

# Engenharia de software

**João Caldeira**

**Professor Convidado**

E-mail. [joaocarlos.caldeira@my.istec.pt](mailto:joaocarlos.caldeira@my.istec.pt)

Mob. +351 917769544

23 de novembro de 2022



# Requirements Engineering

23 de novembro de 2022



# Engenharia de Requisitos

## Quais são os requisitos ?

- **UMA projeto de software começa** quando alguém recebe um **ideia ou demanda** apresenta-se
  - nova empresa/organização
  - novo departamento
  - novo produto
  - novas regras e regulamentos
  - novos processos de negócios
- Os requisitos **diga o que** um sistema **deveria fazer** **sob quais restrições**
- **Requisitos** **área fundamental** parte de **comunicação** entre o cliente e a equipe de engenharia de software
  - Muitas vezes, eles fazem parte de um contrato legal assinado

# Engenharia de Requisitos

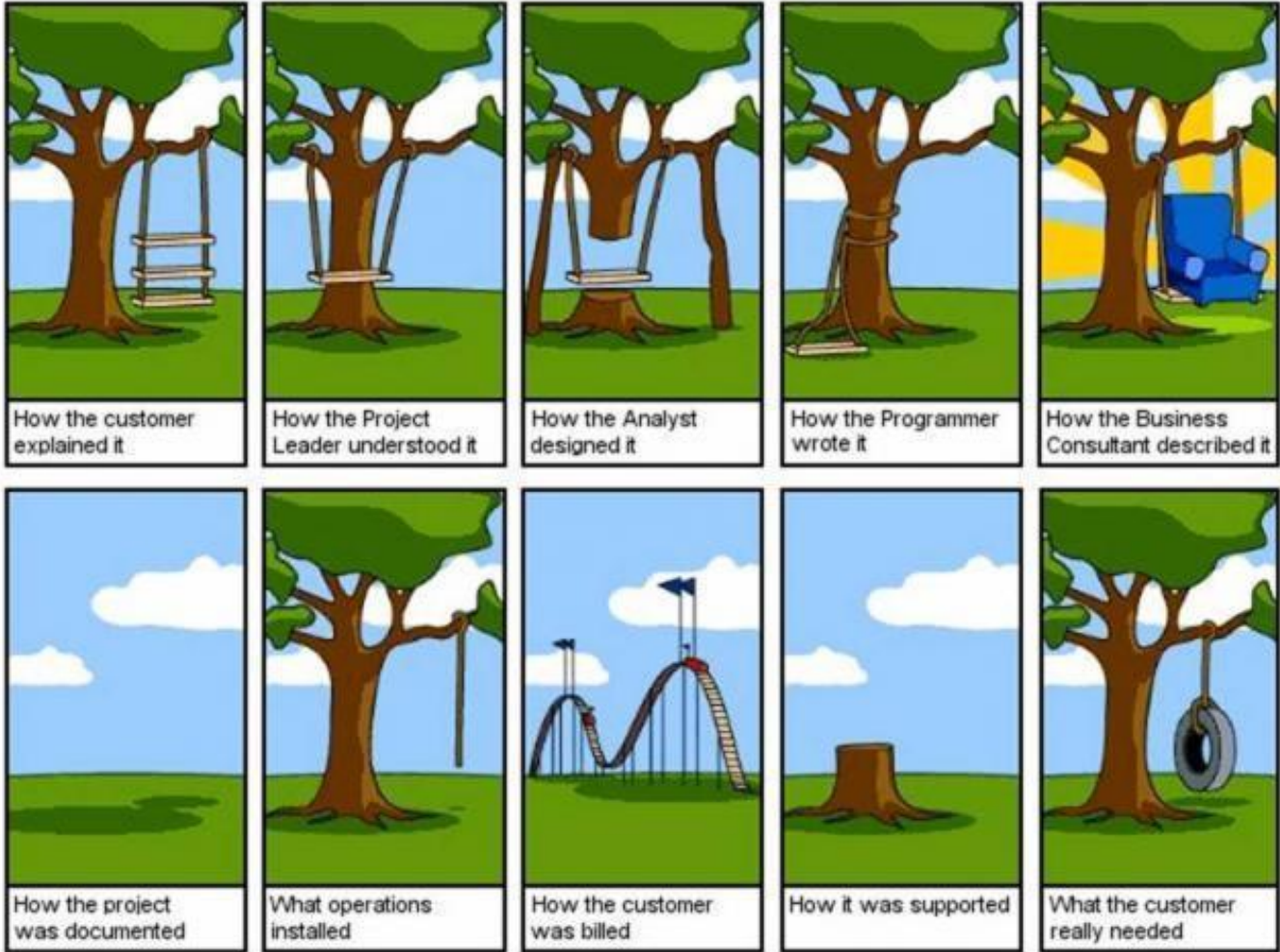
Definição no SWEBOK

- .....um **requisito de software** é uma **propriedade** que deve ser **exibido** em ordem **resolver** algum **problema** no mundo real.....



# Engenharia de Requisitos

Por que eles são necessários?



