

# Engenharia de Software

**Professor:**

Zady Castaneda Salazar



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SÃO PAULO  
Campus Campinas

# Aula 7 Engenharia de Requisitos



# Aula 7 Engenharia de Requisitos

## Agenda:

- **Conceitos básicos**
- **Definição de Requisito de software**
- **Classificação de Requisitos do sistema**



- A Engenharia de Software envolve as seguintes fases:
  1. Definição
  2. Desenvolvimento do Software
  3. Verificação, Entrega e Manutenção
  4. Integração (opcional)

- A Engenharia de Software envolve as seguintes fases:

## 1. Definição

- É iniciada com o **Planejamento do Desenvolvimento do Software**, obtendo um documento de **Plano do Software**, o qual será produzido e revisado pelo Gerente do Projeto;
- Ainda nesta fase é realizada a **Análise de Requisitos do Software**, a qual vai permitir que funções, dentro do sistema como um todo serão atribuídas ao software;
- A última tarefa relacionada a esta fase é a revisão da **Especificação de Requisitos do Software**, o qual é o documento resultante da Análise de Requisitos;

## 2. Desenvolvimento do Software

- É a fase que tem, como ponto de partida, os documentos produzidos na fase anterior, particularmente, o **Plano do Software** e a **Especificação de Requisitos do Software**;
- Com base nestes documentos, inicia-se a etapa de **Projeto do Software**, onde serão descritos aspectos relacionados ao funcionamento do software como a sua arquitetura e as estruturas de dados;
- Após avaliada esta definição, inicia-se a etapa de **Projeto Detalhado**, onde os aspectos algorítmicos e comportamentais do software são definidos;
- Finalmente, a **etapa de codificação** é encaminhada, seja com base no uso de uma linguagem de programação clássica ou com o auxílio de uma ferramenta CASE, o resultado desta etapa sendo a listagem dos **programas-fonte do software em desenvolvimento**;

## 3. Verificação, Entrega e Manutenção

- É a última fase do processo, a qual envolve as atividades de **teste do software**, preparando-o para a **entrega**;
- Uma vez entregue, inicia-se, ao longo de toda a vida útil do software a **etapa de manutenção**, a qual permitirá manter o software em funcionamento a partir da **correção de novos erros** que tenham sido detectados com o software em funcionamento, da introdução ou **melhorias de funções do software**, da adaptação do software para novas plataformas de hardware existentes.

## 4. Integração de Sistemas (opcional)

- O conjunto de atividades a ser desenvolvido nesta etapa é o de **conexão dos diferentes** subsistemas construídos ou adquiridos para compor o sistema.
- É uma atividade bastante complexa, devido principalmente, à grande diversidade de tecnologias envolvidas na concepção dos diferentes sistemas.
- Um problema comumente encontrado nesta etapa é o **mal funcionamento de um subsistema como consequência de uma definição imprecisa de funcionalidade de outro subsistema.**



- Um dos resultados da Engenharia de Software é a definição de aspectos como **funcionalidade e desempenho do software**.
- O trabalho essencial do analista de sistemas é **acomodar os requisitos de funcionalidades e desempenho** da forma mais eficiente possível, tendo que, para isto, adquirir e/ou desenvolver os componentes de software.

# Definição de Requisitos

---

## 1. O que são os requisitos de software ?

- Um requisito consiste na **definição documentada de uma propriedade ou comportamento que um produto ou serviço particular deve atender.**
- Requisitos são uma especificação **do que deve ser construído.**
- É o processo de estabelecer os **serviços que o cliente requer de um sistema** e as **restrições sob as quais deve ser desenvolvido e operar.**

# Definição de Requisitos

- Requisitos são as descrições das **funções** e as suas respectivas **restrições**.
- Pode variar de uma:
  - **declaração abstrata de alto nível** de uma função que o sistema deve fornecer ou de uma restrição do sistema,
  - a uma **definição detalhada**, matematicamente formal, de uma função do sistema.



# Importância dos Requisitos

---

O cenário de desenvolvimento de software atual e o cenário idealizado junto à engenharia de software ainda estão distantes.

Vários fatores contribuem para isso, podemos citar:

- O **não uso** e o **mau uso** dos fundamentos da engenharia de software para apoiar as atividades do desenvolvimento.

Isso tem como consequência o **crescente custo** com manutenção dos sistemas, **retrabalho** (em nível de requisitos, projeto, codificação, teste) causado por uma **definição** do domínio do problema **mal elaborada** nas fases iniciais do desenvolvimento.

