



ENGENHARIA DE REQUISITOS



Priscila Lábamca

ENGENHARIA DE REQUISITOS

1ª edição

Londrina
Editora e Distribuidora Educacional S.A.
2020

© 2020 por Editora e Distribuidora Educacional S.A.

Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Editora e Distribuidora Educacional S.A.

Presidente

Rodrigo Galindo

Vice-Presidente de Pós-Graduação e Educação Continuada

Paulo de Tarso Pires de Moraes

Conselho Acadêmico

Carlos Roberto Pagani Junior
Camila Braga de Oliveira Higa
Carolina Yaly
Giani Vendramel de Oliveira
Henrique Salustiano Silva
Juliana Caramigo Gennarini
Mariana Gerardi Mello
Nirse Ruscheinsky Breternitz
Priscila Pereira Silva
Tayra Carolina Nascimento Aleixo

Coordenador

Henrique Salustiano Silva

Revisor

Douglas Fabiano Lourenço

Editorial

Alessandra Cristina Fahl
Beatriz Meloni Montefusco
Gilvânia Honório dos Santos
Mariana de Campos Barroso
Paola Andressa Machado Leal

Dados Internacionais de Catalogação na Publicação (CIP)

Làbamca, Priscila.

L113e Engenharia de requisitos / Priscila Làbamca. –
Londrina: Editora e Distribuidora Educacional S.A., 2020.
44 p.

ISBN 978-65-87806-39-6

1. Engenharia. 2. Requisitos. 3. Análise. I. Título.

CDD 620

Raquel Torres – CRB 6/2786

2020

Editora e Distribuidora Educacional S.A.
Avenida Paris, 675 – Parque Residencial João Piza
CEP: 86041-100 — Londrina — PR
e-mail: editora.educacional@kroton.com.br
Homepage: <http://www.kroton.com.br/>

SUMÁRIO

Fundamentos da Engenharia de Requisitos _____	05
Tipos e classificação dos requisitos _____	21
Levantamento de requisitos _____	35
Análise, validação, mudança e gerência de requisitos _____	52

Fundamentos da Engenharia de Requisitos

Autoria: Priscila Lábamca

Leitura crítica: Douglas Fabiano Lourenço



Objetivos

- Revisar os conceitos sobre o ciclo de desenvolvimento de software.
- Conceituar e explorar a disciplina de Engenharia de Requisitos.
- Relacionar as atividades desempenhadas pelo Analista de Requisitos com os Métodos Ágeis.
- Apresentar o profissional que atua na área de Engenharia de Requisitos.

1. O ciclo de desenvolvimento de software e o Analista de Requisitos

O ciclo de desenvolvimento de software pode ser considerado uma importante forma de organização para a elaboração do software. Esta organização ocorre nos mais diversos setores, cargos e funções dentro de uma equipe de projetos. As metodologias mudam, ferramentas para desenvolvimento se atualizam ou são substituídas, mas o ciclo de vida em sua definição não muda. Assim, é possível – dependendo da metodologia adotada — que ele seja incrementado, mas sua definição perdura sempre.

O ciclo de desenvolvimento de software pode ser comparado à construção de uma casa, confira o quadro a seguir que ilustra essa similaridade.

Quadro 1 – Similaridades: construção de uma casa e de um software


Ciclo	Construção da casa	Construção do software
Ideia	O proprietário descreve a visão da casa para o projetista.	O cliente descreve as necessidades que o sistema deve sanar.
Planejamento	A ideia é colocada em forma de croqui e desenhos esquemáticos.	A ideia é transferida para um documento sem se preocupar com os detalhes.
Análise	A partir do croqui e desenhos esquemáticos são inseridos os detalhes.	O documento de planejamento é incrementado com detalhes.
Projeto/Design	São consolidadas todas as exigências do cliente e materializadas em uma linguagem de modelagem técnica.	São consolidadas todas as exigências do cliente e materializadas em uma linguagem de modelagem técnica (UML).

Implementação	Construção da casa propriamente dita. É a materialização das exigências realizadas pelo cliente.	Construção do software propriamente dito. É a tradução da linguagem de modelagem técnica numa linguagem de programação.
Entrega	Uma vez finalizada a construção, a casa é entregue ao cliente.	Uma vez finalizado o processo de tradução para a linguagem de programação, o software fica disponível para uso do cliente.
Mudanças	A casa é construída de acordo com o croqui e desenhos esquemáticos. Frequentemente há algumas alterações e tomadas de decisão pelo cliente enquanto a casa é construída.	O software é construído de acordo com a linguagem de modelagem. Frequentemente, há algumas alterações e tomadas de decisão pelo cliente enquanto o software é construído.

Fonte: elaborado pela autora.

Você consegue perceber que a cada passo que avançamos tanto no Modelo de Processos de Requisitos quanto no ciclo de desenvolvimento de software estamos tratando da ideia ou a necessidade do cliente de maneira mais detalhada? Além disso, esse mecanismo pode ser comparado ao estudo de idiomas, primeiro você aprende a estrutura e depois, aos poucos, você vai aprendendo a traduzir, por exemplo, do português para o inglês.

No caso dos requisitos, essa tradução é realizada em duas partes, sendo a primeira através da aplicação de técnicas conhecidas como Técnicas de Levantamento de Requisitos, já a segunda acomodando o resultado da utilização da primeira em um documento onde contém a descrição



das fases de execução para a tradução que inicia com as anotações feitas, utilizando as Técnicas de Levantamento de Requisitos (“língua do cliente”), seguidas das atividades de leitura, identificação e classificação do discurso em requisitos (“língua do cliente”), finalizando com a tradução para uma linguagem técnica (nas literaturas as “traduções” são conhecidas como Níveis de Abstração).

Desse modo, tão importante quanto conhecer a ideia do cliente é saber detalhá-la e modelá-la de maneiras íntegras, concisas, objetivas e claras. Para isto, contamos com um profissional que ocupa a função de analista de requisitos e a ferramenta que ele trabalha é a disciplina de Engenharia de Requisitos.

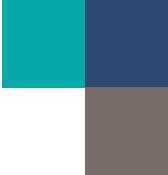
2. O analista de requisitos

Assim como toda área de atuação que é composta por determinados perfis (responsabilidades, deveres e habilidades) e diferentes cargos e funções, em Engenharia de Requisitos isso não seria diferente. Mas, você sabe qual é a diferença entre cargo e função?

Pois bem, segundo o dicionário online Michaelis (2020, [s.p.]) cargo é a “responsabilidade assumida em relação a alguém ou a alguma coisa; obrigação”. Assim, exemplos de cargo seriam: cargo de chefia, cargo de confiança etc.

Já a palavra função, segundo o citado dicionário, significa “conjunto das ações e atividades atribuídas esperadas ou exigidas de uma pessoa ou de um grupo” (MICHAELIS, 2020, [s.p.]).

Dependendo do tamanho da empresa de Tecnologia da Informação – TI (ou da empresa que possui um departamento de TI), um profissional pode desempenhar várias funções; uma delas é a função de analista de requisitos. A função de analista de requisitos é considerada uma



das funções do cargo de analista de sistemas, pois ele é responsável pelo levantamento, identificação, análise e modelagem dos requisitos que comporão o software. Logo, essa é uma função de grande responsabilidade, pois este profissional deve compreender bem os objetivos e necessidades do cliente, desenvolver ideias e sugestões, modelar e documentar o novo sistema, além de manter sempre a documentação atualizada em caso de mudanças no transcorrer do projeto. Ele também é a ligação entre o cliente e o restante da equipe de projetos.

Por sua vez, para executar a função de analista de requisitos é preciso que o profissional possua os seguintes pré-requisitos:

- a. **Tem que ser desinibido! (claro, no bom sentido):** este profissional não pode ter vergonha de falar em público, até mesmo, porque além de fazer o papel de mediador em reuniões, ele eventualmente terá que ministrar treinamentos para aqueles que utilizarão o sistema que foi modelado por ele.
- b. **Ser excelente comunicador:** de que adianta ser desinibido se não souber ser claro, conciso e coeso em seu discurso? E pior, não sabe escrever! Bons comunicadores são excelentes leitores, portanto, quanto mais prática do hábito da leitura, melhor comunicador o profissional se torna tanto na fala quanto na escrita.
- c. **Possuir excelentes conhecimentos em técnicas de elicitação de requisitos:** um sinônimo para elicitação é levantamento. Nesse sentido, há várias técnicas que auxiliam o profissional a fazer bem o seu trabalho. Assim, dependendo da complexidade do projeto é possível adotar uma ou mais delas.
- d. **Possuir excelentes conhecimentos em modelagem de sistemas:** quem domina a linguagem de modelagem técnica (UML) é um sério candidato ao rótulo de profissional por excelência. Além de saber falar bem e escrever bem a nossa

língua, saber traduzi-la para uma linguagem técnica de maneira correta, faz do profissional uma pessoa diferenciada.

- e. **Dominar o idioma:** saber muito bem ler, escrever e interpretar. Um texto mal escrito, por exemplo, dá a impressão de relaxo e falta de comprometimento com o projeto. Gírias, ambiguidades são muito malquistas tanto na escrita quanto na fala. Portanto, o analista de requisitos não precisa ter um vocabulário sofisticado, mas um vocabulário que seja simples, claro, coeso e objetivo é o ideal. Se o cliente fala gíria, o problema é dele e não do analista, o analista precisa ser profissional.

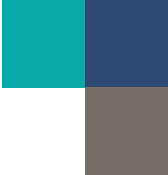
3. O conceito de Engenharia de Requisitos

Antes mesmo de iniciar o assunto sobre a Engenharia de Requisitos é importante compreender os conceitos que envolvem essas duas palavras, para que assim possamos melhorar nossa visão sobre a disciplina e os detalhes que a circundam.

Iniciando pela palavra engenharia, ela tem origem do latim *machinalis scientia* (ou *ingegneria* ou *ingeniarius* ou *ingeniarii*), que pode ser compreendida como (CONCEITO..., 2010, [s.p.]):

O estudo e a aplicação dos vários ramos da tecnologia. As funções do engenheiro consistem na materialização de uma ideia na realidade. Noutros termos, através de técnicas, desenhos e modelos, e com o conhecimento proveniente das ciências, a engenharia pode resolver problemas e satisfazer necessidades humanas.

Já a palavra requisito advém do latim *virtus dis* que, segundo Michaelis (2020, [s.p.]) significa “condição básica e necessária para se obter alguma coisa ou para alcançar determinado propósito”.



No entanto, concatenando as duas palavras (engenharia e requisito) é possível definir da seguinte maneira a Engenharia de Requisitos: uma disciplina como o estudo e a aplicação de técnicas, desenhos e modelos a fim de resolver e satisfazer determinadas condições necessárias para alcançar determinado objetivo. Neste caso, o nosso objetivo é materializar uma solução computacional para resolver/sanar necessidades dos clientes que podem ser de várias áreas do conhecimento, como a área da saúde (por exemplo, sistema para acompanhar a curva de progressão de doenças neurais, sistemas para controles epidêmicos etc.), área financeira (por exemplo, sistemas para bolsa de valores, sistemas de tesouraria de uma determinada instituição financeira etc.). Além disso, outras definições são muito valiosas, como é o caso da definição de Sommerville (2011, p. 71), que afirma esse “é o processo de entender e definir quais serviços são necessários do sistema e identificar as restrições na operação e desenvolvimento do sistema.” Por sua vez, Thayer e Dorfman (2016 *apud* PRESSMAN, 2016, p. 284) afirmam que:

A engenharia de requisitos fornece o mecanismo apropriado para entender o que o cliente deseja, analisar a necessidade, avaliar a viabilidade, negociar uma solução razoável, especificar a solução de forma inequívoca, validar a especificação e gerenciar os requisitos à medida que são transformados em um sistema operacional.

Perceba que a disciplina de Engenharia de Software se preocupa com a “tradução” da ideia de que o cliente possui de sistema que atenda às suas necessidades em um sistema de informação, de maneira mais fiel possível. Para que essa “tradução” seja realizada de maneira fiel, existem técnicas que o profissional que atua no levantamento de requisitos. Estes modelos e técnicas estão inseridos num processo denominado Processo de Engenharia de Requisitos.



