

**UNIVERSIDADE FEDERAL DO AMAPÁ**

**DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**

**COORDENAÇÃO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO**

LEONARDO DE ANDRADE OLIVEIRA

*A importância da rastreabilidade de requisitos de software para garantir a qualidade do produto final*

Prof. Dr. Júlio Cezar Costa Furtado

Orientador

Macapá, Maio de 2023

**SUMÁRIO**

[1 INTRODUÇÃO 5](#_Toc134817344)

[2 CONTEXTUALIZAÇÃO E TERMINOLOGIAS DO TRABALHO 7](#_Toc134817345)

[2.1. Engenharia de Requisitos 7](#_Toc134817346)

[2.1.1. O que é a Engenharia de Requisitos 7](#_Toc134817347)

[2.1.2. Atividades da Engenharia de Requisitos 8](#_Toc134817348)

[2.1.2.1. Elicitação e análise 8](#_Toc134817349)

[2.1.2.2. Especificação 9](#_Toc134817350)

[2.1.2.3. Validação 9](#_Toc134817351)

[2.1.3. Gerenciamento de Requisitos 10](#_Toc134817352)

[2.2.1 Trabalhos Relacionados 11](#_Toc134817353)

**LISTA DE ABREVIATURAS E SIGLAS**

IEEE *Institute of Electrical and Electronics Engineers*

ISO *International Organization for Standardization* (Organização Internacional de Normalização)

RSL Revisão Sistemática da Literatura

TCC Trabalho de Conclusão de Curso

**LISTA DE FIGURAS**

[Figura 1. Processo da Elicitação **Erro! Indicador não definido.**](#_Toc68635270)

[Figura 2. Notação para escrever requisitos do sistema **Erro! Indicador não definido.**](#_Toc68635270)

1 INTRODUÇÃO

2 CONTEXTUALIZAÇÃO E TERMINOLOGIAS DO TRABALHO

2.1. Engenharia de Requisitos

2.1.1. O que é a Engenharia de Requisitos

Sommerville (2019), considera que, de modo geral, um requisito, dentro do escopo global da engenharia de software, é uma relação entre os serviços ou as tarefas que um sistema se propõe a fazer e suas restrições a sua operação. Requisitos, exprimem aquilo que os clientes necessitam dentro de um sistema de software, que atendem a um determinado propósito; pode-se citar exemplos como demandar de um recurso do sistema ou controlar um dispositivo. Esse conceito também pode significar em alguns contextos somente a descrição documentada ou não de um serviço de um sistema de software ou de uma restrição sem um detalhamento preciso, ou seja, apenas uma declaração abstrata de ambos (SOMMERVILLE, 2019).

A Engenharia de Requisitos, por sua vez, é considerada uma disciplina da Engenharia de Software que compõe o uso pautado e contínua de técnicas para cobrir atividades de obtenção, documentação e manutenção de um conjunto de requisitos para software que atendam aos objetivos de negócio e sejam de qualidade (VAZQUEZ *et al.*, 2016).

Assim, para Sommerville (2019), a Engenharia de Requisitos pode ser considerada o primeiro estágio do processo de engenharia de software, porém em alguns casos se faz necessário produzir algum nível de detalhamento e compreensão dos requisitos dos sistemas antes mesmo de adquirir ou desenvolver um sistema, quase como um estudo de viabilidade, pois esses requisitos previamente verificados podem indicar se o produto é financeiramente e tecnicamente viável (SOMMERVILLE, 2019).

Para VAZQUEZ *et al*. (2016), sugere que as atividades de requisitos sejam distribuídas durante todo o processo de desenvolvimento do sistema, dando maior atenção em um primeiro momento no entendimento dos objetivos para o desenvolvimento e de suas restrições.

Para Pressman (2006), não há maneira inquestionável de garantir que a especificação construída pela construção de requisitos se ajuste completamente às necessidades dos *stakeholders*, e que seja capaz de atender às suas necessidades. Este é certamente o desafio experimentado na construção de requisitos, e para isso um conjunto de atividades deve ser construído e conduzido ao longo de um programa de extensão.

