

# **Requisitos de Software**

**Profº Danilo Rocha.**

Sommerville - 9 ed. cap 4.

# **O que são Requisitos de Software?**

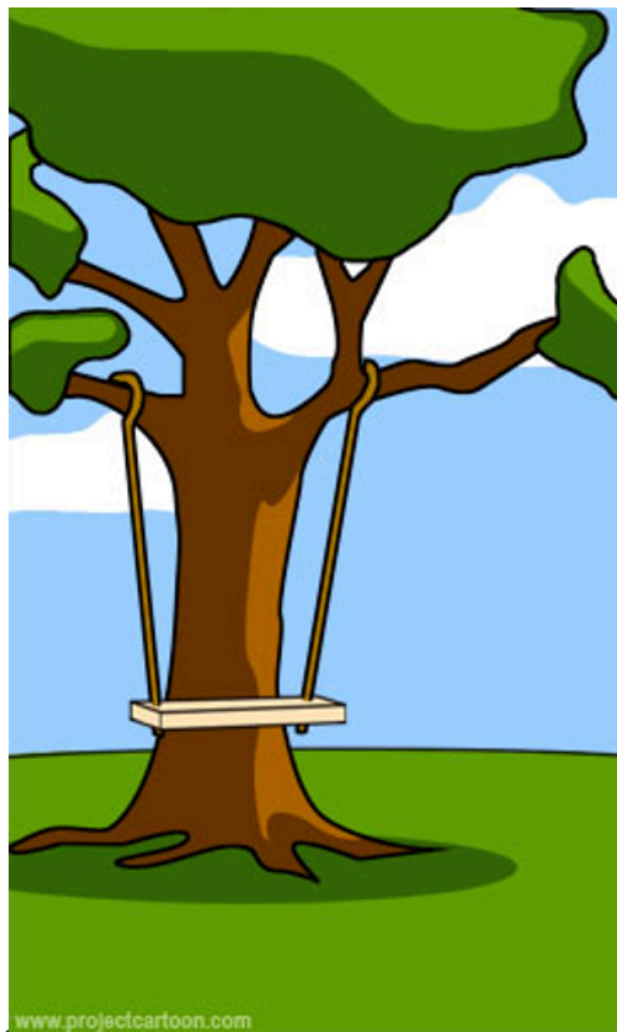
# Requisitos de Software

- Descrições do que **o sistema deve fazer**
- Condição necessária para a obtenção de um certo objetivo, ou para o preenchimento de certo fim. (AURÉLIO)
- Inclui: os **serviços** fornecidos pelo sistema, suas **qualidades específicas** e suas **restrições operacionais**
- Esses requisitos refletem as **necessidades dos clientes** de um sistema

# Por que é difícil entender os requisitos de um software?



Como o cliente  
explicou



Como o líder de projeto  
entendeu



O que o cliente realmente  
necessitava

# Por que é difícil entender os requisitos?

Diferentes níveis de descrição, ex:

## Requisitos de usuário

1. O sistema **deve** gerar relatórios mensais que mostrem o custo dos medicamentos prescritos por clínica durante cada mês

## Requisitos de sistema

1. No último dia de cada mês **deve** ser gerado um resumo dos medicamentos prescritos por clínica durante aquele mês
2. Um relatório por clínica **deve** ser gerado, listando nome dos medicamentos, total de prescrições e o custo total
3. Se os medicamentos estão disponíveis em diferentes unidades de dosagem (10mg, 20mg) **devem** ser criados relatórios separados

# **Tipos de Requisitos**

- Requisitos Funcionais
- Requisitos Não-Funcionais

# Requisitos Funcionais

- Requisitos diretamente ligados a...
  - **Funções** que o sistema deve fornecer.
  - Como o sistema deve **reagir a entradas específicas**.
  - Como o sistema deve se **comportar em determinadas situações**.
  - Podem também declarar o que o sistema **não deve fazer**.

