

Apresentação

Você sabia que o levantamento de requisitos pode ajudar a evitar a frustração de clientes ao final de um projeto de software? Muitas vezes, o cliente não sabe muito bem do que precisa, tornando mais difícil criar um projeto que satisfaça às suas necessidades. A identificação de requisitos no início do projeto é muito importante para o bom desenvolvimento do sistema. Um requisito pode ser uma condição, capacidade, função, objetivo, propriedade ou restrição que caracterize um sistema e satisfaça uma regra de negócio ou contrato.

Nesta Unidade de Aprendizagem, você irá conhecer os conceitos básicos de requisitos e as diferenças entre requisitos funcionais e não funcionais. Também irá estudar o que são regras de negócio e quais os principais problemas enfrentados pelos engenheiros de software ao coletar os requisitos.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Listar as técnicas de coleta de dados.
- Diferenciar requisitos funcionais e não funcionais.
- Identificar as regras de negócio.

Desafio

A companhia aérea YXZ está com aumento de vendas de bilhetes aéreos e precisa que você projete e implemente um sistema de controle de reservas e vendas de bilhetes.



Existe um grande número de voos com conexões entre diferentes aeroportos no país. Além disso, **cada bilhete ou reserva pode conter um ou vários voos e cada voo pode ter uma quantidade limitada de assentos** (isso vai depender do tipo de aeronave utilizada).

Você deve listar e descrever, de forma clara, quais são os requisitos funcionais e não funcionais que este sistema pode ter (informe no mínimo 2 de cada tipo).

Os requisitos podem variar, dependendo da interpretação.

Os requisitos funcionais devem ser relacionados com o comportamento do sistema. Já os não funcionais, devem ser voltados para aspectos mensuráveis, técnicas e necessidades voltadas para a implementação (segurança, acessos, ferramentas, etc.).

Para este caso, os requisitos podem ser estes:

Requisitos funcionais:

RF 1. Os clientes poderão realizar compras e reservas de bilhetes pelo sistema.

RF 2. Os clientes poderão realizar o cancelamento de reservas de bilhetes.

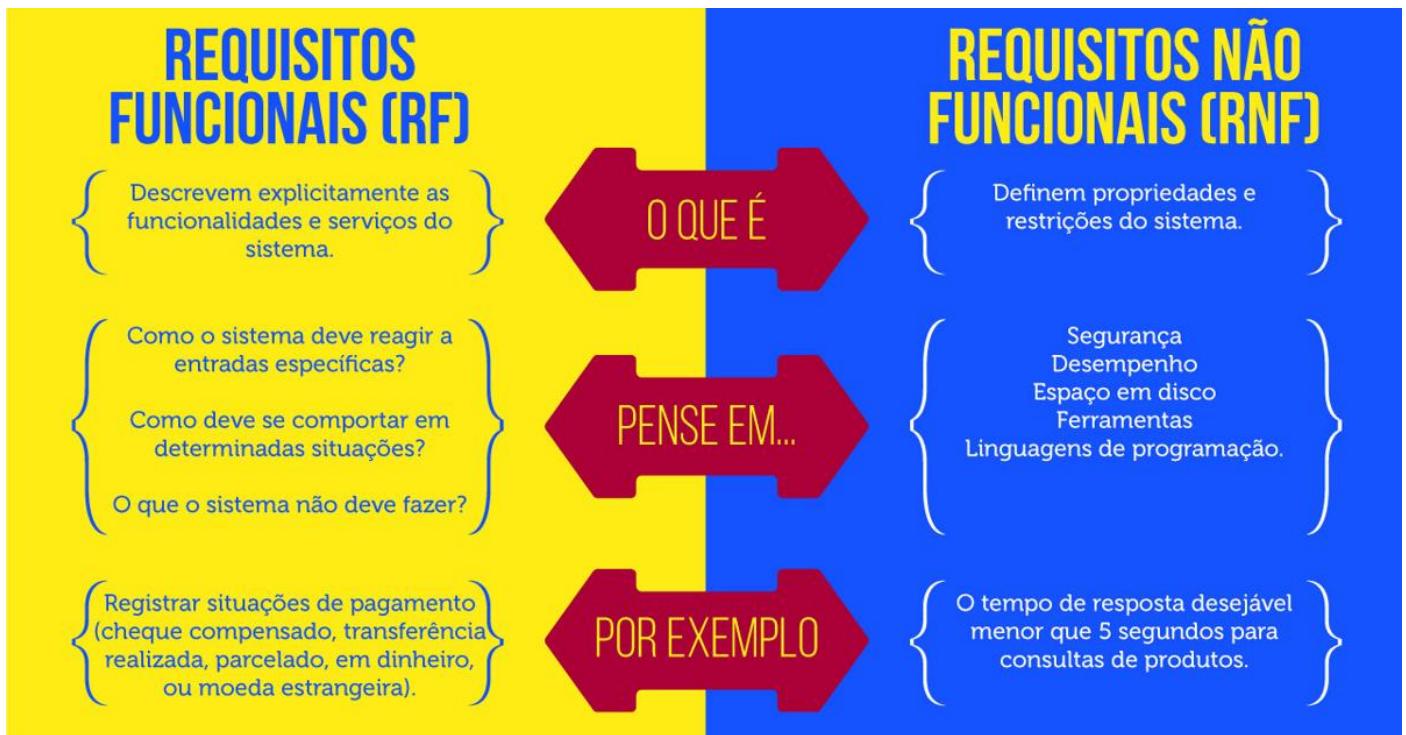
Requisitos não funcionais:

