Indicador visual deve mostrar se há fases pendentes
- CA09: Botão "Nova Iniciativa" visível apenas para admins

- **US02.03 - Visualizar Detalhes da Iniciativa**

Como usuário Quero visualizar detalhes completos de uma iniciativa incluindo fases e participantes Para entender o progresso e contexto do projeto

Critérios de Aceite:

- CA01: Título e descrição completa devem ser exibidos
- CA02: Badge de status deve ser exibido
- CA03: Data de criação formatada deve ser mostrada
- CA04: Barra de progresso deve indicar

- CA05: Lista de 5 fases deve ser exibida com indicadores de status
- CA06: Lista de participantes deve mostrar nome e tipo (stakeholder/técnico)
- CA07: Cada fase deve ter botão para acessar detalhes
- CA08: Checkbox deve permitir marcar/desmarcar fase como concluída (admin)
- CA09: Participantes podem apenas visualizar, não editar.

- **US02.04 - Convidar Participantes para Iniciativa**

Como administrador Quero convidar usuários aprovados para participar de uma iniciativa Para formar a equipe que trabalhará no discovery.

Critérios de Aceite:

- CA01: Lista de usuários disponíveis deve excluir já participantes
- CA02: Lista deve exibir nome, CPF e tipo de usuário
- CA03: Admin pode selecionar múltiplos usuários de uma vez
- CA04: Após convite, usuário aparece na lista de participantes
- CA05: Data do convite deve ser registrada
- CA06: Participante vê a iniciativa em sua lista imediatamente
- CA07: Mensagem de sucesso deve confirmar adição
- CA08: Não deve ser possível convidar usuário duas vezes

- **US02.05 - Alterar Status da Iniciativa**

Como administrador Quero alterar o status de uma iniciativa (planejamento, em progresso, concluída, arquivada) Para refletir o estágio atual do projeto

Critérios de Aceite:

- CA01: Dropdown deve listar 4 opções de status
- CA02: Status atual deve estar pré-selecionado
- CA03: Ao alterar, mudança deve ser salva imediatamente
- CA04: Badge de status deve atualizar visualmente
- CA05: Data de última atualização deve ser registrada

- CA06: Mensagem de confirmação deve ser exibida
- CA07: Status deve refletir em todas as listagens
- CA08: Apenas admins podem alterar status

- **US03.01 - Definir Visão do Produto**

Como participante Quero preencher template estruturado de visão do produto Para documentar claramente público-alvo, necessidade, categoria, benefício e diferencial.

Critérios de Aceite:

- CA01: Template deve ter 6 campos: público-alvo, necessidade, nome do produto, categoria, benefício principal, diferencial
- CA02: Todos os campos são obrigatórios
- CA03: Cada campo deve ter limite de 200 caracteres
- CA04: Preview da frase de visão deve ser exibido em tempo real
- CA05: Botão salvar deve persistir informações
- CA06: Mensagem de sucesso deve confirmar salvamento
- CA07: Dados devem ser recuperáveis ao retornar à página
- CA08: Múltiplas visões podem ser criadas
- CA09: Visões anteriores devem ser listadas

- **US03.02 - Definir Produto (Is/Is Not, Does/Does Not)**

Como participante Quero documentar o que o produto é, não é, faz e não faz Para estabelecer limites claros do escopo do produto.

Critérios de Aceite:

- CA01: Formulário deve ter 4 campos de texto longo
- CA02: Campos: "O produto É", "O produto NÃO É", "O produto FAZ", "O produto NÃO FAZ"
- CA03: Cada campo deve suportar múltiplas linhas (textarea)
- CA04: Limite de 1000 caracteres por campo

- CA05: Ao menos um campo deve ser preenchido
- CA06: Dados devem ser salvos ao clicar em "Salvar"
- CA07: Loading indicator durante salvamento
- CA08: Múltiplas definições podem coexistir
- CA09: Lista de definições anteriores deve ser exibida

- **US03.03 - Definir Metas e Objetivos**

Como participante Quero registrar metas de negócio e objetivos mensuráveis do produto Para estabelecer critérios claros de sucesso.

Critérios de Aceite:

- CA01: Campo para meta principal (obrigatório, até 500 caracteres)
- CA02: Lista dinâmica de objetivos mensuráveis
- CA03: Cada objetivo deve ter: descrição, métrica e meta numérica
- CA04: Botão "Adicionar Objetivo" para inserir múltiplos objetivos
- CA05: Mínimo de 1 objetivo é obrigatório
- CA06: Botão remover para cada objetivo
- CA07: Validação de campos numéricos para metas
- CA08: Dados salvos persistem entre sessões
- CA09: Múltiplas versões de metas podem ser criadas

- **US03.04 - Criar Personas**

Como participante Quero criar personas detalhadas com características demográficas, objetivos e desafios Para ter empatia e entender profundamente os usuários do produto.

Critérios de Aceite:

- CA01: Formulário deve incluir: nome, idade, profissão, bio, objetivos, desafios, comportamentos
- CA02: Nome, idade e profissão são obrigatórios

- CA03: Idade deve ser numérica entre 1 e 120
- CA04: Objetivos e desafios devem aceitar múltiplas entradas (lista)
- CA05: Botão "Adicionar Objetivo/Desafio" para incluir itens
- CA06: Foto placeholder deve ser exibida
- CA07: Múltiplas personas podem ser criadas
- CA08: Lista de personas deve mostrar card resumido
- CA09: Cada persona deve ser editável e deletável
- CA10: Confirmação antes de deletar

- **US03.05 - Mapear Jornadas do Usuário**

Como participante Quero mapear jornadas completas dos usuários identificando etapas, ações, sentimentos e oportunidades Para visualizar e otimizar a experiência do usuário.