# Requisitos Funcionais - Exemplos

- “O usuário **deve** conseguir fazer buscas em todo o acervo de materiais bibliográficos.”
- “O sistema **deve** fornecer **telas apropriadas** para o usuário ler documentos disponíveis no repositório de documentos.”
- “O sistema **deve** permitir o cadastro dos fornecedores da loja”
- “O sistema **deve** utilizar os dados obtidos a partir dos sensores e interpretá-los para realizar a navegação”



# Requisitos Funcionais - Qualidade e Precisão

- Surgem vários problemas quando os requisitos não são declarados de forma precisa.
- Requisitos ambíguos podem ser interpretados de diferentes maneiras pelos desenvolvedores e usuários.
- Considere o termo ‘telas apropriadas’.
  - Intenção do Usuário: telas especiais para cada diferente tipo de documento.
  - Interpretação do Desenvolvedor: fornecer uma tela texto que mostra o conteúdo do documento.

# Requisitos Funcionais

## Qualidade - Completeza e Consistência

- Os requisitos devem ser completos e consistentes.
  - **Completo**
    - Eles devem incluir descrição de todas as facilidades que estão sendo requeridas.
  - **Consistente**
    - Eles não devem apresentar conflitos ou contradições entre as descrições das facilidades fornecidas pelo sistema.
- Na prática, é impossível produzir um Documento de Requisitos completo e consistente.
  - Importante a validação do Documento de Requisitos!!

