



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DO RIO GRANDE  
DO SUL  
Campus Canoas

# Engenharia de Requisitos

Mariano Nicolao  
[mariano.nicolao@canoas.ifrs.edu.br](mailto:mariano.nicolao@canoas.ifrs.edu.br)

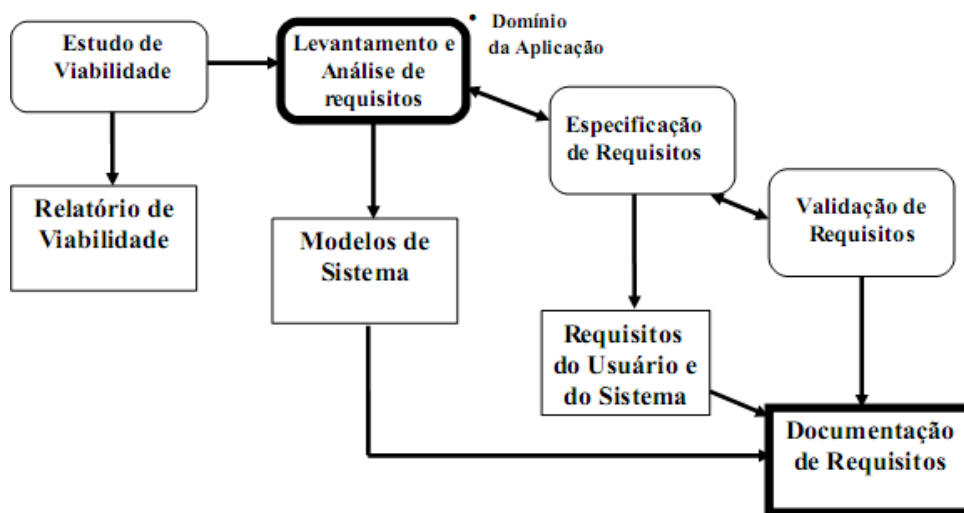
## **1. Introdução**

O desenvolvimento de um sistema de software que não atenda as reais necessidades de um cliente não será útil. Para evitar isto, deve se comunicar com o cliente e compreender as reais necessidades desse cliente para que se possibilite a construção de um sistema de software adequado que resolva e satisfaça as reais necessidades do cliente.

“A parte individual mais difícil da construção de um sistema de software é decidir o que construir. Nenhuma parte do trabalho danifica tanto o sistema resultante se for feita errado. Nenhuma outra parte é mais difícil de consertar depois.” (Fred Brooks apud PRESSMAN, 2010)

Neste contexto, a engenharia de requisitos (Figura 1), objetiva uma melhor compreensão do problema, a ser resolvido, por parte dos analistas ou engenheiros de software (PRESSMAN, 2010). A engenharia de requisitos (SOMMERVILLE, 2007) é o processo de desenvolvimento de uma especificação de software, na qual a funcionalidade do software deve ser definida. Ela se refere a desenvolver uma especificação que possa ser compreendida pelos usuários do sistema e uma especificação mais detalhada para os desenvolvedores do sistema.

A importância da engenharia de requisitos é possibilitar a criação de uma base sólida para o projeto e a implementação correspondente (PRESSMAN, 2010). Sem essa base, o software construído, provavelmente, não atenderá as necessidades do cliente.



**Figura 1: Engenharia de requisitos.**

Na seção 2, requisitos de software são apresentados. Na seção 3, se refere ao estudo de viabilidade. Na seção 4, se aborda a elicitação e análise de requisitos. Na seção 5, a especificação de requisitos é apresentada, na seção 6, considerações sobre a validação de requisitos são realizadas, e finalmente na seção 7, o documento de requisitos é apresentado.

## 2. Requisitos

Segundo SOMMERVILLE (2007), os requisitos estão diretamente associados às necessidades dos clientes de um sistema. Os requisitos de um sistema definem o que o sistema deve fazer, descrições das funções, e as suas restrições operacionais.

