

# Engenharia de Requisitos

# O Problema de Requisitos

Satisfação do usuário

=

Produto adequado

+

Máxima qualidade

+

Entrega dentro do orçamento e cronograma

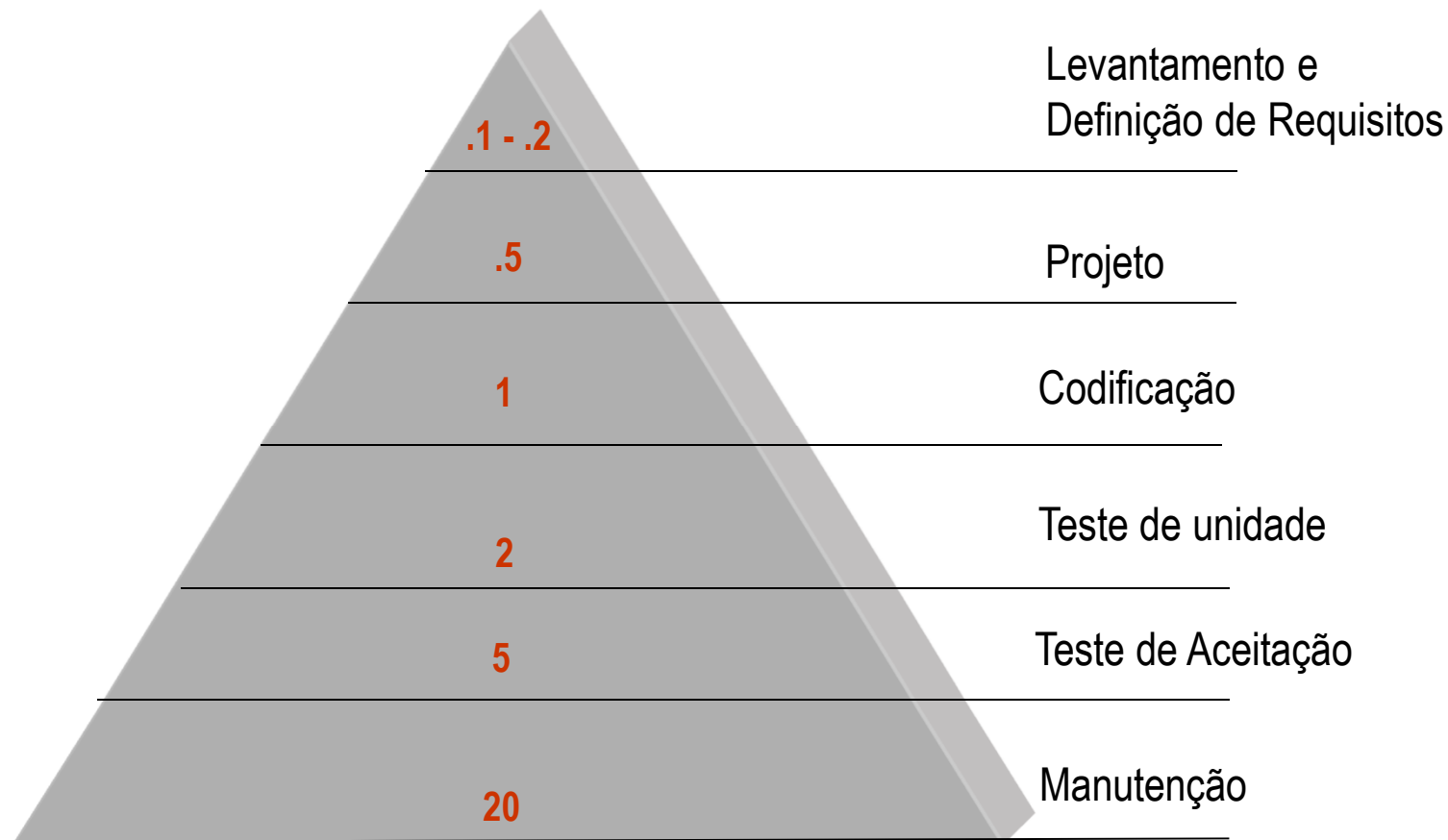
# O Problema de Requisitos

A mudança de requisitos é inevitável!

- O sucesso de um projeto depende de uma adequada Engenharia de requisitos

# O Problema de Requisitos

- O alto custo de erros nos requisitos:



# O Problema de Requisitos

- Erros de requisitos costumam ser a classe de erros mais comuns em projetos.
- Erros de requisitos costumam ser os mais caros de se consertar.

# Requisito de Software

- Definições:
  - Uma capacidade do software necessária ao usuário para resolver um problema ou atingir um objetivo.
  - Uma capacidade do software que deve ser alcançada para satisfazer um contrato, padrão, especificação ou outra documentação imposta formalmente.

# Engenharia de Requisitos – O que é?

- Abordagem sistemática para **elicit**, **organizar** e **documentar** requisitos de sistemas, e um processo que **estabelece e mantém acordos entre clientes e a equipe de projeto** no que se refere a **mudanças** nestes requisitos.

# Engenharia de Requisitos – O que é?

- Elicitar:
  - Entender os problemas, cultura e linguagem dos usuários para construir sistemas que atendam às suas necessidades.
- Organizar:
  - Centenas ou mesmo milhares de requisitos podem estar associados a um sistema.
- Documentar:
  - Registro dos requisitos e meio de comunicação entre os envolvidos no projeto através de documentos, modelos, bases de dados etc.



# Engenharia de Requisitos – O que é?

- Gerenciar:
  - Quem são os responsáveis pelo requisito #278 e quem está autorizado a modificar ou removê-lo?
  - Se o requisito #278 for modificado, que outros requisitos serão afetados?
  - Como podemos assegurar que alguém escreveu o código para atender ao requisito #278 e quais os casos de teste são necessários para verificar se o requisito foi atendido?

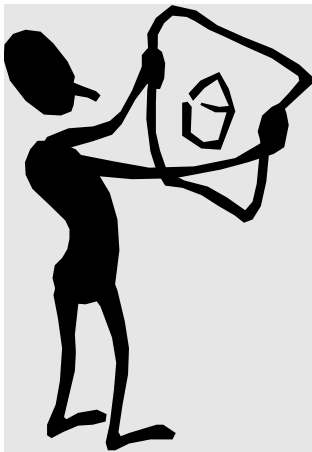
# Engenharia de Requisitos

- Identificação Domínio do Problema
- Entendimento das necessidades dos usuários
- Especificação de requisitos

# Engenharia de Requisitos

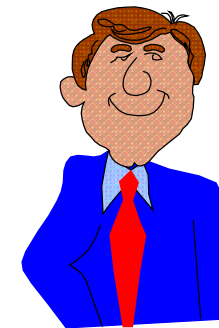
Domínio do Problema

**Conceitos, processos,  
necessidades  
objetivos, termos...**



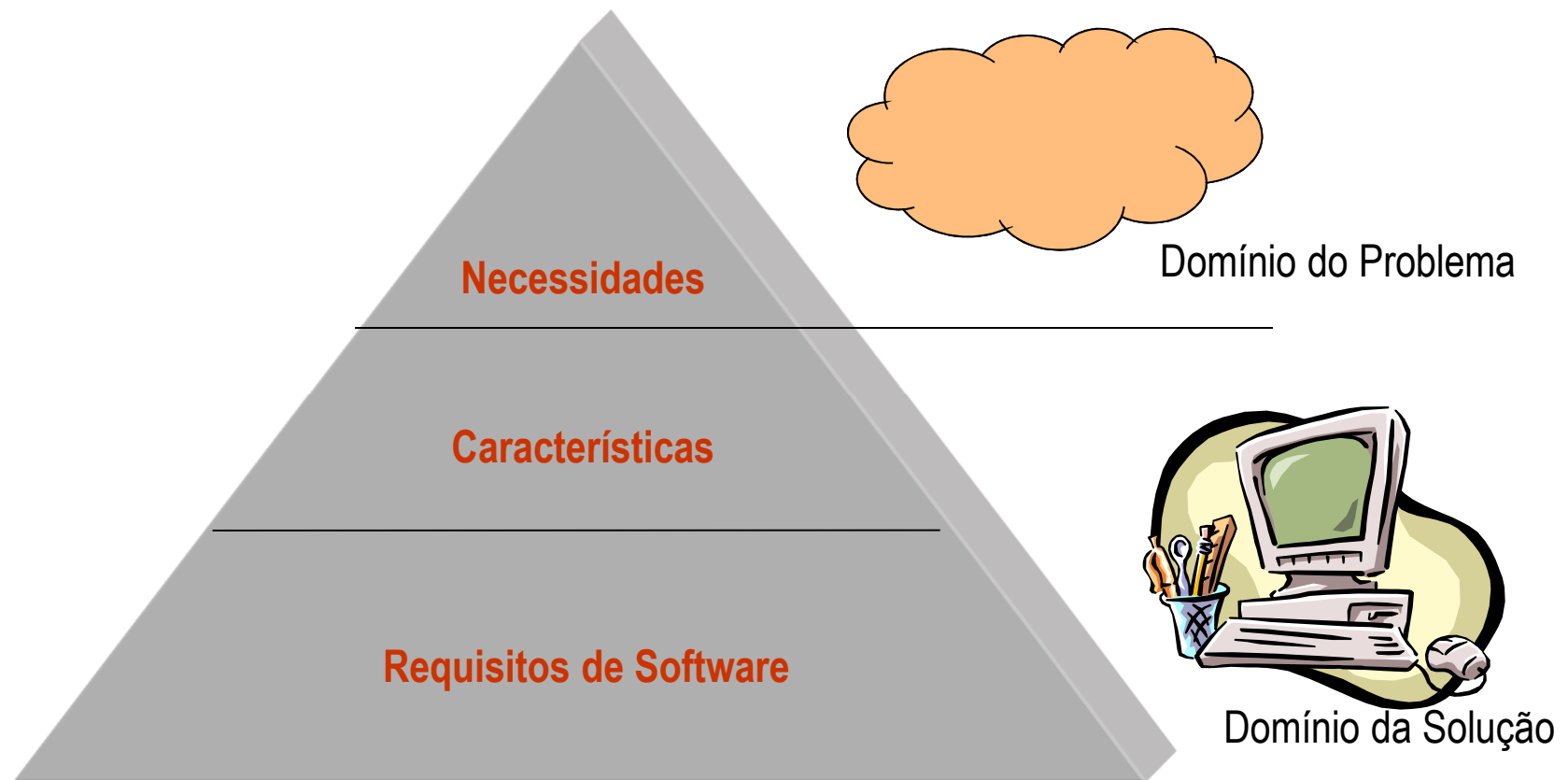
Desenvolvedor

- Torna-se *nosso* problema, compreender o problema dos usuários (de negócio/técnicos)



Usuário

# Engenharia de Requisitos

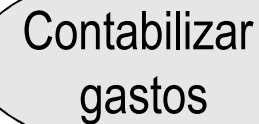


# Engenharia de Requisitos

- Características do software
  - O serviço que um sistema faz para prover ou atingir uma ou mais necessidades dos usuários
    - “O carro terá janelas automáticas.”
    - “Diagramas serão utilizados para apresentar o progresso da execução.”
    - “Os pedidos de compra serão realizados via Internet.”
    - ...