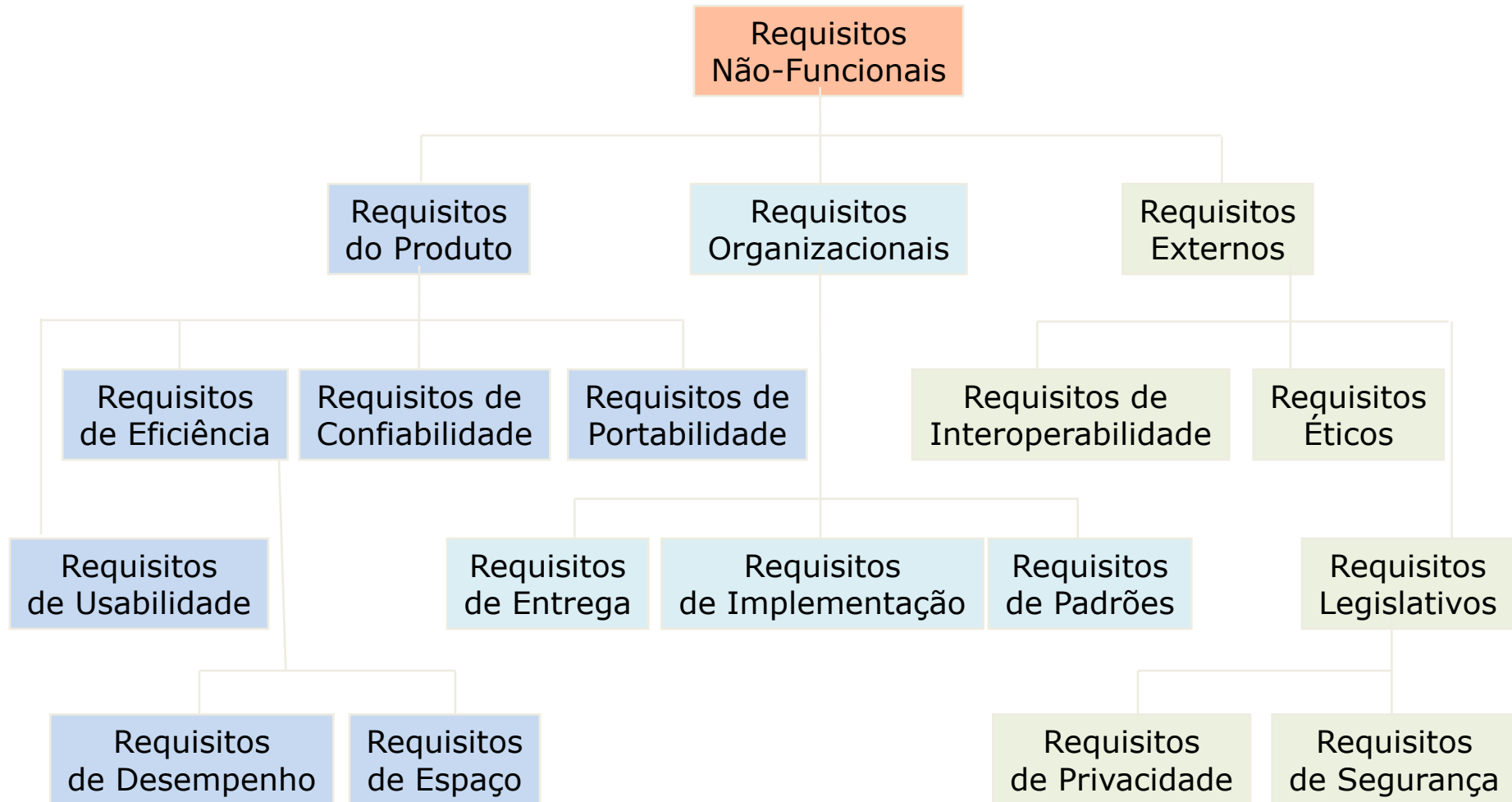
# **Requisitos Não-Funcionais**

# Requisitos Não-Funcionais

- São requisitos que expressam:
  - Restrições que o software deve atender
  - Qualidades específicas que o software deve ter

# Requisitos Não-Funcionais

## Tipos



# Requisitos Não-Funcionais

## Exemplos

- **Requisitos do Produto**

- O sistema deve ser robusto e tolerante a falhas, de forma a continuar sua operação ou abortar de forma segura o modo autônomo caso haja falha de um ou mais sistemas essenciais

- **Requisitos Organizacionais**

- O processo de desenvolvimento do sistema e os produtos liberáveis devem estar em conformidade com o padrão empresarial XYZ.

- **Requisitos Externos**

- Os operadores do sistema não devem ter acesso a qualquer dado que não necessitem.

# Requisitos Não-Funcionais

## Declaração de Requisitos Verificáveis

- Requisitos Não-Funcionais podem ser muito difíceis de serem declarados precisamente.
  - **Requisito Não-Funcional Verificável**
    - Declaração que usa alguma **métrica** que possa ser objetivamente testada.
    - **Exemplo:**
      - Controladores experientes devem ser capazes de usar todas as funções do sistema depois de duas horas de treinamento. Depois desse treinamento, o número médio de erros feito por um usuário experiente não deve exceder dois erros por dia.

# Requisitos Não-Funcionais

## Métricas para Verificar Requisitos

Requisito	Métrica
Velocidade	Transações processadas/segundo Tempo de resposta ao usuário/evento Tempo de refresh da tela
Tamanho	Bytes Tamanho específico de memória
Facilidade de uso	Tempo de treinamento Número de telas de help
Confiabilidade	Tempo médio de falha Disponibilidade Taxa de ocorrência de falhas
Robustez	Tempo de reinício depois de falha Porcentagem de eventos que causam falhas Probabilidade de que dados sejam corrompidos por falhas
Portabilidade	Porcentagem de declarações dependentes do sistema alvo Número de sistemas alvo



# Requisitos Não-Funcionais

## Metas

- Requisitos Não-Funcionais podem ser muito difíceis de serem declarados precisamente.
- Podem ser utilizadas “Metas”.
  - Transmitem as intenções dos usuários do sistema.
  - **Exemplo:** O sistema de controle de aeronave deve ser fácil de ser usado por controladores experientes e deve estar organizado de tal maneira que os erros dos usuários sejam minimizados.

# Requisitos Não-Funcionais

## Interação entre Requisitos

- Em sistemas complexos são comuns conflitos entre diferentes Requisitos Não-Funcionais.
- Exemplo: Sistema para aeronaves.
  - Para minimizar o peso, o número de chips do sistema deve ser minimizado.
  - Para minimizar o consumo de energia, chips de menor potência devem ser usados.
  - *Entretanto, usar chips de menor potência pode significar que mais chips devem ser usados. Qual é o requisito mais crítico?*

# Exercícios

1- Identifique os **requisitos funcionais** e **não funcionais**.

2- Aponte possíveis incertezas nessa descrição.

