

# Especificação de Requisitos

Karin Becker

Engenharia de Software N  
Instituto de Informática - UFRGS

Inclui material público ou cedido com permissão de autoria  
dos professores Marcelo Pimenta e Karin Breitmann

# Introdução

# Requisitos

- “A principal medida de sucesso de um sistema de software é o grau em que ele atende o propósito para o qual foi concebido” (Nusebeich & Easterbrook)



# Requisitos

- “A parte mais difícil na construção de um sistema de software é decidir precisamente o que construir. Nenhuma outra parte do trabalho conceitual é tão difícil quanto estabelecer requisitos técnicos detalhados, incluindo todas as interfaces para pessoas, máquinas e outros sistemas de software. Nenhuma outra parte do trabalho prejudica o sistema resultante caso seja feita incorretamente. Nenhuma outra parte é tão difícil de corrigir mais tarde.”
  - F. Brooks



Mundo Real

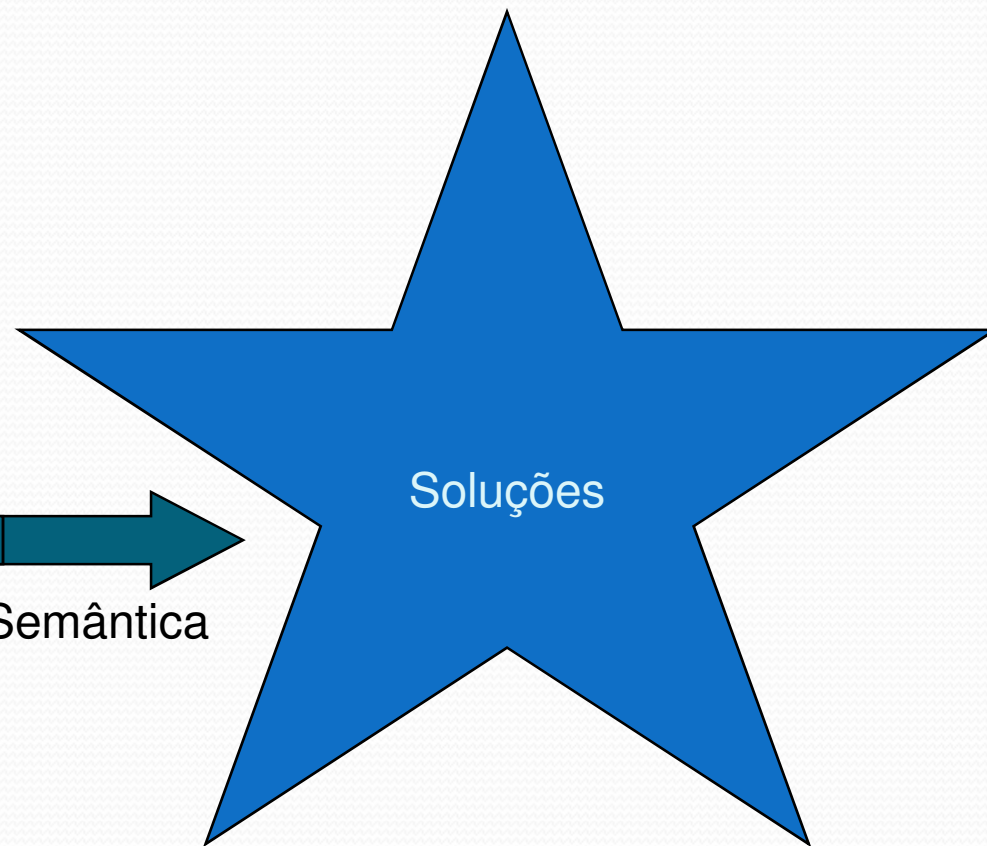


Problemas



Lacuna Semântica

Soluções



Mundo Computacional

# Exercício

- Elaborar uma lista (individual) com 5 características desejadas de uma bicicleta
  - Crie um pseudônimo para você
- Uma outra pessoa irá usar esta lista para adquiri-la para você...
  - Submeta via moodle
    - A identificação do seu amigo, e os critérios
    - uma foto da bicicleta escolhida, mostrando como ela atende os requisitos
- Discussão !!

# Visão Histórica : Análise

- Análise:
  - Estudo do Universo: entender o problema e seu contexto
  - Especificação do sistema: **O QUÊ** o sistema deve fazer
- **Abordagens Principais:**
  - Orientadas a Atividades (Funções): Análise Estruturada (SSA), Análise Estruturada Moderna, SADT,, etc.
  - Orientadas a Dados: ER, JSD, Engenharia da Informação, etc.
  - Orientadas a Estados: Statecharts, Redes de Petri, etc.
  - **Orientadas a Objetos:** OOA (Coad/Yourdon), OMT, Objectory, Fusion, **UML** etc.