RN 1. O sistema deve ser desenvolvido para Web.

RN 2. O sistema deve ter uma resposta rápida, para que os clientes não desistam da compra.

Infográfico

Veja, no infográfico a seguir, o conceito de requisitos de *software* e entenda a diferença entre os requisitos funcionais e não funcionais.



Conteúdo do Livro

A coleta e identificação de requisitos de *software* pode ser considerada uma das tarefas mais importantes no processo de criação de um sistema. Um levantamento de requisitos mal feito gera um risco muito alto para qualquer projeto, podendo levá-lo ao fracasso. Por isso, é muito importante entender como é feito o levantamento de requisitos, seus tipos e características.

Acompanhe a leitura do capítulo Conhecer requisitos, da obra *Engenharia de Software* e conheça os conceitos básicos de requisitos e as diferenças entre requisitos funcionais e não funcionais, além de saber o que são regras de negócio e os problemas enfrentados pelos Engenheiros de *software* ao coletar os requisitos.

Boa leitura!

ENGENHARIA DE *SOFTWARE*



Aline Zanin

Izabelly Soares
de Moraes

Revisão técnica:

Jeferson Faleiro Leon

Desenvolvimento de Sistemas

Especialista em Formação Pedagógica de Professores



M827e Morais, Izabelly Soares de.

Engenharia de software [recurso eletrônico] / Izabelly Soares de Morais, Aline Zanin ; revisão técnica : Jeferson Faleiro Leon. – Porto Alegre : SAGAH, 2017.

ISBN 978-85-9502-253-9

Engenharia. 2. Engenharia de software auxiliada por computador. I. Zanin, Aline. II. Título.

CDU 004.41

Conhecer requisitos

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Listar as técnicas de coleta de dados.
- Diferenciar requisitos funcionais e não funcionais.
- Identificar as regras de negócio.

Introdução

Você sabia que o levantamento de requisitos pode ajudar a evitar a frustração de clientes ao final de um projeto de software? Muitas vezes, o cliente não sabe muito bem do que precisa, tornando mais difícil criar um projeto que satisfaça às suas necessidades. A identificação de requisitos no início do projeto é muito importante para o bom desenvolvimento do sistema. Um requisito pode ser uma condição, uma capacidade, uma função, um objetivo, uma propriedade ou uma restrição que caracterize um sistema e satisfaça uma regra de negócio ou contrato.

Neste capítulo, você vai conhecer os conceitos básicos de requisitos e as diferenças entre requisitos funcionais e não funcionais. Também vai estudar o que são regras de negócio e quais os principais problemas enfrentados pelos Engenheiros de software ao coletar os requisitos.

