

"CADA SONHO QUE VOCÊ DEIXA  
PARA TRÁS, É UM PEDAÇO DO  
SEU FUTURO QUE DEIXA DE  
EXISTIR."

- STEVE JOBS -





# **ENGENHARIA DE SOFTWARE**

Requisitos de Software

*Professor: Charles Leite*

# Engenharia de requisitos

- Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento.
- O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado de engenharia de requisitos.

# O que é um requisito?

- Pode ser uma descrição abstrata de alto nível de um serviço, uma restrição de sistema ou até uma especificação matemática, entre outras coisas
- O problema cujo desenvolvimento do sistema deve resolver
  - O sistema tem que ser construído de modo a satisfazer **todos** os seus requisitos

# Tipos de requisitos

- Requisitos de usuário

- Declarações em linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar.

- Requisitos de sistema

- Um documento estruturado estabelecendo descrições detalhadas das funções, serviços e restrições operacionais do sistema de software.
- Define o que deve ser implementado e pode até ser parte de um contrato entre o cliente e o desenvolvedor.

# Exemplo: O sistema LIBSYS

- Um sistema de biblioteca que fornece uma interface única para uma série de banco de dados de artigos em bibliotecas diferentes.
- Os usuários podem pesquisar, baixar e imprimir estes artigos para estudo pessoal.

# Definições e especificações

## Definição de requisitos de usuário

1. LIBSYS deve manter o acompanhamento de todos os dados exigidos pelas agências de licenciamento de direitos autorais no Reino Unido e em outros lugares.

## Especificação dos requisitos de sistema

1.1 Ao solicitar um documento ao LIBSYS, deve ser apresentado ao solicitante um formulário que registra os detalhes do usuário e da solicitação feita.

1.2 Os formulários de solicitação do LIBSYS devem ser armazenados no sistema durante cinco anos, a partir da data da solicitação.

1.3 Todos os formulários do LIBSYS devem ser indexados por usuário, nome do material solicitado e fornecedor da solicitação.

1.4 O LIBSYS deve manter um registro de todas as solicitações feitas ao sistema.

1.5 Para materiais aos quais se aplicam os direitos de empréstimo dos autores, os detalhes do empréstimo devem ser enviados mensalmente às agências de licenciamento de direitos autorais que se registraram no LIBSYS.

# Requisitos funcionais e não funcionais

## •Requisitos funcionais

- Serviços que o sistema deve fornecer
- Como o sistema deve reagir a entradas específicas
- Como o sistema deve se comportar em determinadas situações.

## •Requisitos não funcionais

- Restrições sobre serviços ou funções oferecidos pelo sistema tais como restrições de *timing*, restrições sobre o processo de desenvolvimento, padrões.
- Muitas vezes aplicam-se ao sistema como um todo.



# Exemplos de requisitos funcionais

- O usuário deve ser capaz de pesquisar em todo o conjunto inicial de banco de dados ou selecionar um subconjunto a partir dele.
- Para todo pedido deve ser alocado um identificador único que o usuário possa copiar para a área de armazenamento permanente da sua conta.
- O sistema deve fornecer **telas apropriadas** para o usuário ler os documentos no repositório de documentos.

# Imprecisão de requisitos

- Problemas surgem quando os requisitos não são precisamente definidos.
- Requisitos ambíguos podem ser interpretados de maneiras diferentes pelos desenvolvedores e usuários.
- Considere o termo ‘telas apropriadas’
  - Intenção do usuário – tela de propósito especial para cada tipo diferente de documento;
  - Interpretação do desenvolvedor – fornece uma tela de texto que mostra o conteúdo do documento.

# Requisitos completos e consistentes

- Em princípio, requisitos devem ser completos e consistentes:
  - **Completude**
    - Eles devem incluir descrições de **todos** os recursos requeridos.
  - **Consistência**
    - Não deve haver **conflitos** ou **contradições** nas descrições dos recursos de sistema.