Um requisito é uma característica do sistema ou uma descrição de algo que o sistema é capaz de fazer em relação a atender os propósitos do sistema (PFLEEGER, 1998).

Frequentemente os requisitos de sistemas de software são classificados em requisitos funcionais, requisitos não funcionais ou requisitos de domínio (SOMMERVILLE, 2007).

### 1. Requisitos funcionais

Os requisitos funcionais correspondem a declarações de serviços que o sistema deve fornecer (SOMMERVILLE, 2007).

Exemplos de requisitos funcionais: apresentar catálogo de produtos, registrar venda, e autorizar pagamento a crédito.

## **2.2. Requisitos não funcionais**

São restrições sobre os serviços ou as funções oferecidos pelo sistema (SOMMERVILLE, 2007), incluem restrições sobre o tempo de resposta, sobre o processo de desenvolvimento e padrões.

Exemplos de requisito funcional: tempo de resposta apropriado nas interações com o sistema.

## **3. Requisitos de domínio**

São derivados do domínio de aplicação do sistema, refletem os fundamentos do domínio da aplicação (SOMMERVILLE, 2007), se esses requisitos não forem satisfeitos, o sistema não funcionará satisfatoriamente.

Exemplo: Deve existir uma interface com o usuário-padrão para todos os bancos de dados e que deverá ser baseada no padrão Z39.50

## **4. Requisitos de usuário e de sistema**

Também são considerados os requisitos de usuário e requisitos de sistema (SOMMERVILLE, 2007).

Os requisitos de usuário correspondem a descrição dos requisitos funcionais e não funcionais, de forma que sejam fácil de entender por parte dos usuários do sistema (não especialistas).

Os requisitos de sistema (SOMMERVILLE, 2007), correspondem a versões expandidas dos requisitos de usuário, adicionam detalhes e explicam como os requisitos de usuário devem ser fornecidos pelo sistema. São usados pelos engenheiros de software como ponto de partida para o projeto do sistema. Correspondem a especificação do sistema, isto é, o que. Eles não correspondem a como o sistema pode ser projetado ou implementado.

## **3. Estudo de viabilidade**

O estudo de viabilidade representa o início do processo de engenharia de requisitos. O estudo de viabilidade tem como entrada os requisitos de negócio, um esboço da descrição do sistema e como o sistema pretende apoiar os processos de negócios (SOMMERVILLE, 2007).

O estudo de viabilidade (SOMMERVILLE, 2007) é um estudo que responde, principalmente, a seguinte questão: o sistema contribui para os objetivos gerais da organização?, outras questões que devem ser respondidas são: o sistema pode ser implementado com tecnologia atual e dentro das restrições definidas de custo e prazo?, e o sistema pode ser integrado a outros sistemas já implantados?

Se um sistema não contribui para os objetivos da organização, esse sistema não é relevante para a organização e para esse sistema deve ser realizada uma recomendação de não prosseguir no relatório de estudo de viabilidade (SOMMERVILLE, 2007).

#### **4. Elicitação e análise de requisitos**

O domínio da aplicação, serviços a serem fornecidos pelo sistema e restrições são compreendidos pelos engenheiros de software através das interações com os clientes e usuários (SOMMERVILLE, 2007). As atividades do processo de elicitação e análise de requisitos são (SOMMERVILLE, 2007):

- 1.Obtenção de requisitos: representa o processo de interação com os stakeholders<sup>1</sup> para coletar seus requisitos.
- 2.Classificação e organização de requisitos: Se refere a agrupar os requisitos relacionados em grupos coerentes.
- 3.Priorização e negociação de requisitos: atividade focada a priorização dos requisitos e à resolução de conflitos (devido a participação de vários stakeholders) via negociação.
- 4.Documentação de requisitos: implica a produção de documentos de requisitos formais ou informais.