4. O Processo de Engenharia de Requisitos

A Engenharia de Requisitos possui um processo composto por uma estrutura de atividades que objetivam derivar, validar e manter o documento de especificação de requisitos.

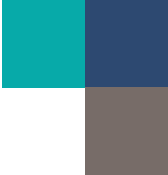
Para que o processo seja executado, utiliza-se um modelo denominado Modelo de Processos de Requisitos, onde a estrutura das atividades são chamadas de elicitação, especificação e revisão.

O modelo funciona seguindo a sequência de execução das atividades de elicitação, especificação e revisão. A sequência se inicia com a ideia ou a necessidade do cliente, e a partir disso são executados os seguintes procedimentos:

Passo 1 – Estudo de viabilidade: este estudo objetiva realizar a análise sob o ponto de vista de estrutura tecnológica (software e hardware) inserida nas dependências do cliente. Além disso, são consideradas as questões sobre o valor agregado sob o ponto de vista do negócio do cliente. O resultado deste estudo norteará a decisão de desenvolver ou não um novo software para o cliente.

Passo 2 – Aquisição de requisitos: este passo é composto pela atividade chamada Elicitação. Ao executar a atividade de Elicitação, por sua vez, são utilizados métodos e técnicas (técnicas de levantamento de requisitos). Existem vários tipos de modelos e técnicas, no entanto, tanto os modelos quanto às técnicas não são engessados, isto é, podem ser combinadas; porém isso dependerá do perfil e complexidade do projeto. Por exemplo, é possível adotar o método JAD (*Joint Application Designer*) e aplicar dentro deste modelo as técnicas de questionário e entrevista.

Passo 3 – Especificação de requisitos: este passo é composto pela atividade chamada Análise de Requisitos. A execução desta atividade se dá em dois tempos: o primeiro realiza-se a especificação (ou descrição)



dos requisitos coletados no Passo 2 e tem como dinâmica a identificação de incompletudes, omissões, redundâncias e distinguir entre requisitos desejados e requisitos necessários. Por requisitos desejados compreende-se os requisitos que o cliente gostaria que o sistema tivesse, na prática, em 99% das vezes não são adequados/necessários ao sistema. Aqui o cliente de maneira inocente não consegue distinguir aquilo que é necessário daquilo que não é necessário ou, até mesmo, redundante. Já o segundo tempo é destinada a criar um documento denominado Documento de Especificação de Requisitos.

Passo 4 – Revisão dos Requisitos/Validação dos Requisitos/

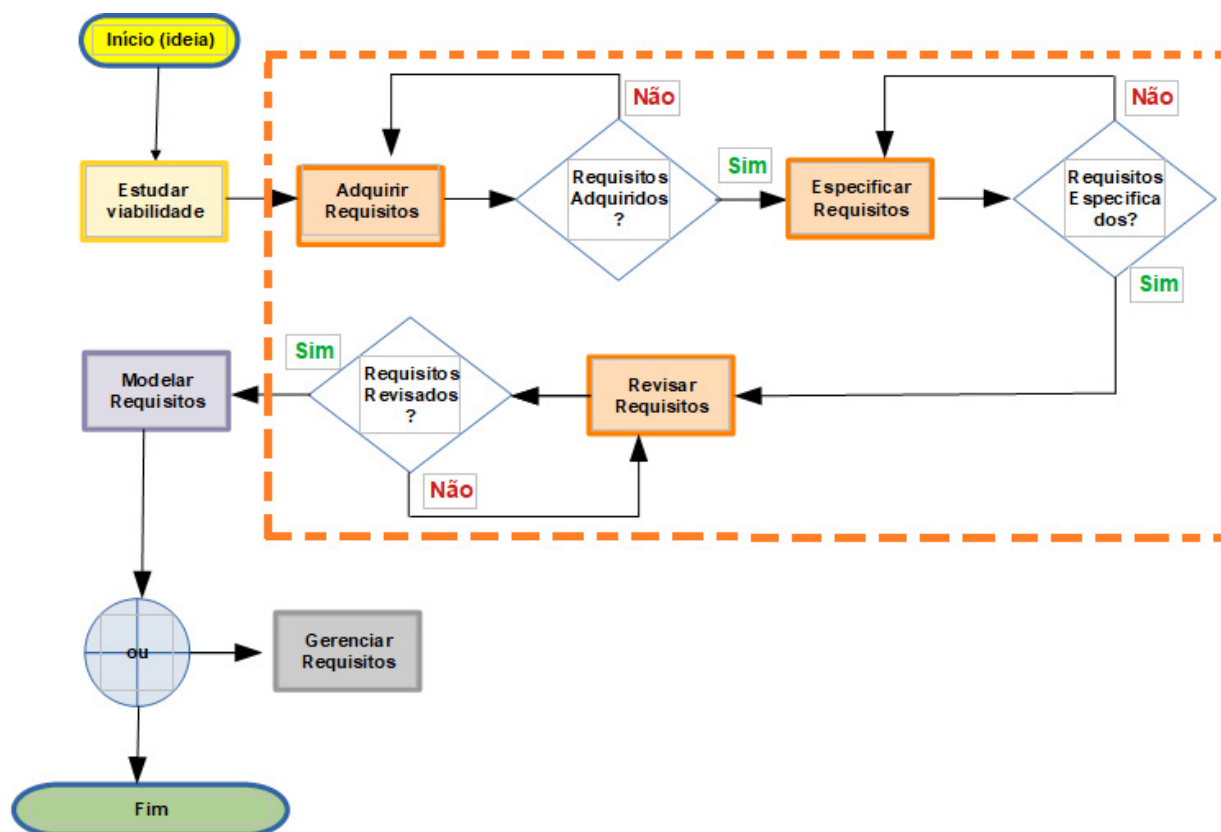
Homologação de Requisitos: neste Passo o analista de requisitos e o cliente se certificam de que os requisitos coletados estão consistentes e atingem os objetivos esperados sob o ponto de vista do software a ser desenvolvido, caso haja alguma dúvida ou identificação de alguma inconsistência, retorna-se à execução para o Passo 2. Esta fase pode ser entendida como uma revisão geral do software (a última revisão antes de seguir para o Passo 5) e o documento a ser analisado é o Documento de Especificação de Requisitos.

Passo 5 – Modelagem dos Requisitos: o analista de requisitos analisa o Documento de Especificação de Requisitos validado (passo 4) e o “traduz” para uma linguagem de modelagem (linguagem técnica) denominada UML. Neste momento, o Documento de Requisitos é incrementado, podendo ou não ser modificado o seu nome para Documento de Design de Requisitos ou apenas mantendo o seu nome. Isso dependerá da metodologia de trabalho da equipe que executará todos os ciclos para o desenvolvimento do software. Independentemente do nome que se dá a este documento, esse é encaminhado para duas equipes dentro do ciclo de desenvolvimento de software: a equipe de testes e a equipe de programação. A primeira lerá e elaborará os casos de testes (ou validação); já a segunda equipe “traduzirá” este documento para uma linguagem que o computador compreende: a linguagem de programação!

Passo 6 – Gerenciamento dos Requisitos: em boa parte da dinâmica diária do Analista de Requisitos, ele se depara com algumas surpresas. Essas surpresas, geralmente, consistem em contratos do cliente informando que “estava pensando melhor e acha que o sistema deve conter mais uma coisinha” (você se lembra da construção da casa no tópico 1?). Por mais que o Passo 4 tenha sido executado com sucesso, não adianta: o cliente deseja “mais uma coisinha”; dependendo da “coisinha”, isso pode ou não gerar impactos no fluxo de desenvolvimento do sistema. Portanto, neste Passo, cabe ao analista de requisitos observar novamente o Documento de Especificação de Requisitos e verificar se a “coisinha” está aderente/se atende aos objetivos do software que está sendo desenvolvido.

A figura a seguir mostra um fluxograma exemplificando a dinâmica execução destes passos:

Figura 1 – Fluxo de execução do modelo de processos de requisitos



Fonte: elaborada pela autora.

Na Figura 1 é possível visualizar que os Passos 2, 3 e 4 são cíclicos; em termos práticos, isso significa que enquanto estes três passos não forem bem definidos, não é possível avançar para os demais passos, onde é possível verificar um efeito dominó, ou seja, se uma peça não estiver em pé e firme, todas as peças correm o risco de caírem uma sobre as outras, em outras palavras, as consequências são perdas de tempo com o retrabalho e prejuízos. Observe que a atividade desempenhada pelo analista de requisitos é de extrema importância e crítica. Tudo gira em torno de requisitos!

5. As metodologias ágeis e o analista de requisitos

Inicialmente denominada de Métodos Livres, o marco das Metodologias Ágeis ocorreu entre os anos de 1980 e 1990, justificando que as metodologias atuais eram bastantes burocráticas e formais. Neste período, portanto, foram criadas muitas metodologias por pesquisadores e profissionais de diferentes perfis.

Em 2001, os pesquisadores (Kent Beck, Mike Beedle, Alistair Cockburn etc.) e profissionais reuniram-se para trocar experiências sobre as suas criações; durante a reunião perceberam que suas metodologias haviam pontos em comum e então resolveram reunir todas estas semelhanças num documento denominado Manifesto Ágil.

O Manifesto Ágil é um documento cujo conteúdo tem o objetivo de encorajar a utilização de melhores métodos de desenvolvimento de sistemas, obedecendo um conjunto de critérios que norteiam os processos de desenvolvimento ágil de softwares. Assim, a ideia é abandonar métodos antigos que se mostravam ultrapassados devido ao uso de hardwares mais avançados, linguagens de programação, ambientes de desenvolvimento e necessidades organizacionais. Ele

é composto por 12 (doze) princípios que regem o documento, que podem ser encontrados no website <http://agilemanifesto.org> (acesso em: 14 jul. 2020), sendo idealizado para redefinir continuamente as propriedades contidas em um projeto de desenvolvimento de software, ou seja, revisar processos, ferramentas e documentos de maneira rápida e objetiva; seu foco principal é nas pessoas que compõem a equipe de projetos. O Quadro 2 ilustra algumas vantagens de se utilizar as Metodologias Ágeis se comparadas às metodologias tradicionais.

Quadro 2 – Comparativo entre metodologias

Quesito	Metodologias tradicionais	Metodologias ágeis
Adaptabilidade	<p>1-Levantamento de todos os requisitos – poucas mudanças de requisitos admitidas (“engessada”).</p> <p>2 – Entrega realizada no final do projeto.</p>	<p>1 – Levantamento de requisitos – as mudanças são realizadas apenas se o requisito agrega valor ao negócio do cliente.</p> <p>2 – Entregas são realizadas por etapas, dando melhor compreensão da funcionalidade do sistema.</p>
Tipo de orientação	<p>1 – Pessoas são substituíveis.</p> <p>2 – O conhecimento encontra-se nos processos.</p> <p>3 – Não leva em conta a falta de motivação e produtividade.</p> <p>4 – Apenas uma pessoa é o líder (gerente de projetos).</p>	<p>1 – Pessoas são componentes.</p> <p>2 – O conhecimento encontra-se nas pessoas.</p> <p>3 – Levam em conta a motivação e produtividade.</p> <p>4 – O grupo é o líder (decisão tomada em conjunto).</p>

Planejamento, metodologia e cronograma	<p>1 – Planejamento: “engessado”.</p> <p>2- Metodologia: dividida em fases.</p> <p>3 – Cronograma: poucas alterações.</p>	<p>1 – Planejamento: conforme entregas.</p> <p>2- Metodologia: orientada ao código fonte.</p> <p>3 – Cronograma: conforme entregas.</p>
--	---	---

Fonte: elaborado pela autora.

Embora pautada no documento Manifesto Ágil, cada Metodologia Ágil possui algumas particularidades, principalmente quando se trata de pessoas; por exemplo, dependendo da Metodologia Ágil adotada, o nome do analista de requisitos muda, porém as atividades a serem executadas possuem adaptações quanto ao procedimento, mas não fogem àquelas definidas desde que a função foi criada.

Para melhor visualizar onde o analista de requisitos atua, veja o quadro a seguir, que contém o nome da metodologia ágil, como o analista de requisitos é conhecido e qual a metodologia de desenvolvimento de projetos empregada.

