

# Modelagem de Software

*Prof.: Sônia A Santana*



# Como Identificar Requisitos?

- Etapa na qual o analista vai buscar as informações sobre o software a ser desenvolvido:
  - Buscar junto ao usuário, seus sistemas e documentos todas as informações possíveis sobre as funções que o sistema deve executar e as restrições sobre as quais o sistema deve operar.
  - **Assume-se pouco conhecimento do analista sobre o sistema.**

# Como Identificar Requisitos?

- Você percebe alguma relação?



# Como Identificar Requisitos?

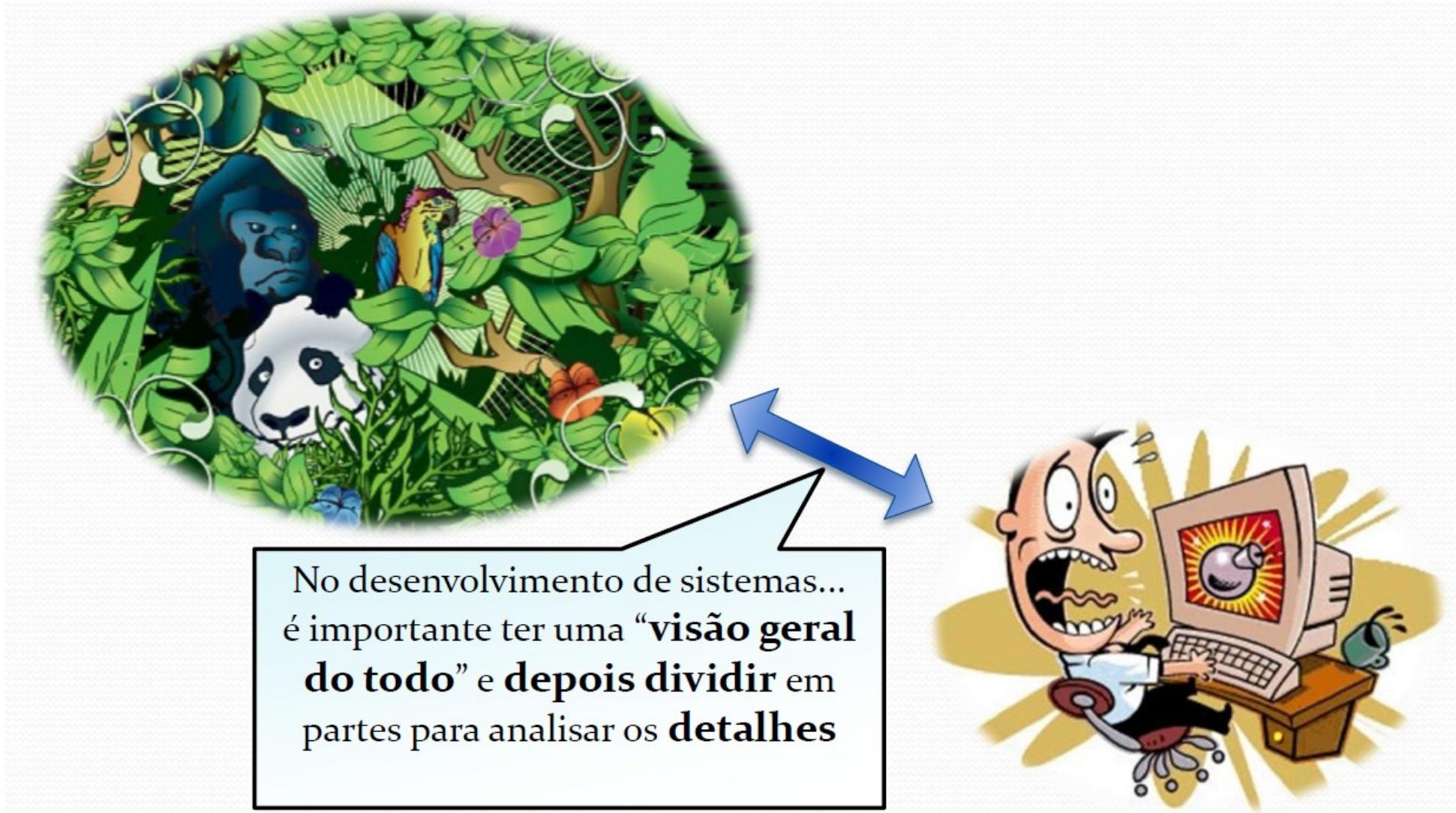


Um sistema a ser analisado é como uma floresta

Não é possível, em um primeiro momento, conhecer cada planta e cada animal



# Como Identificar Requisitos?



No desenvolvimento de sistemas...  
é importante ter uma “**visão geral do todo**” e **depois dividir** em  
partes para analisar os **detalhes**

# Como Identificar Requisitos?

No desenvolvimento de sistemas...  
é importante ter uma “**visão geral do todo**” e depois  
**dividir** em partes para analisar os **detalhes**

## Exemplo 1:

**Visão do Todo:** O Serviço desejado consiste em um Portal Colaborativo, onde qualquer usuário Web deseja obter Informações sobre as Olimpíadas de Londres em 2012

### **Visão das Partes:**

- 1º) Que tipo de informação esse usuário deseja visualizar?
- 2º) Quem detém essa informação e vai disponibilizá-la no sistema....

# Como Identificar Requisitos?

No desenvolvimento de sistemas...  
é importante ter uma “*visão geral do todo*” e depois  
**dividir** em partes para analisar os **detalhes**

## Exemplo 2:

**Visão do Todo:** A Universidade precisa de um sistema que controle a reserva e utilização de recursos áudio visuais

### ***Visão das Partes:***

- 1º) Qual é o procedimento atual para reserva de recursos?
- 2º) Quem pode efetuar reservas de recursos?
- 3º) ...

# Como Identificar Requisitos?

No desenvolvimento de sistemas...  
é importante ter uma “**visão geral do todo**” e **depois dividir** em partes para analisar os **detalhes**

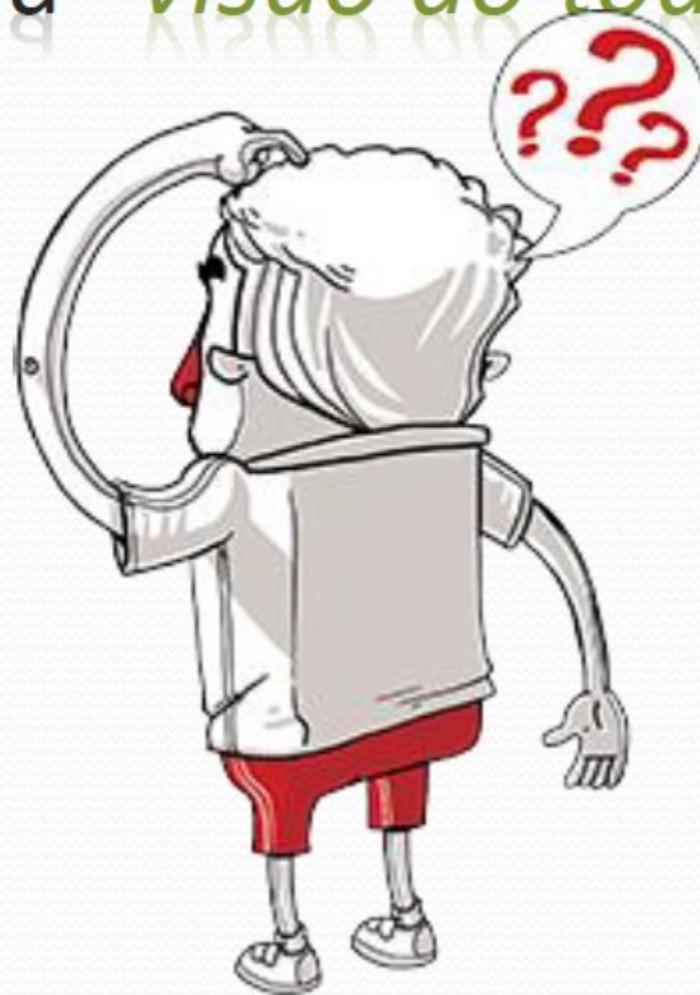
## Exemplo 3:

**Visão do Todo:** A empresa deseja controlar seu relacionamento com os clientes

**Visão das Partes:**

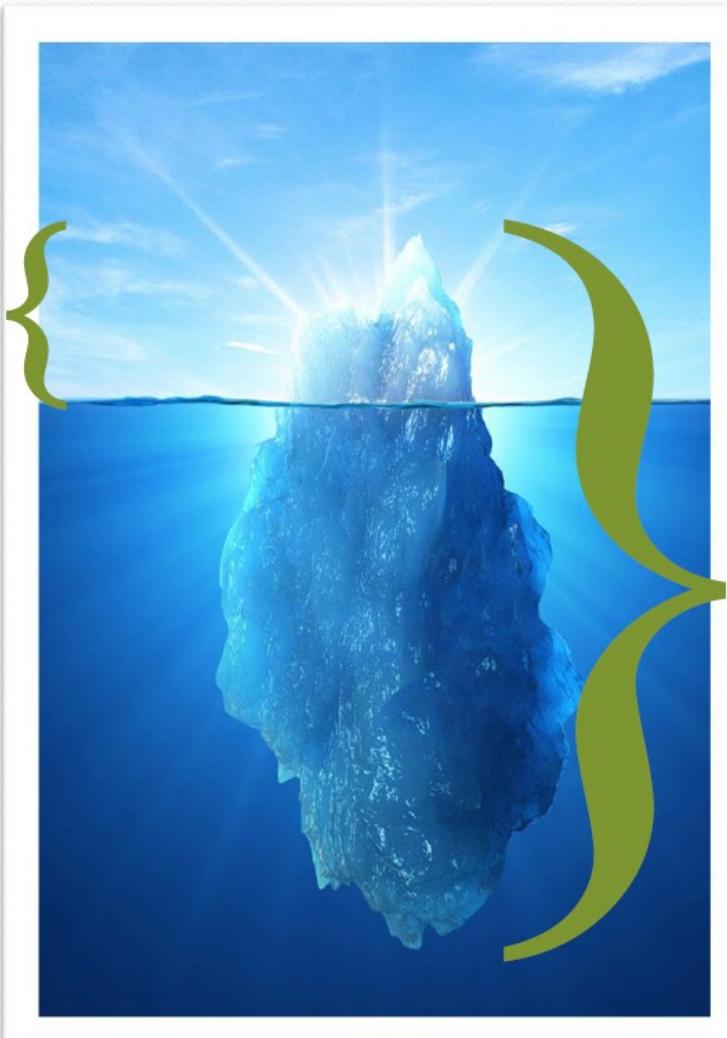
- 1º) Atualmente, existe alguma forma de controlar esse contato?
- 2º) Qual o procedimento realizado?
- 3º) ...

Mas, e se o usuário não sabe explicar  
uma “*visão do todo*”?