Segundo SOMMERVILLE (2007), a elicitação e análise de requisitos são um processo iterativo (Figura 2). A compreensão dos requisitos por parte do analista aumenta a cada volta do ciclo do processo iterativo. Para cada volta do processo são realizadas as atividades: obtenção de requisitos, classificação e organização de requisitos, priorização e negociação de requisitos e documentação de requisitos.

---

<sup>1</sup> Stakeholder: termo usado para se referir a qualquer pessoa ou grupo afetado pelo sistema (exemplos: usuários finais, engenheiros, gerentes de negócios e especialistas do domínio).



**Figura 2: Processo de elicitação e análise de requisitos.**  
**Fonte: (SOMMERVILLE, 2007)**

### **1. Obtenção de requisitos**

Os requisitos de usuário e de sistema são obtidos através do processo que reúne informações sobre o sistema proposto (SOMMERVILLE, 2007). Várias fontes de informação são consideradas, nesta fase: documentos, usuários, analistas e especificações de sistemas similares.

Entrevistas são utilizadas nas interações com os stakeholders, também observações são úteis. O uso de cenários e protótipos auxilia na obtenção de requisitos (SOMMERVILLE, 2007).

#### **1. Entrevista**

A realização de entrevistas (formais ou informais) faz parte do processo de engenharia de requisitos. Os requisitos são obtidos a partir das respostas dos usuários ou clientes às questões colocadas pelo engenheiro de requisitos sobre o sistema existente ou novo sistema a ser desenvolvido (SOMMERVILLE, 2007).

A utilidade das entrevistas reside na compreensão geral de o que os usuários fazem, como interagem com o sistema, e dificuldades e necessidades atuais.

Em relação ao domínio da aplicação, é necessária a realização de entrevistas com um especialista do domínio, preferencialmente que faz parte da organização, ou a consulta de documentos ou informações pertinentes ao domínio.

As entrevistas possibilitam a obtenção de informações que complementam informações obtidas a partir de observações, documentos consultados, etc.

#### 4.1.2 Cenários

Um cenário representa um exemplo de uma sessão de interação do usuário com o sistema de software. Cada cenário implica várias interações possíveis.

Na elaboração de um cenário se deve, principalmente, incluir (SOMMERVILLE, 2007):

1. Uma descrição do que os usuários esperam do sistema no início do cenário.
2. Uma descrição do fluxo normal (interações sem quaisquer entradas fora do normal ou condições de erros) de eventos do cenário.
3. Uma descrição do que pode dar errado (erros do usuário e falhas) como isso é tratado.
4. Informações sobre outras atividades que ocorrem simultaneamente.

A seguir um exemplo de cenário de fluxo normal (Figura 3) e um exemplo de cenário com exceções (Figura 4) no contexto de um caixa eletrônico de um banco (ATM)(RUMBAUGH ET AL., 1994) com acesso a vários bancos.

A ATM (caixa eletrônico de um banco 24 horas com acesso a vários bancos) solicita que o usuário introduza um cartão; o usuário introduz um cartão magnético.  
A ATM aceita o cartão e lê seu número de série.  
A ATM solicita a senha; o usuário introduz "1234".  
A ATM verifica o número de série e a senha com o consórcio; o consórcio faz a confirmação com o banco "39" e notifica a ATM sobre a aceitação.  
A ATM solicita que o usuário escolha o tipo de transação (retirada, depósito, transferência, consulta); o usuário especifica retirada.  
A ATM solicita a quantia em dinheiro; o usuário introduz 100.  
A ATM verifica que o valor está dentro dos limites pré-fixados e solicita que o consórcio processe a transação; o consórcio passa a solicitação para o banco; que normalmente confirma o sucesso e informa o novo saldo da conta.  
A ATM entrega o dinheiro e solicita que o usuário o recolha; o usuário recolhe o dinheiro.

