

SOFTWARE REQUIREMENTS

Ian Sommerville, 8^o edição – Capítulo 6

Aula de Luiz Eduardo Guarino de Vasconcelos

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 abordados

- 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 + **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

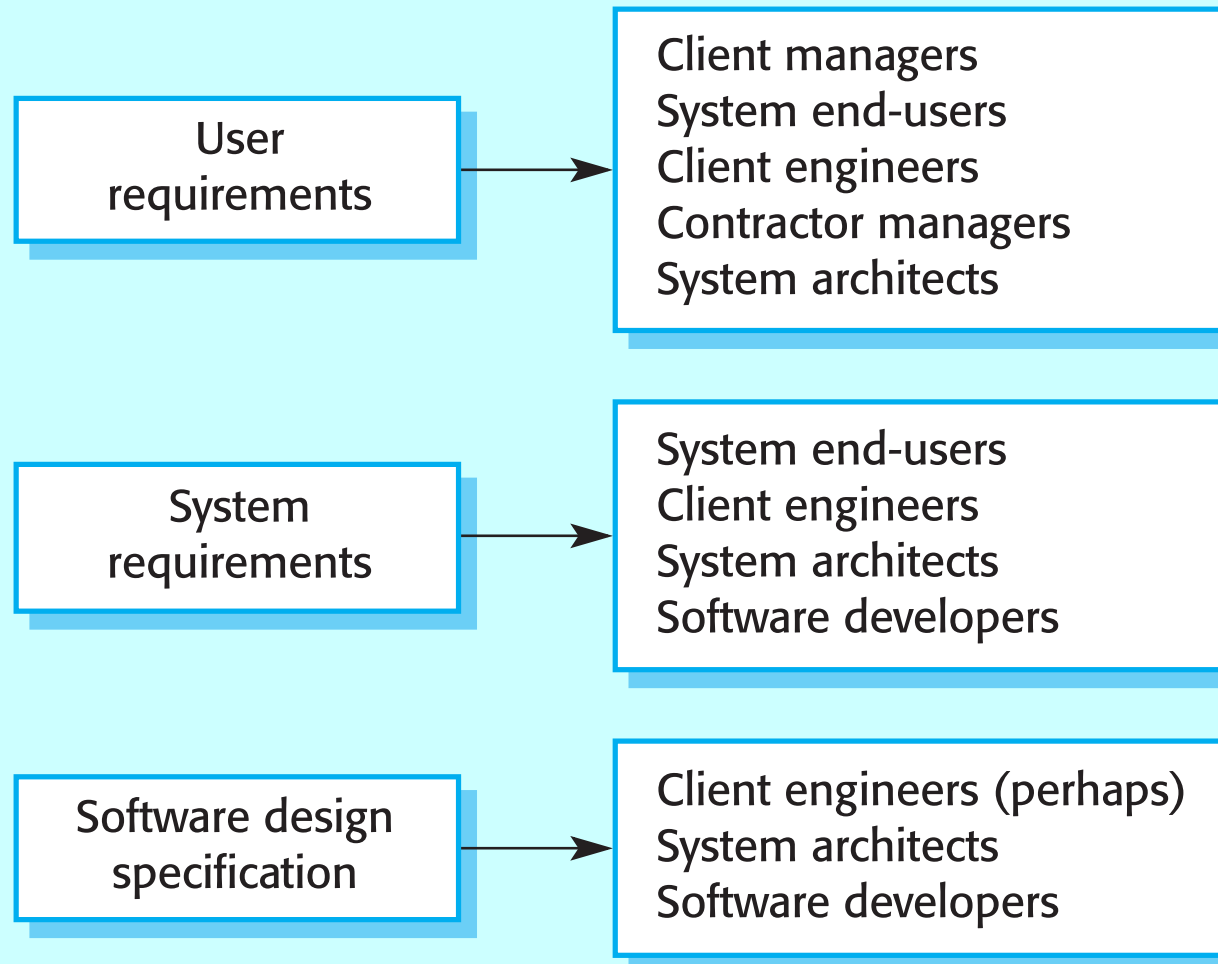
Definição dos requisitos do usuário

1. O software deve oferecer um meio de representar e acessar arquivos externos criados por outras ferramentas.

Especificação dos requisitos de sistema

- 1.1 O usuário deve dispor de recursos para definir o tipo dos arquivos externos.
- 1.2 Cada tipo de arquivo externo pode ter uma ferramenta associada que pode ser aplicada a ele.
- 1.3 Cada tipo de arquivo externo pode ser representado como um ícone específico na tela do usuário.
- 1.4 Devem ser fornecidos recursos para o ícone que representa um arquivo externo, a ser definido pelo usuário.
- 1.5 Quando um usuário seleciona um ícone que representa um arquivo externo, o efeito dessa seleção é aplicar a ferramenta associada com o tipo de arquivo externo ao arquivo representado pelo ícone selecionado

