

Análise de Requisitos de Software

Blau B. da Rosa Filho, Brena N. Oliveira, Caio S. Trindade, Erick D. Aurelio,
Felipe H. Scherer, Heverton Vilas Boas Tavares

¹Universidade Federal do Pampa - UNIPAMPA
Alegrete, RS - Brasil

Abstract. *Requirements analysis is an essential process for identifying, understanding, and documenting the needs and expectations of users in a software system. Its main objective is to ensure that the software is designed and implemented in a way that meets user needs and business requirements. During this process, requirements analysts often face common challenges such as cognitive understanding, frequent changes, conflicts, and incomplete or ill-defined requirements in relation to business requirements. These challenges can result in rework and additional costs after the software is launched.*

The aim of this article is to address these challenges, present best practices, and describe the essential stages of the requirements analysis process, which include identification, analysis, verification, and validation of requirements. Additionally, emerging trends in the market and available tools to assist in this process will also be explored.

Keywords: Requirements Analysis (AR), Requirements Engineering (ER), Software Engineering (ES).

Resumo. *A análise de requisitos é um processo essencial para identificar, compreender e documentar as necessidades e expectativas dos usuários em um sistema de software. Seu principal objetivo é garantir que o software seja projetado e implementado de maneira a atender às necessidades dos usuários e aos requisitos de negócios. Durante esse processo, os profissionais de análise de requisitos frequentemente enfrentam desafios comuns, como compreensão cognitiva, mudanças frequentes, conflitos e requisitos incompletos ou mal definidos em relação aos requisitos de negócios. Esses desafios podem resultar em retrabalho e custos adicionais após o lançamento do software.*

O objetivo deste artigo é abordar esses desafios, apresentar as melhores práticas e descrever as etapas essenciais do processo de análise de requisitos, que incluem a identificação, análise, comprovação e validação dos requisitos. Além disso, também serão exploradas as tendências emergentes no mercado e as ferramentas disponíveis para auxiliar nesse processo.

Palavras-Chave: Análise de Requisitos (AR), Engenharia de Requisitos (ER), Engenharia de Software (ES).

1. Introdução

Na Engenharia de Software, é crucial ter medidas eficazes e uma equipe eficiente para produzir um produto final de qualidade e com maior probabilidade de sucesso. [Sommerville 2015]. Para alcançar esse objetivo, é necessário adotar uma metodologia

adequada. A falta de esforço não é o principal problema, mas sim a utilização de uma metodologia inadequada. Muitos desenvolvedores iniciam projetos sem uma compreensão clara dos requisitos necessários.

De acordo com [Júnior 2012], um projeto é um conjunto de ações coordenadas, planejadas de forma metódica com o objetivo de atingir determinados objetivos, ideais ou materiais, visando o maior sucesso e satisfação possível. Infelizmente, a falta de análise adequada e a ausência de metodologias eficazes são fatores comuns que contribuem para o fracasso de projetos. [sotille 2004]. Além disso, as mudanças nos sistemas e requisitos podem levar à inconsistência na especificação e implementação, resultando na perda de credibilidade dos requisitos. Diante disso, é necessário adotar práticas de Análise de Requisitos e Controle de Mudanças [Sommerville 2015].

A análise de requisitos de software é o processo de identificar, entender e documentar as necessidades e expectativas dos usuários e de um sistema de software. O objetivo principal da análise de requisitos é garantir que o software seja projetado e implementado para atender às necessidades dos usuários e aos requisitos de negócios. Essa atividade ajuda a minimizar o retrabalho e os custos associados à correção de problemas do projeto após o lançamento, sendo assim, uma etapa crucial do processo de desenvolvimento. Isso evidencia a necessidade de uma abordagem cuidadosa e rigorosa para a análise de requisitos de software. [Sommerville 2015].

O constante aprimoramento da tecnologia é essencial para atender às demandas da sociedade atual, especialmente na área da Análise de Requisitos. A adaptação às mudanças nos requisitos de software requer uma organização prática e flexível do projeto. Este artigo visa apresentar uma revisão sobre a análise de requisitos de software, abrangendo as melhores práticas e técnicas, bem como explorar as várias etapas do processo de análise de requisitos, incluindo a identificação, análise, especificação e validação de requisitos. Além disso, serão abordadas as ferramentas e tecnologias disponíveis para apoiar a análise de requisitos de software, bem como os desafios comuns enfrentados pelos profissionais e as tendências emergentes no mercado.

Na análise de requisitos, alguns dos principais desafios são compreensão inadequada, mudanças frequentes, conflitos, requisitos incompletos ou mal definidos, requisitos não realistas e comunicação inadequada com os usuários. É necessário que os analistas estejam cientes desses problemas e tentem mitigá-los o máximo possível para obter um bom resultado no final. Com o intuito de promover um melhor conhecimento sobre a análise de requisitos, este artigo visa compreender o papel exercido pela área, tendo como objetivo final fornecer uma visão geral abrangente da análise de requisitos de software e ajudar profissionais a entender a importância dessa atividade crítica e como executá-la com sucesso. Espera-se melhorar.