Técnicas para coleta de dados

A complexidade de um software está associada a todas as suas particularidades, que vão desde as suas funcionalidades até os recursos que este necessita para desempenhar sua função com eficácia. O desenvolvimento de um sistema exige que algumas etapas sejam seguidas e executadas com lógica. Dessa forma, um planejamento de um software tem como uma de suas principais atividades a obtenção dos requisitos, ou seja, todas as funcionalidades do software. Porém, apesar de aparentemente ser fácil definir o que um sistema irá fazer, não é,

pois dependemos diretamente do cliente, ou seja, daquele que está contratando nosso serviço para desenvolver um software que irá suprir suas necessidades.

Este primeiro momento é crucial para todas as demais etapas, pois imagine se a equipe desenvolve todo o software e, no final, ele não atende às necessidades do cliente? É por isso que essa etapa tem também cunho investigativo, no qual a equipe, além de ter que desenvolver o software, deve saber extrair do cliente tudo que ele precisa, ou seja, os requisitos do software. Um fato importante, e que merece ser citado, é o de que nem sempre o cliente comprehende o que é desenvolver um software, dessa forma, provavelmente ele não terá noção de como deve expor suas ideias, de forma organizada e coerente, e muito menos saberá relacionar o trabalho que será empenhado em seu projeto com os custos e os prazos finais. Para ele, é tudo muito simples, como inserir uma janela, mudar uma funcionalidade de um botão, ou até mesmo acrescentar uma lista de produtos, por exemplo. Mal sabe ele que, na verdade, tudo requer investimento de tempo e trabalho de uma equipe com diversas *expertises* diferentes.

Algumas técnicas acabaram sendo desenvolvidas para auxiliar a equipe de desenvolvimento neste processo. Conforme Pressman e Maxim (2016, p. 143), muitas abordagens para a coleta colaborativa de requisitos foram propostas. Cada uma faz uso de um cenário ligeiramente diferente, porém todas aplicam alguma variação das seguintes diretrizes básicas:

- As reuniões (reais ou virtuais) são conduzidas com a participação tanto dos Engenheiros de software quanto de outros envolvidos.
- São estabelecidas regras para a preparação e a participação.
- É sugerida uma agenda suficientemente formal para cobrir todos os pontos importantes; porém, suficientemente informal para estimular o fluxo livre de ideias.
- Um “facilitador” (pode ser um cliente, um desenvolvedor ou uma pessoa de fora) dirige a reunião.
- É utilizado um “mecanismo de definições” (planilhas, *flip charts*, adesivos de parede ou um boletim eletrônico, salas de bate-papo ou fóruns virtuais).

O momento em que os requisitos estão sendo coletados é nomeado como descoberta de requisitos, que, segundo Sommerville (2011, p. 72-76), é o processo de reunir informações sobre o sistema requerido e os sistemas existentes e separar dessas informações os requisitos de usuário e de sistema. Para isso, o autor lista algumas formas de executar tal ação:

1. Entrevistas. Podem ser de dois tipos:
 - Entrevistas fechadas, em que um conjunto predefinido de perguntas é respondido pelo cliente.
 - Entrevistas abertas, em que não existe uma agenda predefinida, e a equipe desenvolve uma série de questões com o intuito de compreender melhor as necessidades do cliente.
2. Cenários. Podem ser úteis para a obtenção de mais detalhes na visão geral dos requisitos, em que cada cenário cobre determinados números de interações. Um cenário começa por meio de um esboço da interação. Durante o processo de elicitação, são adicionados detalhes ao esboço, para criar uma descrição completa dessa interação. Um cenário pode incluir:
 - Uma descrição do que o sistema e os usuários esperam quando o cenário se iniciar.
 - Uma descrição do fluxo normal de eventos no cenário.
 - Uma descrição do que pode dar errado e como isso é tratado.
 - Informações sobre outras atividades que podem acontecer ao mesmo tempo.
 - Uma descrição do estado do sistema quando o cenário acaba.
3. Casos de uso. Trazem definições estabelecidas pela linguagem de modelagem unificada (UML, do inglês *Unified Modeling Language*). São documentados por um diagrama de casos de uso de alto nível. Identificam as interações individuais entre o sistema e seus usuários ou outros sistemas.
4. Etnografia. É uma técnica de observação que pode ser usada para compreender os processos operacionais e ajudar a extrair os requisitos de apoio para esses processos, em que um analista faz uma imersão no ambiente de trabalho em que o sistema será usado. O trabalho diário passa por diversas observações, nas quais contêm anotações sobre as tarefas em que os participantes estão envolvidos são realizadas.