Crítérios de Aceite:

- CA01: Jornada deve ter: nome, persona associada, descrição
- CA02: Jornada composta por múltiplas etapas
- CA03: Cada etapa tem: nome, ações do usuário, sentimento (escala 1-5), pontos de dor, oportunidades.
- CA04: Botão "Adicionar Etapa" para criar nova etapa
- CA05: Etapas devem ser reordenáveis (drag and drop)
- CA06: Escala de sentimento deve ter indicador visual (emoji/cor)
- CA07: Múltiplas jornadas podem ser criadas
- CA08: Dropdown para selecionar persona já criada
- CA09: Visualização de timeline das etapas
- CA10: Edição e exclusão de etapas individuais

- **US04.01 - Brainstorm de Features**

Como participante Quero cadastrar ideias de funcionalidades com título, descrição e categoria Para capturar todas as possibilidades antes de priorizar

Critérios de Aceite:

- CA01: Formulário com campos: título (obrigatório, 5-100 caracteres), descrição (opcional, até 500 caracteres), categoria
- CA02: Categorias pré-definidas: Autenticação, Interface, Integração, Relatórios, Outros
- CA03: Botão "Adicionar Feature" abre modal de criação
- CA04: Lista de features exibe cards com título, categoria e descrição resumida
- CA05: Features devem ser editáveis
- CA06: Features devem ser deletáveis com confirmação
- CA07: Contador de total de features
- CA08: Filtro por categoria
- CA09: Campo de busca por título

- **US04.02 - Avaliação 360º de Features**

Como participante Quero avaliar cada feature em 5 dimensões (valor, complexidade, alinhamento, viabilidade, risco) Para ter visão completa sobre cada funcionalidade antes de priorizar.

Critérios de Aceite:

- CA01: Avaliação deve ter 5 dimensões com escala de 1-5
- CA02: Dimensões: Valor para Usuário, Complexidade Técnica, Alinhamento Estratégico, Viabilidade, Risco.
- CA03: Cada dimensão deve ter slider ou radio buttons
- CA04: Campo opcional para comentários/observações (500 caracteres)
- CA05: Feature deve estar previamente cadastrada
- CA06: Múltiplas avaliações por feature (diferentes participantes)

- CA07: Média das avaliações deve ser calculada e exibida
- CA08: Visualização gráfica (radar chart) das dimensões
- CA09: Histórico de avaliações deve ser mantido

- **US05.01 - Criar Épicos**

Como participante Quero criar épicos com título, descrição e objetivo de negócio Para organizar features relacionadas em agrupamentos lógicos.

Critérios de Aceite:

- CA01: Formulário com: título (obrigatório, 5-100 caracteres), descrição (até 500 caracteres), objetivo de negócio (até 300 caracteres)
- CA02: Todos os campos são obrigatórios
- CA03: Épicos exibidos em cards com título e descrição resumida
- CA04: Épicos podem ser editados
- CA05: Épicos podem ser excluídos se não tiverem features associadas
- CA06: Contador de features por épico
- CA07: Status do épico (planejado, em andamento, concluído)
- CA08: Lista ordenável por nome ou data de criação

- **US05.02 - Vincular Features a Épicos**

Como participante Quero associar features brainstormadas a épicos específicos Para organizar o backlog de forma estruturada

Critérios de Aceite:

- CA01: Visualização de épico deve listar features vinculadas
- CA02: Botão "Adicionar Feature" abre seleção de features disponíveis
- CA03: Features já vinculadas não aparecem na lista de seleção
- CA04: Feature pode estar vinculada a apenas um épico
- CA05: Drag and drop para mover feature entre épicos
- CA06: Confirmação ao desvincular feature

- CA07: Contador de features atualiza automaticamente
- CA08: Filtro para mostrar apenas features não vinculadas

- **US05.03 - Criar User Stories**

Como participante Quero criar histórias de usuário seguindo formato padrão "Como [papel], quero [ação], para [benefício]" Para detalhar requisitos de forma compreensível para toda a equipe.

Critérios de Aceite:

- CA01: Formulário com 3 campos obrigatórios: papel, ação, benefício
- CA02: Template "Como [papel], quero [ação], para [benefício]" exibido
- CA03: Preview da história formatada em tempo real
- CA04: Campo para título curto da história (até 100 caracteres)
- CA05: Story deve ser vinculada a uma feature
- CA06: Campo para pontos de estimativa (Fibonacci: 1,2,3,5,8,13,21)
- CA07: Campo para prioridade (Alta, Média, Baixa)
- CA08: Múltiplas stories por feature
- CA09: Stories devem ser editáveis e deletáveis

- **US05.04 - Definir Critérios de Aceite**

Como participante Quero documentar critérios de aceite detalhados para cada user story Para estabelecer condições claras de conclusão.

Critérios de Aceite:

- CA01: Lista dinâmica de critérios por user story
- CA02: Cada critério tem: ID sequencial (CA01, CA02...), descrição
- CA03: Botão "Adicionar Critério" para incluir novos
- CA04: Mínimo de 3 critérios por story
- CA05: Critérios podem ser reordenados
- CA06: Critérios podem ser editados inline

- CA07: Checkbox para marcar critério como implementado (dev)
- CA08: Contador de critérios implementados vs total
- CA09: Exportação de critérios em formato checklist

- **US06.01 - Definir Contexto do MVP**

Como participante Quero documentar premissas, restrições e objetivos específicos do MVP Para alinhar expectativas sobre a primeira versão do produto.