## 1. Por que os requisitos são importantes?

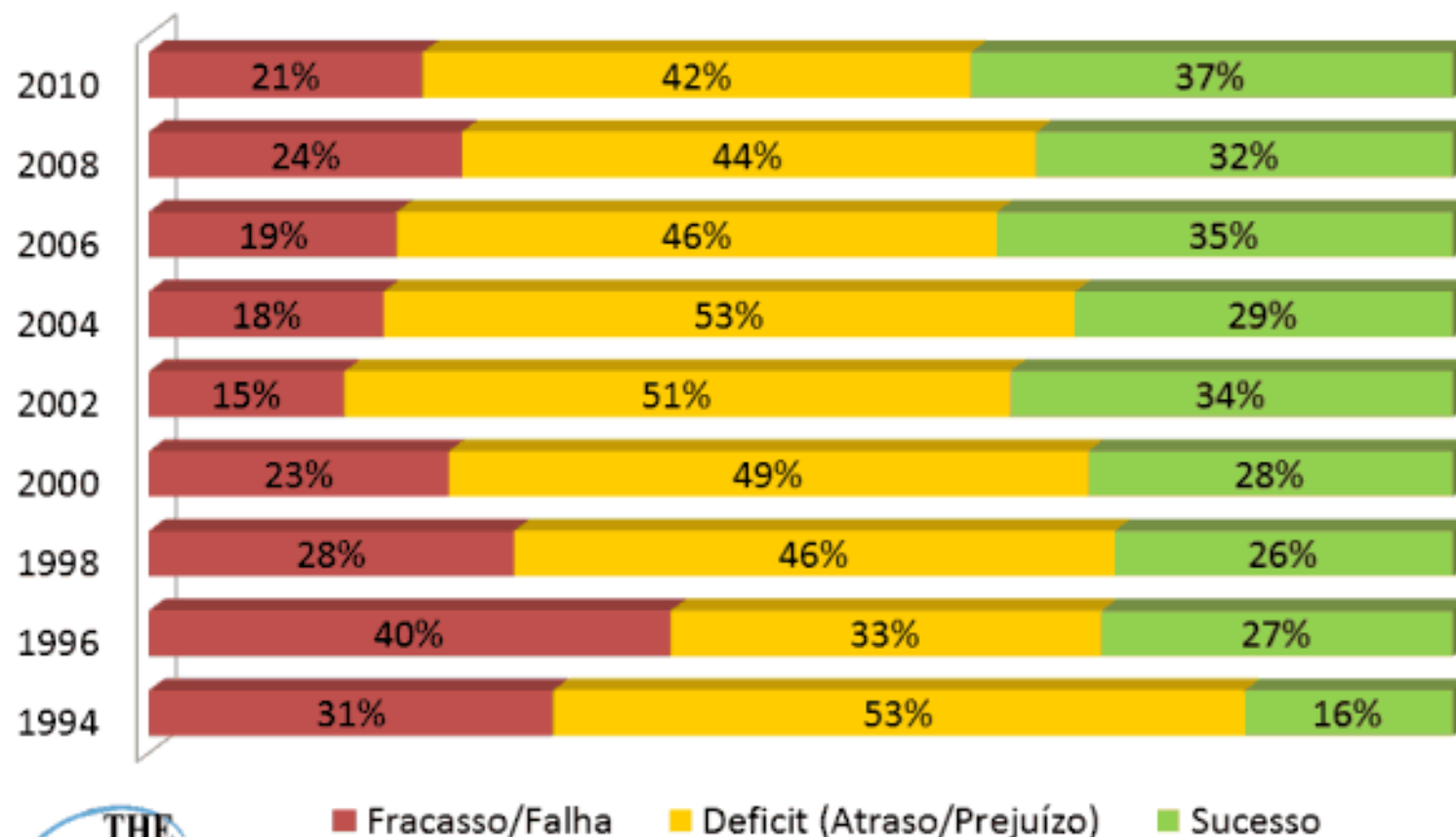
Pesquisa em mais de 350 empresas sobre os seus mais de 8.000 Projetos de software – 30 % dos projetos foram cancelados. Dos concluídos, 9% entregues dentro do prazo e do valor estimado(Standish Group –1994).

Fatores principais relatados como causas das falhas:

1. Requisitos incompletos (13.1%)
2. Falta de envolvimento por parte do usuário (12.4%)
3. Falta de recursos (10.6%)
4. Expectativas não realistas (9.9%)
5. Falta de apoio dos executivos (9.3%)
6. Modificações nos requisitos e nas especificações (8.7%)
7. Falta de planejamento (8.1%)
8. O sistema não era mais necessário (7.5%)

# Importância dos Requisitos

## 1. Por que os requisitos são importantes?



Fonte: Standish Group; CHAOS Manifesto 2011, CHAOS Summary for 2010, Extreme CHAOS 2001.

[CHAOS Report: Métodos Ágeis Aumentam Taxa de Sucesso de Projetos](#)

# Requisitos

## 1. Por que os requisitos são importantes?

### Modelo de Kano

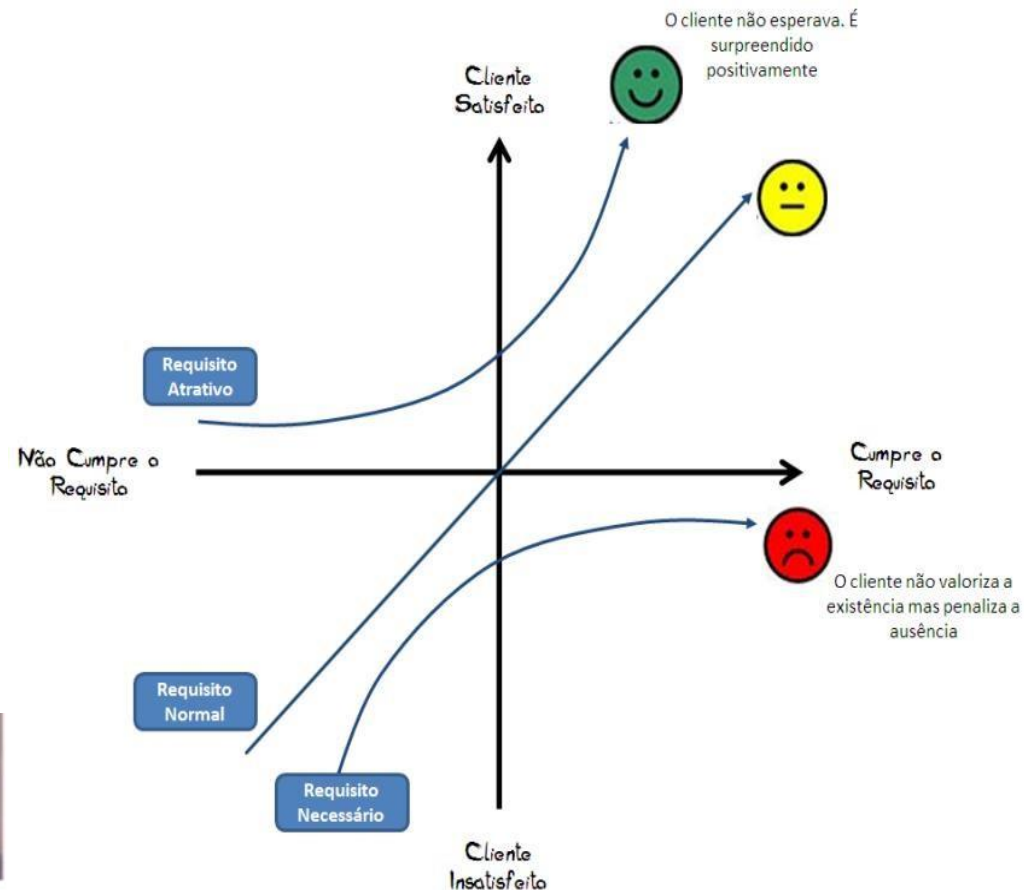
Modelo teórico que se baseia na relação do grau de desempenho (**eixo horizontal**) com o grau de satisfação (**eixo vertical**), sendo que cada atributo resultante das diferentes relações entre os graus é descrito na sequência.