# Iceberg de Requisitos

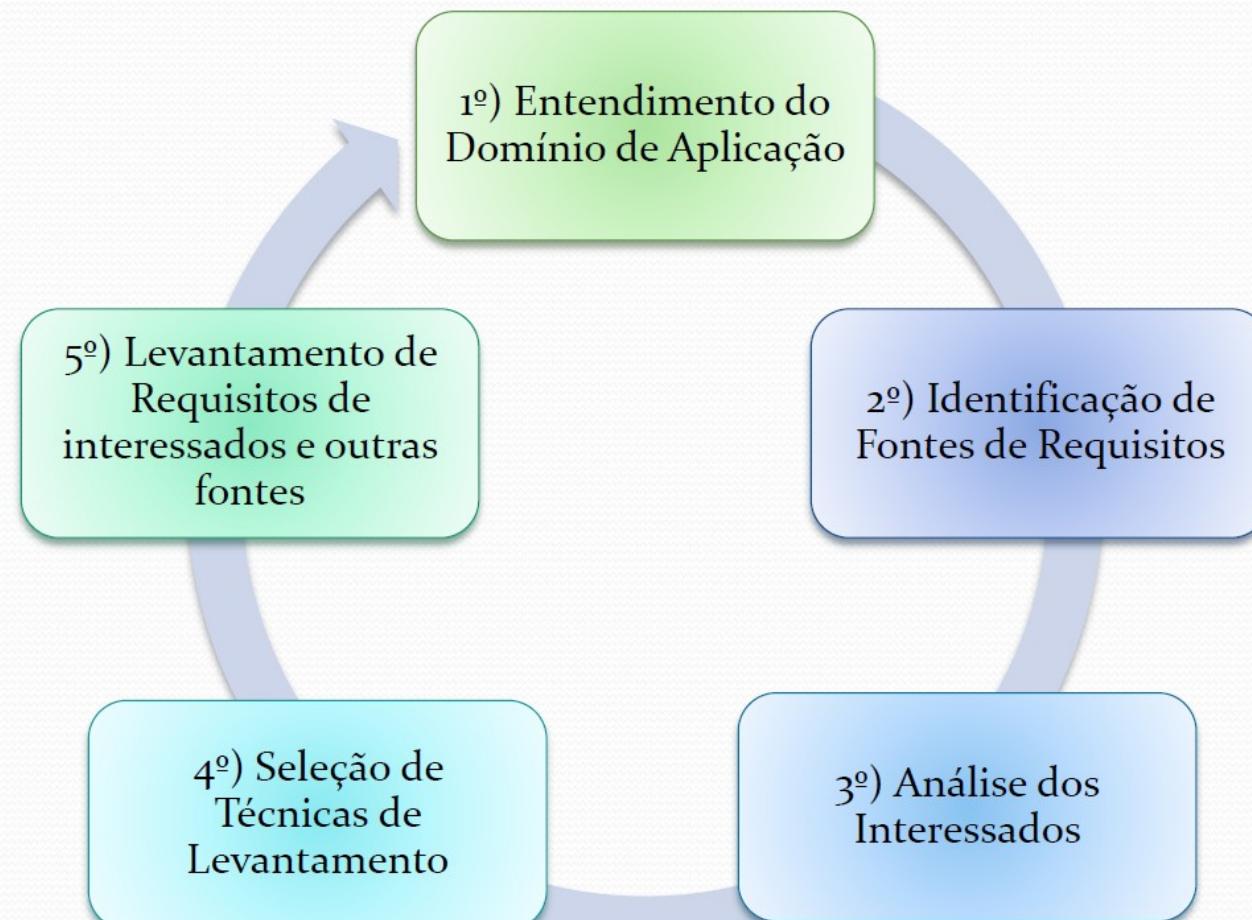
Visão  
do  
Cliente



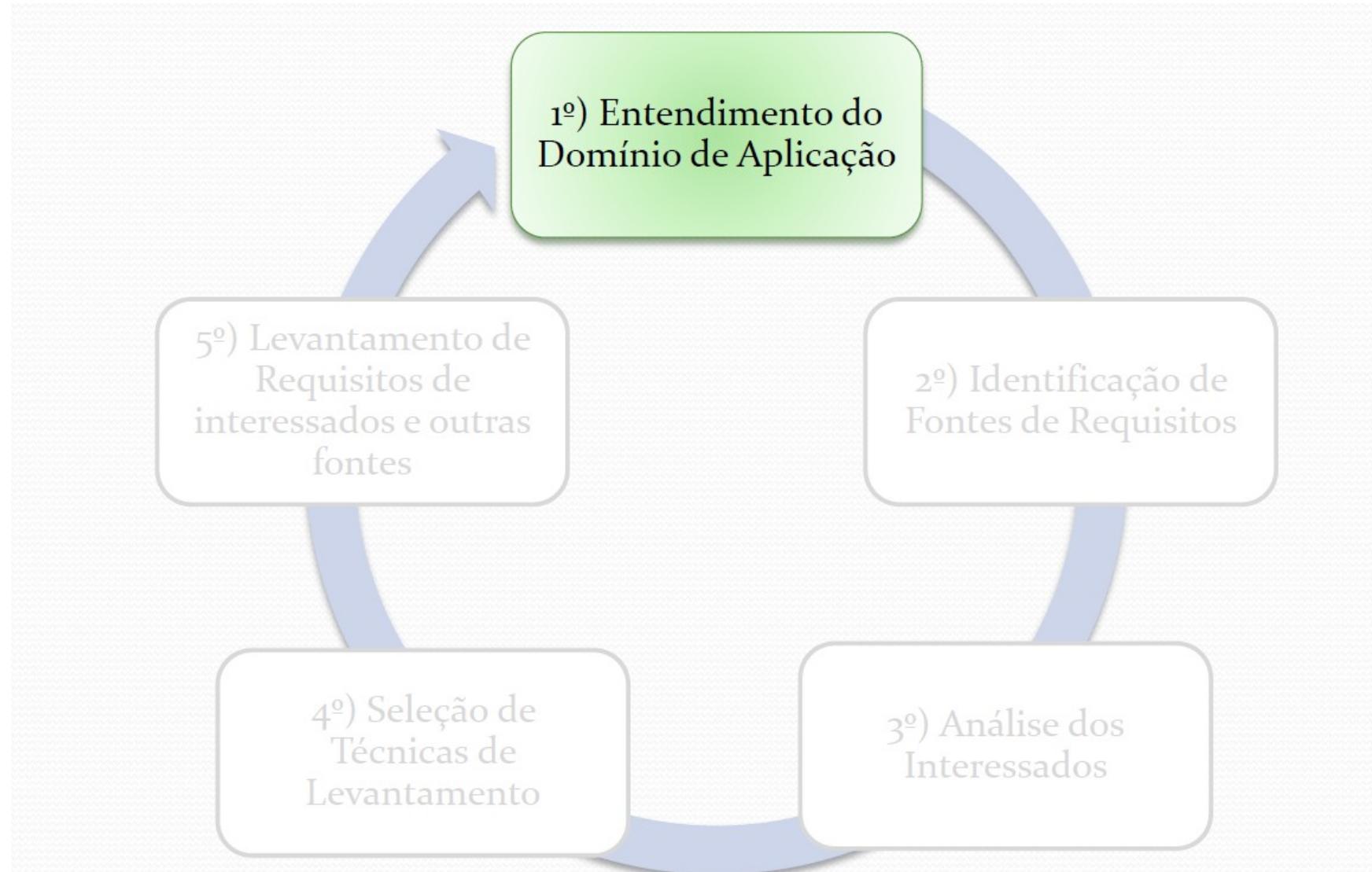
Requisitos  
do  
Sistemas

# Como Identificar Requisitos?

- Atividades básicas: Faça de forma incremental (etapas)



# Como Identificar Requisitos?



# Como Identificar Requisitos?

- Antes de iniciar a “*descoberta*” de requisitos propriamente dita, é importante...
  - Compreender os objetivos gerais do negócio a ser apoiado,
  - *Esboçar uma descrição* do problema a ser resolvido e identificar *por que o sistema é necessário* e quais são suas restrições
  - Levantar informações do contexto do desenvolvimento, informações sobre o domínio da aplicação e informações sobre *sistemas que estão em uso e serão substituídos* pelo sistema em desenvolvimento
  - Identificar as Regras de Negócio