# Engenharia de Requisitos

Assuntos abordados

- **produto e processo** requisitos
  - **Funcional e não funcional** requisitos
- **Do utilizador** requisitos
- **Sistema** requisitos
- O software **documento de requisitos**

# Engenharia de Requisitos

## Tipos de requisitos

- **Requisitos do usuário**
  - Declarações em **linguagem natural e diagramas do Serviço** o sistema fornece e seu funcionamento **restrições**
  - Escrito para clientes
- **requisitos de sistema**
  - Um documento estruturado estabelecendo **descrições detalhadas das funções do sistema, serviços & operacional restrições**
  - Define o que deve ser implementado, **pode fazer parte de um contrato entre cliente e contratante**

# Engenharia de Requisitos

## Tipos de requisitos

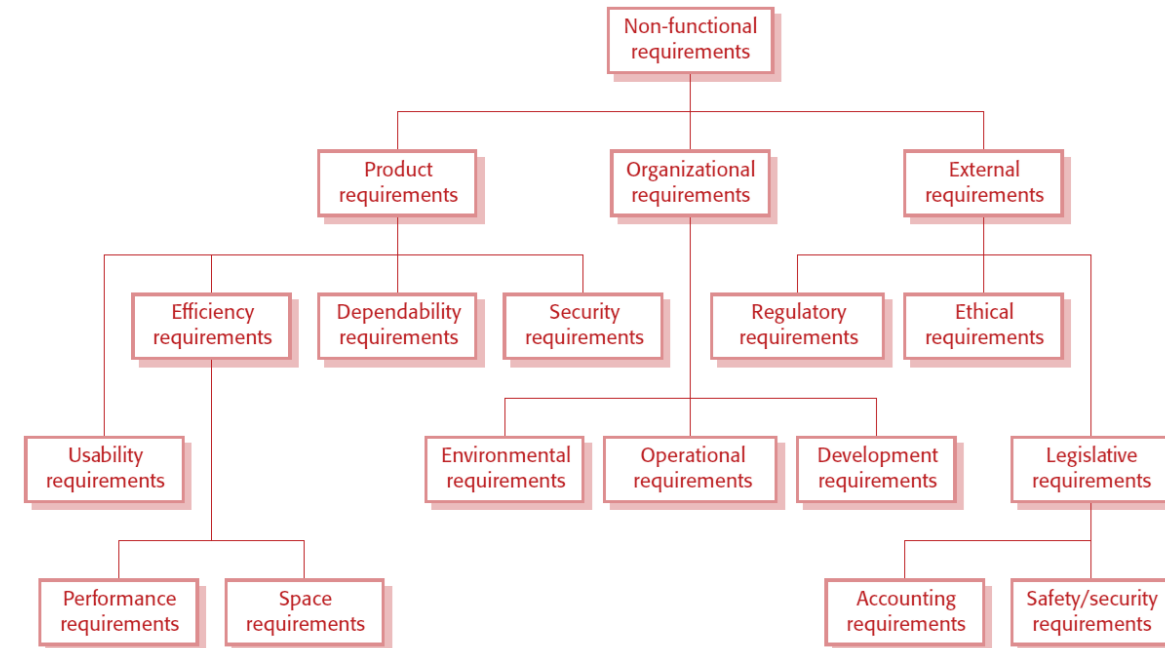
- **Requisitos funcionais**
  - Declarações sobre os serviços do sistema **deve fornecer**
    - Como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em situações específicas
  - Às vezes, também declara explicitamente o que o sistema **não deveria fazer**