*“Um sistema automático de emissão de passagens vende passagens de trem. A partir de uma lista de possíveis destinos, os usuários escolhem seu destino e apresentam um cartão de crédito e um número de identificação pessoal. Os destinos possíveis devem ser organizados de modo a facilitar a escolha. Após a escolha do destino, o sistema deve responder prontamente se há espaço disponível no trem. A passagem é emitida e o custo dessa passagem é incluído em sua conta do cartão de crédito. Quando o usuário pressiona o botão para iniciar, uma tela de menu com os possíveis destinos é ativada, juntamente com uma mensagem para que o usuário selecione um destino. Uma vez selecionado um destino, pede-se que os usuários insiram seu cartão de crédito. A validade do cartão é checada e o usuário então deve fornecer um número de identificação pessoal. Quando a transação de crédito é validada, a passagem é emitida. O formato do bilhete de passagem deve seguir ao padrão definido pelo Sistema Nacional de Tráfego Ferroviário”.*

# Exercícios: Respostas

## 1- Requisitos funcionais (RF)

RF2: listar os possíveis destinos

*RF1: Quando o usuário pressiona o botão para iniciar, uma tela de menu com os possíveis destinos é ativada, juntamente com uma mensagem para que o usuário selecione um destino.*

RF2: receber pagamento de cartão de crédito

*RF2: Uma vez selecionado um destino, pede-se que os usuários insiram seu cartão de crédito.*

RF3: verificar se existem vagas no destino escolhido

*RF3: O sistema deve informar se existem vagas no destino escolhido*

RF4: checar a validade do cartão e receber número de identificação pessoal

*RF4: A validade do cartão é checada e o usuário então deve fornecer um número de identificação pessoal.*

RF5: emitir passagem e debitar custo no cartão de crédito

*RF4: Quando a transação de crédito é validada, a passagem é emitida e o custo dessa passagem é incluído em sua conta do cartão de crédito.*

# Exercícios: Respostas

## 1- Requisitos não funcionais (RNF):

### RNF do Produto

RF1: Usabilidade: facilidade de uso:

*RNF1: As telas devem **facilitar** a escolha do destino*

RF2: Desempenho: tempo de resposta adequado:

*RNF2: O **tempo de resposta** sobre vaga no trem deve ser adequado*

### RNF Organizacional

RF3: Padrão definido pelo SNTF

*RNF3: O formato do bilhete de passagem deve seguir ao **padrão** definido pelo Sistema Nacional de Tráfego Ferroviário”.*

# Exercícios: Respostas

## 2- Aponte possíveis incertezas nessa descrição.

“Um sistema automático de emissão de passagens vende passagens de trem. Os usuários escolhem seu destino e apresentam um cartão de crédito e um número de identificação pessoal. A passagem é emitida e o custo dessa passagem é incluído em sua conta do cartão de crédito. Quando o usuário pressiona o botão para iniciar, uma tela de menu com os possíveis destinos é ativada, juntamente com uma mensagem para que o usuário selecione um destino. Uma vez selecionado um destino, pede-se que os usuários insiram seu cartão de crédito. A validade do cartão é checada e o usuário então deve fornecer um número de identificação pessoal. Quando a transação de crédito é validada, a passagem é emitida. O formato do bilhete de passagem deve seguir ao padrão definido pelo Sistema Nacional de Tráfego Ferroviário”.

# Problemas com Requisitos

- Erros mais comuns cometidos no desenvolvimento:
  - ➡ Ignorar um grupo de clientes
  - ➡ Ignorar um único cliente
  - ➡ Omitir um grupo de requisitos
  - ➡ Permitir inconsistências entre grupos de requisitos
  - ➡ Aceitar requisito inadequado
  - ➡ Aceitar requisito incorreto, indefinido, ou impreciso
  - ➡ Aceitar um requisito ambíguo e inconsistente

# Engenharia de requisitos

- Os requisitos e as formas de obtê-los e documentá-los **variam** drasticamente de um projeto para o outro



Dilbert by Scott Adams, 2006



# Engenharia de Requisitos

- *Engenharia de Requisitos (E.R.)* estabelece o processo de definição de requisitos.
- Este processo deve lidar com diferentes pontos de vista, e usar uma combinação de métodos, ferramentas e pessoal.
- O produto desse processo é um modelo, do qual um documento de requisitos é produzido.
- Este processo acontece num contexto previamente definido a que chamamos de Universo de Informação. (Júlio Leite, 1994)

# Atividades genéricas comuns em Engenharia de Requisitos:

- Extração (elicitação) de requisitos;
- Análise de requisitos;
- Validação de requisitos;
- Gerenciamento de requisitos.

# Atividades genéricas comuns em Engenharia de Requisitos:

- Extração (elicitação) de requisitos;
- Análise de requisitos;
- Validação de requisitos;
- Gerenciamento de requisitos.

