

# Requisitos

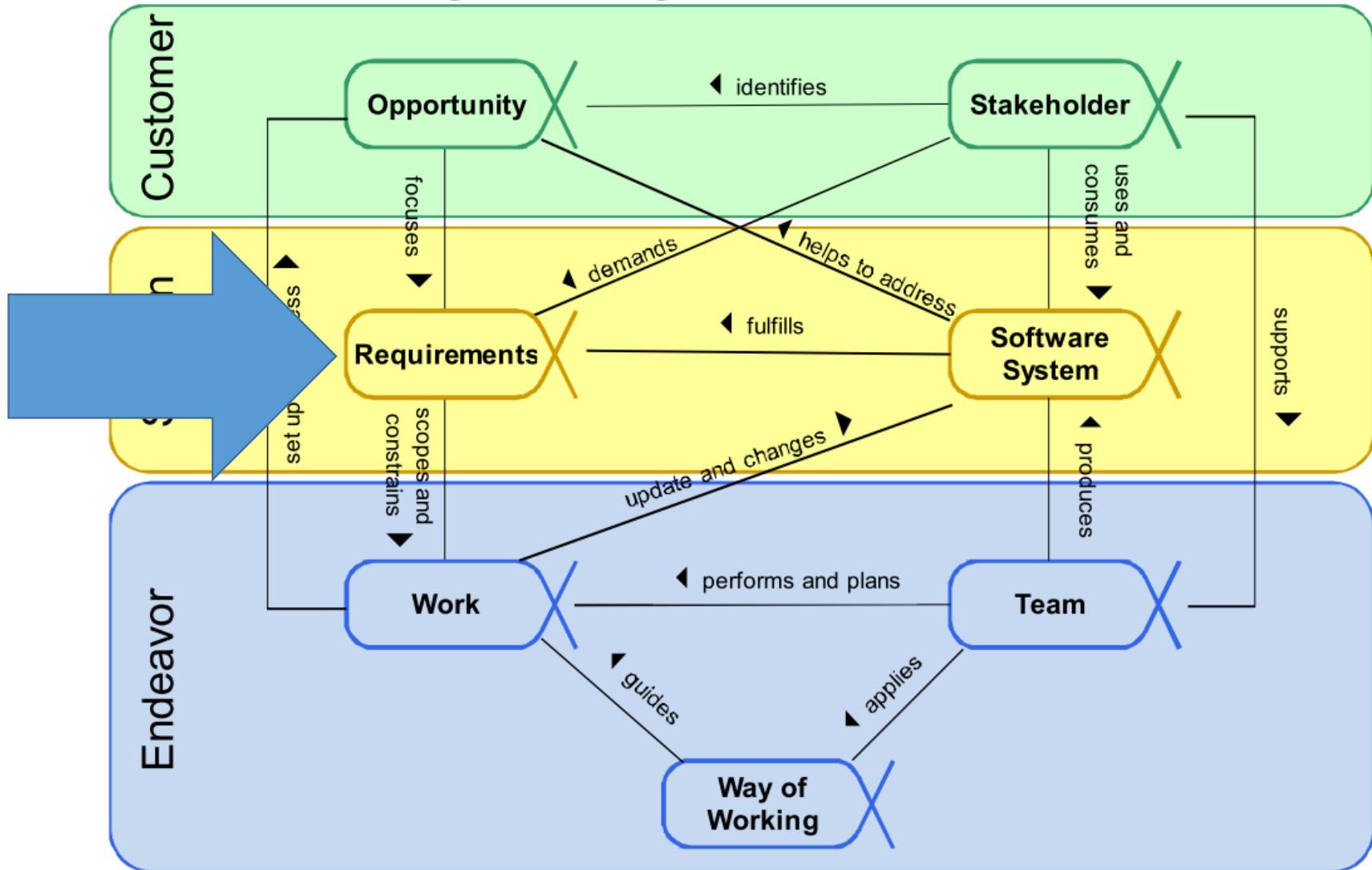
Geraldo Xexéo  
DCC/IM & PESC/COPPE  
Universidade Federal do Rio de Janeiro

xexeo@cos.ufrj.br

<http://www.cos.ufrj.br/~xexeo>

<http://www.xexeo.net/>

# O Kernel - Alphas



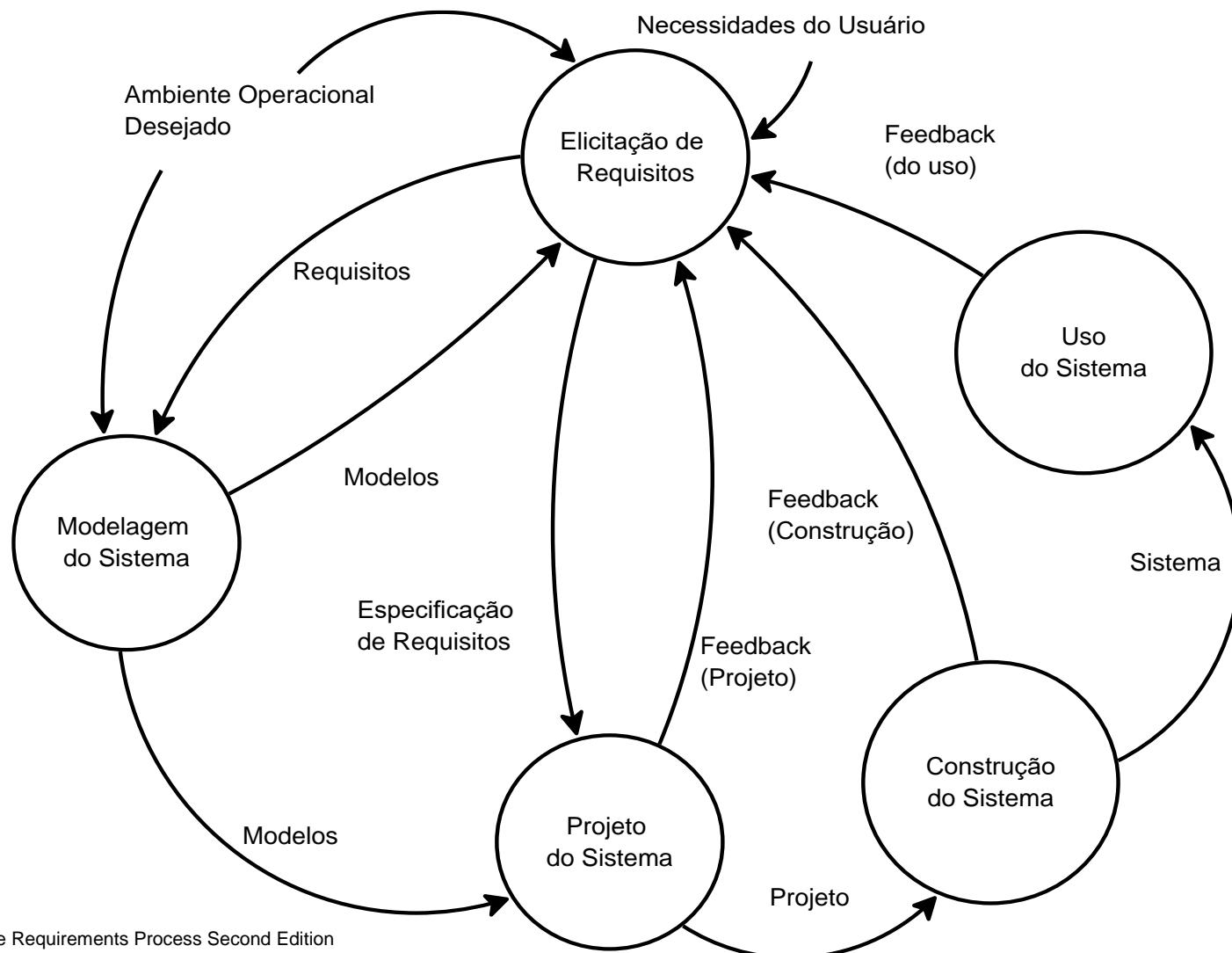
# **Motivação para Atenção aos Requisitos**

Geraldo Xexéo, D.Sc.

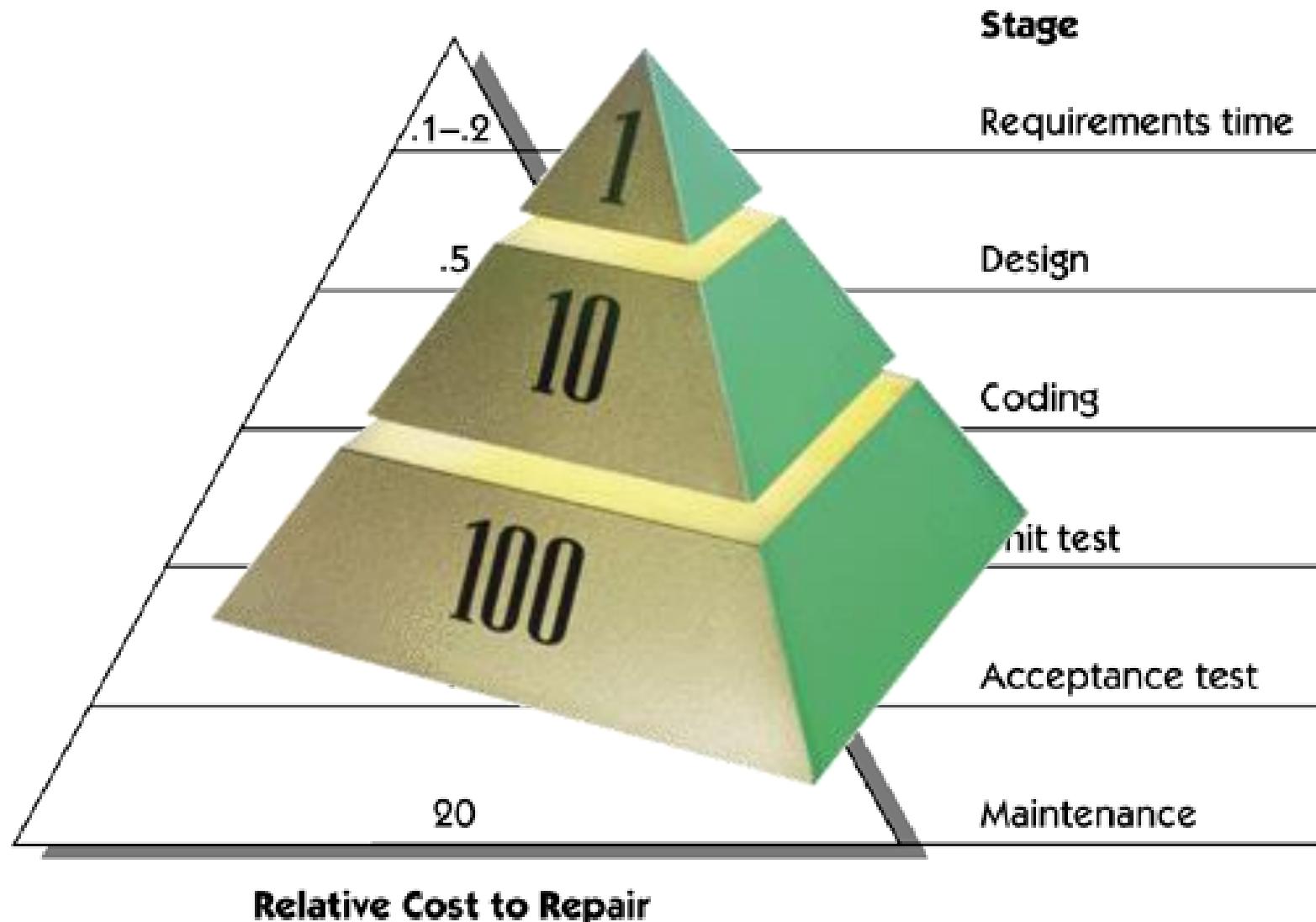
DCC/IM/UFRJ

PESC/COPPE/UFRJ

# Papel dos Requisitos



# Custo de Defeitos



# **Custo de Erros de Requisitos**

- São o tipo mais comum de erro
- São o tipo mais caro para consertar
- 25% a 40% do custo de um projeto provêm de erros de requisitos

# Mudança

- Requisitos inevitavelmente mudam durante o desenvolvimento e após.
- Não se deve esperar defini-los completamente, mas gerenciar sua mudança.
- A única constante é a mudança