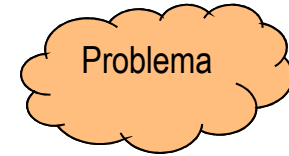
# Engenharia de Requisitos

- Requisitos do Software
  - Detalhamento das características em direção à solução (especificação).
    - Casos de uso: descreve a sequência de ações executadas pelo sistema que oferecem um resultado para o usuário.



Contabilizar  
gastos

# Análise do Problema



- Objetivo: entender o problema a ser resolvido antes de iniciar o desenvolvimento da aplicação.
- Atividades:
  - Atingir acordo sobre a definição do problema
  - Entender as raízes do problema.
  - Identificar usuários e pessoas envolvidas cujos julgamentos determinam o sucesso ou fracasso do sistema.
  - Determinar as fronteiras do sistema.
  - Compreender as restrições impostas à solução.

# Análise do Problema

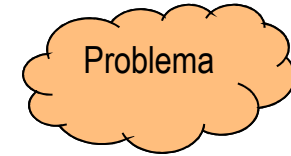


- Atividade: Atingir acordo sobre a definição do problema
  - Consiste em definir o problema e obter acordo entre as partes envolvidas.
  - Elementos de Definição do Problema:

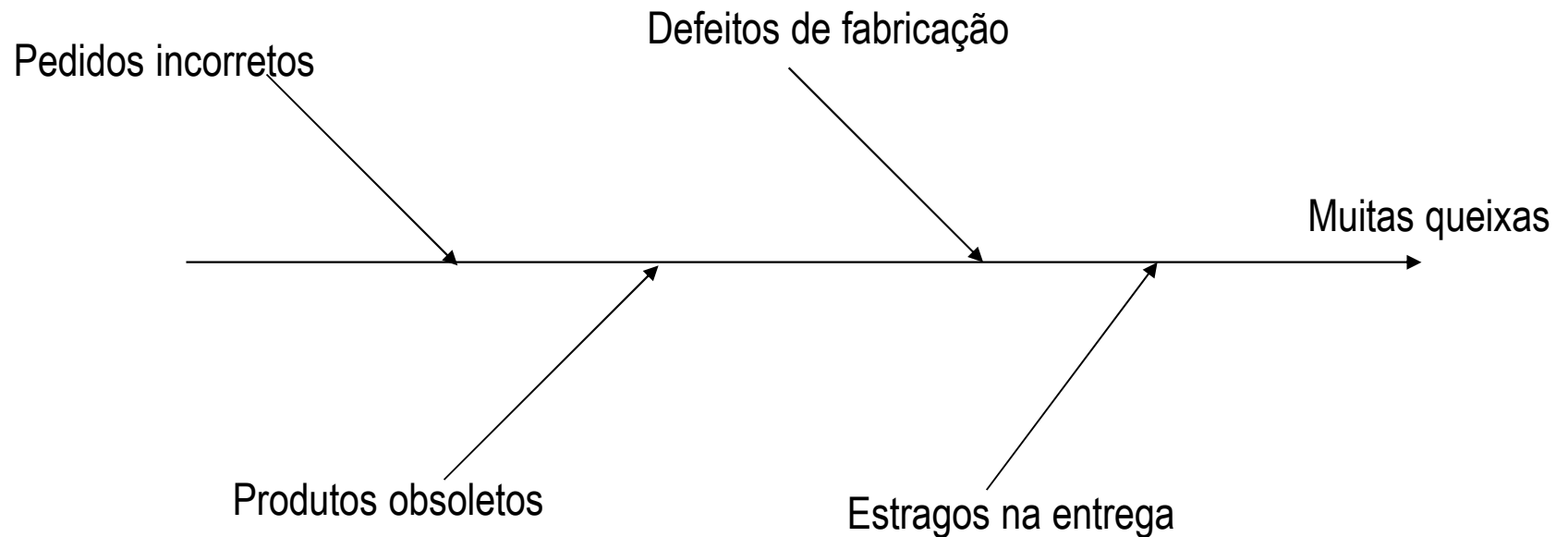
<b>O problema afeta e resulta em</b>	<ul style="list-style-type: none"><li>- descrição do problema</li><li>- identificação dos elementos afetados</li><li>- descrição do impacto deste problema sobre os elementos afetados e atividades de negócio.</li></ul>
<b>Benefícios</b>	<ul style="list-style-type: none"><li>- indicação da solução proposta e uma lista de benefícios chave.</li></ul>



# Análise do Problema



- Atividade: Entender as raízes do problema.
  - Consiste em identificar as causas do problema em questão.
  - Uso de diagramas de causa e efeito (“espinha de peixe”):



# Análise do Problema



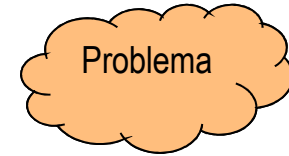
**O problema  
afeta  
e resulta em**

- pedidos de compra incorretos
- pessoal de vendas, clientes, manufatura, entregas, SAC
- muitas queixas, custos para reposição, insatisfação dos clientes, diminuição de renda.

**Benefícios**

- um novo sistema para endereçar este problema incluiria:
  - \* verificação dos pedidos no momento de sua realização
  - \* melhoria do relatório de vendas para a Engenharia
  - \* aumento de rendimentos

# Análise do Problema



- Atividade: Identificação de usuários e envolvidos.
  - Quem são os usuários do sistema?
  - Quem é o cliente do sistema?
  - Quem mais será afetado pelas saídas oferecidas pelo sistema?
  - Quem avaliará e aprovará o sistema quando entregue?
  - Quem irá manter o sistema?
  - Alguém mais?

# Análise do Problema



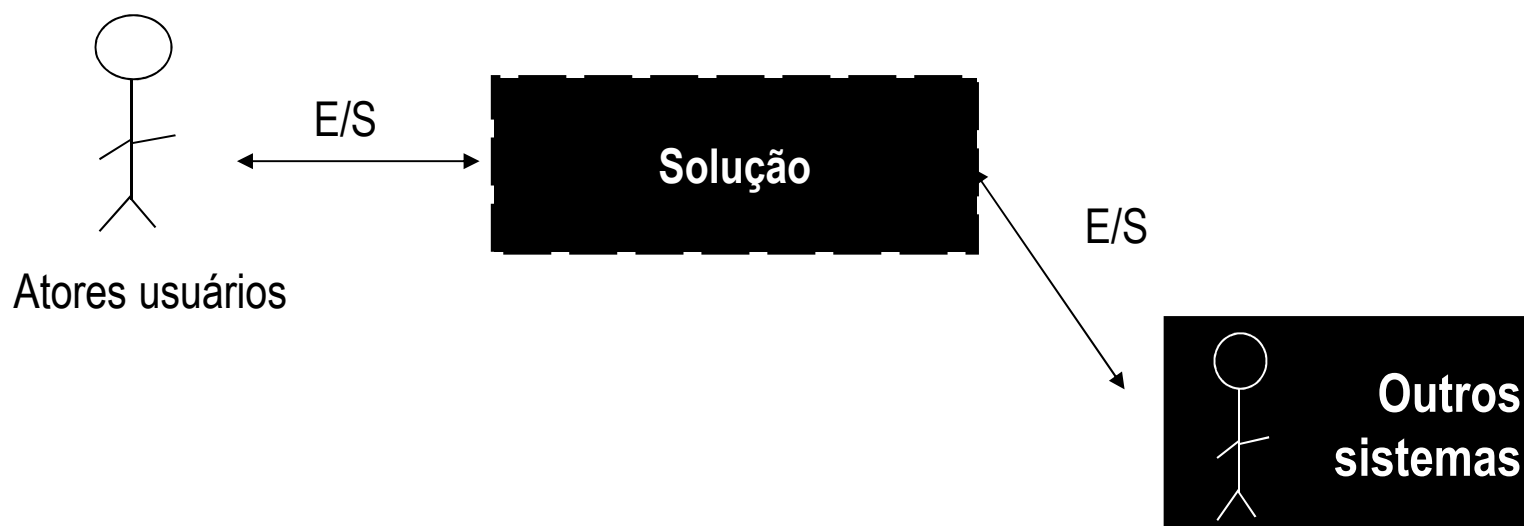
☀ Atividade: **Identificação de usuários e envolvidos.**

Usuários	Envolvidos
Atendentes	Equipe de desenvolvimento
Supervisores de venda	Gerente de finanças
Controle de produção	Gerente de Produção
Faturamento	

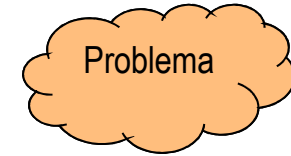
# Análise do Problema



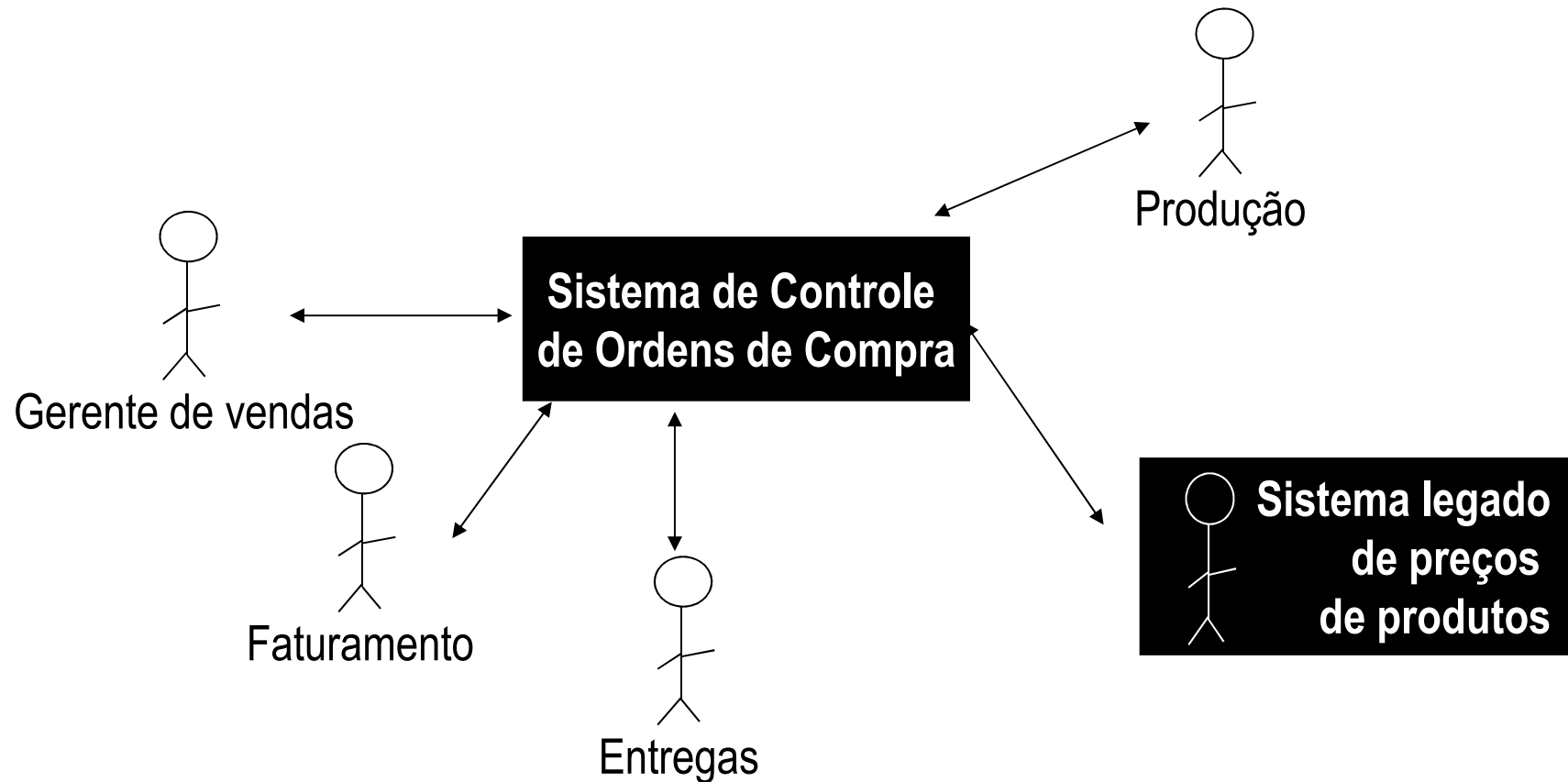
- Atividade: Determinar as fronteiras do sistema.