# Extração (Elicitação) de Requisitos de Software

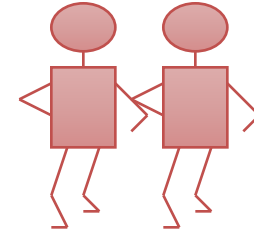
**ELICITAR** = Eliciar + Clarear + Extrair + Descobrir , tornar explícito, obter o máximo de informação para o conhecimento do objeto em questão.

- Pode envolver:
  - Usuários finais
  - Gerentes
  - Engenheiros envolvidos na manutenção
  - Especialistas de domínio
  - Representantes de sindicato, etc.
    - Estes são chamados **stakeholders** (partes interessadas)



# Extração de Requisitos de Software

## Problemas



### Stakeholders:

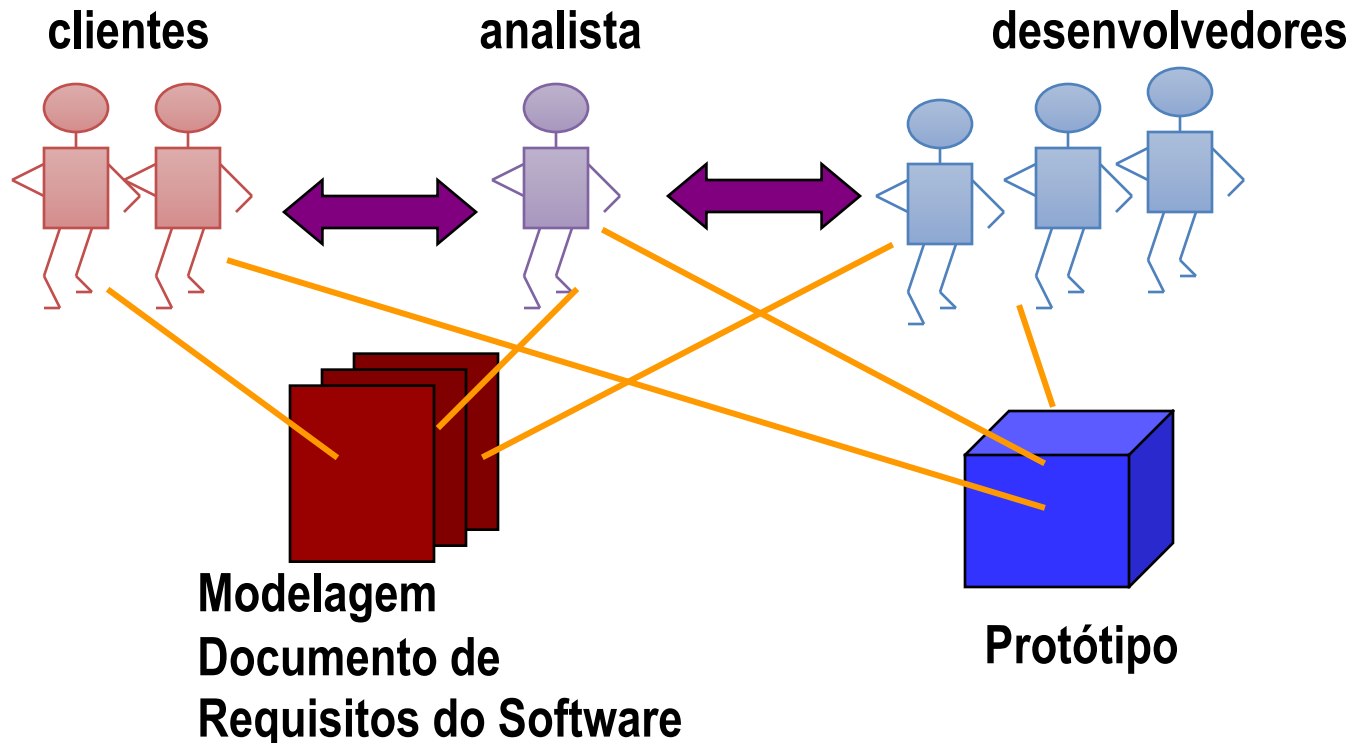
- não sabem o **querem** do software
- não sabem **explicar** o **querem** do software
- usam sua própria **linguagem**
- podem ter requisitos **conflitantes**