A parte inicial desse artigo apresenta uma introdução sobre este trabalho. Na Seção 2, serão abordados a fundamentação teórica para o embasamento deste artigo. Na Seção 3, serão apresentados exemplos de aplicação prática na análise de requisitos. Na Seção 4, serão apresentadas algumas ferramentas que auxiliam na análise de requisitos. Na Seção 5, serão apresentadas as vantagens e desvantagens da utilização da análise de requisitos. Na Seção 6, serão apresentados os desafios encontrados durante a análise de requisitos. E na Seção 7, é apresentada uma conclusão sobre esse artigo.

2. Fundamentação Teórica

Na área de desenvolvimento de software, a análise é uma etapa fundamental com o objetivo de compreender as necessidades e expectativas do cliente em relação ao produto final [Pressman 2019]. Esta seção apresenta uma base teórica que serve de guia para as análises e o conteúdo deste trabalho.

2.1. Engenharia de Requisitos

É um campo de estudo importante na Engenharia de Software, que se preocupa com identificar, validar e documentar funções e restrições do Software. Esta etapa pode ocorrer em qualquer ciclo de vida de um software, tanto no seu nascimento, quanto na sua evolução. Em suma, a Engenharia de Requisitos é uma atividade que garante a qualidade e a satisfação do software na entrega [Lopes 2002].

2.2. Requisitos

Os requisitos de software são as necessidades e expectativas do cliente em relação ao produto final. Eles descrevem as funcionalidades, características e restrições que o software deve possuir para atender às necessidades do cliente. Desse modo, fornecem um conjunto claro e preciso de objetivos e especificações que devem ser alcançados durante o processo de desenvolvimento, orientando e definindo o que o software deve fazer e como ele deve se comportar. Os requisitos permitem com que desenvolvedores identifiquem possíveis problemas e limitações do sistema [Sommerville 2015].

2.2.1. Requisitos Funcionais

São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações e descreve tarefas e serviços que serão fornecidos pelo sistema aos seus usuários [Sommerville 2015].

2.2.2. Requisitos Não Funcionais

Os requisitos não funcionais são restrições aos serviços ou funções oferecidos pelo sistema, os quais descrevem limitações de ordem geral aos requisitos funcionais. Esses requisitos estabelecem níveis de serviço esperados para o funcionamento do software, como interoperabilidade, segurança, privacidade, sigilo, local, hardware alvo, padrões, entre outros [Lopes 2002]. Portanto, os requisitos não funcionais são importantes para uma especificação completa e ampla de software.

2.3. Análise de Requisitos

É uma atividade crítica, pois a falta de compreensão ou documentação adequada dos requisitos pode levar a erros de projeto, atrasos na entrega e insatisfação. É de importância para gerenciar expectativas e definir metas realistas para o software. Envolve a identificação, documentação e verificação das necessidades em relação ao sistema que está sendo desenvolvido.

Existem diversas abordagens e metodologias para a análise de requisitos, todas com objetivo de identificar e documentar os requisitos do sistema. A análise de requisitos é realizada por meio da comunicação e colaboração entre desenvolvedores e usuários finais, por meio de diversas técnicas [Sommerville 2015].

2.3.1. Etapas de Análise de Requisitos:

- **Identificação dos requisitos:** é feita a coleta informações sobre o que o cliente deseja que o software faça. Isso pode ser feito por meio de diversas técnicas.
- **Classificação dos requisitos:** É a parte de classificá-los em diferentes categorias. Isso pode ser feito com base na sua importância, na sua prioridade ou em outros critérios relevantes.
- **Análise dos requisitos:** Analisa-se cada requisito individualmente, verificando se ele é viável do ponto de vista técnico e se atende às necessidades do cliente.
- **Especificação dos requisitos:** Depois de analisar os requisitos, é preciso documentá-los de forma clara e precisa. Isso pode ser feito por meio de documentos ou outras técnicas.
- **Validação dos requisitos:** por fim, é preciso validar os requisitos junto ao cliente, para garantir que eles atendem às suas necessidades e expectativas. Isso pode ser feito por meio de protótipos, testes ou outras técnicas de validação.

2.4. Técnicas

Nesta seção, será abordado, três técnicas comumente usadas na análise de requisitos de software, essas técnicas são discutidas no decorrer do artigo.

2.4.1. Fluxograma

É um diagrama que descreve um processo, sistema ou algoritmo de computador. É amplamente utilizados na área de Análise para documentar, estudar, planejar, melhorar e comunicar processos complexos por meio de diagramas claros e fáceis de entender.

2.4.2. Análise de lacunas

É usado para avaliar o desempenho real face ao potencial, através de comparações entre uma lacuna de desempenho e a futura.

2.4.3. Prototipagem

Prototipar é criar uma versão modelo do produto final de um projeto voltado para a etapa de aprovação e validação.

3. Aplicação Prática

Os requisitos são a chave para o sucesso ou o fracasso de um projeto de software. Com isso, nesta seção, veremos aplicações praticas e estudos de casos sobre aAnálise de requisitos de software.

3.1. Método para Análise de Sistemas de Conhecimento, Inspirado no Princípio da Complementariedade de Niels Bohr

O texto de [Selner 1999] aborda a importância dos requisitos para o sucesso de um projeto de software. Na seção do artigo de [Selner 1999] sobre boas práticas de análise de requisitos, é destacada a importância da representação variada dos requisitos para identificar problemas de maneiras diferentes. Em seguida, são apresentados três estudos de

caso. O primeiro destaca a diferença entre software de folha de pagamento e software de recursos humanos, enfatizando a importância dos relatórios legais.