# Requisitos: Análise Tradicional

- Parte de um conjunto de premissas que dificilmente ocorrem
  - Há um **problema bem definido** e estável
  - Especificação de Requisitos = **contrato** “congelado” entre analista e cliente ( $\neq$ usuário)
  - **Cliente** conhece bem seu domínio e **sabe bem o que quer**
    - basta saber lhe interrogar corretamente
    - Requisitos são “coletados”
  - **Processo linear** baseado em abstrações: é possível atingir um conjunto completo, coerente e não ambíguo de requisitos mesmo sem a participação do usuário



# Os conflitos ...

- Envolvidos (“Stakeholders”)
  - ▣ Não sabem o que realmente querem
  - ▣ Não conseguem articular o que querem
  - ▣ Pensam que sabem o que querem, mas não o reconhecem quando entregue
- Analistas
  - ▣ Pensam que entendem os problemas dos usuários melhor que os próprios
- Todos
  - ▣ Acreditam que todos estão politicamente motivados



# Requisitos : na verdade ...

- O problema a ser resolvido (e seu domínio) não é claramente delimitado nem descrito e está em constante mudança
  - Os requisitos mudam mesmo durante o desenvolvimento
- O cliente não sabe bem o que quer
  - em geral, suas exigências emergem ou se modificam a partir de interações com o analista e versões preliminares do sistema
  - Suas prioridades mudam, seu entendimento sobre o sistema também
- Nem sempre se chega a especificações corretas e completas
  - deve-se lidar com diferentes graus de incompletude e inconsistência e atribuir prioridades diferentes aos requisitos
- Requisitos existem em um contexto organizacional
  - Existem interesses políticos diversos que influenciam o processo
  - Conflitos, negociações

# Engenharia de Requisitos

- “A engenharia de requisitos estabelece o processo de definição de requisitos como um processo no qual o que deve ser feito deve ser elicitado, modelado e analisado. 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 chamado requisitos é produzido. “ (Júlio Leite)
- Enfoque sistemático para identificação e manutenção dos requisitos
- Requisitos devem ser gerenciados



# Processo: Atividades Genéricas

- **DETERMINAÇÃO:** Entender o problema
  - Utilizar técnicas para identificar requisitos: questionários, entrevistas, documentos...
- **EXPRESSÃO:** Representar o problema
  - Representar nosso entendimento do problemas
  - técnicas para modelagem: DFD, MER, casos de uso...
- **AValiação :** Avaliar a representação
  - Verificar e Validar a informação capturada

# Tipos de requisitos

# Requisitos

- Requisitos são objetivos, funções, propriedades, restrições que o sistema deve possuir/obedecer para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s)
- Requisitos expressam comportamentos e propriedades que o sistema deve apresentar
  - Devem ser **compreendidos** pela equipe de desenvolvimento e **validados** pela parte interessado e comunidade de usuários
  - Deve ser possível **verificá-los**



# Tipos de Requisitos

- Requisitos Funcionais
  - RF são requisitos diretamente ligados a funcionalidade do software.
- Requisitos Não Funcionais
  - RNF são requisitos que expressam restrições que o software deve atender ou qualidades específicas que o software deve ter.
- Requisitos<sup>-1</sup> (Requisitos Inversos)
  - RIN estabelecem condições que nunca podem ocorrer.

# Exemplos

- O sistema deve prover um formulário de entrada para a inserção dos resultados dos testes clínicos de um paciente. (RF)
- O sistema deve emitir um recibo para o cliente, com o tempo máximo de 8 segundos após a transação. (RF, RNF de performance)
- Os pagamentos devem ser realizados através do PagSeguro (RNF)
- O sistema não pode apagar informação de um cliente (RIN).

# Requisitos Funcionais : Exercício

Dê alguns exemplos de RFs para:

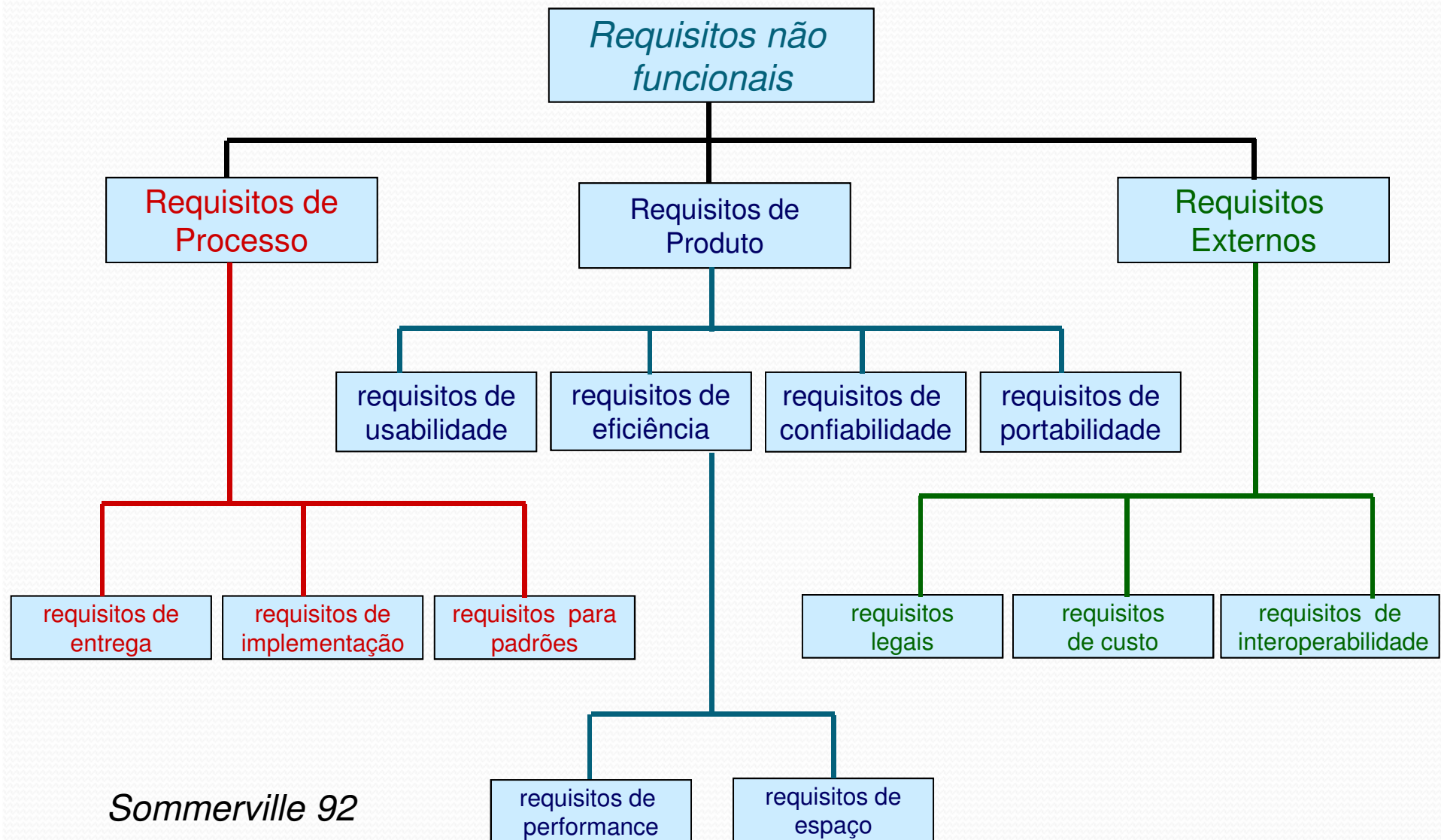
- 1. Sistema de agendamento de reuniões
- 2. Sistema de reserva de sala para reuniões