# Como Identificar Requisitos?

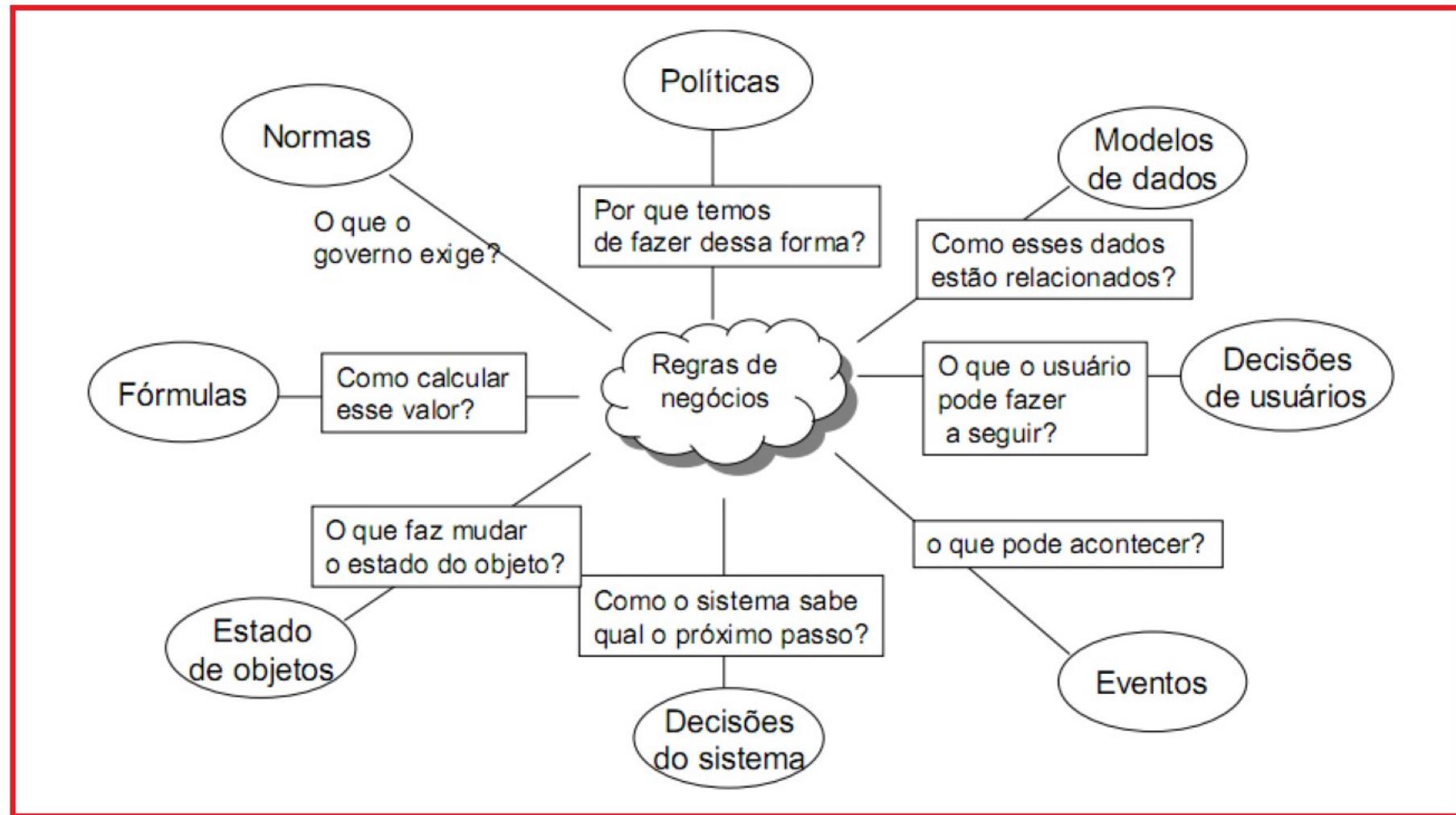
## ➤ *Regras de Negócio*

- Regras de negócios ou propriedades do domínio **não são requisitos**
- São fatos sobre o meio em que o sistema vai ser inserido.
- são verdadeiros mesmo se não houver um sistema
- São relevantes para o **desenvolvimento correto** do sistema

## ➤ *Exemplos de Regras de Negócio*

- Um aluno não pode se matricular em mais de 20 créditos por semestre
- Ativar o motor do videocassete faz os rolamentos girarem e puxarem a fita em uma direção
- Cheques vencem 30 dias após sua emissão
- A temperatura do paciente tem de ser medida a cada 2 horas

# Como Identificar Requisitos?



# Como Identificar Requisitos?

## ➤ Tipos de Regras de Negócio

- Fatos
- Restrições
- Gatilhos
- Inferências
- Cálculos
  - Especificam fórmulas ou algoritmos
    - ex: *toda compra acima de mil reais recebe um desconto de 10%*

# Como Identificar Requisitos?



# Como Identificar Requisitos?

## ➤ *Stakeholders*

- Cliente
- Usuários
- Equipe de Desenvolvimento
- Equipe de Comunicação

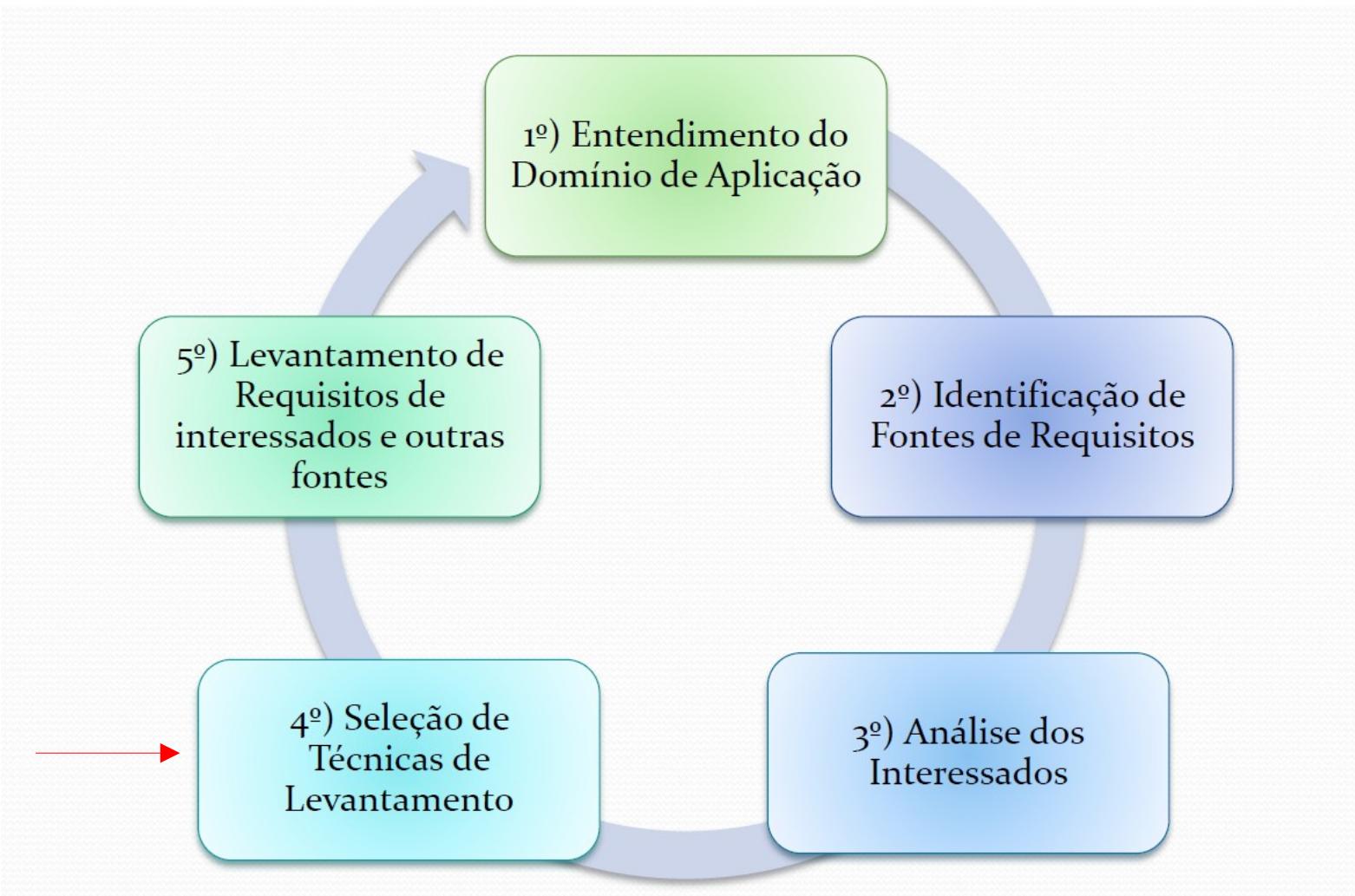
## ➤ *Fontes de Requisitos*

- Em relação aos Stakeholders, as principais fontes de requisitos são os usuários.