O segundo estudo de caso aborda a falta de controle na empresa "Netville" e como problemas foram identificados sem que o cliente falasse diretamente com o analista. O terceiro caso envolve a análise de requisitos para uma empresa de serviços de usinagem chamada "Vama". O texto também menciona o processo de levantamento de dados e análise documental em cada um dos casos. Por fim, é proposta a criação de um arcabouço interdisciplinar e um modelo de análise de requisitos mais adequado para sistemas de informações em organizações sociais.

3.2. Aplicação de um *survey* para Identificação de Problemas no Processo de Coleta e Análise de Requisitos de Sistema

Trata-se de um estudo [Belusso et al. 2018] que tem como objetivo avaliar o impacto do processo de análise de requisitos no desenvolvimento de sistemas em cenários reais, verificando a ocorrência de requisitos mal especificados ou mal interpretados. Para isso, foram coletados relatos de profissionais que trabalham em empresas de desenvolvimento de software, a fim de identificar os problemas encontrados nessa etapa.

Inicialmente, foi realizada uma pesquisa utilizando questionários e entrevistas como ferramentas de investigação *survey* Figura 1. O questionário foi elaborado em colaboração com especialistas em engenharia de requisitos e direcionado a analistas de requisitos de diferentes empresas de desenvolvimento de software. A pesquisa envolveu também a realização de entrevistas com base nas respostas coletadas no questionário, utilizando um modelo semi-estruturado.

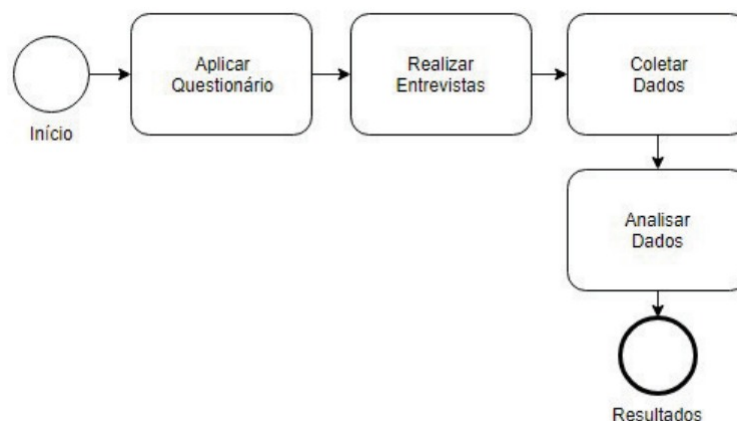


Figura 1. Esquematização da pesquisa de trabalho. Retirado de: [Belusso et al. 2018].

Nesta etapa inicial da pesquisa [Belusso et al. 2018] tem como objetivo avaliar o impacto do processo de análise de requisitos no desenvolvimento de sistemas em cenários reais, verificando a ocorrência de requisitos mal especificados ou mal interpretados. Para isso, foram coletados relatos de profissionais que trabalham em empresas de desenvolvimento de software, a fim de identificar os problemas encontrados nessa etapa.

A segunda etapa da pesquisa envolveu a realização de entrevistas em plataformas online com os profissionais, por meio das respostas coletadas no questionário. Para a

realização das entrevistas, foi utilizado um modelo semi-estruturado, definindo perguntas essenciais para todas as empresas e, durante a entrevista, novas perguntas foram exploradas conforme a necessidade do entrevistador.

Os resultados da pesquisa incluem respostas de profissionais de empresas de diferentes portes que desenvolvem sistemas *desktop*, *web* e *mobile*. A coleta de requisitos é feita por meio de entrevistas, protótipos e observação. A maioria dos participantes relata problemas com requisitos mal especificados ou mal interpretados, resultando em retrabalho e atrasos na entrega dos projetos.

Com esse estudo de [Belusso et al. 2018], foi visto problemas relatados pelos participantes da pesquisa através de questionário e entrevistas, como relatos de problemas nas entrevistas que não foram registrados no questionário. Os resultados indicam frequentes problemas com requisitos mal especificados ou mal interpretados, gerando retrabalho e impactando custos e prazos de entregas. O estudo descreve como o processo de análise de requisitos afeta o produto final do ponto de vista da empresa e do cliente.

4. Ferramentas

De acordo com [Wiegers and Beatty 2013], as ferramentas de análise de requisitos são importantes no desenvolvimento de software para verificar, documentar e construir as necessidades dos clientes e usuários finais no produto final. Elas permitem que as equipes mapeiem requisitos funcionais e não funcionais, gerenciem e rastreiem necessidades ao longo do ciclo de vida do projeto, e forneçam documentos claros e precisos para comunicá-los a outras partes interessadas no projeto, o que proporciona uma visão mais tangível do produto como um todo.

4.1. JIRA

- *Open Source*: Não
- Área de Contribuição: Gestão de Projetos

O Jira é uma ferramenta que ao longo dos anos evoluiu. Nos anos iniciais de seu lançamento, seu maior ponto forte era ser utilizado em conjunto com outras ferramentas. No entanto, com o tempo e grandes atualizações, o Jira se tornou uma ferramenta completa, com a possibilidade de os integrantes do projeto se comunicarem entre si por meio de uma interface fluida e inovadora. Seu preço varia conforme o tamanho da equipe [Alexander 2019].