# Requisitos Não-Funcionais

- Os requisitos não funcionais **são tão importantes** quanto os requisitos funcionais
- Estes requisitos são aqueles que geralmente restringem o comportamento do produto ou condicionam o projeto
- Como qualquer requisito, deve ser verificável
  - deve-se associar uma forma de medida/referência

# Taxonomia



*Sommerville 92*

# Requisitos Não Funcionais

- Requisitos não funcionais devem ser elaborados até que se tornem verificáveis

Requisito Inverificável	Requisito Verificável
“ O banco de dados ZZ deve ser flexível”	<ul style="list-style-type: none"><li>• O banco de dados ZZ deve possuir oito campos por registro.</li></ul>
“MNOP deve ser seguro”	<ul style="list-style-type: none"><li>• MNOP deve parar sua operação se uma pessoa se aproximar a mais de 2 metros de uma de suas partes móveis</li><li>• MNOP deve desligar os aquecedores se a temperatura da água exceder 37 °C</li><li>• MNOP deve estar dentro dos padrões estabelecidos pela norma N567 seção 3.6 para temperaturas de superfícies externas.</li></ul>
“O sistema CE deve processar depósitos rapidamente”	<ul style="list-style-type: none"><li>• O sistema CE deve escanear os dados do usuário e conta de cada folheto de depósito em 2 segundos ou menos”</li></ul>



# Exemplos de formas de mensuração de RNF

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

(Sommerville, 2003)

# Processso

# Requisitos : Ciclo de Vida

- Requisitos **NÃO** são uma preocupação somente no início do projeto
- Requisitos têm papel central em todo o ciclo
  - *Definição de critérios de aceitação e validação (pelos stakeholders)*
  - *Definição de O QUÊ o sistema deve fazer (especificação para a equipe)*
  - *Verificação do sistema sendo desenvolvido*
  - *Alocação de tarefas para a equipe*
  - *Estimativa de custo/esforço/cronograma*
  - *Acompanhamento e Controle do andamento do projeto*
  - *Informações para gerenciamento de **mudanças** (Rastreabilidade, análise de impacto)*
- **Importantíssimo: PA Requirements Management (nível 2) e Requirements Development (nível 3) do CMMI**

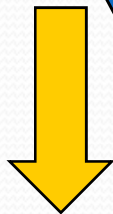
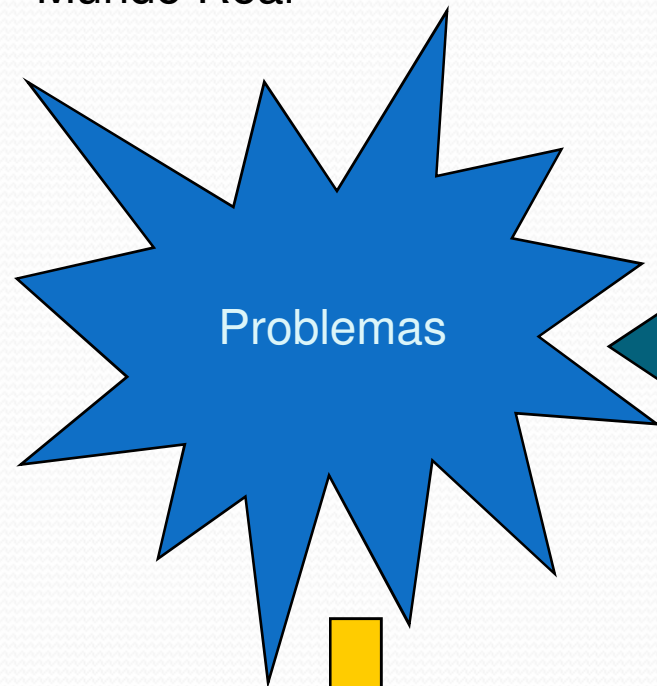