Requirements readers



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, ou seja, fixam **restrições sobre como os requisitos funcionais serão implementados**
- **Em muitos casos não existe distinção entre ambos**
- 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.
- Tem efeito **localizado**, apenas onde as funcionalidades definidas pelo requisito foram implementadas

Exemplo 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

- Requisitos devem ser 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, é impossível produzir um documento de requisitos completo e consistente.

Requisitos não-funcionais

- 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.
- Tem um efeito **global**. A satisfação deste requisito **afeta vários componentes**.
- Raramente são considerados no processo de desenvolvimento

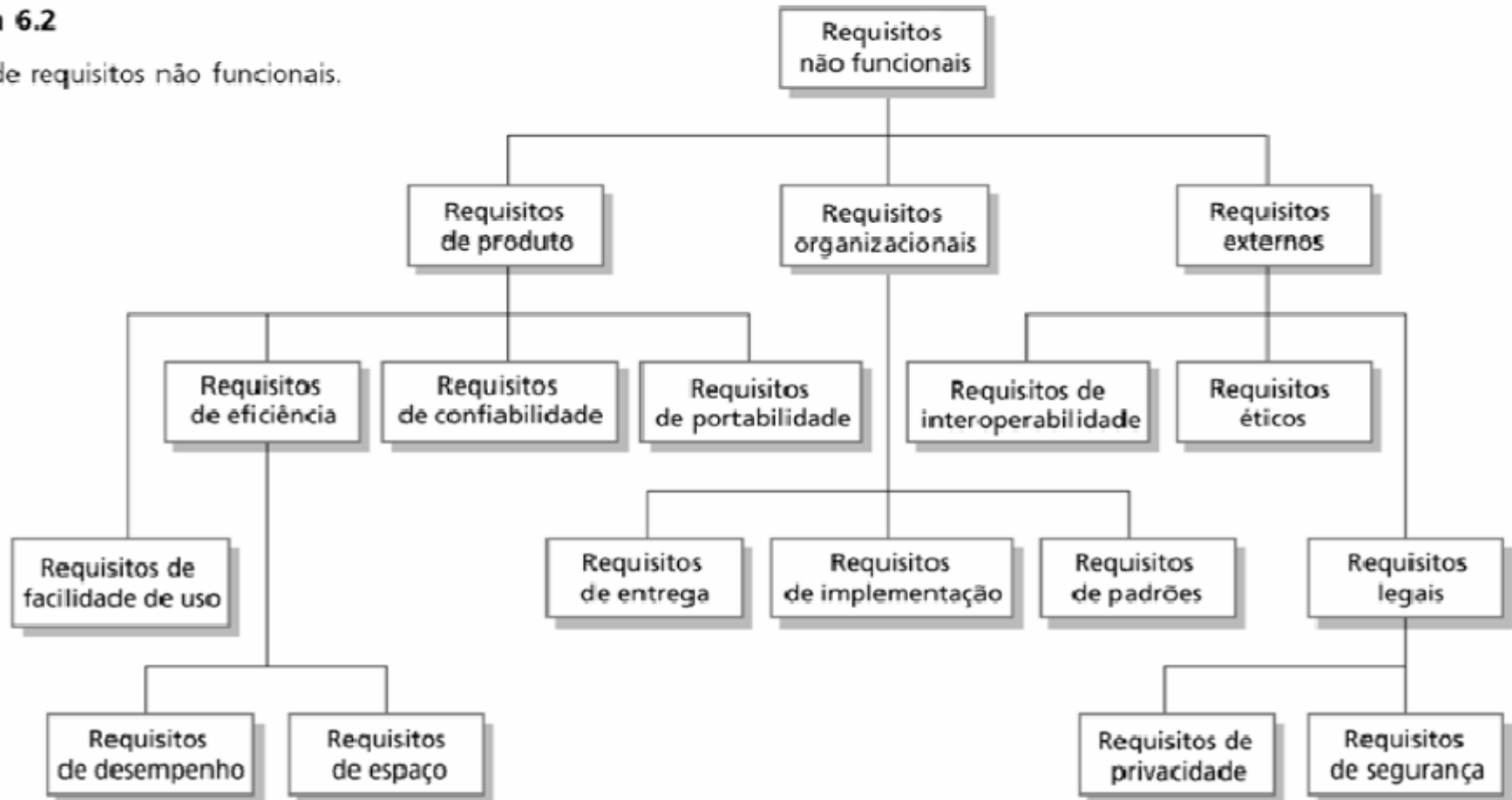
Classificação de requisitos não-funcionais