# **Definição de Requisitos**

# Requisito

- Uma **condição ou capacidade necessária** para um **usuário** resolver um problema ou atingir um objetivo
- Uma **condição ou capacidade** que precisa ser possuída ou alcançada por um sistema ou componente de uma sistema para **satisfazer** um contrato, padrão, especificação ou outro documento formalmente imposto
- Uma representação documental de uma condição ou capacidade no sentido das duas definições anteriores
- IEEE Std 610.12-1990

# O Problema do Termo

- Muitas definições, de acordo com o processo, modelo de processo, ou padrão
  - Termo usado de forma inconsistente na indústria
  - As vezes, altamente abstrato
  - Outras vezes, detalhado

# Requisito

- No processo unificado: **casos de uso**
- No SCRUM, XP: **histórias de usuário**
- Tudo que é pedido do software em algum momento é um requisito para os momentos posteriores
  - Algumas coisas são requisitos para os momentos anteriores (mudança)

# **Formas de Requisitos**

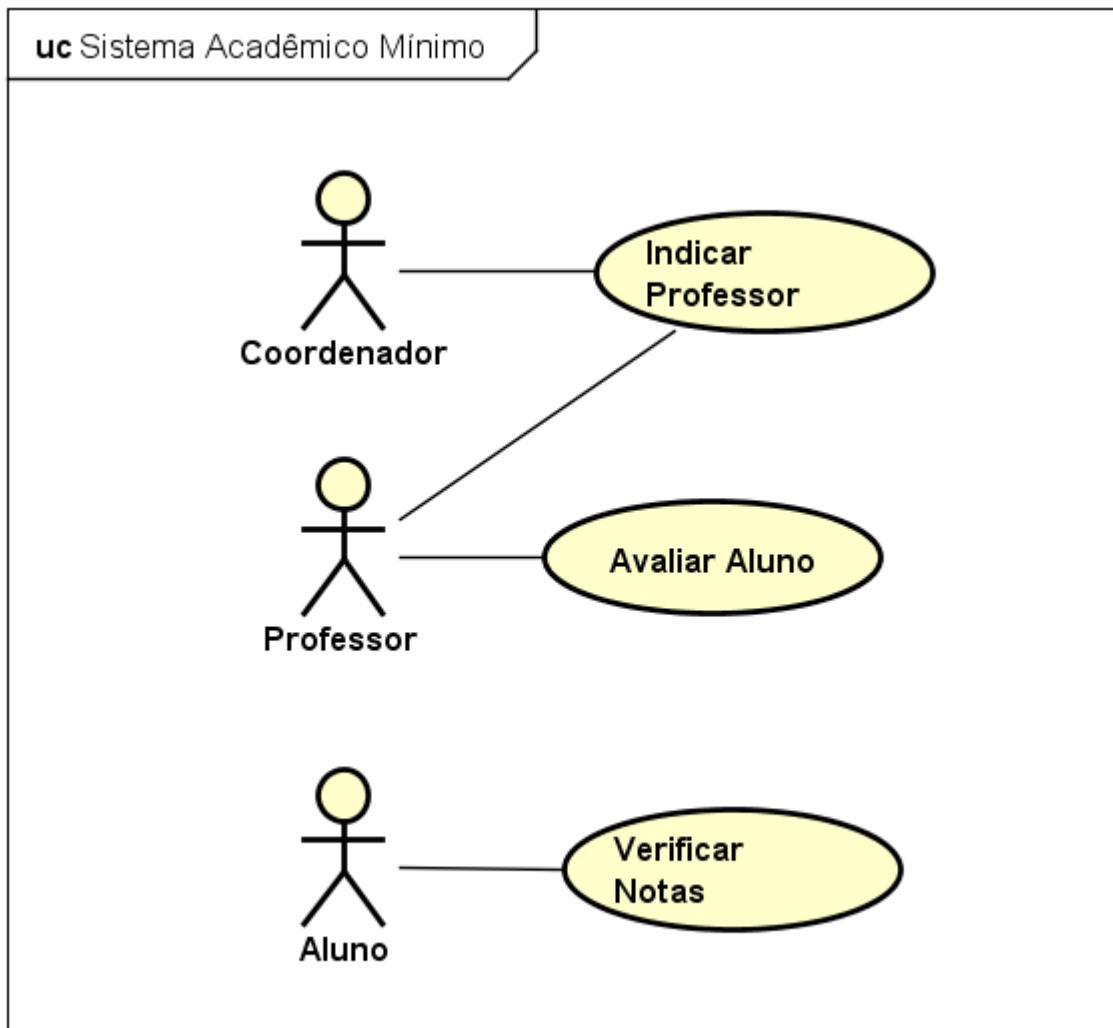
# Formas dos Requisitos

- Sentenças em linguagem natural
- Sentenças em linguagens naturais restritas/estruturadas
  - Usando formas padrão
  - Ex: Volere
- Linguagens de Descrição de Projeto
- Notações Gráficas
  - UML
- Especificações Matemáticas
  - Linguagens Formais
    - VDM, Z
  - Linguagens Algébricas
- Árvores de Requisitos

# **Requisitos como Sentenças**

- O Sistema deverá permitir que o coordenador indique um professor para uma turma
- O Sistema deverá emitir, ao fim do mês, o relatório de vendas por loja, individualizado por SKU
- O Sistema deverá possuir uma tela de ajuda para cada tela, especificando para que servem campos e botões
- O Sistema deverá estar pronto antes do dia 1/12/2017

# Requisitos como Gráficos



# Cartão Volere

Requirement #: 75

Requirement Type: 9

Event/Use Case #: 7, 9

Description: The product shall record all the roads that have been treated.

Rationale: To be able to schedule untreated roads and highlight potential danger.

Originator: Arnold Snow, Chief Engineer

Fit Criterion: The recorded treated and untreated roads shall agree with the drivers' road treatment logs.

Customer Satisfaction: 3

Customer Dissatisfaction: 5

Priority:

Conflicts:

Supporting Materials:

History: Created February 29, 2006

**Volere**

Copyright © Atlantic Systems Guild

# Req. Como Especificações Formais

```
types

Trace = seq of [Event];

Event = <A> | <V>;

state Pacemaker of
    aperiod : nat
    vdelay : nat
init p == p = mk_Pacemaker(15,10)
end
operations

Init: nat * nat ==> ()
Init(aperi,vdel) ==
    (aperiod := aperi;
     vdelay := vdel);

IdealHeart: () ==> Trace
IdealHeart() ==
    return [ if i mod aperiod = 1
        then <A>
        elseif i mod aperiod = vdelay + 1
        then <V>
        else nil
    | i in set {1,...,100}];
```

# Narrativa de um Caso de Uso

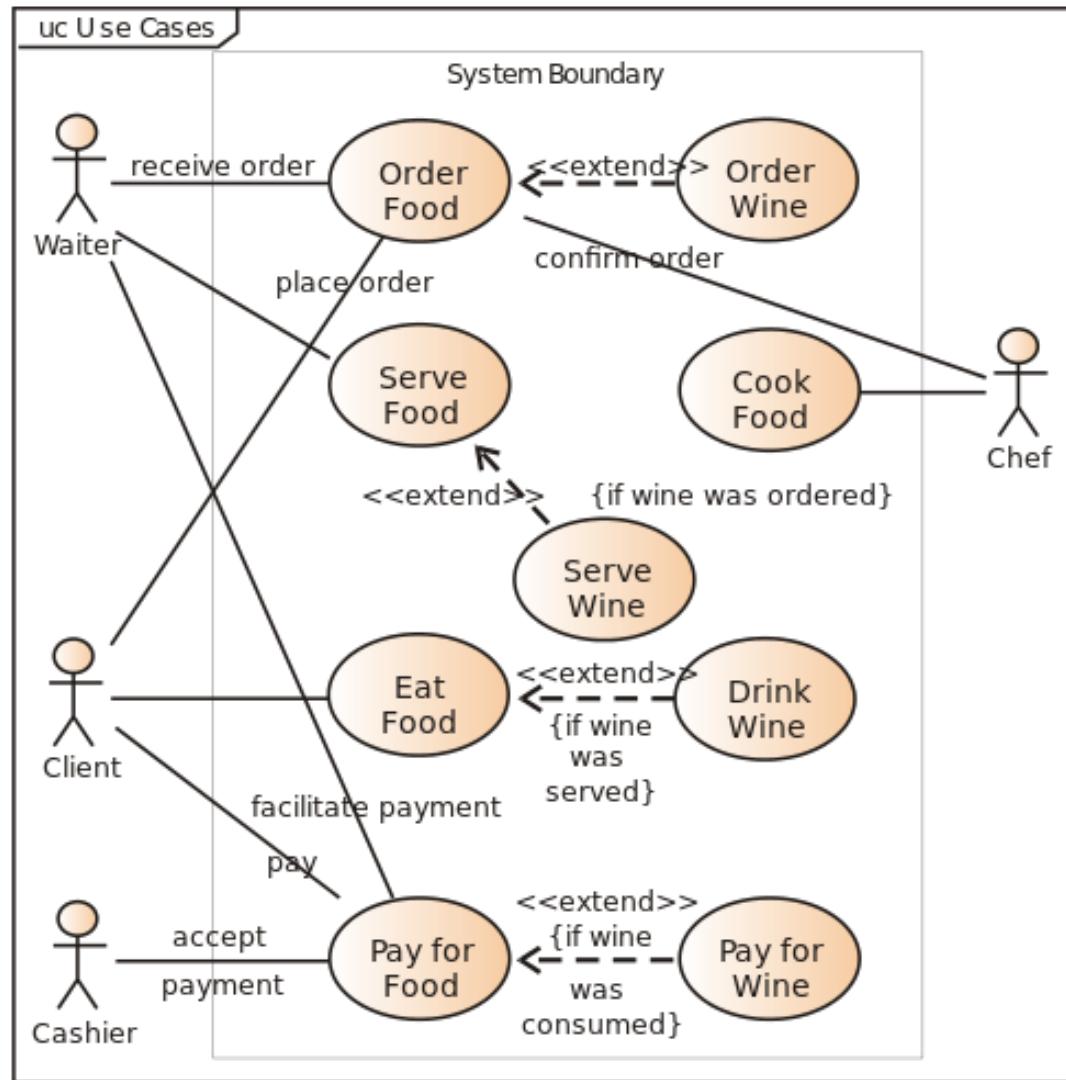
## basic flow

1. insert card
2. validate card
3. select cash withdrawal
4. select account
5. confirm availability of funds
6. return card
7. dispense cash

## alternative flows

- A1 invalid card
  - A2 non-standard amount
  - A3 receipt required
  - A4 insufficient funds in ATM
  - A5 insufficient funds in account
  - A6 would cause overdraft
  - A7 card stuck
  - A8 cash left behind
- etc.

# Diagrama de Casos de Uso



This file is licensed under the [Creative Commons Attribution-Share Alike 3.0 Unported](#) license.

Attribution: Kishorekumar 62

# Caso de Uso Detalhado

## Caso de Uso:

Efetuar Reserva.

### Descrição Geral:

Descreve o processo de reserva, para um cliente, de um determinado apartamento, em um determinado hotel, num determinado período.

### Atores:

Operadora e Cliente.

### Início:

Operadora solicita uma nova reserva.

## Fluxo Típico

Nº	Ação
1	Sistema solicita cidade.
2	Operadora informa cidade desejada.
3	Sistema apresenta os hotéis da região informada.
4	Sistema solicita hotel e período.
5	Operadora informa hotel desejado e período.
6	Sistema apresenta disponibilidade de apartamentos e o preço unitário.
7	Operadora seleciona o apartamento desejado e informa o período.
8	Sistema apresenta valor total da reserva.
9	Operadora confirma a reserva.
10	Sistema solicita a identificação do cliente.
11	Operadora informa a identificação do cliente.
12	Executar <i>Solicitar Autorização de Cartão de Credito</i> .
13	Sistema emite o Voucher para o cliente.

## Fluxos Alternativos

**Alternativa 1:** Caso nenhum hotel seja encontrado na cidade que o Cliente solicitou.

Nº	Ação
3	Encerrar Caso de Uso.

**Alternativa 2:** Caso não exista um apartamento disponível para reserva no período informado.

Nº	Ação
6	Encerrar Caso de Uso.

**Alternativa 3:** Caso o Cliente desista da reserva.

Nº	Ação
7	Encerrar Caso de Uso.

**Alternativa 4:** Caso o Cliente não aceite o valor da reserva.

Nº	Ação
9	Encerrar Caso de Uso.

