

---

# Requisitos de Software

---

# Objetivos

---

- Apresentar os conceitos de requisitos de usuário e de sistema
- Descrever requisitos funcionais e não funcionais
- Explicar como os requisitos de software podem ser organizados em um documento de requisitos

# Tópicos cobertos

---

- Requisitos funcionais e não funcionais
- Requisitos de usuário
- Requisitos de sistema
- Especificação de interface
- O documento de requisitos de software

# Engenharia de requisitos

---

- O processo de estabelecer os serviços que o cliente requer a partir de um sistema e as restrições sob as quais ele opera e é desenvolvido.
- Os próprios requisitos são as descrições dos serviços de sistema e das restrições que são geradas durante o processo de engenharia de requisitos.

# O que é um requisito?

---

- Pode variar de uma declaração abstrata de alto nível de um serviço ou de uma restrição de sistema para uma especificação matemática funcional.
- Isto é inevitável quando os requisitos podem servir uma função dual
  - Pode ser a base para uma proposta de um contrato – portanto deve ser aberta para interpretação;
  - Pode ser a base para o contrato em si – portanto deve ser definido em detalhe;
  - Ambas as declarações podem ser chamadas requisitos.

# Abstração de requisitos (Davis)

---

*“Se uma empresa deseja estabelecer um contrato para um projeto de desenvolvimento de software de grande porte, deve definir suas necessidades de forma suficientemente abstrata, para que uma solução não esteja pré-definida. Os requisitos devem ser escritos de tal forma que vários fornecedores possam apresentar propostas para o contrato, oferecendo, talvez, diferentes formas de atender às necessidades organizacionais do cliente. Uma vez que o contrato for aprovado, o fornecedor deve escrever uma definição de sistema para o cliente, em mais detalhes, tal que o cliente compreenda e possa validar o que o software irá fazer. Ambos os documentos podem ser chamados de documento de requisitos do sistema.”*

# Tipos de requisitos

- **Requisitos de usuário**
  - Declarações em linguagem natural mais diagramas de serviços que o sistema fornece e suas restrições operacionais. Escritos para os clientes.
- **Requisitos de sistema**
  - Um documento estruturado estabelecendo descrições detalhadas das funções, serviços e restrições operacionais do sistema. Define o que deve ser implementado e assim, pode ser parte de um contrato entre o cliente e o desenvolvedor.

# Definições e especificações

## Quadro 6.1

Requisitos de usuário e de sistema.



### 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.

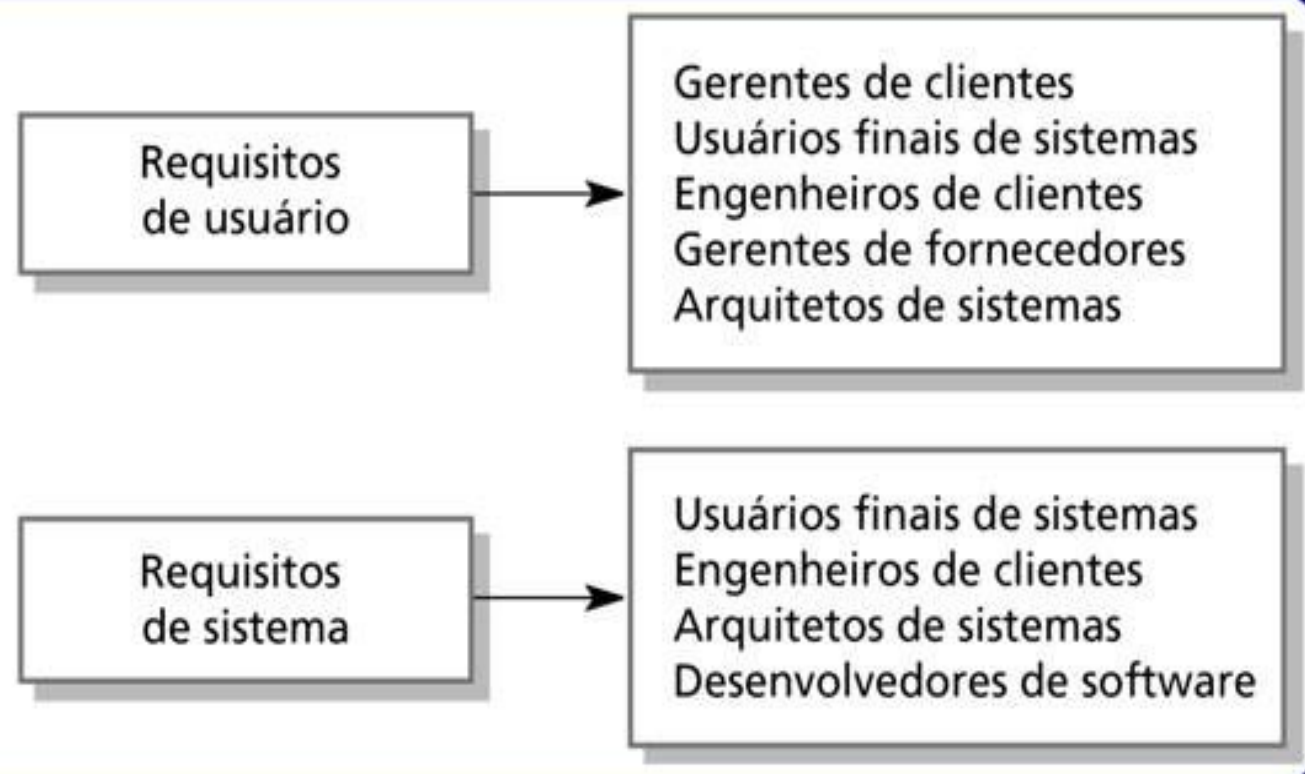
1.5 Para materiais aos quais se aplicam os direitos de empréstimo dos autores, os detalhes do empréstimo devem ser enviados mensalmente às agências de licenciamento de direitos autorais que se registraram no LIBSYS.