**Fatores organizacionais** podem influenciar

**Requisitos** mudam durante a engenharia de requisitos

# Extração dos Requisitos

- Extração de requisitos é o processo de **transformação** das ideias que estão na mente dos clientes (a entrada)
  - em um documento formal (saída)
- A meta é o reconhecimento dos **elementos básicos do problema**, conforme percebidos pelo cliente



# Extração dos Requisitos

- **Processo crítico** em um projeto de software
- Requisitos incompletos, incorretos ou mal entendidos são as causas mais frequentes **da baixa qualidade, excesso de custo e atrasos nas liberações do software**
- Pesquisas têm mostrado que a maioria dos softwares vendidos **não satisfaz as necessidades do usuário**

# Extração dos Requisitos

- HÁ TRÊS ATIVIDADES PRINCIPAIS:
  - Identificação de fontes de informação
  - Coleta de Fatos
  - Comunicação



# **Extração dos Requisitos**

## ***IDENTIFICAÇÃO DAS FONTES DE INFORMAÇÃO***

- Universo de Informação: contém toda informação necessária
- Agentes / Stakeholders (Atores, Usuários)
- Outras fontes de Informação:
  - Documentação do macrosistema
  - Políticas
  - Manuais
  - Memos, atas, contratos...
  - Livros sobre tema relacionado
  - Outros sistemas da empresa
  - Outros sistemas externos

# Extração dos Requisitos

## *IDENTIFICAÇÃO DAS FONTES DE INFORMAÇÃO*

- *Importante:*

- Priorizar as Fontes de Informação!
- Ex:
  - Atores mais importantes
  - Documentos mais mencionados
  - Rede de comunicações entre os componentes do macro-sistema

# Extração dos Requisitos

## *COLETA DE FATOS*

- Leitura de documentos
- Observação
- Entrevistas
- Questionários
- Análise de Protocolos
- Participação ativa dos stakeholders
- Reuniões
- Reutilização
- Recuperação (eng. reversa) do projeto do software

# Extração dos Requisitos

## *COMUNICAÇÃO*

- Atividade fundamental para que a fase de elicitação tenha sucesso
- Trata-se da comunicação entre clientes/agentes e os engenheiros de software
  - *Entendimento*: estabelecimento de um contexto comum.
  - *Linguagem*
  - *Nível de Abstração*
  - *Retro-alimentação (Feedback)*

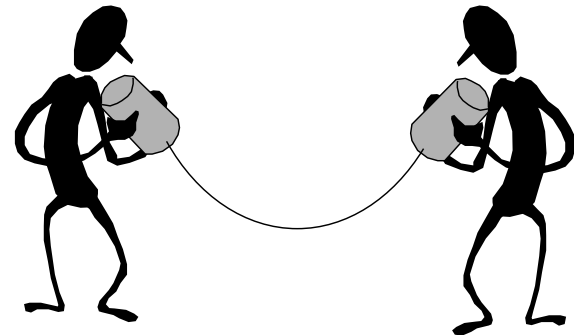
# Extração de Requisitos

É possível minimizar os problemas de comunicação nesta fase?

UTILIZAÇÃO DE TÉCNICAS DE APOIO  
PARA A ELICITAÇÃO DE REQUISITOS

# Extração de Requisitos

- Algumas técnicas são propostas visando auxiliar a comunicação e a extração dos requisitos
  - *Entrevistas*
  - *Cenários*
  - *Estórias do usuário*
  - *Etnografia*
  - *Prototipação*

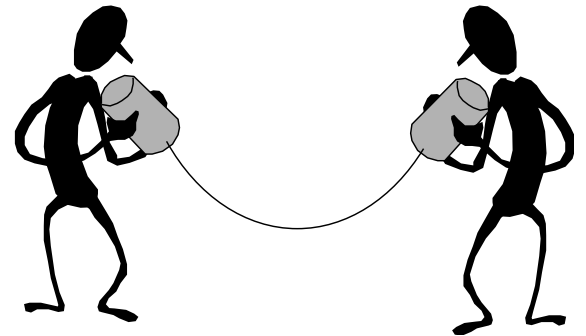


# Extração de Requisitos

- Algumas técnicas são propostas visando auxiliar a comunicação e a extração dos requisitos

