

REQUISITOS DE APLICAÇÃO

Ceça Moraes

ROTEIRO

1. ENGENHARIA E ENGENHARIA DE SOFTWARE
2. ENGENHARIA DE REQUISITOS
3. FUNDAMENTOS DE REQUISITOS DE SOFTWARE
4. PROCESSO DE REQUISITOS
5. ELICITAÇÃO DE REQUISITOS
6. ANÁLISE E NEGOCIAÇÃO DE REQUISITOS
7. DOCUMENTAÇÃO DE REQUISITOS
8. VALIDAÇÃO DE REQUISITOS
9. PROBLEMAS NA ENGENHARIA DE REQUISITOS
10. BOAS PRÁTICAS
11. DOCUMENTO DE VISÃO
12. GLOSSÁRIO
13. MODELO DE CASO DE USO – ATORES E CASOS DE USO
14. MODELO DE CASO DE USO – ESTRUTURAÇÃO

Engenharia

- “A engenharia refere-se à prática das atividades organizadas de **projeto, construção e operação** de qualquer artefato que transforme os mundos físico e social ao nosso redor para **atender** a alguma **necessidade reconhecida...**”
- “Desenvolvimento e aplicação sistemática de **modelos** de comprovada eficácia gerando **soluções tecnológicas** para problemas da humanidade” (Prof. Jorge Fernandes – UFRN)
- “Engenharia é a **arte**, a **ciência** e a **técnica** de conjugar os conhecimentos especializados (científicos) de uma dada área com a sua viabilidade técnico-econômica, para produzir novas utilidades e/ou transformar a natureza, em **conformidade com idéias** bem planejadas.” (Wikipedia)

ENGENHARIA DE SOFTWARE

- Definição:
- A IEEE¹ define a Engenharia de Software como:
 - (1) A aplicação de uma abordagem **sistemática**, **disciplinada** e *quantificável* para desenvolvimento, operação e manutenção de software; isto é, a aplicação da engenharia **para o software**.

¹ “IEEE Standard Glossary of Software Engineering Terminology,”
IEEE std 610.12-1990, 1990.

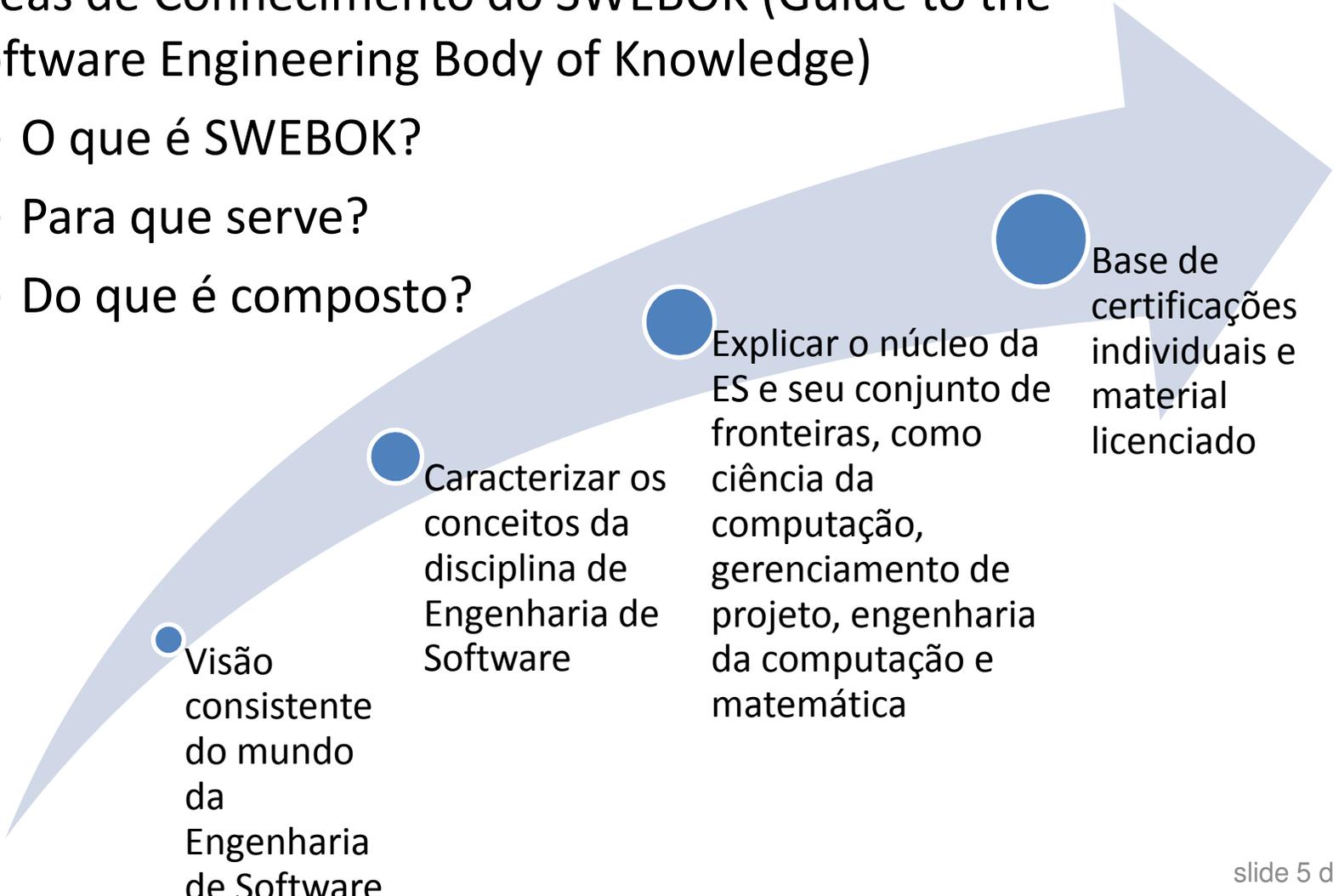
ENGENHARIA DE SOFTWARE - SWEBOK

- Áreas de Conhecimento do SWEBOK (Guide to the Software Engineering Body of Knowledge)

- O que é SWEBOK?

- Para que serve?

- Do que é composto?



Visão consistente do mundo da Engenharia de Software

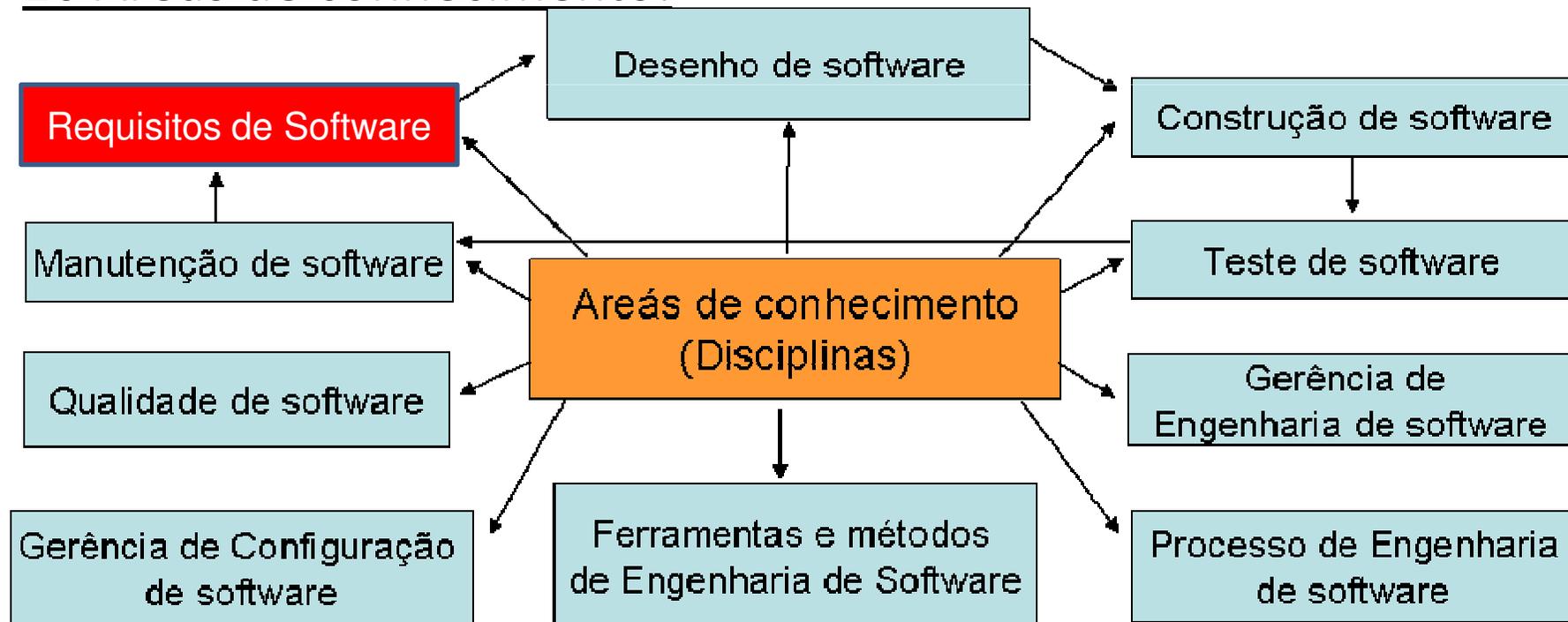
Caracterizar os conceitos da disciplina de Engenharia de Software

Explicar o núcleo da ES e seu conjunto de fronteiras, como ciência da computação, gerenciamento de projeto, engenharia da computação e matemática

Base de certificações individuais e material licenciado

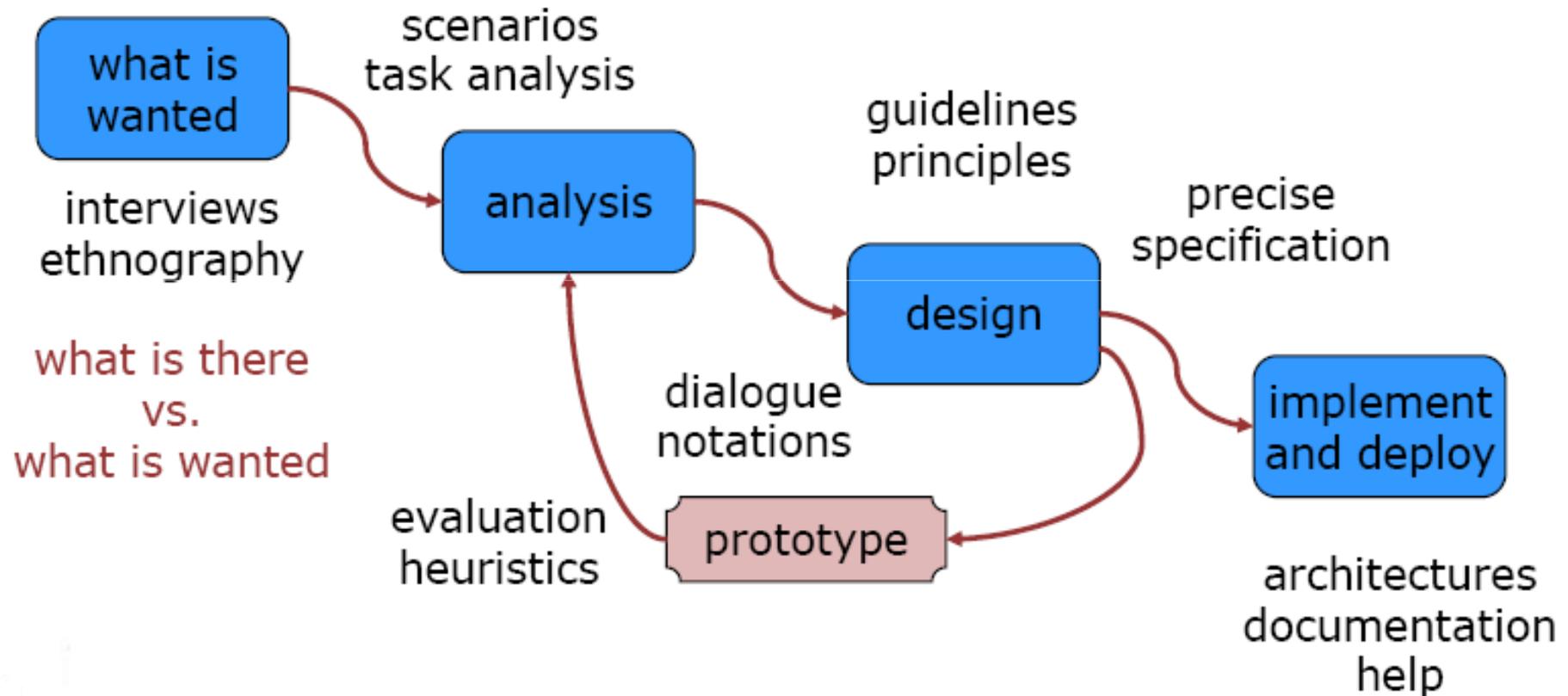
SWEBOK

- SWEBOK = Swebok (*Software Engineering Body of Knowledge*) é uma iniciativa da IEEE Computer Society em criar um consenso sobre as áreas de conhecimento da engenharia de software e seu escopo.
- 10 Áreas de conhecimento:



Requisitos

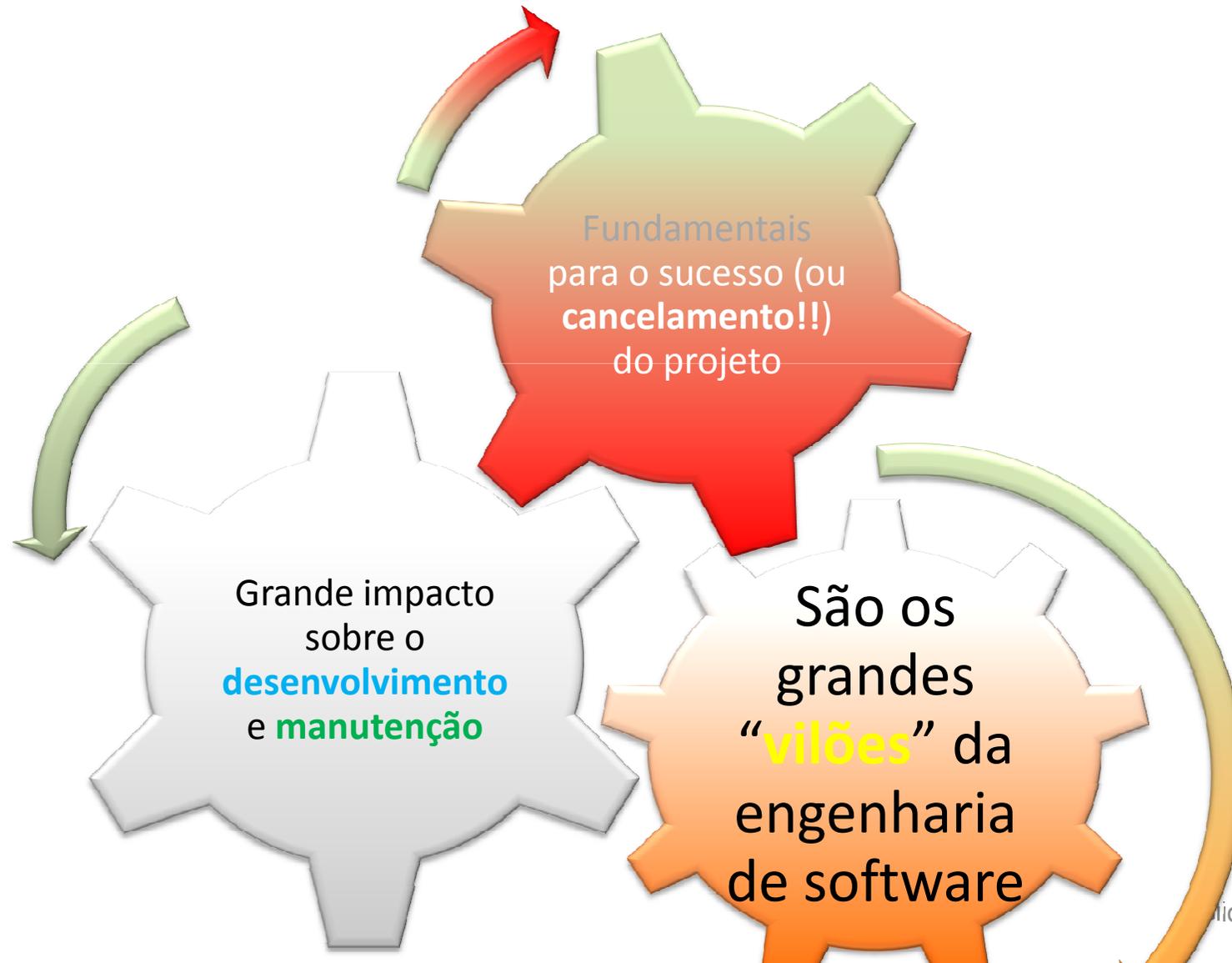
- Ponto de partida do projeto de software



E o que é um “Requisito”???

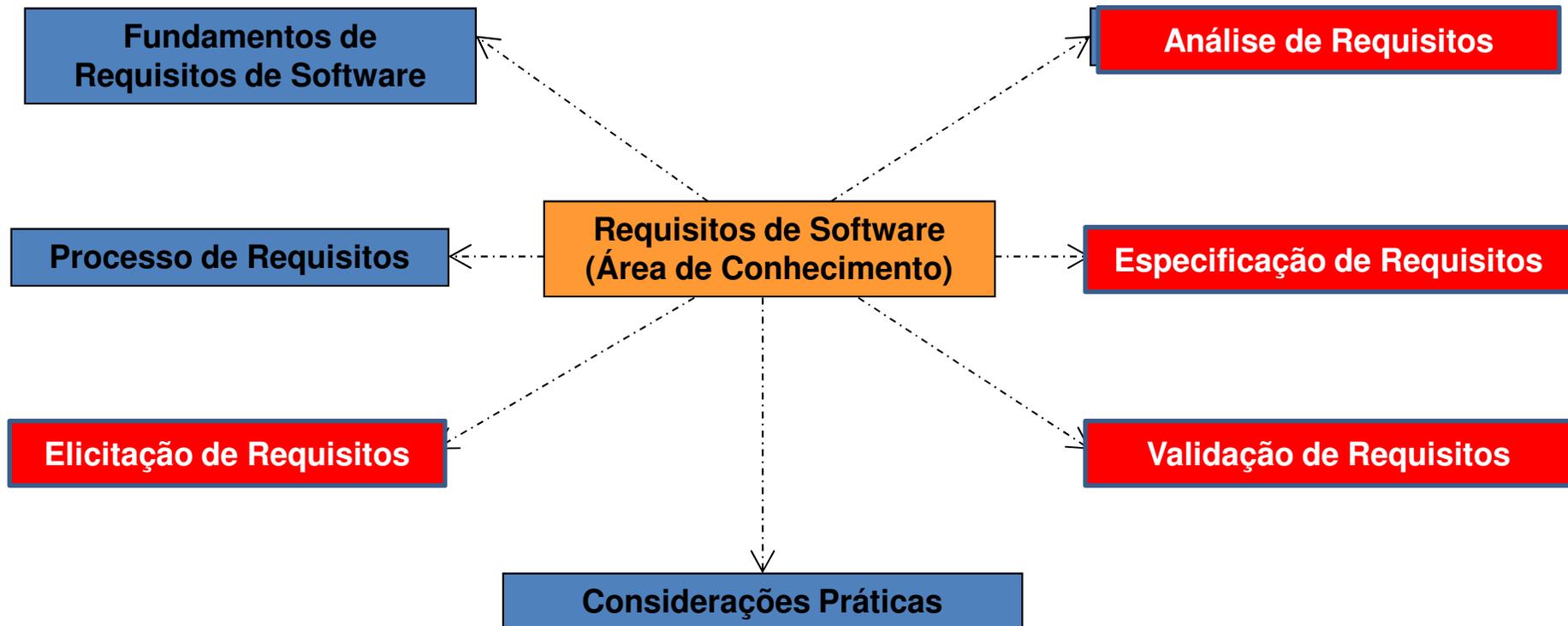
- Houaiss:
 - 1 que foi requisitado, requerido
 - 2 condição para se alcançar determinado fim
 - 3 exigência de ordem legal necessária para a validade de um ato jurídico; condição; formalidade
- De software:
 - uma **condição** ou **capacidade** necessitada por um usuário para resolver um problema ou alcançar um objetivo;
 - é **o que o sistema deve fazer** para implementar uma necessidade de automação requerida pela solução. Desde as **necessidades básicas do cliente**, premissas e restrições obtidas na fase de levantamento do projeto até as condições de negócio explicitadas no contrato com o fornecedor da solução.
 - conjunto de definições que **descreve como o sistema deve ser construído** e testado
 - **Propriedade que o sistema deve apresentar para resolver um problema real (SWEBOK, 2004)**

Características dos requisitos (Castro, 2005)



ENGENHARIA DE REQUISITOS

- Sub-áreas de conhecimento de Requisitos de Software



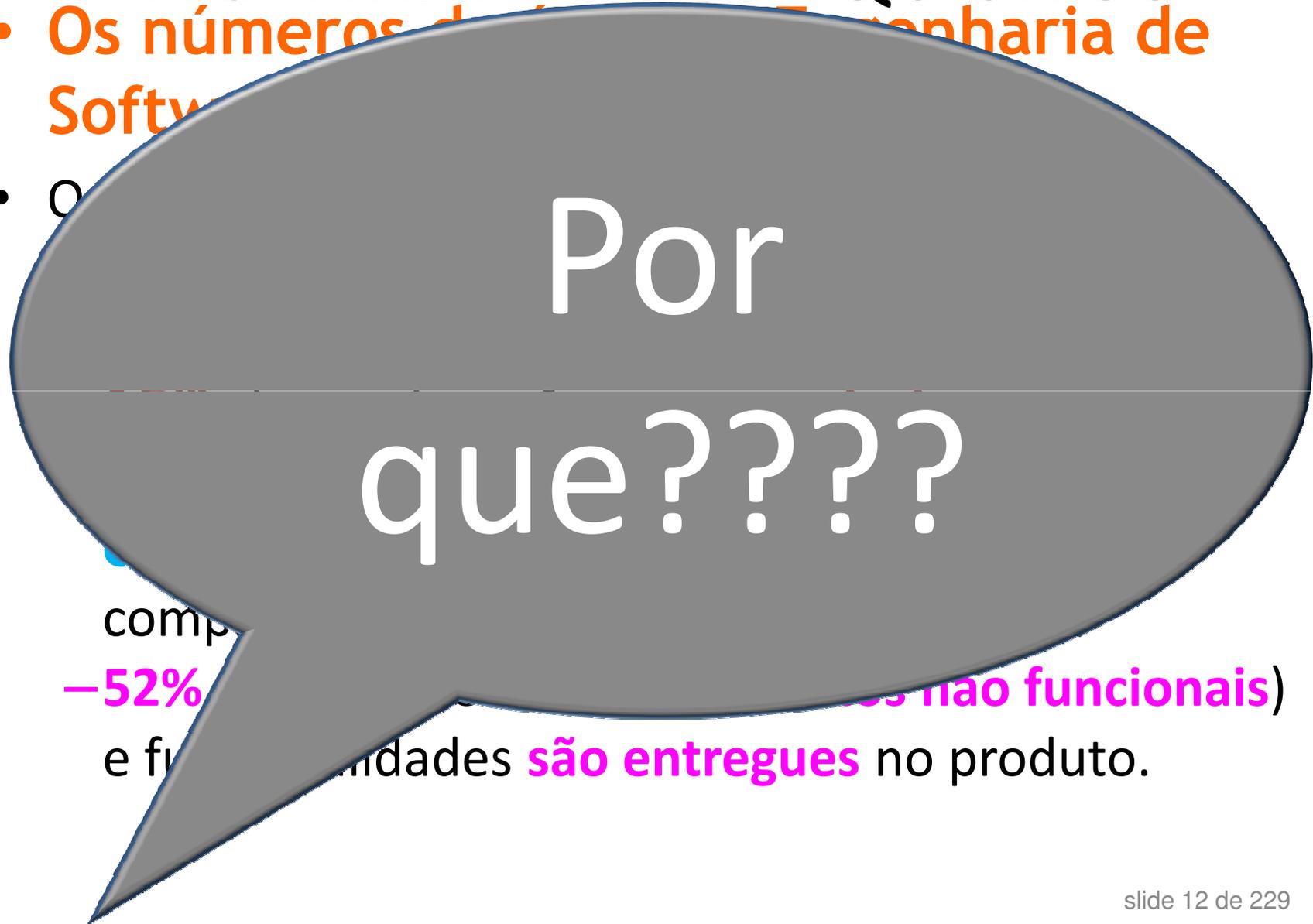
ENGENHARIA DE REQUISITOS

- Os números da área de Engenharia de Software não são animadores...
- O CHAOS Report de 2003 (Standish Group) apresentou as seguintes estatísticas:
 - 34% dos projetos são *bem sucedidos*;
 - 15% dos projetos foram **cancelados**;
 - 43% é o **erro médio** em relação ao **orçamento** do projeto daqueles que foram completados;
 - 52% das características (**requisitos não funcionais**) e funcionalidades **são entregues** no produto.