As técnicas são definidas conforme o contexto em que o projeto será inserido, até mesmo porque não podemos definir qual método de coleta de dados é o mais adequado, tendo em vista que as possibilidades são infinitas.



Fique atento

Ao utilizarmos o termo *stakeholder*, estamos nos referindo a qualquer pessoa que possa estar diretamente ou indiretamente envolvida no projeto, podendo ser desde o cliente até o gerente de projeto. É importante compreendermos que cada um trará um ponto de vista diferente ao projeto, tendo em vista que cada um trará, também, toda a sua bagagem de conhecimentos e experiências. Pode, muitas vezes, ser uma boa contribuição para a excelência do produto final, que será o software.

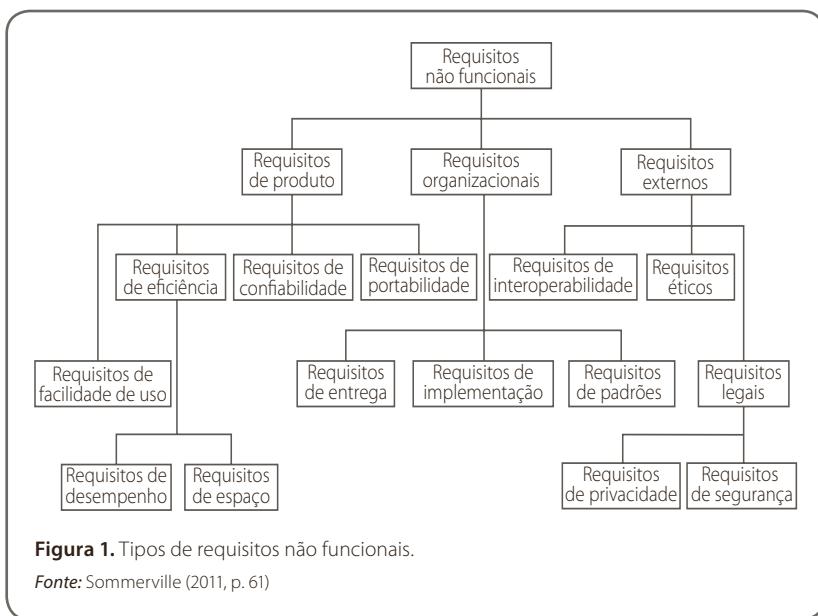
Requisitos funcionais e não funcionais

Um software necessita de recursos computacionais para que possa desempenhar suas funções, ou seja, ele precisa de dispositivos físicos para operar. Para isso, precisamos definir quais restrições e qual poderá ser o limite de alcance que o software que está sendo desenvolvido poderá vir a ter. Um requisito não lista apenas uma funcionalidade, ele traz especificações também, de como deve agir para que o objetivo de sua função seja atingido.

Dessa forma, separamos as necessidades do cliente, ou seja, de seu software, em duas modalidades, que chamamos de requisitos funcionais e não funcionais:

1. Requisitos funcionais (RF ou, em inglês, FR, de *functional requirement*): “São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, também podem explicitar o que o sistema não deve fazer” (SOMMERVILLE, 2011, p. 59).
2. Outros aspectos que norteiam os requisitos funcionais estão ligados à completude, ou seja, se todos as funcionalidades estão definidas, ao realismo quanto à definição das funcionalidades e à consistência, na qual os requisitos não devem trazer ambiguidade ao projeto.
3. Requisitos não funcionais (RNF ou, em inglês, NFR, de *non functional requirement*): “Pode ser descrito como um atributo de qualidade, de desempenho, de segurança ou como uma restrição geral em um sistema” (PRESSMAN; MAXIM, 2016, p. 141). Esses requisitos abrangem muito mais atributos, que podem estar associados a outras áreas do software.