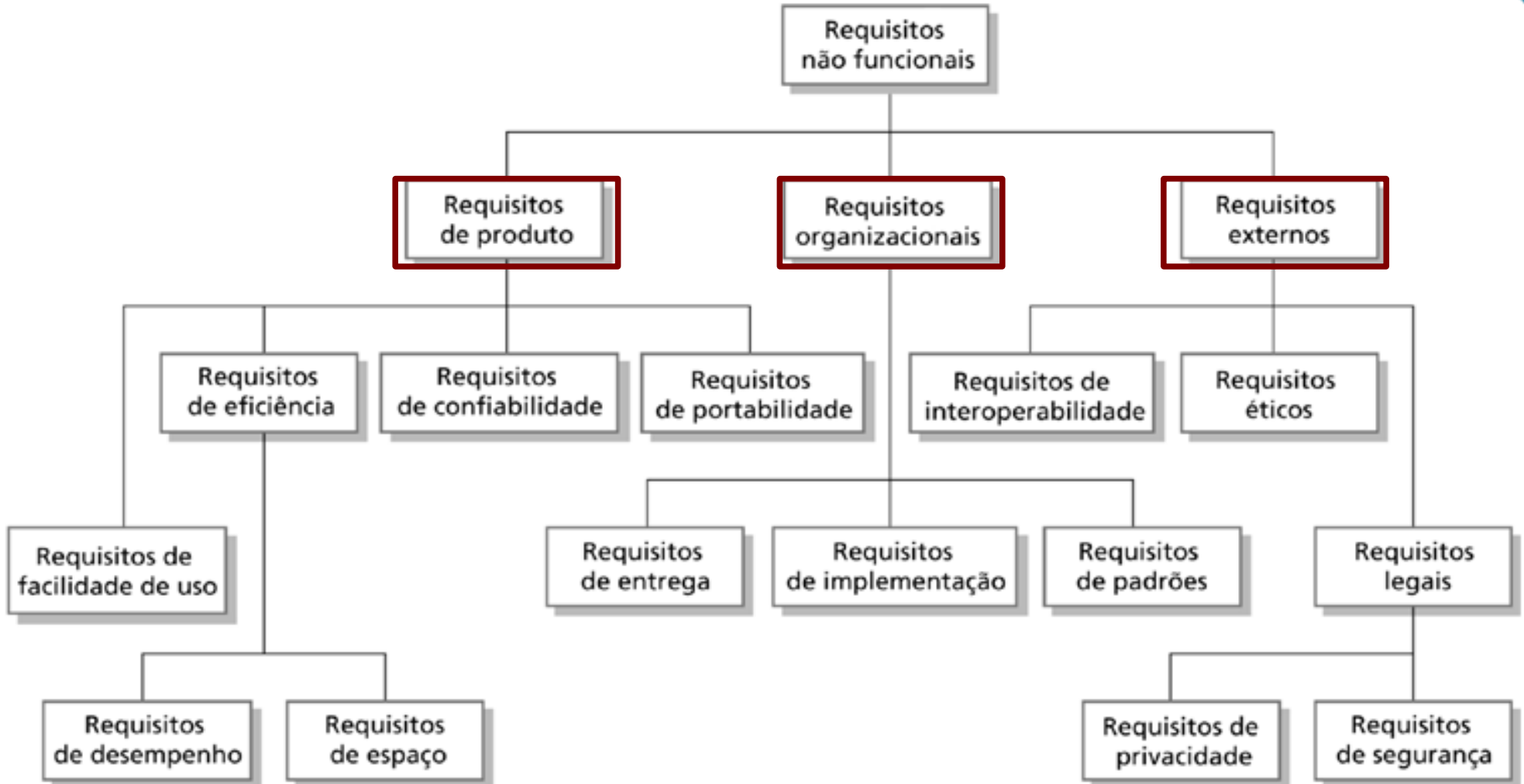
# Requisitos não funcionais

- Definem propriedades e restrições de sistema
- Exemplos incluem confiabilidade, tempo de resposta e requisitos de armazenamento.
- Requisitos de processo podem também ser especificados, impondo uma linguagem de programação, IDE ou método de desenvolvimento particular
- Requisitos não funcionais **podem ser mais críticos** do que os requisitos funcionais.

# Tipos de requisitos não funcionais

- Requisitos de produto: Especificam ou restringem o comportamento do software.
- Requisitos organizacionais: Requisitos gerais de sistemas derivados das políticas e procedimentos da organização do cliente e do desenvolvedor.
- Requisitos externos: Abrange todos os requisitos que derivam de fatores externos ao sistema e seu processo de desenvolvimento.