ENGENHARIA DE REQUISITOS

- Os números da Engenharia de Software

- 0



Por
que?????

comp

–52%

requisitos não funcionais)
e funcionalidades **são entregues** no produto.

ENGENHARIA DE REQUISITOS

- Algo relacionado com a Engenharia de Requisitos?
 - ¹O que disse o SEI - Software Engineering Institute:
 - Dois principais fatores de falhas de **orçamentos** e **cronograma** são problemas de **requisitos**:
 - ✓ Especificação de requisitos inadequada
 - ✓ Mudanças em requisitos
 - **70 a 85% dos erros** encontrados em software podem ser rastreados para problemas de requisitos (Barry Boehm, 1981)
 - Os erros mais caros de se corrigir em um projeto de software são os originados **nesta etapa**
 - Se partimos de algo com problemas, que não atende ao usuário... certamente não chegaremos ao sucesso.

¹SEI National Software Capacity Study, 1990

Custos de Requisitos

- O custo relativo para correção de um problema de requisitos em cada fase do sistema é (Boehm and Papaccio 1988²):
 - \$1 na fase de análise de requisitos
 - \$5 na fase de projeto do sistema
 - \$10 na fase de codificação
 - \$20 na fase de testes unitários
 - \$200 após a entrega do produto

Fatores de Falhas nos Projetos

- Objetivos não estavam claros**
- Ignorar um grupo de clientes**
- Requisitos e especificações incompletos**
- Requisitos e especificações instáveis (mudanças)**
- Omitir um grupo de requisitos**
- Permitir inconsistências entre grupos de requisitos**
- Aceitar requisito inadequado, incorreto, indefinido, ou impreciso**
- Aceitar um requisito ambíguo e inconsistente**

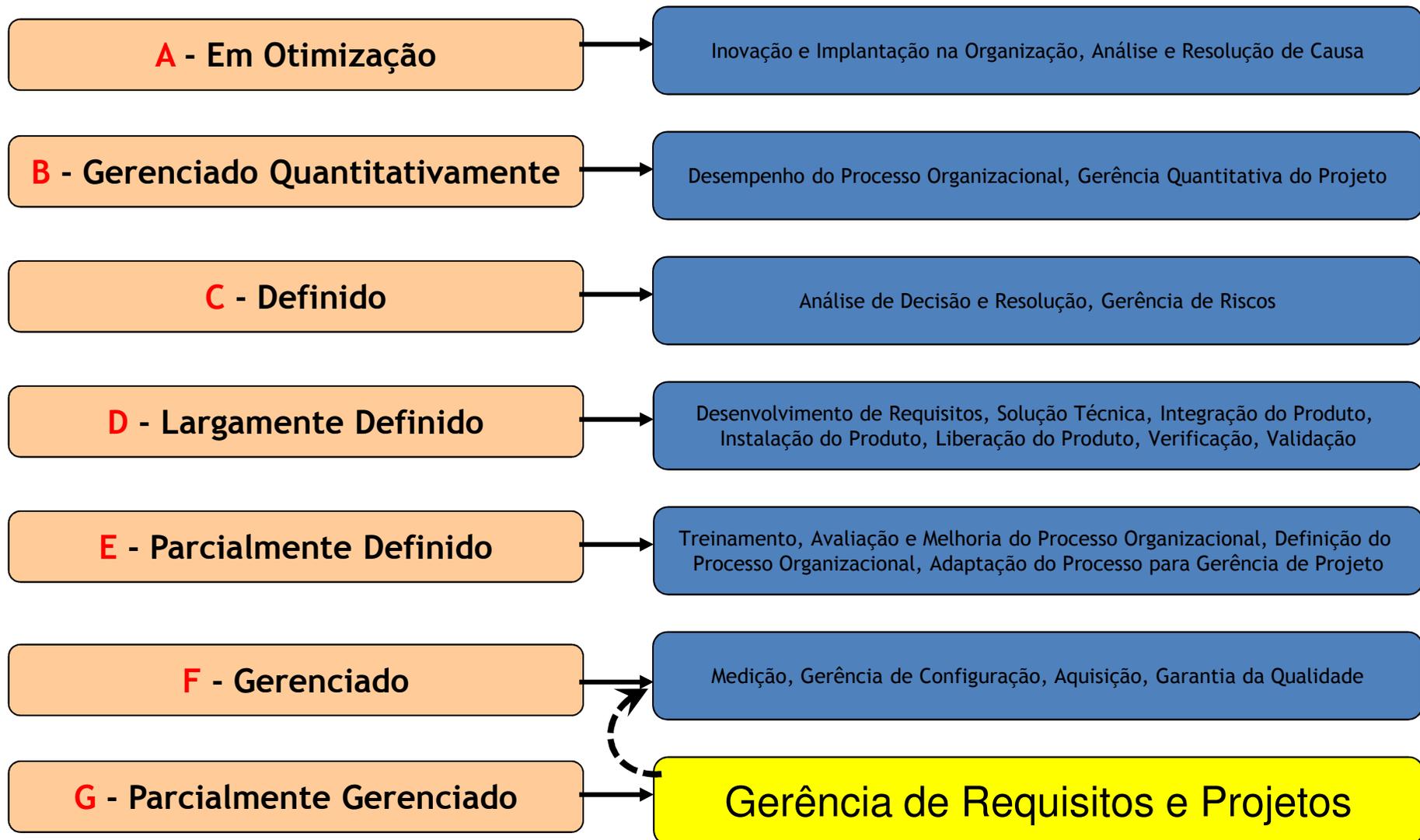
Importância da boa especificação...

- Não importa quão bem projetado ou codificado está um programa, se ele for mal analisado e especificado desapontará o usuário e trará aborrecimentos ao desenvolvedor



Importância da Engenharia de Requisitos

Mps.Br - Melhoria do processo de software brasileiro



CMMI - Capability Maturity Model Integration

5 – Em Otimização

- Foco na melhoria do processo

4 – Gerenciado Quantitativamente

- Processo medido e controlado

3 – Definido

- Processo caracterizado para a organização e proativo

2 – Gerenciado

- Processo caracterizado para o projeto e muitas vezes reativo

1 - Inicial

- Processo imprevisível pobremente controlado e reativo

CMMI - Capability Maturity Model Integration

5 – Em Otimização

4 – Gerenciado Quantitativamente

3 – Definido

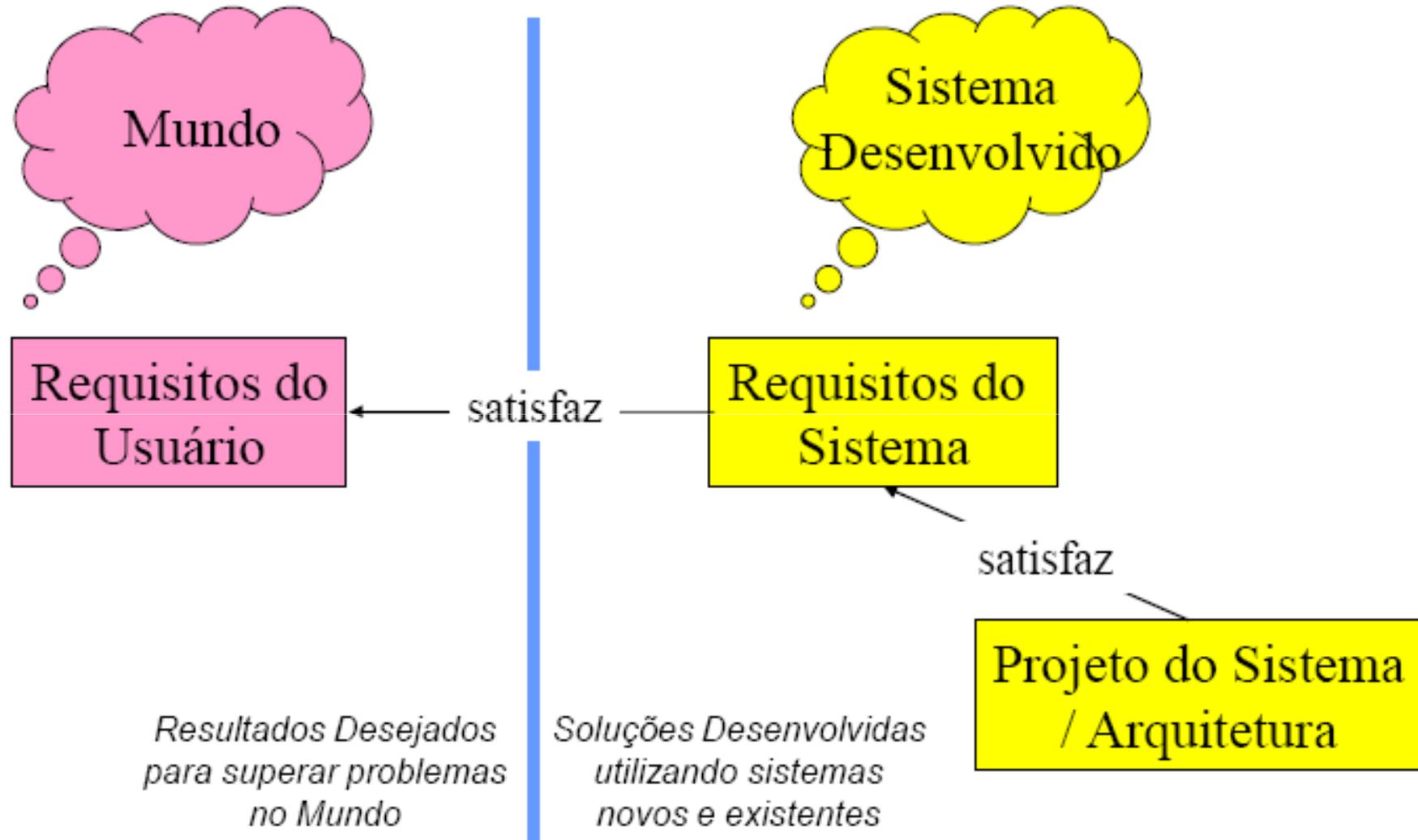
2 – Gerenciado

- **Gerência de requisitos,**
- Planejamento do Projeto,
- Monitoração e Controle do Projeto,
- Garantia da Qualidade do Processo e do Produto,
- Gerência de Configuração,...

1 - Inicial

- Entendimento comum entre o cliente e o fornecedor quanto aos requisitos que serão atendidos no projeto de software
- Acordo entre os envolvidos com o projeto sobre os requisitos e suas mudanças ao longo do mesmo.

Resumindo....



Resumindo....

→ Necessidade

- A origem de tudo!
- Se não há qualquer necessidade, é porque não existe problema
- = problema operacional ou de negócio que precisa ser resolvido

→ O que é um requisito de software?

- Propriedade que deve ser atendida com o objetivo de resolver algum problema no mundo real
- Descrevem comportamento, propriedade e atributo dos softwares
- As vezes permite múltiplas interpretações
- Dever ser testáveis
- Devem ser unicamente identificados

FUNDAMENTOS DE REQUISITOS DE SOFTWARE

→ Requisito de produto x Requisito de processo

- Parâmetros do produto são requisitos do software a ser desenvolvido
 - O software deve verificar se o aluno tem todos os pré-requisitos para o curso que deseja se matricular.
- Parâmetros do processo são essencialmente limitações do processo de desenvolvimento
 - O software deve ser desenvolvido na linguagem Java

FUNDAMENTOS DE REQUISITOS DE SOFTWARE

→ Tipos de requisitos

▪ Requisitos funcionais

- Definem funcionalidades do software
- Operações que clientes e usuários querem, ou precisam, que sejam realizadas pelo sistema
- Exemplo:
 - Possibilitar consulta de Saldo e Extrato em Caixas Eletrônicos e pela Internet
 - Permitir impressão de cheques em Caixas Eletrônicos
 - Permitir solicitação de entrega de talão cheques pela Internet;

FUNDAMENTOS DE REQUISITOS DE SOFTWARE

→ Tipos de requisitos

▪ Requisitos não-funcionais

- Atuam na limitação da solução
- Conhecidos como limitações (no sistema e/ou processo de desenvolvimento) ou requisitos de qualidade
- Dizem respeito a:
 - ✓ aspectos de desempenho,
 - ✓ interfaces com o usuário, confiabilidade,
 - ✓ segurança,
 - ✓ manutenibilidade,
 - ✓ portabilidade,
 - ✓ padrões

Requisitos Não Funcionais

- São críticos para o sucesso de sistemas de software
- Os defeitos provenientes da não elicitação ou elicitação incorreta de Requisitos Não Funcionais (RNFs) estão entre os requisitos mais caros e difíceis de se corrigir.
 - *Softwares* de baixa qualidade
 - Tempo e custo maiores que os planejados
 - Insatisfação dos clientes e da equipe de desenvolvimento
- As conseqüências da desconsideração de um RNF são freqüentemente mais severas do que as decorrentes de um requisito funcional.