#### Origins of the Kano Model



■ Noriaki Kano

□ Professor at Tokyo Rika University



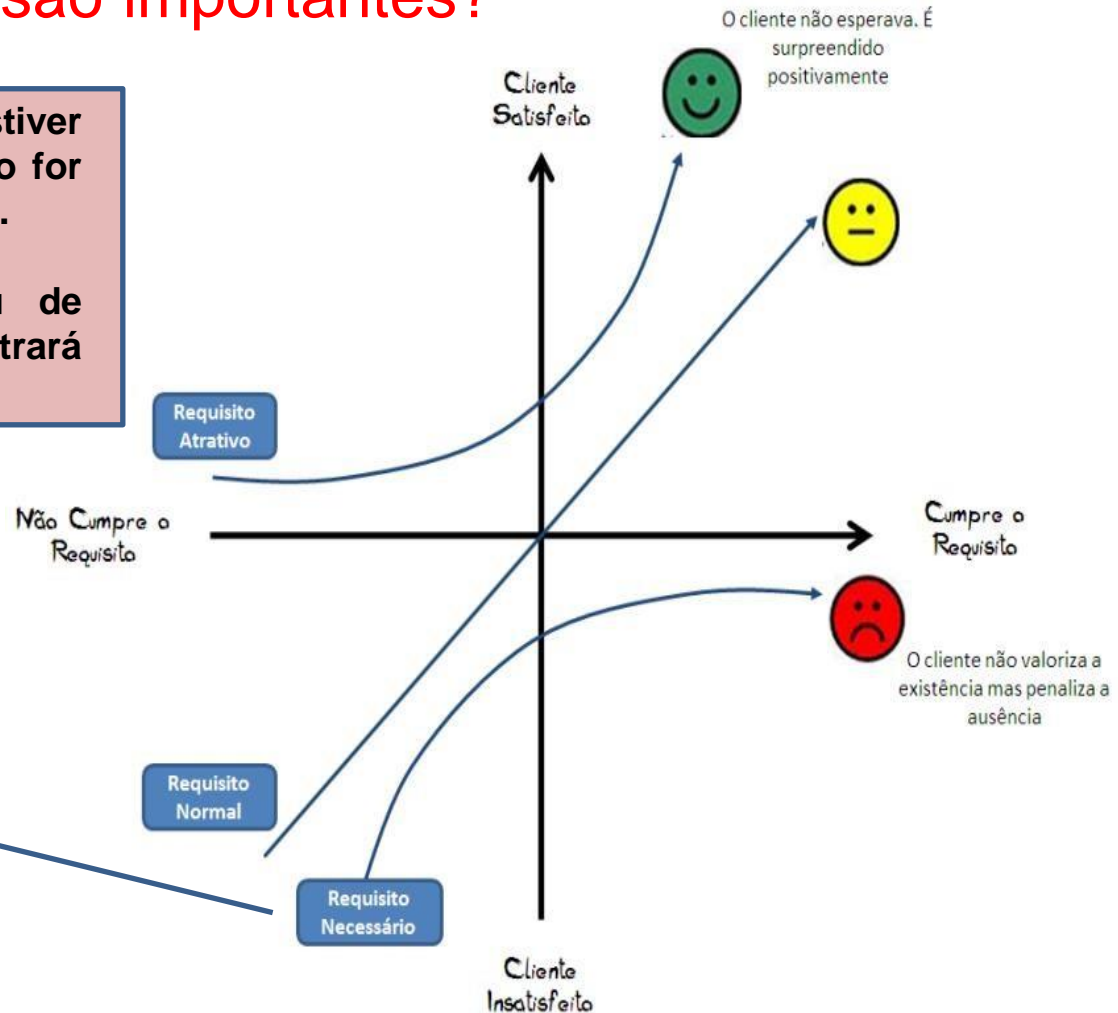


# Requisitos

## 1. Por que os requisitos são importantes?

**Requisito necessário (O):** Se este não estiver presente ou se o grau de desempenho for insuficiente, o cliente ficará insatisfeito.

Se estiver presente ou tiver grau de desempenho suficiente, não trará satisfação;