**Figura 3: Exemplo de cenário normal de um caixa eletrônico (ATM).**

A ATM (caixa eletrônico de um banco 24 horas com acesso a vários bancos) solicita que o usuário introduza um cartão; o usuário introduz um cartão magnético.  
A ATM aceita o cartão e lê seu número de série.  
A ATM solicita a senha; o usuário introduz “9999”.  
A ATM verifica o número de série e a senha com o consórcio, que os rejeita após consultar o banco apropriado.  
A ATM indica senha inválida e solicita que o usuário a reintroduza; o usuário introduz “1234” que a ATM verifica com sucesso com o consórcio.  
A ATM solicita que o usuário escolha o tipo de transação; o usuário escolhe retirada  
A ATM solicita a quantia de dinheiro; o usuário muda de ideia e digita “cancelar”  
A ATM ejeta o cartão e solicita que o usuário o recolha; o usuário assim faz.  
A ATM solicita que o usuário introduza um cartão

**Figura 4: Exemplo de cenário de um caixa eletrônico (ATM) com exceções.**

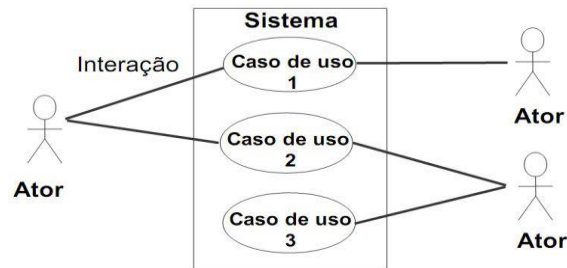
#### **4.1.3 Casos de uso**

Os casos de uso representam uma técnica baseada em cenário para elicitación de requisitos (SOMMERVILLE, 2007).

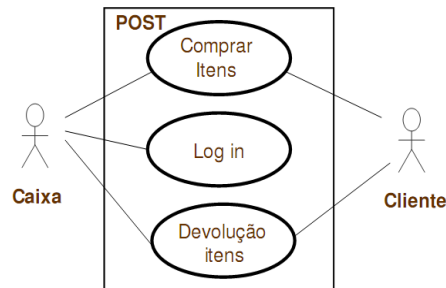
Um caso de uso identifica as interações com o sistema. Um caso de uso abrange vários cenários. Existirá um cenário para representar a interação normal e outros para cada possível exceção (SOMMERVILLE, 2007).

A UML (Linguagem Unificada de Modelagem) (BOOCH ET. AL, 2005) possibilita a representação de casos de uso através do diagrama de casos de uso (BOOCH ET. AL, 2005) (Figura 5). No diagrama de casos de uso da Figura 6, foram colocados três casos de usos (Comprar Itens, Log in e Devolução de itens) (LARMAN, 2000) e dois atores (Caixa e Cliente). Um caso de uso descreve o que um sistema (ou subsistema) faz, define funcionalidades ou comportamentos fornecidos pelo sistema. Um caso de uso representa a descrição de um conjunto de sequências de ações (BOOCH ET. AL, 2005) para produzir um resultado. Um ator, no diagrama de casos de uso, representa um papel que um ser humano, um dispositivo de hardware ou até outro sistema desempenha quando interage com o sistema.





**Figura 5: Diagrama de casos de uso segundo a notação da UML.**



**Figura 6: Exemplo de diagrama de casos de uso para um sistema de controle de venda de produtos no contexto de um supermercado.**

**Fonte: (LARMAN, 2000)**

Um caso de uso pode ser descrito textualmente de forma geral ou alto nível (Figura 7) ou de forma detalhada ou expandida (Figura 8) (LARMAN, 2000).

#### **Exemplo: Caso de uso de alto nível**

**Caso de Uso:** Comprar itens

**Atores:** Cliente, Caixa

**Descrição:** Um Cliente chega a um ponto de pagamento com itens que deseja comprar. O Caixa registra os itens da compra e recebe o pagamento. Quando concluída a compra, o cliente pode ir com os itens.