**Alternativa 5:** Caso a Administradora do Cartão não aceite o débito. Após o passo 12 do Fluxo Típico, acrescentar:

Nº	Ação
13	Sistema informa que o débito no Cartão de Credito não foi aceito.
14	Encerrar Caso de Uso.

# Hierárquica – Organizando funcionalidades

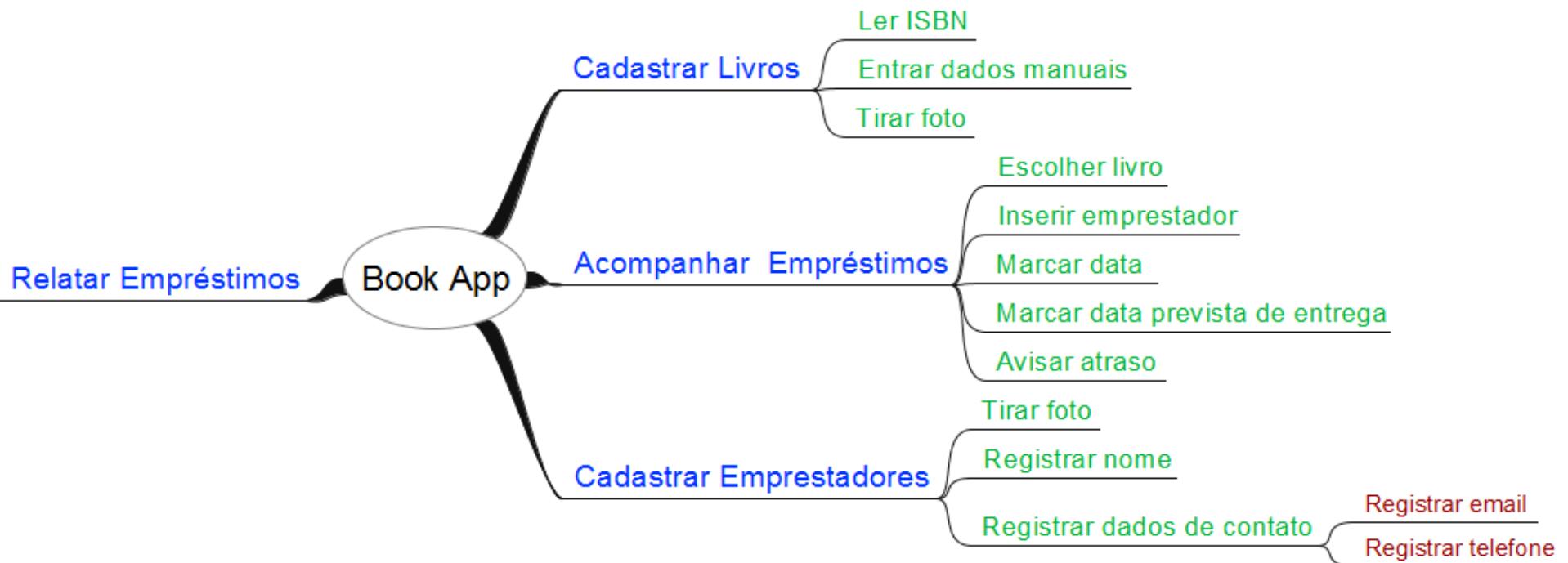
Book App

- Relatar Empréstimos
- Cadastrar Livros
  - Ler ISBN
  - Entrar dados manuais
  - Tirar foto
- Acompanhar Empréstimos
  - Escolher livro
  - Inserir emprestador
  - Marcar data
  - Marcar data prevista de entrega
  - Avisar atraso
- Cadastrar Emprestadores
  - Tirar foto
  - Registrar nome
  - Registrar dados de contato
    - Registrar email
    - Registrar telefone

# Hierárquica – detalhando funcionalidades

- R 1. O sistema deverá permitir o registro de um empréstimo de livro
  - R1.1 O sistema deverá permitir que o livro seja cadastrado no momento do registro do empréstimo
  - R1.2 O sistema deverá permitir que a pessoa que pega o livro emprestado seja cadastrada no momento do registro do empréstimo
  - R1.3 O sistema deverá permitir que seja tirada uma foto da pessoa com o livro como parte dos dados de empréstimo
    - R.1.3.1 O sistema deverá guardar a foto apenas até a devolução do livro

# Mind Map



# CRC Cards

## Candidate, Responsibilities, Collaborators

### MessageBuilder

Builds message from selections

Presents guesses to user

Controls the pacing

Message

Presenter

### MessageBuilder

Purpose: The MessageBuilder is a hub of activity in the application. It coordinates the timing, the presentation of guesses, the message construction. It centralizes control and is a core element of the control architecture.

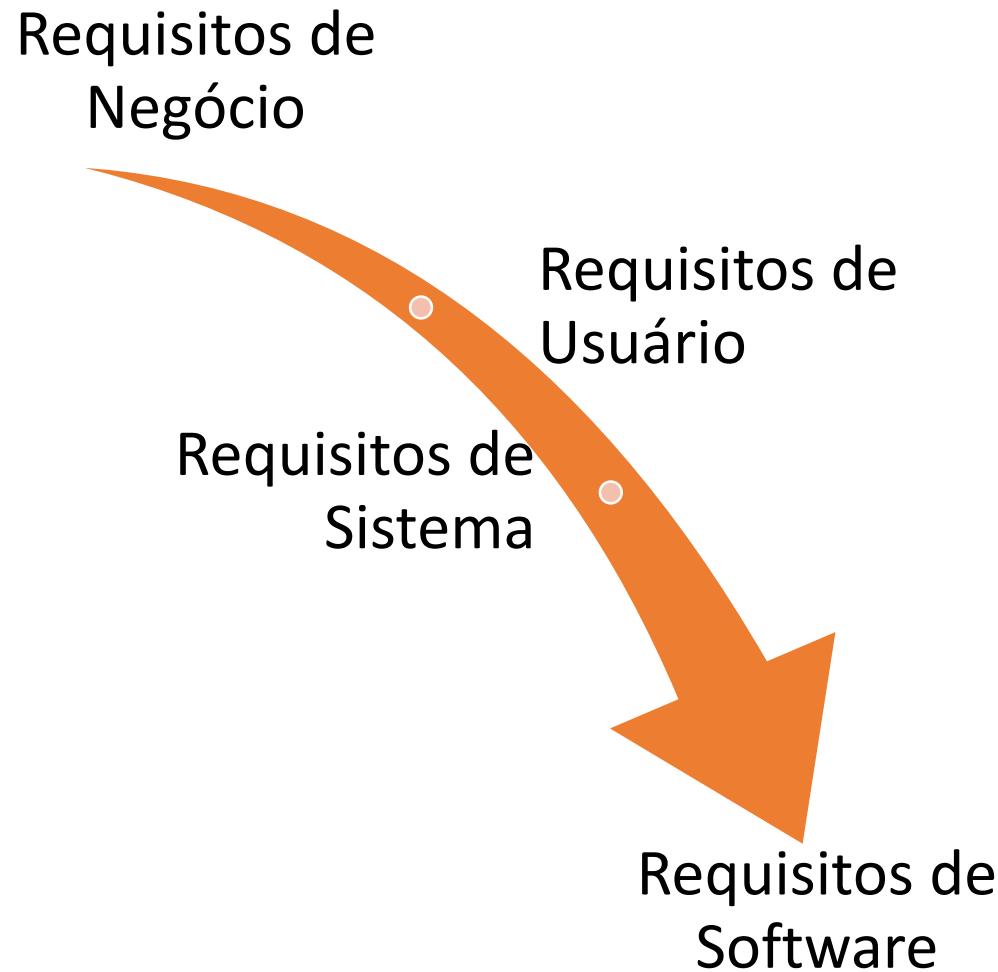
# Product Backlog (Scrum)

## To Do List

ID	Story	Estimation	Priority
7	As an unauthorized User I want to create a new account	3	1
1	As an unauthorized User I want to login	1	2
10	As an authorized User I want to logout	1	3
9	Create script to purge database	1	4
2	As an authorized User I want to see the list of items so that I can select one	2	5
4	As an authorized User I want to add a new item so that it appears in the list	5	6
3	As an authorized User I want to delete the selected item	2	7
5	As an authorized User I want to edit the selected item	5	8
6	As an authorized User I want to set a reminder for a selected item so that I am reminded when item is due	8	9
8	As an administrator I want to see the list of accounts on login	2	10
<b>Total</b>		<b>30</b>	

# **Evolução dos Requisitos**

# Evolução dos Requisitos



# **Requisitos do Negócio**

- Por que o projeto deve ser feito
- Justificativa do projeto para o negócio
- Visão dirigida por objetivos de negócio
- Descrição de alto nível do objetivo do projeto/produto

# Requisito do usuário

- É algum comportamento ou característica que o usuário deseja do software ou o sistema como um todo;
- o que o usuário quer.
- São escritos pelo próprio usuário ou levantados por um analista de sistemas que consulta o usuário.
- “Normalmente na forma de sentenças, em linguagem natural, mais diagramas ou desenhos, de que serviços o usuário espera que o sistema forneça e que restrições o sistema terá para operação”
  - Sommerville

# Requisito do Sistema

- É algum comportamento ou característica exigido do sistema como um todo, incluindo hardware e software.
- O comportamento desejado do sistema.
- São normalmente levantados por engenheiros ou analistas de sistemas, refinando os requisitos dos usuários e os transformando em termos de engenharia.
- “Normalmente descrições mais detalhadas das funções, serviços e restrições operacionais [...] definindo exatamente o que deve ser implementado. Pode ser parte do contrato entre comprador e desenvolvedores de software”
  - Sommerville

# Exemplo

- Usuário
  - O AppBook deverá gerar mensalmente um relatório com cada livro emprestado, indicando quem está com o livro e por quanto tempo.
- Sistema
  - No último dia útil de cada mês, o relatório de livros emprestados, com quem está e a quanto tempo deve ser gerado
  - O sistema deve gerar automaticamente o relatório e enviá-lo por email às 17:30 do último dia útil do mês.
  - O relatório deve citar todos os livros emprestados, seguido da pessoa de posse do livro, seu email, seu telefone e o número de dias que a pessoa está com o livro
  - Se o livro estiver emprestado há mais de 2 meses, deverá aparecer sublinhado

# **Requisito do Software**

- É algum comportamento ou característica que é exigido do software
- Normalmente levantado por analistas de sistemas

# Requisitos de Domínio

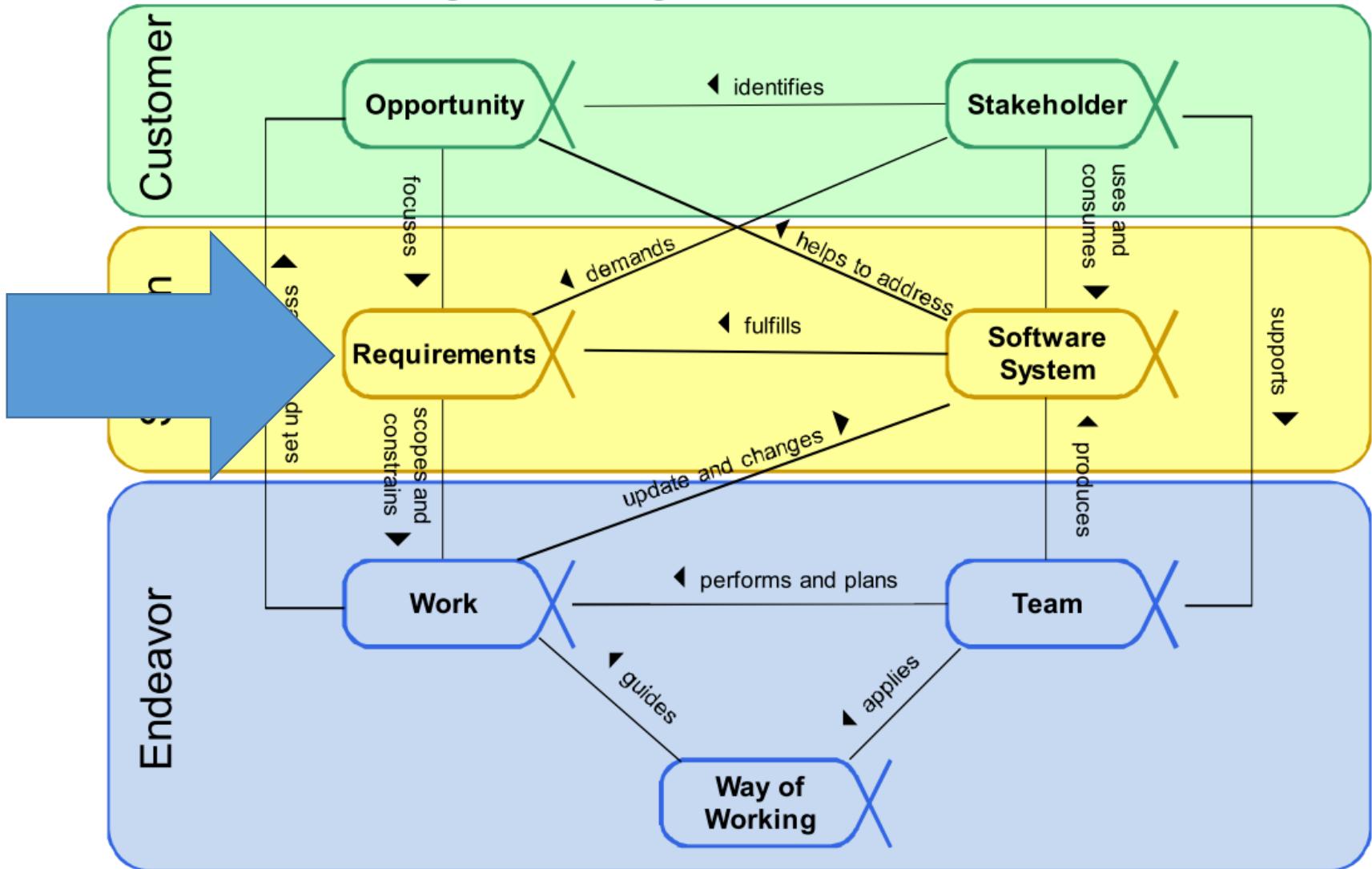
- Um Domínio de Aplicação é um campo de estudo que possui uma terminologia ou jargão, e um conjunto comum de requisitos e funcionalidades
  - Software Financeiro
  - Software Médico
    - Softaware para prontuário médico
- Requisitos de domínio são obtidos diretamente do domínio de aplicação
  - Leitura de documentos, manuais, leis, procedimentos e padrões
  - Experiência do analista
    - Analistas de Domínio, Analistas de Negócio
  - Elicitados com o usuário
- Exige conhecimento do domínio