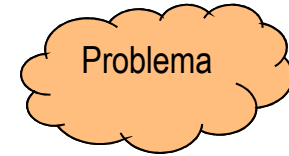
# Análise do Problema



- Atividade: Determinar as fronteiras do sistema.



# Análise do Problema



- Atividade: Identificar e compreender as restrições impostas à solução.
  - Econômicas
  - Políticas
  - Técnicas
  - Sistema
  - Ambientais
  - Cronogramas e recursos

# Análise do Problema

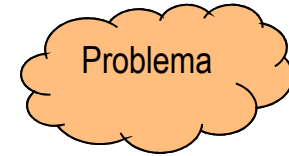


- Atividade: Identificar e compreender as restrições impostas à solução.

Fonte	Restrição	Razão
Operacional	Uma cópia exata da ordem de compra deve ser mantida na base de dados legada por um ano	O risco de perda de dados é grande. Será necessário rodar os sistemas em paralelo por um tempo.
Sistema	O espaço no servidor para a aplicação deve ser de 20 M.	Dispomos de pouco espaço disponível no servidor.
Pessoal	Recursos fixos; sem contratação.	Dificuldades de orçamento.
Tecnológica	Nova metodologia (OO) sendo utilizada.	Acreditamos que esta metodologia trará ganhos em produtividade.

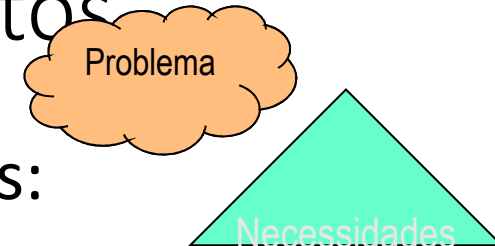


# Análise do Problema



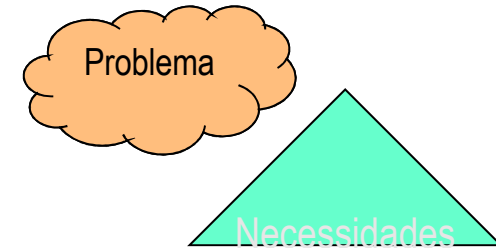
- Outras questões:
  - Modelagem de negócios
  - Modelagem de processos de negócio

# Entendimento das Necessidades dos Usuários – Elicitação de Requisitos



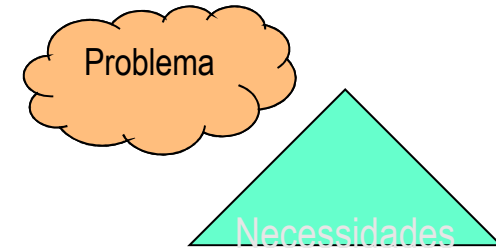
- Dificuldades da Elicitação de Requisitos:
  - “Sim, mas...”
    - Por melhor que seja a elicitação, usuários podem ter reações diferentes ao ver o sistema rodando...
  - “Quantos fósseis ainda não foram descobertos?”
    - Não há como garantir que todos os requisitos foram levantados.
  - Comunicação entre “usuário e desenvolvedor”
    - Diferentes mundos, linguagens, especializações, motivações e objetivos...

# Elicitação de Requisitos



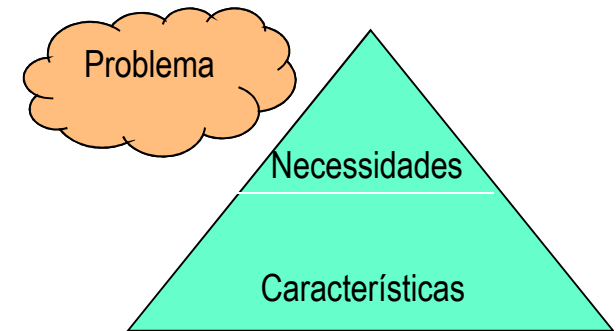
- Problema: usuários nem sempre sabem o que querem, ou sabem o que querem mas não sabem articular.
  - Solução: reconheça o usuário como um especialista em seu domínio. Tente técnicas diferentes de elicitación de requisitos.
- Problema: Usuários pensam que sabem o que querem até que os desenvolvedores entreguem a eles o que eles disseram que queriam
  - Solução: use técnicas diferentes de elicitación
- Problema: Analistas pensam que entendem os problemas melhor que seus usuários.
  - Solução: coloque o analista no lugar do usuário.

# Elicitação de Requisitos



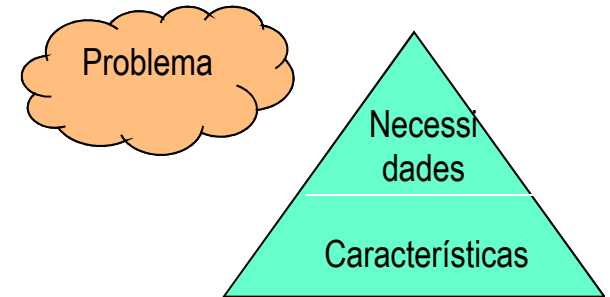
- Técnicas:
  - Questionários e entrevistas
  - Seminários de requisitos
  - Brainstorm
  - *Storyboarding*
  - Casos de Uso
  - *Role playing*
  - Prototipagem

# Elicitação de Requisitos



- Necessidades do usuário X Características do sistema
  - “Eu preciso de uma janela com o formulário para preenchimento da ordem de compra.”
  - “Eu preciso de um carro com direção hidráulica.”
- Estimativa para quantidade de características a serem levantadas: entre 25 e 99 características.

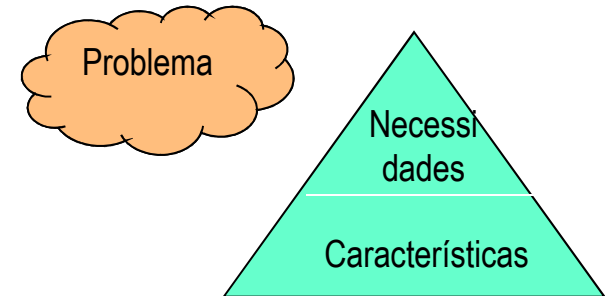
# O documento da “Visão do Sistema”



- Descreve a aplicação em termos gerais: usuários, características do sistema, alvo de mercado etc.
- Descreve, em alto nível, tanto o problema como a solução.
- Todo projeto necessita de um documento de visão do sistema.

# O documento da “Visão do Sistema”

- Exemplo:



## 1. Introdução

### 1.1. Propósito do documento

### 1.2. Referências

## 2. Descrição de usuários

### 2.1. Mercado

### 2.2. Perfis de usuário

### 2.3. Ambiente do usuário

### 2.4. Necessidades básicas dos usuários

### 2.5. Alternativas e Competição

## 3. Visão Geral do Produto

### 3.1. Perspectiva do Produto

### 3.2. Resumo dos Benefícios

## 4. Características do produto

## 5. Principais casos de uso

## 6. Outros requisitos

### 6.1. Padrões aplicáveis ao produto

### 6.2. Requisitos de sistema

### 6.3. Licenças e instalação

### 6.4. Requisitos de desempenho

## 7. Restrições

## 8. Requisitos de documentação

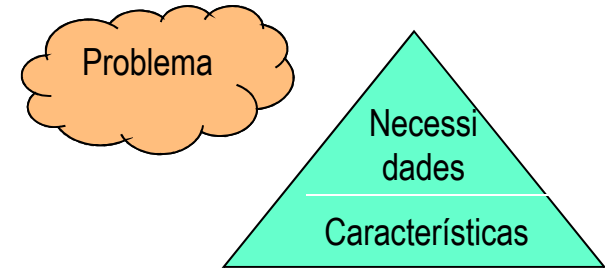
### 8.1. Manual do usuário

### 8.2. Ajuda *on-line*

### 8.3. Guias de instalação e configuração

## 9. Glossário

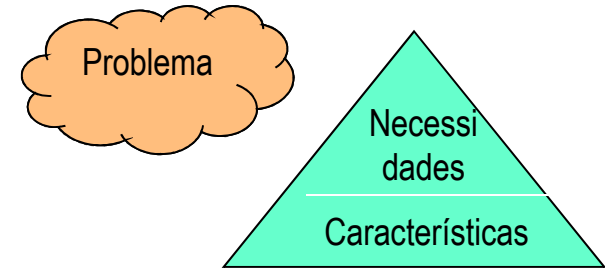
# Engenharia de Escopo do Projeto



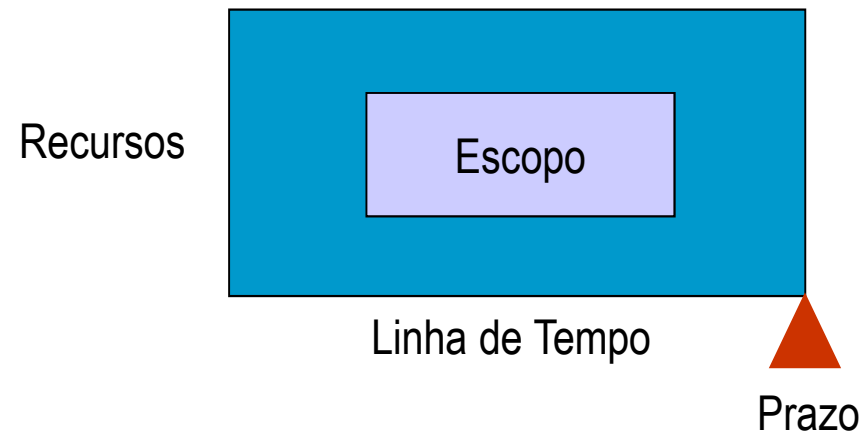
- O Escopo do Projeto é uma combinação de:
  - Funcionalidade do produto,
  - Recursos para o projeto e
  - Tempo disponível.



