

# Projetos de Sistemas de Informação



**Educação a Distância**  
Cruzeiro do Sul Educacional  
*Campus Virtual*



# Material Teórico



## Análise de Requisitos

### **Responsável pelo Conteúdo:**

Prof. Ms. Fábio Peppe Beraldo

### **Revisão Textual:**

Prof. Ms. Claudio Brites



# UNIDADE

## Análise de Requisitos



- Introdução
- Engenharia de Requisitos
- Tópicos de análise de requisitos
- Especificação de requisitos de *software*
- Tipos de requisitos
- A análise de requisitos



O objetivo do estudo da disciplina Projeto de Sistemas de Informação é habilitá-lo(a) a projetar melhorias de qualidade em produtos e serviços; e, acima de tudo, identificar as vantagens da modelagem de sistemas; uso do modelo de conceituar e construir sistemas; estudo interdisciplinar da utilização desses modelos; modelagem de sistemas, análise e esforços de design; modelagem e simulação de sistemas, tais como a dinâmica do sistema; sistemas de linguagem de modelagem específica.

Primeiramente, é necessária a conscientização de que a exigência – tanto de estudo quanto de avaliação – da disciplina online é, no mínimo, a mesma que a presencial.

Assim, é necessário organizar seu tempo para ler atentamente o material teórico e assistir os vídeos, se houver, inclusive do material complementar.

Organize-se também de forma a não deixar para o último dia a realização das atividades de sistematização (pontuada) e fórum (não pontuado). Podem ocorrer imprevistos e, encerrada a Unidade, encerra-se a possibilidade de obter a nota relativa a cada atividade.

Para ampliar seu conhecimento, bem como aprofundar os assuntos discutidos, pesquise, leia e consulte os livros indicados nas Referências e/ou na Bibliografia.

Caso ocorram dúvidas, contate o professor tutor por meio do Fórum de Dúvidas, local ideal, pois assim a explicação poderá ser compartilhada por todos.

## Contextualização

A análise de requisitos em sistemas de engenharia e engenharia de *software* engloba as tarefas que determinam as necessidades e condições para atender a um produto novo ou alterado, tendo em conta as exigências possivelmente conflitantes das diversas partes interessadas, tais como os beneficiários ou usuários.

Nesta Unidade teremos uma visão de como a análise de requisitos funciona, assim como suas ferramentas.

## Introdução



A análise de requisitos em sistemas de engenharia e engenharia de *software* engloba as tarefas que determinam as necessidades e condições para atender a um produto novo ou alterado, tendo em conta as exigências possivelmente conflitantes das diversas partes interessadas, tais como os beneficiários ou usuários.



### Atenção

A análise de requisitos é fundamental para o sucesso de um projeto de desenvolvimento. Os requisitos devem ser objeto de recurso, mensuráveis, verificáveis, relacionados às necessidades de negócio identificados ou oportunidades, e definidos para um nível de detalhe suficiente para o projeto do sistema.

Os requisitos podem ser funcionais e não-funcionais, assim como aplicação de uma abordagem sistemática e disciplinada, quantificável para o desenvolvimento, operação e manutenção de *software*.

Conceitualmente, a análise de requisitos inclui três tipos de atividade, a saber:

- Levante de requisitos: a tarefa de se comunicar com clientes e usuários para determinar quais são as exigências. Esta atividade também é chamada de levantamento de requisitos;
- Análise de requisitos: que determina se os requisitos estabelecidos não são claros, incompletos, ambíguos ou contraditórios para, em seguida, resolver esses problemas;
- Requisitos de gravação: que podem ser documentados de inúmeras formas, tais como documentos de linguagem natural, casos de uso, histórias de usuário ou especificações de processo.

A análise de requisitos pode ser um processo longo e árduo, durante o qual inúmeras habilidades psicológicas delicadas estão envolvidas. Novos sistemas costumam mudar o ambiente e as relações interpessoais, por isso é importante identificar todas as partes interessadas, ter em conta todas as suas necessidades e garantir que compreendam as implicações desses novos sistemas.

Os analistas podem empregar inúmeras técnicas para obter as exigências do cliente. Historicamente, este tem incluído entrevistas, ou grupos focais (neste contexto mais apropriadamente chamados de *workshops* de requisitos) e criado listas de requisitos. Técnicas mais modernas incluem casos de prototipagem e de uso. Sempre que necessário, o analista empregará uma combinação desses métodos para estabelecer os requisitos exatos das partes interessadas, de modo a produza um sistema que atenda às necessidades do negócio.