Critérios de Aceite:

- CA01: Formulário com 3 seções: Premissas, Restrições, Objetivos do MVP
- CA02: Cada seção suporta múltiplas entradas (lista)
- CA03: Campo de texto para cada item (até 300 caracteres)
- CA04: Botão "Adicionar" em cada seção
- CA05: Itens podem ser removidos individualmente
- CA06: Data-alvo de lançamento (campo data)
- CA07: Orçamento estimado (campo numérico opcional)
- CA08: Mínimo de 2 premissas, 2 restrições e 3 objetivos
- CA09: Dados salvos persistem entre sessões

- **US06.02 - Priorizar Features para MVP**

Como participante Quero selecionar e priorizar features essenciais usando método MoSCoW Para definir o que entra ou não na primeira versão.

Critérios de Aceite:

- CA01: Lista de todas as features do backlog
- CA02: Classificação MoSCoW para cada feature: Must have, Should have, Could have, Won't have
- CA03: Drag and drop para mover features entre categorias
- CA04: Contadores por categoria
- CA05: Validação: pelo menos 3 features "Must have"

- CA06: Validação: "Must have"+ "Should have" não pode exceder capacidade estimada
- CA07: Campo de justificativa ao marcar como "Won't have"
- CA08: Visualização de % de capacidade utilizada
- CA09: Exportação da priorização em PDF

- **US07.01 - Visualizar Dashboard Consolidado**

Como participante Quero visualizar resumo executivo com contadores de todos os artefatos criados Para ter visão rápida da completude do discovery

Critérios de Aceite:

- CA01: Cards com contadores: Personas, Jornadas, Features, Épicos, Backlog, User Stories, MVP
- CA02: Cada card é clicável e navega para seção correspondente
- CA03: Indicador visual de fases completadas vs pendentes
- CA04: Porcentagem de conclusão geral
- CA05: Alertas para seções obrigatórias não preenchidas
- CA06: Data de última atualização
- CA07: Informações organizadas visualmente (grid)
- CA08: Hover nos cards mostra detalhes adicionais

- **US07.02 - Validar Completude do Discovery**

Como administrador Quero validar se todas as fases obrigatórias foram preenchidas adequadamente Para garantir qualidade antes da finalização

Critérios de Aceite:

- CA01: Lista de checklist com todos os requisitos mínimos
- CA02: Indicadores visuais: verde (completo), amarelo (incompleto), vermelho (não iniciado)
- CA03: Validações: 1 visão, 1 definição, 3+ metas, 3+ personas, 2+ jornadas, 10+ features, 3+ épicos, 20+ stories, 1 contexto MVP, priorização MVP

- CA04: Botão "Validar Agora" força revalidação
- CA05: Mensagens de erro específicas para cada validação falhada
- CA06: Exportação bloqueada se validação falhar
- CA07: Alerta destacado para itens faltantes
- CA08: Opção de override para admin (com justificativa)

- **US07.03 - Exportar Discovery em PDF**

Como participante Quero gerar documento PDF profissional com todo o conteúdo do discovery Para compartilhar resultados com stakeholders e equipe de desenvolvimento.

Critérios de Aceite:

- CA01: PDF deve ter capa com logo, título da iniciativa e data
- CA02: Índice clicável com todas as seções
- CA03: Cabeçalho em todas as páginas: "UPLean - Discovery"+ nome da iniciativa
- CA04: Rodapé com número de página e data de exportação
- CA05: Seção 1: Alinhamento (visão, definição, metas, personas, jornadas)
- CA06: Seção 2: Features e avaliações
- CA07: Seção 3: Backlog (épicos e stories)
- CA08: Seção 4: MVP (contexto e priorização)
- CA09: Formatação profissional com títulos, subtítulos, tabelas
- CA10: Botão "Download PDF" inicia download automaticamente
- CA11: Loading indicator durante geração
- CA12: Tratamento de erros com mensagem clara

- **US07.04 - Navegar Entre Artefatos via Contadores**

Como participante Quero clicar nos contadores do dashboard e ser direcionado à seção específica Para acessar rapidamente qualquer artefato do discovery

Critérios de Aceite:

- CA01: Click em "Personas = X" navega para fase de Personas
- CA02: Click em "Jornadas = X" navega para fase de Jornadas
- CA03: Click em "Features = X" navega para fase de Features
- CA04: Click em "Épicos = X" navega para fase de Backlog (tab Épicos)
- CA05: Click em "Backlog = X" navega para fase de Backlog
- CA06: Click em "User Stories = X" navega para fase de Backlog (tab Stories)
- CA07: Click em "MVP Features = X" navega para fase de Priorização MVP
- CA08: Navegação deve manter contexto da iniciativa
- CA09: Botão voltar retorna ao dashboard consolidado

- **US08.01 - Dashboard Administrativo**

Como administrador Quero ter dashboard com acesso rápido a gestão de usuários, iniciativas e configurações Para administrar a plataforma eficientemente.

Critérios de Aceite:

- CA01: 3 cards principais: Gerenciar Usuários, Iniciativas, Configurações
- CA02: Card "Gerenciar Usuários" mostra contador de pendências
- CA03: Card "Iniciativas" mostra total de iniciativas ativas
- CA04: Cards são clicáveis e navegam para seções respectivas
- CA05: Saudação personalizada com nome do admin
- CA06: Ícones visuais para cada card
- CA07: Hover effect nos cards
- CA08: Layout responsivo (mobile e desktop)
- CA09: Card "Configurações" desabilitado com label "Em breve"

- **US08.02 - Dashboard do Participante**

Como participante Quero visualizar dashboard personalizado mostrando minhas iniciativas Para acessar rapidamente os projetos dos quais faço parte.

Critérios de Aceite:

- CA01: Saudação personalizada com nome do participante
- CA02: Card "Ver Iniciativas" com descrição
- CA03: Contador de iniciativas ativas do usuário
- CA04: Ícone visual representativo
- CA05: Click no card navega para lista de iniciativas
- CA06: Layout limpo e focado
- CA07: Mensagem motivacional ou dica de uso
- CA08: Responsivo para mobile e desktop

- **US08.03 - Indicador de Progresso da Iniciativa**

Como usuário Quero visualizar barra de progresso mostrando % de fases completadas Para acompanhar visualmente o andamento do discovery.