Quadro 3 – As principais metodologias ágeis e a atuação do analista de requisitos

Metodologia Ágil	Breve definição da Metodologia Ágil	como o analista de requisitos é conhecido	diagramas utilizados pelo analista de requisitos	metodologia de desenvolvimento do projeto
XP (SOMMERVILLE, 2018)	Metodologia para pequenas e médias equipes, desenvolvendo software com requisitos vagos e em constante mudança.	Redator técnico.	Diagrama de Casos de Uso, Diagrama de Classes, Diagrama de Sequência ou Diagrama de Atividades.	Modelo Iterativo – incremental.
SCRUM (SOMMERVILLE, 2018)	Framework usado para gerenciar o desenvolvimento de produtos complexos.	Scrum Master.	Diagrama de Casos de Uso e Diagrama de Classes.	Modelo Iterativo.
<i>Adaptative Software Development</i> (ASD) (KEITH, 2002)	Modelo de desenvolvimento de padrões de projeto. Os requisitos são levantados através de histórias.	Engenheiro de Requisitos.	Diagrama de Casos de Uso, Diagrama de Classes e Diagrama de Atividades.	Modelo Iterativo.

<i>Feature Driven Development</i> (FDD) (SCHILLER, 2008)	Método alternativo do método cascata, fornece informações precisas e significativas acerca do progresso do desenvolvimento do sistema, com o mínimo de sobrecarga e interrupções da equipe de desenvolvimento.	Assume um ou dois papéis dependendo do projeto: Especialista do domínio e Arquiteto chefe.	Diagrama de Classes, Diagrama de Sequência (opcional).	Modelo Iterativo.
"Família" Crystal (MARQUES, 2012)	É uma família de metodologias comum "código" genético comum, que tem ênfase na entrega frequente, comunicação próxima e melhoria recíproca.	Assume um ou dois ou os três papéis dependendo do projeto: Analista de Negócios, Designer/Projetista e Redator.	Diagrama de Casos de Uso.	Modelo Iterativo-incremental.
<i>Dynamic Systems Development Method</i> (DSDM) (VOIGT, 2004)	Estrutura voltada para o gerenciamento de projetos ágeis, ajudando a entregar resultados de forma rápida e eficaz.	Analista de Negócios.	Diagrama de Estados e protótipos: negócio, usabilidade, desempenho/capacidade, capacidade/técnica.	Modelo de Iteração Funcional.

Fonte: elaborado pela autora.

Existem mais Metodologias Ágeis além das mostradas no Quadro 3; porém elas não são tão utilizadas, devido à falta de divulgação.

Referências Bibliográficas

CONCEITO de **engenharia**. 2010. Disponível em: <https://conceito.de/engenharia>. Acesso em: 31 mar. 2020.

CUNNINGHAM, W. (org.). **Manifesto for agile software development**. 2001. Disponível em: <http://agilemanifesto.org/>. Acesso em: 2 abr. 2020.

KEITH, E. R. **Agile software development process: a different approach to software design**. Nova Iorque: Universidade de Nova Iorque, 2002. Disponível em: <https://cs.nyu.edu/courses/spring03/V22.0474-001/lectures/agile/AgileDevelopmentDifferentApproach.pdf>. Acesso em: 31 mar. 2020.

MAGALHÃES, F. **Dicionário português-latim**. 13. ed. Rio de Janeiro: Edições Lep, 1960. (Lep Bolso). Disponível em: <https://docero.com.br/doc/nc181c>. Acesso em: 31 mar. 2020.

MARQUES, A. N. **Metodologias ágeis de desenvolvimento: processos e comparações**. 2012. Monografia (Tecnólogo em Processamento de Dados)– Faculdade de Tecnologia de São Paulo, São Paulo, 2012. Disponível em: www.fatecsp.br/dti/tcc/tcc00064.pdf. Acesso em: 31 fev. 2020.

MICHAELIS. **Dicionário Online Michaelis**. Disponível em: <https://michaelis.uol.com.br/>. Acesso em: 31 mar. 2020.

PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. Tradução de João Eduardo Nóbrega Tortello. 8. ed. Porto Alegre: AMGH, 2016.

SCHILLER, J. **Agile techniques for project management and software engineering**. Munique: Universidade Técnica de Munique, 2008. Disponível em: <http://csis.pace.edu/~marchese/CS616/Agile/FDD/fdd.pdf>. Acesso em: 31 fev. 2020.

SOMMERVILLE, I. **Engenharia de software**. Tradução de Luiz Cláudio Queiroz. 10. ed. São Paulo: Pearson Education do Brasil, 2018.

VOIGT, B. J. **Dynamic system development method**. Zurique: Universidade de Zurique, 2004. Disponível em: https://files.ifi.uzh.ch/rerg/arvo/courses/seminar_ws03/14_Voigt_DSMD_Ausarbeitung.pdf. Acesso em: 31 mar. 2020.

Tipos e classificação dos requisitos

Autoria: Priscila Lábamca

Leitura crítica: Douglas Fabiano Lourenço



Objetivos

- Compreender os conceitos e os tipos de requisitos de software.
- Compreender as diferenças entre os requisitos de software.
- Aprender como obter os requisitos de software.



1. Os requisitos

A palavra requisito vem do latim *virtus dis* que, segundo Michaelis (2020, [s.p.]) significa: “condição básica e necessária para se obter alguma coisa ou para alcançar determinado propósito”. Em outras palavras, tudo o que é preciso para que o software funcione.

Dentro do modelo de processo de engenharia de requisitos, os requisitos são identificados numa fase denominada fase de aquisição. Nesta fase, por sua vez, são utilizadas técnicas para conversar com o cliente com a finalidade de saber em detalhes qual é a visão que ele possui de software e como este software auxiliará nas atividades executadas em sua empresa (empresa do cliente).

Na literatura é possível observar uma quantidade considerável de requisitos, mas cuidado! A palavra requisitos é bastante genérica e, infelizmente, boa parte da literatura disponível trata tudo como requisito. Veja bem, é preciso mais do que saber ler textos literários: é preciso saber interpretar. A interpretação tanto de textos quanto de discursos é uma tarefa que requer muita atenção, caso contrário, o analista de requisitos ou qualquer outro profissional que esteja interessado em aprender sobre o tema, está fadado ao fracasso quando colocado de maneira literal tudo aquilo que está escrito na literatura.

Por este motivo, tenha em mente que existem apenas 3 (três) tipos de requisitos, sendo eles: os requisitos funcionais, os requisitos não funcionais e requisitos de negócio/organizacionais. Os demais requisitos apresentados na literatura, são na verdade, uma espécie de facilitador, ou seja, servem apenas para você diferenciar quais são as ações humanas das ações do sistema e só isso. Esses requisitos estão “embutidos” nas especificações dos três tipos de requisitos e, se analisados friamente, na verdade, eles não passam de regras de validação.

Nos próximos tópicos, apresentaremos os tipos de requisitos e realizadas as devidas considerações, respectivamente.

2. Os tipos de requisitos

Anteriormente, mencionamos que havia apenas 3 (três) tipos de requisitos: funcionais, não funcionais e organizacionais/de negócio. Esta classificação existe para que seja possível identificar e diferenciar quais aqueles que realmente comporão o software (estarão “embutidos”) no software daqueles que são acessórios aos softwares.

Então, simplificando, pense da seguinte maneira (para ficar fácil – acompanhe o texto com as Figuras de 2 a 4):

Requisitos funcionais: também conhecido como funcionalidades, funções. Para estes requisitos lembre da palavra tela (tela do software). Quais telas o software terá? Quais serão as suas dinâmicas? Quais serão os campos e como devo tratar/validar estes campos? Perceba que estes requisitos são literalmente o “coração” do software. Assim, o conjunto de todos os requisitos funcionais que resultam no software.

Requisitos não funcionais: é tudo aquilo que é preciso para que os requisitos funcionais sejam executados de maneira adequada. Logo, são os insumos: sistemas operacionais, capacidade de armazenamento de dados etc. Perceba que estes requisitos são literalmente o “corpo” que dentro dele acomodará o “coração” (requisitos funcionais).

Requisitos de negócio/organizacionais: são as normas e diretrizes de uma empresa ou de um determinado setor/departamento/seção da empresa. Por sua vez, são as missões, valores, diretrizes e metas. Perceba que estes requisitos são literalmente a “alma”, e esta alma possui um corpo (requisitos não funcionais) e um coração (requisitos funcionais).

Figura 1 – Tipos de requisitos e a sua analogia



Fonte: kovacevicmiro/iStock.com.

Obviamente que estamos tratando de uma visão lúdica (corpo, alma e coração), apenas como uma analogia quando estamos diante de um projeto de desenvolvimento de software. Geralmente, as empresas possuem mais de um sistema e uma configuração de infraestrutura. Porém, apresentamos uma visão da parte e não do todo, ou seja, o sistema que o analista de requisitos desenvolverá (sim, ele desenvolve! A solução computacional sai dele e não do programador. Programadores são “tradutores”, isto é, traduzem da linguagem técnica – UML (*Unified Modeling Language*) – para a linguagem que o computador compreende – linguagem de programação).

Estes três requisitos devem estar sempre em harmonia, não há requisitos funcionais se não houver os requisitos não funcionais e os requisitos de negócio/organizacionais. Ou seja, eles são dependentes uns dos outros. Fácil, não é mesmo? Então, vamos às definições formais de cada um deles.

Requisitos de negócio/organizacionais

Iniciando pela “alma”, os requisitos organizacionais/de negócio são aqueles que tratam das capacidades empresariais que o sistema fornecerá, ou seja, nestes requisitos estão inseridos os fatores culturais, políticos e as exigências legais que afetam o sistema. Nesse sentido, são alguns exemplos destes requisitos: é permitido aos gerentes regionais autorizar o uso de interfaces personalizadas com o usuário dentro de suas unidades, as informações pessoais estão protegidas segundo as prescrições do Banco Central do Brasil etc.

Figura 2 – Exemplos de requisitos de negócio/organizacionais



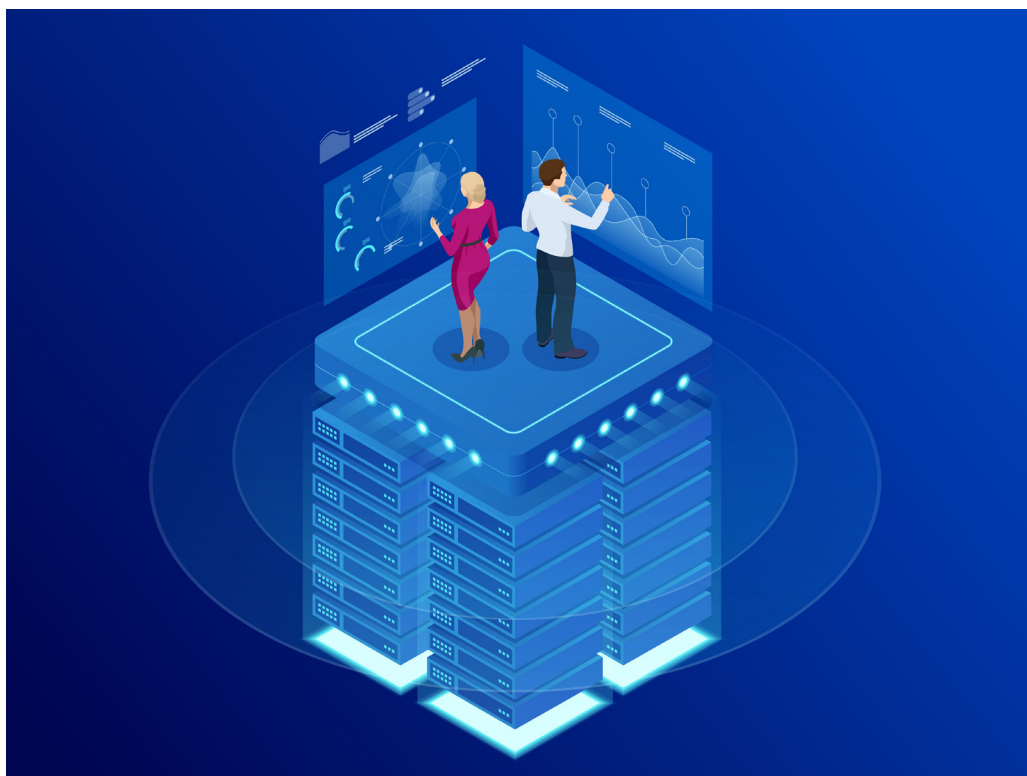
Fonte: nythzl/iStock.com.