# Engenharia de Requisitos

## Tipos de requisitos

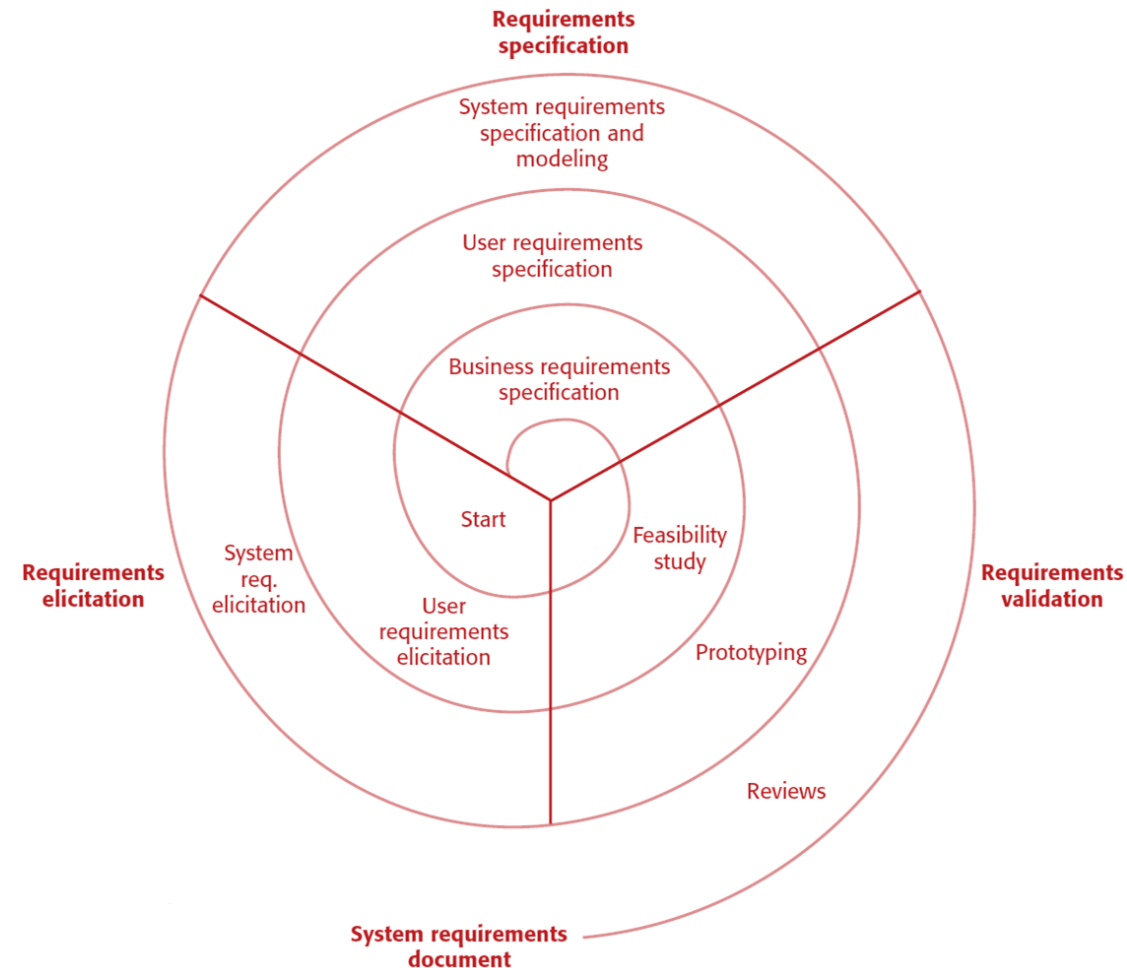
- **requisitos não Funcionais**
  - São restrições nos serviços ou funções oferecidas pelo sistema
    - Inclua restrições de tempo, restrições no processo de desenvolvimento e restrições imposta por padrões
    - Frequentemente se aplicam ao sistema global, não a recursos ou serviços individuais