Critérios de Aceite:

- CA01: Barra de progresso com % calculado (fases completas / total)
- CA02: Animação suave ao atualizar progresso
- CA03: Cores indicativas: 0-30% (vermelho), 31-70% (amarelo), 71-100% (verde)
- CA04: Texto ao lado: "X de 5 fases completadas"
- CA05: Tooltip ao hover mostra quais fases estão completas
- CA06: Atualização automática ao marcar/desmarcar fase
- CA07: Exibido no topo da página de detalhes da iniciativa
- CA08: Responsivo em diferentes tamanhos de tela

- **US01.05 - Recuperar Acesso**

Como usuário Quero recuperar meu acesso solicitando um código temporário Para voltar a usar o sistema com segurança.

Critérios de Aceite:

- CA01: Formulário solicita CPF e e-mail
- CA02: Envio de código de 6 dígitos válido por 15 minutos

- CA03: Máximo de 3 tentativas por código
- CA04: Após validação, acesso é restabelecido e evento é auditado
- CA05: Mensagens claras de sucesso e erro.

- **US01.06 - Consultar Auditoria de Acessos**

Como administrador Quero visualizar registros de login, logout e tentativas falhas Para auditar segurança e investigar incidentes.

Critérios de Aceite:

- CA01: Listagem com filtros por usuário, data e evento
- CA02: Exportação CSV dos registros
- CA03: Retenção mínima de 90 dias
- CA04: Paginada para grandes volumes
- CA05: Campos: data/hora, usuário, IP, tipo do evento, status

- **US02.06 - Definir Papéis dos Participantes**

Como administrador Quero atribuir papéis (PO, Facilitador, Convidado) aos participantes da iniciativa Para deixar claras as responsabilidades de cada um.

Critérios de Aceite:

- CA01: Dropdown de papéis por participante
- CA02: Múltiplos participantes por papel
- CA03: Registro de histórico de alterações
- CA04: Filtro por papel na lista de participantes
- CA05: Somente admins podem alterar papéis

- **US02.07 - Gerenciar Anexos da Iniciativa**

Como participante Quero enviar e organizar arquivos da iniciativa (PDF, imagens, planilhas) Para centralizar documentos do discovery

Critérios de Aceite:

- CA01: Upload com barra de progresso

- CA02: Tipos suportados: PDF, PNG/JPG, XLSX, DOCX (até 10MB)
- CA03: Exibição em lista com nome, tipo, tamanho e autor
- CA04: Download e exclusão com confirmação
- CA05: Organização por pastas ou tags

- **US03.06 - Versionamento das Definições**

Como participante Quero criar e comparar versões das visões, definições e metas Para rastrear evolução e recuperar versões anteriores.

Critérios de Aceite:

- CA01: Botão "Criar nova versão" em cada artefato
- CA02: Histórico com data, autor e comentários
- CA03: Comparação lado a lado entre versões
- CA04: Restauração de versão com confirmação
- CA05: Controle de permissões por perfil

- **US04.03 - Gerenciar Tags de Features**

Como participante Quero adicionar e remover tags em features para melhor organização Para filtrar e agrupar funcionalidades por tema

Critérios de Aceite:

- CA01: Cadastro de tags com cor
- CA02: Autocomplete de tags existentes
- CA03: Filtro por múltiplas tags
- CA04: Edição e remoção de tags
- CA05: Contadores por tag

- **US05.05 - Definir Workflow das User Stories**

Como participante Quero atribuir status às histórias (A Fazer, Em Progresso, Concluída) Para acompanhar execução do backlog.

Critérios de Aceite:

- CA01: Board kanban simples por status
- CA02: Drag and drop entre colunas
- CA03: Histórico de movimentações
- CA04: Filtro por status e prioridade
- CA05: Somente membros da iniciativa podem alterar status

- **US07.05 - Personalizar Template de Exportação**

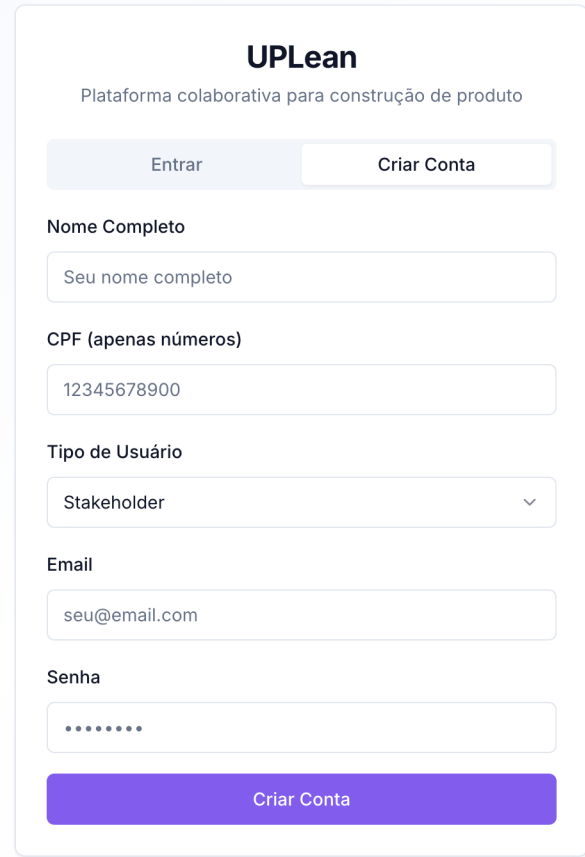
Como participante Quero customizar capa, sumário e seções do PDF antes da exportação Para adequar o documento ao público-alvo da apresentação.