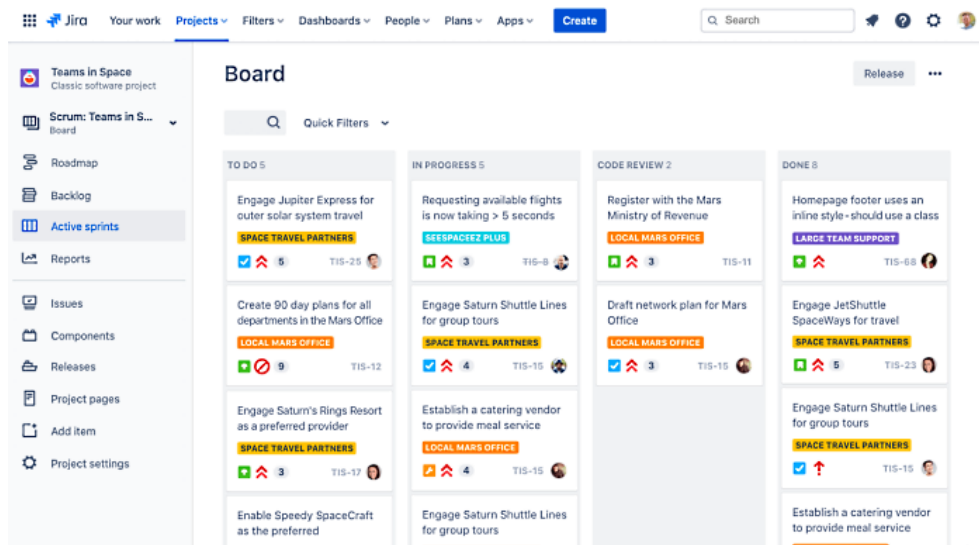


Figura 2. Interface Jira [Santos 2023]

4.2. Aha!

- *Open Source*: Possui uma versão gratuita e outras pagas
- Área de Contribuição: Gestão, planejamento e gerenciamento

O Aha! é um software de gestão de produto que permite que as equipes capturem ideias, priorizem recursos, planejem roadmaps e gerenciem lançamentos em uma plataforma integrada. Ele ajuda a alinhar estratégias de negócios e objetivos, colaborar em tempo real e manter todas as partes atualizadas sobre o progresso do produto final. O software também oferece recursos para análise de concorrência, rastreamento de problemas e bugs, gerenciamento de documentos e outras funcionalidades [capterra 2023].

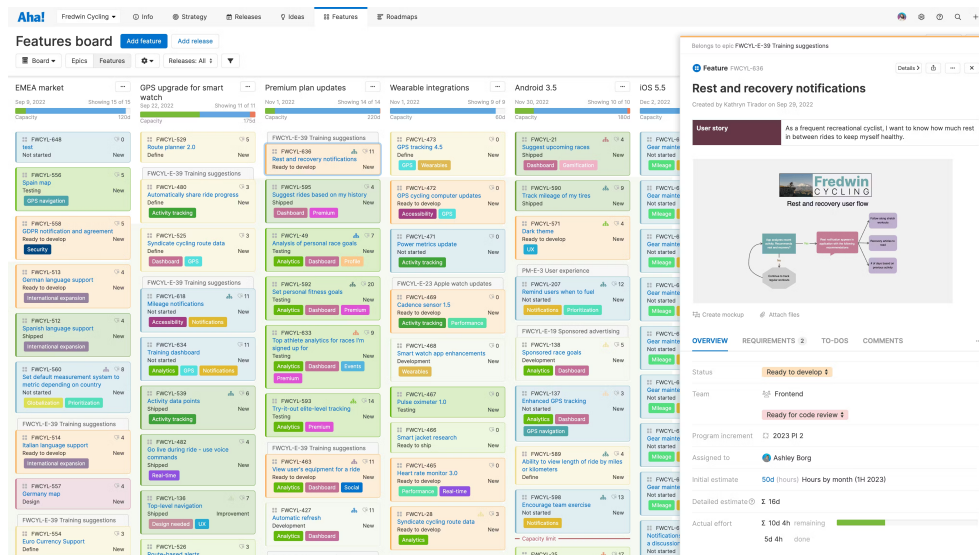


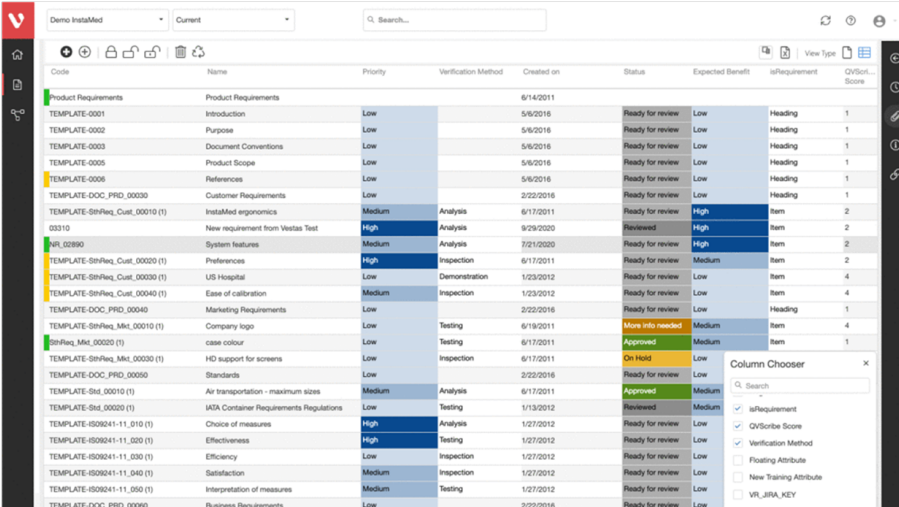
Figura 3. Interface Aha!