# Leitores de requisitos

**Figura 6.1**

Leitores de diferentes tipos de especificação.



# Requisitos funcionais e não funcionais

---

- Requisitos funcionais
  - Declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e 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, etc.
- Requisitos de domínio
  - Requisitos que vêm do domínio de aplicação do sistema e que refletem as características desse domínio.

# Requisitos funcionais

---

- Descrevem a funcionalidade ou serviços de sistema.
- Dependem do tipo de software, dos usuários esperados e o tipo de sistema onde o software é usado.
- Requisitos funcionais de usuário podem ser declarações de alto nível do que o sistema deve fazer mas os requisitos funcionais de sistema devem descrever os serviços de sistema em detalhe.

# 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.

# 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.
- O sistema deve fornecer telas apropriadas para o usuário ler os documentos no repositório de documentos.
- Para todo pedido deve ser alocado um identificador único (ORDER\_ID) no qual o usuário deve ser capaz de copiar para a área de armazenamento permanente da sua conta.

# 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 ambos, completos e consistentes.
- Completeza
  - Eles devem incluir descrições de todos os recursos requeridos.
- Consistência
  - Não deve haver conflitos ou contradições nas descrições dos recursos de sistema.
- Na prática, para sistemas grandes e complexos, é impossível produzir um documento de requisitos completo e consistente.

# Requisitos não funcionais

---

- Estes definem propriedades e restrições de sistema, por exemplo, confiabilidade, tempo de resposta e requisitos de armazenamento. Restrições são capacidade de dispositivos de E/S, representações de sistema, etc.
- Requisitos de processo podem também ser especificados impondo um sistema CASE particular, linguagem de programação ou método de desenvolvimento.
- Requisitos não funcionais podem ser mais críticos do que os requisitos funcionais. Se estes não forem atendidos, o sistema é inútil.