2.1.2. Atividades da Engenharia de Requisitos

Segundo Sommerville (2019), as atividades da Engenharia de Requisitos são compostas de três principais etapas, são elas: **elicitação** **e análise**, **especificação** e por fim **validação**, sendo que nos dois primeiros estágios ocorre a descoberta de requisitos pela interação com os *stakeholders*, onde é normalmente esperado um maior esforço em questão de entendimento das regras de negócio de alto nível, requisitos não funcionais e requisitos de usuário para o sistema, porém, isso depende de alguns fatores como: o estágio do processo, o tipo de sistema a ser desenvolvido e o orçamento disponível.

2.1.2.1. Elicitação e análise

No dicionário Aurélio (2020), o termo elicitar pode significar: fazer sair; lançar para fora, expulsar; ou por extensão: causar o aparecimento de; fazer com que algo passe a existir, em complementar, para Sommerville (2019), a elicitação dentro do contexto da engenharia de software, é buscar e descobrir requisitos, trazendo à tona o que de fato o produto de software é composto.

O processo de Elicitação e análise de requisitos tem como objetivo entender as funções que os *stakeholders* realizam juntamente a esclarecer o propósito do produto final e como esse produto, entende-se como o software, auxiliará nas atividades e consequentemente no seu trabalho como *stakeholders.* Durante a coleta de requisitos, os engenheiros de software trabalham com as partes interessadas para aprender mais sobre os domínios e atividades do produto. Conforme é mostrado na figura abaixo (SOMMERVILLE, 2019):

Figura 1. Processo da elicitação (Engenharia de Software, 2019, Sommerville).

Diagrama

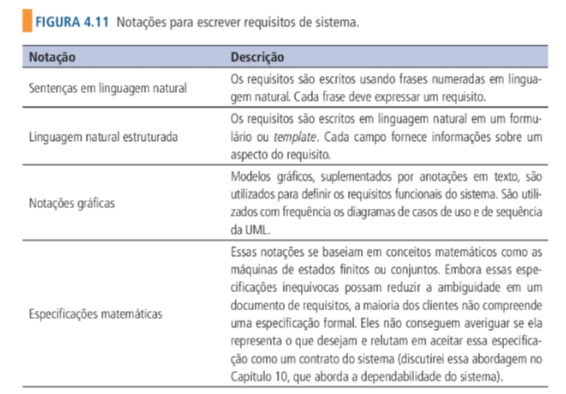
Descrição gerada automaticamente

VAZQUEZ *et al*. (2016), afirma que na análise de requisitos é trabalhado o que foi descoberto na elicitação, procurando dar sentido e forma aos requisitos. O objetivo nesta etapa então, é o de aumentar o entendimento atual da informação e revisar, ou seja, refinar e completar os requisitos descobertos na etapa anterior, essas descobertas são os insumos para a análise e de certa forma se busca revisitar o que foi obtido, mas sem se preocupar em valida-los.

2.1.2.2. Especificação

Na especificação de requisitos o processo pode ser visto como a documentação inicial dos requisitos do usuário e do sistema em um documento de requisitos. Os requisitos do usuário são na maior parte do tempo, escritos em linguagem natural complementada por diagramas e tabelas apropriados no documento de requisitos. Os requisitos do sistema também podem ser escritos em linguagem natural, mas outras notações baseadas em formulários, modelos de sistema gráficos ou matemáticos também podem ser usados. Conforme mostra na figura 2 demonstra os possíveis cenários para a escrita de requisitos (SOMMERVILLE, 2019):

Figura 2. Notação para escrever requisitos do sistema (Engenharia de Software, 2019, Sommerville).



2.1.2.3. Validação

Segundo Sommerville (2019), o processo de validação se dá ao verificar se os requisitos de fato representam o que o cliente deseja, além de tentar encontrar possíveis falhas no entendimento ou outros problemas que venham à tona, como inviabilidade de um ou mais requisitos ou inconsistências. Sommerville (2019) também defini as verificações a serem feitas nesta etapa:

“1. Verificações de validade: estas verificam se os requisitos refletem as necessidades reais do sistema usuários. Devido à mudança de circunstâncias, os requisitos do usuário podem ter alteradas desde que foram originalmente eliciadas.

2. Verificações de consistência: os requisitos do documento não devem entrar em conflito. Aquilo é, não deve haver restrições contraditórias ou descrições diferentes da mesma função do sistema.

3. Verificações de integridade: o documento de requisitos deve incluir requisitos que definem todas as funções e as restrições pretendidas pelo usuário do sistema.

4. Verificações de realismo: usando o conhecimento das tecnologias existentes, os requisitos devem ser verificados para garantir que possam ser implementados dentro do orçamento do sistema. Estas verificações devem também ter em conta o orçamento e cronograma para o desenvolvimento do sistema.