Requisitos não funcionais

Já o “corpo”, isto é, os requisitos não funcionais são aqueles que tratam das restrições de um sistema ou em um de seus componentes. Eles se caracterizam por tratar sobre aspectos de desempenho e possuem

uma classificação, cada qual trata de algum ponto específico e está intimamente conectado aos requisitos organizacionais/negócio.

Figura 3 – Exemplo de requisitos não funcionais



Fonte: Ostapenko Olena/iStock.com.

Além disso, existem algumas classificações de requisitos não funcionais, alguns exemplos são apresentados no quadro a seguir:

Quadro 1 – Alguns exemplos de classificação dos requisitos não funcionais

Classificação	Descrição	Exemplos
Desempenho	Trata da velocidade, confiabilidade e capacidade do sistema.	<ul style="list-style-type: none">- O sistema deve suportar 300 usuários simultâneos das 9-11 horas da manhã.- O sistema não deve ultrapassar o tempo de qualquer interação entre o usuário e o sistema por mais de 2 segundos.

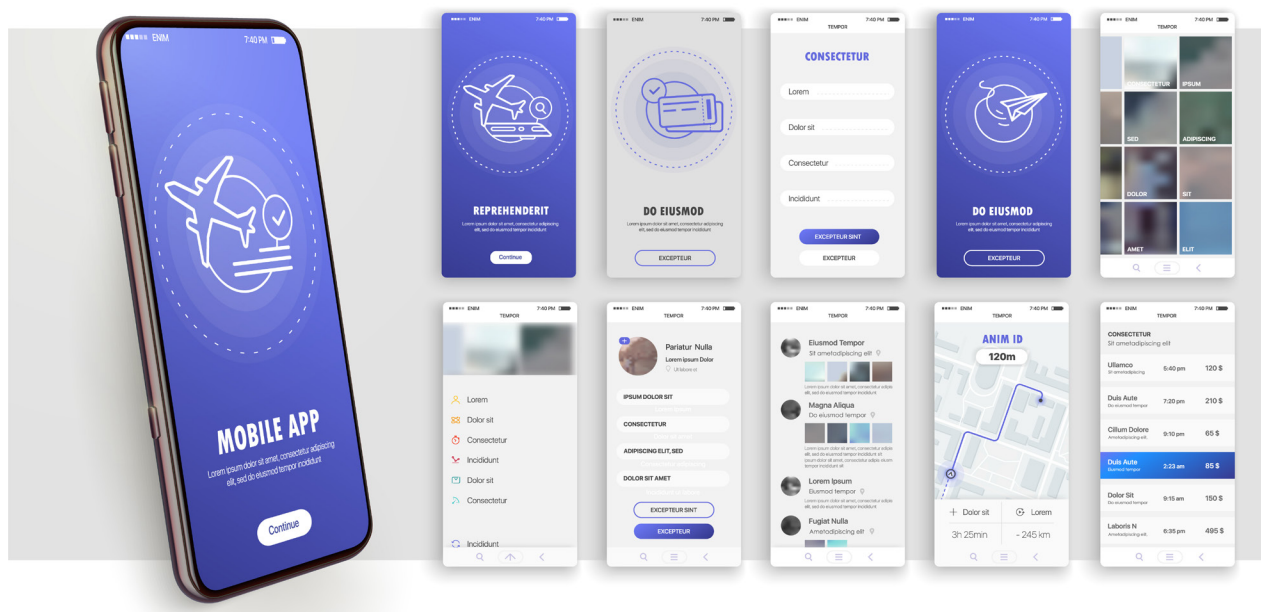
Operacional	Trata dos ambientes físico e técnico no qual o sistema será instalado.	<ul style="list-style-type: none"> - O sistema deve ser capaz de ser executado em qualquer tipo de dispositivos portáteis. - O sistema deve ser capaz de integrar com o sistema de almoxarifado existente.
Segurança	Trata das questões de acesso e suas respectivas circunstâncias.	<ul style="list-style-type: none"> - O sistema incluir todos os dispositivos de segurança contra vírus, <i>worms</i> e cavalos de Troia (<i>trojan</i>, <i>horses</i>) etc. - Apenas os gerentes diretos podem ver os registros pessoais dos funcionários.

Fonte: elaborado pela autora.

Requisitos funcionais

Finalmente, o “coração” (requisitos funcionais) é composto por aqueles que definem as funções de um sistema ou componente de um sistema e como devem funcionar. Desse modo, são descritas as transformações a serem realizadas nas entradas de um sistema ou em um de seus componentes, a fim de que se produzam saídas. Portanto, este requisito está intimamente conectado aos requisitos organizacionais/negócio (missão, valores e diretrizes do cliente) e aos requisitos não funcionais (sistemas operacionais, questões de segurança da informação etc.).

Figura 4 – Exemplo de requisitos funcionais



Fonte: vector_s/iStock.com.

Quando identificados, estes requisitos devem ser descritos, e não especificados, pois a especificação é outra atividade que depende fortemente desta da maneira mais completa e consistente possível. Por exemplo, o sistema fornecerá um conjunto de telas para que o usuário tenha acesso aos documentos armazenados no repositório de documentos.

Os requisitos funcionais possuem uma classificação (que na literatura são chamados também de requisitos, porém, como mencionado anteriormente, devemos tomar cuidado, pois **servem apenas para você diferenciar quais são as ações humanas das ações do sistema** dentro da dinâmica de execução do software). No Quadro 2, apresentamos algumas classificações de requisitos funcionais.

Quadro 2 – Alguns exemplos de classificação dos requisitos funcionais

Classificação	Descrição	Exemplos
Requisitos de usuário	São declarações escritas em linguagem natural (Português, Inglês, Alemão etc.), que descrevem quais ações o software deverá conter para que seja possível a interação entre ele e o usuário.	<ul style="list-style-type: none"> - O sistema deve conter uma tela de cadastro de produtos. - O sistema deve conter uma tela para inserir dados de fornecedores.
Requisitos de sistema	São declarações escritas em linguagem natural (Português, Inglês, Alemão etc.), e que, dependendo do software a ser desenvolvido, poderão conter fórmulas matemáticas. Eles descrevem os detalhes de cada um dos requisitos de usuário que podem ser validações ou regras de negócio a serem tratadas pelo software.	<ul style="list-style-type: none"> - O sistema deve validar o número de CPF após acionar a opção Consultar. - O sistema deve verificar se o campo nome contém apenas letras. Caso contrário, deverá apresentar a mensagem: por favor, digite apenas letras no campo nome.
Requisitos orientados a processos	São aqueles que tratam de processos que o software deve executar.	<ul style="list-style-type: none"> - O sistema deve permitir que os clientes registrados revejam o histórico de pedidos nos últimos três anos. - O sistema deve permitir que os alunos vejam a programação de um curso quando estiverem se matriculando nas disciplinas.

Requisitos orientados por informações	São as informações que o sistema deve conter.	<ul style="list-style-type: none"> - O sistema deve conservar o histórico dos pedidos dos clientes por três anos. - O sistema deve manter um cadastro de clientes para que as faturas/boletos de pagamento sejam enviadas através dos Correios.
---------------------------------------	---	---

Fonte: elaborado pela autora.

Até agora, aprendemos o que são requisitos, suas definições, tipos e características, aparentemente é bastante fácil, mas, será mesmo? Como identificamos os requisitos? O próximo capítulo apresentará dicas de como fazer isto.

3. Como identificar os requisitos?

Os requisitos são identificados a partir da conversa com o cliente. Nesse sentido, para que esta conversa flua de maneira adequada utilizamos algumas técnicas denominadas técnicas de levantamento de requisitos. Elas podem ser utilizadas individualmente, ou seja, é possível utilizar apenas uma técnica para obter os requisitos ou ainda é possível combiná-las. Isso dependerá da complexidade e perfil do software que será construído.

Para isso, relacionamos a seguir algumas boas práticas para facilitar a obtenção dos requisitos.

1. A obtenção dos requisitos tem por objetivo descobrir e entender problemas, pois a solução será dada na atividade de modelagem. Você só é capaz de realmente entender os problemas do cliente quando demonstrar que conhece esses requisitos, logo, procure extrair o máximo de conhecimento dele, utilizando técnicas adequadas e não “na orelhada”. Quanto mais você praticar sua atenção, senso crítico e interpretação de discurso melhor será a obtenção dos requisitos e, conseqüentemente, maiores chances de sucesso na modelagem e posteriormente na “tradução” para uma linguagem de programação.
2. Os clientes costumam confundir aquilo de que precisam com o que desejam (acredite, eles não fazem isso de propósito). Durante a fase de aquisição de requisitos (ou levantamento dos requisitos ou ainda elicitação de requisitos), por meio de questionamentos e entendimento, será possível separar o necessário do desejável.
3. Sempre documente os problemas e as necessidades do cliente e peça para validar a sua precisão e completude. Esse é o meio de demonstrar que você entende as necessidades do cliente (e também demonstra profissionalismo).
4. Analise cuidadosamente os requisitos para que não haja conflito entre eles. Uma sugestão é elaborar mapas de requisitos, em que a estrutura pode ser semelhante ao Quadro 3. Para utilizar o mapa de requisitos, você precisa da lista de requisitos que obteve durante as reuniões com o cliente. Primeiramente, você cria a lista de requisitos, na frente de cada um deles, você pode dar um “apelido”, por exemplo: REQ1, REQ2.... Após isso, você completa o mapa de requisitos. Este mapa, além de auxiliar na identificação de conflitos entre requisitos, dá a oportunidade de visualizar melhor os relacionamentos entre eles (os requisitos). Assim, você também pode utilizar ferramentas apropriadas para isso, existem algumas muito boas oferecidas comercialmente, ou simplesmente abrir um arquivo em planilha eletrônica. Normalmente, o mapa de requisitos é utilizado apenas para requisitos funcionais, porém


nada impede que o analista de requisitos elabore um mapa para cada um dos demais tipos de requisitos.

Quadro 3 – Exemplo de mapa de requisitos funcionais

Requisitos	REQ1	REQ2	REQ3	REQ4
REQ1	– 0 –	Depende de	Não depende de	Depende de
REQ2	Depende de	– 0 –	Depende de	Não depende de
REQ3	Depende de	Depende de	– 0 –	Depende de
REQ4	Não depende de	Depende de	Não depende de	– 0 –

Fonte: elaborado pela autora.

5. Quantifique “quão bem” o cliente espera a funcionalidade do sistema como parte dos requisitos. Quanto mais detalhes o cliente fornece sobre os requisitos, menores serão as chances de retrabalho no transcorrer do ciclo de desenvolvimento do software. Aqui, é preciso tomar a atenção e garantir que todos os requisitos identificados devem ser testáveis.
6. Formule os requisitos de maneira clara, objetiva e livre de ambiguidades. Desta forma, você elimina o “retrabalho” e facilita a entrega do produto no prazo. Simplificando, a formulação deve seguir a nomenclatura: sujeito + verbo + complemento. Um exemplo de escrita “ideal” pode ser: o sistema deve conter uma tela para inserir dados de fornecedores. Os detalhes serão especificados em outro momento, após a validação desta primeira atividade (na fase de especificação de requisitos).
7. Quando houver pressão para mudanças de requisitos (funcionais), antes de prosseguir, envolva o gerente de projeto para fazer uma



avaliação do impacto e custo da alteração. Não queira levar o mundo em suas costas! Todos fazem parte do desenvolvimento do software e existem certos momentos que devemos deixar que o gerente de projeto decida e não o analista de requisitos, certo?

8. Para executar as atividades de coleta e documentação dos requisitos, procure pessoas que trabalham diretamente com o cliente e as convide para ajudar a identificar os requisitos. Verifique se estas pessoas sabem “escutar, falar e escrever” bem, elas não precisam ser doutores em letras, mas precisam saber se expressar bem. Isso facilitará o trabalho de compreensão e melhorará ainda mais a execução das próximas fases que você deve atingir dentro do modelo de processos de requisitos.
9. Procure trabalhar com requisitos que julga serem “bons o suficiente”. A busca pela perfeição pode aumentar riscos e ocasionar impactos.
10. Se você assumir um projeto que está em andamento ou numa fase avançada, é interessante não confiar cegamente nos conhecimentos passado pela equipe de projetos. Por mais que eles estejam engajados no projeto, muitas vezes, deixam-se passar alguns detalhes que podem ser muito importantes para a realização do trabalho. Outro motivo para não confiar cegamente é que ninguém sabe em maiores detalhes sobre o projeto do que o cliente. Neste caso, não custa nada convidá-lo para algumas reuniões e sanar o máximo de dúvidas possível. Com esta atitude, você ganhará a confiança do cliente e do superior imediato (gerente de projeto) e, ainda, garantirá a qualidade no desenvolvimento do projeto.