# Atividades Genéricas

- Determinação
  - identificação das fontes de informação
  - eliciação, refinamento e integração de informações
- Expressão
  - representação das informações obtidas
  - representação das várias versões dos requisitos
  - modelagem
- Avaliação
  - avaliação da informação recolhida e representada quanto à correção, completude, coerência para o(s) usuário(s)

# Processo: Determinação

Mundo Real

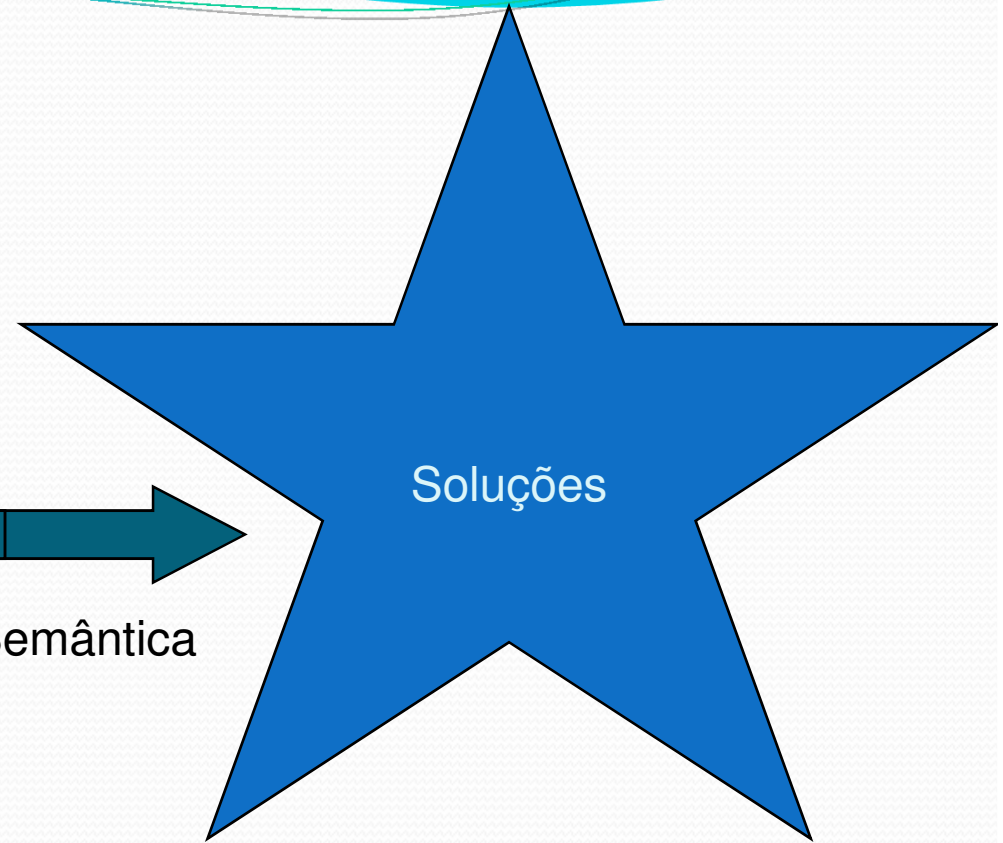


Determinação de  
Requisitos



Lacuna Semântica

Soluções



Mundo Computacional



# Determinação de Requisitos

- Coleta, Captura, Eliciação, Elicitação
- “Eliciação e captura (de requisitos) deveriam ser ilícitos” (Mike Cohn)
  - Requisitos estão prontos, basta fazermos as perguntas certas
- Metáfora do Arrastão (trawling)
  - Malhas de diferentes tamanhos
  - Tempo de vida
  - Múltiplas passadas
  - Pescadores experientes



# Determinação

- Identificar as fontes de informação
- Busca de fatos



# Identificação de Fontes de Informação

- Atores do Universo de Informações (UdI)
  - Clientes
  - Usuários
  - Desenvolvedores
- Documentos
- Livros
- Sistemas de Software



# Heurísticas

- Quem é o cliente?
- Quem é o dono do sistema?
- Existe alguma solução (pacote) disponível?
- Existem sistemas similares?
- Quais são os livros relacionados à aplicação em discussão?
- Existe a possibilidade de reutilizar os artefatos (software)?

# Técnicas para identificar requisitos

- **Técnicas Baseadas em Comunicação**
  - Comunicação Direta: Entrevistas, Reuniões, Workshops
  - Comunicação Indireta: Questionários, Textos de requisitos
- **Técnicas Baseadas em Estudo**
  - Estudo de documentos e formulários
  - Estudo de Sistemas Existentes (Engenharia reversa)
  - Estudo Bibliográfico
- **Técnicas Baseadas em Observação**
  - Observação Direta (pessoal ou vídeo)
  - Observação Verbalizada
  - Observação seguida de diálogo

# Vilão: Conhecimento Tácito

- Trivial
- Internalizado
- Nunca é lembrado!
- Problema de comunicação – Não de requisitos!!!!