- ***Entrevistas***

- *Cenários*
    - *Estórias do usuário*
    - *Etnografia*
    - *Prototipação*



# Extração de Requisitos:

## Entrevista

- Em entrevista formal ou informal, a equipe de ER formula questões para os *stakeholders* sobre os sistemas que eles usam e o sistema a ser desenvolvido.
- Tipos de entrevistas:
  - **Entrevistas fechadas (Estruturadas):** um conjunto de questões predefinidas são respondidas
  - **Entrevistas abertas (Não estruturadas):** conversa informal, uma variedade de assuntos são explorados com os stakeholders
  - **Semi-estruturadas:** questões abertas e outras objetivas



# Extração de Requisitos:

## Entrevista

- **Planejamento** da entrevista!
- **Início:** Questões livres de contexto (**quebrar o gelo!**)
  - *Quem está por trás da solicitação deste trabalho?*
  - *Quem vai usar a solução?*
  - *Qual será o benefício econômico para uma solução bem-sucedida?*

# Extração de Requisitos:

## Entrevista

- Questões que ajudam a entender o problema:
  - *Você pode me mostrar ou descrever o ambiente no qual a solução será usada?*
  - *Que tipo de saídas você considera importante?*
  - *Que problemas existem para a solução de software?*
  - *Existem questões de desempenho ou restrições que podem afetar o software?*

# Extração de Requisitos:

## Entrevista

- **Final: Questões que focalizam a efetividade da reunião**
  - *Você é a pessoa certa para responder a essas questões? Suas respostas são “oficiais”?*
  - *Minhas questões são relevantes para o problema que você tem?*
  - *Estou formulando muitas questões?*
  - *Alguém mais pode fornecer informação adicional?*
  - *Tem alguma questão que não fiz que você julga pertinente?*

# Extração de Requisitos:

## Entrevista

- Entrevistas são boas para obtenção de um **entendimento geral** do que os **stakeholders** fazem e como eles podem interagir com o sistema.
- Entrevistas não são boas para a compreensão de **requisitos de domínio**
  - Os engenheiros de requisitos podem não entender a **terminologia** específica de domínio;
  - Alguns conhecimentos de domínio são tão **específicos** que as pessoas acham **difícil explicar** ou pensam que não valem a pena **mencioná-los**

# Extração de Requisitos:

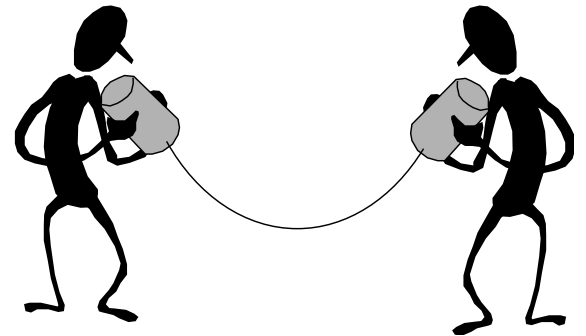
## Entrevista

### ENTREVISTAS EFETIVAS:

- Os **entrevistadores** devem ter mente aberta, desejarem **ouvir** os stakeholders e **não ter ideias preconcebidas** sobre os requisitos.
- Eles devem **induzir** os entrevistados com uma questão ou uma **proposta**, e não simplesmente esperar que eles respondam a uma questão tal como ‘o que você quer?’

# Extração de Requisitos

- Algumas técnicas são propostas visando auxiliar a comunicação e a extração dos requisitos
  - *Entrevistas*
  - **Cenários**
  - *Estórias do usuário*
  - *Etnografia*
  - *Prototipação*



# Extração de Requisitos:

## Cenários

- Cenários são exemplos reais de como um sistema pode ser usado
- Eles devem incluir
  - Uma descrição da situação **inicial**
  - Uma descrição do **fluxo normal** de eventos
  - Uma descrição do que pode dar **errado**
  - Informação sobre outras atividades **concorrentes**
  - Uma descrição do estado quando o cenário **termina**

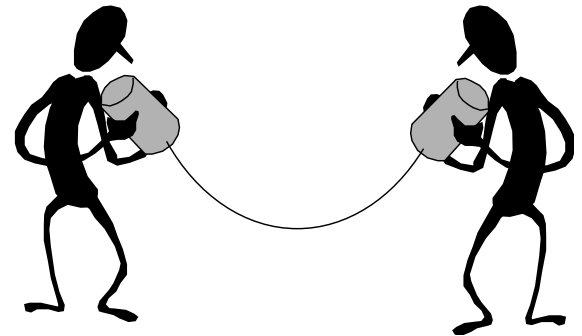
# **Extração de Requisitos:** **Cenários**