## ➤ *Análise dos Interessados Fontes de Requisitos*

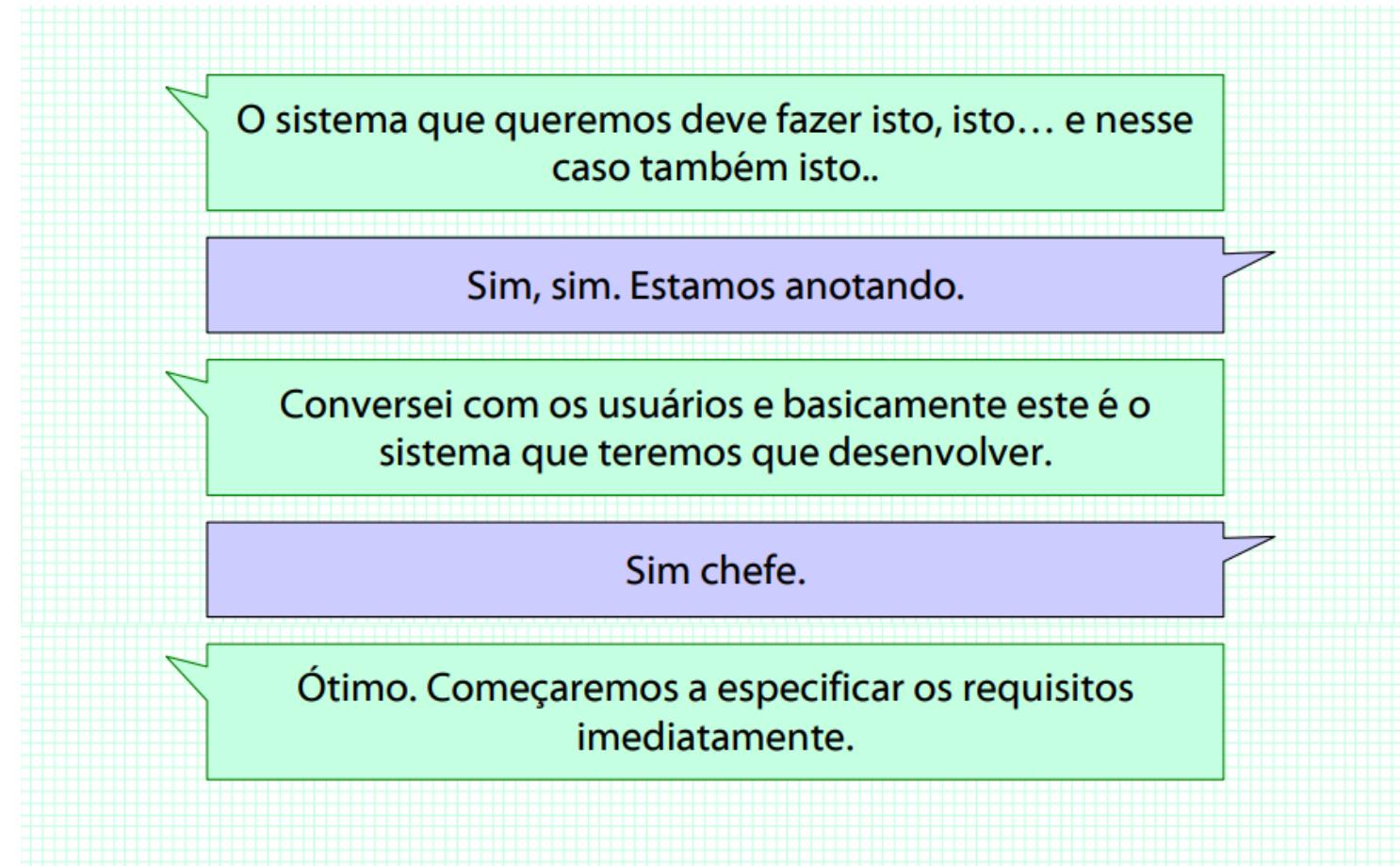
- Levantar o perfil dos interessados.
  - Ex.: Aquele usuário, que trabalha em determinado setor, que deverá interagir com o sistema diariamente. Ele é o responsável por realizar os cadastros de outros usuários e de recursos ou produtos (informação/dados importantes), além de gerenciar todos os cadastros já realizados.

# Como Identificar Requisitos?



# Um pouco de história....

- Já aconteceu com você?



# Dois Meses Depois....

Pessoal, após o emprego das mais modernas técnicas de especificação, com nossa melhor equipe, produzimos este documento que descreve minuciosamente o sistema.

Ótimo! Bom. Hum... É um documento com 300 páginas e todos esses gráficos, tabelas... Enfim, vamos analisá-lo e voltamos a nos falar!

# Sete Meses Depois....

Pessoal, o cliente não está satisfeito com o sistema pois diz que está fazendo as coisas diferentes do que ele pediu, e faltam algumas coisas...

Mas seguimos o documento que foi especificado com exatamente o que eles queriam...

Sim, mas o cliente disse que não tinha dito exatamente aquilo que estava no documento e que somente agora sentiu falta de algumas coisas...

COMO NÃO?? ESSES CLIENTES COMO SEMPRE NÃO SABEM O QUEREM..

# Dificuldades da Elicitação de Requisitos

- Usuários podem não ter uma idéia precisa do sistema por eles requerido;
- Usuários têm dificuldades para descreverem seu conhecimento sobre o domínio do problema
- Usuários e analistas têm diferentes pontos de vista do problema
- Não participação efetiva das pessoas que de fato usarão o sistema
- Entre outros...

# Seleção da Técnica de Levantamento de Requisitos (Elicitação)

Grupo de Usuários	Individual
Oficina de Requisitos	Entrevista
Entrevista (em grupo)	Protótipo
Protótipos	Questionário
Questionário	...
<i>Brainstorming</i>	
Cenários	
Observação	
...	

# Seleção da Técnica de Levantamento de Requisitos (Elicitação)

## - Entrevista

- Forma ***mais comum*** de levantamento (elicitação) de requisitos.
- **Objetivo:** *entender* os ***processos existentes*** e o domínio da aplicação
- Podem ser ***individuais*** ou em ***grupo***
- Deve-se ***preparar*** as perguntas ***antecipadamente***
  - Não se deve assumir qualquer conhecimento prévio sobre o problema
  - Chama-se essa postura de “***IGNORANTE INTELIGENTE***”

# Seleção da Técnica de Levantamento de Requisitos (Elicitação)

## - Questionários

- Muito utilizado quando os analistas identificam a necessidade de coletar informações de **multiusuários**
- FOCO:
  - Para obter **evidências** é necessário elaborar **questões fechadas**;
  - Para capturar **opiniões** e/ou **sugestões**, perguntas **abertas** podem ser incluídas
- Potencial **Ambiguidade**
  - Questionários podem apresentar problemas de interpretação (durante a aplicação e análise das respostas)
    - A **pessoa** que responde pode **não entender** as perguntas
    - O **analista** pode **não compreender** as respostas