Observe que, com atitudes como as elencadas nos 10 (dez) passos, é possível ter sucesso em todos os sentidos dentro do projeto de construção de um sistema de informação; e ainda, poupar o retrabalho e eventuais desconfortos tanto com a equipe que está inserido quanto com o cliente.

Nesse sentido, somente se alcança a excelência praticando massivamente estes conceitos e dicas. Lembre-se que o sucesso não vem na mesma velocidade que uma mensagem de WhatsApp, o sucesso é planejado e atingido paulatinamente (por que você acha que os japoneses, chineses, indianos são tão bons no que fazem? É porque eles treinam, estudam, se dedicam; eles não são gênios, eles são pessoas como você).

Neste momento, você deve ter percebido o quão cauteloso e responsável é preciso que o analista de requisitos seja quando se trata de executar as atividades do modelo de processo de requisitos.

Não é à toa que, frequentemente, nos deparamos com sistemas (seja ele qual for: sistemas do governo, sistemas de instituições financeiras etc.) que possuem erros ou são incongruentes ou não possuem boa navegabilidade. Isso tudo ocorre pela falta de atenção na execução das atividades contidas no modelo de processo de requisitos.

Referências Bibliográficas

FAULK, S. R. Software requirements: a tutorial. **Naval Research Laboratory**, Washington, p. 1-36, 14 nov. 1995. Disponível em: https://www.researchgate.net/publication/235192346_Software_Requirements_A_Tutorial. Acesso em: 6 abr. 2020.

MAGALHÃES, F. **Dicionário Português-Latim**. 13. ed. Rio de Janeiro: Edições Lep, 1960. (Lep Bolso). Disponível em: <https://docero.com.br/doc/nc181c>. Acesso em: 31 mar. 2020.

MICHAELIS. **Dicionário Online Michaelis**. Disponível em: <https://michaelis.uol.com.br/>. Acesso em: 31 mar. 2020.

PRESSMAN, R. S. **Engenharia de software**: uma abordagem profissional. Tradução de João Eduardo Nóbrega Tortello. 8. ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, I. **Engenharia de software**. Tradução de Luiz Cláudio Queiroz. 10. ed. São Paulo: Pearson Education do Brasil, 2018.

Levantamento de requisitos

Autoria: Priscila Lábamca

Leitura crítica: Douglas Fabiano Lourenço



Objetivos

- Adquirir noções sobre o processo de comunicação.
- Conhecer as técnicas de levantamento de requisitos e suas respectivas dinâmicas.
- Aprender a especificar requisitos.



1. O levantamento de requisitos

A atividade de elicitación (ou levantamento) de requisitos é pautada na comunicação entre o cliente e o analista de requisitos. A comunicação advém da palavra em latim *comunicatus*, que significa “ato que envolve a transmissão e a recepção de mensagens entre o transmissor e o receptor, através da linguagem oral, escrita ou gestual, por meio de sistemas convencionados de signos e símbolos” (MICHAELIS, 2020, [s.p.]).

Assim, é possível perceber tanto em nossas relações pessoais quanto em nossas relações profissionais, que a estrada da comunicação para a compreensão nem sempre é uma via de boa pavimentação, às vezes, há algumas crateras ou buracos ocasionando más interpretações. Para amenizar a falta de compreensão no processo de comunicação, é preciso convidar os envolvidos para uma conversa. No caso das relações entre cliente e analista de requisitos, esta primeira conversa pode parecer um pouco desconfortável, afinal, vocês não se conhecem pessoalmente e é natural que haja algumas resistências. Neste caso, seria interessante fazer uma analogia com o processo de conquistar o coração de alguém. Por mais tenso que seja o primeiro encontro entre o cliente e o analista de requisitos, é preciso criar alguns mecanismos para “quebrar o gelo”, fazendo com que o cliente fique à vontade para falar e, aos poucos, obter confiança no analista de requisitos. Para isso, é preciso “criar um clima” e este clima é conhecido como técnicas de reunião com o cliente (ou usuário).


1.1 As técnicas de reunião com o cliente

Nesse sentido, há várias técnicas para conduzir reuniões, e elas podem ser classificadas em um dos grupos a seguir.

Quadro 1 – Classificação das técnicas de reunião

Classificação	Descritivo
Cenários	São situações que simulam a execução de tarefas. Podem ser divididos em dois momentos: o primeiro, “fazer de conta” que o cliente está executando as atividades do dia a dia sem o software, então ele vai descrevendo (contando uma história) estas atividades; no segundo momento, eles simulam estas mesmas atividades só que colocadas num software “imaginário”.
Técnicas tradicionais	São técnicas utilizadas mais comumente. Elas são utilizadas em todos os setores empresariais e foram absorvidas no processo de desenvolvimento de software.
Técnicas de elicitação de grupo	São técnicas que envolvem grupos de pessoas.
Prototipação	Utilizada principalmente quando existe alto grau de incerteza e necessita de rápido feedback. Normalmente, é utilizada durante todo o processo de levantamento de requisitos, quando percebe-se que o cliente possui dificuldades em comunicar-se.
Técnicas cognitivas	São técnicas utilizadas para aquisição de conhecimento para desenvolver sistemas baseados em conhecimento (inteligência artificial).
Técnicas contextuais	São técnicas que analisam a etnografia e a análise social.

Fonte: elaborado pela autora.



Esta classificação está inserida num processo denominado como processo de levantamento de requisitos, que pode ser dividida em dois momentos:

O **primeiro momento** é marcado com as primeiras reuniões. Neste contexto, são realizadas reuniões que objetivam conhecer a dinâmica do dia a dia do cliente, a visão que ele possui do software que será construído e o relacionamento entre a sua dinâmica e o software. Este levantamento pode durar até 3 (três) reuniões, isso depende da complexidade do projeto e do tempo total estipulado pelo gestor de projetos. Aqui, o analista de requisitos precisa saber administrar este tempo total para que não extrapole e, conseqüentemente, colocando o projeto em risco.

O resultado final deste momento é uma lista de requisitos que devem ser especificados em alto nível, ou seja, ele deve ser livre de detalhes. Antes de seguir para o segundo momento, o analista de requisitos identifica e classifica na lista de requisitos adquirida, os tipos de requisitos (requisitos funcionais, requisitos não funcionais e requisitos de negócio/ organizacionais). Uma vez identificados e classificados, eles são colocados em listas separadas ou são acomodados em algum capítulo dentro do documento de requisitos (a forma como devem ser distribuídos os capítulos fica a cargo do analista de requisitos ou em um template já definido pela empresa de TI em que o analista de requisitos trabalha). Finalizada esta atividade, o analista de requisitos retorna à lista de requisitos funcionais e faz um mapa de requisitos (este mapa facilita a compreensão do analista de requisitos e dá a ideia de quais requisitos ele poderá trabalhar no segundo momento). Estes requisitos podem ser trabalhados individualmente ou em grupo, tudo dependerá da complexidade do projeto.

No **segundo momento** são apresentados os requisitos funcionais para obter mais detalhes. Observe que este momento também possui um ciclo: enquanto não se esgotar a especificação de todos os detalhes de todos os requisitos funcionais, não é possível passar para a atividade de modelagem (isto do ponto de vista de metodologias tradicionais; caso o analista de requisitos esteja inserido numa equipe onde foi estabelecido

que a dinâmica de trabalho (metodologia) seja ágil, então o analista de requisitos deverá eleger melhor os requisitos/grupo de requisitos para que seja executada a metodologia ágil de maneira adequada).

Assim como no primeiro momento, aqui também são utilizadas as técnicas de levantamento de requisitos. Dependendo da complexidade do projeto, é possível combinar as técnicas ou ainda, utilizar uma ou mais técnicas no primeiro momento e no segundo momento, utilizar outras técnicas. Quem decide é o analista de requisitos.

Os requisitos funcionais neste momento são especificados em baixo nível, ou seja, eles são detalhados.

Independentemente do momento em que o analista de requisitos se encontre dentro do processo de levantamento de requisitos, ele pode optar por uma ou mais técnicas. As técnicas utilizadas são executadas de acordo com o perfil e complexidade do projeto. Como existem diversas técnicas, no quadro a seguir você pode verificar algumas que se imagina que sejam mais utilizadas.

Quadro 2 – Algumas técnicas de reunião

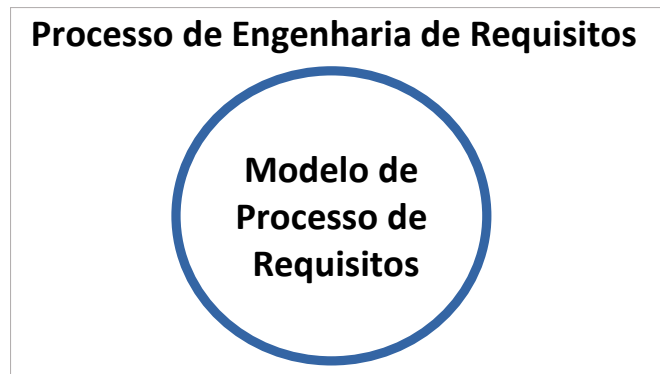
Nome da técnica	Definição	Classificação
Brainstorm	É uma técnica não estruturada para geração de ideias que consiste em duas fases: 1 — Geração de ideias: ideias são apresentadas sem discussão. 2 — Consolidação: ideias são discutidas, revisadas e organizadas.	Técnicas de elicitación de grupo.
Entrevista	É uma discussão do projeto desejado com diferentes grupos de pessoas.	Técnicas tradicionais.

PIECES	<p>Conjunto de categorias de perguntas que ajudam na extração de requisitos.</p> <p>P→ performance (tempo de resposta).</p> <p>I→ informações e dados.</p> <p>E→ economia (questões de demanda).</p> <p>C→ controle (acesso).</p> <p>E→ eficiência (custo-benefício).</p> <p>S→ serviços (que tipo de serviço é necessário).</p>	Técnicas contextuais.
Prototipação	Provê a avaliação das interfaces junto aos clientes, com o auxílio de técnicas apropriadas (usabilidade).	Prototipação.
Questionário	As questões são dirigidas por escrito aos participantes com o objetivo de ter conhecimento sobre suas opiniões. Podem ou não ser autoaplicável ou ainda pode ser combinada com a técnica de Entrevista.	Técnicas tradicionais.
Roleplaying	Determina os atores, explica o que acontece com eles e descreve a forma como isso acontece.	Cenários.

Fonte: elaborado pela autora.

Neste ponto, pode ser que você esteja um pouco confuso, afinal, temos o processo de engenharia de requisitos, a fase de aquisição, o processo de elicitação (levantamento de requisitos) e tantas outras coisas. Acompanhe as Figuras de 1 a 4 e as respectivas considerações, você se sentirá mais seguro.

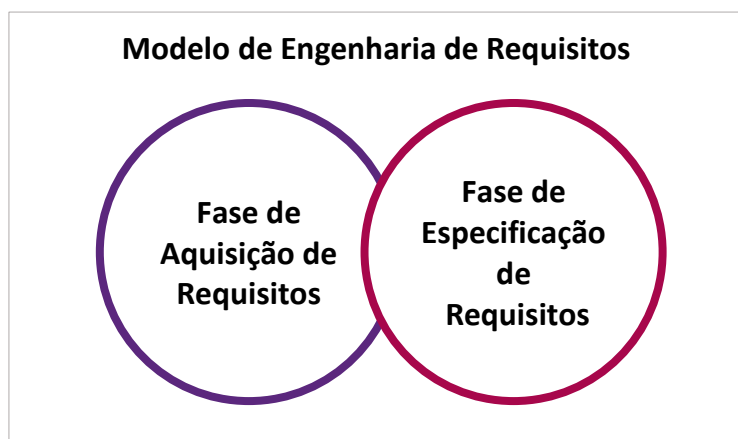
Figura 1 – Modelo macro



Fonte: elaborado pela autora.

