



UNIVERSIDADE ESTADUAL DE GOIÁS - UEG

WELTON LEITE DA PAIXÃO

**ANÁLISE DE REQUISITOS NO PROCESSO LEGISLATIVO DA CÂMARA
MUNICIPAL DE TRINDADE - GO**

TRINDADE - GO
NOVEMBRO DE 2023

WELTON LEITE DA PAIXÃO

**ANÁLISE DE REQUISITOS NO PROCESSO LEGISLATIVO DA CÂMARA
MUNICIPAL DE TRINDADE - GO**

Monografia final de conclusão de curso apresentada ao Colegiado de Ciência da Computação – UESB como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Área de concentração: Engenharia de Software

Orientador (a): Prof. Cláudio Garcia

TRINDADE - GOIÁS
NOVEMBRO DE 2023

WELTON LEITE DA PAIXÃO

**ANÁLISE DE REQUISITOS NO PROCESSO LEGISLATIVO DA CÂMARA
MUNICIPAL DE TRINDADE - GO**

Monografia aprovada em ____/____/____ para obtenção do título de Bacharel em
Sistema da Informação.

Comissão Examinadora

Prof. Cláudio Garcia
Universidade Estadual de Goiás

Prof. Cláudio Garcia
Universidade Estadual de Goiás

Prof. Cláudio Garcia
Universidade Estadual de Goiás

TRINDADE - GOIÁS
NOVEMBRO DE 2023

*Em memória dos meus pais e
a minha esposa e meus filhos
pela enorme força e paciência
durante este caminho e sempre.*

AGRADECIMENTOS

Aos professores Helber Lima Frazão, que foi um grande mentor, ao professor Olegário Correa Neto que, muito somou em minha jornada acadêmica, ao professor e mestre Fábio que é um referencial de carreira, postura e conhecimento, à professora Nágela que muito me ajudou, principalmente no começo acadêmico, ao mestre Paulo Cesar de Almeida que é uma fonte infinita de conhecimento, ao professor Cláudio Garcia, simplesmente o maior gênio que já conheci em minha vida. Professores extraordinários que me incentivaram e me fizeram lutar pelo melhor em minha vida, à vocês, minha eterna gratidão.

Seria impossível não mencionar minha esposa Sara, meus filhos Khallel e Heloá pelo apoio incondicional em meus estudos. Eu apenas posso mostrar o caminho aos meus filhos, para que possam seguir o meu exemplo de estudar e seguir no melhor caminho possível. Também ao amigo Alexandre Souza que foi a primeira pessoa que me motivou a iniciar uma vida de estudos acadêmicos, você mudou minha história, meu irmão de coração.

Aos amigos que conheci e muito me ajudaram na caminhada como o Lucas Henrique, Diogo Vicente, Denis Salles, Guilherme Daniel, Abílio, Ederlan, Mayke, e tantos outros que já não estão conosco devido o covid, como o grande amigo Lima. Às minhas irmãs na qual amo demasiadamente.

À Deus, pois ele é a minha força e minha inspiração, nos momentos mais difíceis em minha vida.

LISTAS DE FIGURAS

Figura 1 - Diagrama de planejamento	6
Figura 2 - Tabela de tempos para desenvolvimento	9
Figura 3 - Quadro de perguntas de usuário para escopo.....	10
Figura 4 - Evolução dos requisitos	21
Figura 5: Passos para a utilização da técnica de questionário	24
Figura 6 - Diagrama de casos de uso	25
Figura 7 - Descrição contínua.....	26
Figura 8: Caso de uso processar venda	27
Figura 8: Caso de uso processar venda (continuação)	28

LISTA DE GRÁFICOS

LISTA DE ABREVIATURAS

IEEE	<i>Institute of Electrical and Electronics Engineers</i>
DC	<i>Desenvolvimento Conhecido</i>
DNC	<i>Desenvolvimento não Conhecido</i>
MAN	<i>Manutenção de Sistemas</i>

RESUMO

SUMÁRIO

1 INTRODUÇÃO	11
2 A ENGENHARIA DE REQUISITOS	14
2.2 REQUISITOS	16
2.3 GERENCIAMENTO DE REQUISITOS	16
2.4 REQUISITO FUNCIONAL	17
2.6 DOCUMENTAÇÃO NO PROCESSO DE REQUISITOS	18
2.7 VALIDAÇÃO	20
2.8 CONTROLE DE QUALIDADE	20
2.8 GERENCIANDO OS REQUISITOS	21
3 DAS TÉCNICAS DE LEVANTAMENTO DE REQUISITOS	23
3.1 BRAINSTORM	23
3.2 ENTREVISTA	23
3.3 QUESTIONÁRIO	24
3.3.2 QUESTIONÁRIO NPS (NET PROMOTER SCORE) OBS	24
3.4 OBSERVAÇÃO DIRETA	24
3.5 ANÁLISE DOCUMENTAL	24
3.6 REUTILIZAÇÃO DE REQUISITOS	24
3.7 PROTOTIPAÇÃO	24
3.8 CASO DE USO	24
3.9 PONTO DE VISTA	28
3.10 ETNOGRAFIA	28
3.11 WORKSHOPS	28
3.12 JAD - JOINT APPLICATION DESIGN	28
4 METODOLOGIA	28
REFERÊNCIAS	30

1 INTRODUÇÃO

Com a grande expansão nos últimos anos de desenvolvimento de software tanto no Brasil quanto no mundo, é possível notar as mais variadas e diversas técnicas adotadas no levantamento de requisitos, “o mundo moderno não poderia existir sem o software” (SOMMERVILLE, 2013). Nesse quesito, a análise de requisitos e o seu levantamento é uma tarefa bastante essencial dentro do processo de criação de um software. Segundo Reinehr (2020, p. 13) “é inquestionável a presença da tecnologia em nosso dia a dia. Produtos de software fazem parte de praticamente todas as atividades humanas, sejam pessoais, sejam profissionais”.