# **Documentos de requisitos**

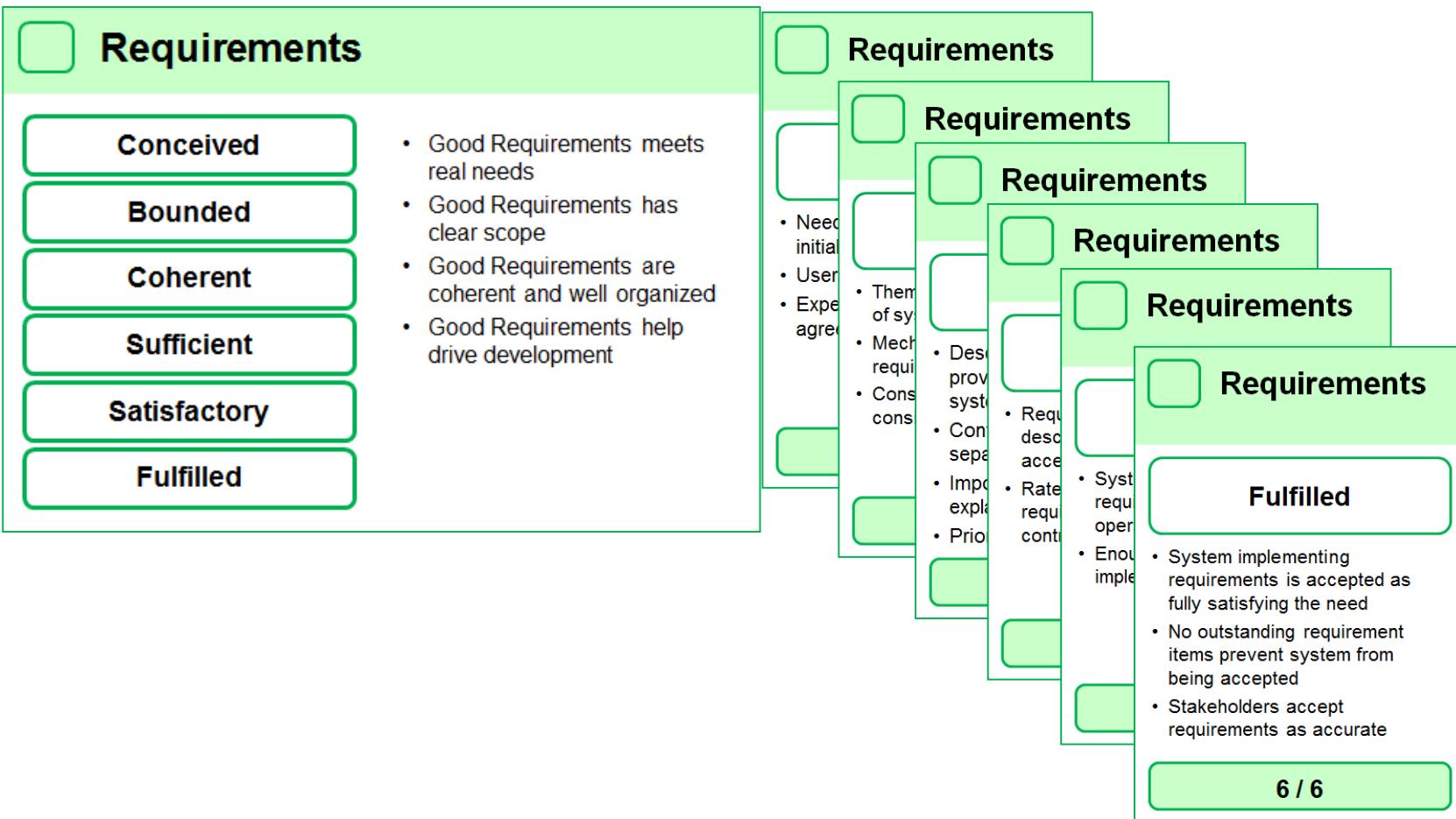
- Definição dos requisitos: listagem completa de tudo que o cliente espera que o sistema proposto faça
- Especificação dos requisitos: redefine os requisitos em termos técnicos apropriados para o desenvolvimento do projeto do sistema
- Gerência de configuração: correspondência direta entre os dois requisitos

# **Requisitos no SEMAT**

# O Kernel - Alphas



# Orientado a resultado



Γ

# **Tipos de Requisitos**

Λ

**Funcionais/Não  
Funcionais**

Λ

Λ

# Requisitos funcionais versus não-funcionais

- Funcional: descreve uma interação entre o sistema e seu ambiente
  - Exemplos:
    - o sistema dever ter comunicação com um sistema 'x' externo
    - Quais estados devem ser encontrados para uma mensagem ser enviada
  - Coisas que o sistema deve fazer
- Não-funcional: descreve uma restrição do sistema que limita nossas opções para criar uma solução para o problema
  - Exemplos:
    - Contracheques distribuídos em não mais que quatro horas depois de os dados iniciais terem sido lidos
    - Qualidades que o sistema deve ter

# **Requisito Funcional**

- Representa algo que o sistema deve fazer,
- Uma função esperada do sistema que agregue algum valor a seus usuários.
- Exemplos típicos:
  - a emissão de relatórios
  - a realização e manutenção de cadastros

# **Exemplos de Req. Funcionais**

- O sistema deverá fornecer ao gerente de entregas a capacidade de escolher o entregador que entregará cada pacote.
- O sistema deverá permitir que o gerente de estoque busque por itens disponíveis no estoque
- O sistema deverá notificar o operador quando a temperatura exceder o limite máximo especificado
- O sistema deve guardar um registro de todas as reclamações feitas pelos clientes do help-desk

# Categorias de Funções

- Evidente
  - deve ser executada com conhecimento do usuário
- Oculta
  - deve ser executada mas não necessariamente visível para os usuários

- Enfeite/Decoração
  - opcional, sua adição não afeta significativamente o custo ou outras funções

# Requisito Não Funcional

- Fala da forma como os requisitos funcionais devem ser alcançados.
- Define propriedades e restrições do sistema.
  - Muitos requisitos não funcionais são também requisitos de qualidade, como exigências de desempenho e robustez.
  - Outros são restrições ou exigências de uso de uma ou outra tecnologia.
- Muitas vezes chamados de “requisitos de qualidade” ou “atributos de qualidade”
- Qualidade fala de atendimento a expectativas, requisito fala de atendimento a especificações
- Difíceis de tratar formalmente

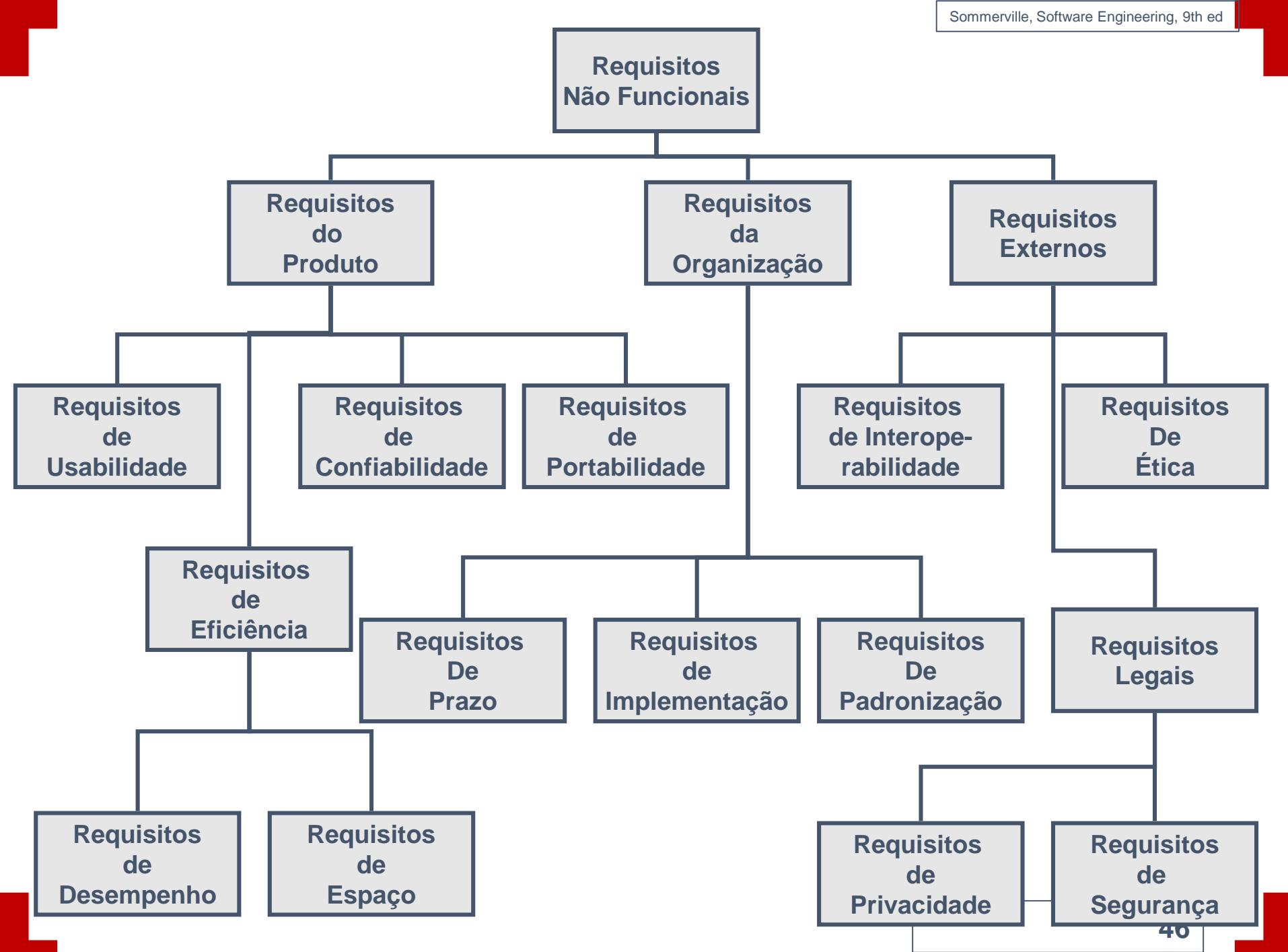
# **Tipos de Requisitos Não-Funcionais**