Critérios de Aceite:

- CA01: Seleção de seções a incluir/excluir
- CA02: Upload de logomarca e escolha de cores
- CA03: Campos para notas de rodapé e disclaimers
- CA04: Preview antes de exportar
- CA05: Configuração lembrada por iniciativa

A.4 Telas do Protótipo - UPLearn

Figura A.23: Tela de Cadastrar Novo Usuário



UPLearn
Plataforma colaborativa para construção de produto

Entrar Criar Conta

Nome Completo
Seu nome completo

CPF (apenas números)
12345678900

Tipo de Usuário
Stakeholder

Email
seu@email.com

Senha
.....

Criar Conta

Fonte: Elaborada pela autora (2025)

Figura A.24: Tela de Login do Sistema



UPLearn
Plataforma colaborativa para construção de produto

Entrar Criar Conta

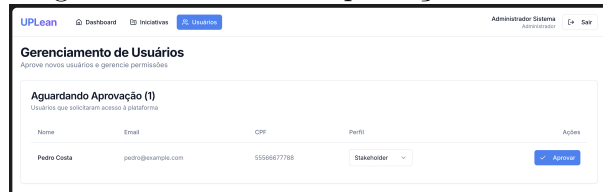
Email
seu@email.com

Senha
.....

Entrar

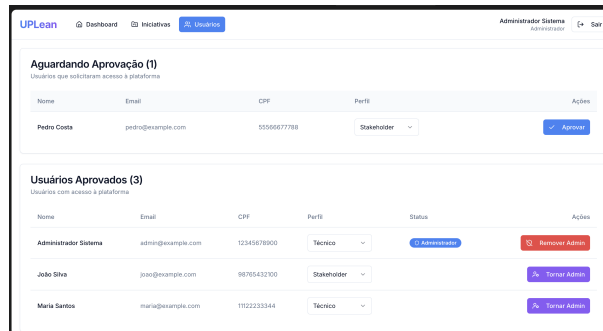
Fonte: Elaborada pela autora (2025)

Figura A.25: Tela de Aprovação Pendente



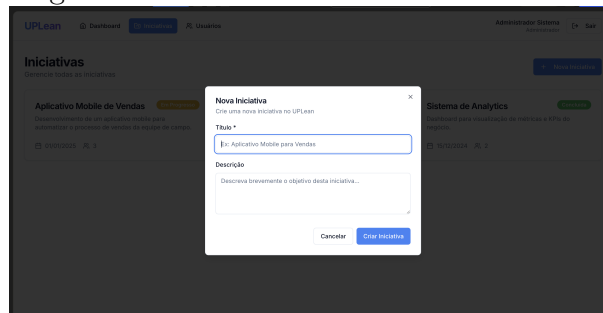
Fonte: Elaborada pela autora (2025)

Figura A.26: Tela de Gerenciamento de Usuário Ativo



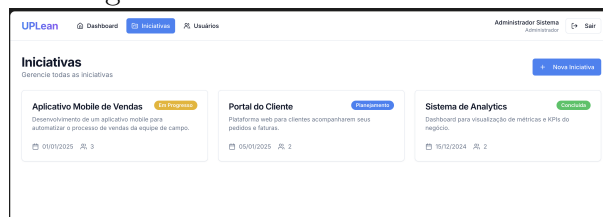
Fonte: Elaborada pela autora (2025)

Figura A.27: Tela de Criar Nova Iniciativa



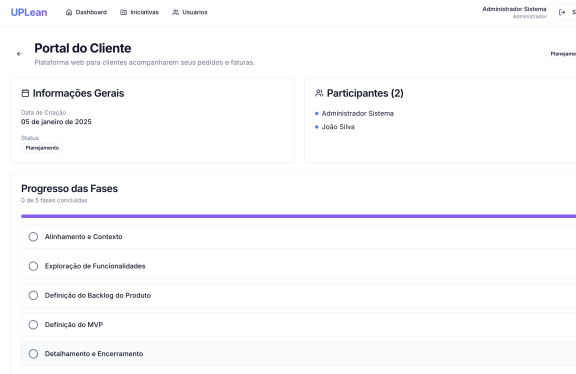
Fonte: Elaborada pela autora (2025)

Figura A.28: Tela Listar Iniciativa



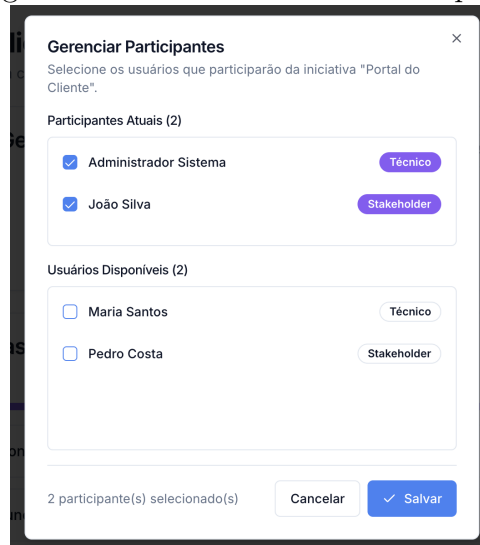
Fonte: Elaborada pela autora (2025)

Figura A.29: Tela Visualizar Detalhe da Iniciativa



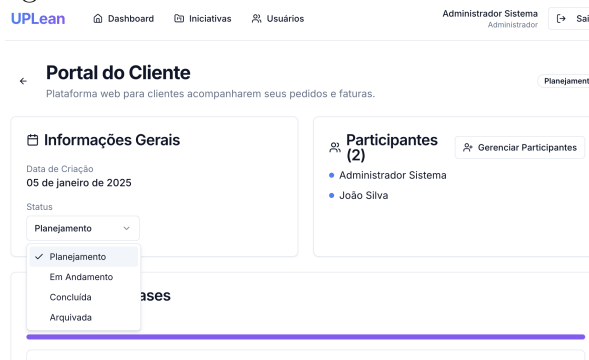
Fonte: Elaborada pela autora (2025)

Figura A.30: Tela Convidar Participante



Fonte: Elaborada pela autora (2025)

Figura A.31: Tela Alterar Status da Iniciativa



Fonte: Elaborada pela autora (2025)

Figura A.32: Tela Visão do Produto

UPLEAN Dashboard Iniciativas Usuários Administrador Sistema

Alinhamento e Contexto
Portal do Cliente

1.1 Visão 1.2 Definição 1.3 Metas 1.4 Personas 1.5 Jornadas

Visão do Produto
Em algum lugar entre a ideia e o lançamento do MVP, a visão do produto ajuda você a trilhar o caminho inicial.