4.3. Visure Requirements

- *Open Source*: Não

- Área de Contribuição: Gerenciamento de Requisitos

Visure Requirements é uma ferramenta avançada de Engenharia de Requisitos da *Visure Solutions* que oferece suporte completo ao processo de requisitos, incluindo captura, análise, especificação, validação, verificação, rastreabilidade, gerenciamento e reutilização. Ela permite colaboração em equipe e melhoria da qualidade dos produtos de software, possuindo integrações com outras ferramentas e recursos como rastreabilidade e controle de acesso baseado em funções e relatórios [Visure 2023].



Code	Name	Priority	Verification Method	Created on	Status	Expected Benefit	isRequirement	QVSort Score
Product Requirements	Product Requirements			6/14/2011				
TEMPLATE-0001	Introduction	Low		5/5/2016	Ready for review	Low	Heading	1
TEMPLATE-0002	Purpose	Low		5/5/2016	Ready for review	Low	Heading	1
TEMPLATE-0003	Document Conventions	Low		5/5/2016	Ready for review	Low	Heading	1
TEMPLATE-0005	Product Scope	Low		5/5/2016	Ready for review	Low	Heading	1
TEMPLATE-0006	References	Low		5/5/2016	Ready for review	Low	Heading	1
TEMPLATE-DOC_PRD_00030	Customer Requirements	Low		2/22/2016	Ready for review	Low	Heading	1
TEMPLATE-ShiReq_Cust_00010 (1)	InstaMed ergonomics	Medium	Analysis	6/17/2011	Ready for review	High	Item	2
03310	New requirement from Vestas Test	High	Analysis	9/29/2020	Reviewed	High	Item	2
NRI_02890	System features	Medium	Analysis	7/21/2020	Ready for review	High	Item	2
TEMPLATE-ShiReq_Cust_00020 (1)	Preferences	High	Inspection	6/17/2011	Ready for review	Medium	Item	2
TEMPLATE-ShiReq_Cust_00030 (1)	US Hospital	Low	Demonstration	1/23/2012	Ready for review	Low	Item	4
TEMPLATE-ShiReq_Cust_00040 (1)	Ease of calibration	Medium	Inspection	1/23/2012	Ready for review	Low	Item	4
TEMPLATE-DOC_PRD_00040	Marketing Requirements	Low		2/22/2016	Ready for review	Low	Heading	1
TEMPLATE-ShiReq_Mkt_00010 (1)	Company logo	Low	Testing	6/18/2011	More info needed	Medium	Item	4
ShiReq_Mkt_00020 (1)	case colour	Low	Testing	6/17/2011	Approved	Medium	Item	1
TEMPLATE-ShiReq_Mkt_00030 (1)	HD support for screens	Low	Inspection	6/17/2011	On Hold	Low		
TEMPLATE-DOC_PRD_00050	Standards	Low		2/22/2016	Ready for review	Low		
TEMPLATE-Shd_00010 (1)	Air transportation - maximum sizes	Medium	Analysis	6/17/2011	Approved	Medium		
TEMPLATE-Shd_00020 (1)	ATA Container Requirements Regulations	Low	Testing	1/13/2012	Reviewed	Medium		
TEMPLATE-IS09241-11_010 (1)	Choice of measures	High	Analysis	1/27/2012	Ready for review	Low		
TEMPLATE-IS09241-11_020 (1)	Effectiveness	High	Testing	1/27/2012	Ready for review	Low		
TEMPLATE-IS09241-11_030 (1)	Efficiency	Low	Inspection	1/27/2012	Ready for review	Low		
TEMPLATE-IS09241-11_040 (1)	Satisfaction	Medium	Inspection	1/27/2012	Ready for review	Low		
TEMPLATE-IS09241-11_050 (1)	Interpretation of measures	Medium	Testing	1/27/2012	Ready for review	Low		
TEMPLATE-DOC_PRD_00060	Business Requirements	Low		2/22/2016	Ready for review	Low		

Figura 4. Interface Visure

4.4. Figma

- *Open Source*: Possui uma versão gratuita e outra paga
- Área de Contribuição: Prototipagem

O Figma é uma plataforma inovadora de prototipagem que permite às empresas criar designs personalizados e compartilhar protótipos entre os membros da equipe. Usando tecnologia em nuvem e local, essa solução fornece recursos avançados de colaboração, registro de atividades e análise/geração de relatório. Com a ajuda do Figma, empresas de todos os tamanhos podem criar designs de internet sob medida para suas necessidades, o que possibilita maior eficiência e resultados mais positivos. Além disso, o fato de não ser necessária a instalação de um software é sensacional. Além disso, o armazenamento em nuvem é muito útil e funcional [capterra 2023].