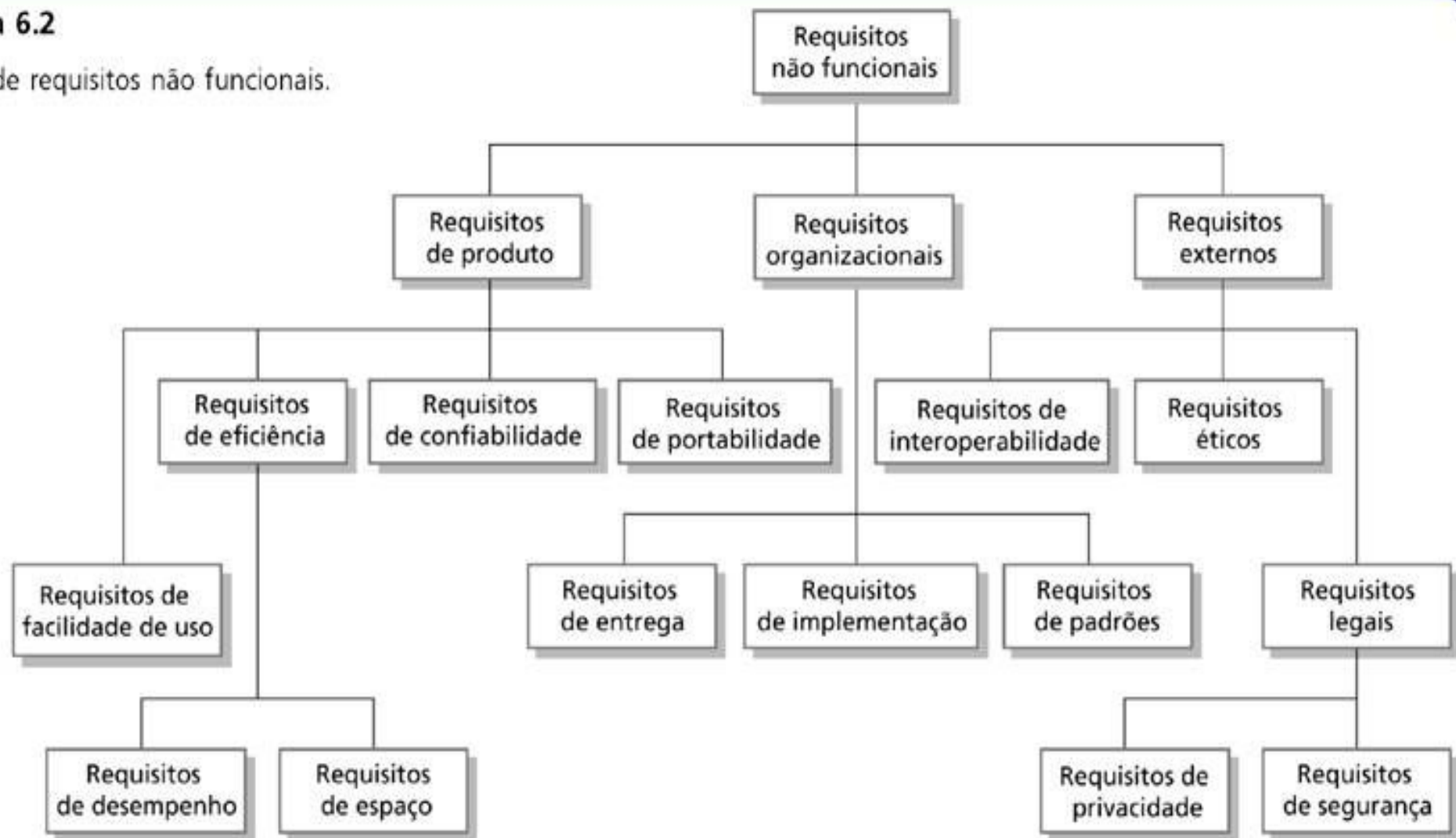
# Classificações de requisitos não funcionais

- Requisitos de produto
  - Requisitos que especificam que o produto entregue deve se comportar de uma maneira particular, por exemplo, velocidade de execução, confiabilidade, etc.
- Requisitos organizacionais
  - Requisitos que são uma consequência de políticas e procedimentos da organização, por exemplo, padrões de processo usados, requisitos de implementação, etc.
- Requisitos externos
  - Requisitos que surgem a partir de fatores externos ao sistema e seu processo de desenvolvimento, por exemplo, requisitos de interoperabilidade, requisitos legais, etc.

# Tipos de requisitos não funcionais

Figura 6.2

Tipos de requisitos não funcionais.



# Exemplos de requisitos não funcionais

## Quadro 6.2

Exemplos de requisitos não funcionais.



### Requisito de produto

8.1 A interface de usuário para o LIBSYS deve ser implementada como simples HTML, sem frames ou applets de Java.

### Requisito organizacional

9.3.2 O processo de desenvolvimento do sistema e os documentos a serem entregues devem estar em conformidade com o processo e produtos a serem entregues definidos em XYZCo-SP-STAN-95.

### Requisito externo

10.6 O sistema não deve revelar quaisquer informações pessoais sobre os usuários do sistema ao pessoal da biblioteca que usa o sistema, com exceção do nome e número de referência da biblioteca.