# Tipos de requisitos não funcionais



# Metas e requisitos

- Requisitos não funcionais podem ser difíceis de definir precisamente, sendo que requisitos imprecisos podem ser difíceis de verificar.
- Meta
  - Uma intenção geral do usuário, tal como facilidade de uso.
- Requisito não funcional verificável
  - Uma declaração usando alguma medida que pode ser objetivamente testada.
- Metas são úteis para desenvolvedores quando exprimem as intenções dos usuários do sistema.

# Exemplo

## **Meta do sistema**

O sistema deve ser fácil de ser usado pelos controladores experientes e ser organizado de modo que os erros dos usuários sejam minimizados.

## **Requisito não funcional verificável**

Os controladores experientes devem ser capazes de usar todas as funções do sistema depois de um treinamento no total de duas horas. Após esse treinamento, o número médio de erros cometidos pelos usuários experientes não deve exceder dois por dia.



# Medidas de requisitos

**Tabela 6.1** Métricas para especificar requisitos não funcionais

Propriedade	Medida
Velocidade	Transações processadas/segundo Tempo de resposta de usuário/evento Tempo de atualização da tela
Tamanho	Kbytes Número de chips de RAM
Facilidade de uso	Tempo de treinamento Número de frames de ajuda
Confiabilidade	Tempo médio de falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo para reiniciar após falha Porcentagem de eventos que causam falhas Probabilidade de corrupção de dados por falhas
Portabilidade	Porcentagem de declarações dependentes do sistema-alvo Número de sistemas-alvo

# Interação de requisitos

- Conflitos entre os diferentes requisitos não funcionais são comuns em sistemas complexos.
- Sistema de aeronave
  - Para minimizar o peso, o número de *chips* separados no sistema deve ser minimizado.
  - Para minimizar o consumo de energia, *chips* de baixa potência devem ser usados.
    - E o **desempenho** pode ser impactado!
  - Contudo, o uso de *chips* de baixa potência pode significar que mais chips devem ser usados . Qual é o requisito mais crítico?

# Requisitos de usuário

- Requisitos funcionais e não funcionais descritos de modo a ser compreensíveis por usuários que não têm conhecimento técnico detalhado.
- São definidos usando uma linguagem simples, tabelas e diagramas quando estes podem ser compreendidos por todos os usuários.
- Histórias de usuários são similares a requisitos de usuários

# Diretrizes para escrever requisitos

- Usar um formato padrão para todos os requisitos.
- Usar a linguagem de uma forma consistente.
  - ‘deve’ para requisitos obrigatórios, e
  - ‘deveria’ para requisitos desejáveis.
- **Realçar** o texto para identificar as partes principais do requisito.
- Evitar o uso de jargões de computação.

# Requisitos de sistema

- Especificações mais detalhadas das funções do sistema, dos serviços e das restrições
- Visam fornecer uma base para o desenvolvimento do sistema
  - Em XP: **histórias de usuário** + **tarefas de desenvolvimento**.
- Eles podem ser incorporados no contrato de sistema.
- Requisitos de sistema podem ser definidos ou ilustrados usando **notações gráficas**

# Requisitos e Projeto

- Requisitos devem definir o que o sistema deve fazer e o projeto deve descrever como ele faz isto.
  - Requisitos => **problema**
  - Projeto => **solução**
- Na prática, requisitos e projeto são **inseparáveis**
  - Uma arquitetura de sistema pode ser projetada para estruturar os requisitos;
  - O sistema pode ter que interoperar com outros sistemas que geram novos requisitos;
  - O uso de uma solução de projeto específica pode ser um requisito de domínio.

# Problemas com linguagem natural

- Falta de clareza
  - É difícil atingir uma precisão sem tornar o documento difícil de ler e ambíguo
- Confusão de requisitos
  - Requisitos funcionais e não-funcionais tendem a estar misturados.
- Fusão de requisitos
  - Vários requisitos diferentes podem ser expressos juntos
- Dificuldade de estruturar a especificação

# Especificações em linguagem estruturada

- A liberdade do elaborador de requisitos é limitada por um *template* pré-definido para requisitos.
- Todos os requisitos são escritos de maneira padronizada.
- A terminologia usada na descrição pode ser limitada.
- A vantagem é que a maior parte da expressividade da linguagem natural é mantida
  - Mas o grau de uniformidade é imposto na especificação.