- Atributos de Qualidade
- Restrições de Projeto
- Restrições de Implementação
- Interfaces Externas

# Métricas

- Requisitos não funcionais exigem métricas de avaliação

Requisito	Métrica
Desempenho: Velocidade	Transações por segundo Tempo de resposta ao usuário Tempo de resposta a eventos
Desempenho: Memória	Memória ocupada pelo código Memória ocupada pelos dados
Facilidade de Uso	Tempo de treinamento Número de páginas de ajuda
Confiabilidade	MTBF: Tempo médio entre falhas Probabilidade de não-disponibilidade Taxa de ocorrência de falhas Disponibilidade (%)
Robustez	Tempo de reínicio após falha MTTR: Tempo



# **Exemplos de Req. Não Funcionais**

- O tempo de resposta para ler um código de barras tem que ser menor ou igual a 3s
- O sistema deverá permitir o acesso simultâneo de 500 usuários.
- O sistema deve estar disponível todos os dias da semana, ininterruptamente entre 8:00 e 17:00 horas
- O sistema deverá apresentar uma disponibilidade de 99,99%
- O sistema deverá funcionar com um servidor Linux na versão Ubuntu 14.0.1

# Restrições

- Requisitos Globais
- Limitam o projeto
- Limitam o produto
- **Um outro tipo de requisito não funcional**
- Ex:
  - BookApp tem que seguir as normas de interface para ser aprovado na AppStore

# **Req. Não Funcionais x Restrições**

## **Requisitos Não Funcionais**

- Descrevem o que queremos que aconteça
- Normalmente top-down
- Normalmente falam sobre necessidades
- Normalmente desejos das partes interessadas

## **Restrições**

- Impõe limites ou fronteiras sobre o que queremos que aconteça
- Normalmente bottom-up
- Normalmente falam sobre a limites a solução
- Normalmente impostos as partes interessadas

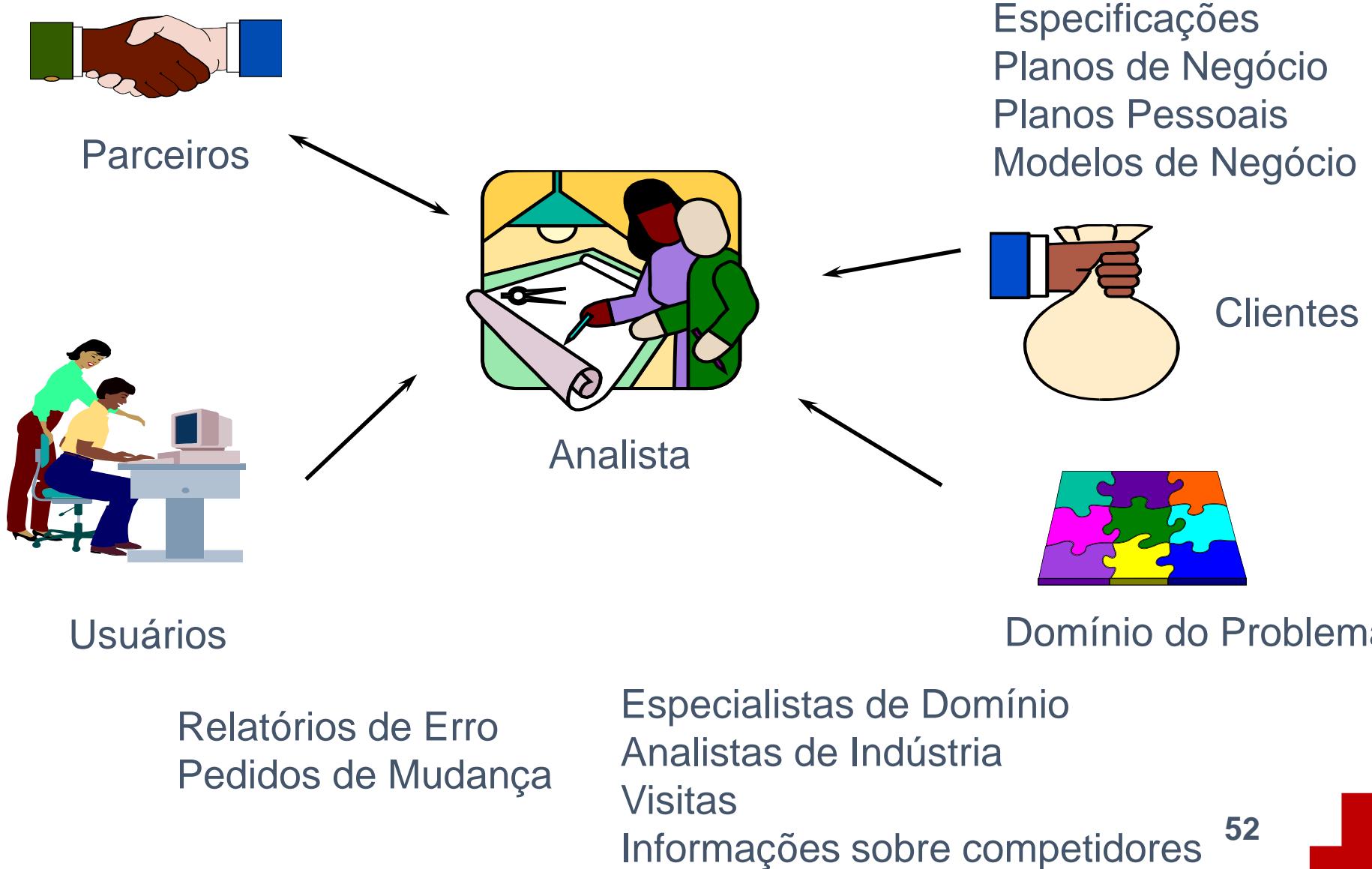
A diferença pode ser sutil em alguns casos.

# **Outros Tipos de requisitos**

- Ambiente físico
- Interfaces
- Usuários e fatores humanos
- Funcionalidade
- Documentação
- Dados
- Recursos
- Segurança
- Garantia de qualidade

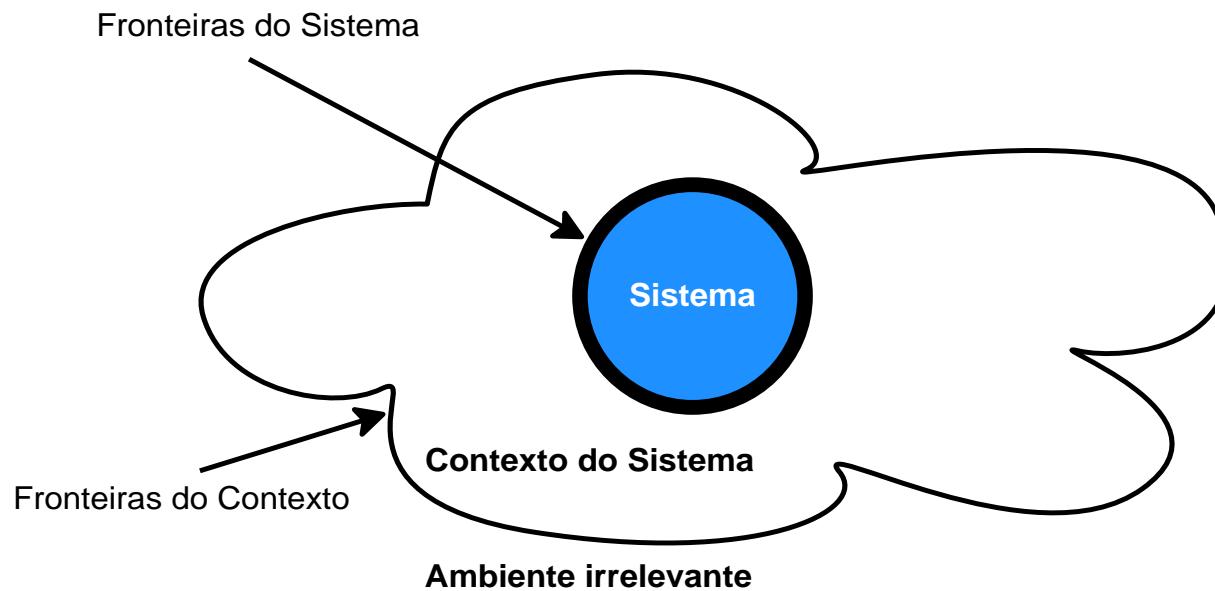
# **Fontes de Requisitos**

# Fontes de Requisitos

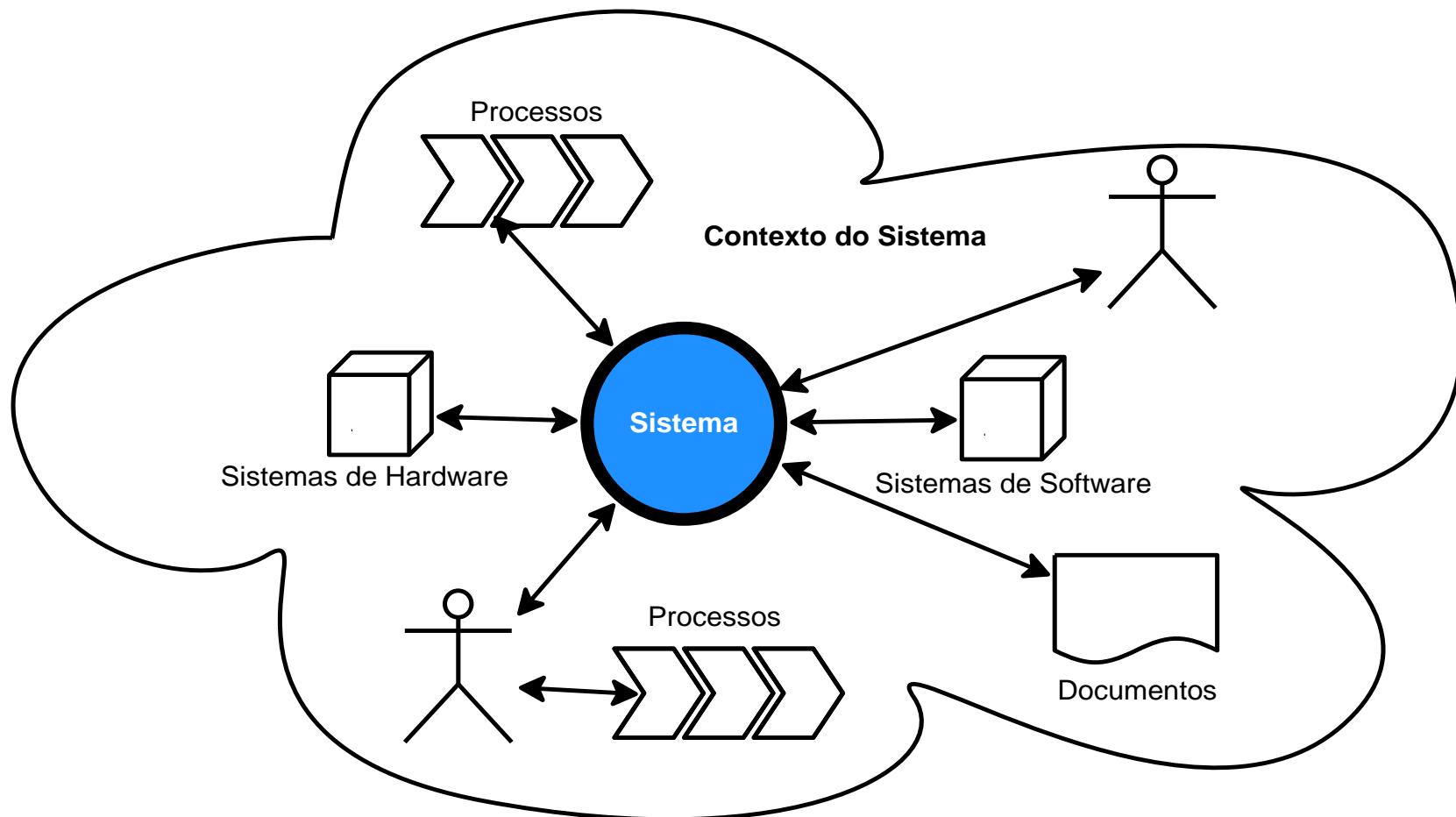


# Fronteira do Sistema

- A fronteira do sistema separa o sistema a ser desenvolvido de seu ambiente, i.e., separa a parte da realidade que pode ser modificada ou alterada pelo processo de desenvolvimento
  - dos aspectos do ambiente que não podem ser modificados ou alterados pelo processo de desenvolvimento
- O que podemos mudar diretamente
  - E não os efeitos