# Metas e requisitos

- Requisitos não funcionais podem ser muito difíceis de definir precisamente e 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.

# Exemplos

## Quadro 6.3

Metas do sistema e requisitos verificáveis

### 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 Spacecraft system
  - 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.
  - Contudo, o uso de *chips* de baixa potência pode significar que mais chips devem ser usados . Qual é o requisito mais crítico?

# Requisitos de domínio

---

- Derivados do domínio de aplicação e descrevem características de sistema que refletem o domínio.
- Podem restringir os requisitos funcionais existentes ou estabelecer como cálculos específicos devem ser realizados.
- Se os requisitos de domínio não forem satisfeitos, o sistema pode não funcionar.



# Requisitos de domínio do sistema de bibliotecas

---

- Deve existir uma interface de usuário padrão para todos os bancos de dados que será baseada no padrão Z39.50.
- Devido às restrições de direitos autorais, alguns documentos devem ser excluídos imediatamente na chegada. Dependendo dos requisitos de usuário, esses documentos serão impressos localmente no servidor de sistema para serem encaminhados manualmente para o usuário ou direcionados para uma impressora de rede.

# Sistema de proteção de trem

## Quadro 6.4

Requisito de domínio de um sistema de proteção de trens.

A desaceleração do trem deve ser calculada como:

$$D_{\text{trem}} = D_{\text{controle}} + D_{\text{gradiente}}$$

onde  $D_{\text{gradiente}}$  é  $9,81 \text{ ms}^2 * \text{gradiente compensado/alfa}$  e onde os valores de  $9,81 \text{ ms}^2/\text{alfa}$  são conhecidos para diferentes tipos de trens.

# Problemas de requisitos de domínio

---

- **Facilidade de entendimento**
  - Requisitos são expressos na linguagem do domínio de aplicação;
  - Isso não é, freqüentemente, compreendido pelos engenheiros de software que estão desenvolvendo o sistema.
- **Implícito**
  - Especialistas em domínio compreendem a área tão bem que não pensam em tornar os requisitos de domínio explícitos.

# Requisitos de usuário

---

- Deve descrever requisitos funcionais e não funcionais, de tal modo que sejam compreensíveis pelos usuários de sistema que não têm conhecimento técnico detalhado.
- Requisitos de usuário são definidos usando uma linguagem simples, tabelas e diagramas quando estes podem ser compreendidos por todos os usuários.

# Problemas com linguagem natural

---

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

# Requisito do LIBSYS

## Quadro 6.5

Requisito de usuário para um sistema de contabilidade no LIBSYS.

4.5 O LIBSYS deve fornecer um sistema de contabilidade financeira que mantenha registros de todos os pagamentos realizados pelos usuários do sistema. Os gerentes podem configurar esse sistema de modo que os usuários frequentes possam receber descontos.

# Requisito de grade de editor

## Quadro 6.6

Requisito de usuário para uma grade de editor.

**2.6 Recursos de grade** Para ajudar no posicionamento de entidades sobre um diagrama, o usuário pode ativar uma grade em centímetros ou em polegadas por meio de uma opção no painel de controle. Inicialmente, a grade não estará ativada. A grade poderá ser ativada ou desativada a qualquer instante durante uma sessão de edição, e poderá ser alternada entre polegadas e centímetros, a qualquer instante. Uma opção de grade será fornecida na visão reduzida, mas o número de linhas de grade mostradas será reduzido para evitar o preenchimento do diagrama

# Problemas de requisitos

- Requisitos de banco de dados incluem tanto informação conceitual quanto informação detalhada
  - Descrevem o conceito de um sistema de contabilidade financeira que deve ser incluído no LIBSYS;
  - Contudo, também inclui o detalhe com os quais os gerentes podem configurar esse sistema – isso é desnecessário neste nível.
- O requisito de grade mistura três tipos diferentes de requisitos
  - Requisito funcional conceitual (a necessidade de uma grade);
  - Requisito não funcional (unidades de grade);
  - Requisito não funcional de UI (chaveamento de grade).



# Apresentação estruturada

## Quadro 6.7

Definição de um recurso de grade de editor.

### 2.6.1 Recursos de grade

O editor deve fornecer um recurso de grade no qual uma matriz de linhas horizontais e verticais fornece um fundo para a janela do editor. Essa grade deve ser passiva e o alinhamento das entidades é de responsabilidade do usuário.