# Especificação tabular

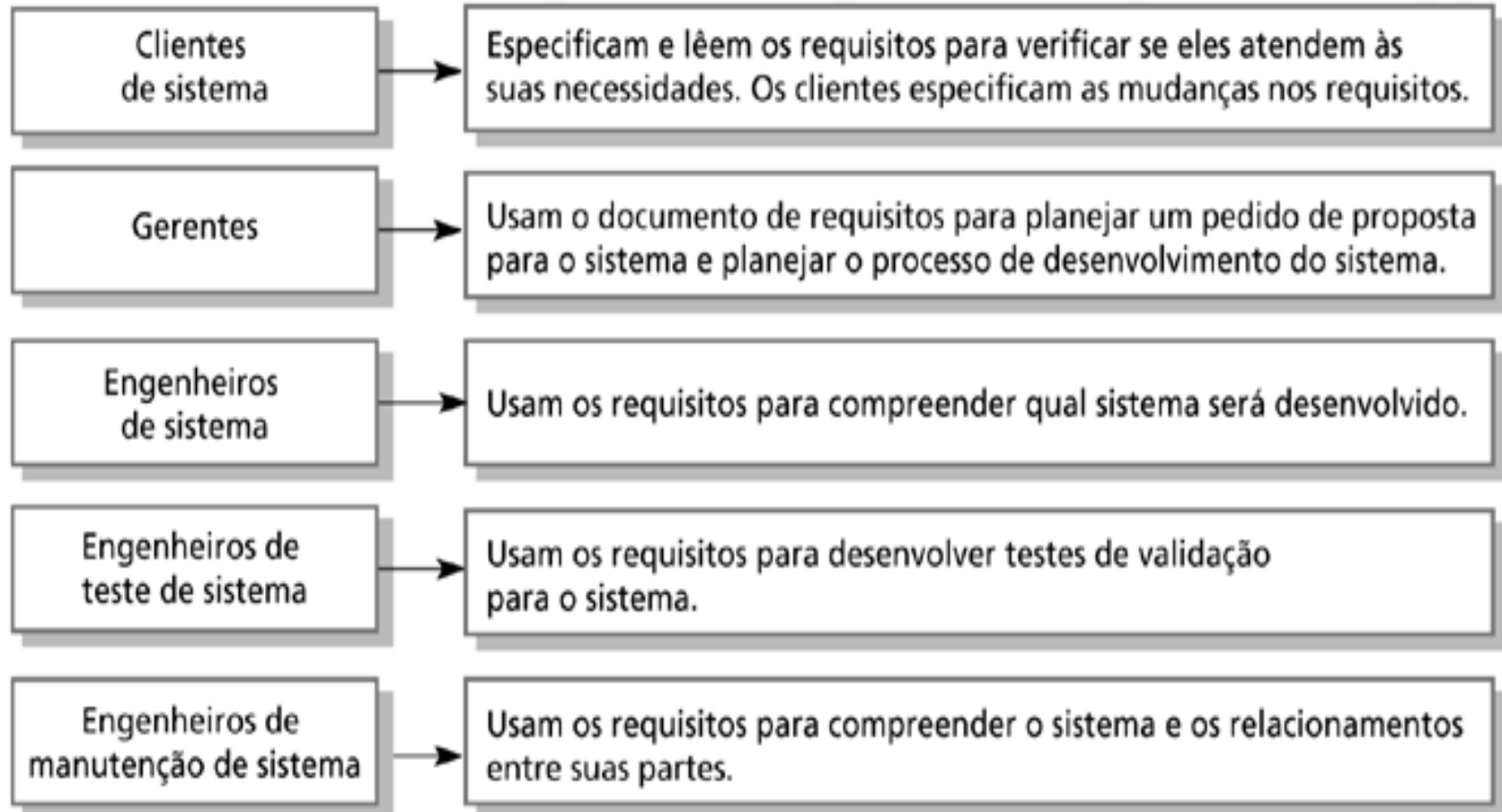
- Usada para suplementar a linguagem natural.
- Particularmente útil quando você tem de definir uma série de possíveis cursos alternativos de ação.



# O documento de requisitos

- O documento de requisitos é a declaração oficial do que é requisitado pelos desenvolvedores do sistema.
  - Em XP é um pouco **diferente**
- Deve incluir ambos, uma definição dos requisitos de usuário e uma especificação dos requisitos de sistema.
- **NÃO É** um documento de projeto.
  - Logo que possível, será preciso definir **como** o sistema deve fazer, ao invés de **o que** deve ser feito.

# Usuários de um documento de requisitos



# DÚVIDAS ...