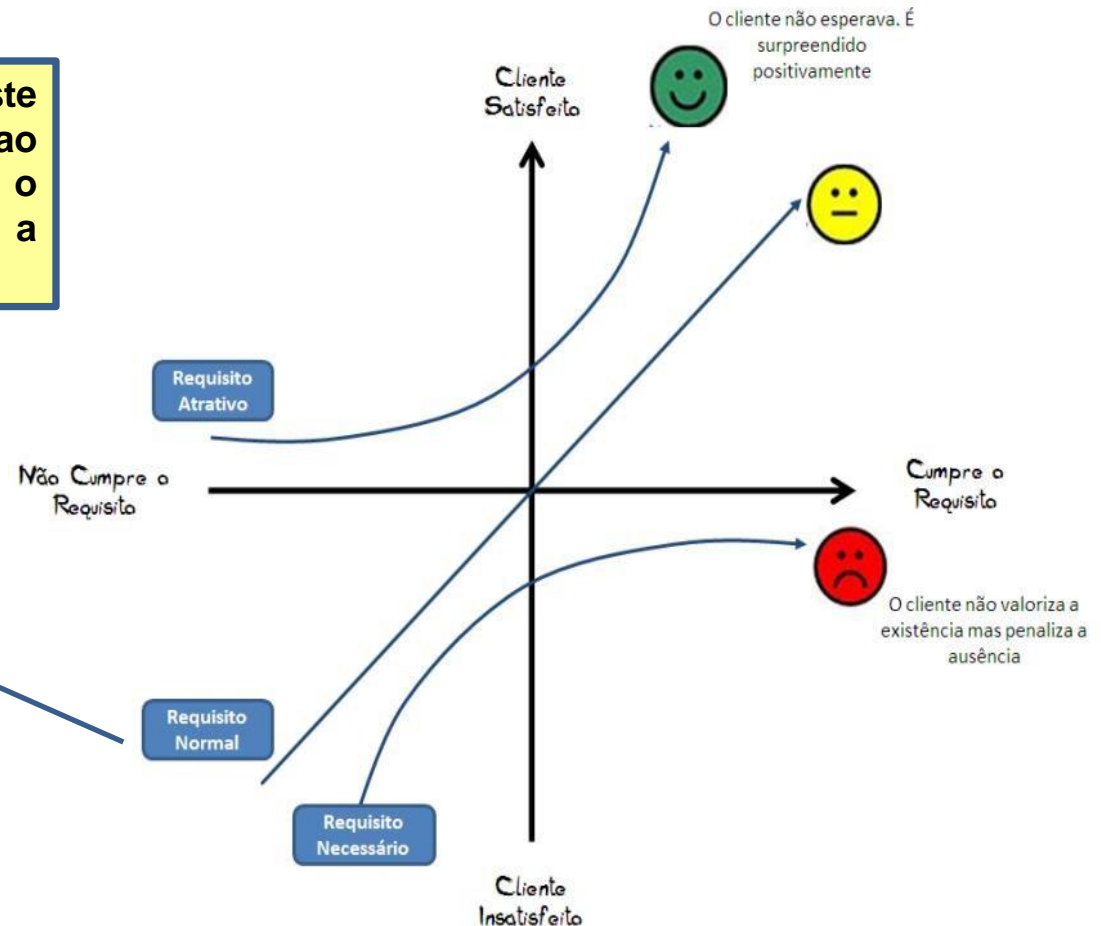




# Requisitos

## 1. Por que os requisitos são importantes?

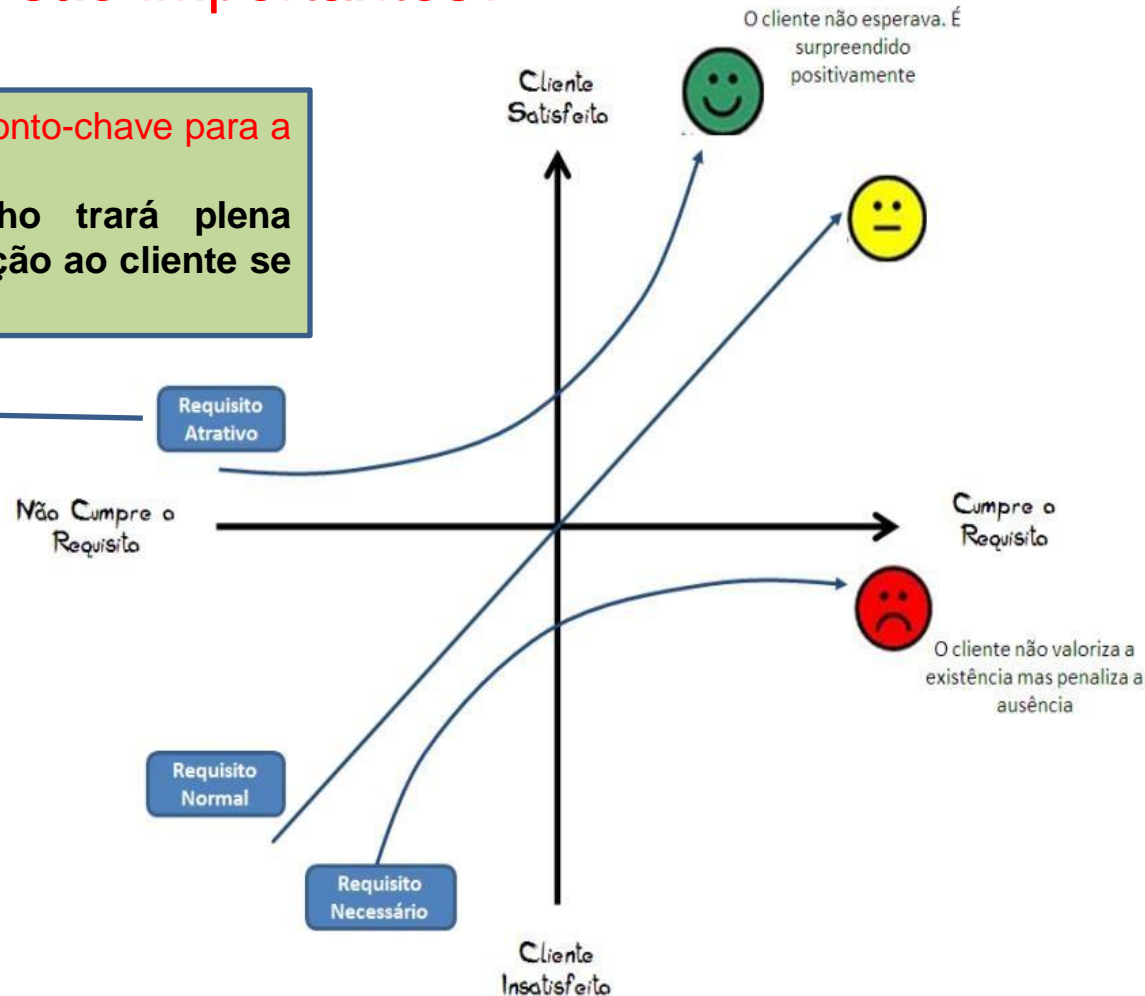
**Requisito normal (U):** quanto a este atributo, a satisfação é proporcional ao grau de desempenho, quanto maior o grau de desempenho, maior será a satisfação do cliente e vice-versa;