# Engenharia de Requisitos

## Sobre o processo

- **Principais atividades**
  - **Elicitação e Análise.** Descobrir requisitos interagindo com as partes interessadas
  - **Especificação.** Convertendo esses requisitos em um formulário padrão
  - **Validação.** Verificar se os requisitos realmente definem o sistema que o cliente deseja

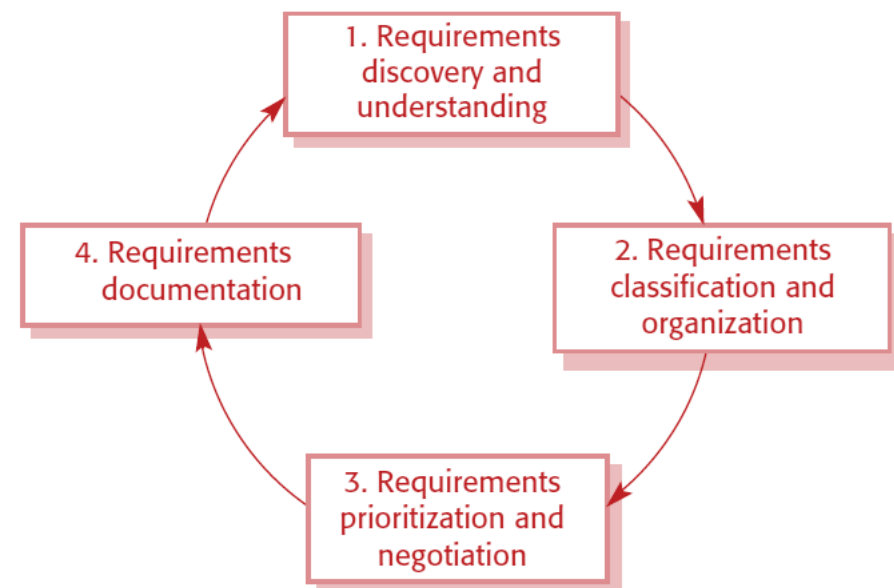


# Engenharia de Requisitos

## Sobre o processo

- **Principais atividades**

1. Interagir com as partes interessadas do sistema para descobrir seus requisitos
2. Pega a coleção não estruturada de requisitos, agrupa requisitos relacionados e os organiza em clusters coerentes
3. Preocupado em priorizar requisitos e encontrar e resolver conflitos
4. Os requisitos são documentados e inseridos na próxima rodada da espiral



# Engenharia de Requisitos

## Elicitação e Análise

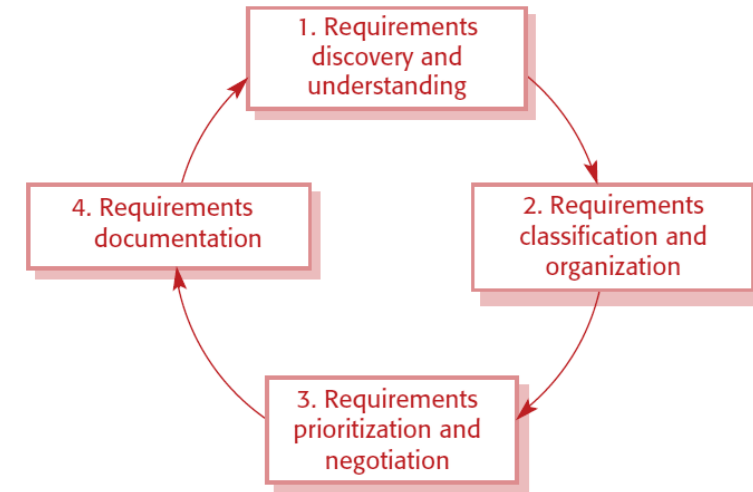
- **Técnicas**

- 1. Entrevistando.