# Contexto de um Sistema



# O Usuário Não Sabe Tudo

- Um problema comum é o usuário pedir algo como requisito porque ele pensa que esta é a forma de implementar um funcionalidade desejada.
  - Esse erro, além de comum, é provavelmente prejudicial ao sistema se não for detectado.
- A verdade é que apesar do analista ter que atender aos stakeholders, ele não tem que atender exatamente ao que eles dizem, mas sim ao que eles realmente precisam.
- O trabalho de análise é um trabalho investigativo,



# O que o usuário sabe?

- Os efeitos dos problemas
- A área de aplicação
- A cultura, o ambiente de trabalho
- As pessoas
- As tentativas anteriores
  - Bem ou mal sucedidas
- Seu descontentamento

# O que o usuário pode desconhecer

- As causas reais do problema
- A estrutura interna da solução
- A complexidade da solução
- Os comportamentos possíveis do software
- O que é inútil
- O que é realmente útil
- Engenharia de Software
- Você e sua equipe

# Categorías de Kano

# Categorias de Kano

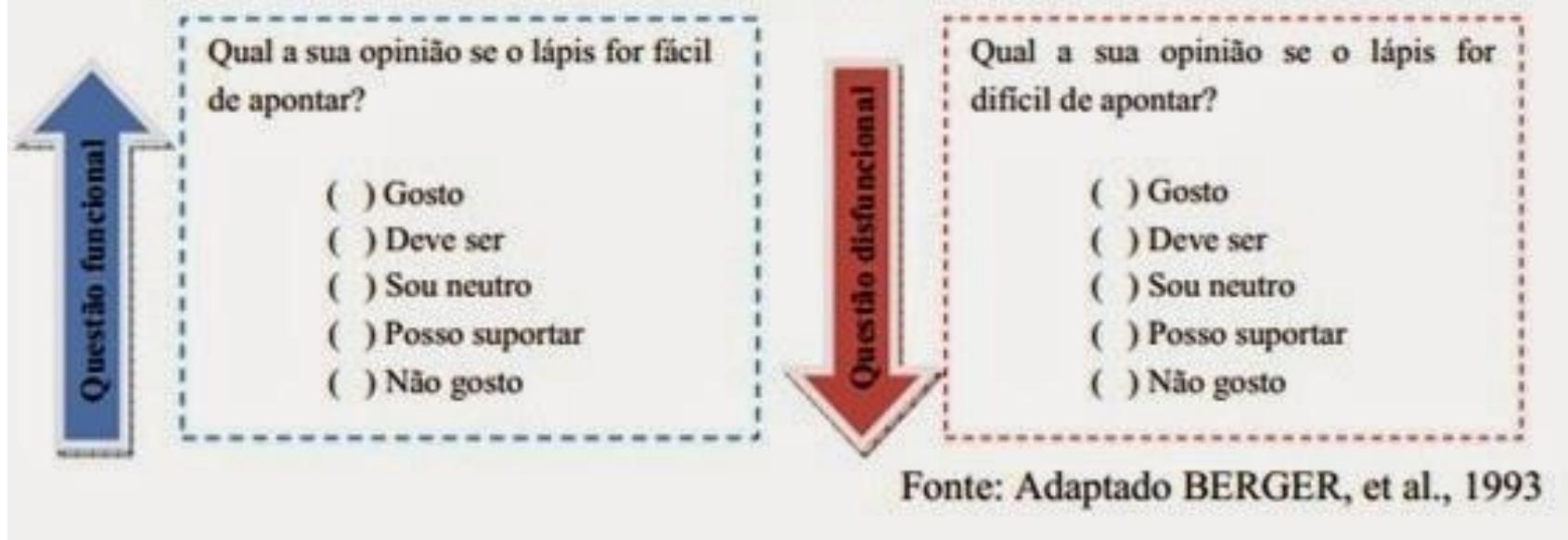
- Unidimensionais (Satisfiers)
  - Atributos de desempenho
  - Satisfação proporcional ao nível de desempenho
  - Normalmente solicitados explicitamente
  - Conscientes
  - Ex: o preço de um prato
- Obrigatórios ou esperados (Dissatisfiers)
  - Não pensamos em mencioná-los até que eles não são encontrados no sistema
  - Sua ausência causa insatisfação
  - Subconscientes
  - Ex. Limpeza do restaurante
- Atrativos (Delighters)
  - Satisfaz, ou não, em relação a presença, ou ausência
  - Sua ausência não traz insatisfação
  - Ex: um brinde surpresa ao fim do jantar
  - Não Conscientes

# Categorias Adicionais

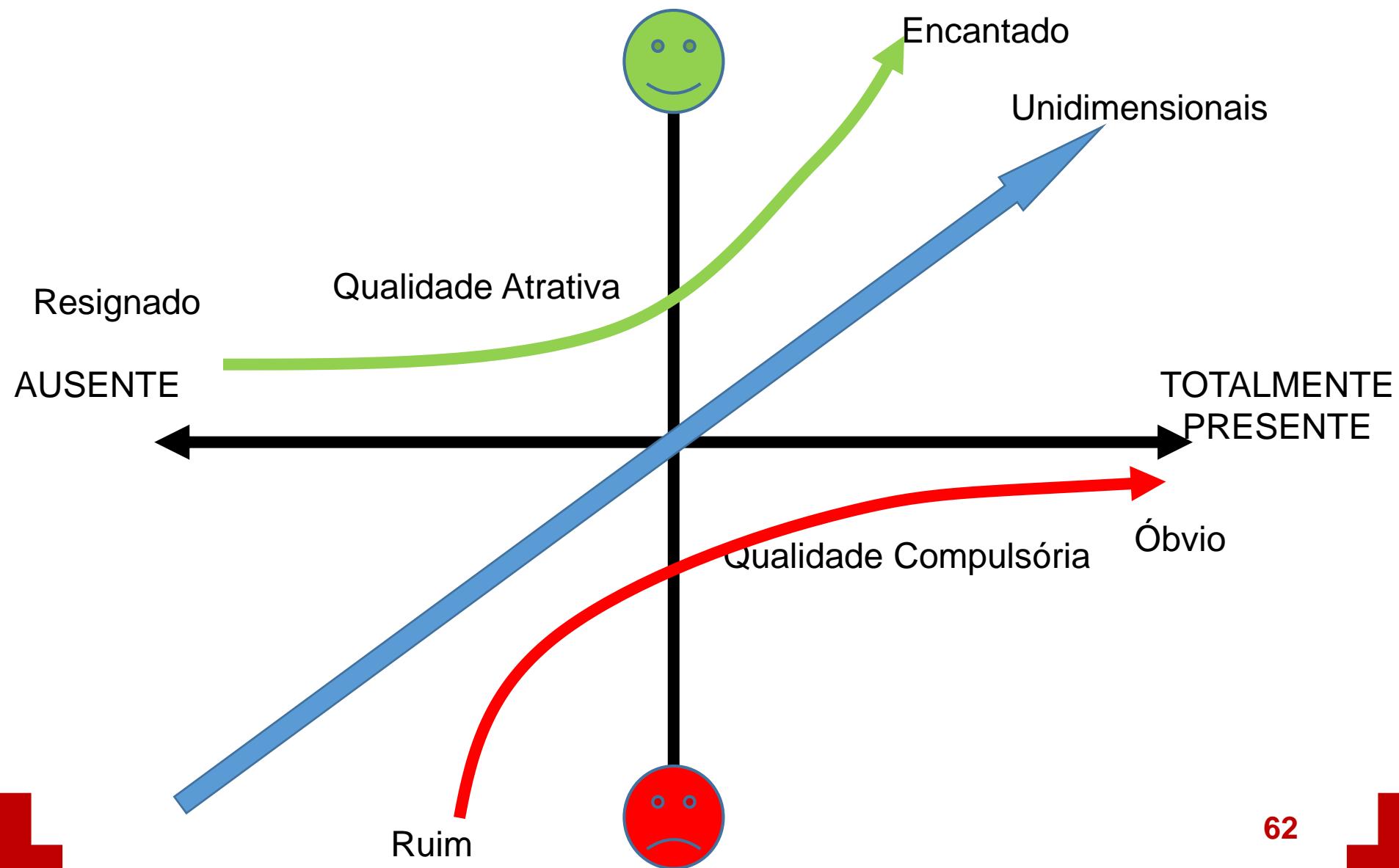
- Indiferentes
  - Não importa se ele atende ou não às expectativas, pois, da mesma forma, não gera satisfação nem tampouco insatisfação.
- Reversos
  - São aqueles que mostram que o julgamento a priori sobre o que é funcional e disfuncional no questionário em relação ao atributo é justamente o reverso do que o cliente realmente sente.
- Questionáveis
  - Demonstram uma contradição nas respostas dos clientes nas partes funcional e disfuncional do questionário.

# Exemplo de Questão

FIGURA 2 - Exemplo de questões funcional e disfuncional do método Kano.



# Gráfico de Kano



# **Variância no Tempo**

- Ao longo do tempo
- Atrativos se tornam Unidimensionais
- Unidimensionais se tornam obrigatórios

# **Bons Requisitos**



# Bons Requisitos

- Necessário
  - se retirado, haverá uma deficiência no sistema
    - não atenderá plenamente as expectativas do usuário.
- Não devem existir requisitos do tipo “seria interessante ter”.
  - O requisito é necessário ou é dispensável.
- Deve ser levado em conta que cada requisito aumenta a complexidade e o custo do projeto
  - não podem ser introduzidos de forma espúria.



# Bons Requisitos

- Conciso
  - Cada requisito define apenas um requisito que deve ser feito e apenas o que deve ser feito
  - Claro e simples.
  - Conciso
    - não inclui explicações, motivações, definições ou descrições do seu uso.
    - Estes textos podem ser mantidos em outros documentos, apontados pelo requisito, de preferência sob o controle de um sistema de gerência de requisitos



# Bons Requisitos

- Alcançável
  - Realizável a um custo definido por uma ou mais partes do sistema
- Completo
  - Não precisa ser explicado ou aumentado, garantindo capacidade suficiente do sistema.
- Rastreável
  - Podemos detectar sua origem



# Bons Requisitos

- Consistente
  - Não contradizendo ou mesmo duplicando outro requisito
  - Utilizando os termos da mesma forma que outros requisitos
- Não-ambíguo
  - Possuindo uma e apenas uma interpretação.
  - Nesse caso é muito importante prestar atenção na linguagem sendo utilizada.



# Bons Requisitos

- Verificável
  - Não sendo vago ou geral
  - Sendo quantificado
    - de uma maneira que permita a verificação de uma das seguintes formas: inspeção, análise, demonstração ou teste.
- Ordenável
  - por estabilidade ou importância (ou ambos).
- Aceito pelos usuários e desenvolvedores.



