

## Requisitos de software

---

- Descrições e especificações de um sistema

## Objetivos

---

- Apresentar os conceitos de requisitos de usuário e de sistema
- Descrever requisitos funcionais e não-funcionais
- Explicar duas técnicas para descrever os requisitos do sistema
- Explicar porquê os requisitos de software devem ser organizados em um documento de requisitos

## Tópicos abordados

---

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

## Engenharia de requisitos

---

- O processo de estabelecer as funções que o cliente requer de um sistema e as restrições sob as quais ele deve funcionar e ser desenvolvido
- Os requisitos são descrições das funções e restrições que são geradas durante o processo de engenharia de requisitos

## O que é um requisito?

- Pode variar desde uma descrição abstrata em alto nível de uma função ou de uma restrição do sistema até uma especificação funcional matemática detalhada
- Isso é inevitável, uma vez que os requisitos podem servir para uma função dupla
  - Podem ser a base para uma licitação de um contrato – portanto, deve ser abertos a interpretações
  - Podem ser a base para o contrato em si – portanto, devem ser definidos em detalhes
  - Ambas as definições podem ser chamadas de requisitos

## Abstração de requisitos (Davis)

Se uma empresa deseja estabelecer um contrato para o desenvolvimento de um grande projeto de software, ela tem de definir suas necessidades de maneira suficientemente abstrata para que uma solução não seja predefinida. Os requisitos devem ser redigidos de modo que os diversos fornecedores possam apresentar propostas, oferecendo, talvez, diferentes maneiras de atender às necessidades organizacionais do cliente. Uma vez estabelecido um contrato, o fornecedor precisa preparar uma definição de sistema para o cliente, com mais detalhes, de modo que o cliente compreenda e possa validar o que o software fará. Esses dois documentos podem ser chamados de *documentos de requisitos* do sistema.

## Tipos de requisitos

- Requisitos de usuário
  - Declarações em linguagem natural acrescidas de diagramas das funções que o sistema fornece e suas restrições operacionais. Elaborados para os clientes
- Requisitos de sistema
  - Um documento estruturado, estabelecendo descrições detalhadas das funções do sistema. Elaborados como um contrato entre o cliente e o contratante
- Especificação de software
  - Uma descrição detalhada do software, que pode servir como base para um projeto ou implementação. Elaborada para desenvolvedores

## Definições e especificações

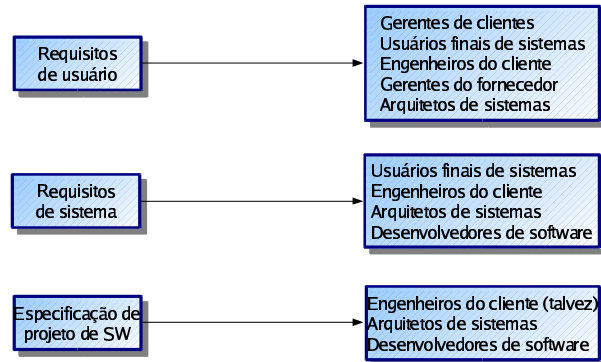
### Definição dos requisitos do usuário:

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

### Definição dos requisitos do usuário:

- 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 associada 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 Quanto 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 arquivo selecionado.

## Leitores de requisitos



## Requisitos funcionais e não-funcionais

- **Requisitos funcionais**
  - Declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em situações particulares
- **Requisitos não-funcionais**
  - Restrições nas funções oferecidas pelo sistema, tais como restrições de tempo, restrições no processo de desenvolvimento, padrões, etc.
- **Requisitos de domínio**
  - Requisitos que vêm do domínio de aplicação do sistema e que refletem características daquele domínio

## Requisitos funcionais

- Descrevem funcionalidades das funções do sistema
- Dependem do tipo de software, usuários esperados e o tipo de sistema onde o software será usado
- Requisitos funcionais do usuário podem ser declarações de alto nível daquilo que o sistema deve fazer
- Requisitos funcionais do sistema devem descrever as em detalhe as funções do sistema