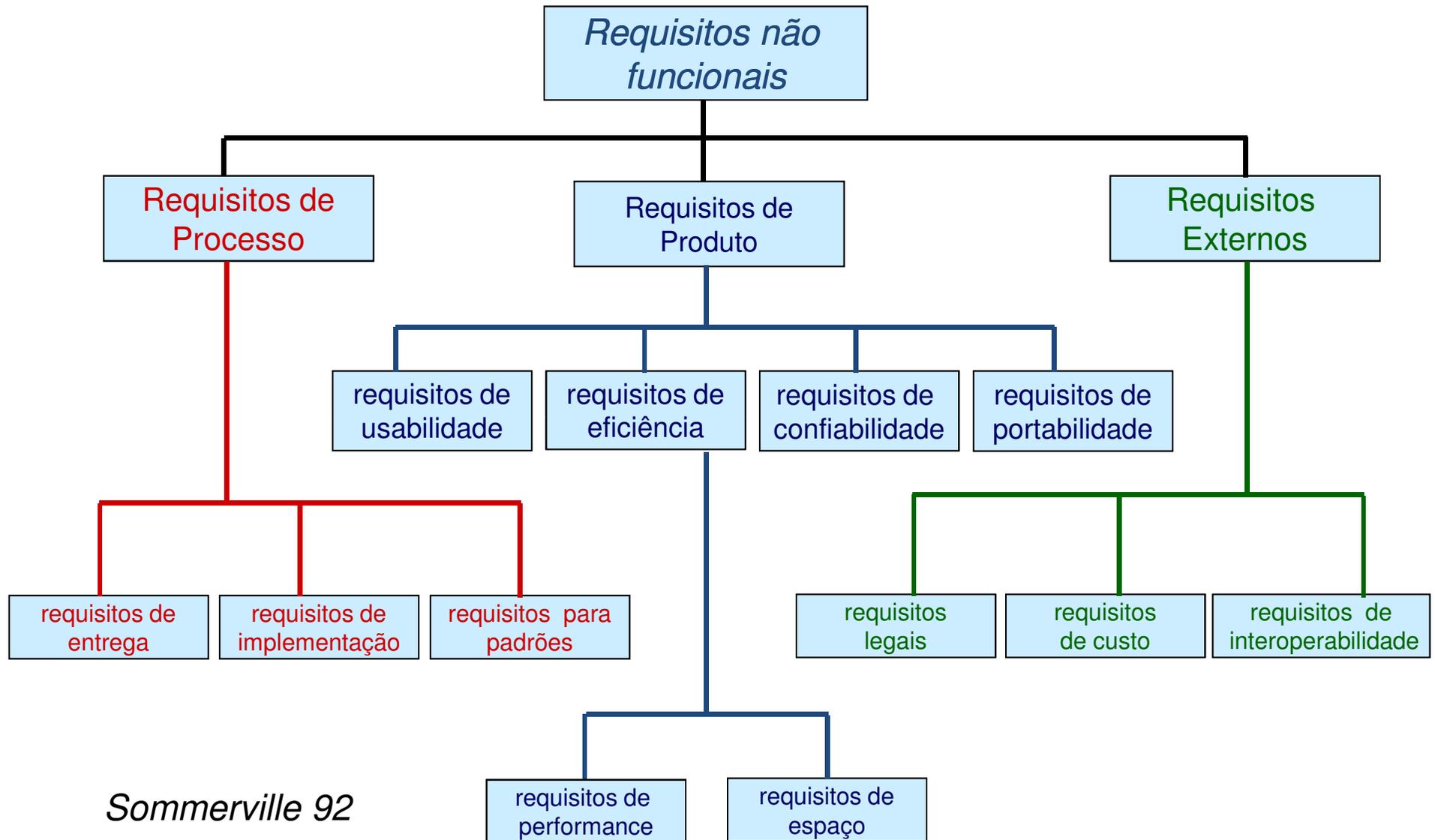
Requisitos Não Funcionais – Exemplos

- Tipo de interface desejada
- Facilidade de uso necessária
- Volume de utilização (número de usuários, número de transações, ...)
- Hardware e software alvo para o produto
- Desempenho
- Segurança
- Compatibilidade com outros produtos/versões e necessidades de migração
- **Necessidades de customização do produto pelo cliente**
- **Suporte**
- **Preço da solução**
- **Documentação necessária**
- **Uso de padrões**
- **Aspectos legais**
- **Integração com outros produtos**
- **Requisitos de instalação**
- **Tolerância a falha**

Requisitos Não Funcionais – Exemplos

Atributo	Detalhes ou condição limite
Tempo de resposta	(Condição limite) Ao registrar um item sendo vendido, a descrição e preço devem aparecer em 2 segundos
Tipo de interface	(Detalhe) Usar formulários para entrada de dados e dialog boxes (Detalhe) Maximizar a facilidade de uso via teclado e não via mouse
Tolerância a falhas	(Condição limite) Deve fazer log dos pagamentos autorizados via cartão de crédito em 24 horas, mesmo com falhas de energia ou de dispositivo

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

Requisitos Não Funcionais

- Algumas palavras levam a requisitos impossíveis de serem verificados

Palavras não Verificáveis	Possíveis substitutos
Amigável	Número máximo de passos Lista de funcionalidades presentes em outras aplicações Menus ou prompts para auxiliar usuários
Portável	Dimensões Requisitos mínimos de hardware Sistemas operacionais em que deve funcionar
Pequeno	Dimensões aceitáveis (em número de Bytes)
Flexível	Variáveis que podem acomodar uma gama de mudanças de valores Funções que implementam uma de várias possibilidades

Fundamentos de Requisitos (Conceitos)

Requisitos

- Definem o que é solicitado ao sistema fazer
- Com quais limitações ele deve operar

Gerência de Requisitos

- Mudanças que ocorrem nos requisitos já acordados;
- Relacionamentos entre os requisitos;
- Dependências entre os documentos de requisitos e outros documentos

Engenharia de Requisitos

- Métodos, técnicas e ferramentas que auxiliam o processo de descoberta, documentação e gestão dos requisitos que o software deve atender

PROCESSO DE REQUISITOS

O Sistema que queremos
deve fazer isto, isto ..., e
nesse caso também isto

Usuário



Sim, sim, estou anotando.
Pode continuar...o que
mais você quer?



PROCESSO DE REQUISITOS



Conversei com os usuários e basicamente este é o sistema que teremos que desenvolver

Ótimo, começaremos a especificar os requisitos agora mesmo!!!



PROCESSO DE REQUISITOS



Conversei com os usuários e basicamente este é o sistema que teremos que desenvolver.

Ótimo, começaremos a desenvolver o sistema agora mesmo!!!



PROCESSO DE REQUISITOS

Meses depois



Sr. Usuário, após o emprego das mais modernas técnicas de especificação, produzimos este documento que descreve minuciosamente o sistema

Ótimo... 150 páginas. Tudo isso? OK, vamos analisar e o mais breve possível voltamos a conversar

Usuário



PROCESSO DE REQUISITOS

5 semanas depois

Sra. Analista, nosso pessoal analisou com cuidado o documento. Tivemos muita dificuldade e dúvidas em entendê-lo. Mas o que percebemos é que **NÃO FOMOS CORRETAMENTE ENTENDIDOS!!!**

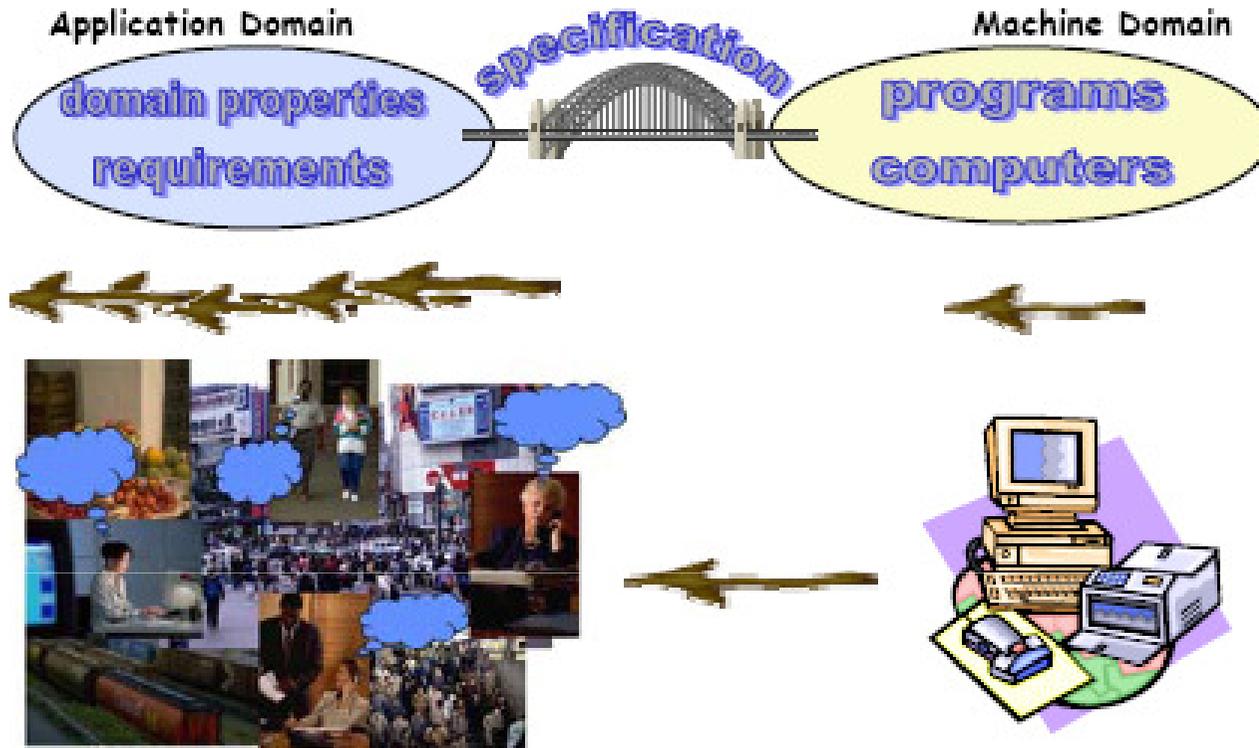
Usuário



Como não? Tudo que aí está, foi fruto de nosso entendimento pessoal. **REALMENTE VOCÊS NÃO SABEM O QUE QUEREM!!!**