Na construção de um software, a especificação de requisitos tem sua grande importância e contribuição no processo de desenvolvimento. Os requisitos podem ser explícitos, normativos e requisitos implícitos. No implícito, são indesejáveis pois não sendo documentados, eles podem ser considerados desenho do produto. Mas, mesmo assim, nos requisitos documentados podem apresentar problemas, decorrente de linguagem natural, normalmente usada para expressá-los. Já outros, recorrem a técnicas deficientes nesses levantamentos e especificações(PÁDUA FILHO, 2019).

A fase inicial é uma das etapas mais importantes no desenvolvimento de um sistema. Basicamente, significa entender o que o cliente deseja ou o que ele pensa ser importante, além das regras e processos de negócio. O principal objetivo do levantamento de requisitos é que usuários e desenvolvedores possam ter uma mesma visão do problema a ser resolvido(O que é..., 2018).

A engenharia de requisitos fornece um mecanismo adequado para entender o que o cliente deseja, analisar as necessidades, avaliar a exequibilidade, negociar uma solução razoável, especificar a solução de maneira não ambígua, validar a especificação e administrar os requisitos à medida que eles são transformados num sistema em operação. (PRESSMAN, 2006, p.250)

Neste caso, a engenharia de requisitos vem cumprindo assim sua respectiva atribuição dentro do processo de construção de software. Esse processo de

engenharia de requisitos nos garante que o sistema gerado por esses requisitos possa atender muito bem todas as necessidades satisfazendo as demais expectativas dos respectivos clientes (Pressman, 2006).

A necessidade que existe em uma engenharia de requisitos é de grande valia para que o processo de construção do software ocorra com toda cautela e sucesso. Isso porque a engenharia de requisitos define o sucesso na construção do software pela elicitación de requisitos. “O levantamento ou elicitación de requisitos de um sistema consiste de uma etapa muito importante no desenvolvimento de projetos” (AMOROSO, 2023, online).

A tecnologia tornou-se bastante presente em nosso cotidiano, para Reinehr(2020) os softwares hoje em dia passam por grandes problemas devido levantamentos de requisitos errôneos, que não foram levantados de acordo com a solução real do problema. Nesta linha de raciocínio, para a autora, a engenharia de requisitos ajuda a minimizar esses problemas de requisitos, daí sua importância dentro de um projeto de construção de software.

Pressman (1995) coloca que há alguns atributos da análise de sistema, como a identificação das necessidades de determinado usuário, avaliação da qualidade do sistema, a análise econômica, dentre outros. Ele já identificava a análise de requisitos de um software como uma propriedade bastante importante dentro do âmbito do projeto, enfatizando-o na criação do mesmo.

Tal associação tem um grande impacto nos projetos atuais, dentro da análise de requisitos de software e seus processos. “A engenharia de requisitos oferece controles e padrões para que as exigências do projeto sejam claras” (MONITORA TEAM, 2020, online). Esta é uma ferramenta essencial na construção de software para estes autores e dentre outros, e a necessidade de aplicação e estudo aprofundado do mesmo pode ser o sucesso do projeto ou não.

Com tal premissa, este trabalho visa a corroborar com este tema aprofundando nos levantamentos de requisitos dentro do âmbito de projeto requerido por modelos definidos e regulamentados seguindo as normas técnicas de regulamentação. Propõe-se, a aplicabilidade das análises de requisitos dentro de um determinado órgão de funcionamento legislativo e municipal, com a aplicabilidade dessa análise propostos em sua composição.

Ao propor tal tema, temos o enfoque de ressaltar a importância da análise de requisitos no ambiente legislativo, bem como os conceitos práticos e conceituais na

elicitação dos requisitos, de maneira que seja uma bússola que aponte para uma direção necessária para levantamento do mesmo.

Propomos ao mesmo projeto, que seja realizada com as devidas cautelas e cuidados dentro de um projeto real para uso futuros, tendo em vista que o órgão legislativo possui suas necessidades tecnológicas de software que ajudariam no processo de digitalização do mesmo. Tal processo, melhoraria a transparência do processo legislativo.

Desde os requerimentos de entradas quanto às ordens de serviços e prestação de contas do legislativo à população, passando pela votação em plenário e requerimentos aprovados, votados e os não votados em sessão. Projetos de lei aceitos e demandas do executivo para o legislativo.

Tal processo mostrará a transparência do mesmo, e o acompanhamento por parte da população pelos projetos e análise das pautas em sessão ordinária em execução. Também viabiliza o processo de envio do mesmo ao jurídico da casa viabilizando e digitalizando o processo democrático em questão. A usabilidade é garantir o processo e melhorar a transparência do mesmo.

Sendo assim, qualquer pessoa dentro ou fora do âmbito da tecnologia independentemente de seu nível de abstração conseguirá verificar as informações do legislativo e acompanhar as demandas de seu candidato no qual realizou seu voto. O mesmo processo contará com as entrevistas visando o procedimento do mesmo e analisando os passos dentro do processo legislativo atual.

2 A ENGENHARIA DE REQUISITOS

A análise de requisitos define o escopo num projeto, ela funciona como uma bússola apontando a direção a seguir. Essa definição é necessária antes de se iniciar a construção do software, para entender tal importância e complexidade do mesmo.