# Requisitos

## 1. Por que os requisitos são importantes?

**Requisito atrativo (A):** este atributo é ponto-chave para a satisfação do cliente, Se tiver alto grau de desempenho trará plena satisfação, porém, não trará insatisfação ao cliente se não for atendido;



## Problemas no momento de elicitar os requisitos

- Falta de cuidado com os requisitos pode levar a construção de um sistema que resolve o problema errado;
- Não funciona como esperado;
- Difícil para os usuários entenderem e utilizarem;
- Alto custo.



# Exemplo:

Uma empresa deseja estabelecer um contrato para o desenvolvimento de um grande projeto de software, ela precisa:

- **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, diferentes maneiras de atender às necessidades organizacionais do cliente.

Após a aprovação do contrato,

- **o fornecedor deve redigir uma definição mais detalhada do sistema para o cliente**, de modo que o cliente possa compreender e validar o que o software fará.

**ESSES DOIS NIVEIS DE DESCRIÇÃO ESTÃO PRESENTES  
NO DOCUMENTO DE REQUISITOS DO SISTEMA.**

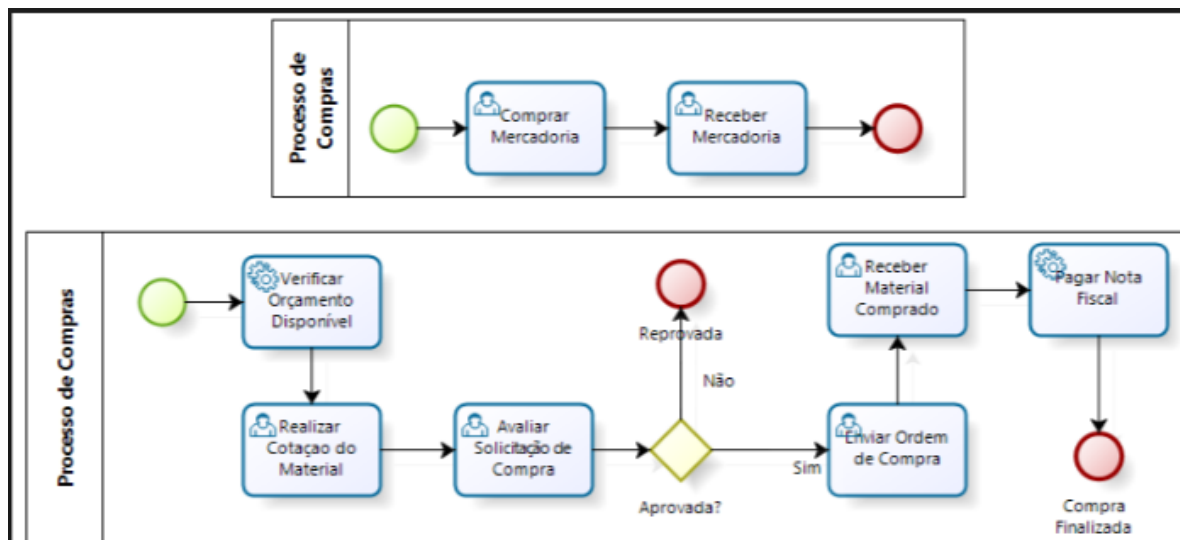
Durante o processo de engenharia de requisitos surgem problemas pela falta de uma clara separação entre esses níveis de descrição.

☐ Requisitos do usuário:

☐ Requisitos do sistema:

## Requisitos do usuário:

- Para requisitos abstratos de alto nível.
- **Declarações em linguagem natural** e em diagramas sobre as funções que o sistema deve fornecer e as restrições sob as quais deve operar.



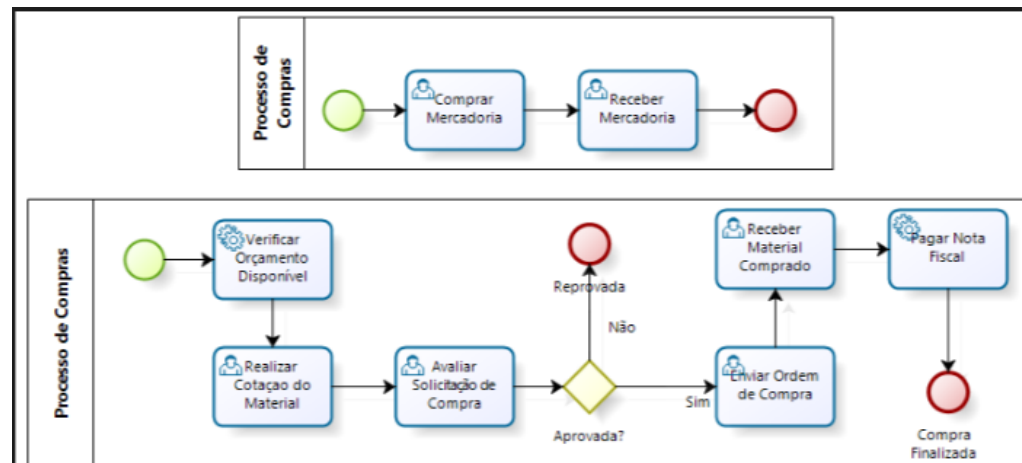
# Requisitos de Usuário

- Devem descrever os requisitos funcionais e não funcionais de modo compreensível pelos usuários do sistema, que não tem conhecimentos técnicos detalhados.
- Devem especificar apenas o comportamento externo do sistema e evitar, sempre que possível características de projeto do sistema.
- Consistência para separar requisitos obrigatórios de desejáveis.