## Exemplos de requisitos funcionais

- O usuário deve ser capaz de efetuar consultas sobre todo o conjunto inicial de bancos de dados ou selecionar um subconjunto deste
- O sistema deve fornecer visualizadores apropriados para o usuário ler documentos no arquivo de documentos
- Cada pedido deve possuir um identificador único (NUM\_PEDIDO), que o usuário deve ser capaz de copiar para a área de armazenamento permanente de contas

## Imprecisão dos requisitos

- Surgem problemas quando os requisitos não são definidos com precisão
- Requisitos ambíguos podem ser interpretados de maneiras diferentes por desenvolvedores e usuários
- Considere-se o termo ‘visualizadores apropriados’
  - Intenção do usuário – visualizador de fim específico para cada tipo diferente de documento
  - Interpretação do desenvolvedor – fornecer um visualizador de texto que mostra o conteúdo do documento

## Completude e consistência dos requisitos

- Em princípio, os requisitos deveriam ser ao mesmo tempo completos e consistentes
- Completos
  - Eles deveriam incluir descrições de todas as funções requeridas
- Consistentes
  - Não deveriam haver conflitos ou contradições nas descrições dos recursos do sistema
- Na prática, é impossível produzir um documento de requisitos completo e consistente

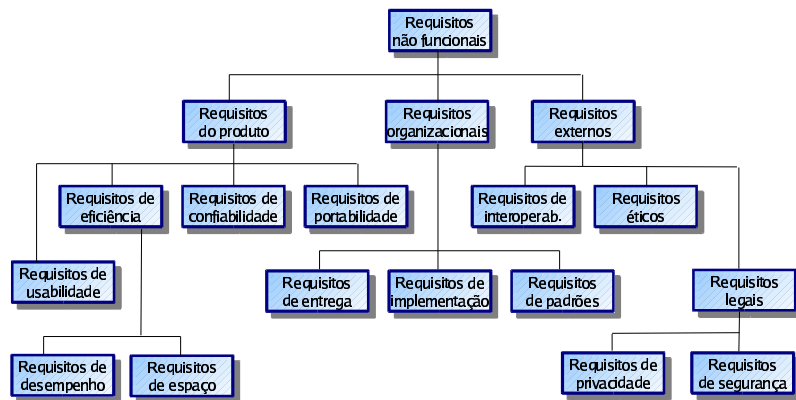
## Requisitos não-funcionais

- Define as propriedades e restrições do sistema, p. ex., confiabilidade, tempo de resposta e requisitos de armazenamento. Restrições são a capacidade do dispositivo de I/O, representações do sistema, etc.
- Os requisitos do processo podem também ser especificados, definindo uma ferramenta CASE, linguagem de programação ou método de desenvolvimento específicos
- Requisitos não-funcionais podem ser mais críticos que requisitos funcionais. Se eles não forem satisfeitos, o sistema é inútil

## Classificações não-funcionais

- Requisitos do produto
  - Requisitos que especificam que o produto entregue deve ter um comportamento específico. Exemplos: velocidade de execução, confiabilidade, etc.
- Requisitos organizacionais
  - Requisitos que são consequência de políticas e procedimentos organizacionais. Exemplos: padrões de processo utilizados, requisitos de implementação, etc.
- Requisitos externos
  - Requisitos que podem surgir de fatores que são externos ao sistema e ao seu processo de desenvolvimento. Exemplos: requisitos de interoperabilidade, requisitos legais, etc.