# Seleção da Técnica de Levantamento de Requisitos (Elicitação)

## - ***Brainstorming (Chuva de Ideias)***

- Um ***grupo de pessoas*** é reunido, um ***cenário simulado*** e um assunto é ***discutido*** para ***elicitar os requisitos***
- Importante que os ***participantes*** da reunião se sintam ***confortáveis*** o bastante para ***discutir*** sobre o assunto levantado ***sem se sentirem ridículos***.
  - Na primeira sessão ***não*** se deve ***julgar*** as ideias
  - ***Sessões posteriores*** são usadas para ***priorizar*** o que foi ***levantado***

# Seleção da Técnica de Levantamento de Requisitos (Elicitação)

## - Prototipação

- ***Utilizada com frequência*** na elicitação de requisitos
  - Tipos de protótipos: descartáveis (**papel/digital**) e evolutivos
  - Protótipos **Descartáveis**
    - Função de ilustrar para os usuários e/ou clientes do sistema o que o analista entendeu sobre os requisitos – **Sem layout, preto e branco!**
  - Protótipos **Evolutivos**
    - **Contém os requisitos** do sistema no âmbito de implementação incremental.
    - Quais os **Benefícios** da utilização?
      - **Redução dos riscos** na construção
      - Baixo valor de correção, etc...

# Seleção da Técnica de Levantamento de Requisitos (Elicitação)

- **Seminários:**

- Reuniões com usuários dos diversos departamentos envolvidos.
- Planejar as reuniões.
- Escolher criteriosamente as pessoas.
- Conhecer previamente o assunto.
- Atuar como um líder imparcial.
- Esta técnica possibilita a quebra de feudos.

# Seleção da Técnica de Levantamento de Requisitos (Elicitação)

- **Observação**

- Vivenciar a rotina de trabalho dos usuários.
- Participar das atividades operacionais em conjunto com o usuário.
- Constatar problemas de relacionamentos entre os usuários e entre os departamentos.

# Seleção da Técnica de Levantamento de Requisitos (Elicitação)

- **Pesquisas**
  - Solicitação de todos os documentos usados pelos departamentos.
  - Documentos emitidos e recebidos pelo departamento.
  - Confrontar e analisar as informações dos documentos com as informações obtidas nas outras técnicas.

# Perguntas para Identificar Requisitos

- Quais são alguns motivos pelos quais você e seus colegas usariam o novo sistema?
- Que objetivos você tem em mente que esse sistema o ajudaria a cumprir?
- Que problemas você espera que esse sistema resolva para você?
- Que eventos externos estão associados ao sistema?
- Que palavras você usaria para descrever o sistema?
- Algumas partes do produto são mais importantes que outras por motivos de desempenho, segurança, robustez, disponibilidade, ou alguma outra característica?
- Há alguma restrição ou regra à qual o sistema deve estar de acordo?
- Como você imagina que o sistema deve ser semelhante à forma como você trabalha hoje? Como ele deve ser diferente?
- Que aspectos dos seus processos de negócios você quer manter? quais você quer substituir?
- Que aspecto do sistema você acha mais interessante?
- Que aspecto terá mais valor pra você? e menos valor?
- Como você vai julgar se o sistema é um sucesso?
- Você pode descrever o ambiente onde o sistema vai ser usado?

# Verificação e Validação de Requisitos

- Após sua especificação, os requisitos devem ser **verificados, validados e priorizados (aconselhável)**.
- O objetivo é garantir que eles estejam corretos, precisos, completos, consistentes e verificáveis.

# Verificação e Validação de Requisitos

- Requisitos devem ser verificáveis, isto é, deve ser possível testar se os requisitos estão sendo atendidos. Um contra-exemplo é um requisito que apenas requer que o sistema seja amigável. Como os desenvolvedores vão saber se estão atendendo a essa expectativa dos clientes?

# Verificação e Validação de Requisitos

- **Requisitos devem estar corretos.** Um contra-exemplo é a especificação de forma incorreta da fórmula para remuneração das cadernetas de poupança em um sistema bancário. Evidentemente, uma imprecisão na descrição dessa fórmula irá resultar em prejuízos para o banco ou para seus clientes.

# Verificação e Validação de Requisitos

- Requisitos devem ser precisos, isto é, não devem ser ambíguos. No entanto, ambiguidade ocorre com mais frequência do que gostaríamos quando usamos linguagem natural.

# Verificação e Validação de Requisitos

- **Requisitos devem ser completos.** Isto é, não podemos esquecer de especificar certos requisitos, principalmente se eles forem importantes no sistema que se pretende construir.
- **Requisitos devem ser consistentes.** Um contra-exemplo ocorre quando um stakeholder afirma que a disponibilidade do sistema deve ser 99,9% e outro considera que 90% já é suficiente.

# Priorização de Requisitos

- Por fim, os requisitos devem ser priorizados. Às vezes, o termo requisitos é interpretado de forma literal, isto é, como uma lista de funcionalidades e restrições obrigatórias em sistemas de software.
- No entanto, nem sempre aquilo que é especificado pelos clientes será implementado nas *releases* iniciais. Por exemplo, restrições de prazo e custos podem postergar a implementação de certos requisitos.

# Atores do Processo de Elicitação

- **Analista de Sistemas/Negócios**
  - Entender as necessidades dos clientes/usuários
  - Definir os limites do sistema.
  - Identificar e documentar os requisitos do sistema.
  - Usar um vocabulário comum entre usuários e demais envolvidos.
  - Desenvolver e manter o documento de requisitos.
  - Ter boas habilidades de comunicação (verbal e escrita)

# Artefatos Produzidos na Elicitação

- **Documento de Especificação de Requisitos**
  - A Visão Geral do Sistema (texto corrido que contém as principais ideias do cliente sobre o sistema)
  - Lista de requisitos funcionais e não funcionais
  - Outros artefatos:
    - *Glossário (para alinhar termos utilizados entre analista, cliente e usuários – **Evitando assim o Pedantismo**)*
    - *Protótipos (ajudam a esboçar ideias sobre as funcionalidades)*

# Documento de Especificação de Requisitos

- Não deve conter informações sobre soluções técnicas.
- Procura responder a questão: Quais são as necessidades do usuário?
- Serve como um termo de consenso entre a equipe técnica (desenvolvedores) e o cliente.
- Estabelece o escopo do sistema, ou seja, define o que é sistema e o que não é sistema.
- Mostrar os requisitos pelo grau de prioridade (aconselhável).