Deve haver muita comunicação para que não gere redundâncias de informação, e para não haver interpretações errôneas ou falsas informações (PRESSMAN, 2006). Para Reinehr (2020), dentre os vários problemas relacionados aos requisitos são:

- Estar incompleto;
- Com níveis insuficientes dos detalhes para etapas de ciclo de desenvolvimento;
- Estar incompatíveis com outros requisitos;
- Inviável tecnicamente;
- Testes e validação com mais dificuldades;
- Serem conflitantes na visão dos stakeholders.

Tais problemas podem acontecer dentro da engenharia de requisitos variando de acordo com, projetos. A engenharia viabiliza o processo de especificação da função do software, “a análise de requisitos proporciona ao projetista uma representação da informação e da função” (PRESSMAN, 2006).

Os requisitos são obtidos diretamente com o cliente ou usuários, e, caso não haja uma comunicação correta, problemas podem vir à tona nos requisitos e construção do software. Reinehr (2020), nos afirma que “é comum ouvir que os usuários não sabem o que querem, mas, na verdade, por mais que eles saibam, sua perspectiva pode mudar pelo simples fato de entrarem em contato com a equipe”.

Isso significa que usuários podem sim ter uma visão de requisitos onde podem ser alteradas no meio desse levantamento. Geralmente, temos que guiar o usuário para e conduzi - lo para onde ele quer chegar. A engenharia de requisitos está de certa maneira, unida na aquisição de conhecimento para sua criação, também com aperfeiçoamento e implementações dele.

Tal afirmativa de Vazquez & Simões (2016), define a primeira etapa do processo em buscar e identificar problemas para resolvê-los. Desenhos de diagramas são essenciais nesse processo, como abaixo em um diagrama onde é retratado um determinado fluxograma no processo de coletas de informações.

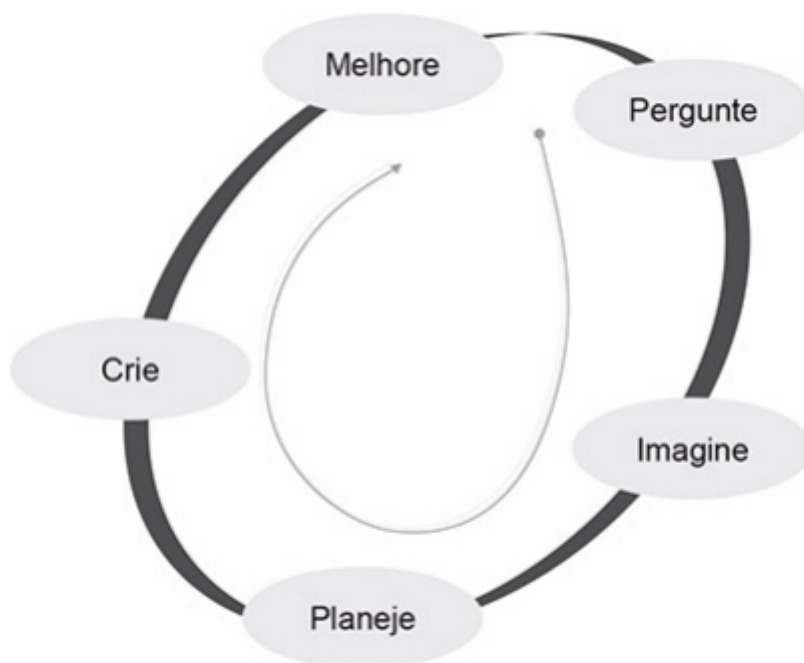


Figura 1 - Diagrama de planejamento.

Fonte: Eng. de Requisitos(Vazquez & Moura, 2016, p.29)

Analisando tal diagrama, vemos o início de um determinado ciclo a partir de perguntas para coleta de informações dentro do processo na construção de software. Logo em seguida o demais procedimentos nessa entrevista, é um fluxo contínuo. Nisso, sabemos para Machado(2014), que os requisitos são o início na construção de software, tornando - se essenciais dentro deste processo.

Segundo defende Freitas:

Considerando-se os requisitos como o alicerce para a construção do software em questão, é preciso garantir que eles representem efetivamente as necessidades e acordos 12 realizados para que o seu desenvolvimento seja seguro e com qualidade. (Freitas, 2006, p. 11 e 12)

Iniciar um projeto tão grande sem um alicerce o torna vulnerável a erros dentro do processo de construção tornando - se base para o mesmo. Esse é o

marco zero na construção em uma jornada que requer atenção, e o mínimo de erros na coleta dos requisitos mediante o usuário.

2.2 REQUISITOS

Os requisitos, de maneira bem geral, são basicamente divididos em requisitos funcionais, que nos definem o que um software irá fazer, basicamente está ligado à sua função de “o quê”. Já os por outro lado, temos o requisito não funcional, este, define as restrições do mesmo no ambiente de uso para o usuário, este, está ligado sob que restrições, por isso é um requisito não funcional (VALENTE, 2010).

Tal definição nos demonstra a grande importância relativa que é o tema abordado. Falar de requisitos é de tamanha importância dentro da análise de requisitos que, torna - se uma parte capaz de definir o sucesso e a abordagem do software. Um software sem o devido levantamento de requisitos, torna -se obsoleto.

Para Soares e Zanin(2010), o software precisa de recursos para que suas funções possam operar pelo dispositivo físico. Neste caso, será necessário definir qual a restrição e o limite de alcance que poderá ter. Tal requisito, não lista apenas a sua funcionalidade mas pode nos trazer sua especificação sobre como agir, assim, o objetivo será alcançado.

Segundo consta na obra de Gallotti (2016, p. 122), “especificar requisitos é a etapa em que escrevemos os requisitos de usuário e de sistemas no documento de requisitos”. Isso em tese, com os mesmos claros e objetivos precisos, também devem estar completos, consistentes, porém isso não é tão simples (GALLOTTI, 2016).