# Determinação

Perguntar *por quê?*

*“A cafeteira deve ser feita de aço”*

- qual a razão disto?
- pode me explicar o porquê?
- qual o pensamento por trás disto?

# Determinação

*“Porque se for de vidro, pode quebrar”*

## Requisito real

- A cafeteira deve ser feita de material inquebrável
  - Plástico
  - Poliuretano
  - Até mesmo aço
- Os usuários misturam a solução com os requisitos
  - Separar NECESSIDADE da solução proposta ou atual!

# Exercício

- “a cafeteira tem uma luz vermelha que pisca quando está ligada, quando a água chega na temperatura certa ela fica ligada (sem piscar)”
  - Quais seriam os “reais” requisitos?

Dica: Separar requisito de solução/implementação



# Técnicas para identificar requisitos

- **Técnicas Baseadas em Comunicação**
  - Comunicação Direta: Entrevistas, Reuniões, Workshops
  - Comunicação Indireta: Questionários, Textos de requisitos
- **Técnicas Baseadas em Estudo**
  - Estudo de documentos e formulários
  - Estudo de Sistemas Existentes (Engenharia reversa)
  - Estudo Bibliográfico
- **Técnicas Baseadas em Observação**
  - Observação Direta (pessoal ou vídeo)
  - Observação Verbalizada
  - Observação seguida de diálogo

# Entrevistas

- Popular
- Escolha bem quem entrevistar
- Perguntas abertas e livres de contexto
- Tomar cuidado com a diferença entre desejo e necessidade
- +
  - contato direto com atores
  - possibilidade de validação imediata
- -
  - conhecimento tácito
  - diferenças culturais

# Reuniões

- Extensão da entrevista
  - Brainstorm
  - Workshop de Requisitos
- +
  - dispor de múltiplas opiniões
  - criação coletiva
  - Prototipação, storyboard → stakeholders
- -
  - dispersão
  - custo



# Observação

- +
  - baixo custo
  - pouca complexidade da tarefa
  - Palavras tomam outro sentido
- -
  - dependência do ator (observador)
  - superficialidade decorrente da pouca exposição ao universo de informações

# Leitura de Documentos

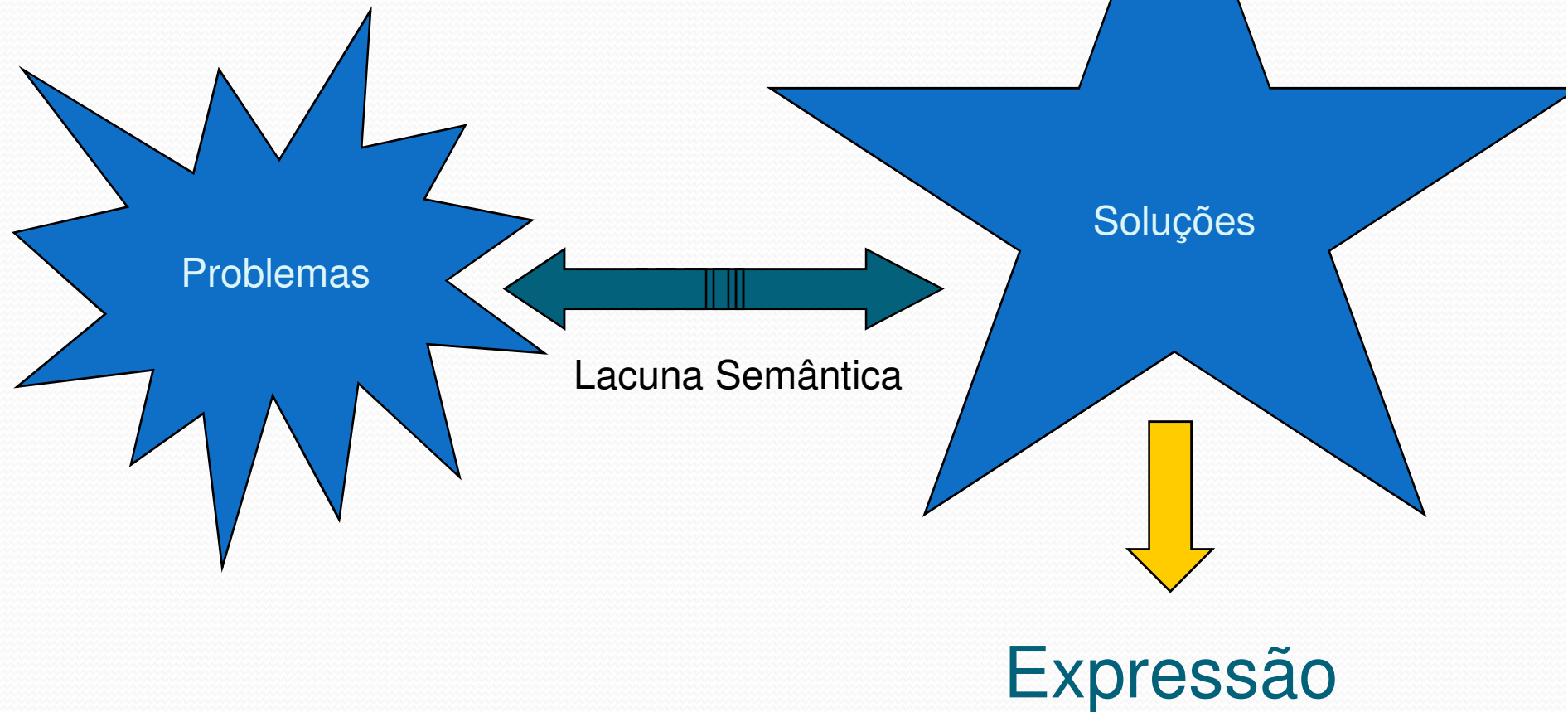
- +
  - facilidade de acesso às fontes de informação
  - volume de informação
- -
  - dispersão das informações
  - volume de trabalho requerido para identificação dos fatos
  - Representação estática
  - Dependente de expressão e de interpretação