*Justificativa lógica:* Uma grade ajuda o usuário a criar um diagrama bem organizado com entidades bem espaçadas. Embora uma grade ativa, na qual as entidades 'saltam' as linhas de grade, possa ser útil, o posicionamento é impreciso. O usuário é a melhor pessoa para decidir onde as entidades devem ser posicionadas.

*Especificação:* ECLIPSE/WS/Tools/DE/FS Seção 5.6

*Fonte:* Ray Wilson, escritório de Glasgow

# Diretrizes para escrever requisitos

---

- Inventar um formato padrão e usá-lo para todos os requisitos.
- Usar a linguagem de uma forma consistente. Use 'deve' para requisitos obrigatórios, e 'deveria' para requisitos desejáveis.
- Realce o texto para identificar as partes principais do requisito.
- Evitar o uso de jargões de computação.

# Requisitos de sistema

---

- Mais especificações detalhadas das funções do sistema, dos serviços e das restrições do que requisitos de usuário.
- Eles pretendem ser uma base para o desenvolvimento do projeto de sistema.
- Eles podem ser incorporados no contrato de sistema.
- Requisitos de sistema podem ser definidos ou ilustrados usando modelos de sistema discutidos no Capítulo 8.

# Requisitos e projeto

---

- Em princípio, requisitos devem definir o que o sistema deve fazer e o projeto deve descrever como ele faz isto.
- Na prática, requisitos e projeto são inseparáveis
  - Uma arquitetura de sistema pode ser projetada para estruturar os requisitos;
  - O sistema pode interoperar com outros sistemas que geram requisitos de projeto;
  - O uso de um projeto específico pode ser um requisito de domínio.

# Problemas com especificação em linguagem natural

- Ambigüidade
  - Os leitores e os escritores dos requisitos devem interpretar as mesmas palavras da mesma maneira. Linguagem natural é naturalmente ambígua , por isso, muito difícil.
- Flexibilidade excessiva
  - A mesma coisa pode ser dita de várias maneiras diferentes na especificação.
- Falta de modularização
  - Estruturas de linguagem natural são inadequadas para estruturar requisitos de sistema.

# Alternativas para especificação em linguagem natural

**Tabela 6.2** Notações para especificação de requisitos

Notação	Descrição
Linguagem natural estruturada	Esta abordagem depende da definição de formulários ou templates-padrão para expressar a especificação de requisitos.
Linguagens de descrição de projeto	Esta abordagem usa uma linguagem semelhante à linguagem de programação, porém com mais características abstratas, para especificar os requisitos por meio da definição de um modelo operacional do sistema. Essa abordagem não é amplamente usada hoje em dia, embora possa ser útil para especificações de interfaces.
Notações gráficas	Uma linguagem gráfica, complementada com anotações de texto é usada para definir os requisitos funcionais do sistema. Um antigo exemplo dessa linguagem gráfica é SADT (Ross, 1977) (Schoman e Ross, 1977). Atualmente, as descrições de casos de uso (Jacobsen, et al., 1993) e os diagramas de seqüência são comumente usados (Stevens e Pooley, 1999).
Especificações matemáticas	São notações baseadas em conceitos matemáticos, como máquinas de estados finitos ou conjuntos. Essas especificações não ambíguas reduzem discussões entre cliente e fornecedor. No entanto, a maioria dos clientes não compreende as especificações formais e são relutantes em aceitá-las no momento da contratação.

# 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.

# Especificações baseadas em formulários

---

- Descrição da função ou da entidade.
- Descrição das suas entradas e de onde elas se originam.
- Descrição das suas saídas e para onde elas vão.
- Indicações das outras entidades requisitadas.
- Pré e pós-condições (se apropriados)
- Efeitos colaterais da operação (se existirem).