# Requisitos de Usuário

- Requisitos de usuário são declarações, em linguagem natural e diagramas, sobre os serviços que o sistema oferece e as restrições para a sua operação.
- Escrito para os **clientes**.
- Exemplos:
  1. Documento de Visão
  2. Declaração de Escopo
  3. Mapeamento de processos





# Regras para Requisitos de Usuário

- Adote um formato padrão e use-o para todos os requisitos.
- O formato pode mostrar :
  - ☐ **requisito inicial em negrito**,
  - ☐ uma declaração da justificativa lógica de cada requisito de usuário,
  - ☐ uma referencia à especificação detalhada de requisito do sistema,
  - ☐ informações de quem propôs o requisito de modo que se saiba a quem consultar se o requisito tiver de ser mudado.
- Use linguagem consistentemente. Use **deve** para requisitos obrigatórios e **pode** para os desejáveis
- Use **texto em negrito**, *itálico* ou cor para identificar partes chave dos requisitos
- Evite jargão de computação.

# Exemplo para Requisitos de Usuário

- 1 (procedimento no.): 1. Separação de mercadorias
- 2 (especificação): O sistema deve emitir guia de separação de mercadorias com suas localizações
- 3 (requisitos):
  1. O usuário deve solicitar a emissão de guias em tela
  2. O usuário pode selecionar opção da operação fiscal
  3. O usuário deve selecionar o tipo de transporte
  3. O usuário pode selecionar o período para emissão guias
  4. O usuário deve confirmar a emissão
- 4 (lógica): O usuário emite as guias para separação e arrumação para transporte de acordo com tipo de operação fiscal
- 5 (usuário): Os usuários são depósitos e filiais
- 6 (restrições): Não existe
- 7 (data): Última definição 29/09/2008

## Requisitos do sistema

- Para a descrição detalhada do que o sistema deve fazer.
- Definem detalhadamente as funções, os serviços e as restrições operacionais do sistema.
- Deve definir o que será implementado.
- No documento de requisitos os requisitos do sistema devem de ser precisos.
- Pode ser parte do contrato entre o comprador do sistema (cliente) o desenvolvedor do software.

## Definição dos requisitos do usuário

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

## 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.

## Classificação de Requisitos do sistema

- Requisitos funcionais
- Requisitos não funcionais
- Requisitos de domínio

# Tipos de Requisitos de Sistema

---

## Requisitos funcionais

- Declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações.

## Requisitos não funcionais

- Restrições sobre os serviços ou as funções oferecidas pelo sistema.

## Requisitos de domínio

- Requisitos que se originam do domínio da aplicação do sistema e que refletem características desse domínio (Podem ser requisitos funcionais e não funcionais).



- **O que é ?**
- Descrevem a **funcionalidade** ou os serviços do sistema.
- Dependem do **tipo de software**, das **expectativas dos usuários** e do **tipo de sistema** que está sendo desenvolvido.
- **Requisitos funcionais do usuário são descritos de forma bem geral, mas os requisitos funcionais de sistema descrevem a função de sistema detalhadamente.**



# Requisitos Funcionais: Exemplos

RF1. O usuário deve ser capaz de manipular arquivos no drive virtual

RF2. O usuário deve ser capaz de compartilhar arquivos e pastas com outros usuários.

RF3. O administrador deve ser capaz de manipular usuários.

RF4. O administrador deve administrar a utilização do disco virtual pelos usuários.

Id	Descrição
RF1	O usuário pode enviar arquivos para seu disco virtual
RF2	O usuário pode recuperar arquivos de seu disco virtual
RF3	O usuário pode excluir arquivos do seu disco virtual
RF4	O usuário pode compartilhar arquivos ou pastas com outros usuários
RF5	O usuário pode tornar dado arquivo público para ser recuperado por qualquer pessoa
RF6	O usuário pode pesquisar arquivos por nome ou conteúdo
RF7	O administrador dos usuários pode criar, modificar ou excluir usuários
RF8	O administrador precisa de ferramentas para tarifar o usuário de acordo com o uso de seu disco virtual (espaço usado, bytes transferidos, etc.)

Tabela 1 – Requisitos Funcionais



# Precisão dos Requisitos Funcionais

- A especificação de requisitos funcionais deve ser **completa** e **consistente**

- **Completeza**
- Todas as funções requeridas pelo usuário devem estar definidas
- **Consistência**
- Não devem haver definições contraditórias de requisitos.



- Problemas se originam da imprecisão na especificação de requisitos.
- Requisitos ambíguos podem ser interpretados de maneira diferente pelos desenvolvedores e usuário.

# Requisitos Não Funcionais

- **O que é ?**
- Requisitos não funcionais especificam ou restringem as propriedades emergentes do sistema como por exemplo desempenho, proteção, disponibilidade, linguagem de programação e outras propriedades emergentes do sistema.
- **Podem ser mais críticos que requisitos funcionais.**
- A falha em atender um requisito não funcional de sistema pode inutilizar o sistema. Exemplo:
  - ❑ Se uma aeronave não atende aos requisitos de confiabilidade, ela não será certificada como segura para operação.

# Requisitos Não Funcionais : Exemplos

- [RNF1] O sistema deve imprimir o relatório em até 5 segundos.
- [RNF2] Todos os relatórios devem seguir o padrão de relatórios especificado pelo setor Contabilidade da empresa.
- [RNF3] O sistema deve ser implementado em Java.