2.3 GERENCIAMENTO DE REQUISITOS

Em um projeto de desenvolvimento de software a produção de informações são amplas:

- Para Pádua Filho(2019), em projetos complexos, necessitam de um maior investimento na engenharia de requisitos. Ela se torna bastante complexa em novos produtos, e, em casos de versões anteriores, o usuário pode pela sua experiência identificar as necessidades mais prioritárias.

- O software deve atender às necessidades do cliente: Neste caso é o resultado final fazendo sua relação custo/benefício. Para Hirama(2012), leva - se em consideração o custo e o prazo. É necessário que haja um gerenciamento desses requisitos desde sua especificação até a entrega do software ao cliente.O gerenciamento de requisitos trata das entregas e das mudanças nesses requisitos.
- Os requisitos devem estar bem identificados para que suas atividades tenham sucesso (REINEHR, 2020).
- A engenharia de requisitos torna a relação cliente x software facilitada pela sua interação, neste caso, irá entender suas necessidades e identificá-las como parte da solução que será entregue (SIMÕES e VASQUEZ, 2016).

2.4 REQUISITO FUNCIONAL

O requisito dentro de uma construção de software é de extrema importância, é uma necessidade de, pelo requisito, definir todos os detalhes na construção do software, isso definirá sua funcionalidade. Ao falarmos de requisito funcional, dizemos à sua funcionalidade, tanto no que irá atender ou realizar (VENTURA, Ebook).

Kerr(2015) definiu os requisitos funcionais como definições sobre funcionalidades, tal definição é fornecida pelo sistema, mostrando - o sobre o que deve fazer. Para Rezende(2005), “a relação dos requisitos funcionais são definidos para determinação do tempo”. Neste caso, sua distribuição se dá por tipos como: telas, arquivos, relatórios, processamento além de testes.Ele, continua na relação desses requisitos, tais como:

- Planejar;
- Levantar as necessidades;
- Determinar os requisitos funcionais;
- Elaborar a macro proposta;
- Relatar requisitos funcionais;
- Elaborar os diagramas;
- Descrever a lógica dos requisitos funcionais;
- Desenhar as telas, relatórios e arquivos;

- Análise de integrações;
- Definição da tecnologia e elaboração de plano logístico;
- Análise de custos, benefícios e viabilidade;
- Definir os procedimentos de segurança;
- Documentar o manual para o cliente e o manual técnico;
- Planejar a implementação, treinamento do cliente;
- Finalizar e acompanhar pós-implementação;

1. Criação de Arquivos	DC	DNC	MAN
<i>Cadastro</i>			
- arquivo físico pequeno	0,06	0,07	0,06
<i>Movimento</i>			
- arquivo físico grande	0,15	0,20	0,15
2. Definição de Telas			
<i>Lista</i>			
- tela média	0,15	0,20	0,15
<i>Detalhe</i>			
- tela grande	0,25	0,40	0,25

Figura 2. Tabela de tempos para desenvolvimento

Fonte: Rezende(2006)

Como nesta tabela acima, o autor sugere que deve ser atribuído um determinado tempo em minutos na elaboração, neste caso, o DC é o desenvolvimento conhecido, o DNC é o desenvolvimento não conhecido e o MAN é a manutenção de sistemas. Ela mostra alguns dados gerados no tempo pré definido anteriormente.

2.5 Elicitação de Requisitos

Não sei se coloco este tópico no projeto

2.6 DOCUMENTAÇÃO NO PROCESSO DE REQUISITOS

Há uma variedade de possibilidades nesta fase do projeto, Ledur(2017) nos diz que antes que seja iniciado o projeto é necessário estar definido o escopo que será criado. Nesta definição, é levantado alguns diagramas de casos de uso , onde

as técnicas abordadas pelo gerente de projeto seguem perguntas cruciais na sua definição.

Perguntas como: Quem? O quê? Quando? Como? Onde? Por quê? devem ser palavras chaves para direcionar para um escopo bem elaborado e definido.

Com estas perguntas realizadas a respeito do sistema, dificilmente ficarão dúvidas quanto ao escopo do projeto, já que elas tentam induzir o gerente de projetos a pensar em todo o processo de uso e interação que o software terá com o mundo real. Por exemplo, veja no Quadro 1 como são feitos os questionamentos para fazer o levantamento de escopo de uma aplicação real para celulares (LEDUR, 2017, p. 40).

O quadro abaixo do autor demonstra bem sua idéia:

Quem?	O quê?	Quando?	Como?	Onde?	Por quê?
Usuário	Escolhe uma aplicação para rodar	A qualquer momento	Através de uma interface de seleção de aplicações	No celular (ou outro dispositivo móvel)	Porque deseja executar a aplicação
Usuário	Executa uma aplicação	Quando a aplicação estiver rodando	Preenchendo formulários, navegando entre as páginas da aplicação, enviando informação...	No celular (ou outro dispositivo móvel)	Para utilizar a aplicação disponibilizada

Figura 3 - Quadro de perguntas do usuário para escopo

Fonte: Ledur(2017)

Segundo consta a Norma IEEE (1996) que é a prática recomendada para especificações de exigências de software, descreve o conteúdo e a qualidade de uma boa especificação de exigências de software. Nele, são descritas as abordagens para uma especificação de requisitos de software, e, são exigências deste mesmo software.

Em relação a natureza do documento, quem o descreve deve ter em conta pontos bases como funcionalidades, interfaces externas, performance, atributos e restrições. Seu propósito, harmoniza a definição de conteúdo para resultados do

processo de ciclo de vida do software entre padrões de engenharia de software IEEE.