4. Para Sommerville (2011, p. 60), esses requisitos podem “estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta, e ocupação de área. [...] Os requisitos não funcionais, como desempenho, proteção ou disponibilidade, normalmente especificam ou restringem as características do sistema como um todo. [...] Esses requisitos podem afetar a arquitetura geral de um sistema em vez de apenas componentes individuais”. A Figura 1 a seguir traz um diagrama com alguns dos requisitos não funcionais.



O diagrama desenvolvido por Sommerville (2011, p. 61) expõe requisitos não funcionais que podem ser provenientes das características requeridas para o software (requisitos do produto), da organização que desenvolve o software (requisitos organizacionais) ou de fontes externas:

- Requisitos do produto: “Requisitos que especificam ou restringem o comportamento do software. Exemplos incluem os requisitos de desempenho quanto à rapidez com que o sistema deve executar e quanta memória ele requer, os requisitos de confiabilidade que estabelecem

a taxa aceitável de falhas, os requisitos de proteção e os requisitos de usabilidade”.

- Requisitos organizacionais: “São os requisitos gerais de sistemas derivados das políticas e procedimentos da organização do cliente e do desenvolvedor. Exemplos incluem requisitos do processo operacional, que definem como o sistema será utilizado, dentre outros”.
- Requisitos externos: “Abrangem todos os requisitos que derivam de fatores externos ao sistema e seu processo de desenvolvimento. Exemplos incluem requisitos reguladores, que definem o que deve ser feito para que o sistema seja aprovado para uso, dentre outros”.

O detalhamento e as demais especificações acerca de cada vertente de um requisito, seja ele funcional ou não funcional, são descritos em uma documentação, que deve ser bastante detalhada e concisa. É nessa documentação que todas as práticas de levantamento de requisitos que foram utilizadas irão mostrar presteza para todo o processo. Outro fator importante é o de conhecermos também as regras de negócio que devem ser inseridas no sistema.



Link

Este link traz um vídeo desenvolvido por Fabrício Laguna (2016), que apresenta um breve resumo sobre os requisitos, assunto em destaque neste capítulo.

<https://goo.gl/mh8orc>



Identificação das regras de negócio

A identificação das regras de negócio que fazem parte do contexto do projeto compartilham das mesmas técnicas da coleta de dados, até mesmo porque é o momento em que os dados estão sendo coletados e as restrições do sistema estão sendo impostas, automaticamente, em que é necessário se ter conhecimento, também, das regras de negócio do cliente.

O entendimento dessas regras deve ser exposto em um modelo de negócio, pois “é ele que gera um entendimento do negócio do cliente como um todo. Com esse conhecimento, os desenvolvedores poderão aconselhá-lo sobre quais partes de seu negócio eles deveriam informatizar. De modo alternativo, se a tarefa for estender um produto de software existente, os desenvolvedores terão de entender o negócio existente como um todo para determinar como incorporar a extensão e aprender quais partes, se realmente alguma, do produto existente precisam ser modificadas para acrescentar o novo trecho” (SCHACH, 2010, p. 288).

Ainda sob o ponto de vista de Schach (2010, p. 288), “para construir um modelo de negócio, o desenvolvedor precisa ter um entendimento detalhado dos diversos processos de negócio. Tais processos serão refinados, isto é, analisados de forma mais detalhada. Podem ser usadas várias técnicas diferentes para obter as informações necessárias para criar o modelo de negócio, principalmente entrevistas”.

Além das possibilidades citadas no início deste capítulo, podemos acrescentar o uso de formulários e questionários no processo de levantamento de requisitos e, consequentemente, das regras de negócio. De acordo com Morgan (2001 apud ALVARENGA, 2007), a definição de uma regra de negócio requer a especificação de alguns detalhes operacionais, dentre os quais podemos destacar:

- **Quem** invoca uma regra: esta informação geralmente é descrita em um caso de uso ou em uma descrição de processo do negócio.
- **Quando** a regra é executada: normalmente isto é descrito por um evento, caso de uso ou descrição de processo do negócio.
- **Onde** a regra é executada: esta decisão é pertinente ao *design* do sistema e, mais especificamente, ao empacotamento do software.
- **Como** a regra é implementada: também é uma decisão da fase de projeto.