- Requisitos de produto
 - ▣ especificam que o produto entregue deve se comportar de uma maneira particular
 - ▣ por exemplo, velocidade de execução, confiabilidade
- Requisitos organizacionais
 - ▣ consequência de políticas e procedimentos da organização
 - ▣ por exemplo, padrões de processo usados, requisitos de implementação
- Requisitos externos
 - ▣ 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

□ Requisito de produto

8.1 A interface do usuário para o software deve ser implementada como simples HTML, sem frames ou applets 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

7.6.5 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

Métricas para requisitos (não funcionais)

Propriedade	Métrica
Velocidade	Transações processadas por segundo Tempo de resposta ao usuário/evento Tempo de refresh da tela
Tamanho	K bytes Número de chips de memória
Facilidade de uso	Tempo de treinamento Número de frames de ajuda
Confiabilidade	Tempo médio para falhar Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício depois de uma falha Porcentagem de eventos que causam falha Probabilidade de que dados sejam corrompidos por falhas
Portabilidade	Porcentagem de declarações dependentes de 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.

Sistema para trem

- A desaceleração do trem deve ser computada através da fórmula
 - A desaceleração do trem é dada por:
 - $D(\text{trem}) = D(\text{controle}) + D(\text{gradiente})$
 - Onde: $D(\text{gradiente}) = 9.81 \text{ ms}^2 * \text{gradiente compensado/alfa}$ e onde os valores de $9,81 \text{ ms}^2/\text{alfa}$ são conhecidos para diferentes tipos de trens
- Uso de linguagem específica do domínio do sistema
- Geralmente o engenheiro de sistema não possui domínio sobre o assunto
- Por ser muito óbvio para o usuário, ele pode não passar informações completas para o engenheiro de sistema

Problemas de requisitos de domínio

- Facilidade de entendimento
 - ▣ Requisitos são expressos na linguagem do domínio de aplicação;
 - ▣ Isso não é frequentemente 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.

Requisitos para grade de editor

Recursos de grade: para ajudar no posicionamento de entidades em um diagrama, o usuário pode acionar uma grade em centímetros ou em polegadas, por meio de um opção no painel de controle. Inicialmente, a grade está desativada. Ela pode ser ligada e desligada a qualquer momento durante uma sessão de edição e pode ser alterada entre polegadas e centímetros a qualquer momento. Uma opção será fornecida na visão reduzida do diagrama, mas o número de linhas da grade mostrado diminuirá, para evitar preencher o diagrama menor com linhas de grade

Problemas de requisitos

- Requisitos de banco de dados incluem tanto informação conceitual quanto informação detalhada
 - ▣ Descrevem o conceito de um sistema de contabilidade financeira;
 - ▣ Contudo, também inclui o detalhe com os quais os gerentes podem configurar esse sistema isso é desnecessário neste nível.
- O requisito de grade mistura 3 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).

Diretrizes para escrever requisitos

- Definir 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.

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

- Ambiguidade
 - ▣ 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.