**Figura 7: Descrição geral ou alto nível do caso de uso Comprar itens.**

- **Caso de uso expandido (mais detalhes)**

<b>Use case:</b>	<b>Comprar itens com dinheiro</b>
<b>Atores:</b>	<b>Cliente, Caixa</b>
<b>Finalidade:</b>	<b>Capturar uma venda e o pagamento correspondente.</b>
<b>Descrição:</b>	<b>Um cliente chega ao ponto de pagamento com itens que deseja comprar. O caixa registra os itens da compra e recebe um pagamento, o qual pode necessitar autorização. Quando concluída a compra o Cliente pode ir com os itens.</b>

- **Sequência de eventos**

<u>Ação-Ator</u>	<u>Resposta - Sistema</u>
1. Começa quando um Cliente chega a um ponto de pagamento com vários itens	
2. O Caixa registra o identificador de cada item, também a quantidade	3. Determina o preço dos itens e acrescenta informação sobre o item à transação de vendas em andamento.
4. Quando concluída a entrada de itens O Caixa indica ao Post que a entrada de itens foi concluída	5. Calcula e mostra o Total da Venda

6. O Caixa informa ao Cliente o total
7. O Cliente faz pagamento em dinheiro
  - a - se pagamento em dinheiro - ver seção  
Pagar com Dinheiro
  - b - se pagamento com cartão - ver seção  
Pagar com Cartão de Crédito
  - c - se pagamento com cheque - ver seção  
Pagar com Cheque
8. O Caixa registra a venda completada
9. Atualiza estoque
10. Gera um recibo
11. O Caixa dá ao Cliente o recibo da compra.
12. O Cliente sai com os itens comprados.

## ● Sequência de eventos - Seção pagar com Dinheiro

<u>Ação-Ator</u>	<u>Resposta - Sistema</u>
1. O Caixa faz um pagamento com dinheiro “Valor fornecido” possivelmente maior que o total da venda.	
2. O Caixa registra a quantia fornecida	3. Apresenta o valor do troco
4. O Caixa deposita o dinheiro recebido e retira o troco. O caixa dá o troco ao Cliente.	

**Figura 8: Descrição detalhada ou expandida do caso de uso Comprar itens.**

### 4.1.4 Negociação de requisitos

Acontece na realidade que clientes e usuários as vezes solicitam mais, em relação aos recursos disponíveis, do que é possível realizar. Também requisitos conflitantes podem ser propostos por diferentes usuários. Neste contexto é necessário um processo de negociação. Os envolvidos, clientes, usuários e outros interessados são convidados a participar de reuniões e discutir os conflitos sobre os requisitos e estabelecer prioridades nos requisitos (PRESSMAN, 2010).

## 5. Especificação de requisitos

Frequentemente a linguagem natural é usada para redigir especificações de requisitos de sistema como de usuário.

É necessário escrever os requisitos de usuário em uma linguagem que os não especialistas consigam compreender. Neste contexto, notações para a especificação de requisitos têm sido propostas (SOMMERVILLE, 2007), como: linguagem natural estruturada (todos os requisitos são definidos de forma padronizada), modelos gráficos dos requisitos como os casos de uso (seção 4.1.3), até especificações matemáticas.

No contexto da linguagem natural estruturada consiste na definição de formulários padrão para expressar a especificação de requisitos.

Um exemplo de formulário padrão para a especificação de requisitos é apresentado na Figura 9 e um exemplo de especificação de requisito, pagamento no contexto de um sistema de gerenciamento de vendas, usando esse formulário padrão é ilustrado na Figura 10.

<b>Identificação do requisito:</b>
<b>Função:</b>
<b>Descrição:</b> descrição da função
<b>Entradas:</b> descrição de entradas
<b>Origem:</b> descrição da origem da entrada
<b>Saídas:</b> descrição de saídas
<b>Destino:</b> para onde as saídas prosseguirão
<b>Ação:</b> descrição da ação a ser tomada
<b>Requer:</b> quais outras entidades são usadas
<b>Precondição:</b> o que deve ser verdadeiro antes da chamada da função
<b>Pós-condição:</b> o que é verdadeiro após a chamada da função