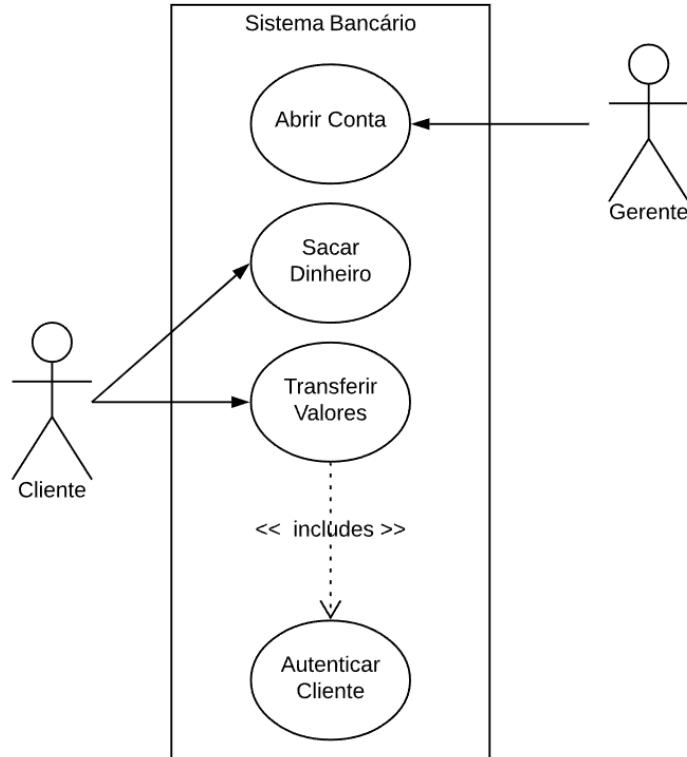
# Desafios da Engenharia de Requisitos

- Requisitos incompletos ou não-documentados (48%)
- Falhas de comunicação entre membros do time e os clientes (41%)
- Requisitos em constante mudança (33%)
- Requisitos especificados de forma abstrata (33%)
- Restrições de tempo (32%)
- Problemas de comunicação entre os próprios membros do time (27%)
- Stakeholders com dificuldades de separar requisitos e soluções (25%)
- Falta de apoio dos clientes (20%)
- Requisitos inconsistentes (19%)
- Falta de acesso às necessidades dos clientes ou do negócio (18%)

# Documentação de Requisitos

- “Documentações” de Requisitos
  - Documento de Especificação de Requisitos;
  - Casos de Uso;
- História de Usuários;

# Casos de Uso (UML)



“

## **Transferir Valores entre Contas**

**Ator:** Cliente do Banco

### **Fluxo normal:**

- 1 - Autenticar Cliente
- 2 - Cliente informa agência e conta de destino da transferência
- 3 - Cliente informa valor que deseja transferir
- 4 - Cliente informa a data em que pretende realizar a operação
- 5 - Sistema efetua transferência
- 6 - Sistema pergunta se o cliente deseja realizar uma nova transferência

### **Extensões:**

- 2a - Se conta e agência incorretas, solicitar nova conta e agência
- 3a - Se valor acima do saldo atual, solicitar novo valor
- 4a - Data informada deve ser a data atual ou no máximo um ano a frente
- 5a - Se data informada é a data atual, transferir imediatamente
- 5b - Se data informada é uma data futura, agendar transferência

# Histórias de Usuários

- No desenvolvimento de software e gerenciamento de produto, uma história de usuário é uma explicação informal e geral sobre um recurso de software escrito a partir da perspectiva do usuário final ou cliente
- É uma descrição simples e genérica de uma funcionalidade de software a partir da experiência do usuário. Ela define o que um usuário precisa da sua empresa, o que ajuda a priorizar o trabalho e a aumentar o valor para o cliente.

“

*História de Usuário = Cartão + Conversas + Confirmação*

# 3Cs

Em 2001, *Ron Jeffries* propôs a fórmula dos **Três C**. A história cumpre bem o seu papel quando atende às 3 características C, que são:

## Cartão:

É pequena o suficiente para caber em um cartão

## Confirmação

O comportamento esperado para confirmar o seu escopo. Também conhecido como plano de testes, escrito no verso do cartão

## Conversação:

Consegue promover a comunicação entre o usuário e a equipe dando um entendimento comum da funcionalidade a ser entregue

# Histórias de Usuários: Detalhando...

- **Cartão**, usado pelos clientes para escrever, na sua linguagem e em poucas sentenças, uma funcionalidade que esperam ver implementada no sistema.

# Histórias de Usuários: Detalhando...

- **Conversas** entre clientes e desenvolvedores, por meio das quais os clientes explicam e detalham o que escreveram em cada cartão. Como dito antes, a visão de métodos ágeis sobre Engenharia de Requisitos é pragmática: como especificações textuais e completas de requisitos não funcionam, elas foram eliminadas e substituídas por comunicação verbal entre desenvolvedores e clientes. Por isso, métodos ágeis incluem nos times de desenvolvimento um representante dos clientes, que participa do time em tempo integral.

# Histórias de Usuários: Detalhando...

- **Confirmação**, que é basicamente um teste de alto nível — de novo especificado pelo cliente — para verificar se a história foi implementada conforme esperado. Portanto, não se trata de um teste automatizado, como um teste de unidades, por exemplo. Mas a descrição dos cenários, exemplos e casos de teste que o cliente irá usar para confirmar a implementação da história. Por isso, são também chamados de testes de aceitação de histórias. Eles devem ser escritos o quanto antes, preferencialmente no início de uma iteração. Alguns autores recomendam escrevê-los no verso dos cartões da história.

# Histórias de Usuários

*Exemplos:*

Como um operador  
do hotel, eu quero  
estabelecer taxas ótimas  
para os quartos no  
meu hotel para  
maximizar meus ingressos

Como vice-presidente de  
marketing e vendas,  
eu quero rever o desempenho  
histórico de vendas  
para identificar regiões  
geográficas e produtos de  
melhor desempenho

# Características de Boas Histórias de Usuários

<b>I</b>	<b>independente</b>	Quanto mais independente a história, maior a facilidade de implementá-la em qualquer ordem e evitar “gargalos” na implementação. Deve conter informações suficientes sem depender da finalização de outras histórias.
<b>N</b>	<b>negociável</b>	Deve ser flexível. A história possui a essência e seus detalhes devem ser discutidos pelos envolvidos nos projetos. (3 C's – Cartão, conversa, confirmação)
<b>V</b>	<b>valiosa</b>	Deve ter valor para o cliente, produto ou equipe do projeto. O valor apoiará para priorizar o backlog. Vale pensar se realmente há valor na história, caso não, vale repensar se realmente deve ser feita.
<b>E</b>	<b>estimável</b>	Deve ter a quantidade de informações suficientes nos 3 C's que permita a estimativa pelo time, custo de desenvolvimento, demais métricas necessárias. Não precisa ser exato, mas se não conseguir estimá-la, é importante refinar para realizar a divisão.
<b>S</b>	<b>pequena</b>	Boas histórias são pequenas. É importante o acordo entre o Time e PO para o tamanho da história. Além disso quando são menores, há mais chance de precisão nas estimativas.
<b>T</b>	<b>testável</b>	A história sempre deve ser testada! Os testes devem ser realizados principalmente através dos critérios de aceite, pois se não se saber o que deve ser testado, provavelmente a história não está clara o suficiente ou não agrega o valor aos olhos do cliente.

# Características de Boas Histórias de Usuários

- Histórias devem ser **independentes**: dadas duas histórias X e Y, deve ser possível implementá-las em qualquer ordem. Para isso, idealmente, não devem existir dependências entre elas.

# Características de Boas Histórias de Usuários

- Histórias devem ser abertas para **negociação**.  
Frequentemente, costuma-se dizer que histórias (o cartão) são convites para conversas entre clientes e desenvolvedores durante um sprint.