Imagine um grande conjunto chamado de processo de engenharia de requisitos. Dentro deste conjunto existe um outro conjunto denominado modelo de engenharia de requisitos que, por sua vez, contém outros dois conjuntos (Figura 2).

Figura 2 – Modelo Intermediário

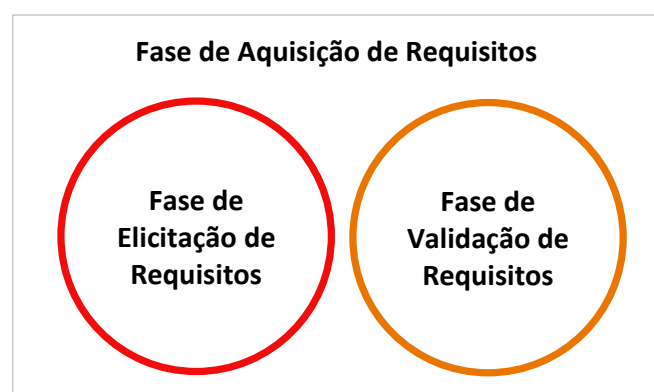


Fonte: elaborado pela autora.

A fase de aquisição de requisitos é composta pelas atividades de elicitação de requisitos e validação de requisitos (Figura 3). Nesta fase, o analista de requisitos executa o processo de levantamento de requisitos que é composto por técnicas e elas podem ser utilizadas individualmente ou combinadas. A utilização das técnicas combinadas

ou não resulta numa lista de requisitos que ainda não foram classificados. Eles serão classificados após o término da reunião (esta tarefa é somente do analista de requisitos fazer, pois o cliente não tem a mínima ideia de que existe classificação de requisitos e muito menos o que são requisitos). Após a identificação e classificação de requisitos, são executadas as atividades já descritas em parágrafos anteriores.

Figura 3 – Modelo intermediário-micro

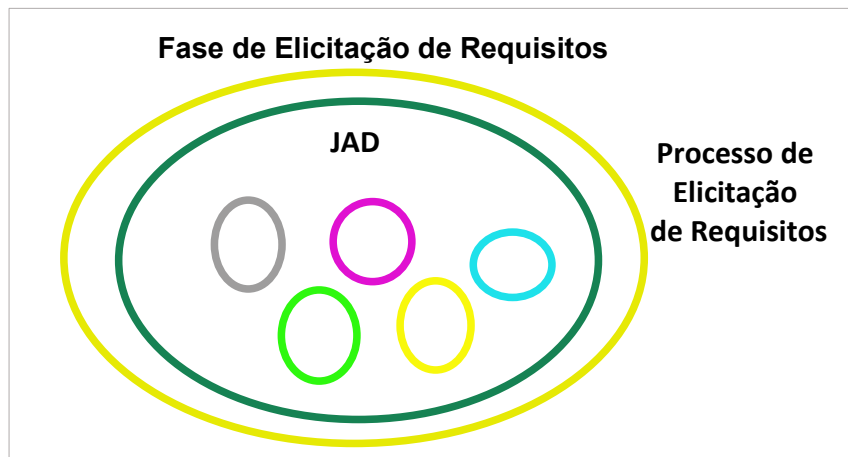


Fonte: elaborado pela autora.

Já a fase de especificação de requisitos é composta pelas atividades de análise e modelagem de requisitos, que serão apresentadas em detalhes na próxima leitura digital. A intersecção entre as duas fases é a lista de requisitos obtidos pelo analista de requisitos.

O processo de levantamento de requisitos possui um conjunto de técnicas para obtenção dos requisitos, porém elas estão inseridas num conjunto maior que nada mais é que um método para planejar e organizar reuniões. Assim, há vários métodos, porém, existe um bastante usual (que boa parte não só dos analistas de requisitos, mas boa parte dos profissionais de quaisquer outras áreas utilizam e nunca sabem o nome) chamado JAD (*Joint Application Design*), ilustrado na Figura 4, onde cada círculo colorido (conjunto) representa uma técnica de levantamento de requisitos.

Figura 4 – Modelo micro



Fonte: elaborado pela autora.

A metodologia JAD (*Joint Application Design*) é, segundo Pimentel (2007, p. 3), o “[...] conjunto de técnicas para promover cooperação, entendimento e trabalho em equipe entre usuários e desenvolvedores a fim de se obter uma melhor extração de requisitos”.

Logo, o objetivo é chegar a um consenso sobre os requisitos obtidos. Cada encontro (reunião) é definido por uma agenda que é disponibilizada ao cliente antes da realização da reunião. Esta agenda deve ser seguida com o máximo rigor possível.

Portanto, a metodologia JAD requer um ritual de execução composto por 5 (cinco) etapas, cada qual com seus objetivos e procedimentos, conforme Quadro 3.

Quadro 3 – Etapas da metodologia JAD

Etapa	Nome da etapa	Objetivo (s) da etapa	Como proceder?
1	Orientação inicial.	Obter os requisitos em alto nível.	O Analista de Requisitos deve: a) Revisar o Documento de Escopo do Projeto. b) Verificar as áreas envolvidas (cliente e a equipe de TI).
2	Familiarização com a área/aplicação.	Analisar os procedimentos atuais do negócio do cliente ou do departamento do negócio do cliente que o software será utilizado.	Documentar os procedimentos, dando ênfase: - Aos dados de entrada e saída. - A dinâmica dos dados de entrada e saída. - A descrição do processo que acomoda os dados de entrada e saída.

3	Preparação do material para a reunião (ou workshop).	Elaborar o material para ser levado à reunião.	<ul style="list-style-type: none"> - Esboçar de acordo com o Documento de Escopo do Projeto, algumas telas para facilitar a comunicação. - Eleger o local para a realização da reunião - Obter material de apoio (caneta, blocos de anotações, fita, <i>flowcharts</i>, canetas para flowcharts). - Elaborar agenda de convocação, relatório de avaliação, plano de sessão, carta (e-mail) de convocação. - Definir a técnica de elicitação de requisitos.
4	Reunião (ou workshop).	Realizar a reunião efetivamente.	<ul style="list-style-type: none"> - Informar aos participantes sobre as regras de condução da reunião, seus respectivos papéis na reunião e o tempo de duração da reunião. - Iniciar efetivamente a reunião (aplicar as técnicas de elicitação de requisitos).

5	Eleição das pessoas que deverão comparecer à reunião (workshop).	Verificar quem são as pessoas que devem e as pessoas que podem participar da reunião.	<p>- Quem deve participar da reunião: cliente e sua equipe ou algum representante de sua equipe, Analista de Requisitos (geralmente faz o papel de mediador e escrita, mas se houver a possibilidade de ter um escriba, melhor).</p> <p>- Quem pode participar da reunião: Gerente de Projetos, Pessoal da equipe de TI, Diretor da empresa (Patrocinador), outras pessoas que achar conveniente convidar, porém estas pessoas somente assistirão a reunião, não poderão em hipótese alguma dar palpites.</p> <p>- Esta etapa pode ser absorvida pela etapa de Preparação do material para a reunião (ou workshop).</p>
---	--	---	---

Fonte: elaborado pela autora.

A dinâmica pormenorizada das reuniões, como elaborar materiais e a agenda podem ser conferidas nos trabalhos de: Rottmann (2001), Paludo (2004), David (2002) e Mushtaq (2016).

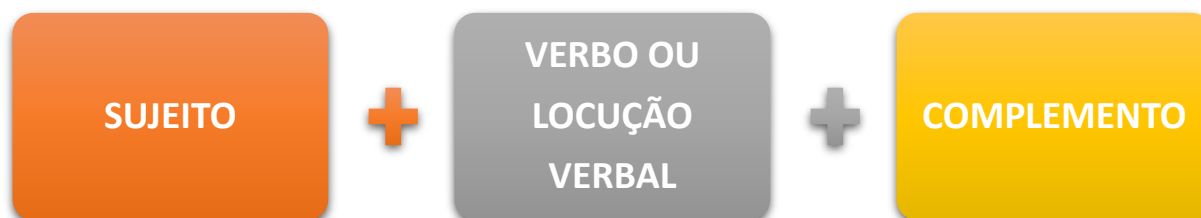
► 2. A especificação dos requisitos

Até aqui, apresentamos as técnicas e os procedimentos para executar as técnicas de elicitação de requisitos e as atividades a serem realizadas após a execução das técnicas. Diante disso, é o momento de especificar todos os requisitos que foram obtidos.

Conforme informado, neste momento os requisitos obtidos são especificados no mais alto nível, ou seja, aqui não são verificados ainda os detalhes, deixando esta atividade para ser executada na fase de especificação de requisitos.

Nesse sentido, a forma de especificar qualquer tipo de requisitos é a seguinte:

Figura 5–Forma de especificar um requisito



Fonte: elaborada pela autora.

Desta forma, você garante a completude, a consistência, a coerência e a coesão. As chances de você errar na especificação são mínimas ou não existentes. Logo, a qualidade do software a ser desenvolvido inicia no trabalho do analista de requisitos e finaliza com o analista de testes.

Para cada tipo de requisitos, há alguns exemplos de como utilizar a forma de especificação:

Requisitos Funcionais (RF):

- Exemplo 1: o sistema deve conservar o histórico dos pedidos dos clientes por três anos.

Quadro 4–Tipo de RF: orientado por informações

Sujeito	Verbo OU locução verbal	Complemento
O sistema	Deve conservar	O histórico dos pedidos dos clientes por três anos.

Fonte: elaborada pela autora.

- Exemplo 2: o sistema deve permitir que os alunos vejam a programação de um curso quando estiverem se matriculando nas disciplinas.

Quadro 5–Tipo de RF: orientado a processos

Sujeito	Verbo OU locução verbal	Complemento
O sistema	Deve permitir	Que os alunos vejam a programação de um curso quando estiverem se matriculando nas disciplinas.

Fonte: elaborada pela autora.

Requisitos Não Funcionais (RNF):

- Exemplo 1: o sistema deve suportar 300 usuários simultâneos das 9 às 11 horas da manhã.

Quadro 6–Tipo de RNF: desempenho

Sujeito	Verbo OU locução verbal	Complemento
O sistema	Deve suportar	300 usuários simultâneos das 9 às 11 horas da manhã.

Fonte: elaborada pela autora.

- Exemplo 2: o sistema deve ser capaz de ser executado em qualquer tipo de dispositivos portáteis.

Quadro 7–Tipo de RNF: operacional

Sujeito	Verbo OU locução verbal	Complemento
O sistema	Deve ser	Capaz de ser executado em qualquer tipo de dispositivos portáteis.

Fonte: elaborada pela autora.

Requisitos de Negócio/Organizacionais (RO):

- Exemplo 1: o sistema deve produzir relatórios de gestão.

Quadro 8–Exemplo de RO: sobre relatórios gerenciais

Sujeito	Verbo OU locução verbal	Complemento
O sistema	Devem produzir	Relatórios de gestão.

Fonte: elaborada pela autora.

- Exemplo 2: o sistema deve ser capaz de fazer a distinção entre a moeda americana e as moedas de outras nações.

Quadro 9–Exemplo de RO: sobre câmbio

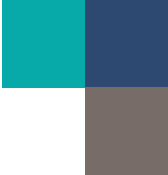
Sujeito	Verbo OU locução verbal	Complemento
O sistema	Deve ser	Capaz de fazer a distinção entre a moeda americana e as moedas de outras nações.

Fonte: elaborada pela autora.

Lembrando que os requisitos funcionais também possuem uma classificação (que na literatura são chamados também de requisitos). Desse modo, devemos tomar cuidado, pois **servem apenas para você diferenciar quais são as ações humanas das ações do sistema** dentro da dinâmica de execução do software.