## Tipos de requisitos não funcionais



©Ian Sommerville 2000

Software Engineering, 6th edition, Chapter 5

Slide 17

## Exemplos de requisitos não-funcionais

- **Requisito de produto**
  - 4.C.8 Deve-se permitir que toda a comunicação necessária entre o APSE e o usuário possa ser expressa no conjunto de caracteres padrão Ada
- **Requisito organizacional**
  - 9.3.2 O processo de desenvolvimento do sistema e documentos entregues deve estar de acordo com o processo e produtos a serem entregues definido em XYZCo-SP-STAN-95
- **Requisito externo**
  - 7.6.5 O sistema não deve abrir nenhuma informação pessoal dos clientes além do seu nome e número de referência para os operadores do sistema

©Ian Sommerville 2000

Software Engineering, 6th edition, Chapter 5

Slide 18

## Metas e requisitos

- Requisitos não-funcionais podem ser muito difíceis de serem definidos com precisão e requisitos imprecisos podem ser difíceis de verificar.
- **Meta**
  - Uma intenção geral do usuário, tal como a facilidade de uso
- **Requisito não-funcional verificável**
  - Uma declaração usando alguma medida que possa ser objetivamente testada
- Metas são úteis para os desenvolvedores à medida em que elas conduzem as intenções dos usuários do sistema

©Ian Sommerville 2000

Software Engineering, 6th edition, Chapter 5

Slide 19

## Exemplos

- **Uma meta do sistema**
  - O sistema deve ser fácil de usar por controladores experientes e deve ser organizado de modo a minimizar os erros dos usuários.
- **Um requisito não-funcional verificável**
  - Controladores experientes devem ser capazes de usar todas as funções do sistema após um total de duas horas de treinamento. Após esse treinamento, o número médio de erros cometidos por usuários experientes não deve exceder dois por dia.

©Ian Sommerville 2000

Software Engineering, 6th edition, Chapter 5

Slide 20

## Métricas de requisitos

Propriedade	Métrica
Velocidade	Transações processadas por segundo Tempo de resposta ao usuário/evento Tempo de <i>refresh</i> da tela
Tamanho	Kbytes Número de chips de RAM
Facilidade de uso	Tempo de treinamento Número de telas de ajuda
Confiabilidade	Tempo médio para falhar Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício após falha Porcentagem de eventos que causam falhas Probabilidade de que os dados sejam corrompidos por falhas
Portabilidade	Porcentagem de declarações dependentes de sistemas-alvo Número de sistemas-alvo

## Interação de requisitos

- Conflitos entre diferentes requisitos não-funcionais são comuns em sistemas complexos
- Sistema de nave espacial
  - Para minimizar o peso, o número de chips separados no sistema deve ser minimizado
  - Para minimizar o consumo de energia, deve-se utilizar chips de menor capacidade
  - Entretanto, o uso de chips de menor capacidade significa que mais chips terão que ser utilizados. Qual deles é o requisito mais crítico?

## Requisitos de domínio

- Derivados do domínio da aplicação
- Descrevem características e recursos do sistema que refletem o domínio
- Podem ser novos requisitos funcionais, restrições nos requisitos existentes ou definições de novas computações
- Se os requisitos de domínio não forem satisfeitos, o sistema pode não ser trabalhável

## Requisitos de domínio de um sistema de biblioteca

- Deve haver uma interface de usuário padrão para todos os bancos de dados, que devem estar baseados no padrão Z39.50.
- Em função de restrições de direitos autorais, alguns documentos devem ser excluídos imediatamente após a chegada. Dependendo dos requisitos do usuário, esses documentos poderão ser imprimidos localmente no servidor do sistema para envio manual ao usuário ou roteados para uma impressora de rede.

## Sistema de proteção de trens

- A desaceleração do trem deve ser computada como:
  - $D_{\text{trem}} = D_{\text{controle}} + D_{\text{gradiente}}$onde  $D_{\text{gradiente}}$  é  $9.81\text{ms}^2 * \text{gradiente}/\alpha$  compensado e onde os valores de  $9.81\text{ms}^2 / \alpha$  são conhecidos para diferentes tipos de trens.