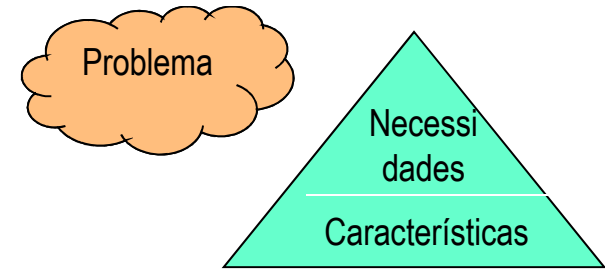
# Engenharia de Escopo do Projeto



- O Escopo do Projeto é uma combinação de:
  - Funcionalidade do produto,
  - Recursos para o projeto e
  - Tempo disponível.

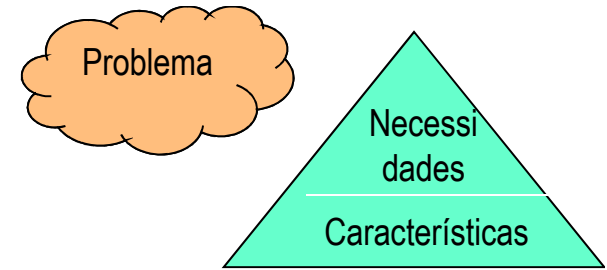


# Estabelecendo o Escopo do Projeto

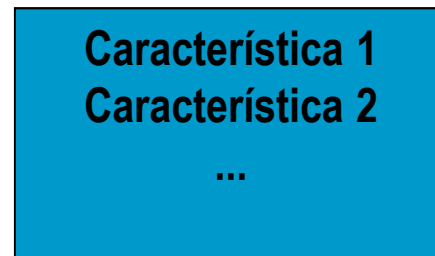


- Estabelecer uma *baseline* de requisitos
  - Conjunto de características a serem oferecidas em uma dada versão do sistema
- Estabelecer o nível de esforço requerido para cada característica identificada na *baseline*
- Estimar riscos para cada característica
  - Probabilidade que sua implementação cause impactos adversos no cronograma ou orçamento

# Baseline de Requisitos

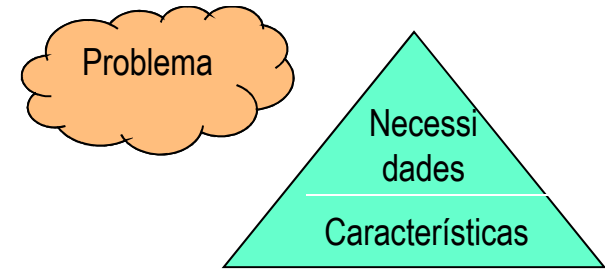


- Uma lista de características do sistema a serem entregues em uma versão específica do sistema.



*Baseline para*  
Versão 1.0

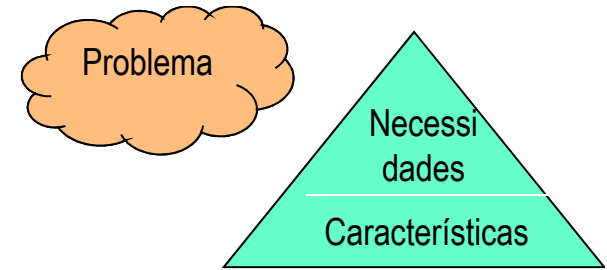
- A *baseline* deve ser ao menos “aceitável” pelo cliente e
- ter uma razoável probabilidade de sucesso, do ponto de vista da equipe.



# *Baseline* de Requisitos

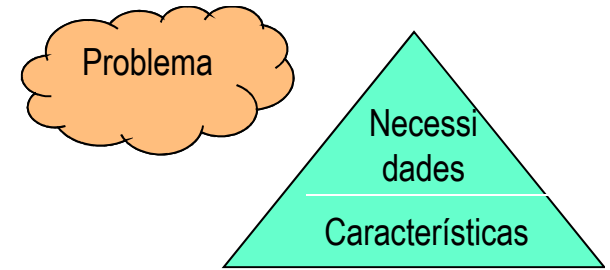
- Exemplo:
  - Característica 1: suporte por banco de dados relacional
  - Característica 2: Controle de segurança para acesso multiusuário
  - Característica 3: Habilidade de “clonar” um projeto
  - Característica 4: Portabilidade para nova versão de sistema operacional
  - Característica 5: Assitente de criação de novos projetos
  - Característica 6: Importação de dados externos
  - Característica 7: Implementação de “dicas” de ferramentas
  - Característica 8: Integração com subsistema de Engenharia de versões
  - ...

# Priorização



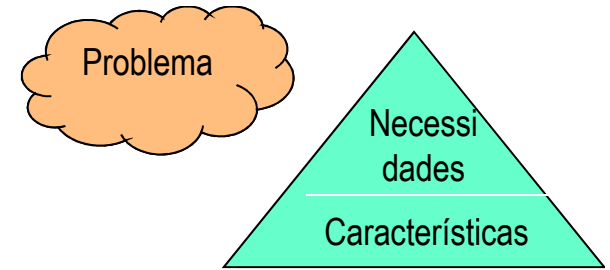
- Críticas:
  - Característica 1: suporte por banco de dados relacional
  - Característica 4: Portabilidade para nova versão de sistema operacional
  - Característica 6: Importação de dados externos
- Importantes:
  - Característica 3: Habilidade de “clonar” um projeto
  - Característica 2: Controle de segurança para acesso multiusuário
  - Característica 5: Assistente de criação de novos projetos
- Úteis:
  - Característica 7: Implementação de “dicas” de ferramentas
  - Característica 8: Integração com subsistema de Engenharia de versões

# Priorização



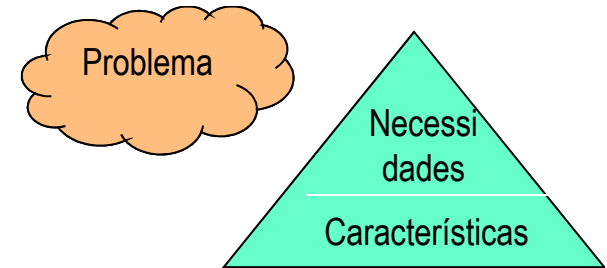
- Idealmente, a priorização deve ser feita:
  - Pelos usuários/clientes
  - Sem considerar questões técnicas
- Estima-se que pelo menos 1/3 dos itens na lista de características devem ser críticos

# Esforço



- Desafios:
  - Não há requisitos detalhados para se aplicar técnicas de estimativas formais.
  - Uma avaliação de esforço é a única maneira de determinar se o escopo do sistema poderá ser atendido.
- Solução:
  - Estimativa “alto nível” do esforço necessário para desenvolver cada característica.
  - Baseado na experiência da equipe!

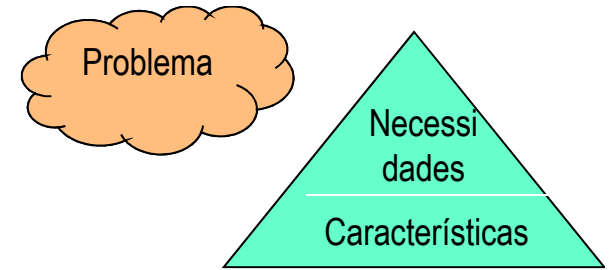
# Esforço



Características:	Prioridade:	Esforço
Característica 1: suporte por banco de dados relacional	Crítica	Médio
Característica 4: Portabilidade para nova versão de sistema operacional	Crítica	Alto
Característica 6: Importação de dados externos	Crítica	Baixo
Característica 3: Habilidade de “clonar” um projeto	Importante	Alto
Característica 2: Controle de segurança para acesso multiusuário	Importante	Baixo
Característica 5: Assistente de criação de novos projetos	Importante	Baixo
Característica 7: Implementação de “dicas” de ferramentas	Útil	Baixo
Característica 8: Integração com subsistema de Engenharia de versões	Útil	Alto

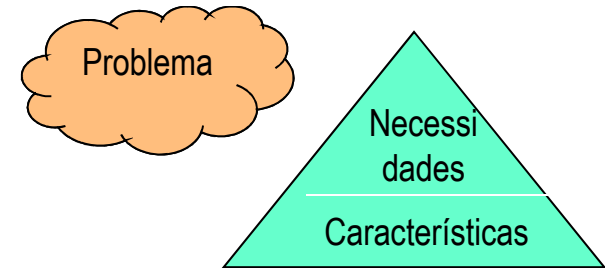


# Adicionando o Risco

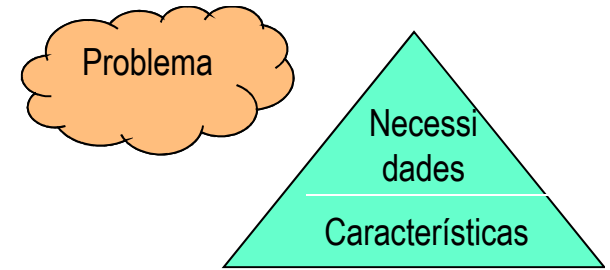


- Probabilidade que a implementação de uma característica cause impactos no cronograma ou orçamento do projeto

# Adicionando Risco



Características:	Prioridade: Esforço		Risco
Característica 1: suporte por banco de dados relacional	Crítica	Médio	Baixo
Característica 4: Portabilidade para nova versão de sistema operacional	Crítica	Alto	Médio
Característica 6: Importação de dados externos	Crítica	Baixo	Alto
Característica 3: Habilidade de “clonar” um projeto	Importante	Alto	Médio
Característica 2: Controle de segurança para acesso multiusuário	Importante	Baixo	Alto
Característica 5: Assitente de criação de novos projetos	Importante	Baixo	Baixo
Característica 7: Implementação de “dicas” de ferramentas	Útil	Baixo	Alto
Característica 8: Integração com subsistema de Engenharia de versões	Útil	Alto	Baixo



# Reduzindo Escopo

## ☀ Uma primeira estimativa:

### Atributos:

### Considere:

*Prioridade:* Crítica  
*Esforço:* Alto  
*Risco:* Alto

Alarme! Estabeleça uma estratégia de diminuição de risco; aloque recursos imediatamente.

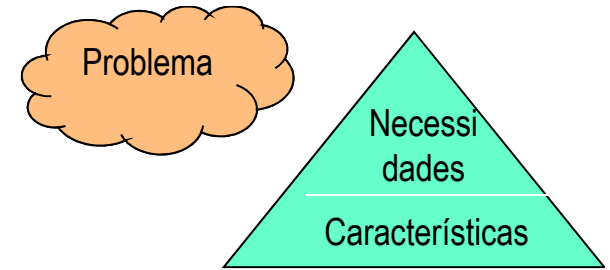
*Prioridade:* Crítica  
*Esforço:* Alto  
*Risco:* Baixo

Aloque recursos imediatamente.