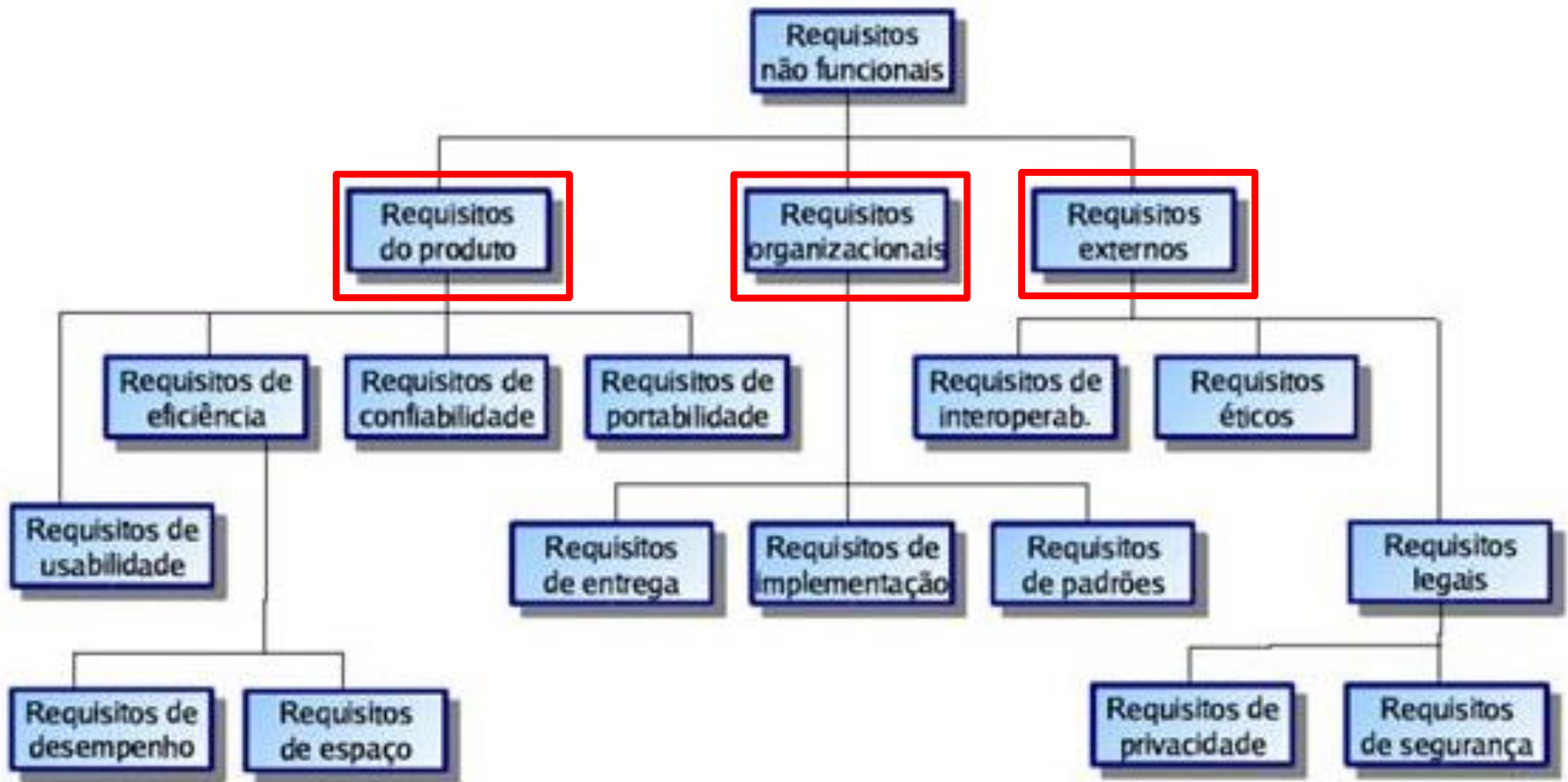
- [RNF 21]O software deve ser compatível com os browsers IE (versão 5.0 ou superior) e Firefox (1.0 ou superior);
- [RNF22]O software deve garantir que o tempo de retorno das consultas não seja maior do que 5 segundos.

- [RNF45] O sistema deverá possuir meios de impedir a cópia e divulgação de listas de clientes para pessoas não autorizadas.

- ❑ Surgem devido às necessidades do usuário, às restrições de orçamento, às políticas organizacionais, à necessidade de interoperabilidade com outros sistemas de software ou hardware ou a fatores externos.



# Tipos de Requisitos Não Funcionais



- **Requisitos de produto**
- Requisitos que especificam o comportamento do produto. Exemplo:
  - velocidade de execução,
  - confiabilidade,
  - portabilidade,
  - facilidade de uso,
  - espaço, etc.

- **Requisitos organizacionais**
- Requisitos que são consequência de políticas de procedimentos nas organizações do cliente e do desenvolvedor. Exemplo:
  - padrões de processos que devem ser utilizados,
  - requisitos de implementação,
  - linguagem de programação ou método de projeto usado,
  - requisitos de entrega que especificam quando o produto e a sua documentação devem ser entregues.etc.

- **Requisitos externos**
- Requisitos procedentes de fatores externos ao sistema e a seu processo de desenvolvimento.  
Exemplo:
  - requisitos de interoperabilidade,
  - requisitos legais,
  - requisitos éticos
  - requisitos de privacidade,
  - requisitos de segurança.



- Um problema comum é que eles **são difíceis de verificar**. Os usuários ou clientes definem esses requisitos como metas gerais.
- Sempre que possível, você deve **escrever os requisitos não funcionais quantitativamente**, de modo que eles possam ser testados objetivamente.

# Requisitos Não Funcionais

- Métricas usadas para especificar as propriedades não funcionais do sistema.

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

- **O que é ?**
- São derivados do domínio de aplicação do sistema, e descrevem características do sistema e qualidades que refletem o domínio.
- Podem ser novos requisitos funcionais em si, podem restringir os requisitos funcionais existentes, ou estabelecer como **realizar cálculos específicos** que devem ser realizados.
- Se não forem satisfeitos, poderá ser impossível fazer o sistema operar satisfatoriamente.

# Requisitos de Domínio: Exemplos

[RD45] Para implementar o Sistema de proteção de trens devera ser calculada a desaceleração do trem como:

- A desaceleração do trem deve ser calculada como:

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

onde

$D_{\text{gradiente}}$  é  $9.81\text{ms}^2 * \text{gradiente compensado}/\alpha$   
e onde os valores de  $9.81\text{ms}^2 / \alpha$  são conhecidos  
para diferentes tipos de trem.