## Problemas de requisitos de domínio

- Compreensibilidade
  - Os requisitos são expressos na linguagem do domínio da aplicação
  - Isso é frequentemente mal compreendido pelos engenheiros de software que estão desenvolvendo o sistema
- Requisitos implícitos
  - Especialistas de domínio compreendem a área tão bem que eles não se preocupam em tomar os requisitos de domínio explícitos

## Requisitos de usuário

- Devem descrever requisitos funcionais e não-funcionais de tal forma que eles sejam compreensíveis pelos usuários do sistema que não possuem conhecimento técnico detalhado
- Os requisitos de usuário são definidos utilizando linguagem natural, tabelas e diagramas

## Problemas com a linguagem natural

- Falta de clareza
  - É difícil ser preciso sem tornar um documento difícil de ser lido
- Confusão de requisitos
  - Requisitos funcionais e não-funcionais tendem a estar misturados
- Fusão de requisitos
  - Vários requisitos diferentes podem ser expressos juntos

## Requisito de banco de dados

**4.A.5** O banco de dados deve suportar a geração e controle de objetos de configuração, isto é, objetos que são eles mesmos agrupamentos de outros objetos no banco de dados. Os recursos de controle de configuração devem permitir acesso aos objetos em um grupo de versão pelo usuário com um nome incompleto.

## Requisito para a grade de um editor

**2.6 Recursos de grade** Para auxiliar no posicionamento de entidades em um diagrama, o usuário pode ligar uma grade em centímetros ou polegadas, por meio de uma opção no painel de controle. Inicialmente, a grade está desligada. A grade pode ser ligada e desligada a qualquer tempo durante uma sessão de edição e pode ser alternada entre polegadas e centímetros a qualquer tempo. Uma opção da grade será fornecida para a visão “reduzir para caber”, mas o número de linhas de grade mostradas será reduzido para evitar o preenchimento de diagramas menores com linhas de grade.

## Problemas com requisitos

- Os requisitos de banco de dados incluem tanto informação conceitual quanto informação detalhada
  - Descreve o conceito de recursos de controle de configuração
  - Inclui o detalhe que objetos podem ser acessados usando um nome incompleto
- Os requisitos de grade misturam três tipos diferentes de requisitos
  - Requisitos funcionais conceituais (a necessidade de uma grade)
  - Requisitos não-funcionais (unidades da grade)
  - Requisitos não-funcionais de IU (seleção de grade)

## Representação estruturada

### 2.6 Recursos de Grade

**2.6.1 O editor deverá fornecer um recurso de grade, em que uma matriz de linhas horizontais e verticais constitua um fundo da janela do editor.** Essa grade deverá ser uma tela passiva em que o alinhamento de entidades é de responsabilidade do usuário. *Lógica:* uma grade ajuda o usuário a criar um diagrama 'limpo', com entidades bem espaçadas. Embora uma grade ativa, em que as entidades saltam as linha 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/Ferramentas/DE/FS, Seção 5.6.



## Requisitos de usuário detalhados

### 3.5.1 Adicionando nós a um desenho

3.5.1.1 O editor deve fornecer um recurso aos usuários para adicionar nós de um tipo especificado a seu desenho

3.5.1.2 A sequência de ações para acrescentar um nó deve ser como se segue:

1. O usuário deve selecionar o tipo de nó a ser acrescentado
2. O usuário deve mover o cursor para a posição aproximada do nó no diagrama e indicar que o símbolo do nó deve ser adicionado naquele ponto.
3. O usuário deve então arrastar o símbolo do nó para a sua posição final.

*Lógica:* o usuário é a melhor pessoa para decidir onde posicionar o nó no diagrama. Essa abordagem dá ao usuário o controle direto sobre a seleção do tipo de nó e seu posicionamento.

*Especificação:* ECLIPSE/WS/Ferramentas/DE/FS, Seção 3.5.1.

