



ENGENHARIA DE *SOFTWARE*

Aline Zanin

Izabelly Soares
de Morais

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

Analisar e preparar lista de requisitos

Objetivos de aprendizagem

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

- Diferenciar requisitos de negócio, de cliente, de sistema e de usuários.
- Elaborar a documentação de requisitos e regras de negócio.
- Classificar os requisitos por prioridade (essencial, importante e desejável).

Introdução

Você sabe por que é importante diferenciar requisitos de negócio, de cliente, de sistema e de usuários? Cada um desses requisitos corresponde a elementos que compõem o sistema, então a definição de cada um é essencial para o sucesso do projeto.

O engenheiro de software deve elaborar a documentação de requisitos e regras de negócio expressando as necessidades e características do sistema, além de classificar os requisitos por prioridade (essencial, importante e desejável) para que durante o desenvolvimento sejam enfatizados os requisitos essenciais para o funcionamento do sistema.

Neste capítulo, você vai adquirir conhecimentos fundamentais sobre requisitos de software. Vai conhecer a diferença de cada requisito e suas classificações de prioridade, além de aprender como preparar a documentação de requisitos e regras de negócio.

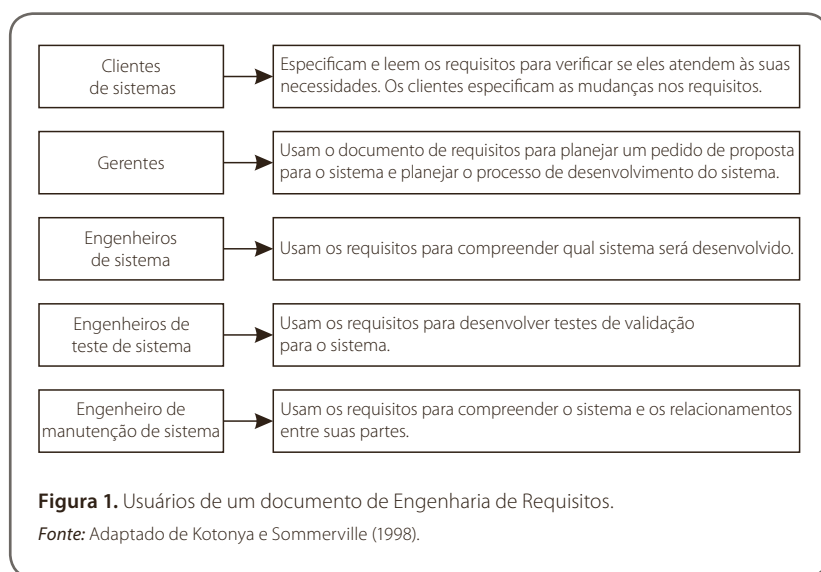
Requisitos

Um sistema computacional atende a diversas demandas quase que simultaneamente. Para isso, ele deve trazer entre suas funcionalidades características que atendam essas diferentes vertentes. Dentre elas, podemos destacar as necessidades de um cliente, de negócio (que traz restrições ao sistema conforme

o negócio do cliente), de sistema e até de usuário, que será o responsável por fazer o uso do sistema quando este for finalizado.

A seleção das funcionalidades de um software se baseia em diversos aspectos, dentre eles temos diversos meios de se obter essas informações do cliente, para que o software atinja suas expectativas. É importante lembrarmos que um software é algo complexo, e dentro dessa complexidade ele traz milhares de minuciosos detalhes que devem estar bem estabelecidos pela equipe de desenvolvimento. Para isso, é interessante que haja a criação de uma documentação de requisitos do software, que trará todos os requisitos que o sistema deverá ter. Isso ajudará a nortear a equipe durante o ciclo de desenvolvimento.

De acordo com Sommerville (2011, p. 63), “o documento de requisitos pode ser chamado também de Especificação de Requisitos de Software (SRS – do inglês *Software Requirements Specification*), é uma declaração oficial de o que os desenvolvedores do sistema devem implementar. Deve incluir tanto os requisitos de usuário para um sistema quanto uma especificação detalhada dos requisitos de sistema”. O autor fomenta que, “em alguns casos, os requisitos de usuário e de sistema são integrados em uma única descrição. Em outros, os requisitos de usuário são definidos em uma introdução à especificação de requisitos de sistema”. A Figura 1 a seguir mostra os possíveis usuários do documento e como eles o utilizam.