2.7 VALIDAÇÃO

Dentro da validação de requisitos, a consistência tem sido de fundamental preocupação no decorrer da elicitação de requisitos. Tais artefatos que são produzidos nesse processo da engenharia de requisitos, são avaliados ao longo do processo, que é comumente chamada de validação de requisitos (PRESSMAN; MAXIM, 2021) .

Para Vetorazzo(2018), produtos defeituosos são evitados pelo controle de qualidade, e, esse detalhe evita que o software seja entregue de maneira defeituosa ao cliente. A garantia visa o gerenciamento e melhoria dos processos existentes, onde as inspeções são realizadas na etapa de validação evitando nos testes que o software seja entregue com erros.

[...]as inspeções verificam a conformidade com uma especificação do software, mas não verificam a não conformidade com os requisitos reais do cliente e nem características não funcionais, como por exemplo, desempenho, usabilidade e adaptabilidade (VETORAZZO, 2018, p. 77).

O teste de software e a depuração, são fatores de atividades relacionados intrinsecamente à validação do software. Neste caso, podemos distingui-las com algumas perguntas: a primeira é se o produto a ser construído é o correto, neste caso, estamos tratando com validação. E a segunda pergunta é, se o produto está correto quanto a sua construção, esta é ligada a verificação (INTHURN, 2001).

2.8 CONTROLE DE QUALIDADE

Para Staa(1983), o controle de qualidade faz o papel de operar do detalhe em sentido ao geral. Primeiramente as métricas serão obtidas para apenas depois quantificar os diversos fatores pelas funções normalistas. Neste caso, poderá haver rejeições nas ocasiões dentro do controle de qualidade, para este caso, somente se os controles caírem abaixo dos limites tolerados.

De acordo com Inthurn (2001, p.27), “a tarefa de revisar a especificação é algo muito importante para que falhas sejam detectadas o mais breve possível”. Para a autora, tal trabalho requererá uma dedicação consistente à equipe

desenvolvedora do projeto, para que o resultado seja satisfatório e logo no início, sejam verificadas as vantagens no desenvolvimento do sistema.

2.8 GERENCIANDO OS REQUISITOS

Dentro de um projeto, podemos destacar a importância de identificar e gerenciar os requisitos, definir seu gerenciamento e suas mudanças definindo neste processo as ferramentas necessárias para a gestão do mesmo. Gerenciar requisitos, torna - se atrativo ao requerer alguns desafios e competências técnicas da equipe, tornando - o um recurso muito importante para o processo de construção do software.

Para Sommerville(2011), “uma vez que um sistema tenha sido instalado e seja usado regularmente, inevitavelmente surgirão novos requisitos[...] quando usuários finais tiverem a experiência de um sistema descobrirão novas prioridades”. Para o mesmo autor fica evidente a tese de que existem várias razões pelas quais as mudanças são inevitáveis.

Segundo (LEFFNGWELL & WIDRIG, 2003 apud LEDUR, 2017), uma pergunta na qual fica e: “porque desenvolvedores falham em construir a coisa certa?”(2017). Tal pergunta ecoa para os desenvolvedores e equipes de projeto. Segundo o autor, a concepção de novos sistemas geram grande parte dessas falhas ou, um gerenciamento ruim desses sistemas de produtos.

Tais requisitos possuem um ciclo de vida definido. É a atividade de elicitação que faz esses ciclos nascerem. Os mesmos serão validados e analisados e somente depois estarão prontos a serem codificados. A partir daí, o requisito é parte fixa dentro do produto, exceto que não seja exigida mais dentro do software (REINEHR, 2020). Logo abaixo veremos um gráfico da evolução dos requisitos para melhor elucidar o tema.

Evolução dos requisitos.

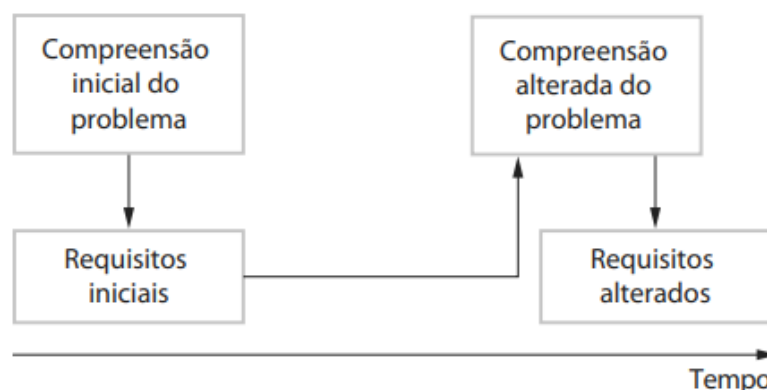


Figura 4 - Evolução dos requisitos

Fonte:(Sommerville, 2011)

Para Sommerville(2011), há três estágios principais em um processo de gerenciamento de mudanças que são:

- Análise de problema e especificação de mudanças. Nesta fase é inicializado com a identificação dos problemas de requisitos. Neste estágio os problemas ou as propostas serão analisados para verificação de validade.
- Análise de mudanças e custos. Todas as mudanças serão avaliadas por informações de rastreabilidade do sistema, bem como conhecimentos gerais dos requisitos.
- Implementação de mudanças. Organiza - se os documentos de requisitos quando necessário com alterações que não haja reformulações.

3 DAS TÉCNICAS DE LEVANTAMENTO DE REQUISITOS

No último capítulo vimos sobre a engenharia de requisitos, nele, vimos a sua importância e necessidade dentro do escopo do projeto. Observamos nele, seus fundamentos, os ciclos de planejamento, o que são os requisitos, seu gerenciamento e uma diferença entre funcionais e não funcionais. Vimos o processo de documentação dessa etapa no projeto, sua validação e controle de qualidade.