# Especificação baseada em formulário

**Tabela 6.3** Especificação de requisitos do sistema com utilização de um formulário-padrão

## Bomba de insulina/Software de controle/SRS/3.3.2

<b>Função</b>	Calcular dose de insulina: nível seguro de açúcar
<b>Descrição</b>	Calcula a dose de insulina a ser liberada quando o nível medido de açúcar atual está na zona segura entre 3 e 7 unidades
<b>Entradas</b>	Leitura atual de açúcar ( $r_2$ ), as duas leituras anteriores ( $r_0$ e $r_1$ )
<b>Origem</b>	Leitura atual de açúcar do sensor. Outras leituras da memória
<b>Saídas</b>	CompDose — a dose de insulina a ser liberada
<b>Destino</b>	Loop de controle principal
<b>Ação:</b>	CompDose será zero se o nível de açúcar estiver estável ou em queda, ou se o nível estiver aumentando, mas a taxa de aumento estiver diminuindo. Se o nível estiver aumentando e a taxa de aumento estiver aumentando, então CompDose será calculado dividindo-se a diferença entre o nível atual de açúcar e o nível anterior por 4, e arredondando o resultado. Se o resultado do arredondamento for zero, então CompDose será definido como a dose mínima que pode ser liberada.
<b>Requer</b>	Duas leituras anteriores de modo que a taxa de mudança do nível de açúcar possa ser calculada.
<b>Precondição</b>	O reservatório de insulina conter, pelo menos, o máximo de dose única permitida de insulina.
<b>Pós-condição</b>	$r_0$ é substituído por $r_1$ , portanto $r_1$ é substituído por $r_2$
<b>Efeitos colaterais</b>	Nenhum

# Especificação tabular

---

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

# Especificação tabular

**Tabela 6.4** Especificação tabular de cálculo



Condição	Ação
Nível de açúcar em queda ( $r_2 < r_1$ )	CompDose = 0
Nível de açúcar estável ( $r_2 = r_1$ )	CompDose = 0
Nível de açúcar aumentando e taxa de aumento diminuindo ( $(r_2 - r_1) < (r_1 - r_0)$ )	CompDose = 0
Nível de açúcar aumentando e taxa de aumento estável ou aumentando ( $(r_2 - r_1) > (r_1 - r_0)$ )	CompDose = arredondar $((r_2 - r_1)/4)$ Se o resultado do arredondamento for = 0 então CompDose = DoseMinima

# Modelos gráficos

---

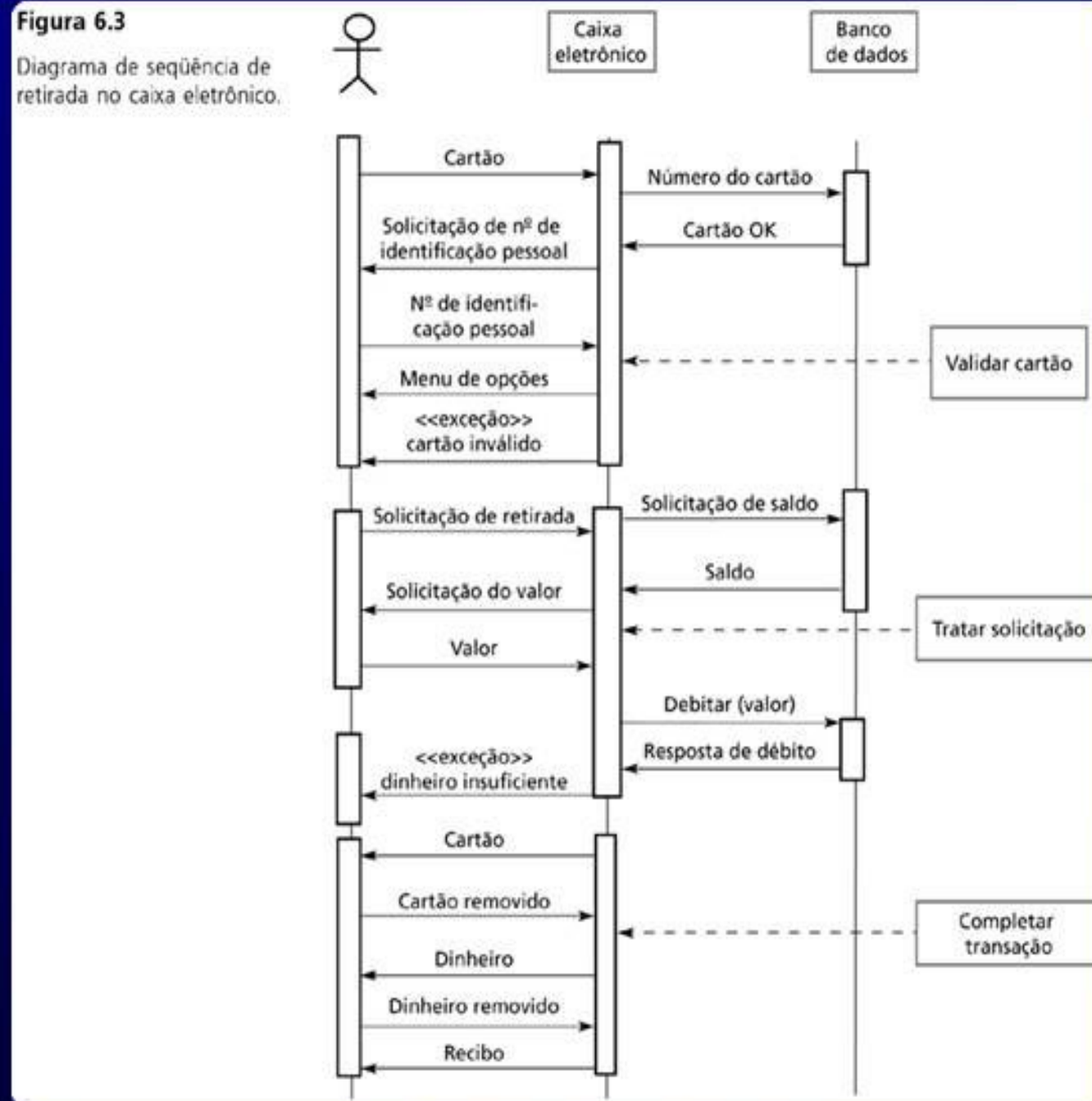
- Modelos gráficos são mais úteis quando você necessita mostrar como o estado muda ou onde você necessita descrever uma seqüência de ações.
- Os diferentes modelos gráficos são explicados no capítulo 8.