Conforme Alvarenga (2007, p.21), uma característica comum às regras de negócio é que elas fazem uso dos chamados “parâmetros do negócio”. Tais parâmetros podem ser definidos e gerenciados da mesma forma que as regras de negócio (MORGAN, 2001, apud ALVARENGA, 2007). O Quadro 1 traz características das Regras de Negócios encontradas na prática em muitas organizações, de acordo com Morgan (2001).

Quadro 1. Características das Regras de Negócio.

Característica: Uma RN deve ser ...	Justificativa
Atômica	Caso uma regra seja definida em termos de subunidades, estas subunidades não serão a mesma RN original. Isto quer dizer que a tentativa de dividir um RN em outras regras mais simples resultará em perda de informação e também em perda semântica.
Declaração do negócio	RN deveriam descrever a lógica do negócio e não a tecnologia que irá implementar a regra.
Declarativa	RN devem contribuir para um objetivo do negócio, ao invés de descrever como o objetivo do negócio será alcançado.
Restritiva	RN limitam as ações que podem ser aplicadas no contexto do negócio.
Representada em linguagem natural	RN devem ser expressas através de idioma natural, pois isto dispensa qualquer tipo de treinamento específico ou uso de ferramentas. Em contrapartida, podem ocorrer expressões ambíguas. Em consequência disso, torna-se necessário mapear as regras escritas em linguagem natural para uma expressão matemática for mal, como uma linguagem de programação, por exemplo.
Rastreável	É preciso saber com as RN se encaixa dentro de um SI, e rastreá-las desde a origem até as suas realizações para manter o acompanhamento de todo o ciclo de vida do SI.
Estruturada	As regras devem ser escritas de tal maneira que facilitem a leitura e o entendimento. Porém, é necessário restringir o número de opções para se escrever a regra, isto é, determinar padrões estruturais para representação de regras. Esta prática pode apoiar o processo de automatização da regra, ou seja, o mapeamento da regra para implementação desejada.

Fonte: Alvarenga (2007, p. 21)