Os requisitos do usuário trazem restrições e serviços que serão ofertados ao cliente. Eles expõem os requisitos funcionais e não funcionais, nos quais todas estas informações são passadas aos clientes por meio claro e simples, tendo em vista que não faz parte do contexto do cliente algumas informações técnicas. Já os requisitos do sistema constituem de todo um detalhamento acerca das funções e de suas respectivas limitações funcionais, já que são direcionados ao uso da equipe em si, podem trazer detalhes técnicos, desde que descrevam apenas o comportamento externo do software. Dessa forma, é notória a quantidade de requisitos que um sistema tem, assim como as diversas áreas as quais eles devem atender, ou seja, não se limitam apenas às funcionalidades gerais do software e da necessidade do cliente, devem atender a outros fatores, como as regras do negócio e até as necessidades do próprio sistema.



Fique atento

Uma forma muito comum de se adquirir os requisitos de um software é por meio do uso de câmeras filmadoras. Dessa forma, a equipe de desenvolvimento pode rever toda a conversa tida com o cliente e, assim, extrair informações que muitas vezes não ficam tão explícitas!

Documento de requisitos

O documento de requisitos deve ser elaborado conforme as particularidades do software que está sendo desenvolvido. Jamais um documento servirá para outro software, pois cada sistema tem suas características próprias, assim como metodologias utilizadas, ferramentas, dentre outros detalhes. Porém, o que pode ser comum em vários projetos diferentes é a estrutura do documento.

Uma das possíveis organizações do documento de requisitos pode ser baseada em uma norma IEEE para documentos de requisitos (IEEE, 2009). Conforme Sommerville (2011, p. 64), essa é uma norma genérica que pode ser adaptada para usos específicos. O autor incluiu informações sobre evoluções que podem ser previstas no sistema, as quais ajudam os Engenheiros e a prestação de suporte em futuros recursos no sistema. Para isso, expôs estas informações no Quadro 1 a seguir.

Quadro 1. Estrutura de um documento de requisitos.

Capítulo	Descrição
Prefácio	Deve definir os possíveis leitores do documento e descrever seu histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das mudanças feitas em cada versão.
Introdução	Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como ele vai funcionar com outros sistemas. Também deve descrever como o sistema atende aos objetivos globais de negócio ou estratégicos da organização que encomendou o software.
Glossário	Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou o conhecimento do leitor.
Definição de requisitos de usuário	Deve descrever os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.
Arquitetura do sistema	Deve apresentar uma visão geral em alto nível da arquitetura do sistema previsto, mostrando a distribuição de funções entre os módulos do sistema. Componentes de arquitetura que são reusados devem ser destacados.
Especificação de requisitos do sistema	Deve descrever em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.

(Continua)

(Continuação)

Quadro 1. Estrutura de um documento de requisitos.

Capítulo	Descrição
Modelos do sistema	Pode incluir modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.
Evolução do sistema	Deve descrever os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário, etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, por exemplo. Os requisitos de hardware definem as configurações mínimas ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.
Índice	Vários índices podem ser incluídos no documento. Pode haver, além de um índice alfabético normal, um índice de diagramas, de funções, entre outros pertinentes.

Fonte: Sommerville (2011, p.65)

Já Pressman e Maxim (2016) apresenta os seguintes campos para comporem um documento de requisitos:

I. Introdução

1. Referências do sistema
2. Descrição geral
3. Restrições de projeto do software

II. Descrição da informação

1. Representação do fluxo de informação
 - a) Fluxo de dados
 - b) Fluxo de controle
2. Representação do conteúdo de informação
3. Descrição da interface com o sistema

III. Descrição funcional

1. Divisão funcional em partições
2. Descrição funcional
 - a) Narrativas
 - b) Restrições/limitações
 - c) Exigências de desempenho
 - d) Restrições de projeto
 - e) Diagramas de apoio
3. Descrição do controle
 - a) Especificação do controle
 - b) Restrições de projeto

IV. Descrição comportamental

1. dados do sistema
2. Eventos e ações

V. Critérios de validação

1. Limites de desempenho
2. Classes de testes
3. Reação esperada do software
4. Considerações especiais

VI. Bibliografia

VII. Apêndice

Podemos ver que Pressman e Maxim deixa os campos de seu modelo de documento de requisitos um pouco mais detalhados do que o de Sommerville, sem contar que cada autor acrescenta ou reduz alguma informação que julga supérflua para tal artefato. Os autores Humble e Faley (2013) trazem uma versão alternativa para as que foram apresentadas pelos outros autores, na qual destacam os seguintes tópicos:

- Sumário
- Visão geral do produto
- Ambientes de desenvolvimento, operação e manutenção
- Interfaces externas e fluxo de dados
- Requisitos funcionais
- Requisitos de desempenho
- Tratamento de exceções
- Prioridades de implementação
- Antecipação de mudanças e extensões
- Dicas e diretrizes de *design*
- Critérios de aceitação
- Índice remissivo
- Glossário

Apesar de possuir algumas características que podem ser vistas como padrões, por serem estabelecidas pela IEEE, um documento de requisitos deve remeter às características do projeto pelo qual foi desenvolvido. O interessante de termos esse modelo é que podemos acrescentar ou retirar tópicos que não condizem com nosso projeto. Os autores Humble e Faley trazem conceitos que remetem ao design do sistema e até ao fluxo de dados. Perceba que a Engenharia de Software nos traz meios de organização para desenvolvermos um software que são totalmente adaptáveis.