RD2] Deve haver uma interface padrão com o usuário para todos os bancos de dados, que ter á como base o padrão Z39.50

**Z39.50** é um protocolo cliente servidor de padrão internacional que permite pesquisa e recuperação de informação em redes de computadores distribuídos. Corresponde à norma ISO 23950:1998

[RD1] O calculo da média final de cada aluno é dado pela fórmula:  $(\text{Nota1} * 2 + \text{Nota2} * 3)/5$ ;

[RD2]Um aluno pode se matricular em uma disciplina desde que ele tenha sido aprovado nas disciplinas consideradas pré-requisitos.

- **Principal problema com requisitos de domínio**
- Declarações implícitas
- **Especialistas** em um domínio podem **deixar de fornecer informações** em um requisito, simplesmente por acharem que essas informações são muito óbvias.
- Para os **desenvolvedores do sistema** **podem não ser óbvias** e implementar o requisito de forma equivocada.

Requisitos **documentados** são chamados de **explícitos**.

Requisitos que **não tenham sido declarados** por alguns ***STAKEHOLDERS***, ausentes da documentação são chamados de **implícitos**.

# Como validar os Requisitos?

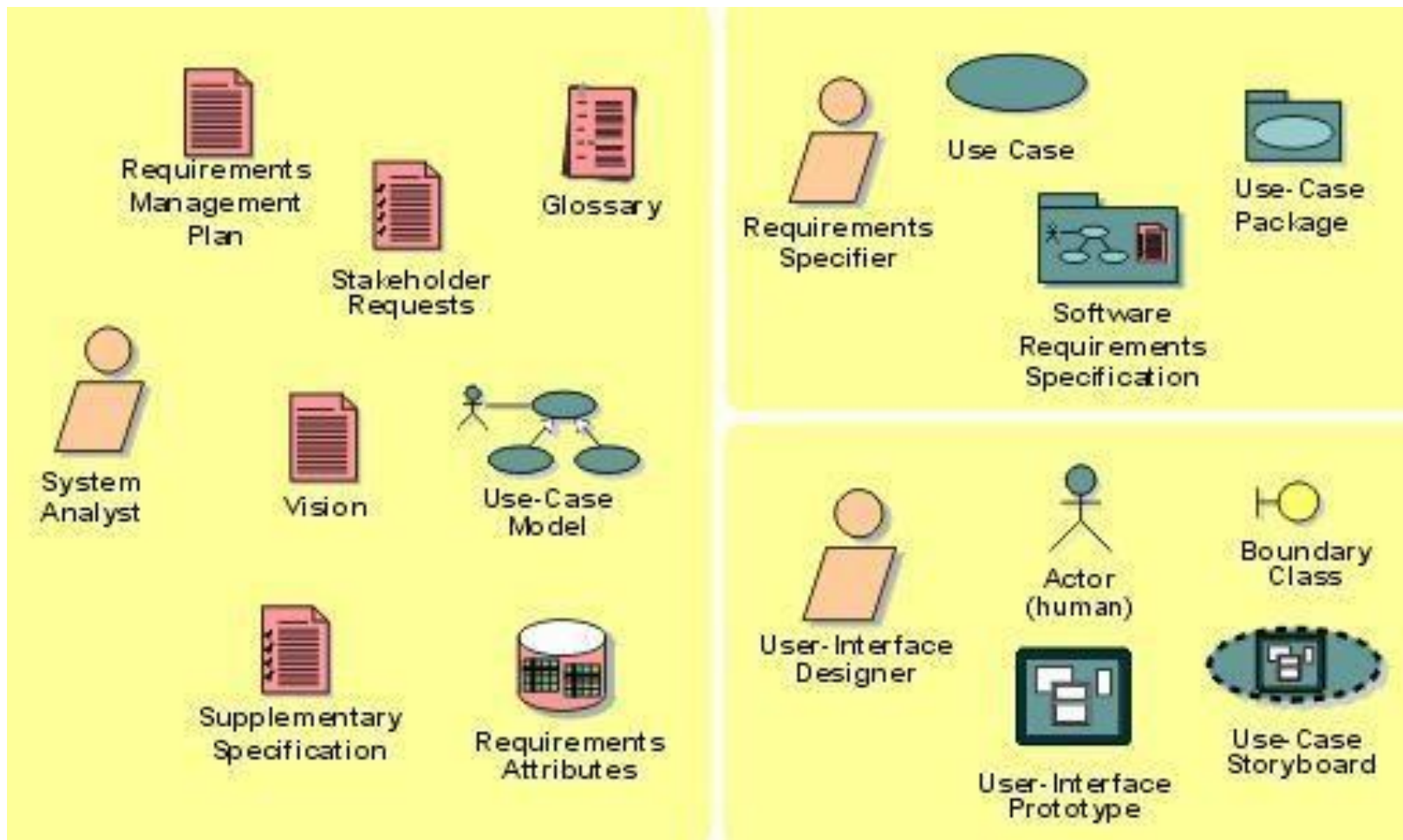
---

Os testes de aceite são os testes executados pelo cliente, baseados nos requisitos para que ele possa aprovar o sistema e avaliar se é exatamente o que ele precisava.

**Acceptance Tests**



# Principais Atores e Artefatos





- **Bibliografia**

- Paulo Filho, Wilson de Pádua. *Engenharia de Software*. LTC, 2003
- Pressman, Roger S. *Engenharia de Software*. 6ª edição. McGraw-Hill, 2006.
- Sommerville, Ian. *Engenharia de Software*. 8ª edição. Pearson Education, 2007.
- Carvalho, Ariadne M. B. Rizzoni & Chiossi, Thelma C. dos Santos. *Introdução à Engenharia de Software*. Unicamp, 2001.