- Exemplos de cenário?
  - Saque em caixa eletrônico
  - Empréstimo de livro em biblioteca
  - Compra de livro na internet
  - Pilotar um avião



# Extração de Requisitos

- Algumas técnicas são propostas visando auxiliar a comunicação e a extração dos requisitos
  - *Entrevistas*
  - *Cenários*
  - ***Estórias do usuário***
  - *Etnografia*
  - *Prototipação*



# Extração de Requisitos:

## Estórias

- São frases escritas **pelo cliente** na sua linguagem, sobre algo que a aplicação **deve fazer**
- As estórias conduzem novas reuniões com usuários que podem ocorrer durante a fase de desenvolvimento.
- Feitas em cartões (manuscritas) que serão fixados em painéis
  - Ajudam a acompanhar o desenvolvimento (estória concluída, em desenvolvimento, não iniciado)
- Auxiliam durante os **testes de aceitação**

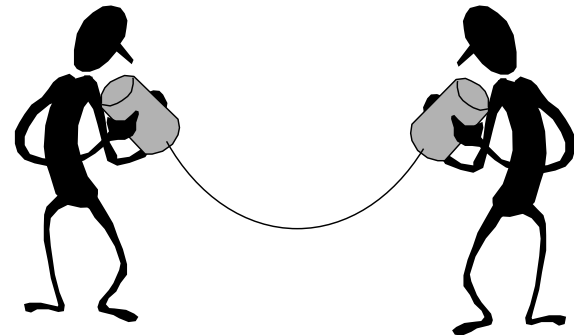
# Extração de Requisitos:

## Estórias

- Exemplo de estórias - para uma loja virtual:
  - “Um usuário possui um carrinho de compras no qual ele adiciona produtos que quer comprar”
  - “Um usuário faz o pagamento com cartão de crédito ou boleto bancário”
  - “Um usuário lê comentários feitos por outros sobre os produtos da loja”
  - “Um usuário recebe um e-mail de confirmação de compra quando efetua um pagamento”.

# Extração de Requisitos

- Algumas técnicas são propostas visando auxiliar a comunicação e a extração dos requisitos
  - *Entrevistas*
  - *Cenários*
  - *Estórias do usuário*
  - ***Etnografia***
  - *Prototipação*



# Extração de Requisitos:

## Etnografia

- Um analista **observa e analisa** como as pessoas realmente trabalham.
  - As pessoas não explicam seu trabalho.
- Fatores sociais e organizacionais de importância podem ser observados.
- Estudos de etnografia têm mostrado que o trabalho é, geralmente, mais rico e mais complexo do que o sugerido pelos modelos simples de sistema.

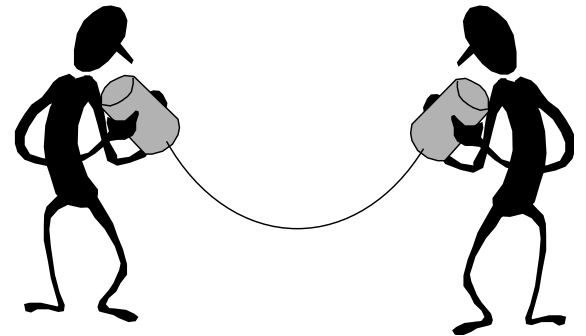
# Extração de Requisitos:

## Etnografia

- Requisitos do sistema se originam do **modo como** as pessoas realmente **trabalham**
  - Independem de como definições de processo sugerem que elas devam trabalhar.
- Ideal complementar com prototipação

# Extração de Requisitos

- Algumas técnicas são propostas visando auxiliar a comunicação e a extração dos requisitos
  - *Entrevistas*
  - *Cenários*
  - *Estórias do usuário*
  - *Etnografia*
  - ***Prototipação***



# Extração de Requisitos

## Prototipação

- Construção de modelos que representam o software
- Protótipos executáveis (papel ou software)
- Deve ser usado em conjunto com outras abordagem (ex caso de uso)



# Extração de Requisitos

## Prototipação