## Linhas-guia para a elaboração de requisitos

- Definir um formato padrão e usá-lo para todos os requisitos
- Utilizar o idioma de forma consistente. Usar “deve” para requisitos obrigatórios, “deveria” para requisitos desejáveis
- Usar texto destacado para identificar as partes principais do requisito
- Evitar o uso do jargão de computação

## Requisitos de sistema

- Especificações mais detalhadas de requisitos de usuário
- Servem como base para projetar o sistema
- Podem ser utilizados como parte do contrato de sistema
- Os requisitos de sistema podem ser expressos utilizando os modelos de sistema descritos no Capítulo 7

## Requisitos e projeto

- Em princípio, os requisitos deveriam dizer o que o sistema deveria fazer e o projeto deveria dizer como ele deveria fazê-lo
- Na prática, requisitos e projeto são inseparáveis
  - Uma arquitetura do 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 especificações em linguagem natural (LN)

- Ambigüidade
  - Os leitores e redatores de requisitos devem interpretar as mesmas palavras da mesma maneira. A LN é naturalmente ambígua, portanto isso é muito difícil
- Excesso de flexibilidade
  - Pode-se dizer a mesma coisa em uma série de formas diferentes na especificação
- Falta de modularização
  - As estruturas da LN são inadequadas para estruturar requisitos de sistemas

## Alternativas à especificação em LN

Notação	Descrição
Linguagem natural estruturada	Essa abordagem depende da definição de formulários-padrão ou <i>templates</i> para expressar a especificação de requisitos.
Linguagem de descrição de programa	Essa abordagem utiliza uma linguagem semelhante a uma linguagem de programação, mas com recursos mais abstratos para especificar os requisitos pela definição de um modelo operacional do sistema.
Notações gráficas	Uma linguagem gráfica, complementada com anotações de texto, é utilizada para definir os requisitos funcionais do sistema. Um exemplo anterior dessa linguagem gráfica foi SADT (Ross, 1977; Schoman e Ross, 1977). Mais recentemente, as descrições de casos de uso (Jacobson et al., 1993) foram utilizadas.
Especificações matemáticas	São notações com base em conceitos matemáticos, como uma máquina de estados finitos e conjuntos. Essas especificações não são ambíguas e reduzem as discussões entre cliente e fornecedor sobre a funcionalidade do sistema. Contudo, a maioria dos clientes não compreende as especificações formais, e reluta em aceitá-las no momento de uma contratação de sistema.

## Especificações em linguagem estruturada

- Uma forma restrita da linguagem natural pode ser usada para expressar requisitos
- Isso remove alguns dos problemas resultantes da ambigüidade e da flexibilidade e impõe um grau de uniformidade sobre a especificação
- Frequentemente suportada usando uma abordagem baseada em formulários

## Especificações baseadas em formulários

- Definição da função ou entidade
- Descrição de entradas e de onde elas vêm
- Descrição de saídas e para onde elas vão
- Indicação de outras entidades exigidas
- Pré e pós condições (se for apropriado)
- Os efeitos colaterais (se houver)

## Especificação de nó baseada em formulários

ECLIPSE/Workstation/Ferramentas/DE/FS/3.5.1

**Função** Adicionar nós.

**Descrição** Adiciona um nó a 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 de 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 tela 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

**Definição:** ECLIPSE/Workstation/Ferramentas/DE/RD/3.5.1

## Definição de requisitos baseada em *Program Description Language – PDL*

- Requisitos podem ser definidos operacionalmente usando uma linguagem similar a uma linguagem de programação, mas com mais flexibilidade de expressão
- Mais apropriada para duas situações
  - Onde uma operação é especificada como uma sequência de ações e a ordem é importante
  - Quando interfaces de hardware e software têm que ser especificadas
- As desvantagens são
  - A PDL pode não ter poder de expressão suficiente para definir conceitos do domínio
  - A especificação será confundida com um projeto

## Parte de uma especificação de ATM

```
class ATM {
// área de declarações
public static void main (String args[] throws CartaoInvalido {
    try {
        thisCard.read(); // pode disparar a exceção CartaoInvalido
        pin = KeyPad.readPin(); attempts = 1;
        while ( !thisCard.pin.equals (pin) & attempts < 4)
        { pin = KeyPad.readPin(); attempts = attempts + 1;
        }
        if ( !thisCard.pin.equals(pin))
            throw new CartaoInvalido("Senha errada");
        thisBalance = thisCard.getBalance();
        do { Screen.prompt("Selecione um serviço:");
            service = Screen.touchKey();
            switch (Service) {
                case Services.withdrawalWithReceipt:
                    receiptRequired = true;
            }
        } while (true);
    }
}
```