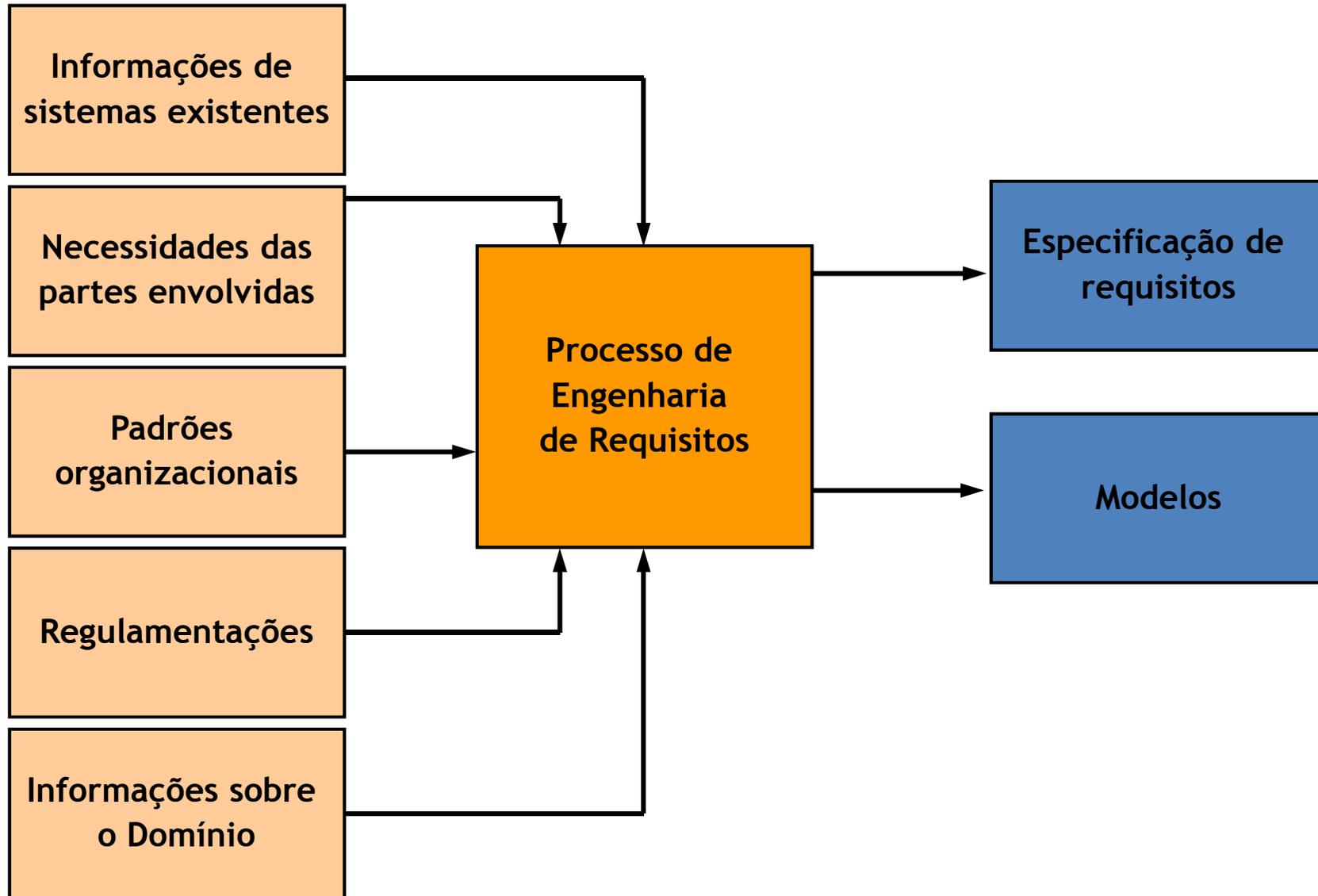


O Grande Desafio x Kickoff



- Stakeholders
 - Quem são os stakeholders?
- Fronteiras
 - Como definir o escopo do problema?
- Objetivos e cenários
- Viabilidade

PROCESSO DE REQUISITOS: Entradas e Saídas



PROCESSO DE REQUISITOS

1. Processo de requisitos variam nas empresas

- Maturidade Técnica
- Cultura Organizacional
- Domínio de aplicação

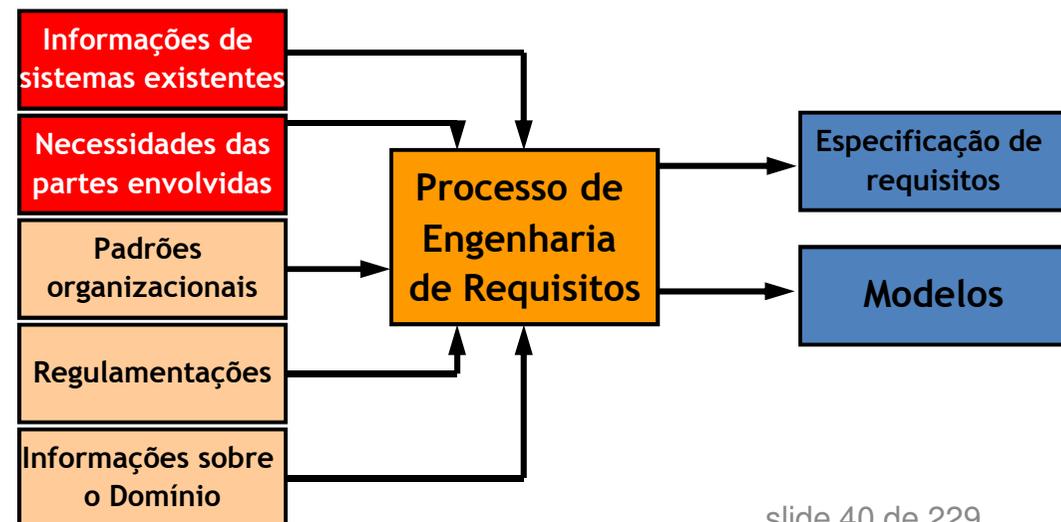
2. Não existe um processo ideal

3. Objetivos do processo de requisitos

- Estabelecer e manter concordância com os stakeholders
- Oferecer uma compreensão melhor dos requisitos do sistema
- Definir as fronteiras do sistema
- Fornecer uma base para estimar o custo e tempo
- Definir uma interface de usuário para o sistema

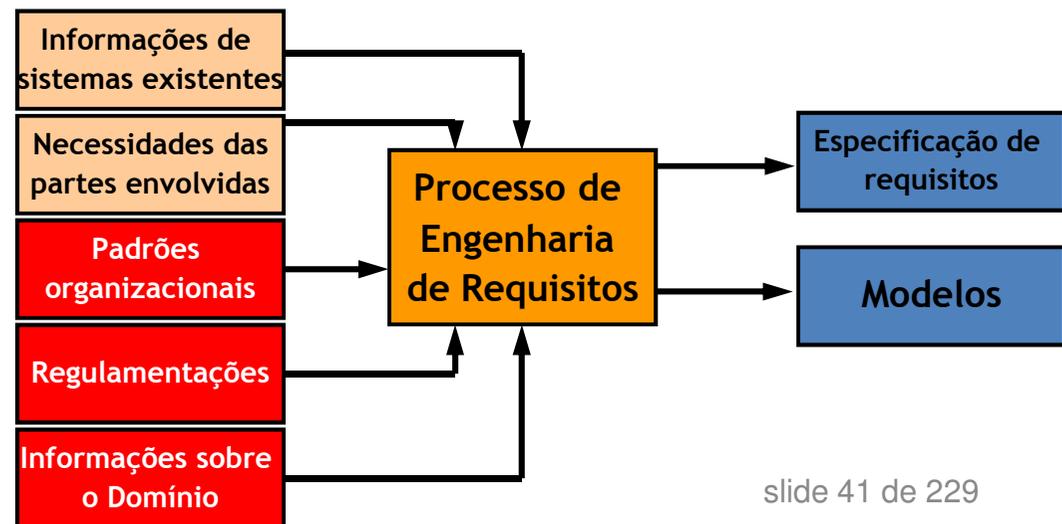
PROCESSO DE REQUISITOS – Entradas

- **Informações de sistemas existentes**
 - Informação sobre os requisitos dos softwares a serem substituídos, ou de outros software que interagem com o sistema que está sendo especificado
- **Necessidades das partes envolvidas**
 - Descrições do que os stakeholders precisam do sistema



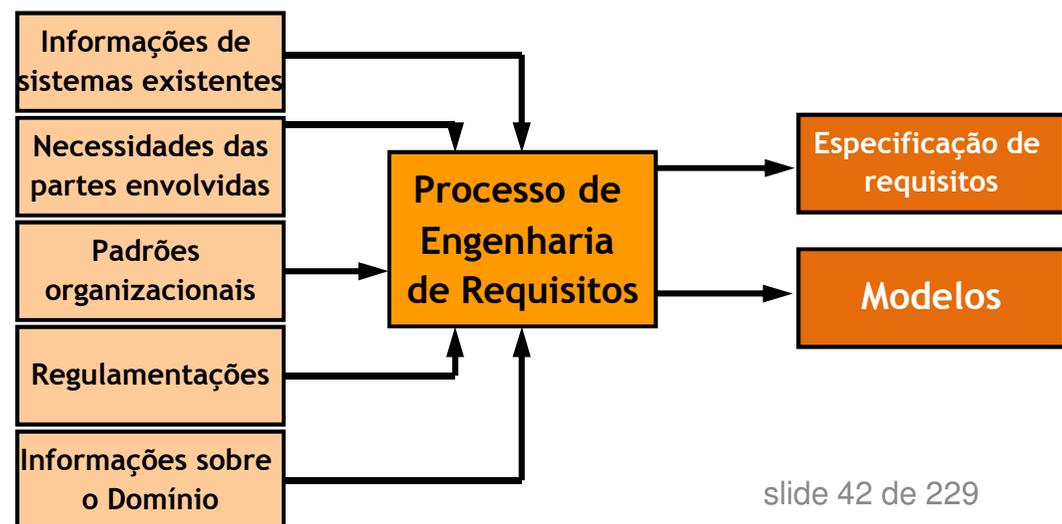
▪ PROCESSO DE REQUISITOS – Entradas

- **Padrões organizacionais**
 - Tanto os padrões utilizados na organização relacionados com práticas de desenvolvimento, qualidade etc como com o ambiente organizacional
- **Regulamentações**
 - Leis, regulamentações externas, como de saúde, ambiental, de segurança e que se aplicam ao sistema
- **Informações sobre o Domínio**
 - Informações gerais sobre o domínio de aplicação do sistema



PROCESSO DE REQUISITOS – Saídas

- Especificação de requisitos
 - Descrição dos requisitos que tenha sido entendida, acordada e validada pelos participantes
- Modelos
 - Conjunto de modelos, tais como modelo de fluxo de dados, modelos de caso de uso etc



PROCESSO DE REQUISITOS - Stakeholders

→ Processo de requisitos de software

- É um processo interdisciplinar...
- As pessoas envolvidas no processo de Engenharia de Requisitos possuem backgrounds diferentes

→ O que é mesmo um stakeholder?

- São todas as pessoas afetadas direta ou indiretamente pelo sistema.
 - Por exemplo, quem tem conta corrente é stakeholder de um sistema desenvolvido pelo Banco Central do Brasil e que permite o bloqueio de contas pela internet, à disposição dos juízes (BacenJud)
- Para cada sistema temos uma variedade enorme de stakeholders, com diferentes perfis e *backgrounds*
 - O que acaba por dificultar o processo de requisitos

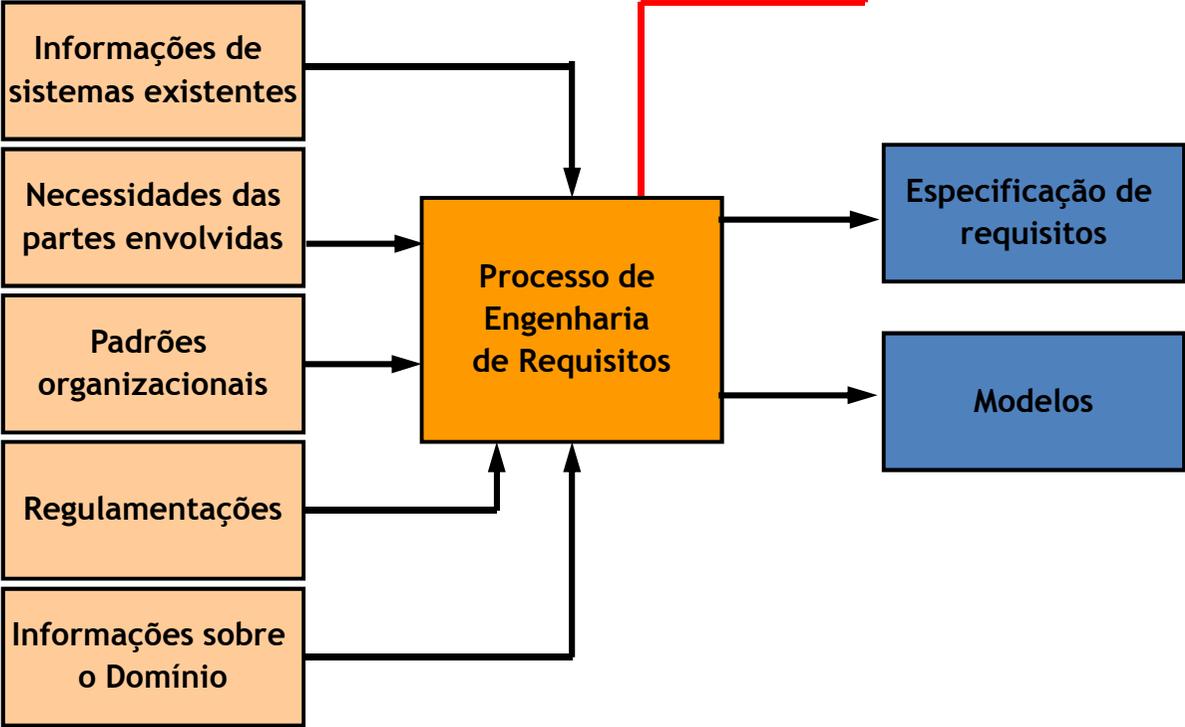
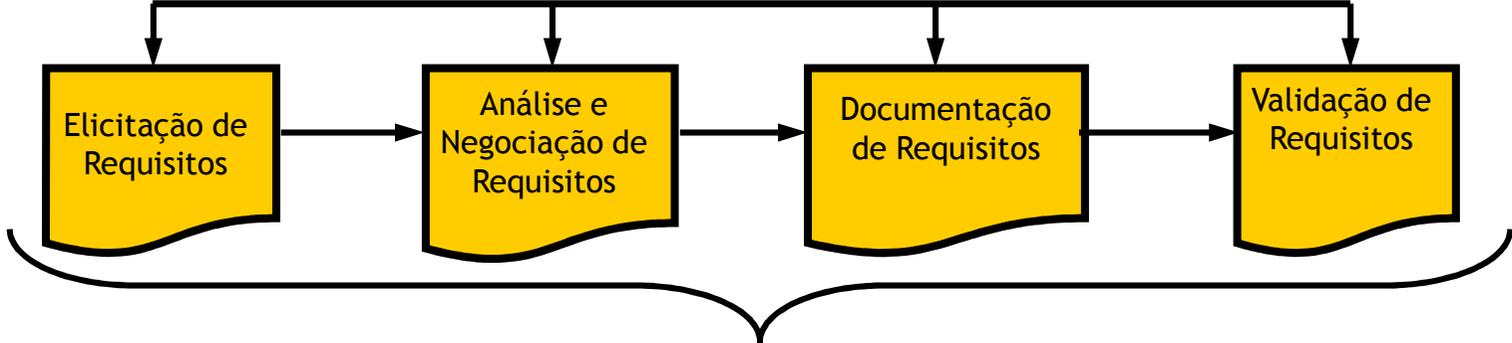
PROCESSO DE REQUISITOS - Stakeholders