**Figura 9: Um formulário padrão para a especificação de requisitos.**

<b>Identificação do requisito: R2.1</b> <b>Função:</b> Pagamento de uma compra de produtos a vista <b>Descrição:</b> Registrar o pagamento e imprimir o recibo <b>Entradas:</b> os itens (produtos) de uma venda <b>Origem:</b> Obtenção do preço dos produtos a partir da leitura do código de barras dos produtos <b>Saídas:</b> comprovante do pagamento <b>Destino:</b> Cliente <b>Ação:</b> Realizar o pagamento de produtos num supermercado, para isto realizar o registro correspondente no sistema, calcular o troco e imprimir o recibo. <b>Requer:</b> <b>Precondição:</b> Total da venda já foi calculado, verificar a disponibilidade do troco. <b>Pós-condição:</b> Pagamento foi realizado, atualização, no sistema, dos produtos vendidos e registro da venda realizada.
--

**Figura 10: Exemplo de especificação do requisito pagamento usando o formulário padrão.**

## 6. Validação de requisitos

A validação de requisitos objetiva mostrar que os requisitos realmente definem o sistema que o usuário deseja. Essa validação procura problemas com os requisitos, se encontrar erros em um documento de requisitos, devem ser corrigidos já que esses erros podem levar a custos excessivos de retrabalho, uma vez descobertos quando o sistema está em operação (SOMMERVILLE, 2007). Técnicas de requisitos podem ser usadas individualmente ou em conjunto: revisões de requisitos, protótipo e geração de casos de teste.

## 7. Documento de requisitos

O que deverá ser implementado pelos desenvolvedores é definido no documento de requisitos de software (SOMMERVILLE, 2007). Deve conter os requisitos de usuário e uma especificação detalhada dos requisitos de sistema.

Os usuários, possíveis, de um documento de requisitos são: clientes, gerentes, analistas e desenvolvedores participando na construção do sistema, testes e manutenção.

A estrutura proposta para um documento de requisitos, segundo o padrão IEEE/ANSI 830-1998, é apresentada na Figura 11.

- 1. Introdução**
  - 1.1 Propósito do documento de requisitos
  - 1.2 Escopo do produto
  - 1.3 Definições, acrônimos e abreviaturas
  - 1.4 Referências
  - 1.5 Visão geral do restante do documento
- 2. Descrição geral**
  - 2.1 Perspectiva do produto
  - 2.2 Funções do produto
  - 2.3 Características dos usuários
  - 2.4 Restrições gerais
  - 2.5 Suposições e dependências
- 3. Requisitos específicos** que abrangem requisitos funcionais, não funcionais e de interface.  
Os requisitos podem estar relacionados a interface, funcionalidades, desempenho do sistema, restrições, e características de qualidade.
- 4. Apêndices**

**Figura 11: A estrutura para um documento de requisitos, padrão IEEE/ANSI 830-1998.**

## **8. Resumo**

Este texto abordou a engenharia de requisitos.

Foi apresentado o que é um requisito, tipos de requisitos, a elicitação e a análise de requisitos.

Três técnicas de obtenção de requisitos foram apresentadas (entrevista, cenários e casos de uso).

Considerações sobre a validação de requisitos foram apresentadas.

Finalmente, foi apresentada a estrutura de um documento de requisitos segundo o padrão IEEE/ANSI 830-1998.

## **Referências Bibliográficas**

Booch G., Rumbaugh J., Jacobson I. UML: Guia do usuário, 2ª edição. Editora Campus. 2005.

Larman C. Utilizando UML e Padrões: uma introdução à análise e ao projeto orientado a objetos. Editora Bookman. Porto Alegre, 2000.

Pfleeger S. L. Software Engineering: theory and practice. Prentice-Hall. 1998.

Pressman R. Engenharia de Software, 6a edição, AMGH Editora. 2010.

Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W. Modelage e Projetos Baseados em Objetos. Editora Campus. 1994.

Sommerville, I. Engenharia de Software. 8a Edição. Addison Wesley. 2007.