## Engenharia de requisitos



A análise de requisitos sistemática também é conhecida como engenharia de requisitos. É por vezes referida por nomes como levantamento de requisitos, requisitos de captura ou especificação de requisitos. A análise de requisitos também pode ser aplicada especificamente para a análise propriamente dita, em oposição a elicitación ou documentação dos requisitos, por exemplo.

Para alguns autores engenharia de requisitos é considerada uma subdisciplina da engenharia de sistemas e *software* de engenharia que se preocupa com a determinação dos objetivos, funções e limitações de sistemas de *hardware* e *software*.

Em alguns modelos de ciclo de vida, o processo de engenharia de requisitos começa com uma atividade de estudo de viabilidade, o que leva a um relatório de viabilidade. Se o estudo de viabilidade sugerir que o produto deve ser desenvolvido, a análise de requisitos é iniciada.

## Tópicos da análise de requisitos



### Identificação dos Stakeholders

A análise dos *stakeholders* é uma discussão sobre seus usos comerciais. *Stakeholders* são pessoas ou organizações (pessoas coletivas, tais como empresas, organismos de normalização etc.), que têm um interesse válido no sistema. Podem ser afetados direta ou indiretamente. Na década de 1990 houve uma maior ênfase na identificação das partes interessadas. Desde então, ficou cada vez mais notório que as partes interessadas não estão limitadas à organização ao empregarem o analista. Outras intervenções incluem:

- O operador do sistema (operadores normais e manutenção);
- Qualquer indivíduo que se beneficie do sistema (beneficiários, políticos, financeiros e sociais funcionais);
- Qualquer pessoa envolvida na compra ou aquisição do sistema. Em uma organização de produtos de mercado de massa, gerenciamento de produto, *marketing* e, às vezes, ato de vendas, como substitutos consumidores (clientes de mercado de massa) para orientar o desenvolvimento do produto;
- Organismos que regulam os aspectos do sistema (financeiro, de segurança e outras autoridades reguladoras);
- Pessoas ou organizações que se opõem ao sistema (partes interessadas negativas; assim como casos de mau uso);
- Organizações responsáveis pelos sistemas que fazem interface com o sistema sob o projeto;
- Organizações que se integram horizontalmente com a organização para quem o analista está projetando o sistema.



## Entrevistas com os *stakeholders*

As entrevistas com os *stakeholders* são uma técnica comum para a análise de requisitos. Essas entrevistas podem revelar exigências não previstas anteriormente no escopo do projeto, assim como podem evidenciar contradições nos requisitos. No entanto, cada interessado terá uma ideia de sua expectativa ou terá visualizado as suas necessidades.

## Exigência de listas de contrato

Uma forma tradicional de documentação dos requisitos é a exigência de listas de contrato. Em um sistema complexo, listas de requisitos podem ter centenas de páginas. Uma metáfora apropriada seria uma lista de compras extremamente longa. Tais listas de análise moderna provam-se malsucedidas em atingir os seus objetivos; embora ainda sejam utilizadas. Quanto as suas vantagens:

- Fornece uma lista de verificação de requisitos;
- Fornece um contrato entre o patrocinador do projeto e seus desenvolvedores;
- Para um grande sistema pode fornecer uma descrição de alto nível.

No que diz respeito às desvantagens:

- Torna impraticável a leitura de documentos como um todo para compreender de forma coerente o sistema;
- Tais exigências listam de forma abstrata todos os requisitos, tirando-lhes todo o contexto, de modo que essa abstração:
- Torna impossível ver como os requisitos se encaixam;
- Dificulta a identificação dos requisitos mais importantes;
- Significa que quanto mais pessoas leem essas exigências, mais interpretações do sistema são formuladas;
- Torna incerta a obtenção da maioria dos requisitos. Necessariamente, esses documentos dizem respeito a generalidades; mas “o diabo”, como dizem, está nos detalhes.
- Criam uma falsa sensação de compreensão mútua entre as partes interessadas e desenvolvedores;
- Essas listas de estilo em contrato dão às partes interessadas uma falsa sensação de segurança sobre o que os programadores devem fazer. No entanto, devido à natureza desses documentos, inevitavelmente omitem requisitos cruciais, identificados tardiamente, ao longo do processo. Em casos assim, os desenvolvedores usam esses requisitos inicialmente desconhecidos às listas para renegociar os termos e condições a seu favor;
- Finalmente, essas listas de requisitos não ajudam compreensão de concepção do sistema, uma vez que não se prestam a tal aplicação.