# Diagramas de seqüência

---

- Mostram a seqüência de eventos que ocorrem durante alguma interação entre usuário e sistema.
- Você lê os eventos de cima para baixo para ver a ordem em que as ações ocorrem.
- Retirada de dinheiro de um caixa eletrônico
  - Validar cartão;
  - Tratar solicitação;
  - Completar transação.

# Diagrama de seqüência de retirada de Caixa eletrônico



© 2007 by Pearson Education

# Especificação de interface

- A maioria dos sistemas devem operar com outros sistemas, e as interfaces que operam devem ser especificadas como parte dos requisitos.
- Três tipos de interface podem ser definidos:
  - Interfaces de procedimentos;
  - Estruturas de dados que são trocadas;
  - Representações de dados.
- Notações formais são uma técnica efetiva para especificação da interface.

# Descrição de interface em PDL

## Figura 6.4

Descrição PDL em Java da interface do servidor de impressão.

```
interface PrintServer {  
  
    // define um servidor abstrato de impressora  
    // requer: interface Printer, interface PrintDoc  
    // fornece: inicia, imprime, apresentaPrintQueue, cancelaPrintJob, trocaPrinter  
  
    void inicia ( Printer p ) ;  
    void imprime ( Printer p, PrintDoc d ) ;  
    void apresentaPrintQueue ( Printer p ) ;  
    void cancelaPrintJob ( Printer p, PrintDoc d ) ;  
    void transferePrinter ( Printer p1, Printer p2, PrintDoc d ) ;  
} //PrintServer
```