→ Exemplos de stakeholders

- Engenheiros de software responsáveis pelo desenvolvimento (Responsáveis pelo desenvolvimento)
- Usuários finais do sistema (Utilização direta)
- Clientes ou patrocinadores (\$\$)
- Os fiscais que verificaram se o sistema satisfaz os requisitos legais
- Especialistas de domínio (Quem realmente entende!)

→ É muito importante correta identificação do conjunto dos stakeholders

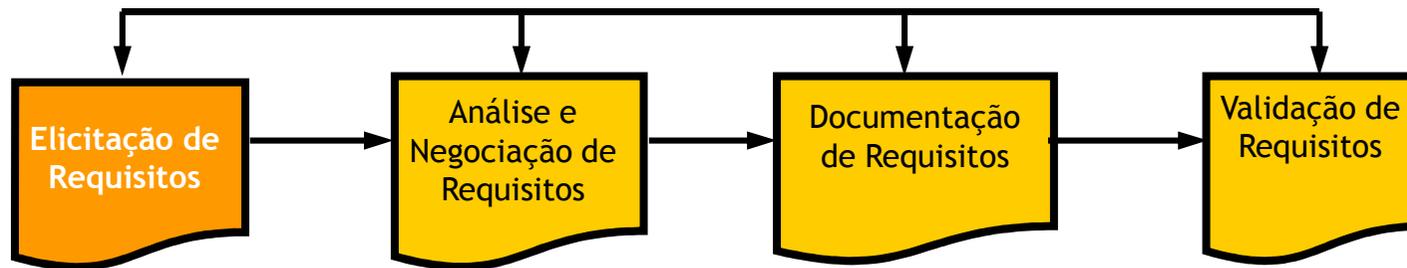
PROCESSO DE REQUISITOS - Atividades



ELICITAÇÃO DE REQUISITOS

Vamos olhar agora especificamente a Elicitação de Requisitos

O Processo de Requisitos



ELICITAÇÃO DE REQUISITOS

→ **Consiste em:**

- **Descobrir**

- **Explicitar**

- Obter o *máximo de informações* para o entendimento do objeto em questão
- Refere-se ao processo de extração de informação sobre a(s) funcionalidade(s) requisitada(s) e outras propriedades do sistema
- “É o **primeiro estágio** na construção e **entendimento do problema (correto e completo!!)** que o software deve resolver”
[SWEBOK]

ELICITAÇÃO DE REQUISITOS

→ Elicitação FAZ

- Identificação de fontes de informação
- Comunicação
- Coleta

→ Elicitação USA

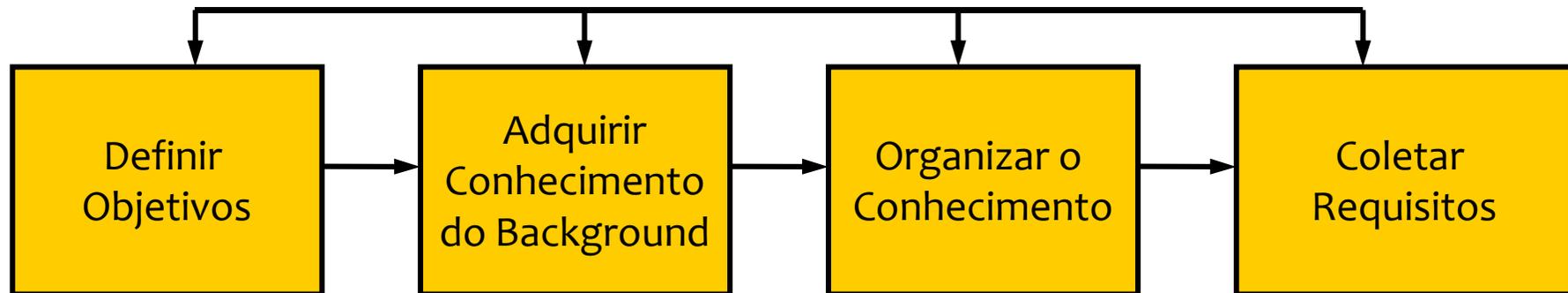
- Pessoal
- Métodos
- Ferramentas

→ Elicitação DEPENDE DE

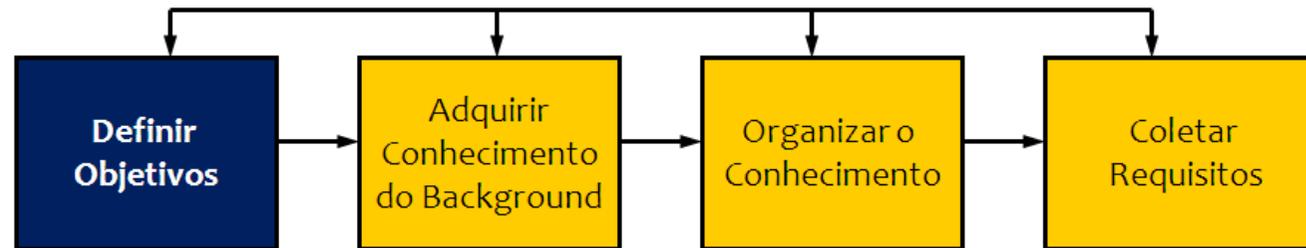
- Pontos de Vista

ELICITAÇÃO DE REQUISITOS

Processo de Elicitação de Requisitos



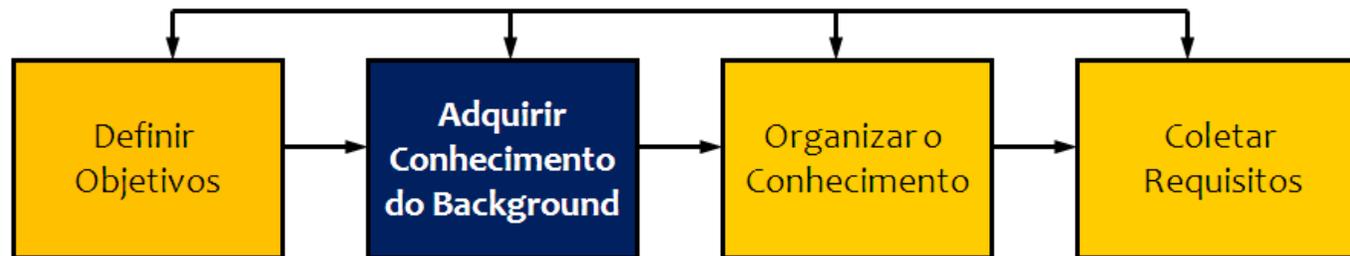
ELICITAÇÃO DE REQUISITOS - Processo



→ Definir Objetivos

- objetivos gerais do negócio
- descrição geral do problema a ser resolvido
- por que o sistema é necessário
- limitações do sistema

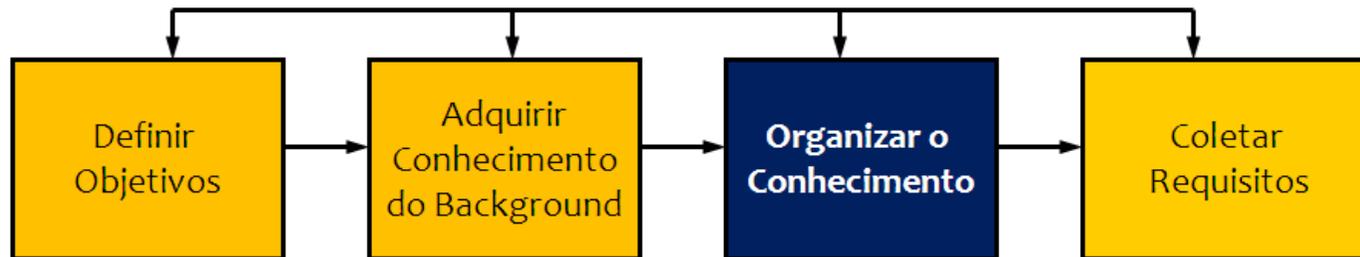
ELICITAÇÃO DE REQUISITOS - Processo



→ **Adquirir Conhecimento do Background**

- informações sobre a organização onde o sistema será instalado,
- o domínio de aplicação do sistema,
- informações de outros sistemas existentes

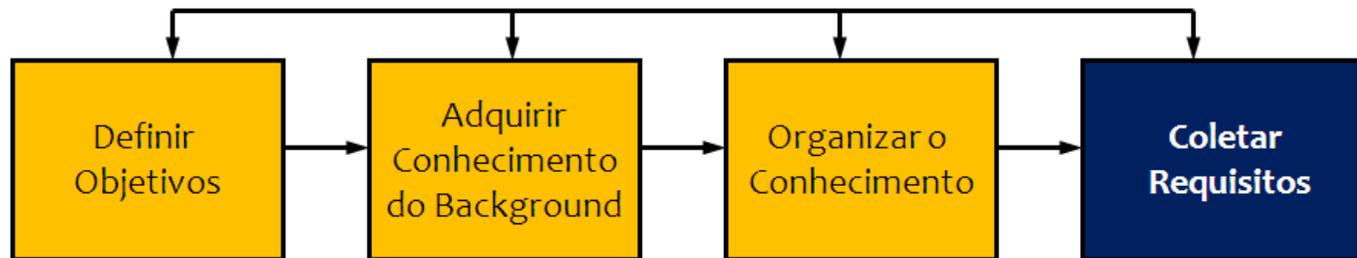
ELICITAÇÃO DE REQUISITOS - Processo



→ Organizar o conhecimento

- O conhecimento coletado deve ser organizado

ELICITAÇÃO DE REQUISITOS - Processo



→ Coletar requisitos

- Os stakeholders do sistema devem ser consultados para a descoberta de seus requisitos
- Técnicas de elicitação

Coleta de Fatos

- Leitura de documentos
- Observação
- Entrevistas
- Reuniões
- Questionários
- Antropologia
- Participação ativa dos atores
- Bases de Requisitos não funcionais
- Engenharia Reversa
- Reutilização

ELICITAÇÃO DE REQUISITOS

→ Técnicas de elicitação de requisitos

- Utilizadas para extrair informações
 - Área delicada: nem sempre o usuário consegue exprimir suas necessidades, tarefas, etc.
- As técnicas são complementares entre si
- Servem para estruturar “conversas”
 - O objetivo maior é **extrair informação**, seja uma ou outra técnica sendo utilizada

ELICITAÇÃO DE REQUISITOS - Técnicas

▪ Entrevistas

- O analista conversa com com diferentes stakeholders para obter um entendimento dos requisitos
- Conduzir entrevistas com um grupo restrito de [stakeholders](#).
- Lista de perguntas elaboradas para se obter uma compreensão dos problemas reais e das possíveis soluções.

▪ Leitura de Documentos

- O analista busca documentação que possa dar suporte ao entendimento e extração dos requisitos

▪ Questionários

- Pode funcionar como uma pesquisa qualitativa
- Permite análises estatísticas

ELICITAÇÃO DE REQUISITOS

▪ Prototipação

- Uma abordagem valiosa para clarear requisitos.
- Levam os usuário a um contexto onde conseguem entender melhor quais informações eles precisam prover
- Geralmente, mais adequado do que apresentar documentos de texto

▪ Workshops, reuniões facilitadas

- Trabalhar com grupos pode agregar mais do que trabalhar individualmente
- Bom porque conflitos nos requisitos surgem antecipadamente
- Uso da técnica de brainstorming, para em seguida o facilitador conduzir o grupo na organização.