## Metas mensuráveis

As melhores práticas para tornar a lista composta de requisitos como roteiros de pistas é perguntar repetidamente “por que?”, até que os reais propósitos de negócios sejam descobertos.

Nesse sentido, *stakeholders* e desenvolvedores podem, em seguida, elaborar testes para medir o nível de atingimento de cada objetivo. Tais objetivos tendem a mudar em uma velocidade menor do que a longa lista de requisitos específicos, com o benefício de que, neste método, passam a ser regularmente medidos.

Uma vez que em um pequeno conjunto de objetivos classificados como críticos as medidas foram devidamente estabelecidas, as fases de desenvolvimento iterativo prototipagem rápida e curta podem continuar a entregar valor real das partes interessadas, inclusive antes que o projeto atinja a metade de seu cronograma.

## Especificação de requisitos de *software*



A especificação de requisitos de *software* (SRS) é uma descrição completa do comportamento do sistema a ser desenvolvido. Inclui um conjunto de casos de uso que descrevem todas as interações que os usuários terão com o *software*. Os casos de uso também são conhecidos como requisitos funcionais. Além de casos de uso, o SRS também contém os requisitos não-funcionais (ou complementares). Os requisitos não-funcionais impõem restrições sobre a concepção ou implementação (tais como requisitos de desempenho, padrões de qualidade, ou restrições de projeto).



### Informação

Abordagens recomendadas para a especificação de requisitos de *software* são descritos por IEEE 830-1998. Esta norma apresenta possíveis estruturas, conteúdos desejáveis e qualidades de uma especificação de requisitos de *software*.

## Tipos de requisitos



Requisitos são classificados de diversas maneiras. A seguir, são categorizados os mais comuns que se relacionam com a gestão técnica:

## Requisitos do cliente

Declarações de fatos e pressupostos que definem as expectativas do sistema em termos de objetivos da missão, ambiente, restrições e medidas de eficácia e adequação (MOE, MOS). Os clientes são aqueles que executam as oito principais funções de engenharia de sistemas, com especial destaque para o operador como cliente-chave. Os requisitos operacionais definirão a necessidade básica e, no mínimo, responderão às questões colocadas na seguinte lista:

- Distribuição ou implantação operacional: onde o sistema será utilizado?
- Missão, perfil ou cenário: como o sistema atingirá seu objetivo de missão?
- Parâmetros de desempenho e afins: quais são os parâmetros críticos do sistema para realizar a missão?
- Ambientes de utilização: como são os componentes do sistema a serem utilizados?
- Requisitos de eficácia: quão eficaz ou eficiente deve ser o sistema para realizar sua missão?
- Ciclo de vida operacional: quanto tempo o sistema estará em uso?
- Ambiente: qual o ambiente que se espera que o sistema funcione de forma eficaz?

## Requisitos funcionais

Os requisitos funcionais explicam o que deve de ser feito através da identificação da tarefa necessária, ação ou atividade que a ser realizada. A análise de requisitos funcionais será utilizada como as funções top-level para a análise funcional.

## Requisitos não-funcionais

Os requisitos não-funcionais especificam os critérios que podem ser empregados para avaliar o funcionamento de um sistema, ao invés de comportamentos específicos.

## Requisitos de desempenho

Na medida em que uma missão ou função deve ser executada, geralmente é mensurada em termos de quantidade, qualidade, cobertura, atualidade, ou prontidão. Durante a análise de requisitos, o requisito de desempenho será desenvolvido de forma interativa em todas as funções identificadas com base em fatores de ciclo de vida do sistema; caracteriza-se em termos de grau de certeza em sua estimativa, o nível de criticidade para o sucesso do sistema e sua relação com outros requisitos.

## Requisitos de projeto

Utiliza os comandos *build to*, *code to* e *buy to* como requerimentos para produtos e *how to execute* como requerimento para processos, operando em pacotes de dados técnicos e manuais técnicos.

## Requisitos derivados

Tratam-se de requisitos que estão implícitos ou transformados a partir de exigência de nível superior. Por exemplo, a exigência de longo alcance ou de alta velocidade pode resultar em uma exigência de projeto para o baixo peso.

## Requisitos alocados

É um requisito estabelecido pela divisão, ou por indução de níveis superiores, nos requerimentos de níveis inferiores. Por exemplo, um produto que possua 100kg, constituído por dois subníveis, pode ser resultado de um produto de 70kg e outro de 30kg de dois outros níveis, quase como se estivéssemos pensando em herança de requisitos de produtos.