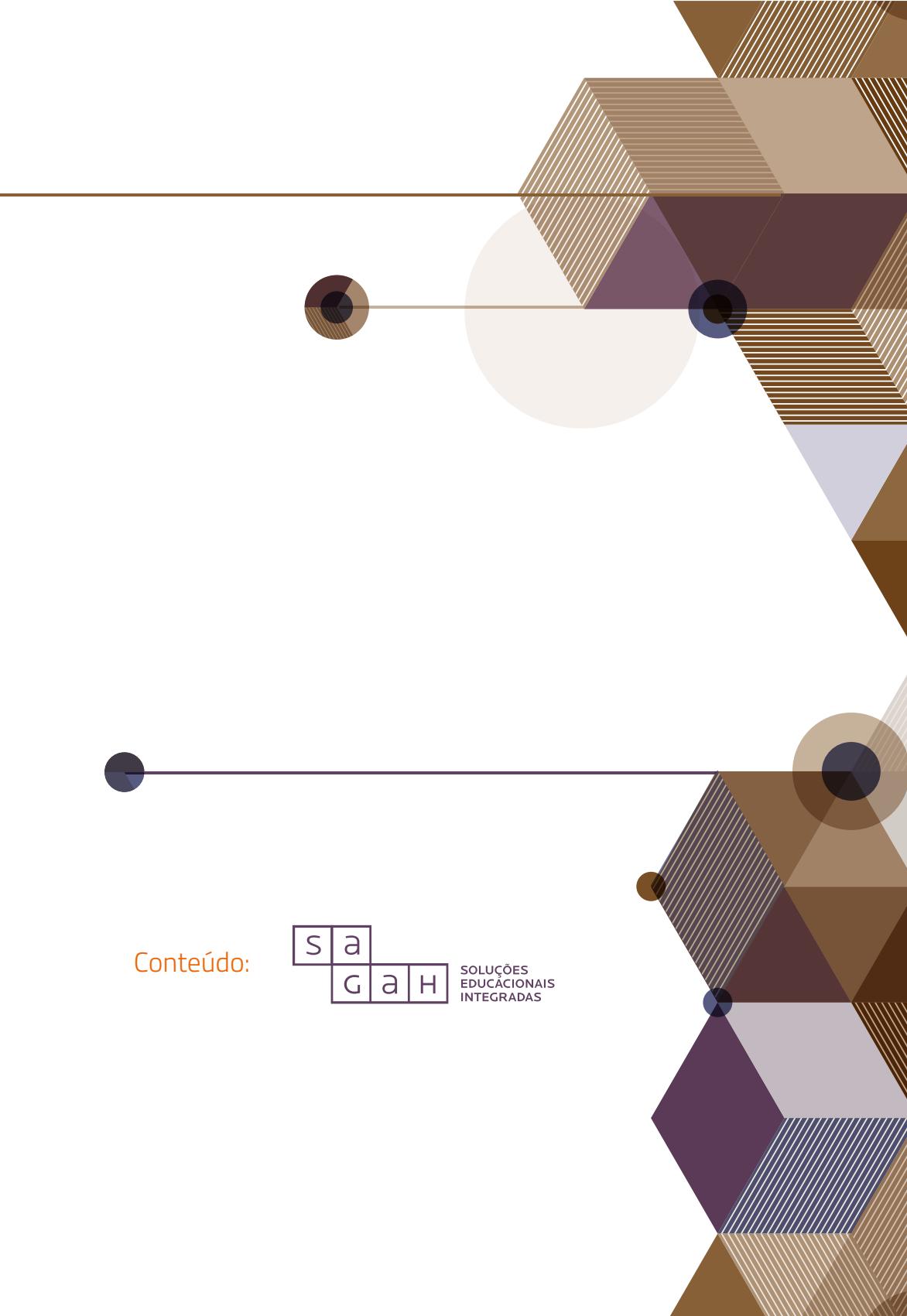
Podemos notar que cada detalhe é descrito juntamente com suas particularidades e pode ser inserida a documentação do projeto por um meio específico, seja por um diagrama, ou por uma descrição textual. As regras de negócio, assim como as demais informações de um software, visam a trazer o desenvolvimento correto de um sistema de informação.



Referências

- ALVARENGA, G. G. *Uma abordagem para tratamento de regras de negócio em sistemas de informação*. 149 f., 2007,. Dissertação (Mestrado em Informática) - Universidade Federal de Goiás, Goiânia, 2007. Disponível em: <http://www.inf.ufg.br/sites/portal.inf.ufg.br.mestrado/files/ds_Geoflavia.pdf>. Acesso em: 27 ago. 2017.
- LAGUNA, F. Gerenciamento de requisitos sem mistério. *YouTube*, 2016. Disponível em: <<https://www.youtube.com/watch?v=jajQyzOpLaE>>. Acesso em: 27 ago. 2017.
- MORGAN, T. *Business Rules and Information Systems: aligning IT with Business Goals*. Boston: Addison Wesley, 2001.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016.
- SCHACH, S. R. *Engenharia de software: os paradigmas clássicos e orientado a objetos*. 7. ed. Porto Alegre: AMGH, 2010.
- SOMMERVILLE, I. *Engenharia de software*. 9. ed. São Paulo: Pearson, 2011.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.



Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Dica do Professor

Neste vídeo você irá conhecer o conceito de requisitos de *software*, seus tipos e características. Você também irá conhecer os principais problemas enfrentados durante a coleta de requisitos.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

1) O que é um requisito de *software*?