- **entrevistas abertas** onde não há agenda
    - **Entrevistas fechadas** quando as partes interessadas respondem a um conjunto predefinido de perguntas

- 2. Etnografia.

- **técnica observacional** que pode ser usado para entender os processos operacionais e ajudar a derivar requisitos de software para apoiá-los
    - **Um analista mergulha** no ambiente de trabalho



# Engenharia de Requisitos

## Especificação

- **Definição**

Processo de escrever os requisitos do usuário e do sistema em um documento de requisitos

### **Documento de Requisitos**

1. A declaração oficial do que é exigido dos desenvolvedores do sistema
2. Deve incluir uma definição de **requisitos do usuário** e um **especificação dos requisitos do sistema**
3. Isso é **NÃO** um documento de projeto. Na medida do possível, deve definir **O QUE** o sistema deve fazer em vez de **COMO AS** deve fazer isso



# Engenharia de Requisitos

## Especificação

- **Especificação de linguagem natural (dois requisitos)**
  - Este método pode ser muito vago. Depende da habilidade do escritor em colocar os requisitos em palavras
  - **Tem pouco formalismo**

3.2 The system shall measure the blood sugar and deliver insulin, if required, every 10 minutes. *(Changes in blood sugar are relatively slow, so more frequent measurement is unnecessary; less frequent measurement could lead to unnecessarily high sugar levels.)*

3.6 The system shall run a self-test routine every minute with the conditions to be tested and the associated actions defined in Table 1. *(A self-test routine can discover hardware and software problems and alert the user to the fact the normal operation may be impossible.)*

# Engenharia de Requisitos

## Especificação

- **Alternativas à especificação de linguagem natural**
  - Às vezes, esses métodos podem ser combinados para ilustrar mais detalhadamente alguns requisitos do cliente
  - **É mais trabalho intensivo**

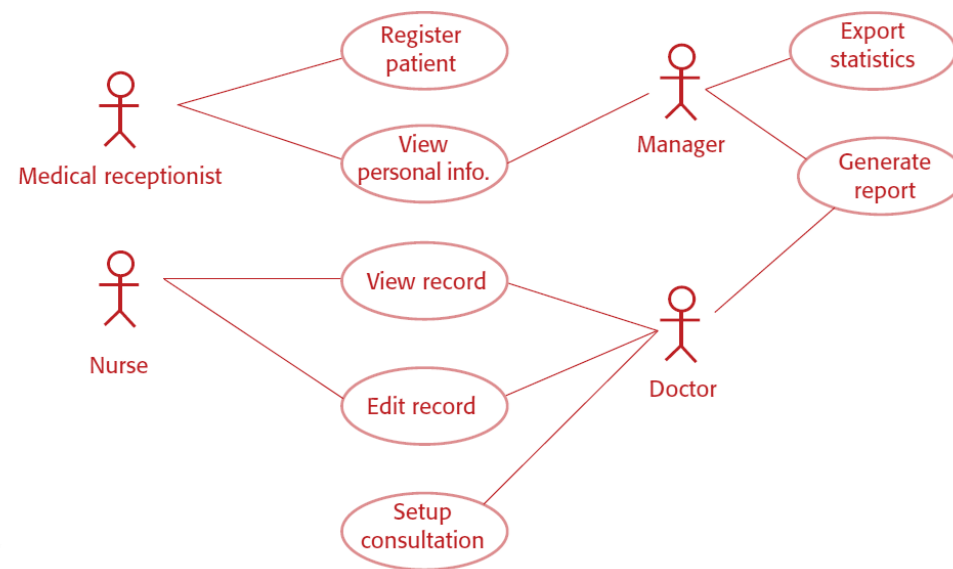
Notation	Description
Natural language sentences	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system. UML (unified modeling language) use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want, and they are reluctant to accept it as a system contract. (I discuss this approach, in Chapter 10, which covers system dependability.)