## Especificação de interface

- A maioria dos sistemas deve interagir com outros sistemas, e as interfaces de operação devem ser especificadas como parte dos requisitos
- Três tipos de interface podem ter que ser definidas
  - Interfaces procedurais
  - Estruturas de dados que são permutadas
  - Representações de dados
- Notações formais são uma técnica efetiva para especificação de interface

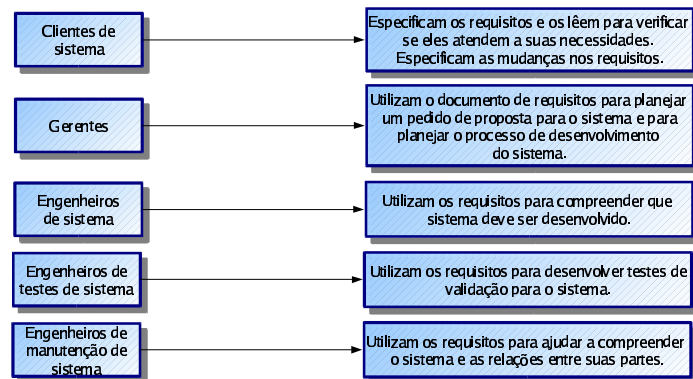
## Descrição de interface em PDL

```
Interface ServidorImpressao {  
  
    // define um servidor de impressão abstrato  
    // requer: impressora de interface Impressora, interface ImprimeDoc  
    // fornece: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter  
  
    void inicializa ( Impressora i );  
    void imprime ( Impressora i, ImprimeDoc d );  
    void mostraFilalmp ( Impressora i );  
    void cancelaTrabalmp ( Impressora i, ImprimeDoc d );  
    void trocalmp ( Impressora i1, Impressora i2, ImprimeDoc d );  
} //ServidorImpressao
```

## O documento de requisitos

- O documento de requisitos é a definição oficial do que é exigido dos desenvolvedores do sistema
- Deve incluir tanto a definição quanto a especificação de requisitos
- NÃO é um documento de projeto. Tanto quanto possível, deve definir O QUÊ o sistema deveria fazer, em vez de COMO ele deve fazê-lo

## Usuários de um documento de requisitos



## Requisitos do documento de requisitos

- Especificar o comportamento externo do sistema
- Especificar restrições de implementação
- Fácil de modificar
- Servir como ferramenta de referência para manutenção
- Registrar a estratégia sobre o ciclo de vida do sistema, ou seja, prever mudanças
- Caracterizar respostas a eventos inesperados

## Padrão de requisitos do IEEE

---

- Introdução
- Descrição geral
- Requisitos específicos
- Apêndices
- Índice
- Esta é uma estrutura genérica, que deve ser instanciada para sistemas específicos

## Estrutura do documento de requisitos

---

- Introdução
- Glossário
- Definição dos requisitos do usuário
- Arquitetura do sistema
- Especificação dos requisitos do sistema
- Modelos do sistema
- Evolução do sistema
- Apêndices
- Índice

## Pontos principais

---

- Os requisitos definem o que o sistema deveria fazer e define restrições a respeito da sua operação e implementação
- Requisitos funcionais estabelecem os serviços que o sistema deveria fornecer
- Requisitos não-funcionais restringem o sistema a ser desenvolvido ou o processo de desenvolvimento
- Requisitos de usuário são declarações de alto nível do que o sistema deveria fazer

## Pontos principais

---

- Requisitos de usuário devem ser escritos em linguagem natural, tabelas e diagramas
- Requisitos de sistema se destinam a comunicar as funções que o sistema deve fornecer
- Os requisitos de sistema podem ser escritos em uma linguagem natural estruturada, uma PDL ou em uma linguagem formal
- Um documento de requisitos de software é uma declaração acordada dos requisitos do sistema