Tivemos uma abordagem introdutória sobre o gerenciamento de requisitos, onde pudemos ver os três estágios de mudança de gerenciamento, seguindo a análise de Sommerville(2011). Nele, dispomos a evolução dos requisitos para melhor elucidar o tópico abordado.

Outro tema importante referente ao capítulo anterior é a validação dos requisitos, tema abordado como um aspecto da elicitacão de requisitos tomando ênfase nos testes de software, como instrumento dessa validação, que está ligada à verificação.

Nas próximas seções, mostrar-se-ão algumas das técnicas de levantamentos de requisitos.

3.1 BRAINSTORM

Brainstorm é uma das técnicas de levantamentos de requisitos que possui um conceito em determinado conjunto de ideias. Tal técnica se vê desprovida de julgamentos ou até mesmo análises. Ela é ideal para a busca de novas ideias e produtos (ZANCHETT, 2015).

Nessa técnica, há uma reunião entre membros da equipe, neste caso, desenvolvedores serão requisitados a fazerem um brainstorm dos grupos de tarefas detalhadamente, visando a meta. Alguns desses exemplos são: tarefas de banco de dados, tarefas de integração de sistemas externos, tarefas de desenvolvimento orientado a objetos da camada de domínio etc (LARMAN, 2007).

3.2 ENTREVISTA

Sommerville(2011), descreve entrevistas e questionários como as técnicas mais usadas na análise de processos. Esta técnica de entrevista pode ser realizada

de maneira formal ou informal com *stakeholders*. Estes, são questionados sobre o sistema usado e sobre o próximo sistema a ser desenvolvido. Os requisitos podem surgir a partir de respostas às perguntas, neste caso, podemos ter dois tipos de entrevistas: fechada e aberta (SOMMERVILLE, 2011).

A diferença para as duas é que, na entrevista fechada haverá um conjunto de perguntas pré definidas pela equipe de análise de requisitos. Para Sommerville(2011), “entrevistas abertas, em que não existe uma agenda predefinida. A equipe de engenharia de requisitos explora uma série de questões com os stakeholders do sistema e, assim, desenvolvem uma melhor compreensão de suas necessidades”.

3.3 QUESTIONÁRIO

Descrito com uma das duas principais técnicas assim como a entrevista (SOMMERVILLE, 2011). Para Zanchett(2015), “questionários visam extrair respostas através de questões subjetivas e objetivas“. O questionário é na realidade um grande instrumento que possibilita a coleta de dados quando o grupo é muito grande, ou, quando cada integrante mora geograficamente distante (REINEHR, 2020).

Para a autora, há alguns passos que necessitam a serem seguidos viabilizando a aplicação da técnica.



Figura 5: Passos para a utilização da técnica de questionário.

Fonte: (Reinehr, 2020).

Em seguida, é necessário identificar o público-alvo da pesquisa. Aplicar o questionário para o público-alvo pode ser constituído de membros internos ou externos à organização. As questões serão elaboradas em função do perfil desse público e será definida a forma mais adequada de aplicação (REINEHR, 2020, p. 76).

3.4 CASO DE USO

Dentre muitas técnicas no processo de desenvolvimento de software destacamos o caso de uso. Se pudéssemos descrevê-lo, seria, como um conto, uma jornada de interação entre o usuário do sistema e um conjunto de circunstâncias possíveis (PRESSMAN; MAXIM, 2021). Pressman e Maxim descrevem algumas características de sua sintaxe como vemos **abaixo**.

1. Quem são atores primários e secundários.
2. As metas do ator.
3. Condições para que aconteça.
4. As tarefas e funções realizadas pelo ator.
5. Exceções ao longo do caminho de percurso.
6. Possíveis variações na interação do ator.
7. Informações adquiridas, produzidas e modificadas pelo ator.
8. Mudanças externas ao ambiente e a informação obtida e informada pelo ator.
9. Informações desejadas pelo ator no sistema.
10. Mudanças inesperadas e se o ator quer ser informado sobre.

Para Fowler(2005), “os casos de uso são uma técnica para capturar os requisitos funcionais de um sistema. Eles servem para descrever as interações típicas entre os usuários de um sistema[...]”. Para ele, é mais fácil descrever os cenários ao invés de iniciar com o caso de uso. Os cenários, segundo Fowler(2005), são sequências de passos descritos na interação entre usuários e sistemas.

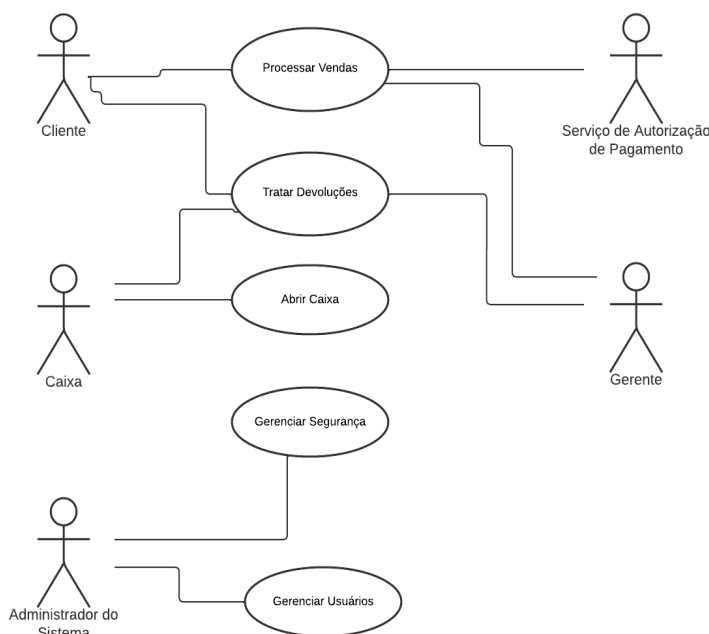


Figura 6: Diagrama de casos de uso

Fonte: Larman(2007)

A figura 6 nos mostra um exemplo de diagrama de caso de uso, onde existem alguns atores envolvidos que são: Cliente, caixa, administrador do sistema, gerente, serviço de autorização de pagamento. Todos se relacionam com o sistema, uns com permissões distintas de outros.