*Prioridade:* Crítica  
*Esforço:* Baixo  
*Risco:* Baixo

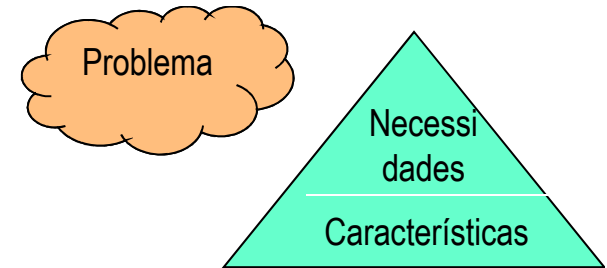
Aloque algum recurso, se houver, ou então aguarde.

# Reduzindo Escopo

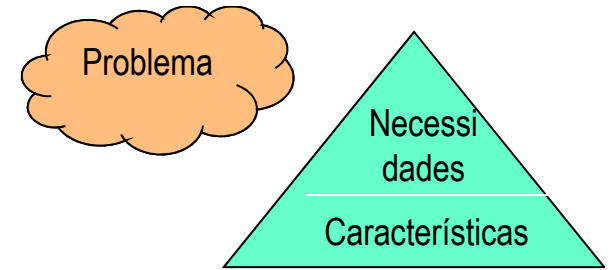


- Podemos fazer todos os itens críticos?
- Sim.
  - Então estamos prontos para detalhar os requisitos e planejar o projeto.
  - Inclua alguns itens importantes/úteis na *baseline*, se possível for.
  - Inclua novos itens importantes/úteis, de acordo com o progresso do projeto.

# Reduzindo Escopo



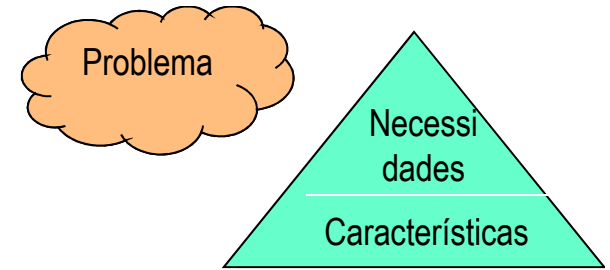
Características:	Prioridade:	Esforço	Risco	
Característica 1: suporte por banco de dados relacional	Crítica	Médio	Baixo	
Característica 4: Portabilidade para nova versão de sistema operacional	Crítica	Alto	Médio	
Característica 6: Importação de dados externos	Crítica		Baixo	Alto
Característica 3: Habilidade de “clonar” um projeto	Importante		Alto	Médio
<b>Baseline:</b> características até aqui estabelecem o compromisso do projeto				
Característica 2: Controle de segurança para acesso multiusuário	Importante	Baixo	Alto	
Característica 5: Assistente de criação de novos projetos	Importante		Baixo	Baixo
Característica 7: Implementação de “dicas” de ferramentas	Útil		Baixo	Alto
Característica 8: Integração com subsistema de Engenharia de versões	Útil		Alto	Baixo



# Reduzindo Escopo

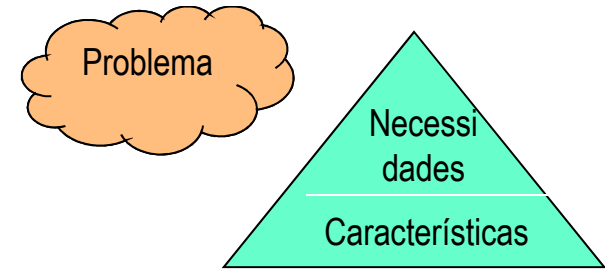
- Podemos fazer todos os itens críticos?
- Não.
  - Então precisamos priorizar novamente o que é crítico por aqui...

# Gerenciando o Cliente



- A Engenharia do escopo deixa evidente um relacionamento “adversário” entre cliente e desenvolvedores.
- Desafio:
  - Se associar ao cliente na Engenharia dos **seus** requisitos e de **seu** escopo de projeto para garantir qualidade e prazos.

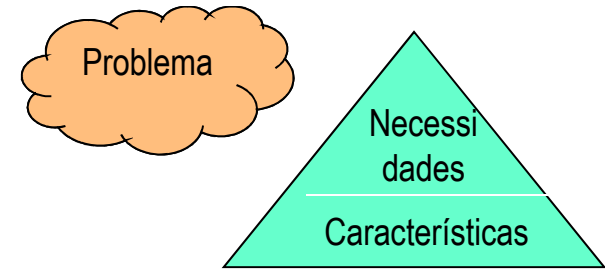
# Gerenciando o Cliente



- Comunicação de resultados
  - Necessidade de se estabelecer compromissos com o cliente
  - O cliente deve saber e negociar a redução de escopo
  - Ao longo do projeto podem ocorrer “más notícias” mas, o ideal é que elas tenham uma justificativa.
- Nada é pior do que perder a credibilidade.

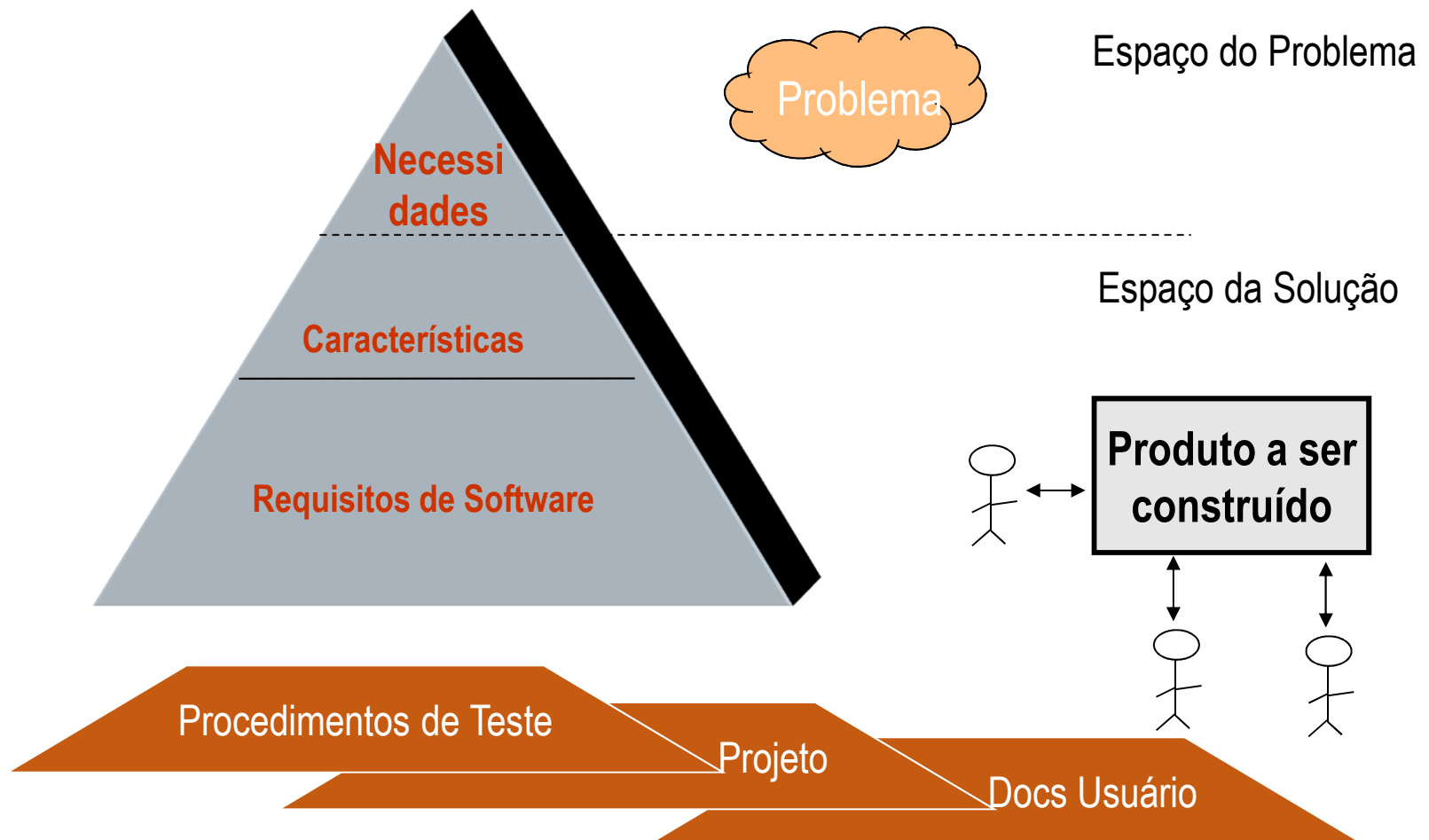


# Gerenciando o Cliente

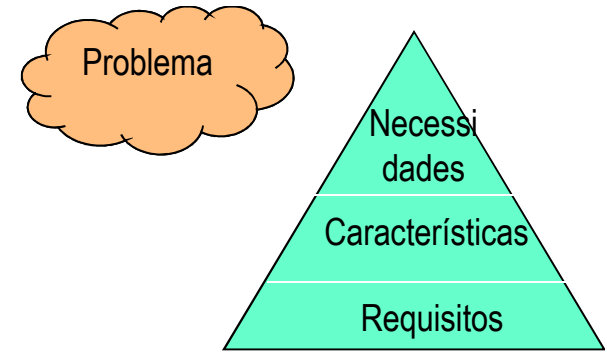


- Negociação
  - Capacidade necessária e de treinamento possível
- Recomendações:
  - Voe alto mas não de forma irracional
  - Separe pessoas de problemas
  - Focalize interesses e não posições
  - Invente opções para ganhos mútuos
  - Aplica critérios objetivos
- Princípio:
  - Prometer menos do que se pode cumprir e
  - Cumprir mais do que se prometeu.

# Especificação de Requisitos

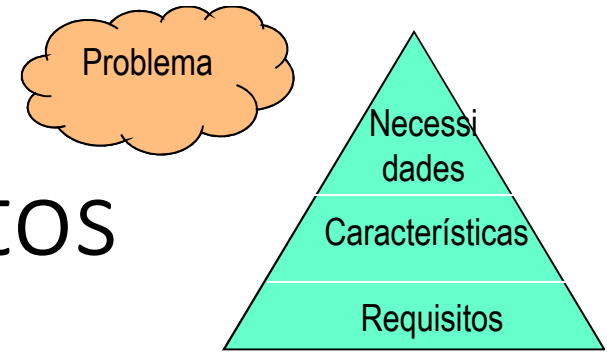


# Especificação de Requisitos

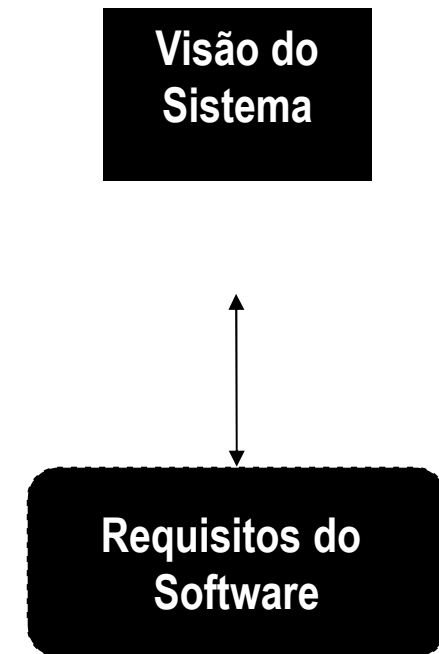


- Requisitos determinados a partir:
  - **Entradas:** não somente o conteúdo mas também os detalhes do dispositivo de entrada.
  - **Saídas:** descrição dos dispositivos de saída a serem suportados bem como o protocolo e formato de saída.
  - **Funções:** o mapeamento entre entradas e saídas e suas combinações.
  - **Atributos do sistema:** requisitos não comportamentais, como: confiabilidade, manutenibilidade, disponibilidade etc
  - **Atributos do ambiente:** requisitos não comportamentais adicionais, como: habilidade de operação sobre restrições , carga e compatibilidade com sistemas operacionais.