# Bons Requisitos Funcionais

- Independente da Implementação
  - Define o que deve ser feito, mas não como.
  - Não reflete um projeto ou uma implementação.
  - Não descreve uma operação.
    - Requisitos de interface são uma exceção a essa regra

# **Elicitação e Análise de Requisitos**

# Elicitação de Requisitos

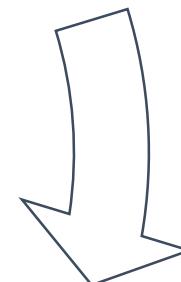


Especificação

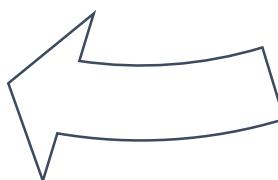
Descoberta



Prioritização  
e Negociação



Classificação  
e  
Organização



# Análise de Requisitos

- Especifica as características funcionais do software
- Indica a interface do software com outros sistemas
- Estabelece as restrições que o software deve seguir
- Guias gerais
  - Deve focar em requisitos que são visíveis dentro dentro do problema ou do domínio de negócio
  - Nível de abstração relativamente alto
  - Cada elemento deve adicionar um entendimento adicional dos requisitos
  - Adiar considerações sobre infraestrutura e outros modelos não funcionais
  - Minimizar acoplamento dentro do sistema
  - Prover valor a todos as partes interessadas
  - Ser o mais simples quanto puder ser

# Elementos da Análise de Requisitos

Modelos baseados em modelos

- Casos de Uso
- Histórias de Usuário

Modelos de Classe

- Diagramas de Classe
- Diagramas de Colaboração

Requisitos de Software

Modelos Comportamentais

- Diagramas de Estado
- Diagramas de Sequência

Modelos de Fluxo

- DFDs
- Modelos de Dados

# Técnicas de Elicitação

- Pesquisas
  - Entrevistas
    - Estruturadas
    - Não estruturadas
  - Questionários
- Criativas
  - Brainstorm
  - Design Thinking
- Centradas em Documento
  - Arqueologia de sistema
- Observação
  - Observação de Campo
  - Observação de Campo com Aprendizado
- JAD
- CRC
- Volere

# **Escolha da Técnica**

- Necessidade de distinção de requisitos unidimensionais, obrigatórios e atrativos
- Limitações de orçamento e tempo
- Disponibilidade das partes interessadas
- Experiência do analista/engenheiro de requisitos com a técnica
- Oportunidades e riscos do projeto

# **Documentos**

# Especificação de Requisitos

- É uma coleção de requisitos, para um sistema ou componente, sistematicamente representada, satisfazendo um critério dado
- 3 perspectivas importantes
  - Dados
  - Funcional
  - Comportamental



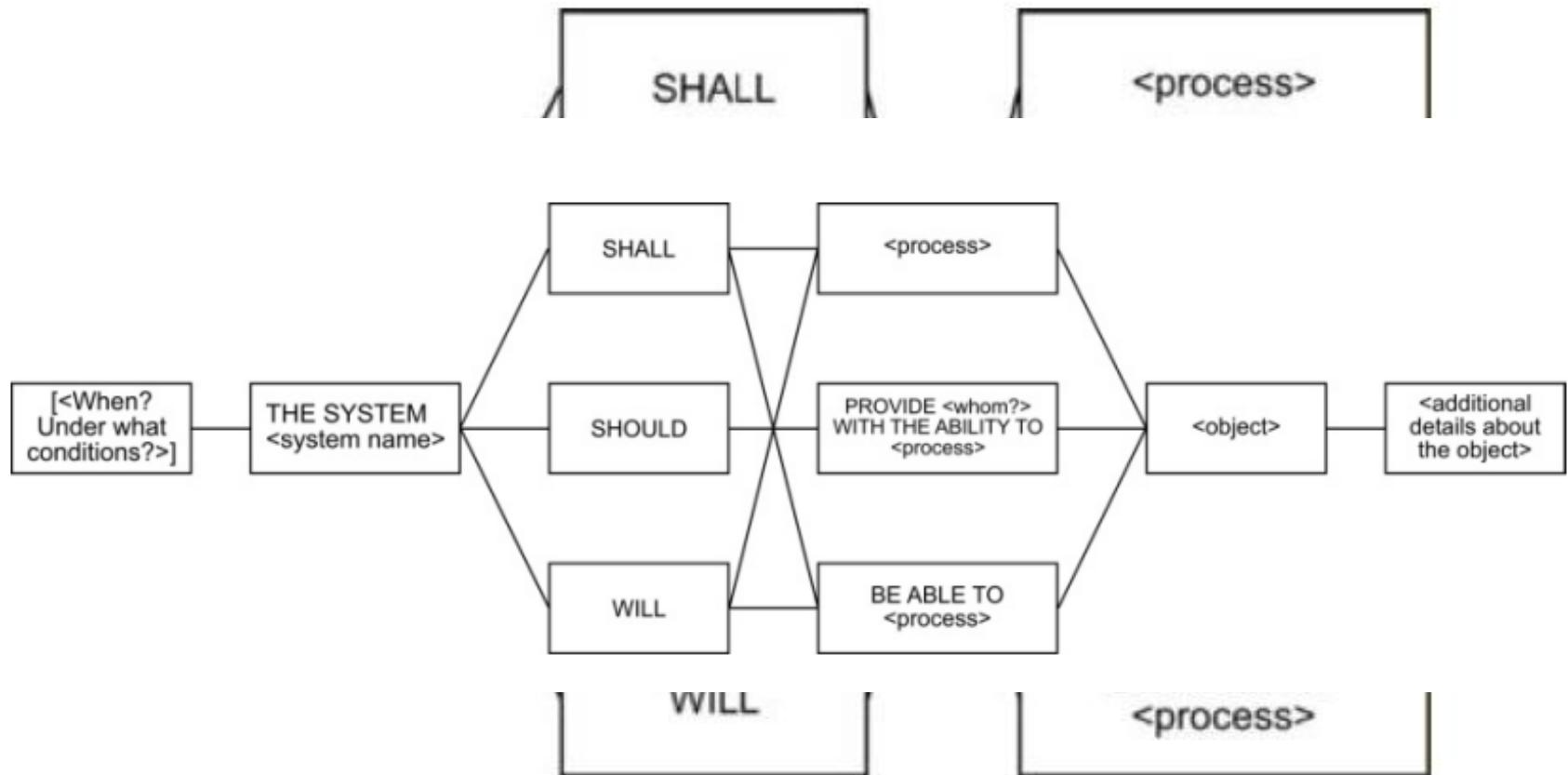
# **Padrões de Requisitos**

- IEEE Standard 830-1998
  - 3 capítulos principais
    - Introdução
      - Objetivo do sistema, fronteiras do sistema
    - Descrições Gerais
      - Perspectivas
      - Usuários
      - Restrições
    - Descrição dos requisitos
      - Funcionais
      - Desempenho
      - ...
- V Model (V Modell 2004) do BMI
  - Especificação de Requisitos do Comprador
  - Especificação de Requisitos do Sistema

# Conteúdo Mínimo

- Introdução
  - Objetivo
  - Cobertura do sistema
  - Partes Interessadas
  - Definições, Acrônimos e Abreviações
  - Referências
  - Visão geral
- Descrição
  - Ambiente do sistema
  - Descrição da arquitetura
  - Funcionalidades do Sistema
  - Usuário e audiência alvo
  - Restrições
  - Premissas
- Requisitos
- Apêndices
- Index

# Forma (em inglês)



# Forma (em português)

- Sempre usar o futuro
- Sempre exigir do sistema
- [Quando] **o sistema deverá** [permitir que o <usuário> <processo> [<objeto> [<detalhes do objeto>]]
  - O sistema deverá permitir que o comprador escolha a forma de pagamento
  - O sistema deverá calcular os juros devidos
  - Ao fim do mês o sistema deverá listar os clientes inadimplentes
  - O sistema deverá permitir ao gerente solicitar a lista de vendas por loja, ordenadas por valor em ordem decrescente

# Cuidado

- Queremos que o sistema possua uma função, usada pelo gerente, que gere uma lista de vendedores
- **O sistema deverá permitir ao gerente solicitar a impressão da lista de vendedores**
  - Indica controle do gerente quanto a quando a função é executada
  - Indica que o sistema deverá atender o pedido
  - Indica que é uma impressão
- ~~O gerente listará vendedores~~
  - Não é o gerente que lista, é o sistema
- ~~O sistema deverá permitir ao gerente listar os vendedores~~
  - O gerente não lista nada, quem lista é o sistema
- ~~O sistema listará vendedores para o gerente~~
  - Quando o sistema fará isso?
- ~~Quando o gerente quiser, o sistema listará os vendedores~~
  - Como o sistema vai saber que o gerente quer?

# Lembrar

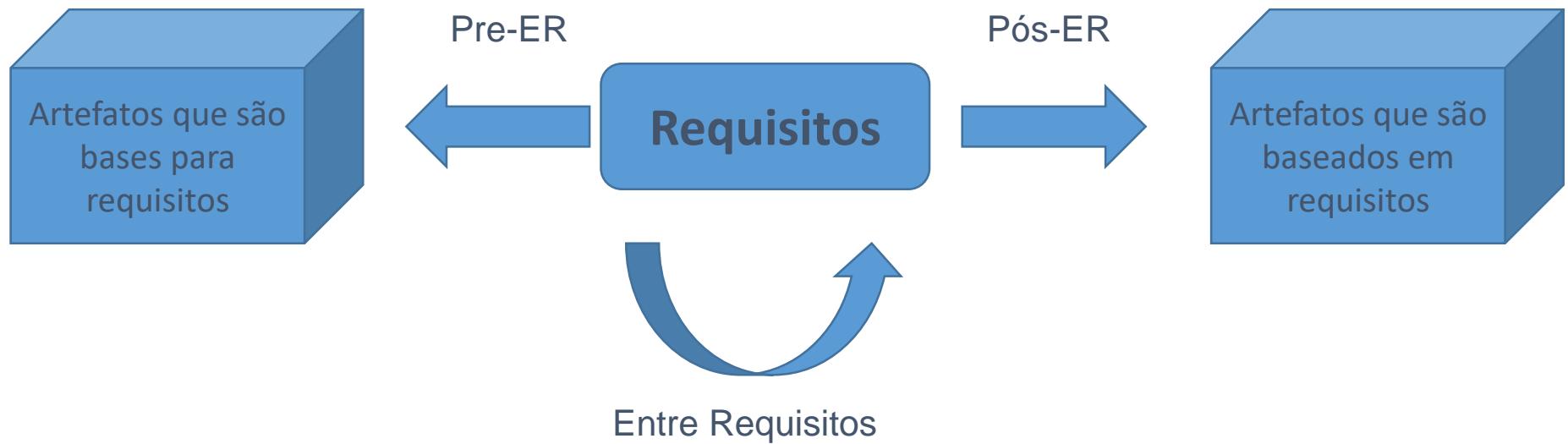
- Os requisitos são do sistema
  - Não pedimos / obrigamos os usuários a nada
  - Correto
    - O sistema deverá permitir que o comprador selecione a forma de pagamento
  - Incorreto
    - ~~O comprador selecionará a forma de pagamento~~

# Rastreabilidade

# Rastreabilidade

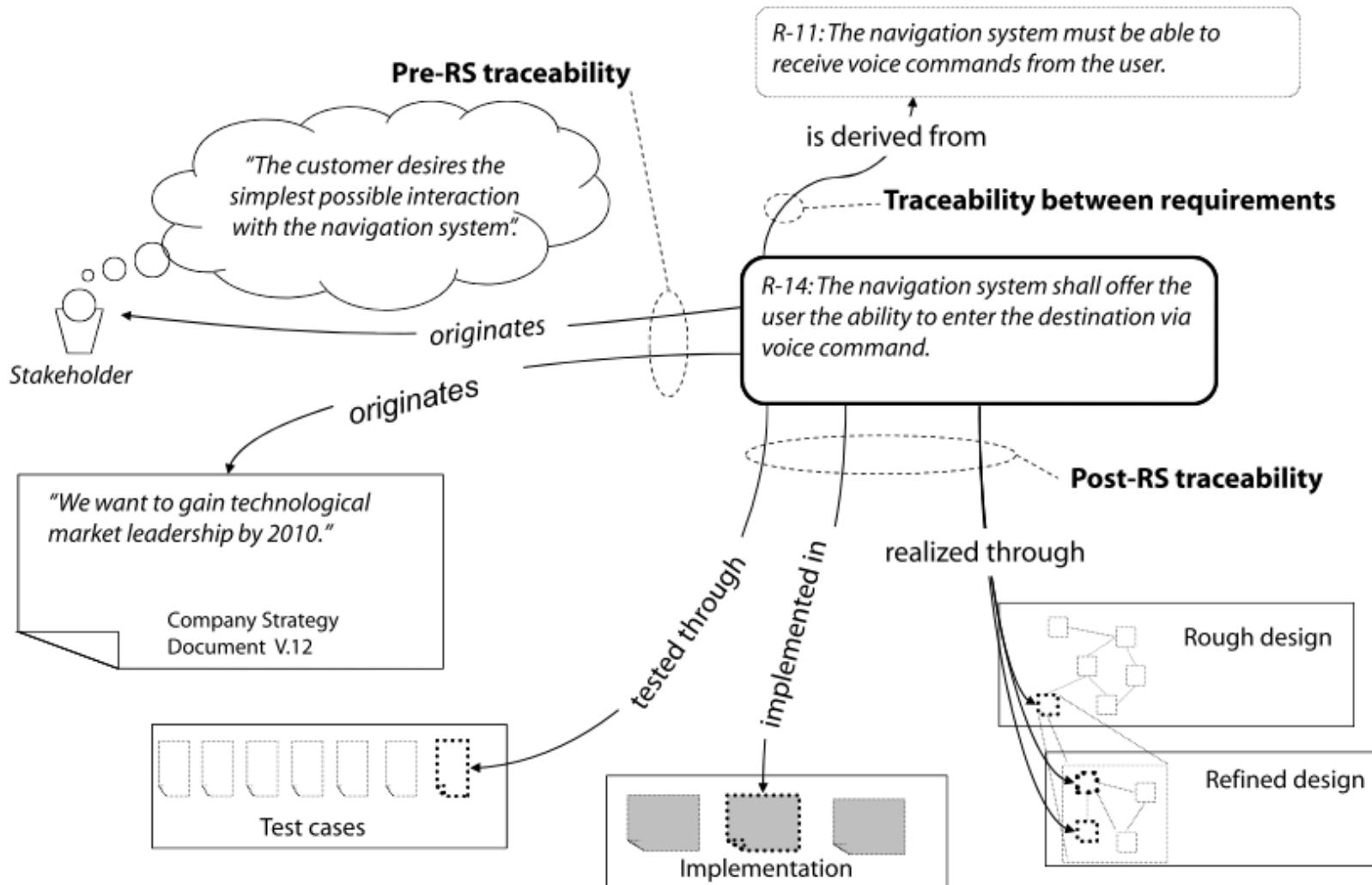
- Habilidade de rastrear o requisito através de todo ciclo de vida do sistema
- Vantagens
  - Verificação
  - Identificação de “gold-plating”
  - Análise de Impacto
  - Reuso
  - Responsabilidades (Accountability)
  - Manutenção