# Características de Boas Histórias de Usuários

- Histórias devem agregar **valor** para o negócio dos clientes. Histórias são propostas, escritas e priorizadas pelos clientes e de acordo com o valor que elas agregam ao seu negócio. Por isso, não existe a figura de uma história técnica, como a seguinte: o sistema deve ser implementado em JavaScript, usando React no front-end e Node.js no backend.

# Características de Boas Histórias de Usuários

- Deve ser viável **estimar o tamanho** de uma história. Por exemplo, quantos dias serão necessários para implementá-la. Normalmente, isso requer que a história seja pequena, como veremos no próximo item, e que os desenvolvedores tenham experiência na área do sistema.

# Características de Boas Histórias de Usuários

- Histórias devem ser **succintas** e pequenas. Na verdade, até se admite histórias complexas e grandes, as quais são chamadas de épicos.

# Características de Boas Histórias de Usuários

- Histórias devem ser **testáveis**, isto é, elas devem ter critérios de aceitação objetivos. Como exemplo, podemos citar: o cliente pode pagar com cartões de crédito. Uma vez definidas as bandeiras de cartões de crédito que serão aceitas, essa história é testável. Por outro lado, a seguinte história é um contra-exemplo: um cliente não deve esperar muito para ter sua compra confirmada. Essa é uma história vaga e, portanto, com um critério de aceitação também vago.

# Épicos

Épicos são grandes histórias de usuário, normalmente aquelas que são grande demais para implementar em uma única iteração e, portanto, precisam ser desagregadas em histórias de usuário menores em algum momento.

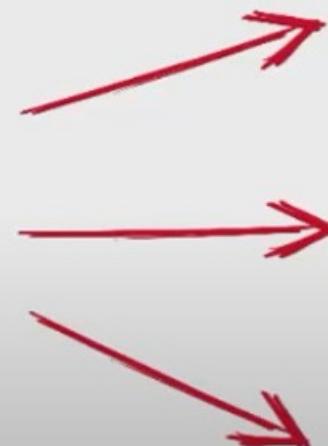
## Épico

O usuário pode pesquisar vagas de emprego

O usuário pode pesquisar vagas de emprego por localização, faixa salarial, título de vaga, nome da empresa e a data no qual a vaga foi publicada

O usuário pode visualizar informações de cada vaga que foram encontradas pela pesquisa

O usuário pode visualizar informações detalhadas sobre a empresa que publicou a vaga



# Exemplos

- Eu, enquanto deficiente visual, desejo que meu ambiente de trabalho seja mais acessível para que eu não dependa tanto de outras pessoas.
- ■ História 1: Eu, enquanto deficiente visual, desejo que os locais que eu tenho que ir sejam acessíveis para que eu não tenha que passar pelo constrangimento de ficar perguntando para as pessoas sobre os locais onde quero ir.
- História 2: Eu, enquanto deficiente visual, desejo chegar facilmente à saída de emergência, pois não quero morrer em um incêndio.

# Exemplo: História de Usuários

- **História 1.1:** Eu, enquanto deficiente visual, desejo **chegar facilmente aos banheiros** para que eu não tenha que passar pelo constrangimento de ficar perguntando para as pessoas sobre os locais aonde quero ir.
- **História 1.2:** Eu, enquanto deficiente visual, **desejo chegar facilmente aos elevadores** para que eu não tenha que passar pelo constrangimento de ficar perguntando para as pessoas sobre os locais aonde quero ir.
- **História 1.3:** Eu, enquanto deficiente visual, **desejo chegar facilmente às salas de reunião** para que eu não tenha que passar pelo constrangimento de ficar perguntando para as pessoas sobre os locais aonde quero ir.

# Exemplo: História de Usuários

- **História 1.1.1:** Eu, enquanto deficiente visual que trabalha no 5º andar, desejo chegar facilmente aos banheiros para que eu não tenha que passar pelo constrangimento de ficar perguntando para as pessoas sobre os locais aonde quero ir.

# Exemplo: Tarefas

- Tarefas são itens técnicos necessários para que uma História de Usuário se transforme em incremento do produto. Exemplo de tarefas da História 1.1.1:
  - Adquirir módulo de sensor de presença;
  - Adquirir alto-falantes;
  - Fazer o design das novas placas;
  - Criar os modelos de engenharia;
  - Montar as placas;
  - Instalar as placas nos banheiros do 5º andar; etc

# Vantagens e Desvantagens das Histórias de Usuários

Vantagens	Desvantagens
<b>Criar Histórias de Usuário é fácil</b> <ul style="list-style-type: none"><li>- Não se precisa de experiência</li><li>- Não é investido muito tempo ou dinheiro</li></ul>	Não é uma documentação detalhada (e não é este seu objetivo)
<b>São escritas pelos usuários</b> <ul style="list-style-type: none"><li>- Comunicação com o cliente</li><li>- Compreensão do que o produto final tem que realizar.</li></ul>	Não aborda explicitamente como documentar requisitos não funcionais
Priorização	Pode não ser adequada para ambientes com regulamentações ou quando o nível de documentação é imposto pelo cliente

# Exercício de Fixação: Solução

- Como especificar requisitos não-funcionais usando histórias de usuários?

# Critérios de Aceite

## CRITÉRIOS DE ACEITE

---

O que é?

- 1. Comportamentos
- 2. Saídas

- 1. Sentenças curtas
- 2. Cenários

Como descrever?

# Exemplo

- Épico:
  - Vender produtos
- História de Usuário
  - Eu, **como** vendedor
  - **quero** consultar pedidos
  - **para** verificar minha comissão.
- Critérios de Aceite
  - O resultado da consulta tem que ser mostrado de 10 em 10 segundos.
  - O valor da comissão de cada pedido tem que ser calculado considerando 1% do lucro de venda.

# Exemplo

- ÉPICO
  - Manter imagens de cartas.
- HISTÓRIA DE USUÁRIO
  - Como um desenvolvedor da Dataprev
  - Eu quero manter imagens de cartas
  - Para que eu possa manter modelos de cartas
- CRITÉRIOS DE ACEITE
  - 1 - A pesquisa de imagens de cartas permite a utilização de filtros conforme documento #L123.
  - 2 - A inclusão de imagens de cartas segue o modelo #M1897
  - 3 - Ao realizar a inclusão de uma imagem, o sistema gera uma sigla única para cada imagem conforme RN001.
  - 4 - Só poderão ser incluídas imagens nos formatos JPG, GIF e PNG, conforme RN003;
  - 5 - Somente funcionários autorizados poderão acessar as imagens e imprimi-las. RNF001

# Referências Bibliográficas

- PRESSMAN, Roger; MAXIM, Bruce. Engenharia de software: uma abordagem profissional. 8.ed. Bookman, 2016. E-book. Disponível em: [https://integrada\[minhabiblioteca\].com.br/books/978858055349](https://integrada[minhabiblioteca].com.br/books/978858055349)
- SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Prentice Hall, 2011. E-book. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/2613 epub/0>
- Anotações de aula cedidas pelos professores: Jader Marques da Silva, Jean Paul Lopes, Osmar de Oliveira Braz Junior, Richard Henrique de Souza