Um cliente interage com o sistema da loja em processar a venda e o serviço de autorização de pagamento interage autorizando ou não esse processamento. Ao mesmo tempo o cliente pode tratar devoluções e o caixa interage no mesmo intuito do tratamento dessas devoluções.

O gerente pode interagir tanto no tratamento das devoluções, quanto em processar as vendas, porém, somente o caixa pode interagir abrindo o caixa. Já o administrador interage tanto no gerenciamento da segurança quanto de usuários do sistema.

Os casos de uso na realidade são textos narrativos de um determinado sistema, que visa atingir os objetivos (LARMAN, 2007). Agora, vejamos um exemplo de caso de uso em formato textual na figura abaixo.

Processar Venda: um cliente chega em um ponto de pagamento com itens que deseja adquirir. O caixa usa o sistema PDV para registrar cada item comprado. O sistema apresenta um total parcial e detalhes de linha de item. O cliente entra os dados sobre o pagamento, que são validados e, em seguida, registrados pelo sistema. O sistema atualiza o estoque. O cliente recebe um recibo do sistema e sai com os itens comprados.

Figura 7: Descrição contínua

Fonte: Larman(2007)

Vejamos abaixo um modelo de um caso de uso definido por (LARMAN, 2007, p. 86).

Caso de uso CDU1: processar venda

Escopo: Aplicação PDV ProxGer

Nível: objetivo do usuário

Ator Principal: Caixa

Interessados e Interesses:

- Caixa: deseja entrada de pagamento rápida, precisa e sem erros, pois a falta de dinheiro na gaveta do caixa será deduzida do seu salário.
- Vendedor: deseja comissões sobre vendas atualizadas.
- Cliente: deseja comprar, receber um serviço rápido e com o mínimo esforço. Deseja a exibição, facilmente visível, dos itens e preços inseridos. Deseja um comprovante da compra, necessário no caso de devoluções de mercadorias.
- Empresa: deseja registrar precisamente as transações e satisfazer aos interesses do cliente. Quer garantir que os pagamentos a receber do Serviço de Autorização de Pagamentos sejam registrados. Deseja algum tipo de proteção contra falhas para permitir que as vendas sejam capturadas mesmo se os componentes do servidor (por exemplo, validação remota de crédito) se encontrarem indisponíveis. Deseja uma atualização automática e rápida da contabilidade e do estoque.
- Gerente: deseja poder realizar rapidamente operações de correção e facilmente corrigir os problemas do Caixa.
- Órgãos fiscais governamentais: desejam cobrar os impostos de cada venda. Podem estar envolvidos vários órgãos, como, por exemplo, federais, estaduais e municipais.
- Serviço de autorização de pagamentos: deseja receber solicitações de autorização digital no formato e protocolo corretos. Deseja contabilizar com precisão seus débitos a pagar para a loja.

Pré-Condições: Caixa está identificado e autenticado.

Garantia de Sucesso (ou Pós-Condições): Venda foi salva. Impostos foram corretamente calculados. Contabilidade e Estoque foram atualizados. Comissões foram registradas. Recibo foi gerado. Autorizações de pagamento foram registradas.

Cenário de Sucesso Principal (ou Fluxo Básico):

1. Cliente chega à saída do PDV com bens ou serviços para adquirir.
2. Caixa começa uma nova venda.
3. Caixa insere o identificador do item.
4. Sistema registra a linha de item da venda e apresenta uma descrição do item, seu preço e total parcial da venda. Preço calculado segundo um conjunto de regras de preços. Caixa repete os passos 3 e 4 até que indique ter terminado.
5. Sistema apresenta o total com impostos calculados.

Figura 8: Caso de uso processar venda

Fonte: Larman(2007)

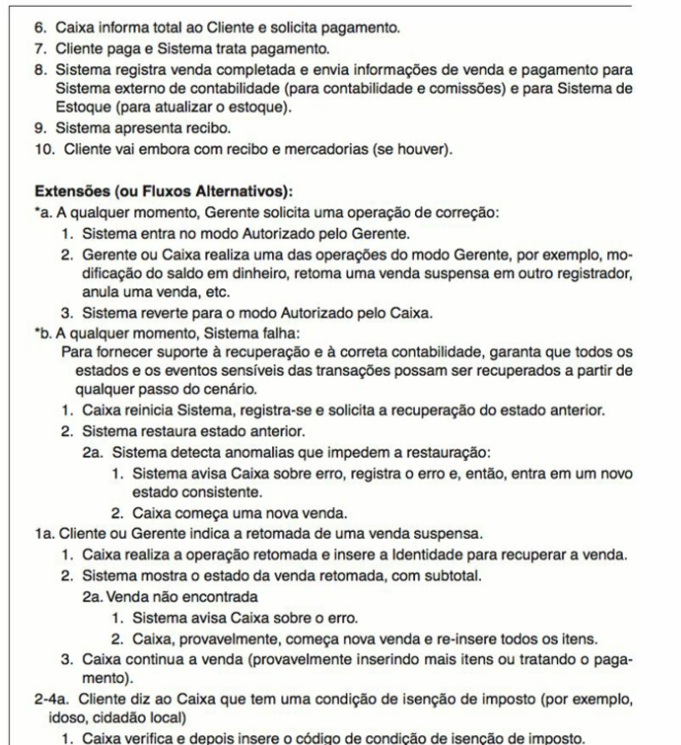


Figura 8: Caso de uso processar venda (continuação)