# Engenharia de Requisitos

## Especificação

- **Casos de uso**

- Os casos de uso são uma maneira de descrever as interações entre os usuários e um sistema usando um modelo gráfico e um texto estruturado
- **É relativamente fácil de realizar**



# Engenharia de Requisitos

## Especificação

- **Especificação estruturada (um requisito)**
  - Este método é **mais detalhado, portanto, mais preciso** sobre os requisitos do cliente.
  - **É mais trabalho intensivo**

### *Insulin Pump/Control Software/SRS/3.3.2*

<b>Function</b>	Compute insulin dose: Safe sugar level.
<b>Description</b>	Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.
<b>Inputs</b>	Current sugar reading (r2), the previous two readings (r0 and r1).
<b>Source</b>	Current sugar reading from sensor. Other readings from memory.
<b>Outputs</b>	CompDose—the dose in insulin to be delivered.
<b>Destination</b>	Main control loop.
<b>Action:</b>	CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered. (see Figure 4.14)
<b>Requires</b>	Two previous readings so that the rate of change of sugar level can be computed.
<b>Precondition</b>	The insulin reservoir contains at least the maximum allowed single dose of insulin.
<b>Postcondition</b>	r0 is replaced by r1 then r1 is replaced by r2.
<b>Side effects</b>	None.

# Engenharia de Requisitos

## Especificação

- **Documento de Requisitos de Software**
  - Depende do tipo de software que está sendo desenvolvido
  - Para um sistema de engenharia complexo que inclui hardware e software desenvolvido por diferentes empresas, o documento de requisitos provavelmente será longo e detalhado
  - Para um produto de software interno, muitos capítulos detalhados serão deixados de fora. O foco estará na definição dos requisitos do usuário e requisitos de sistema não funcionais de alto nível

Chapter	Description
Preface	This defines the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This describes the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This defines the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter presents a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.
System requirements specification	This describes the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. Interfaces to other systems may be defined.
System models	This chapter includes graphical system models showing the relationships between the system components and the system and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This describes the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.
Appendices	These provide detailed, specific information that is related to the application being developed—for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.