# Processo: Expressão



Mundo Computacional

Mundo Real



# Expressão: Sentenças

- O sistema deve + [verbo + objeto | frase verbal ] + [complemento de agente | nulo] + [condições | nulo]
- Três classes de sentenças: {Entrada, Saída, Mudança de Estado}
- Verbo/Frase verbal é uma frase que expressa a funcionalidade do requisito
- Complemento de agente é a identificação de um agente relacionado com o requisito. Agente pode ser uma pessoa, uma instituição, um grupo ou um dispositivo físico externo ao software
- As sentenças podem ser de três tipos: funcionais, não-funcionais e inversas.

# Sentenças de Requisitos

- O sistema deve emitir um recibo para o cliente.
- O sistema deve emitir o recibo para o cliente em no máximo 2 segundos.
- O sistema deve transformar uma fita disponível em fita emprestada, quando a fita for alugada pelo cliente.
- O sistema deve cadastrar bibliotecários.
- O sistema deve verificar a identidade do bibliotecário.
- O sistema não pode deixar que um livro fique na reserva mais de um mês.
- O sistema deve tornar um livro em livro emprestado, quando um usuário pegar este livro emprestado.



# Atributos de uma boa especificação

- Clareza
- Não ambígua
- Completa
- Simples
- Bem escrita

# Expressão

- Histórias
  - “como cliente, gostaria de poder selecionar produtos por categorias para poder adquiri-los”
  - “como aluno, gostaria de poder emitir meu histórico escolar”
- Sem formato definido
- Adotados em métodos ágeis, como recurso de acomodação à mudança
- XP Essencial = CCC
  - Exige a conversa
  - Exige a confirmação
  - Casos de testes tornam-se a expressão do requisito
  - Documentados de forma mais ou menos formal