Problema *
Descreva o problema que o produto resolve
Preencha este campo.

Visão do Produto *
Descreva a visão do produto em uma única frase clara e convincente

Palavras: 0

Para (público-alvo) *
Quem é o público-alvo?

Que (necessidade ou problema) *
Qual necessidade ou problema será resolvido?

O (nome do produto) *
Nome do produto

É (unidade (categoria do produto) *
Categoria do produto (ex: aplicativo mobile, plataforma web)

Que (principal benefício) *
Qual o principal benefício ou valor entregue?

Ao contrário de (alternativa atual) *
Qual a alternativa ou solução atual?

Nosso produto (diferencial) *
Qual o principal diferencial?

Salvar Visão do Produto

Fonte: Elaborada pela autora (2025)

Figura A.33: Tela Definir Visão do Produto

UPLEAN Dashboard Iniciativas Usuários Administrador Sistema

Alinhamento e Contexto
Portal do Cliente

1.1 Visão 1.2 Definição 1.3 Metas 1.4 Personas 1.5 Jornadas

Definição do Produto
Decida o que NÃO fazer e TÃO IMPORTANTE quanto decidir o que fazer.

O que o Produto É *
Descreva o que o produto é

Palavras: 0

O que o Produto NÃO É *
Descreva o que o produto não é

Palavras: 0

O que o Produto FAZ *
Descreva o que o produto faz

Palavras: 0

O que o Produto NÃO FAZ *
Descreva o que o produto não faz

Palavras: 0

Salvar Definição do Produto

Fonte: Elaborada pela autora (2025)

Figura A.34: Tela Definir Metas e Objetivos

UPLEAN Dashboard Iniciativas Usuários Administrador Sistema

Alinhamento e Contexto
Portal do Cliente

1.1 Visão 1.2 Definição 1.3 Metas 1.4 Personas 1.5 Jornadas

Metas do Produto
Se você deseja que cresça o produto em três objetivos de negócios, quais seriam eles?

Objetivo *
Descreva o objetivo
Palavras: 0

Metas * + Adicionar Meta

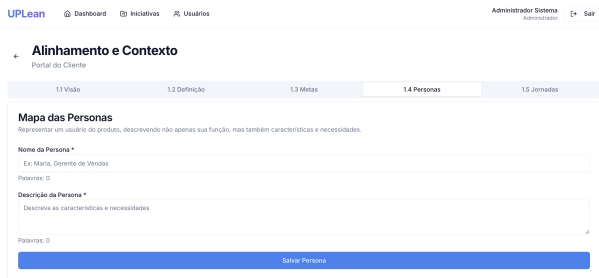
Meta 1

Meta 2

Salvar Objetivos e Metas

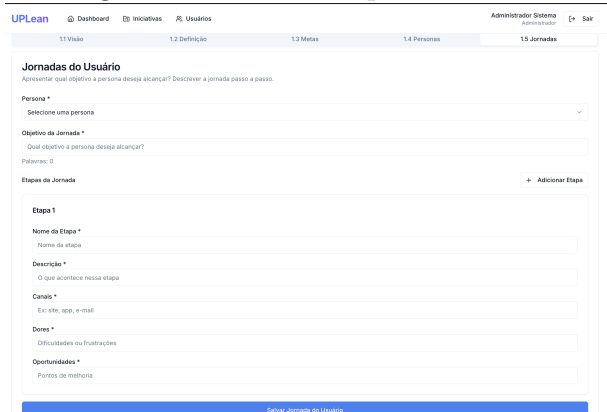
Fonte: Elaborada pela autora (2025)

Figura A.35: Tela Criar Personas



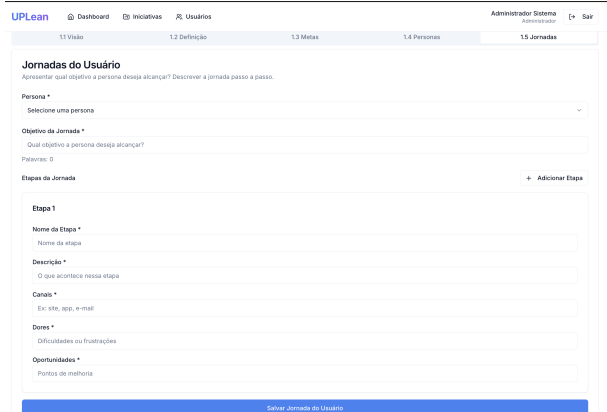
Fonte: Elaborada pela autora (2025)

Figura A.36: Tela Mapear Jornada



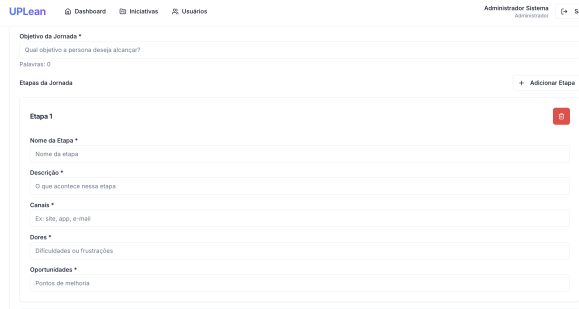
Fonte: Elaborada pela autora (2025)