Referências Bibliográficas

- DAVID, L. **Técnicas para reuniões de JAD (Joint Application Design)**. 2002. Disponível em: <https://slideplayer.com.br/slide/14019/>. Acesso em: 13 abr. 2020.
- MAGALHÃES, F. **Dicionário português-latim**. 13. ed. Rio de Janeiro: Edições Lep, 1960. (Lep Bolso). Disponível em: <https://docero.com.br/doc/nc181c>. Acesso em: 31 mar. 2020
- MICHAELIS. **Dicionário Online Michaelis**. Disponível em: <https://michaelis.uol.com.br/>. Acesso em: 31 mar. 2020.
- MUSHTAQ, J. Different requirements gathering techniques and issues. **International Journal Of Scientific & Engineering Research**, EUA, p. 835-840, set. 2016. Disponível em: <https://www.ijser.org/researchpaper/Different-Requirements-Gathering-Techniques-and-Issues.pdf>. Acesso em: 13 abr. 2020.
- PALUDO, M. **O desenvolvimento de software aplicando a técnica joint application design**. 2004. Disponível em: https://www.academia.edu/1317486/O_Desenvolvimento_de_Software_Aplicando_a_T%C3%A9cnica_Joint_Application_Design. Acesso em: 13 abr. 2020.
- PIMENTEL, A. Vygotsky: uma abordagem histórico-cultural da educação infantil. In: OLIVEIRA-FORMOSINHO, J.; KISHIMOTO, T. M.; PINAZZA, M. A. (orgs.) **Pedagogia(s)**



da infância: dialogando com o passado, construindo o futuro. Porto Alegre: Artmed, 2007. Cap. 9.

PRESSMAN, R. S. **Engenharia de software:** uma abordagem profissional. Tradução de João Eduardo Nóbrega Tortello. 8. ed. Porto Alegre: AMGH, 2016.

ROTTMANN, D. **Joint Application Development (JAD)**. 2001. Disponível em: https://www.umsl.edu/~sauterv/analysis/488_f01_papers/rottman.htm. Acesso em: 13 abr. 2020.

Análise, validação, mudança e gerência de requisitos

Autoria: Priscila Lábamca

Leitura crítica: Douglas Fabiano Lourenço



Objetivos

- Aprender a analisar e formatar os requisitos.
- Conhecer como é o processo de validação dos requisitos.
- Conhecer o processo de avaliação de mudanças em requisitos.
- Conhecer o processo de gerência de requisitos.




1. A análise de requisitos

O processo de engenharia de requisitos está pautado num modelo denominado modelo do processo de requisitos, que possui duas grandes fases: fase de aquisição de requisitos e fase de elicitação de requisitos. A primeira fase é composta pelas atividades de elicitação (ou levantamento de requisitos) e validação. Já a fase de especificação de requisitos é composta pelas atividades de análise e modelagem de requisitos.

Afinal, o que é análise? Análise é um procedimento cíclico que possui 3 (três) etapas: derivar, validar e documentar. Nesse sentido, derivar significa quebrar e dividir os requisitos para que seja possível detalhá-los e, posteriormente, validá-los (com o cliente), documentá-los e, posteriormente, modelá-los. O núcleo da atividade de análise de requisitos é identificar as necessidades dos usuários e possui as seguintes tarefas: avaliar a concepção do sistema quanto a sua exequibilidade, criar definições de sistema que constitua a base para todo o trabalho das atividades subsequentes, atribuir funções ao hardware, pessoas, banco de dados etc.

A partir desta atividade de análise, gera-se um documento onde são elencados todos os requisitos para a construção do sistema. Este documento é normalmente chamado de documento de especificação de requisitos e fundamentado nos documentos de escopo do projeto (aquele redigido pelo gerente de projetos e o cliente) e documento de requisitos do projeto (aquele que o analista de requisitos criou durante a fase de aquisição de requisitos). Nesta fase, o documento criado será denominado documento de especificação de casos de uso.

Além disso, há vários tipos de análise, cada qual com suas características e dinâmicas. O tipo de análise que será apresentado é o tipo análise



orientado a objetos, pois este é o tipo mais comum. Basicamente, este tipo de análise consiste em organizar e classificar os requisitos em grupos; a reunião destes grupos representa o problema a ser resolvido. Após a organização e classificação, eles são verificados como estes grupos se relacionam e interagem uns com os outros.

Apenas um ponto de atenção antes de continuar: os requisitos que trabalhamos na fase de elicitação de requisitos são os requisitos funcionais, ok? Pois somente os requisitos funcionais são passíveis de detalhamento e descrição, são aqueles que realmente serão “traduzidos” posteriormente para uma linguagem de programação.

Uma boa maneira de organizar e classificar os requisitos (funcionais) em grupos é visitando a matriz (ou mapa) de requisitos, construído na fase de aquisição de requisitos. Esta organização consiste na seguinte estrutura: nome do grupo que o requisito funcional faz parte, os elementos que contribuem para os requisitos funcionais serem executados e as ações que estes requisitos funcionais realizam.

Dica: faça uma analogia com a teoria dos conjuntos da matemática, isto é, reunir elementos que possuem características comuns, por exemplo, o grupo das frutas, o grupo das árvores etc. Por exemplo, suponha que um dos requisitos é realizar o cadastro de fornecedores (este cadastro sofrerá de tempos em tempos atualizações, certo? Que tal chamar este grupo de manter fornecedor?). Então, neste caso, observando os requisitos funcionais elencados no documento de requisitos, e o mapa de requisitos, chegamos a seguinte conclusão:

Grupo: Manter Fornecedor

- **Ações que estes requisitos funcionais realizam:** Inserir Fornecedor, Editar (ou Alterar) Fornecedor, Excluir Fornecedor, Pesquisar Fornecedor e Emitir Relatório de Fornecedor (este último

tem como saída os nomes e os detalhes de cadastro de todos os fornecedores).

- **Elementos para a execução (também podem ser chamados de atributos):** nome, CNPJ, número de telefone, inscrição estadual etc.

Agora, você se põe a perguntar: e os demais detalhes, tipo, validação de CNPJ, como será a tela, quais serão os nomes dos botões? Estes detalhes serão inseridos em um documento denominado documento de especificação de casos de uso. Normalmente, este documento é redigido simultaneamente a atividade de modelagem. Ou seja, é um documento dinâmico e resultado da seguinte soma: ciclo de reuniões realizados + documento de escopo + documento de requisitos + modelagem.

Finalizada esta tarefa, já é possível partir para a próxima tarefa: a modelagem de requisitos.

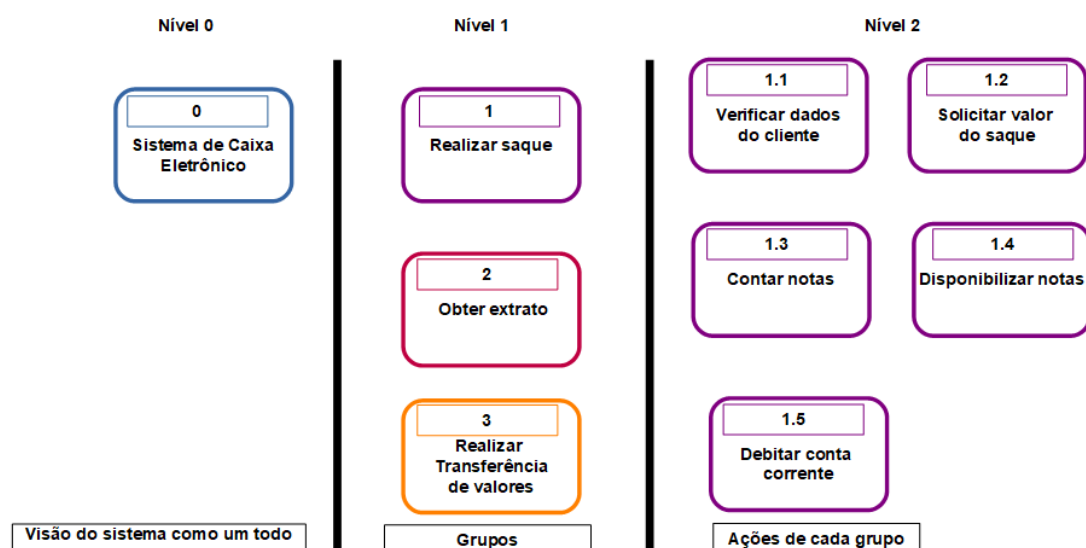
1.1 A modelagem de requisitos

Esta tarefa nem sempre é realizada de maneira trivial, pois depende de uma série de fatores a serem considerados, entre eles a complexidade do projeto. Mas, existe uma maneira bastante fácil de modelar, se você compreender esta maneira, nunca mais errará e tampouco perderá tempo pensando em como modelar. Ela consiste na execução dos seguintes passos:

Passo 1 – Modelar em alto nível: para realizar este modelo, pedimos emprestado do tipo de análise essencial, apenas 1 (um) símbolo, o símbolo que representa a entidade (normalmente uma elipse ou um retângulo com cantos arredondados). A partir disso, definimos os níveis de abstração que são separados por raias. Está confuso? Então, observe a Figura 1. Suponha que você, analista de requisitos esteja engajado num projeto onde será preciso construir 3 (três) opções

(grupos) para um sistema de caixa eletrônico: realizar saque, obter saldo e realizar transferência de valores. Você notou que tanto os grupos quanto as ações foram representadas de maneira gráfica? Todo aquele trabalho que o analista de requisitos realizou na atividade de análise de requisitos, foi representado de maneira visual. Este esquema pode ser facilmente apresentado ao cliente e, com isso, realizar as devidas validações e obter mais detalhes, antes de finalizar a modelagem e sua respectiva especificação e entregar para a equipe de programadores.

Figura 1 – Representação dos requisitos após classificação e agrupamento



Fonte: elaborada pela autora.

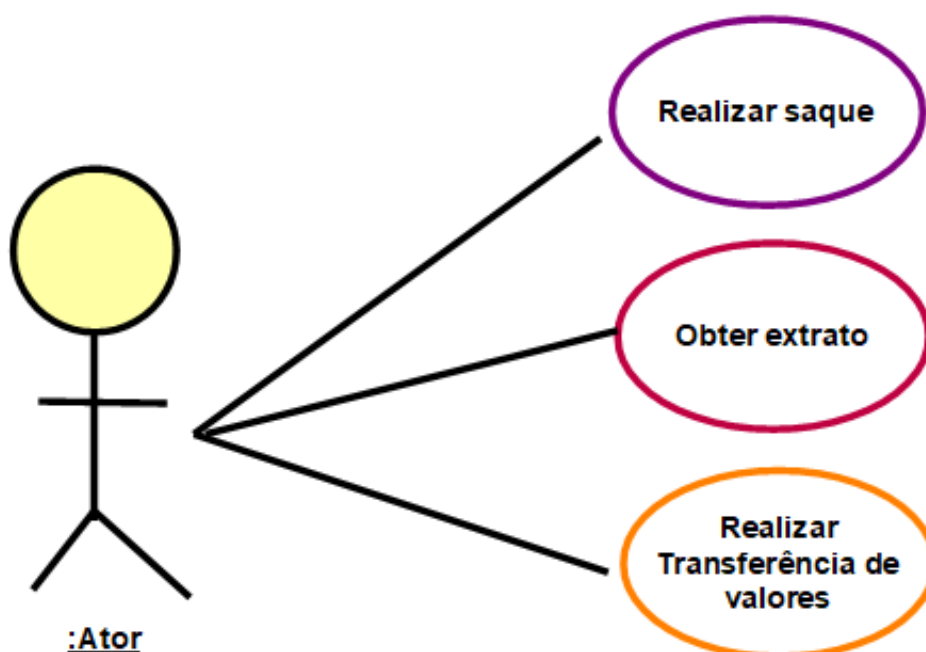
As demais ações de cada grupo ficarão como exercício para você (use a sua imaginação).

Passo 2–“Traduzir” para a linguagem UML (*Unified Modelling Language*): uma vez executado o Passo 1, é hora de realizar a modelagem propriamente dita. Há uma linguagem muito utilizada chamada UML. Esta linguagem possui uma série de diagramas que, dependendo da complexidade do projeto, podem ser utilizados 2 (dois) ou mais deles. O primeiro diagrama a ser utilizado é o diagrama de caso de uso; basicamente, ele é aquele que especifica um comportamento


de um sistema segundo uma perspectiva externa, descrevendo um conjunto de sequências de ações realizadas pelo sistema. Ele possui apenas 3 (três) elementos: um elemento denominado ator (um “bonequinho” que pode representar: uma pessoa, ou grupo de pessoas, ou um sistema ou ainda, um componente de sistema. É aquele que efetivamente utilizará o sistema), o outro denominado caso de uso (elipse) e o último elemento é um traço seccionado denominado associação. Já o segundo diagrama é denominado diagrama de classes (basicamente as regras são idênticas ao diagrama entidade-relacionamento da disciplina de banco de dados).