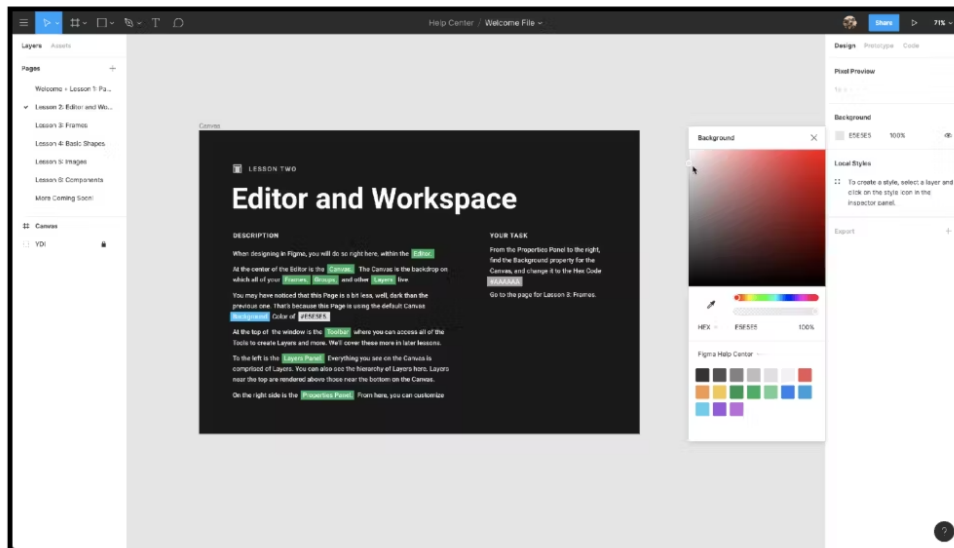


Figura 5. Interface Figma [capterra 2023]

5. Vantagens e Desvantagens

Pressman [Pressman 2019] enfatiza que a compreensão das necessidades do cliente é o primeiro passo crítico no processo de obtenção de requisitos de software. Podemos destacar a importância desse processo no desenvolvimento de um software de qualidade.

Neste contexto, serão discutidas aqui as vantagens e desvantagens dos métodos utilizados na análise de requisitos. Com isso, será possível escolher a técnica mais adequada para cada projeto, garantindo uma análise precisa e completa, fator essencial para o sucesso do desenvolvimento de software.

5.1. Business Process Model and Notation (BPMN):

Vantagens:

- O BPMN é usado para criar modelos gráficos que facilitam a compreensão dos processos de desenvolvimento, tornando-os claros e visualmente compreensíveis. Isso ajuda os envolvidos no projeto a entender o fluxo de trabalho, as atividades e as interações entre stakeholders, incluindo desenvolvedores de software [White 2011]; criação de modelos BPMN identifica lacunas e inconsistências em processos de negócio existentes ou propostos. A visualização dos fluxos de trabalho destaca falhas, redundâncias, atividades desnecessárias ou ausentes, permitindo identificar oportunidades de melhoria e corrigir problemas antes da implementação do software [White 2011]; BPMN é uma linguagem comum que ajuda na comunicação entre stakeholders, como usuários finais, analistas de negócios e desenvolvedores de software. Isso melhora a colaboração e alinha as expectativas entre as partes interessadas [White 2011];
- Modelos BPMN são geralmente fáceis de atualizar, o que facilita a manutenção e a evolução dos requisitos à medida que o projeto avança [White 2011].

Desvantagens:

- O uso do BPMN pode exigir uma curva de aprendizado significativa para analistas e outros stakeholders não familiarizados com a notação. É necessário dominar ferramentas específicas para criação e manipulação de modelos BPMN, exigindo habilidades técnicas adicionais [White 2011];
- O BPMN é focado em representar a lógica de negócio e o fluxo de trabalho, e pode não ser adequado para representar detalhes técnicos específicos de implementação do software, como estruturas de dados, integrações com sistemas externos ou requisitos de desempenho [White 2011].

5.2. Flowcharts ou fluxogramas:

Vantagens:

- As flowcharts são uma representação visual clara dos fluxos de atividades, facilitando a compreensão do fluxo de trabalho do sistema por todos os stakeholders [Jeffrey A. Hoffer 2013];
- Flowcharts ajudam a detectar erros e inconsistências no fluxo de trabalho de um sistema, economizando tempo e recursos na correção de problemas futuros [Jeffrey A. Hoffer 2013];
- São facilmente atualizáveis, permitindo que os analistas de requisitos façam modificações nas atividades ou na lógica de controle do processo de forma rápida e eficiente [Jeffrey A. Hoffer 2013].

Desvantagens:

- Podem se tornar complexas e difíceis de entender quando o processo analisado é muito complexo ou possui muitas ramificações e condições de controle [Jeffrey A. Hoffer 2013];
- Flowcharts não detalham requisitos técnicos e são mais adequadas para representações de alto nível do fluxo de trabalho, complementadas por outras técnicas para detalhar os requisitos de software [Jeffrey A. Hoffer 2013];
- Embora sejam relativamente simples de entender, pode haver uma curva de aprendizado inicial para os membros da equipe que não estão familiarizados com essa técnica [Jeffrey A. Hoffer 2013].

5.3. Análise de Lacunas:

Vantagens:

- Identificar lacunas entre o produto/sistema atual e os requisitos desejados ajuda a entender deficiências e áreas de melhoria, permitindo decisões informadas para atender aos requisitos de software [van Lamsweerde 2009];
- Permite a definição de metas claras e a criação de planos de ação para preencher as lacunas identificadas, melhorando assim o desempenho do produto ou sistema [van Lamsweerde 2009].