Figura A.37: Tela Mapear Jornada



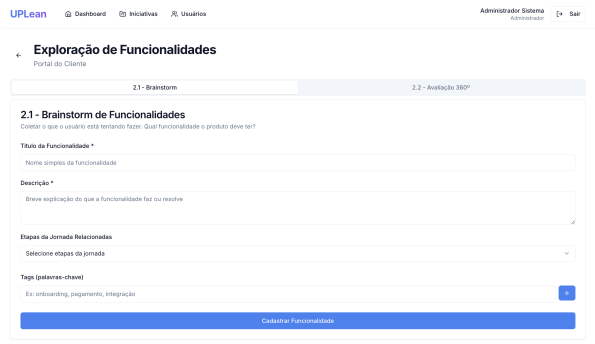
Fonte: Elaborada pela autora (2025)

Figura A.38: Tela Adicionar Etapa



Fonte: Elaborada pela autora (2025)

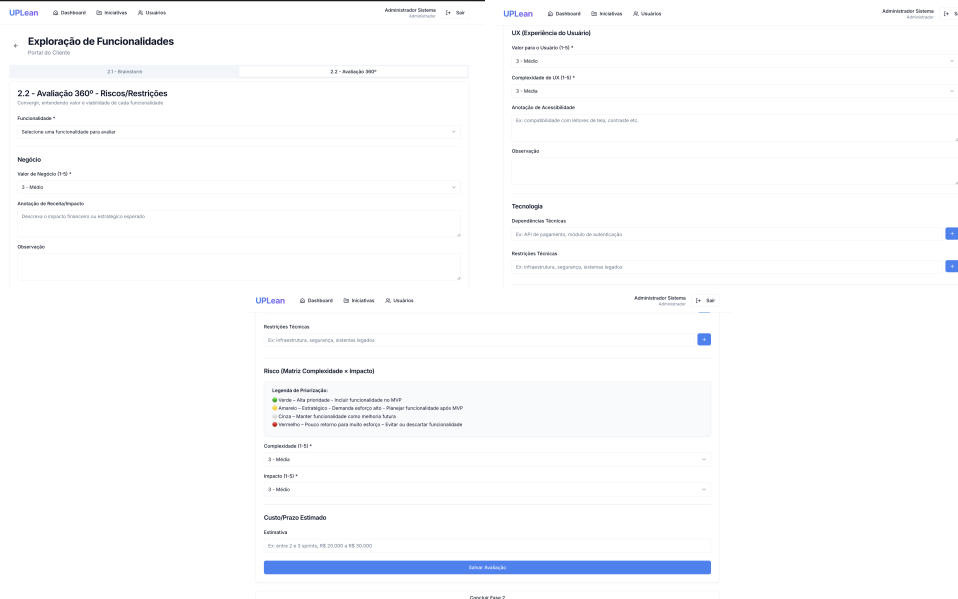
Figura A.39: Tela Brainstorm de Feature



Funcionalidades Cadastradas

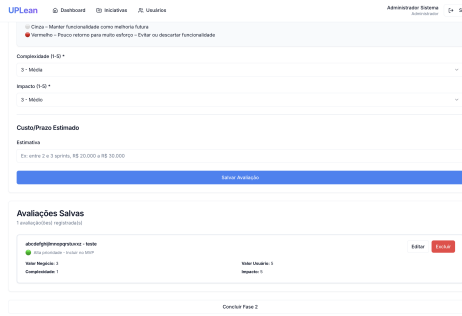
Fonte: Elaborada pela autora (2025)

Figura A.40: Tela Avaliação 360



Fonte: Elaborada pela autora (2025)

Figura A.41: Tela Avaliação Salva



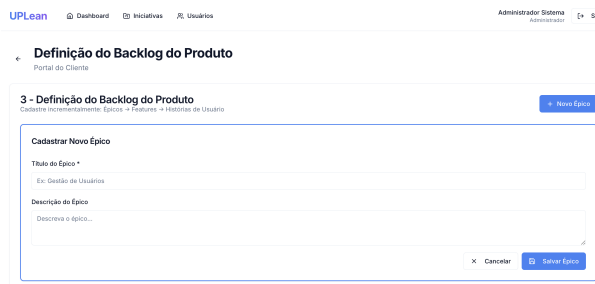
Fonte: Elaborada pela autora (2025)

Figura A.42: Tela Criar Épico



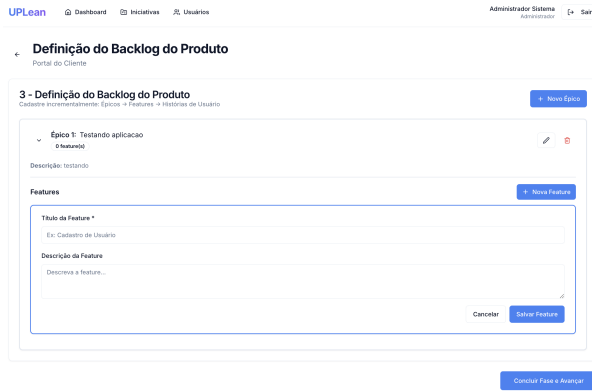
Fonte: Elaborada pela autora (2025)