Fonte: Larman(2007)

3.5 QUESTIONÁRIO NPS (NET PROMOTER SCORE) **OBS**

3.6 OBSERVAÇÃO DIRETA

3.7 ANÁLISE DOCUMENTAL

3.8 REUTILIZAÇÃO DE REQUISITOS

3.9 PROTOTIPAÇÃO

3.10 ETNOGRAFIA

3.11 WORKSHOPS

3.12 JAD - JOINT APPLICATION DESIGN

3.13 PONTO DE VISTA

4 METODOLOGIA

REFERÊNCIAS

A

Amoroso, Edgar. Engenharia de Requisitos de Software. **Católica.edu**. Disponível em:

https://conteudo.catolica.edu.br/conteudos/nbt_cursos/engenharia_requisitos/tema_02/index.html?access_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvY29udGV1ZG8uY2F0b2xpY2EuZWZWR1LmJyYXVkljoiaHR0cHM6XC9cL2NvbnRldWRvLmNhdG9saWNhLmVkdS5icilslmlhdCI6MTU5Mzk4NjE0NiwiImJmljoxNTkzOTg2MTQ1LCJkYXRhIjpbIjNkZWE2OWRlliwiNjVhYTEyNjEiXX0.iZ3ceZBLZEMmEdbijnCqxFQH7GuNtAahhDVLQlgVN6g. Acesso em: 09 Mar. 2023.

ANÁLISE de Requisitos de Software: quais desafios e como fazer?. **Monitora Team**. 21 Jan. 2021. Disponível em:

<https://www.monitoretec.com.br/blog/analise-de-requisitos-de-software/#:~:text=A%20an%C3%A1lise%20de%20requisitos%20de%20software%20ajuda%20o%20cliente%20e,as%20fun%C3%A7%C3%B5es%20do%20novo%20sistema>. Acesso em: 10 Mar. 2023.

F

Gallotti, Giocondo Marino Antonio. Arquitetura de Software. São Paulo: Pearson Education do Brasil, 2016.

Fowler, Martin. UML Essencial: Um Breve Guia para a Linguagem-Padrão de Modelagem de Objetos. 3º Ed. Porto Alegre: Brookman, 2005.

I

Inthurn, Cândida. Qualidade & Teste de Software. Florianópolis: Visual Books, 2001.

H

Hirama, Kechi. Engenharia de Software: Qualidade e Produtividade com Tecnologia. Rio de Janeiro: Elsevier, 2011.

K

Kerr, Eduardo Santos. Gerenciamento de Requisitos. São Paulo: Pearson, 2015.

L

Ledur, Cleverson Lopes. Análise de Projeto de Sistemas. Porto Alegre: Sagah, 2017.

Larman, Craig. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientados a Objetos e ao Desenvolvimento Iterativo. 3 ed. Porto Alegre: Bookman, 2007.

M

Machado, Felipe Nery Rodrigues. Análise e Gestão de Requisitos de Software: Onde Nascem os Sistemas. 2 ed. São Paulo: Érica, 2014.

Morais, Izabelly Soares de. Engenharia de Software. Porto Alegre: Sagah, 2017.

O

O que é Levantamento de Requisitos - Tópicos de Engenharia de Software. [S. l.: s. n.], 2018. 1 vídeo (21 min). Publicado pelo canal Bóson Treinamentos. Disponível em: <https://www.youtube.com/watch?v=VcOeM2AD8Yk>. Acesso em 10 mar. 2023

P

Paula Filho, Wilson de Pádua. Engenharia de Software: Produtos. 4 ed. Rio de Janeiro: LTC, 2019.

Pressman, Roger. Engenharia de Software. 3. Ed. São Paulo: Makron Books, 1995.

Pressman, Roger; Maxim, Bruce. Engenharia de Software: uma abordagem profissional. 9. Ed. Porto Alegre: AMGH, 2021.

R

Reinerh, Sheila. Engenharia de Requisitos. Porto Alegre: Sagah, 2020.

Rezende, Denis Alcides. Engenharia de Software e Sistemas de Informação. 3 ed. rev. e ampl. Rio de Janeiro: Brasport, 2005.

S

SERVIÇO de Engenharia de Requisitos: Entenda como Funciona. **Monitora Team**. 28 Ago. 2020. Disponível em:

<https://www.monitoretec.com.br/blog/analise-de-requisitos-de-software/#:~:text=A%20an%C3%A1lise%20de%20requisitos%20de%20software%20ajuda%20o%20cliente>

[%20e.as%20fun%C3%A7%C3%B5es%20do%20novo%20sistema](#). Acesso em 10 mar. 2023.

Sommerville, Ian. Engenharia de Software. 9 ed. São Paulo: Pearson Prentice Hall, 2011.

V

Valente, Marco Túlio. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade. Belo Horizonte: Independente, 2020. E-book.

Vazquez, Carlos Eduardo; Simões, Guilherme Siqueira. Engenharia de Requisitos: Software Orientado ao Negócio. Rio de Janeiro: Brasport Livros, 2016.

Vetorazzo, Adriana de Souza. Engenharia de Software. Porto Alegre: SAGAH, 2018.

Zanchett, Pedro Simeí. Engenharia e Projeto de Software. Indaial: Uniasselvi, 2015.