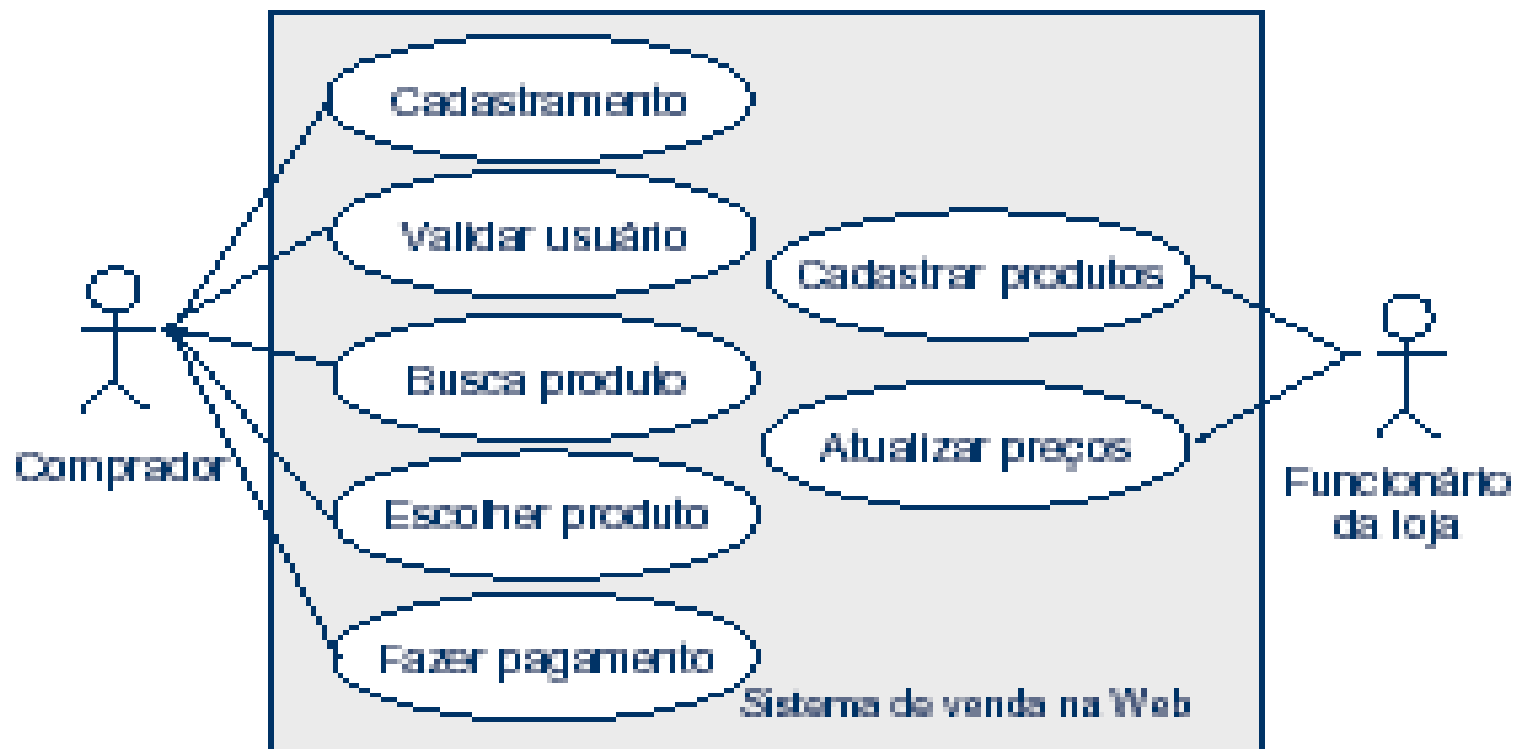
# Modelar

- Para que servem modelos?
  - Compreender
  - Organizar
  - Abstrair, concentrar no que é importante
  - Comunicar
  - Testar idéias



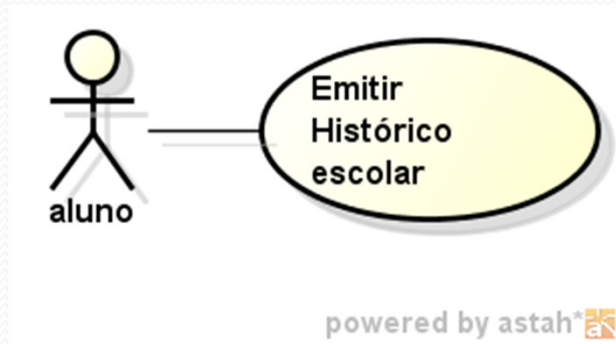
# Expressão : Casos de Uso

- Representam a funcionalidade do sistema do ponto de vista de seus usuários



# Casos de Uso

- Diagrama de casos de uso são semelhantes à histórias no seu poder semântico
  - “Como aluno, desejo emitir meu histórico escolar”
    - Precisa conversar, e documentar, possivelmente através de casos de teste
  - Caso de Uso Emitir Histórico Escolar
    - Diagrama tem de vir acompanhado de descrição



# Descrição de Casos de Uso

- Descrição de casos de uso são informais
- Podem ser usados em vários níveis de abstração
  - Grau de detalhe reflete estágio do projeto de desenvolvimento
  - Conteúdo pode ser adequado às necessidades da aplicação
  - Formato é definido pela organização
  - Diferentes estilos de descrição



# Descrição de Casos de Uso: Tipos

- *Preliminar* (ou alto-nível)
  - Conceitual, abstrato e pouco detalhado (independente de implementação)
  - usados na especificação de requisitos e delimitação de escopo no início da análise
- *Essencial* (ou expandido)
  - Conceitual (independente de implementação)
  - Detalhado em termos de funcionalidade
  - Descreve cenários de interação entre ator e caso de uso
- *Real* (ou concreto)
  - Concreto (dependente de tecnologia)
  - Detalhado em termos de funcionalidade, comportamento (operação) e referências a interface

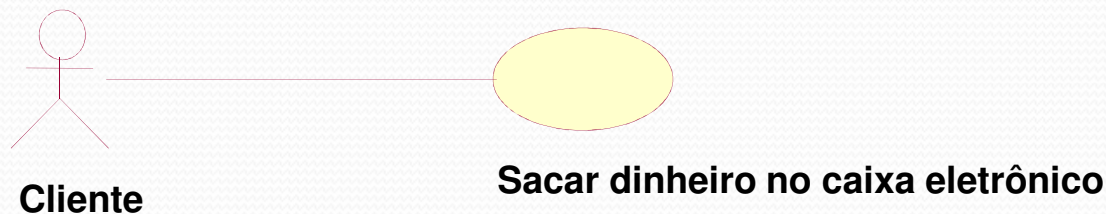
# Exemplo : Caso de Uso Preliminar

**Identificação:** UC1

**Caso de uso:** Sacar dinheiro no caixa eletrônico

**Ator:** Cliente

**Descrição:** Permite ao cliente sacar quantia especificada, uma vez validados cartão e senha. A transação deve ser registrada, o saldo atualizado, e dinheiro liberado.





# Exemplo : Caso de Uso Essencial

**Identificação:** UC1

**Caso de uso:** Sacar dinheiro no caixa eletrônico

**Ator:** Cliente

**Descrição:** Permite ao cliente sacar quantia especificada, uma vez validados cartão e senha. A transação deve ser registrada, o saldo atualizado, e dinheiro liberado.