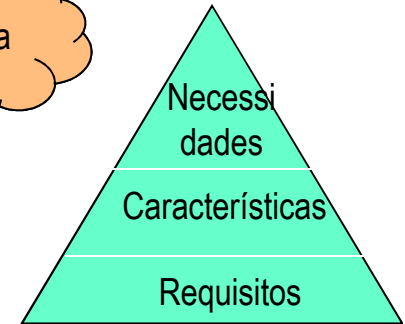
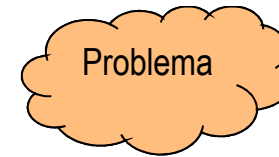
# Especificação de Requisitos



- Mapeamento de Características para Requisitos
  - Características:
    - Descrições simples de um comportamento desejado e útil do sistema
    - Auxiliam na comunicação com clientes/usuários
  - Requisitos:
    - Expressam as características em termos mais detalhados, em um nível mais próximo de sua implementação.
    - Devem ser suficientemente específicos para serem “testados”



# Especificação de Requisitos



- Exemplo:

## Visão do Sistema

**Característica 63:** O sistema de controle de defeitos deve oferecer informação para ajudar o usuário a compreender o status do projeto.

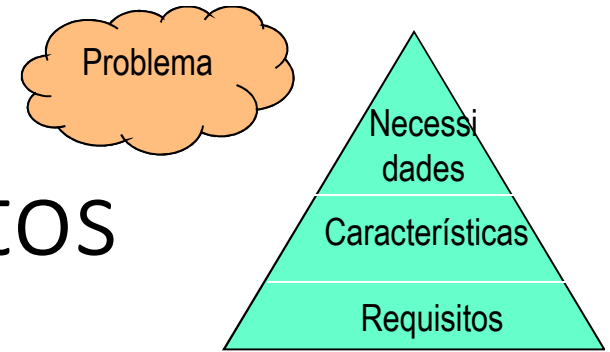
## Especificação de Requisitos

**RS63.1:** Informação será apresentada em relatório de histograma mostrando o tempo no eixo x e o número de defeitos encontrados no eixo y.

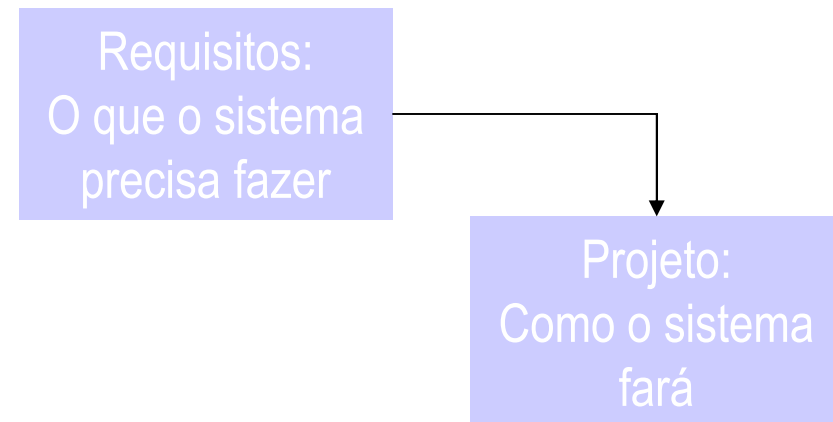
**RS63.2:** O usuário pode entrar com o período em unidades: dias, semanas ou meses.

**RS63.3:** Um exemplo do relatório pode ser visto na figura x.

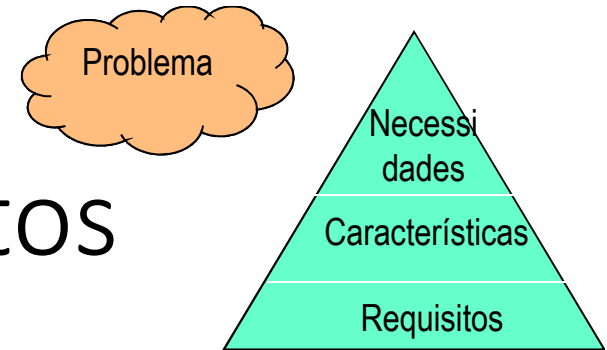
# Especificação de Requisitos



- Requisitos não incluem informações de projeto – custos, prazos etc - (deixe isto para o plano de projeto)
- Requisitos também não devem incluir decisões de projeto – linguagem, arquitetura, estratégia de armazenamento etc

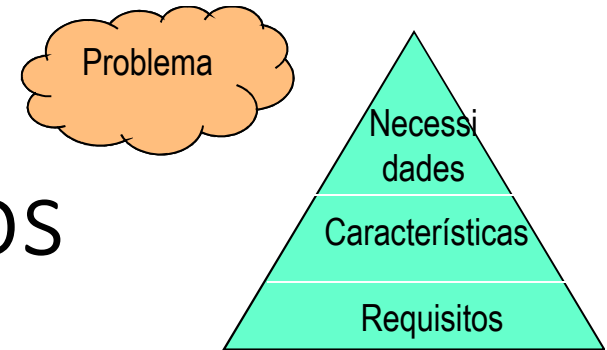


# Especificação de Requisitos



- Restrições:
  - Restrições no projeto do sistema ou no processo de seu desenvolvimento que não afetam seu comportamento externo mas que devem ser atendidas para cumprir obrigações técnicas de negócio ou contratuais.
- Exs:
  - Ambiente de desenvolvimento/operação: O sistema tem que ser desenvolvido em Visual Basic.
  - Compatibilidade com sistemas existentes: O sistema tem que rodar em plataformas existentes e novas plataformas.
  - Aplicação de padrões: Use a biblioteca de class da Biblioteca de Desenvolvimento 99-724 disponível no servidor de TI corporativo.

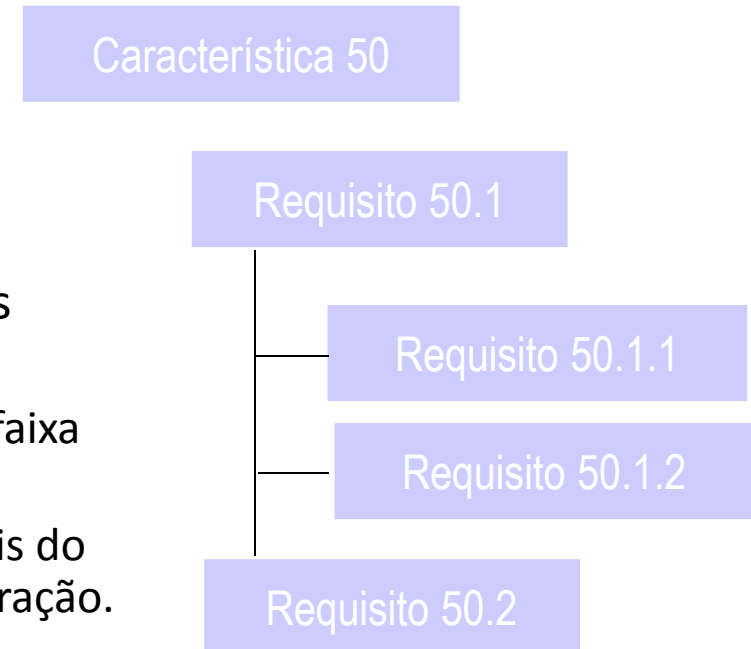
# Organização de Requisitos



- Hierarquia de requisitos

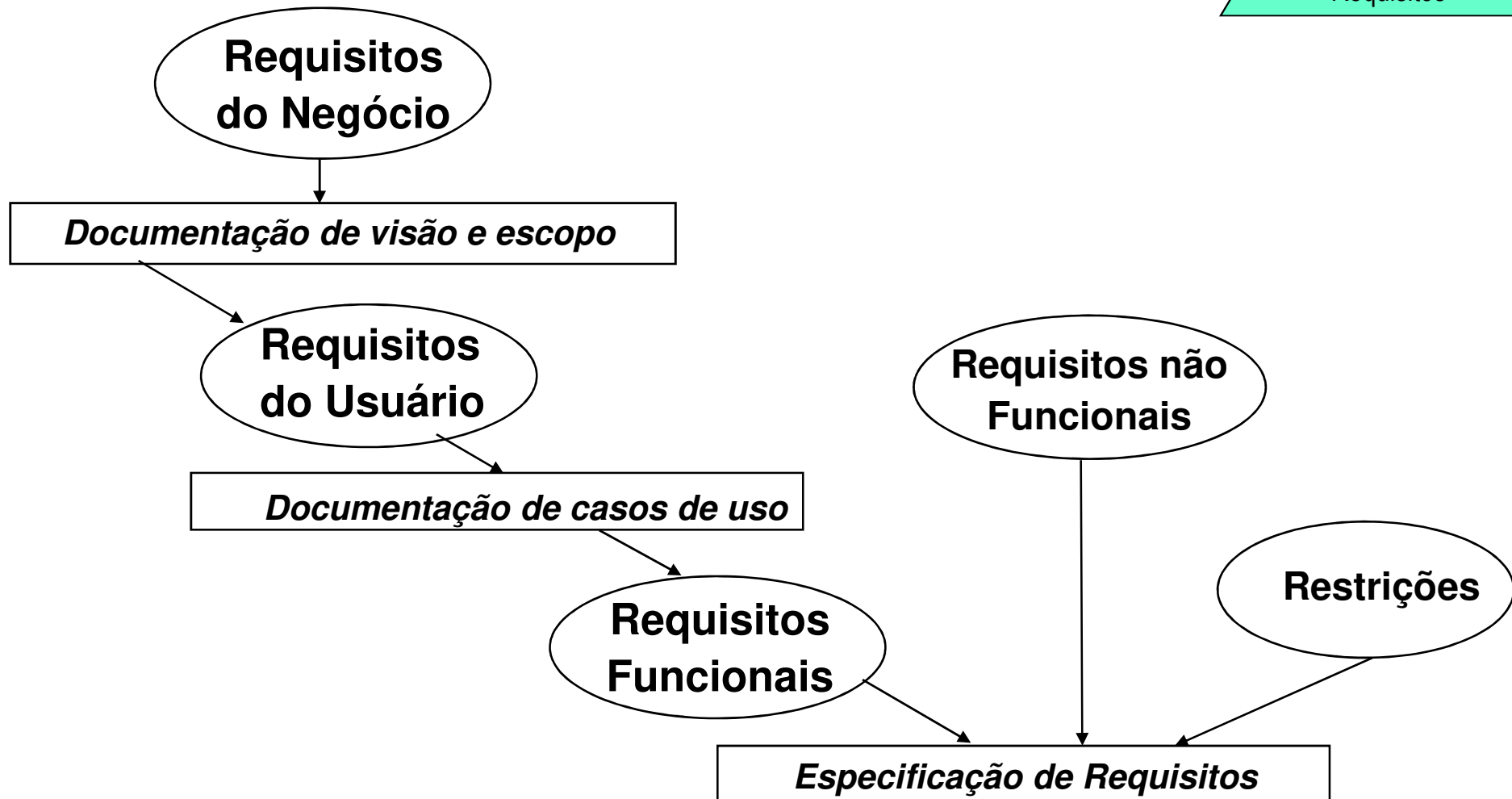
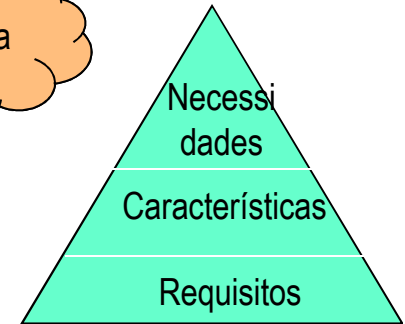
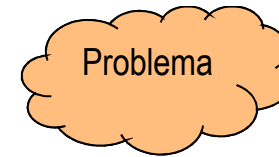
- Ex:

- O dispositivo deve operar dentro das normas nacionais de voltagem
  - O dispositivo deve operar em uma faixa de voltagem entre x-y volts AC.
  - O dispositivo deve requerer não mais do que x AC amperes para correta operação.
  - ...

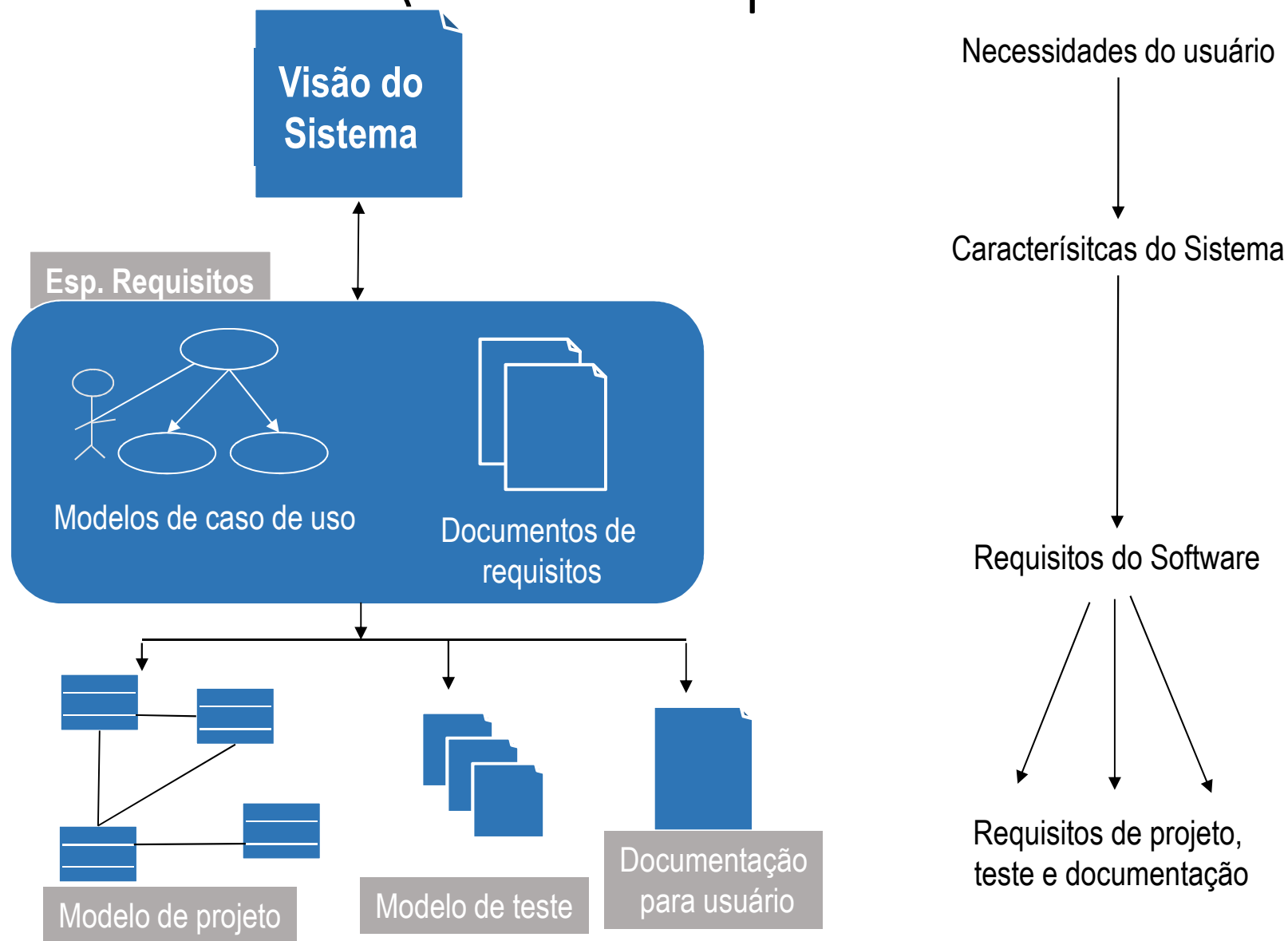




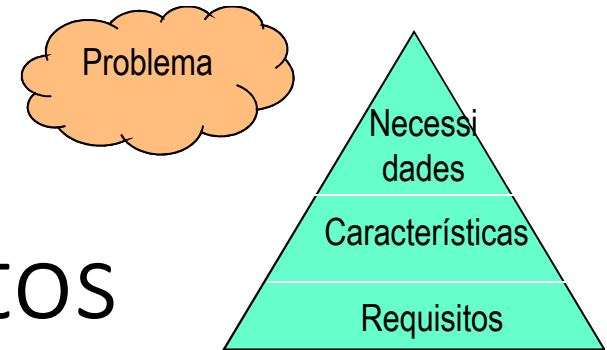
# Especificação de Requisitos



# Documentação de Requisitos



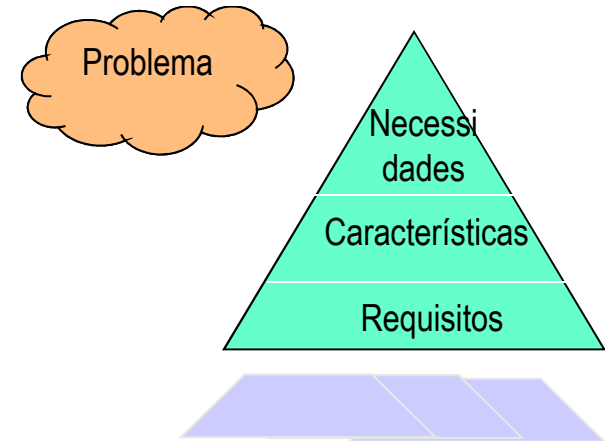
# Modelo de Especificação de Requisitos



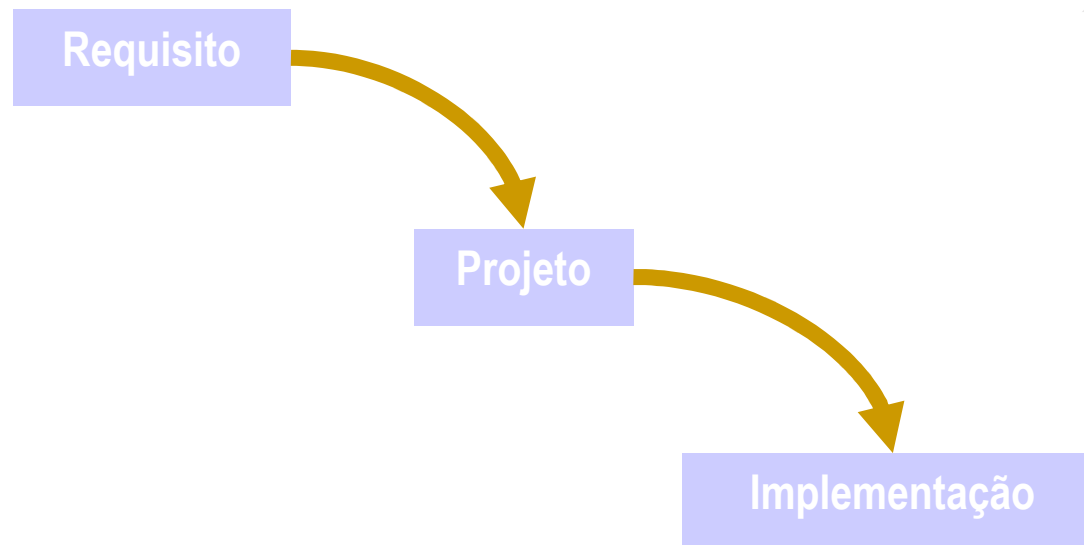
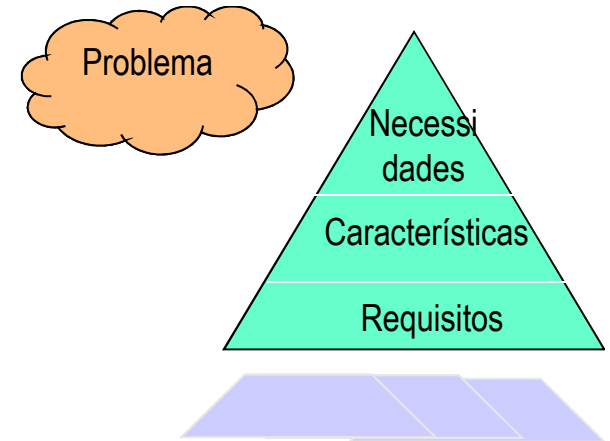
- |  |  |
|--|--|
| 1. Introdução  | 8. Interfaces                            |
| 1.1. Propósito do documento                                    | 8.1. Interfaces com usuário              |
| 1.2. Referências   | 8.2. Interfaces com hardware             |
|  | 8.3. Interfaces com outros software      |
| 2. Lista de Casos de Uso                                       | 8.4. Interfaces de comunicação           |
|  |  |
| 3. Lista de Atores   | 9. Requisitos de licenças                |
|  |  |
| 4. Requisitos  | 10. Legalização, Direitos Autorais etc   |
| 4.1. Requisitos Funcionais                                     |  |
| 4.2. Requisitos não Funcionais                                 | 11. Padrões aplicáveis                   |
|  |  |
| 5. Requisitos de Documentação para Usuário e Sistemas de Ajuda | Glossário                                |
|  |  |
| 6. Restrições de Projeto                                       | Apêndice: Especificações de Casos de Uso |
|  |  |
| 7. Produtos comprados  |  |

# Engenharia do Produto

- Definição detalhada de requisitos
- Uso de técnicas específicas:
  - Modelagem OO
  - Modelagem Funcional
  - Modelagem de Dados
  - ...