Notações para especificação de requisitos

Notation	Description
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 especificação de interfaces.
Notações gráficas	Uma linguagem gráfica, complementada por anotações de texto é usada para definir os requisitos funcionais do sistema. Casos de uso e diagramas são muito usados hoje em dia
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 Natural

- 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.
- Vantagem: 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

Função: adicionar nós

Descrição: adiciona um nó em um desenho existente. O usuário seleciona o tipo de nó e seu posicionamento. Quando adicionado ao desenho, o nó se torna a seleção atual. O usuário escolhe a posição do nó movimentando o cursor para a área em que o nó será adicionado

Entradas: tipo de nó, posição do nó, identificador do desenho

Origem: tipo de nó e posição do nó são entradas fornecidas pelo usuário; identificador de desenho se origina da base de dados

Saídas: identificador do desenho

Destino: o banco de dados do desenho. O desenho é designado para a base de dados, no término da operação

Requer: gráfico de desenho associado ao identificador de desenho de entrada

Pré-condição: o desenho é aberto e exibido na tabela do usuário

Pós-condição: o desenho é imutável, a não ser pela adição de um nó do tipo especificado em dada posição

Efeitos colaterais: nenhum

Especificação tabular

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

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 = DoseMínima

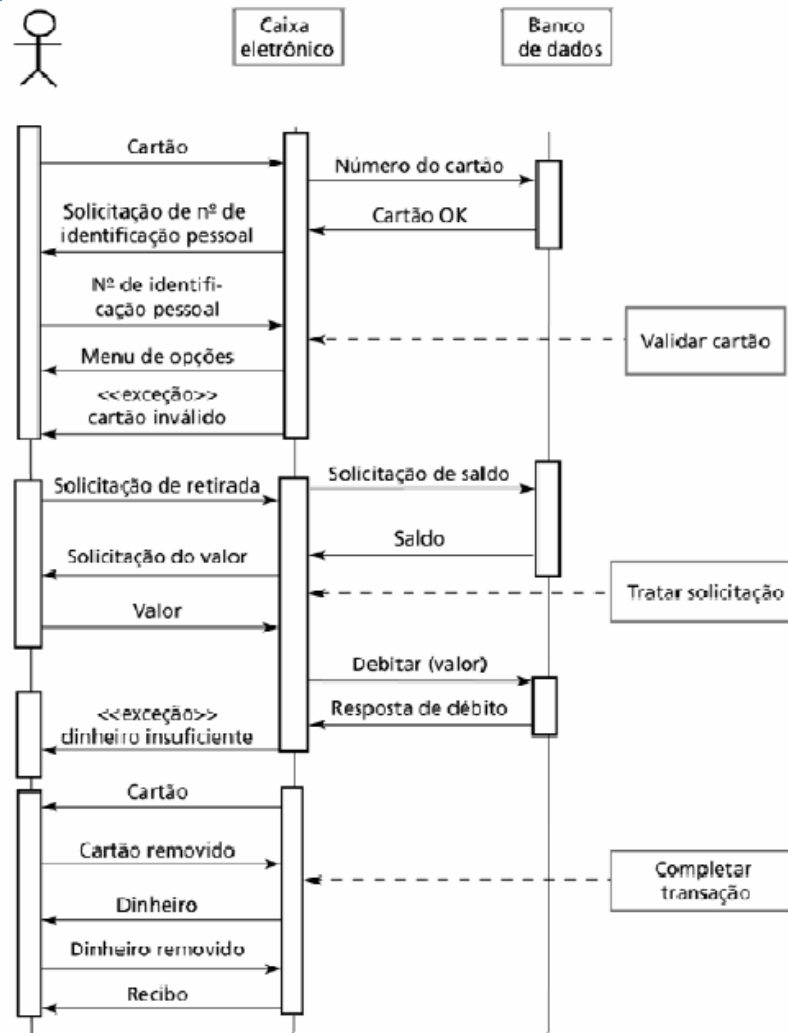
Modelos gráficos

- Modelos gráficos são mais úteis quando você necessita mostrar como o estado muda ou onde você necessita descrever uma sequência de ações.

Diagrama de sequência

- Mostram a sequência 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 sequência de retirada de Caixa eletrônico



Especificação de Interface

- A maioria dos sistemas devem operar com outros sistemas, e as interfaces que operam devem ser especificadas como parte dos requisitos.
- 3 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.