**Pré-Condições:** o Cliente possui cartão do banco e senha cadastrada.

**Pós-Condições:** lançada a transação na conta do Cliente, atualizado o saldo da conta corrente e liberado o dinheiro.

## Seqüência Típica de Eventos (Fluxo Básico):

### Ação do ator

1. Este caso de uso começa quando o Cliente insere o cartão do banco no caixa eletrônico
2. O Cliente informa a sua senha
4. O Cliente informa o valor do saque

### Resposta do sistema

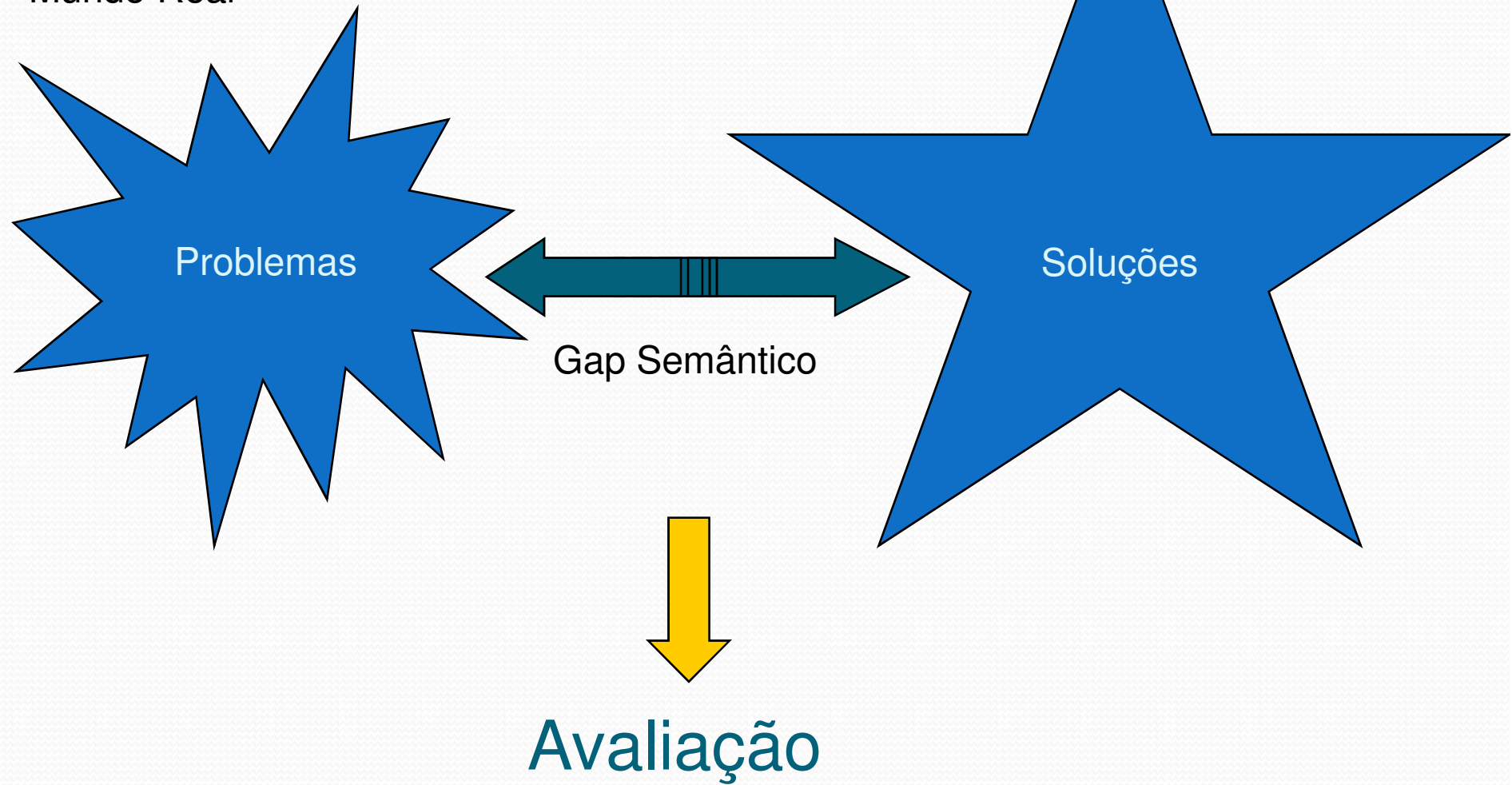
3. O sistema valida a conta corrente e senha do Cliente, autorizando a operação
5. O sistema autoriza o saque e lança o débito na conta corrente do Cliente
6. O sistema libera o dinheiro



# Avaliação

Mundo Computacional

Mundo Real



# Verificação X Validação

## **Verificação**

**estamos construindo o  
produto de maneira  
certa?**

**(em relação a outros  
produtos)**



**entre modelos**

## **Validação**

**estamos construindo o  
produto certo?  
(em relação a intenção dos  
fregueses)**



**entre o UdI e um modelo**



# Avaliação

- Definição de critérios de aceitação
  - Casos de teste de aceitação
  - Evolução de casos de teste para guiar as demais atividades
- Discussão de Cenários
- Protótipos
- Inspeção
- Revisão

# Para saber mais ...

- Leituras (fortemente) recomendadas
  - Sommerville, Ian. Engenharia de software. 8ª edição. Pearson Education. São Paulo:, 2007.
  - Capítulo 6 – discute o conceito de requisitos, os tipos de requisitos, e o conteúdo de um documento de especificação de requisitos
  - Capítulo 7 – discute o processo de engenharia de requisitos
- Outras alternativas
  - Pressman, Roger. Engenharia de Software. Ed. Makron Books, 2006.
  - Capítulo 10 – as atividades da engenharia de requisito
  - Capítulo 11 (seções 11.1 e 11.2) : o processo de coleta e descrição de requisitos