- A) Um requisito pode ser definido como uma condição ou uma capacidade com a qual o sistema deve estar de acordo.
- B) É uma declaração sobre políticas ou condições que devem ser satisfeitas.
- C) É uma técnica para a medição de projetos de desenvolvimento de *software*, visando estabelecer uma medida de tamanho, em Pontos de Função (PF), considerando a funcionalidade implementada, sob o ponto de vista do usuário.
- D) É uma técnica de desenvolvimento de *software* em que se utiliza camadas.
- E) É um conjunto de elementos que um *software* entrega, podendo ser dados ou valores.

2) Qual é a característica de um requisito funcional?

- A) Definem propriedades e restrições do sistema.
- B) Descrevem explicitamente as funcionalidades e serviços do sistema.
- C) É mais voltado para características que podem ser mensuradas e testadas facilmente.
- D) Expressam informações relacionadas com a segurança do sistema.
- E) Expressam informações relacionadas com a arquitetura do sistema.

3) Qual é a característica de um requisito não funcional?

- A) É um tipo de requisito que o usuário geralmente conhece bem.
- B) É um tipo de requisito fácil de estimar.
- C) É um tipo de requisito que define propriedades e restrições do sistema. É mais voltado para características que podem ser mensuradas e testadas facilmente.

- D) É um tipo de requisito que geralmente descreve explicitamente as funcionalidades e serviços do sistema.
- E) É um tipo de requisito que é flexível e não impacta no desenvolvimento.

4) O que é uma regra de negócio?

- A) Regras de negócio são premissas e restrições aplicadas a uma operação comercial de uma empresa, que precisam ser atendidas para que o negócio funcione da maneira esperada.
- B) Definem propriedades e restrições do sistema.
- C) É um tipo de requisito que geralmente descreve explicitamente as funcionalidades e serviços do sistema.
- D) É um requisito que o usuário não conhece muito bem durante a criação de um sistema.
- E) É um tipo de requisito difícil de estimar.

5) Na engenharia de *software*, existe um processo genérico de levantamento e análise que contém as seguintes atividades: compreensão do domínio, coleta de requisitos, classificação, resolução de conflitos, definição das prioridades e verificação de requisitos. Uma das atividades mais importantes deste processo é a coleta de requisitos. Informe quais das descrições a seguir melhor descrevem esta atividade:

- A) Essa atividade considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes.
- B) Quando múltiplos *stakeholders* estão envolvidos, os requisitos apresentarão conflitos. Essa atividade tem por objetivo solucionar esses conflitos.
- C) Nesta atividade, os requisitos são verificados para descobrir se estão completos e consistentes e se estão em concordância com o que os *stakeholders* desejam do sistema.
- D) Em qualquer definição de requisitos, alguns serão mais importantes do que outros. Esse estágio envolve interação com os *stakeholders* para a definição dos requisitos mais importantes.
- E) É o processo de interagir com os *stakeholders* do sistema para descobrir seus requisitos.

Na prática

Ao reconhecer um requisito de *software*, devemos ter atenção para descrevê-lo de maneira clara. Lembre-se que um bom requisito especifica claramente algo que é necessário, verificável ou atingível no sistema.

Observe a seguinte descrição de um sistema:

Conteúdo interativo disponível na plataforma de ensino!

Este é um exemplo de descrição do negócio de uma farmácia e seu escopo, e também os requisitos funcionais e não funcionais coletados para criar este sistema. Com esta lista de requisitos, o engenheiro de *software* pode documentá-los e detalhar melhor o que cada um significa e como será trabalhado durante o desenvolvimento do sistema.

Saiba mais

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Amplie seus conhecimentos sobre uma proposta para modelagem de requisitos não-funcionais em projetos ágeis, na próxima dica de leitura.



Aponte a câmera para o código e accese o link do conteúdo ou clique no código para accesar.

Os requisitos de software podem ser divididos em requisitos funcionais, não funcionais e regra de negócio. Para saber mais sobre requisitos de software, accesse a próxima dica de leitura.



Aponte a câmera para o código e accese o link do conteúdo ou clique no código para accesar.

Acompanhe a leitura do artigo seguinte, que exemplifica como o gerenciamento de requisitos de software pode beneficiar o trabalho de uma equipe em um projeto de desenvolvimento.



Aponte a câmera para o código e accese o link do conteúdo ou clique no código para accesar.