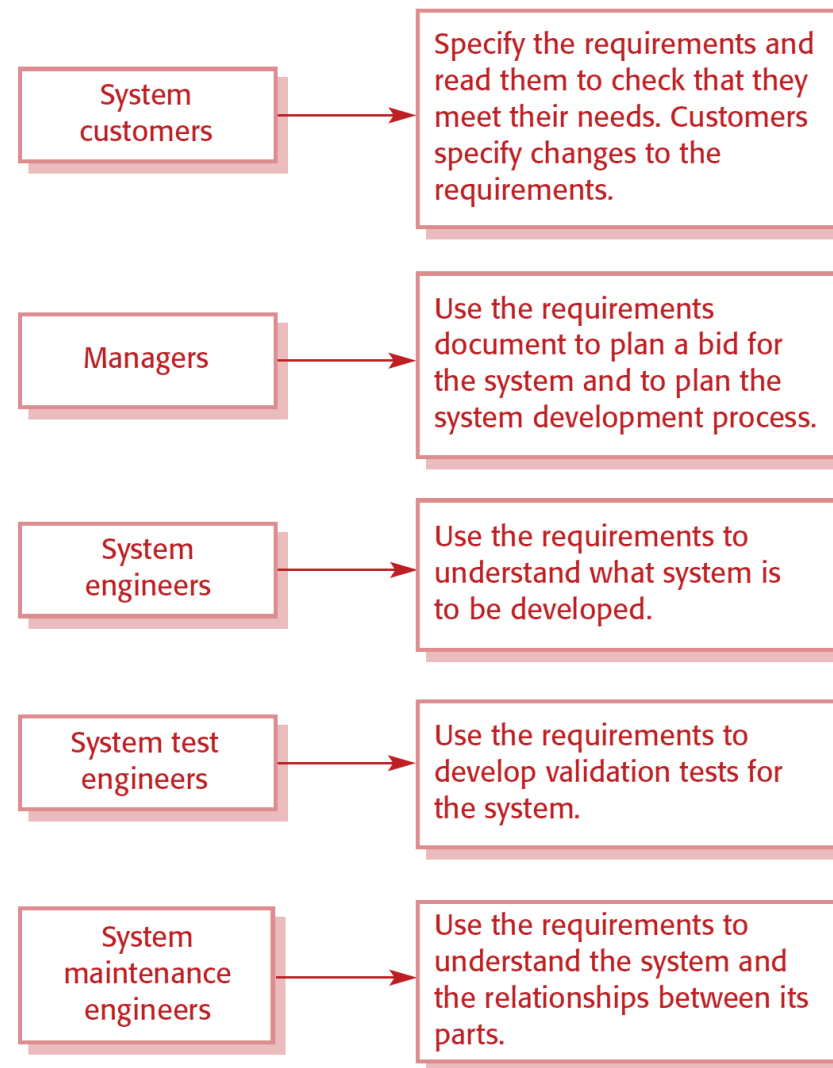


# Engenharia de Requisitos

## Especificação

- **Usuários do documento de requisitos**

- Os documentos de requisitos são essenciais quando os sistemas são terceirizados para desenvolvimento, quando diferentes equipes desenvolvem diferentes partes do sistema e quando uma análise detalhada dos requisitos é obrigatória
- O nível de detalhe que você deve incluir em um documento de requisitos depende do tipo de sistema que está sendo desenvolvido e do processo de desenvolvimento usado



# Engenharia de Requisitos

## Validação de Requisitos

- **Metas**

1. Preocupada em demonstrar que os requisitos definem o sistema que o cliente realmente deseja
2. Os custos de erros de requisitos são altos, então a validação é muito importante
  - A correção de um erro de requisitos após a entrega pode **custar até 100 vezes** o custo de corrigir um erro de implementação

# Engenharia de Requisitos

## Validação de Requisitos

- **Verificação de Requisitos**

- 1.Validade.**O sistema fornece as funções que atendem às necessidades do cliente?
- 2.Consistência.**Existem conflitos de requisitos?
- 3.Completude.**Todas as funções exigidas pelo cliente estão incluídas?
- 4.Realismo.**Os requisitos podem ser implementados de acordo com o orçamento, prazo e tecnologia disponíveis?
- 5.Verificabilidade.**Os requisitos podem ser verificados/testados?

# Engenharia de Requisitos

## Validação de Requisitos

- **Técnicas de Validação**

- 1. **Revisões de requisitos**

- **Revisões/análises manuais sistemáticas e regulares.** Executado enquanto a definição dos requisitos está sendo formulada, envolvendo o cliente. As revisões podem ser formais (com documentos) ou informais
    - **Cheques de revisão.** Verificabilidade (o requisito é testável?), Compreensibilidade (é entendido?), Rastreabilidade (sua origem está claramente declarada?), Adaptabilidade (pode ser alterado sem impacto em outros requisitos?)

- 2. **Prototipagem**

- Usando um modelo executável do sistema para verificar os requisitos

- 3. **Geração de casos de teste**

- Desenvolvimento de testes para requisitos para verificar a testabilidade

# Engenharia de Requisitos

## Validação de Requisitos

- **Gerenciamento de Requisitos**

O processo de gerenciamento de requisitos em mudança durante o processo de engenharia de requisitos e desenvolvimento do sistema

- 1. Ambiente comercial/técnico.** Pode mudar durante o desenvolvimento
- 2. Prioridade de requisitos.** Pode mudar durante o processo de desenvolvimento
- 3. Alguns requisitos podem se tornar obsoletos**