ELICITAÇÃO DE REQUISITOS

■ Cenários

- Busca uma seqüência específica de ações que ilustra comportamentos
 - O mais comum tipo de cenário é o **caso de uso** (descrição de comportamento do sistema em termos de seqüências de ações)
- Existem diversas outras técnicas...
- Entender **necessidades** ⇒ produzir **requisitos**



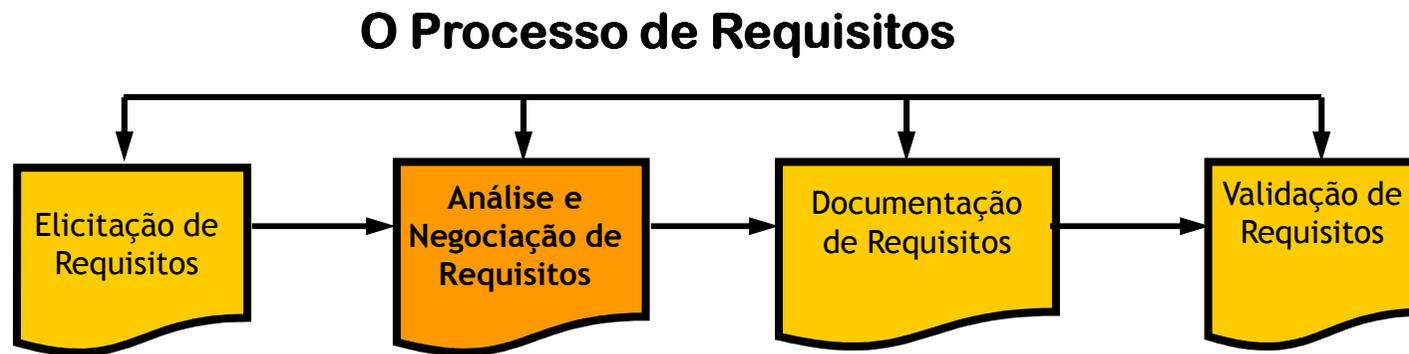
EXERCÍCIO

Mini Workshop de Requisitos

- **Objetivo:** Delimitar o escopo do sistema que vai interligar o controle de estacionamento de um shopping com a central da Polícia Civil, a fim de identificar carros roubados que estejam entrando no shopping
- **Utilização de *brainstorming* (tempestade de idéias)**
- **Regras**
 - Expressar idéias livremente
 - Evitar atitude crítica
 - Registrar idéias
 - Fechar um consenso
 - Documentar o escopo (sugestão: lista de funcionalidades)

ANÁLISE E NEGOCIAÇÃO DE REQUISITOS

Vamos olhar agora especificamente a Análise e Negociação de Requisitos



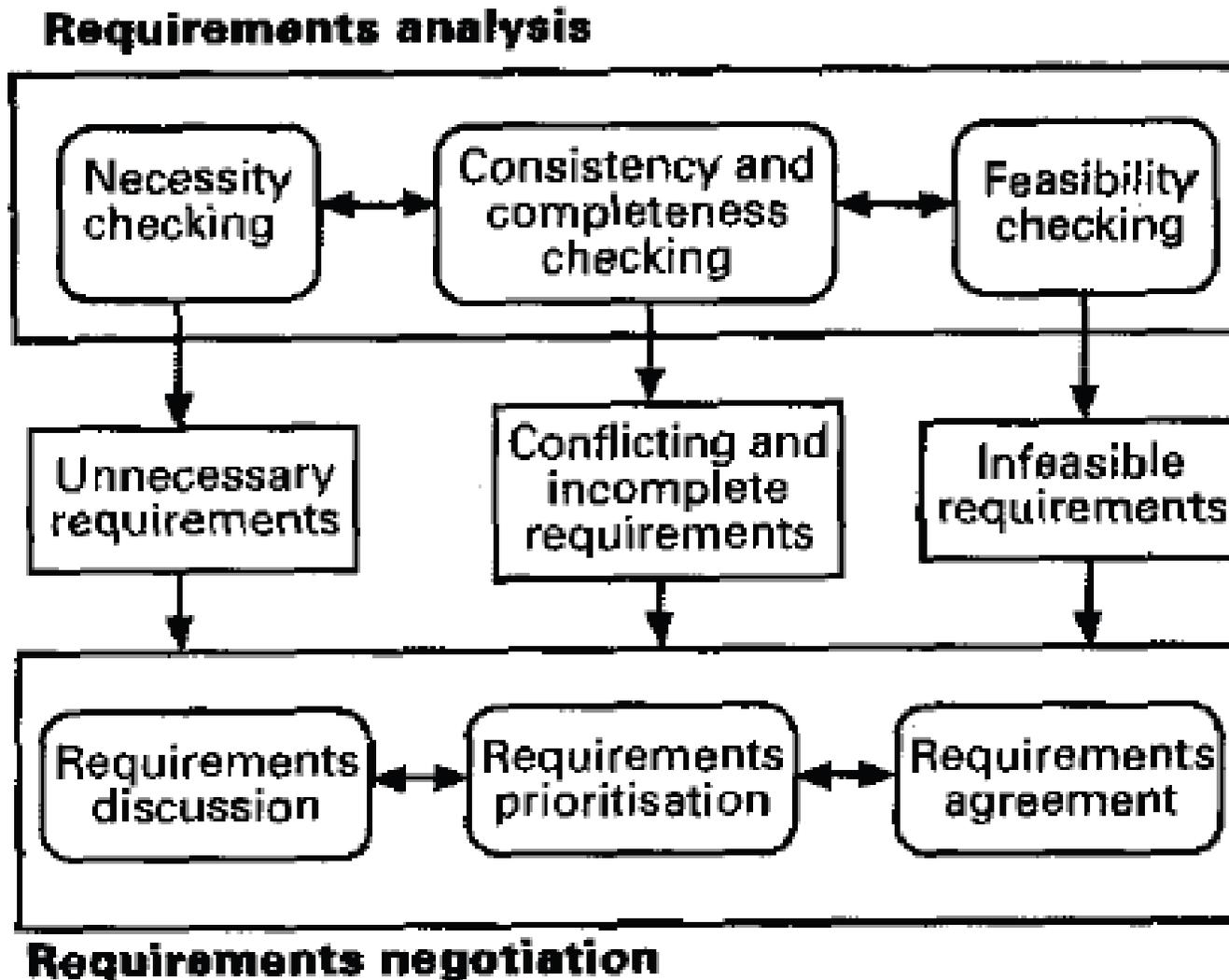
ANÁLISE E NEGOCIAÇÃO DE

- Objetivos: REQUISITOS
 - atividades que visam **descobrir problemas** com os requisitos e chegar a acordos para a sua resolução de forma a satisfazer todos os interessados no sistema
- na fase de elicitação **já se realizam** atividades de análise e negociação
- análise e negociação de requisitos são atividades que incidem sobre **conjuntos incompletos** de requisitos

ANÁLISE E NEGOCIAÇÃO DE REQUISITOS

- a análise e negociação de requisitos é normalmente um processo **complexo**
- requer pessoas com competências **específicas**
- baseia-se muito no **juízo e experiência** dos participantes
- **não é possível** transformar este processo numa abordagem estruturada e sistemática

ANÁLISE E NEGOCIAÇÃO DE REQUISITOS



ANÁLISE E NEGOCIAÇÃO DE REQUISITOS

→ **Análise de requisitos**

- É importante analisar se os requisitos estão precisamente definidos
 - Para que possam ser validados
 - Suas implementações verificadas
 - E seus custos estimados

- O objetivo principal é descobrir incompletude e inconsistências!
 - **Incompletude** quer dizer que nem todos os requisitos foram elicitados, levantados, e surge a sensação que está faltando algo,
 - **Inconsistência** está relacionada com as contradições que podem existir entre os requisitos.

ANÁLISE E NEGOCIAÇÃO DE REQUISITOS

Processo de Análise de Requisitos



ANÁLISE DE REQUISITOS



→ Verificação da necessidade

- Alguns requisitos podem não contribuir ou até mesmo divergir dos objetivos do negócio ou da organização

ANÁLISE DE REQUISITOS



→ Verificação da consistência e completude

- Busca de requisitos contraditórios e de todos os serviços e limitações oriundas das necessidades dos usuários

ANÁLISE DE REQUISITOS



→ Verificação da viabilidade

- Os requisitos são avaliados para saber se são viáveis, levando em consideração o orçamento e o tempo disponível

Atributos de uma Boa Especificação

- Clareza
- ! Ambígua
- Completa
- Simples
- Bem escrita

Clareza

Um requisito claro

Tipo de usuário	O engenheiro de teste...
Resultado desejado	...simula...
Objeto	...erros de componente
Condições	...utilizando as funções de teste QQ e TT.

Um requisito vago

	Em geral o sistema...
Precisa ou não?	... deve ser capaz...
Quais?	...de diagnosticar possíveis erros...
Como verificar isto?	... em um prazo razoável.

Ambiguidade

- “O sistema deve enviar relatórios de produtividade dos programadores, analistas ou desenvolvedores do projeto mensalmente **ou** quando requisitado.”
- "Realizar rotina de importação de dados **periódica** de preço de fluido“
- "Identificar e associar as intervenções que são **complementares às outras**"
- O sistema deve emitir uma mensagem de atenção visual ou auditiva no evento de falha do sistema de refrigeração.

Requisitos Incompletos

- Curva S (Planejado X Realizado) de um projeto
- Cadastro de iniciativas estratégicas
- Cadastro de iniciativas de melhoria
- Acompanhamento das atividades
- Acompanhamento dos projetos (percentual de conclusão)

Requisitos Múltiplos

- No evento de falha da rede elétrica, o sistema deve enviar mensagem de erro ao usuário, salvar a configuração atual do sistema e os dados entrados, até então.
- O sistema deve manter dados estatísticos sobre compra, venda e movimentação do estoque de materiais de limpeza. Informação relativa a comissão de vendedores também deve ser mantida.
- Cadastro das atividades de um projeto e produtos e funcionários alocados na atividade

Requisitos com alternativas

- O sistema deve mostrar o total do pedido a medida em que os códigos dos produtos vão sendo entrados no pedido, a não ser que se trate de um produto promocional.
- Evitar: Mas, com exceção, apesar, quando...

Requisitos mal escritos

- Projetos coordenados por um funcionário
- O sistema poderá ser acessado remotamente por qualquer unidade internacional da Petrobras, com isso, ele deverá ter um desempenho compatível ao acesso.
- Na improvável eventualidade de falha no sistema de refrigeração, o sistema deve mandar mensagem para a chave *admin*

ANÁLISE DE REQUISITOS

→ **A classificação dos requisitos é parte do processo de análise**

- Requisitos funcionais e não funcionais
- Requisitos do produto e do processo
- Quanto a prioridades dos requisitos
- Quanto ao escopo dos requisitos
- Quanto a volatilidade/estabilidade
- Entre outras...

→ **Interessante a classificação quanto ao impacto na arquitetura**

- Atividade priorizar casos de uso (arquiteto de sw, fase de elaboração)
- Identificar aqueles requisitos que são arquiteturalmente críticos

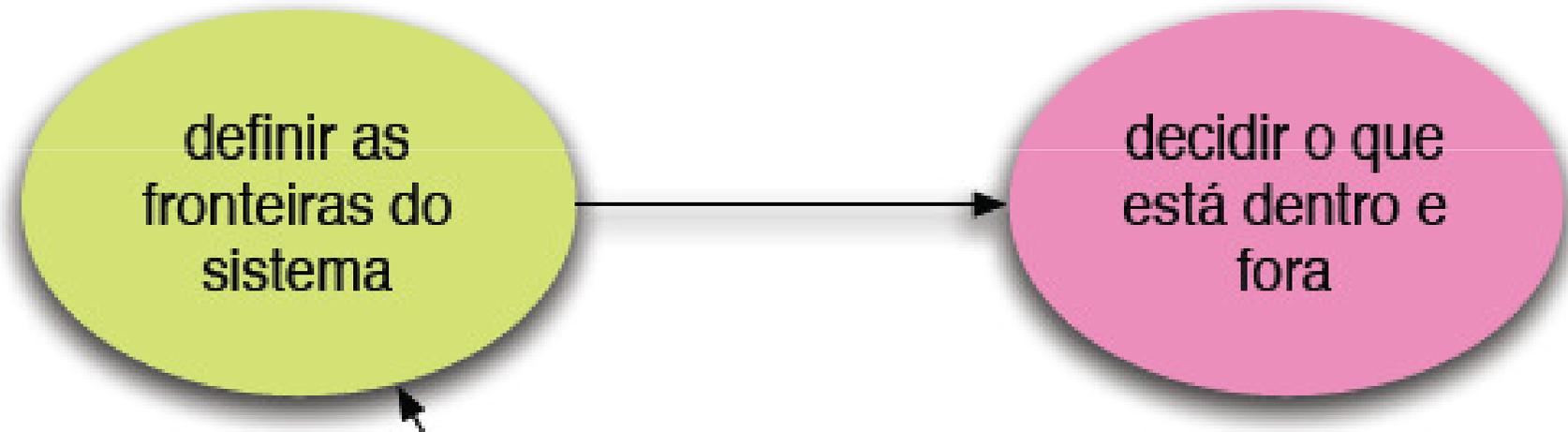
ANÁLISE DE REQUISITOS - Técnicas

Checklist

- ✓ o requisito poderia ser decomposto em sub-requisitos?
- ✓ o requisito é mesmo necessário?...
- ✓ o requisito implica a utilização de software não standard?
- ✓ o requisito está de acordo com os objectivos do negócio?
- ✓ o requisito é ambíguo?
- ✓ o requisito é realista?
- ✓ o requisito é "testável"?