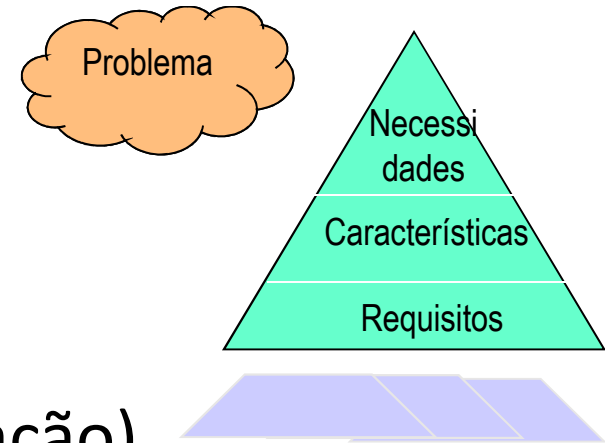


# Engenharia do Produto



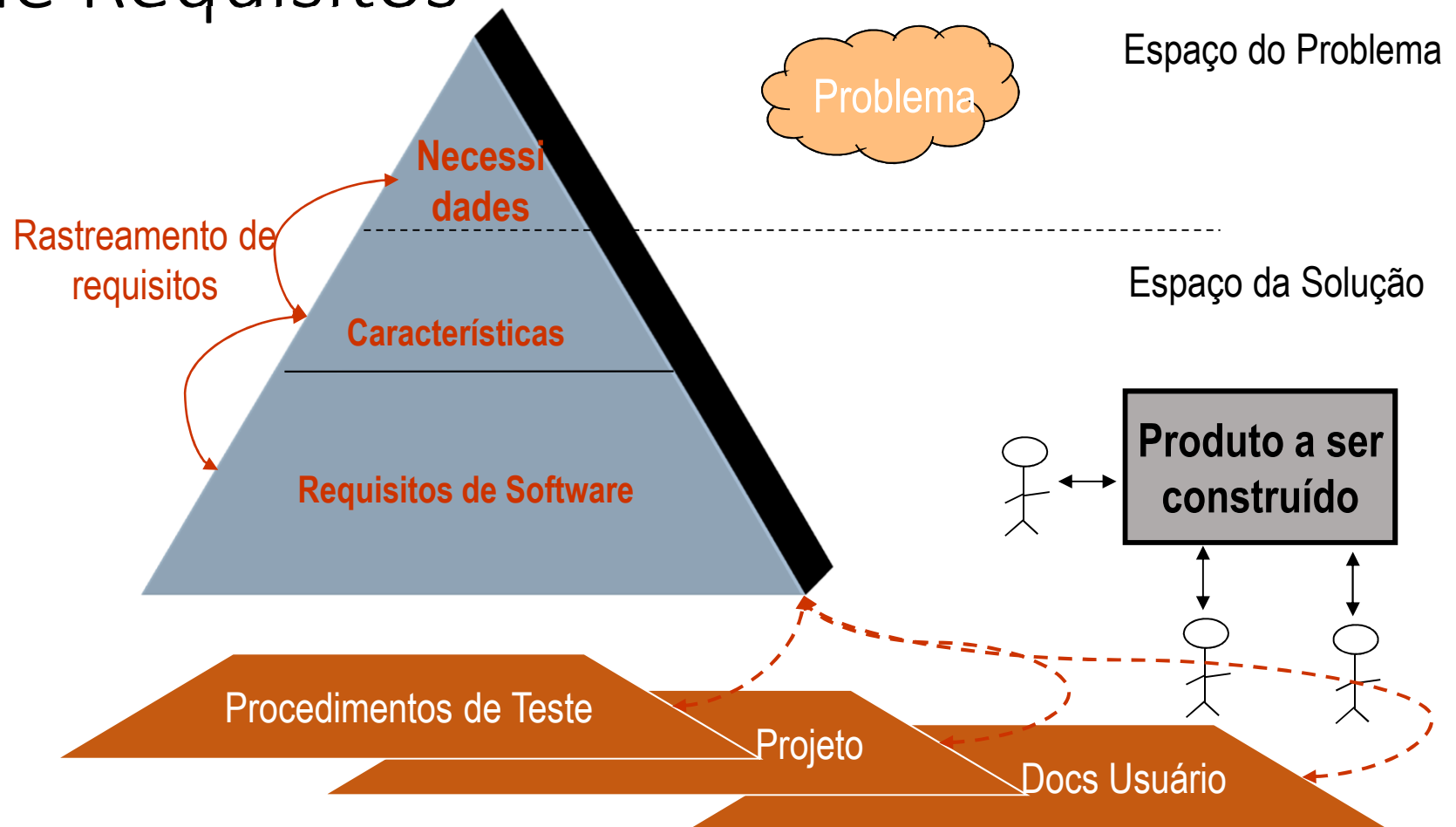
Todos os itens de projeto e implementação partem de um requisito inicial!

# Engenharia do Produto

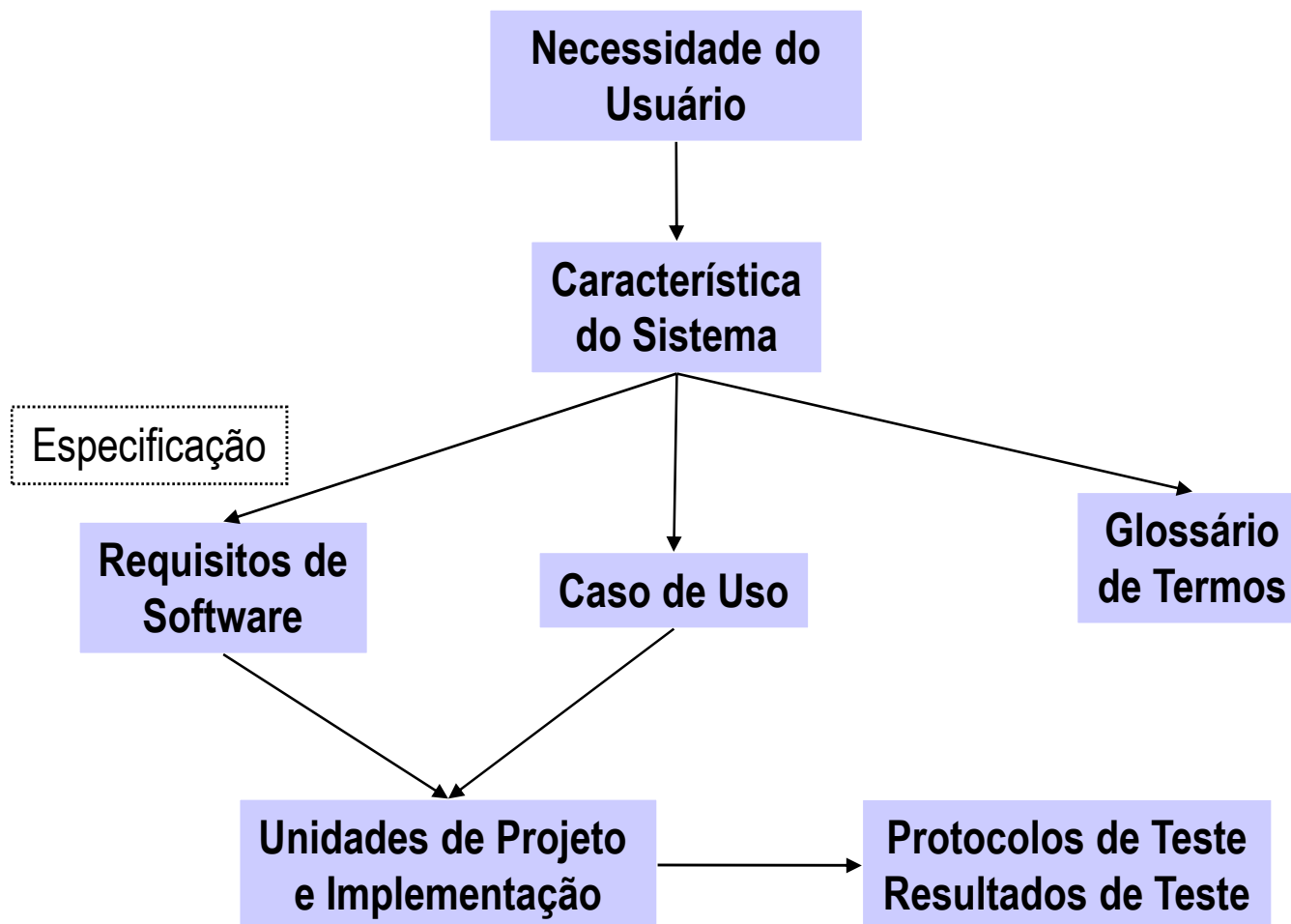


- Construindo certo o produto (verificação)
  - Consistência entre requisitos, projeto, codificação e testes
  - Auxílio dos processos de garantia da qualidade
- Construindo o produto certo (validação)
  - Os itens sendo construídos refletem os requisitos levantados e são aceitos pelo usuário
  - Testes

# Verificação através do Rastreamento de Requisitos



# Rastreamento de Requisitos





# Matriz de Relacionamentos

- Características X Requisitos
- Características X Casos de Uso
- Casos de Uso X Casos de Teste
- ...

# Engenharia de Mudanças

- Razões para mudança de requisitos:
  - Fatores externos:
    - Mudanças no problema
    - O cliente/usuário muda de opinião
    - O ambiente muda
    - Uma versão do sistema é colocada em uso
  - Fatores internos:
    - Requisitos não identificados nas fases iniciais do projeto
    - Dificuldades em gerenciar mudanças (qq coisa pode ser mudada)

# Engenharia de Mudanças

- Recomendações:
  - Reconhecer que a mudança é inevitável e, planejá-la
    - Estar preparado
  - Manter uma *baseline* de requisitos
    - Referência do que era original, o que foi introduzido e o que foi descartado.

# Engenharia de Mudanças

- Recomendações:
  - Estelecer um canal único de controle de mudanças
  - Usar um sistema de controle de mudanças
- Gerenciar a mudança hierarquicamente
  - atualizando os diversos níveis: necessidades, características, requisitos, projeto, implementação, teste e documentação do usuário
- Manter os requisitos sob Engenharia de Configuração

# Engenharia de Requisitos – Quem faz?

- A Engenharia de requisitos atinge todos os membros de uma equipe de projeto, mas de forma distinta.
- A Engenharia de requisitos efetiva só pode ser realizada por uma equipe com habilidades adequadas.

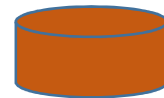
**“Traga-me uma pedra.”**



**“Legal. Mas o que eu realmente queria era uma pedra laranja.”**



**“Puxa, ela podia ser mais redondinha, né?”**



**“Perfeito. Agora sim eu tenho com o que segurar estes papéis que não cabem mais no meu arquivo.”**

# Bibliografia

- LEFFINGWELL, D., WIDRIG, D., 2001, Managing Software Requirements, Addison-Wesley, 1a edição.
- PRESSMAN, R.S., Engenharia de Software, 2002, Mc Graw Hill, 5a. Edição.