5. Verificabilidade: para reduzir o potencial de disputa entre cliente e contratado,

os requisitos do sistema devem sempre ser escritos de forma que sejam verificáveis.

Isso significa que você deve ser capaz de escrever um conjunto de testes que possam demonstrar que o sistema entregue atende a cada requisito especificado.”

De acordo com o autor, esta etapa é importante, pois pode evitar custos de operação e desenvolvimento devido a falhas nas etapas anteriores o que pode prejudicar a entrega do software final, embora não existam garantias de que neste processo não será alterado algum requisito ou seja feita uma perfeita validação avaliando todos os possíveis problemas, por isso é importante mitiga-los na medida do possível.

2.1.3. Gerenciamento de Requisitos

O Gerenciamento de Requisitos de Software (GRS) tem sido amplamente reconhecido como um dos principais elementos do processo de desenvolvimento de software. O GRS é a atividade que garante que os requisitos especificados pelos usuários sejam explicitamente documentados, analisados, gerenciados, validados e verificados, de acordo com as necessidades do cliente (FURTADO, 2016). De acordo com (CAVICHIOLLI, 2010), o GRS tem como objetivo principal permitir que o projeto de software seja implementado de forma eficaz, criando um elo entre as necessidades dos usuários, as características do software e a funcionalidade oferecida. A abordagem de GRS também procura prevenir mudanças imprevistas e inconsistências em relação ao produto final, devido à compreensão incompleta dos requisitos.

Ao lado da abordagem de GRS, a modelagem de requisitos tem sido estabelecida como uma abordagem de design de software efetiva. A modelagem de requisitos é um processo que visa capturar, analisar, modelar e validar requisitos de software relevantes para a implementação de um projeto de software (SILVA et al., 2018). O principal objetivo da modelagem de requisitos é identificar as necessidades dos usuários e descrever as características do software necessárias para atender a essas necessidades.

Para VAZQUEZ *et al*. (2016) gerir requisitos é administrar as informações conhecidas do software, gerindo as mudanças do sistema, de modo que estas mudanças podem ocorrer quando os clientes estão mais esclarecidos do negócio como um todo, dando maturidade a redefinição de um requisito, para os autores manter a documentação dos requisitos durante toda a fase do projeto pode auxiliar no gerenciamento destas mudanças, embora não seja fundamental.

Ainda segundo VAZQUEZ *et al*. (2016) a forma como será gerenciado tais requisitos é definido pelo plano de gerenciamento de requisitos, que pode ser criado pela equipe de gestão de projetos na fase de planejamento do desenvolvimento.

Wiegers (2003), aponta que o processo de gerência de requisitos deve incluir as seguintes atividades: controle de mudanças, controle de versão, acompanhamento do estado dos requisitos e rastreamento de requisitos, mas Pressman (2010), diz que é fundamental gerenciar alterações nos requisitos acordados entre cliente e gestores, relacionamentos e interações entre requisitos e gerenciar dependências entre requisitos.

2.2.1 Trabalhos Relacionados

O primeiro autor a abordar a questão das ferramentas de captura de requisitos de software foi Pressman (2005). Segundo ele, as três principais ferramentas para capturar requisitos de software são entrevistas, documentos e protótipos. As entrevistas permitem a coleta de informações detalhadas do cliente e das partes interessadas, além de permitir a verificação de suposições e o entendimento do problema do cliente. Os documentos são uma forma de registrar formalmente os requisitos e compartilhá-los entre as partes interessadas. Por fim, os protótipos podem ser úteis para demonstrar a funcionalidade do software e validar as suposições do cliente.

Jacobson e outros. (1992) acrescentam à discussão de ferramentas para capturar requisitos de software introduzindo o conceito de casos de uso. Os casos de uso são uma forma de expressar os requisitos de maneira precisa e podem ser usados para garantir que as necessidades do cliente sejam atendidas. Além disso, eles podem ser usados para modelar o sistema do cliente e podem ser usados para gerar casos de teste.

Ambler (2002) sugere o uso de histórias de usuários como forma de capturar os requisitos do cliente. As histórias de usuários são descrições curtas dos requisitos do cliente, escritas em linguagem natural. Eles são uma forma de capturar requisitos de alto nível e podem ser usados para garantir que as necessidades do cliente sejam atendidas.