Figura A.43: Tela Criar Épico



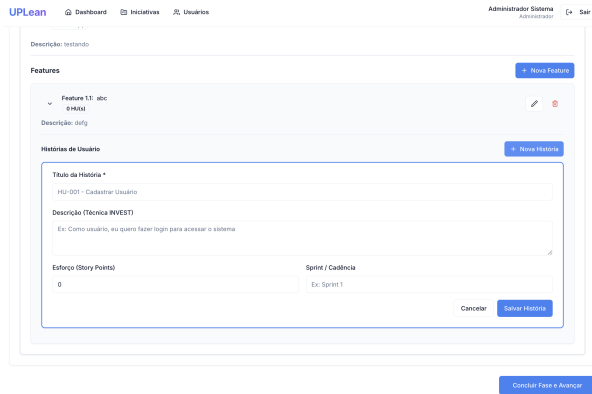
Fonte: Elaborada pela autora (2025)

Figura A.44: Tela Vincular Feature a Epico



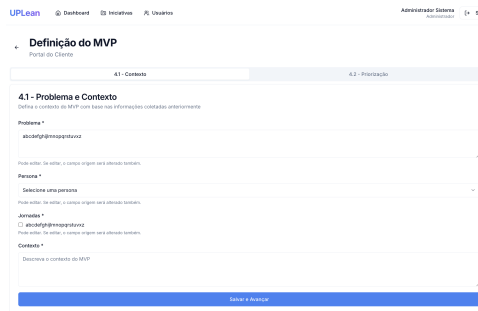
Fonte: Elaborada pela autora (2025)

Figura A.45: Tela Criar User Stories



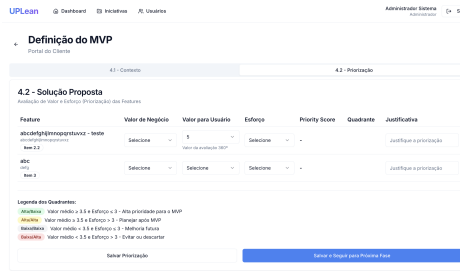
Fonte: Elaborada pela autora (2025)

Figura A.46: Tela Definir MVP



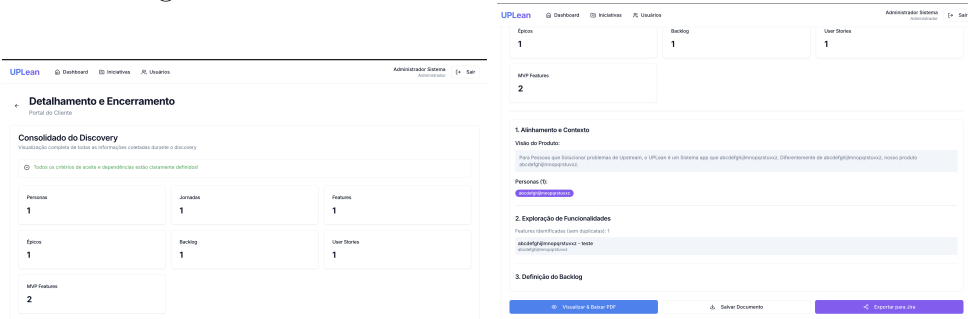
Fonte: Elaborada pela autora (2025)

Figura A.47: Tela Priorizar Feature para MVP



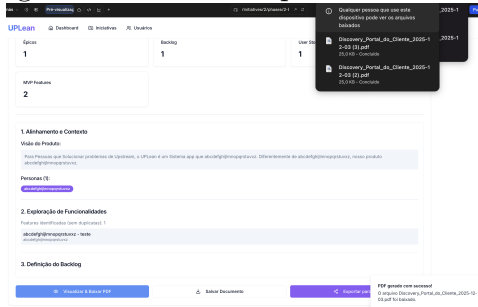
Fonte: Elaborada pela autora (2025)

Figura A.48: Tela Visualizar Dashboard Consolidado



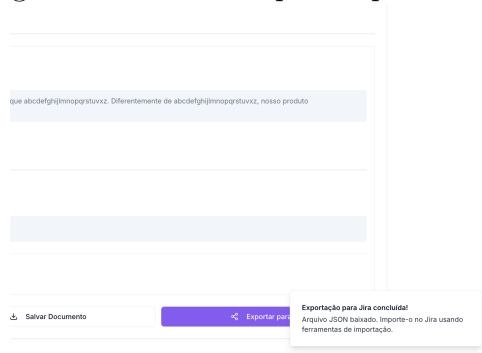
Fonte: Elaborada pela autora (2025)

Figura A.49: Tela Exportar Discovery



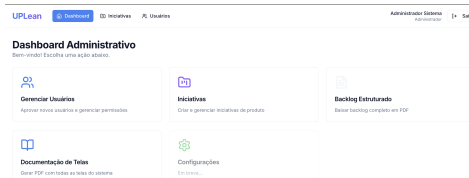
Fonte: Elaborada pela autora (2025)

Figura A.50: Tela Exportar para o Jira



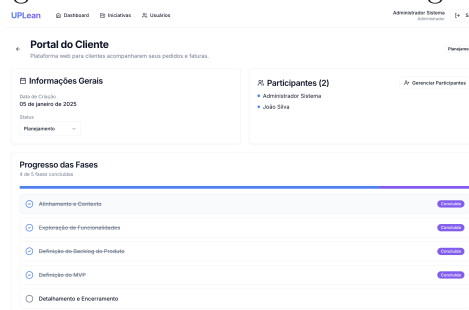
Fonte: Elaborada pela autora (2025)

Figura A.51: Tela Dashboard Administrador



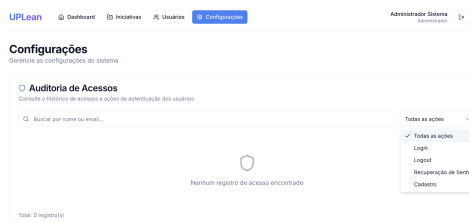
Fonte: Elaborada pela autora (2025)

Figura A.52: Tela Indicador Progresso



Fonte: Elaborada pela autora (2025)

Figura A.53: Tela Consultar Auditoria de Acessos



Fonte: Elaborada pela autora (2025)