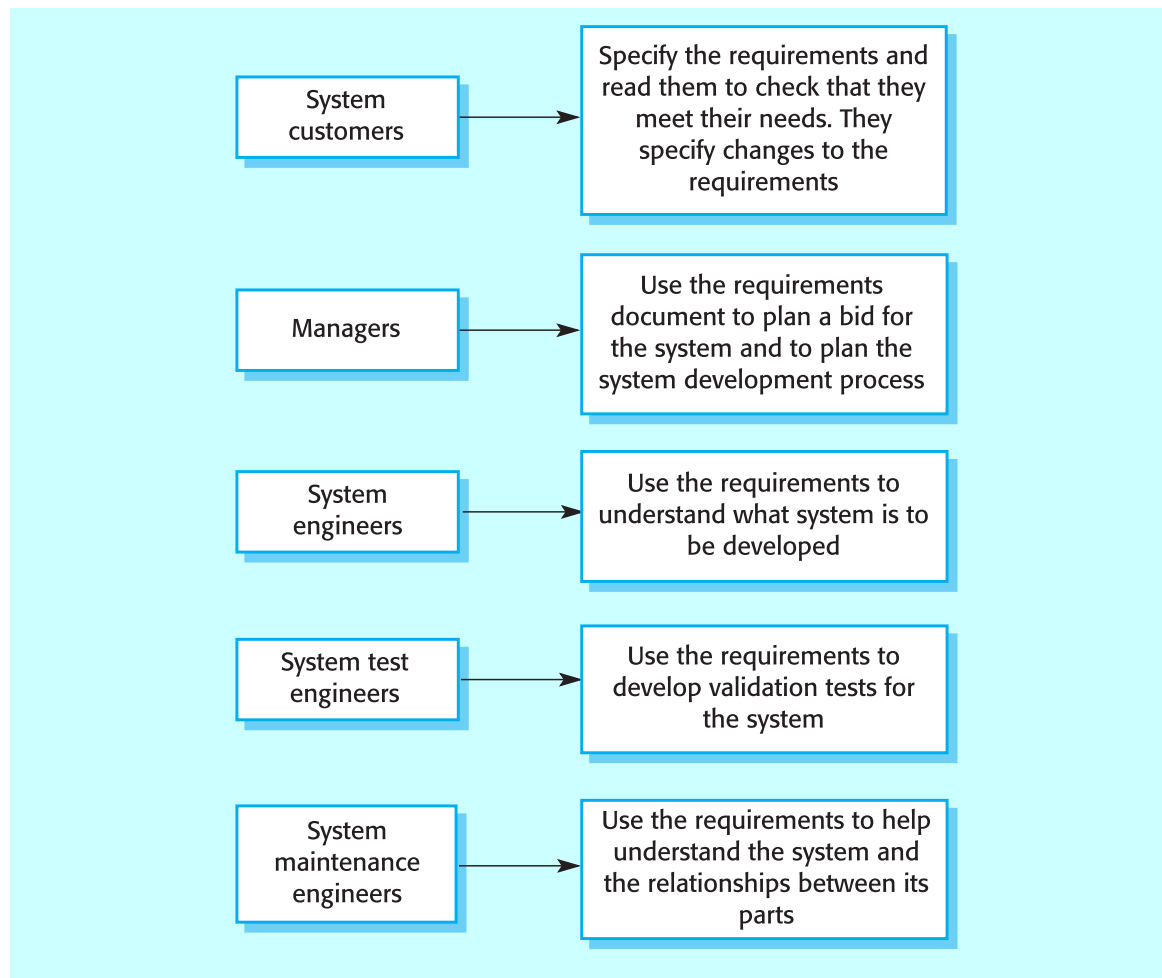
PDL interface description

```
interface PrintServer {  
  
    // defines an abstract printer server  
    // requires:      interface Printer, interface PrintDoc  
    // provides: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter  
  
        void initialize ( Printer p ) ;  
        void print ( Printer p, PrintDoc d ) ;  
        void displayPrintQueue ( Printer p ) ;  
        void cancelPrintJob (Printer p, PrintDoc d) ;  
        void switchPrinter (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 definição dos requisitos de usuário e 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 do documento de requisitos



IEEE requirements standard

- Documento de requisitos de software
- Baseado no padrão IEEE 830-1998
 - ▣ Composto de 5 partes
 - Introdução (Escopo do produto, Definições acrônimas e abreviações, Referências, Visão geral do restante do documento)
 - Descrição geral (Perspectiva do produto, Funções do produto, Características do usuário, Restrições gerais, Suposições, condições e dependências)
 - Requisitos específicos (Interface externa, Funções, Desempenho, Projeto)
 - 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.