# Tipos de Rastreabilidade

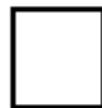
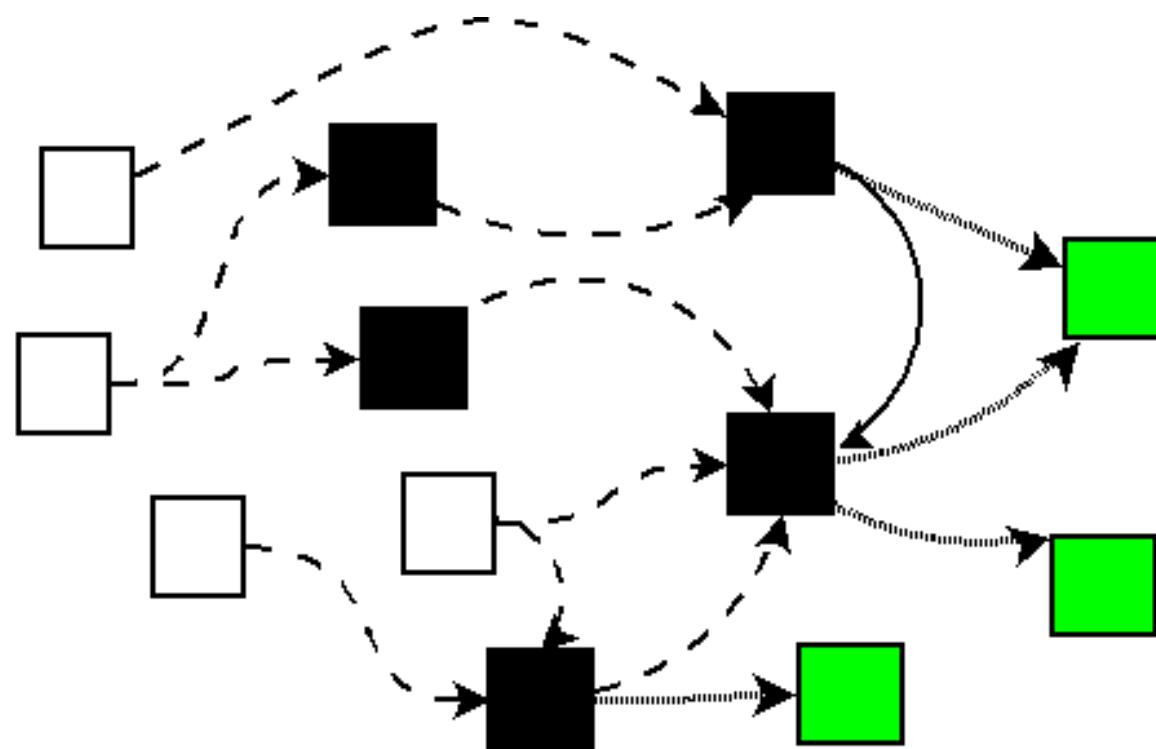


ER = Especificação de Requisitos

# Exemplos de Rastreabilidade



# Representação - Gráfica



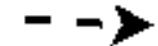
Informação sobre o contexto



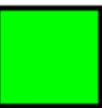
Refina



Requisito



Origina



Componentes



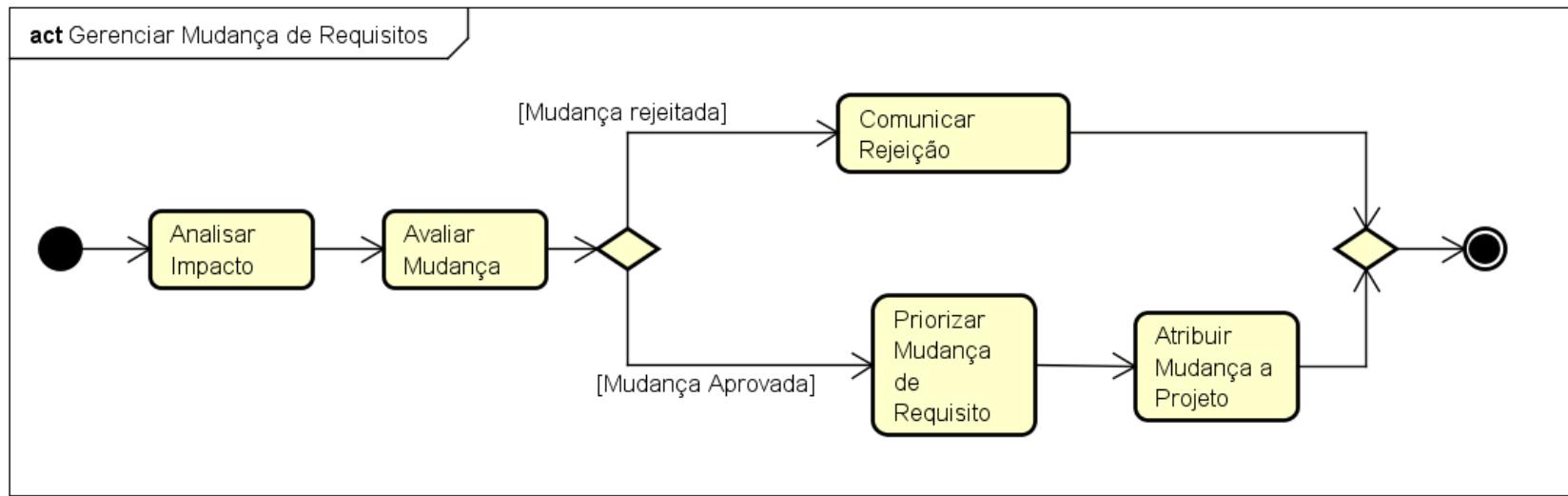
Realizado por

# Representação - Matriz

Depende	Req 1	Req 2	Req 3	Req 4
Req 1	-	X		
Req 2		-	X	X
Req 3			-	X
Req 4				-

# **Mudanças nos Requisitos**

# Mudanças nos Requisitos



powered by Astah

# **Volere Cards**

Requirement #:

Requirement Type:

Event/use case #:

Description:

Rationale:

Source:

Fit Criteria:

Customer Satisfaction:

Customer Disatisfaction:

Dependencies:

Conflicts:

Supporting Materials:

History:

**Volere**

Copyright © Atlantic Systems Guild

# Documentando Requisitos

- Número identificador,
  - para facilitar a discussão, identificamos todos os requisitos unicamente.
- Tipo
  - Classificando-o como funcional, não funcional,...
- Evento que o atende
- Descrição
- Justificativa
- Fonte do requisito
  - A pessoa ou o grupo que o originou
- Critério de aceitação
  - Uma medida que possa ser usada para garantir que o requisito foi alcançado.
- Satisfação do usuário
- Insatisfação do usuário
- Dependências
  - Referências a outros requisitos que dependem de alguma forma desse requisito
- Conflitos
  - Referência aos requisitos que de alguma forma conflitam com esse
- Material de apoio
  - Listagem de material de apoio para atender esse requisito
- Histórico
  - Documentação da criação e das mudanças efetuadas

Requirement #: 75

Requirement Type: 9

Event/Use Case #: 7, 9

Description: The product shall record all the roads that have been treated.

Rationale: To be able to schedule untreated roads and highlight potential danger.

Originator: Arnold Snow, Chief Engineer

Fit Criterion: The recorded treated and untreated roads shall agree with the drivers' road treatment logs.

Customer Satisfaction: 3

Customer Dissatisfaction: 5

Priority:

Conflicts:

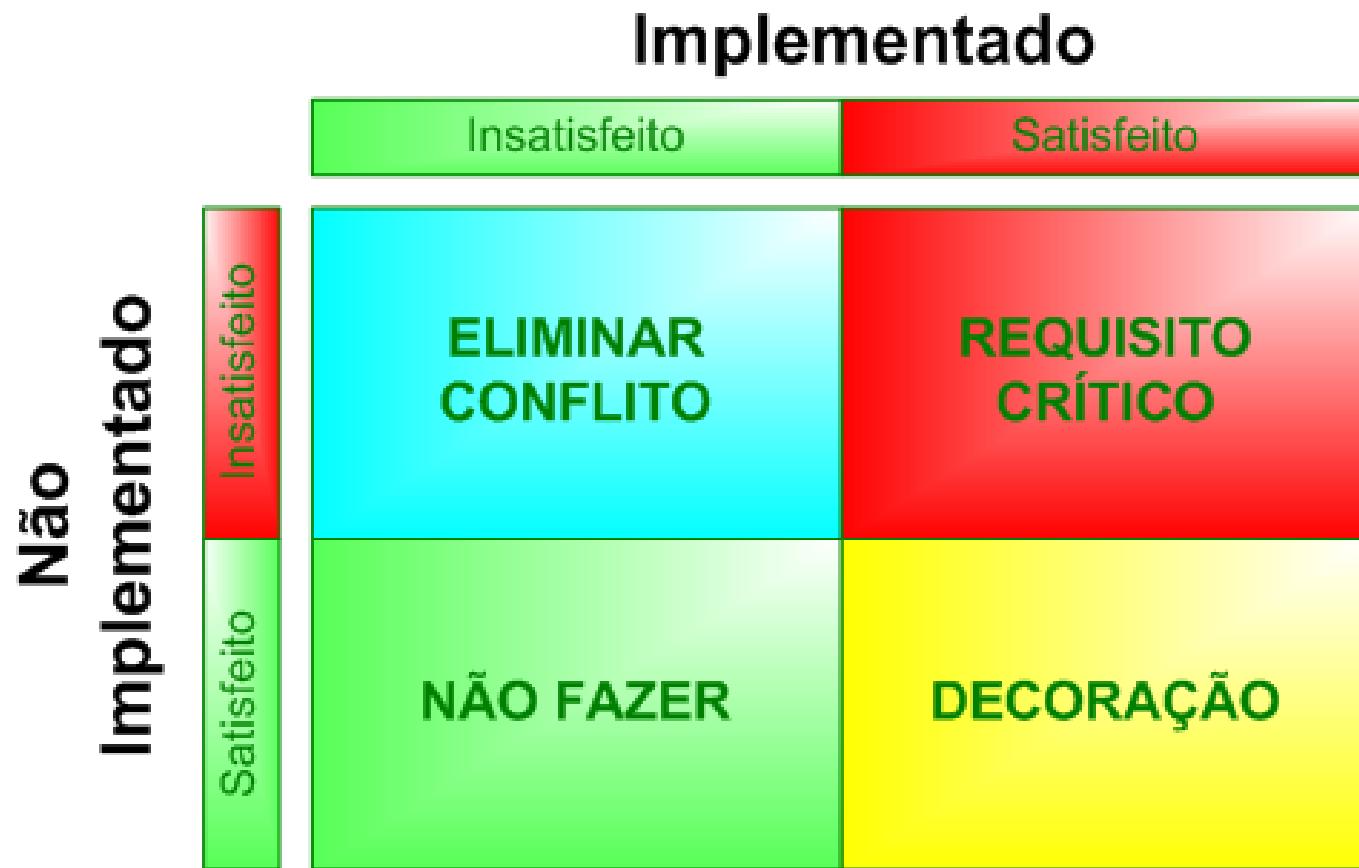
Supporting Materials:

History: Created February 29, 2006

**Volere**  
Copyright © Atlantic Systems Guild

# Após a Classificação

Semelhante ao Modelo de Kano



**FIM**