# O documento de requisitos

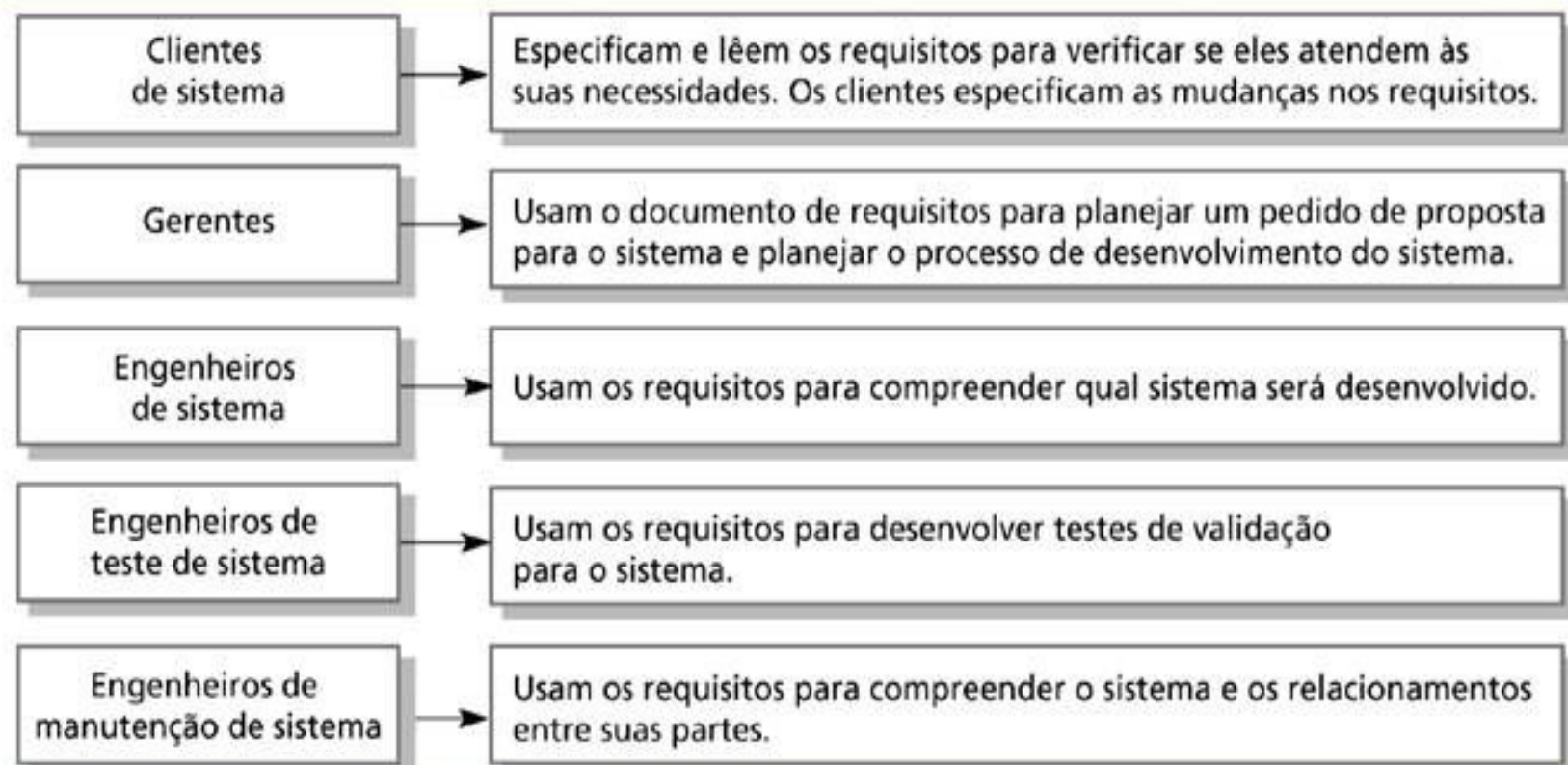
---

- O documento de requisitos é a declaração oficial do que é requisitado pelos desenvolvedores do sistema.
- 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 **O QUÊ** o sistema deve fazer ao invés de **COMO** deve ser feito.

# Usuários de um documento de requisitos

**Figura 6.5**

Usuários de um documento de requisitos.



# Padrão de requisitos do IEEE

---

- Define uma estrutura genérica para um documento de requisitos que deve ser instanciado para cada sistema específico.
  - Introdução.
  - Descrição geral.
  - Requisitos específicos.
  - Apêndices.
  - Índice.

# Estrutura de documento de requisitos

---

- Prefácio
- Introdução
- Glossário
- Definição de requisitos de usuário
- Arquitetura de sistema
- Especificação de requisitos de sistema
- Modelos de sistema
- Evolução de sistema
- Apêndices
- Índice

# Pontos-chave

---

- Os requisitos estabelecem o que sistema deve fazer e definem as restrições sobre suas operações e sua implementação.
- Requisitos funcionais definem os serviços que o sistema deve fornecer.
- Requisitos não funcionais restringem o sistema que está sendo desenvolvido ou o processo de desenvolvimento.
- Requisitos de usuário são declarações de alto nível sobre o que o sistema deve fazer. Esses requisitos devem ser escritos usando linguagem natural, tabelas e diagramas.

# Pontos-chave

---

- Requisitos de sistema se destinam a comunicar quais as funções que o sistema deve fornecer.
- Um documento de requisitos de software é uma declaração acordada dos requisitos de sistema.
- O padrão IEEE é um ponto de partida útil para definição de padrões de requisitos mais detalhados e específicos.