Desvantagens:

- Identificar lacunas entre o produto/sistema atual e os requisitos desejados é desafiador, especialmente com requisitos complexos ou mal definidos. Determinar a extensão das lacunas e áreas de melhoria pode ser difícil [van Lamsweerde 2009];
- A Análise de Lacunas identifica lacunas, mas não oferece soluções detalhadas para preenchê-las. É necessário utilizar outras técnicas para desenvolver soluções específicas e atender aos requisitos de software identificados [van Lamsweerde 2009].

5.4. Prototipagem:

Vantagens:

- A prototipagem permite criar modelos tangíveis ou interativos do sistema, o que pode ajudar na comunicação com as partes interessadas, como clientes, usuários finais e desenvolvedores [Warfel 2009];
- Protótipos coletam feedback rápido das partes interessadas sobre o sistema em desenvolvimento, ajudando a identificar problemas e melhorias necessárias nos requisitos antes da fase de implementação [Warfel 2009];
- A prototipagem envolve as partes interessadas no processo de análise de requisitos, permitindo que visualizem e interajam com o sistema em desenvolvimento, fornecendo feedback valioso [Warfel 2009].

Desvantagens:

- Protótipos podem ter limitações técnicas, como falta de funcionalidades completas, desempenho inferior ou falta de integração com outros sistemas, levando a interpretações incorretas ou incompletas dos requisitos [Warfel 2009];
- Protótipos podem levar a mudanças frequentes e desvios do escopo original do projeto, pois as partes interessadas podem se concentrar mais neles do que nos requisitos reais, criando desafios na gestão de escopo [Warfel 2009].

6. Desafios e Tendências

Uma análise de requisitos completa e eficaz pode melhorar a comunicação e colaboração entre as equipes, alinhar o software com as necessidades do negócio, aumentar a satisfação do usuário e melhorar a qualidade do produto final. No entanto, para garantir que os requisitos sejam precisos, completos e atendam às necessidades dos usuários finais, é importante considerar os desafios e tendências que podem surgir durante o processo de análise de requisitos [Visure 2023]. Alguns desses desafios e tendências incluem:

6.1. Desafios

- **Inteligência artificial:** O desafio consiste em treinar a IA com base em análises anteriores para estabelecer um planejamento o mais realista possível. Isso envolve determinar os custos do projeto, as demandas, as ferramentas necessárias e estipular prazos de entrega [Sky.One 2022].
- **Mudança acelerada dos requisitos:** Essa mudança pode resultar em dificuldades para gerenciá-los de forma eficaz, tornando um desafio equilibrar flexibilidade com disciplina.
- **Compreensão dos requisitos:** Muitas vezes, os stakeholders têm dificuldade em compreender completamente os requisitos, o que pode levar a especificações inadequadas ou incompletas [Monitoratec 2021].
- **Conflitos de requisitos:** Diferentes stakeholders podem ter requisitos conflitantes, o que torna desafiador chegar a um consenso sobre a especificação final [Monitoratec 2021].
- **Ambiguidade:** Os requisitos podem apresentar ambiguidades e serem abertos a interpretações diversas, o que pode levar a entendimentos e resultados diferentes [Visure 2023].

- **Mitigação de riscos:** Em qualquer projeto, existem riscos que devem ser gerenciados. Requisitos mal definidos ou estratégias inadequadas de gerenciamento de riscos podem resultar em retrabalho dispendioso e atrasos no cronograma do projeto [Visure 2023].
- **Tempo e orçamento:** Manter o cronograma planejado e cumprir os prazos é um grande desafio, uma vez que imprevistos podem surgir e causar atrasos e aumento de custos. Não cumprir os prazos estabelecidos pode levar à desconfiança e perda de credibilidade por parte do cliente, o que acarreta em consequências negativas significativas [Visure 2023].
- **Seleção de ferramentas adequadas:** Existem diversas ferramentas disponíveis para auxiliar na análise de requisitos de software, porém escolher a ferramenta adequada pode representar um desafio. É importante considerar os requisitos específicos do projeto e as necessidades da equipe de desenvolvimento [e Chris Rupp 2011].