Link

A Aracele Fassbinder (2013) traz um vídeo demonstrativo sobre o preenchimento do documento de requisitos:

<https://goo.gl/fjRkgB>



Classificação dos requisitos por prioridade

Diante de tamanha complexidade do software, devemos saber que a obtenção dos requisitos não termina após a concepção de sua documentação, inclusive outra informação que um documento de requisitos deve nos trazer é sobre a prioridade dos requisitos, que podem ser classificados como essencial, importante e desejável, ou, até mesmo, como prioridade baixa, média e alta. Essas definições dependem muito da nomenclatura que a equipe decidir utilizar. Porém, o objetivo é o mesmo, classificar o requisito conforme a prioridade de implementação e importância dele perante o sistema. Essa classificação é utilizada no gerenciamento do escopo das etapas do projeto e na definição das prioridades durante o desenvolvimento do sistema.

- **Essencial:** requisito sem o qual o sistema não entra em funcionamento; requisitos essenciais são requisitos imprescindíveis e devem ser implementados desde as primeiras versões do sistema.
- **Importante:** requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória; requisitos importantes devem ser implementados o mais rápido possível; se não o forem, parte do sistema poderá ser implantada mesmo assim.
- **Desejável:** requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele; requisitos desejáveis são requisitos que não comprometem o funcionamento do sistema.

A Figura 2 a seguir traz um exemplo prático da classificação de um requisito quanto a sua prioridade.

[RF001]: Fazer Login

O sistema disponibiliza a opção para o usuário inserir seus dados de acesso.

Prioridade:



Essencial



Importante



Desejável

Figura 2. Exemplo de classificação de requisito quanto à prioridade.

A prioridade de um requisito faz parte dos demais atributos que um requisito pode ter. Ele especifica a prioridade do requisito com relação às propriedades de priorização selecionadas, por exemplo: “importância para aceitação pelo mercado”, “ordem de implementação” e “custo da oportunidade perdida em caso de não implementação” (POHL; RUPP, 2012). No caso do exemplo mostrado, é essencial que o usuário faça o *login* no sistema para que possa ter acesso as suas funcionalidades. Dessa forma, se torna obrigatório que esse requisito seja implementado. Esse procedimento é realizado para todos os requisitos funcionais e não funcionais do sistema.



Referências

FASSBINDER, A. Preenchimento do documento de requisitos. *YouTube*, 2013. Disponível em: <<https://www.youtube.com/watch?v=FeLtUI-zlVQ>>. Acesso em: 15 set. 2017.

HUMBLE, J.; FARLEY, D. *Entrega contínua: como entregar software*. Porto Alegre: Bookman, 2013. 496p.

IEEE STANDARD. *830-1998 IEEE Recommended Practice for Software Requirements Specifications*. Washington: IEEE Computer Society, 2009. Disponível em: <<https://standards.ieee.org/findstds/standard/830-1998.html>>. Acesso em: 25 set. 2017.

KOTONYA, P.; SOMMERVILLE, I. *Requirements Engineering: processes and techniques*. Somerset: John Wiley & Sons, 1998.

POHL, K.; RUPP, C. *Requirements engineering fundamentals: a study guide for the certified professional for Requirements Engineering exam*. Victoria: IREB, 2012.

PRESSMAN, R. S., MAXIM, B. R. *Engenharia de Software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, I. *Engenharia de software*. 9. ed. São Paulo: Pearson, 2011.

Leituras recomendadas

KECHI, H. *Engenharia de Software: qualidade e produtividade com tecnologia*. Rio de Janeiro: Elsevier – Campus, 2011. 232 p.

PFLEEGER, S. *Engenharia de Software: teoria e prática*. 2. ed. Upper Saddle River: Pearson/Prentice-Hall, 2004.

SCHACH, S. R. *Engenharia de software: os paradigmas clássicos e orientado a objetos*. 7. ed. Porto Alegre: AMGH, 2010.

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