### A análise de requisitos



## Questões relacionadas aos *stakeholders*

Existe uma série de formas de se inibir a coleta de requisitos e prejudicar o desenvolvimento:

- Os usuários não entendem o que querem ou não têm uma ideia clara das suas necessidades;
- Os usuários não se comprometem com um conjunto de requisitos registrados;
- Os usuários insistem em novas exigências após a correção de custos e cronograma;
- A comunicação com os usuários é lenta;
- Comumente os usuários não participam de comentários ou são incapazes de fazê-lo;
- Os usuários são tecnicamente sofisticados;
- Os usuários não entendem o processo de desenvolvimento;
- Os usuários não dominam a tecnologia atual.

Tais pontos podem levar a uma situação em que as necessidades dos utilizadores sempre mudarão, mesmo quando o sistema ou produto de desenvolvimento for iniciado.

## Questões do engenheiro/desenvolvedor

Possíveis problemas causados por engenheiros e desenvolvedores durante a análise de requisitos:

- Pessoal técnico e usuários finais podem ter diferentes vocabulários. Consequentemente, podem erroneamente acreditar que estão em perfeito acordo até o fornecimento do produto concluído;

- Engenheiros e promotores podem tentar encaixar os requisitos em um sistema ou modelo existente, em vez de desenvolver um sistema específico para as necessidades do cliente;
- Certas análises podem ser realizadas indevidamente por engenheiros ou programadores, ao invés de pessoal com habilidades em relacionamento interpessoal e domínio em compreender as necessidades do cliente.

## Possíveis soluções

Uma tentativa de solução para problemas de comunicação é o emprego de especialistas em negócios ou análise de sistemas. Além disso, uma nova classe de ferramentas de simulação ou de definição de aplicativos entraram no mercado. Tais recursos são projetados para fazer a ponte de comunicação entre os usuários de negócios e da organização de Tecnologia da Informação (TI), além de permitir que aplicativos sejam testados no mercado antes de qualquer código ser produzido. A melhoria dessas ferramentas eletrônicas deve esboçar os fluxos de aplicação e melhorar a capacidade de:

- Captar as necessidades de lógica de negócios e dados;
- Geração de protótipos de alta fidelidade, que imitam a aplicação final;
- Interatividade;
- Adicionar requisitos contextuais e outros comentários;
- Usuários remotos e distribuídos para executar e interagir.

## Material Complementar

PRESSMAN, R. S. **ENGENHARIA De Software**. 6ª ed. Porto Alegre: Grupo A, 2010. (e-book)

STEPHEN R. S. **Engenharia de Software**. 8ª ed. Porto Alegre: Grupo A, 2008. (e-book)

PADUA, W. **Engenharia de Software**, 3ª ed. Rio de Janeiro: Grupo GEN, 2008. (e-book)

KALINOVSKY, A. **Java Secreto: técnicas de descompilação, patching e engenharia reversa**. São Paulo: Pearson, 2009. (e-book)

PFLEEGER, S. L. **Engenharia de Software: teoria e prática** - 2ª ed. São Paulo: Pearson, 2009. (e-book)

## Referências

- KALINOVSKY, A. **Java secreto: técnicas de descompilação, patching e engenharia reversa**. São Paulo: Pearson, 2009.
- PADUA, W. **Engenharia de software**. 3. ed. Rio de Janeiro: Grupo GEN, 2008.
- PFLEEGER, S. L. **Engenharia de software: teoria e prática**. 2. ed. São Paulo: Pearson, 2009.
- PRESSMAN, R. S. **Engenharia de software**. 6. ed. Porto Alegre, RS: Grupo A, 2010.
- SCHACH, S. R. **Engenharia de software: os paradigmas clássicos e orientado a objetos**. 7. ed. São Paulo: Bookman, 2009.
- SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo: Addison-Wesley, 2007.
- STEPHEN R. S. **Engenharia de software**. 8. ed. Porto Alegre, RS: Grupo A, 2008.
- WAZLAWICK, R. S. **Análise e projeto de sistemas de informação orientados a objetos**. 2. ed. Rio de Janeiro: Elsevier, 2011.



## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



**Educação a Distância**

Cruzeiro do Sul Educacional

*Campus Virtual*

[www.cruzeirodosulvirtual.com.br](http://www.cruzeirodosulvirtual.com.br)

Campus Liberdade

Rua Galvão Bueno, 868

CEP 01506-000

São Paulo SP Brasil

Tel: (55 11) 3385-3000