6.2. Tendências

- **Inteligência artificial:** A aplicação de técnicas de inteligência artificial pode ser benéfica na identificação de requisitos ocultos ou implícitos, na análise de grandes quantidades de dados e na automação de tarefas, melhorando a eficiência do processo. Entretanto, é fundamental lembrar que essa abordagem ainda está em fase de desenvolvimento e requer a atuação de profissionais altamente capacitados e especializados [Maiden 2020].
- **Low-code:** Essa solução possibilita a criação de aplicativos com uma codificação reduzida, simplificando o processo de criação do programa e permitindo que os desenvolvedores dediquem seu tempo em tarefas mais estratégicas. Com isso, outras pessoas do projeto podem colocar um MVP em funcionamento de maneira rápida, tornando possível a criação de soluções de forma ágil e eficiente [Sky.One 2022].
- **Internet das coisas (IoT):** Essa solução, além de colher informações de equipamentos, analisa o comportamento dos usuários conforme sua interação [Sky.One 2022].
- **Utilização de técnicas ágeis:** As metodologias ágeis têm sido cada vez mais utilizadas na análise de requisitos, permitindo uma maior flexibilidade e adaptação às mudanças nos requisitos [ACerT 2022].
- **Interação com o usuário:** A interação com o usuário final tem sido cada vez mais valorizada na análise de requisitos, permitindo uma melhor compreensão das necessidades do usuário e reduzindo a chance de erros na especificação [ACerT 2022].
- **Automação:** A automação de processos de análise de requisitos pode aumentar a eficiência e a precisão, reduzindo a chance de erros humanos [ACerT 2022].
- **Análise de requisitos em tempo real:** A análise de requisitos em tempo real permite que os requisitos sejam alterados e adaptados rapidamente às mudanças nas necessidades do usuário final [ACerT 2022].
- **Segurança e privacidade:** Com a crescente sofisticação das ameaças cibernéticas, é essencial que os requisitos de segurança sejam identificados, documentados e validados para garantir que o software seja projetado e construído com as medidas de segurança adequadas [Anderson 2008].

7. Considerações Finais

Em conclusão, a coleta e análise corretas de requisitos são cruciais para o sucesso dos projetos de desenvolvimento de software. Os exemplos apresentados neste artigo destacam como a coleta e análise inadequadas de requisitos podem resultar em atrasos significativos, retrabalho e custos adicionais. É fundamental envolver as partes interessadas no processo de coleta de requisitos e utilizar técnicas e ferramentas apropriadas para análise.

Além disso, a comunicação e colaboração contínuas entre a equipe de desenvolvimento e as partes interessadas são essenciais para garantir a compreensão e o atendimento adequados dos requisitos.

Com base nos exemplos apresentados neste artigo, fica evidente que o processo de coleta e análise de requisitos não deve ser negligenciado. É necessário direcionar mais pesquisas e desenvolvimento para aprimorar as ferramentas e técnicas utilizadas nessa fase crítica do desenvolvimento de software. Além disso, as organizações devem investir em treinamento e educação para suas equipes de desenvolvimento em relação às técnicas adequadas de coleta e análise de requisitos.

Portanto, diante dos fatos expostos neste artigo, torna-se crucial reconhecer a importância da comunicação e colaboração contínuas entre a equipe de desenvolvimento e as partes interessadas ao longo do ciclo de vida do projeto.

Referências

- ACerT, W. (2022). O que é análise de requisitos de software? [urlhttps://acertbr.com.br/analise-de-requisitos-de-software-quais-os-desafios-e-como-fazer/](https://acertbr.com.br/analise-de-requisitos-de-software-quais-os-desafios-e-como-fazer/).
- Alexander, M. (2019). 5 principais ferramentas de gerenciamento de requisitos. [urlhttps://itforum.com.br/noticias/5-principais-ferramentas-de-gerenciamento-de-requisitos/](https://itforum.com.br/noticias/5-principais-ferramentas-de-gerenciamento-de-requisitos/).
- Anderson, R. J. (2008). *Security Engineering: A Guide to Building Dependable Distributed Systems*.
- Belusso, L., Machado, P., Ribeiro, F., and Schimit, J. (2018). Aplicação de uma survey para identificação de problemas no processo de coleta e análise de requisitos de sistema. In *Anais da II Escola Regional de Engenharia de Software*, pages 17–24. SBC.
- capterra (2023). capterra. [urlhttps://www.capterra.com.br/company/about](https://www.capterra.com.br/company/about).
- e Chris Rupp, K. P. (2011). *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam – Foundation Level – IREB compliant*.
- Jeffrey A. Hoffer, J. F. G. e. J. V. (2013). *Modern Systems Analysis and Design*.
- Júnior, M. R. D. C. (2012). Gestão de projetos: da academia à sociedade. *InterSaberes*, 1.
- Lopes, P. S. N. D. (2002). *Uma taxonomia da pesquisa na área de Engenharia de Requisitos*. PhD thesis, Citeseer.

Maiden, N. (2020). *Requirements Engineering for AI Systems*.

Monitoratec (2021). Análise de requisitos de software: quais os desafios e como fazer?
url<https://www.monitoratec.com.br/blog/analise-de-requisitos-de-software/>.

Pressman, R. (2019). *Software engineering a practitioner's approach*.

Santos, J. M. D. (2023). Jirainterace23. url<https://project-management.com/jira-software-review/>.

Selner, C. (1999). Análise de requisitos para sistemas de informações, utilizando as ferramentas da qualidade e processos de software.

Sky.One, E. (2022). 7 tendências de desenvolvimento de software.
url<https://skyone.solutions/hub/tendencias-desenvolvimento-software/>.

Sommerville, I. (2015). *Software engineering*.

sotille, M. (2004). *Gerenciamento de Projetos da Engenharia de Software*. PhD thesis, CiteSeer.

van Lamsweerde, A. (2009). *Requirements Engineering: From System Goals to UML Models to Software Specifications*.

Visure (2023). Visure. urlhttps://docs.google.com/document/d/1ok3MCYTqgAV6gi6e48C0wJQR0_VY2a.

Warfel, T. Z. (2009). *Prototyping: A Practitioner's Guide*.

White, S. A. (2011). *BPMN 2.0 Handbook Second Edition*.

Wieggers, K. and Beatty, J. (2013). *Software Requirements*.