Nesse sentido, como “traduzir” o Passo 1 em diagrama de caso de uso? Isso muito fácil: tudo o que está na raia denominada Nível 1 será representado por elipses (caso de uso) e as ações representadas na raia 3 (Nível 2) serão especificadas (você fará uma redação descrevendo como cada ação é executada dentro de cada caso de uso), observe a Figura 2.

Figura 2 - “Tradução” do passo 1 para diagrama de casos de uso



Fonte: elaborada pela autora.



Uma observação: as cores de ambos os diagramas (Figuras 1 e 2) são apenas para facilitar a visualização. Não há cores nos diagramas.

E para “traduzir” de diagrama de casos de uso para diagrama de classes? Agora ficou mais fácil ainda: tudo o que está dentro da elipse, você chama de classe e os relacionamentos, você busca na matriz de requisitos! Pronto, está feito!

Estas “traduções” não se esgotam assim facilmente. Ou seja, há algumas análises e pormenores para se construir os diagramas de caso de uso e de classe, assim como os outros diagramas que compõem a UML. Sugere-se fortemente a leitura de Sommerville (2018), a fim de saber mais sobre a construção destes diagramas e dos demais que compõem a linguagem UML.

Passo 3 – Organizando o Documento de Especificação de Casos de Uso: este documento possui uma estrutura própria e dependerá da cultura da empresa onde o analista de requisitos trabalha; existem empresas que possuem templates próprios para este documento, outras não. Às vezes, pode ser que na empresa onde o analista de requisitos trabalhe haja apenas um documento contendo vários capítulos, cada qual tratando da especificação de um requisito que foi “traduzido” para caso de uso; já outras empresas, preferem trabalhar com documentos separados, ou seja, cada especificação fica em um documento, gerando vários documentos. No tocante à estrutura do documento de especificação de casos de uso, existe uma gama de templates no mercado e na literatura. O analista de requisitos elege aquele que achar mais adequado, caso a empresa onde ele trabalhe não possua um template definido. Neste material, apresentamos uma estrutura que será utilizada para realizar os exercícios propostos desta disciplina.

A estrutura de um Documento de Especificação de Caso de Uso

O objetivo é especificar o comportamento de um caso de uso pela descrição textual de seu fluxo de eventos, de modo que alguém de fora possa compreendê-lo. Antes de apresentar a estrutura, apresentaremos algumas considerações sobre a descrição textual:

1. **Cada caso de uso** constitui um **curso completo de eventos** com um ator e especifica a interação (“conversa”) que acontece entre o ator e o sistema.
2. O **conjunto** de todas as **descrições de casos de uso** especifica todas as maneiras de se usar o sistema e, conseqüentemente, a sua **funcionalidade completa**.
3. **Aqui é descrito o que um sistema faz**. Simula-se uma conversa entre o ator e o sistema (caso de uso).
4. **Cada caso de uso possui apenas um único fluxo de eventos principal (curso ou fluxo básico) e vários fluxos alternativos**. É um pecado mortal inserir fluxo de exceção, pois isso não existe! O que existe são fluxos alternativos que tratam de alguma situação especial ou de algum desvio, mas jamais fluxo de exceção.


Com base nestas considerações, abordaremos a estrutura da especificação dos casos de uso que será utilizada para este curso. A estrutura é a seguinte (um documento para cada caso de uso):

Identificação do sistema (nome do sistema que será construído).

Histórico de revisão (não se esqueça de que este documento passará por validações e, conseqüentemente atualizações – geralmente constrói-se uma tabela com as colunas: data, versão, descrição, autor).

Capítulo 1 – Nome do caso de Uso (por exemplo: manter fornecedor).

1.1 Breve descrição de seu objetivo (qual é o objetivo deste caso de uso que o Analista de Requisitos descreverá?).



Capítulo 2 – Precondições (o que é preciso para que este caso de uso seja executado? Pode ser outro caso de uso – mas é preciso mencionar qual; ou ainda outros dados relevantes).

Capítulo 3 – Fluxo de Eventos

3.1 Fluxo básico: é possível dar um nome para ele, mas deve deixar bem claro de que este é o fluxo principal. Em linguagem de programação, este fluxo pode ser entendido como o programa chamado, o *int main()*.

3.2 Fluxos alternativos: aqui é obrigatório o nome para ele, afinal, são vários fluxos alternativos. Em linguagem de programação, este fluxo pode ser entendido como o programa chamado de métodos/funções que o programa principal chama. Para facilitar a escrita, já que serão muitos os Fluxos alternativos, geralmente, seu nome é composto por uma sigla e o nome; por exemplo, FA01 – Selecionar opção Retornar ao Menu Principal.

Capítulo 4 – Regras de Negócio e Validação: neste capítulo são elencados e descritos todos os detalhes acerca deste caso de uso: os elementos que compõem a tela (regra de negócio), a obrigatoriedade de digitação de campos (este é um exemplo de validação), como os dados da tela serão armazenados, quais são as tabelas e os seus respectivos campos etc.

Capítulo 5 – Diagramas UML: este capítulo é dividido em subcapítulos. Cada subcapítulo é a representação de um Diagrama UML, que versa sobre a “tradução” daquele requisito que foi transformado em caso de uso e está sendo descrito no documento. Veja bem: não é o projeto todo e sim um recorte dos diagramas.

Capítulo 6 – Protótipo da tela: você se lembra daquele rascunho que o analista de requisitos apresentou e validou com o cliente? Então, agora é hora de inseri-lo no documento.



2. A validação dos requisitos

A atividade de validação dos requisitos consiste em certificar-se de que os requisitos estão consistentes com as necessidades dos usuários.

A validação é composta por reflexões sobre:

- **Completeza:** todos os requisitos funcionais devem estar muito bem definidos.
- **Compreensível:** todos os usuários devem ser capazes de entender os requisitos.
- **Consistência:** compreende-se que a redação dos requisitos não deve ser definida de maneira contraditória.
- **Rastreamento:** é estabelecer referências entre os requisitos (Mapa de Requisitos), aspectos de projeto e implementação, para possibilitar controlar os efeitos das modificações.
- **Teste:** é garantir que todos os requisitos funcionais possam ser testados.

O objetivo desta atividade é possibilitar ao usuário de ler e entender a especificação de requisitos, e indicar se os requisitos refletem as suas ideias. Para isso, são sugeridas as técnicas de revisão de requisitos funcionais e prototipação.

Nesta atividade, também é possível apresentar além da modelagem construída no Passo 1 da subseção 1.1, o diagrama de casos de uso (Passo 2 da citada subseção). O cliente é capaz de compreender os dois diagramas, muito embora ele não saiba os nomes destes diagramas, eles são bons para melhor visualizar e validar. O analista de requisitos não precisa mencionar os nomes dos diagramas, apenas apresentá-los. Reforçamos que o analista de requisitos deve evitar jargões técnicos,

pois isso frustra – de alguma maneira — a comunicação entre o cliente e o analista de requisitos.

3. A mudança de requisitos

Frequentemente, durante o transcorrer do ciclo de desenvolvimento de sistemas, há algumas alterações (mudanças) solicitadas pelo cliente. Dependendo da solicitação, ela pode ou não gerar impactos em todo o projeto e comprometer todo o planejamento e condução dos trabalhos da equipe de projetos.

Uma maneira de verificar a proporção do impacto diante de uma solicitação de mudança, é a execução de uma atividade denominada análise de impacto, que tem como objetivo identificar quais as alterações serão necessárias e que deverão ser realizadas no sistema, mas também no planejamento do projeto.

A análise de impacto, por sua vez, avalia o esforço e o custo das mudanças toda vez que houver a solicitação de mudança. Para realizar esta análise é preciso que o analista de requisitos em comunhão com o gerente de projetos redija um documento à parte, contendo: a identificação e registro da necessidade de mudança; análise de impacto; e a implementação da mudança. Outros elementos podem ser adicionados neste documento, como o parecer sobre a solicitação de mudança (válida ou não válida e sua respectiva justificativa) e elencar os requisitos que serão afetados e sua respectiva justificativa.

Como saber quais requisitos serão impactados? Lembre-se da matriz de rastreabilidade, o analista de requisitos poderá utilizá-la para auxiliá-lo nesta tarefa, pois verificando os relacionamentos entre os requisitos é possível identificar mais facilmente quais requisitos serão ou não afetados tanto direta quanto indiretamente. Além disso, é importante

identificar todos os documentos elaborados em todas as fases do ciclo. Assim, é possível garantir e ter maior dimensão dos impactos da solicitação de mudança no projeto de construção do sistema.

4. A gerência dos requisitos

Os objetivos desta etapa são administrar e controlar eventuais mudanças nos requisitos funcionais já especificados e eventuais solicitações de criação de novos requisitos. Isso se dá conforme o cliente vai amadurecendo seu entendimento sobre o que ele realmente espera de um sistema que atenda às suas necessidades.

Segundo Dorfman (1997), o gerenciamento de requisitos pode ser definido como o planejamento e o controle de todas essas atividades relacionadas. Assim, caracteriza-se pelos sucessivos rastreios de alterações para manter a concordância entre o cliente e a equipe do projeto.

Portanto, dependendo da empresa/departamento de TI esta atividade pode ser realizada ou pelo gerente de projetos ou pelo analista de requisitos ou, ainda, por ambos.

Nesta etapa, os requisitos são diferenciados entre requisitos permanentes e requisitos voláteis. Os requisitos permanentes são aqueles derivados da atividade principal do cliente, por exemplo, cadastro de clientes, cadastro de fornecedores etc. Já os requisitos voláteis são aqueles que se modificam a medida que o cliente possui melhor visualização de sua necessidade ou, ainda, que de alguma maneira gere impacto na empresa que ele trabalha, por exemplo, regras de planos de saúde, modificação em cálculos de aposentadoria etc. Os requisitos voláteis são identificados de acordo com a seguinte classificação:

- **Requisitos Consequentes:** são aqueles que podem modificar os processos e procedimentos na execução de tarefas diárias da empresa do cliente.
- **Requisitos de Compatibilidade:** basicamente são aqueles que dependem de atualizações de outros sistemas utilizados pelo cliente.
- **Requisitos Emergentes:** são aqueles que se originam a partir da clareza que o cliente possui de sua necessidade.
- **Requisitos Mutáveis:** são aqueles que sofrem modificações devido a fatores externos à empresa do cliente e que de alguma maneira, gere impacto na execução de suas atividades diárias. Eles podem ser regras ou leis governamentais.

Esta é uma etapa cíclica que ocorre durante todo o processo de desenvolvimento do sistema, por sua vez, a execução é realizada através dos seguintes passos:

Passo 1 – Identificação da mudança: a mudança é identificada quando o cliente entre em contato com o analista de requisitos que anota a mudança e executa o Passo 2.

Passo 2 – Documentação da mudança: nesta documentação são colocados os detalhes e o analista de requisitos em comunhão com o gerente de projetos, executa-se o Passo 3.

Passo 3 – Análise da mudança: são realizadas análises de viabilidade, custo e impacto e atualizada a documentação de mudança.

Passo 4 – Implementação da mudança: após a realização das análises em Passo 3, é inserido no documento a decisão de realização da mudança ou não com suas respectivas justificativas.

Perceba que a gestão dos requisitos possui forte relacionamento com a atividade de mudança (seção 3).

Nesse sentido, há algumas ferramentas gratuitas para realizar esta atividade, dentre elas é possível citar: OSRM (*Open Source Requirements Management Tool*/Ferramenta de código aberto para gerência de requisitos), Spider-CL, DotProject, SIGERAR e OpenReq.

Referências Bibliográficas

DORFMAN, M.; THAYER, R. **Software engineering**. Los Alamitos, CA: IEEE Computer Society Press, 1997.

PRESSMAN, R. S. **Engenharia de software**: uma abordagem profissional. Tradução de João Eduardo Nóbrega Tortello. 8. ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, I. **Engenharia de software**. Tradução de Luiz Cláudio Queiroz. 10. ed. São Paulo: Pearson Education do Brasil, 2018.



BONS ESTUDOS!