ANÁLISE DE REQUISITOS - Técnicas

Análise do Escopo



ANÁLISE DE REQUISITOS - Técnicas

Análise de Dependências

- ▶ matriz de dependências

requisito	r1	r2	r3	r4
r1	x	x	x	x
r2	conflito	x	x	x
r3			x	x
r4		sobreposição	sobreposição	x

ANÁLISE DE REQUISITOS - Técnicas

Prioridades

✓ definir **prioridades** na análise e implementação dos requisitos

✓ **classificação:**

✓ alta, média, baixa, n/s

✓ essencial, importante, desejável, a ser decidido

ANÁLISE DE REQUISITOS - Organização

Classificação

- ✓ agrupamento de requisitos para melhor manipulação e visualização
- ✓ exemplos:
 - ✓ atendimento, vendas, pesquisa, fluxo de trabalho, etc.

ANÁLISE DE REQUISITOS - Organização

Identificadores

- ✓ número sequencial
 - ✓ número sequencial numa hierarquia do documento
 - ✓ número sequencial dentro de uma categoria: RF001, RNF005,...

NEGOCIAÇÃO DE REQUISITOS

→ Negociação de requisitos

- Algumas literaturas apresentam a área de negociação como parte da análise de requisitos, como por exemplo o SWEBOK, pois os problemas surgem como resultado da análise
- Resolução de conflitos
 - ✓ 2 stakeholders requisitando RFs incompatíveis
 - ✓ **RNFs de alto desempenho e segurança**: Quanto maior a segurança (mais complexo um algoritmo criptográfico) menor o desempenho da aplicação
- O especialista em requisitos deve ter habilidades para resolver conflitos

NEGOCIAÇÃO DE REQUISITOS

Negociar pra que?

- ✓ chegar a um **acordo** em relação a **opções** mais **adequadas** aos interesses dos stakeholders
- ✓ definir as **prioridades** dos requisitos
- ✓ para novas iterações de **elicitação** e **análise** e para o **desenvolvimento**
- ✓ chegar a um acordo em relação a requisitos que estão em **conflito**

NEGOCIAÇÃO DE REQUISITOS

Problemas:

- ✓ separar os aspectos importantes que serão debatidos das questões pessoais
- ✓ falta de entendimento compartilhado e de pontos de vista pessoais
- ✓ atitudes interpessoais

NEGOCIAÇÃO DE REQUISITOS

Técnicas para gerir problemas:

✓ lidar com os ataques pessoais:

✓ **evitá-los**... se acontecerem: mudar de assunto, fazer um intervalo...

✓ resolver o problema **fora da reunião**

✓ **reações negativas** sem justificativa: "isso não funciona", "não dá pra fazer", "dá muito trabalho", "tá doido?"

✓ incitar os participantes a **justificarem** a negativa

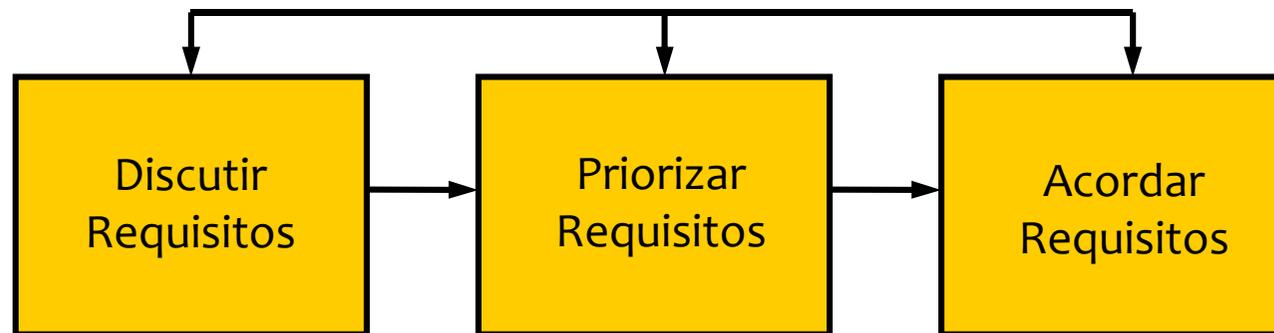
✓ lidar com conflitos entre grupos

✓ **choque de pontos de vista** entre grupos

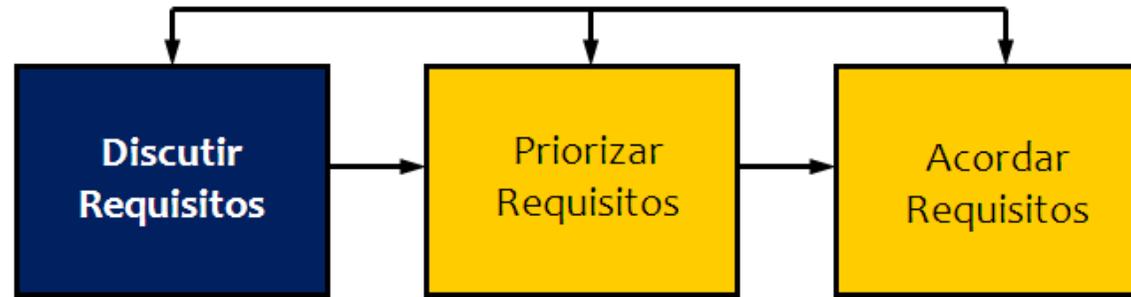
✓ exemplos: qualidade x prazos, custos x desenvolvimento, segurança x acesso, complexidade x usabilidade, etc.

NEGOCIAÇÃO DE REQUISITOS

Processo de Negociação de Requisitos



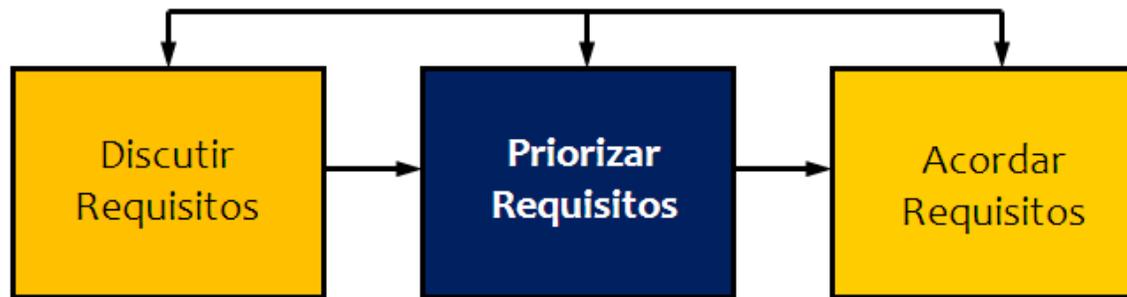
Negociação de Requisitos



→ Discutir requisitos

- problemas identificados nos requisitos são levantados para que os stakeholders apresentem seus pontos de vista

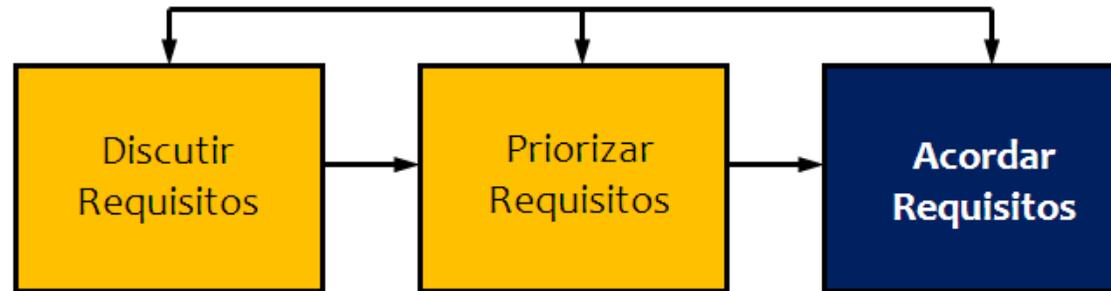
Negociação de Requisitos



→ Priorizar requisitos

- os requisitos são priorizados de acordo com as classificações existentes

Negociação de Requisitos

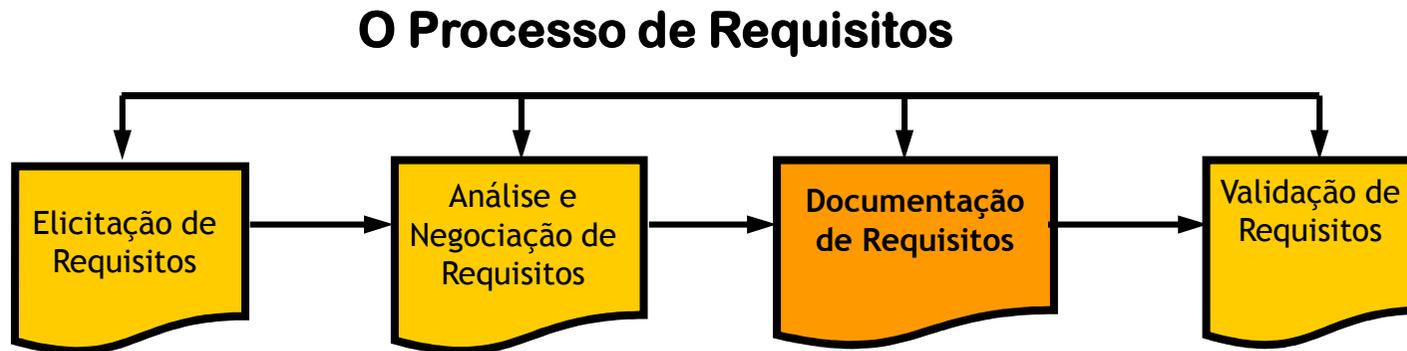


→ Acordar requisitos

- soluções para os problemas dos requisitos são identificadas e acordos são fechados.
- muitas vezes envolve mudanças em alguns dos requisitos

DOCUMENTAÇÃO DE REQUISITOS

Vamos olhar agora especificamente a Documentação de Requisitos



- A documentação de requisitos está relacionada com a produção de um conjunto de documentos que possam ser sistematicamente revisados, evoluídos e aprovados.

DOCUMENTAÇÃO DE REQUISITOS

→ Documento de Requisitos (ou Visão)

- **Especificação do produto** a ser desenvolvido, em termos das **necessidades e características** mais importantes.
- Por conter uma descrição dos requisitos centrais pretendidos, proporciona **a base contratual** para requisitos técnicos mais detalhados

→ Glossário

- Define **termos** importantes usados pelo projeto

→ Modelo de Caso de Uso

- O modelo de casos de uso é um modelo das **funções pretendidas do sistema** e como será sua interação com o ambiente

DOCUMENTAÇÃO DE REQUISITOS

→ Protótipo de Interface com Usuário

- Representação dos campos, comandos e navegabilidade entre as telas da aplicação

→ Especificações de Casos de Uso

- **Seqüência de ações** realizada pelo sistema que **produz um resultado** de valor observável para determinado **ator**.
- Utilizado para **detalhar** cada **requisito** com seus fluxos de processamento.
- As informações contidas em documentos desse tipo serão a base para a implementação e realização de testes.

→ Matrizes de Rastreabilidade

- Repositório de dependências e atributos dos requisitos com o objetivo de facilitar o gerenciamento de requisitos

DOCUMENTAÇÃO DE REQUISITOS

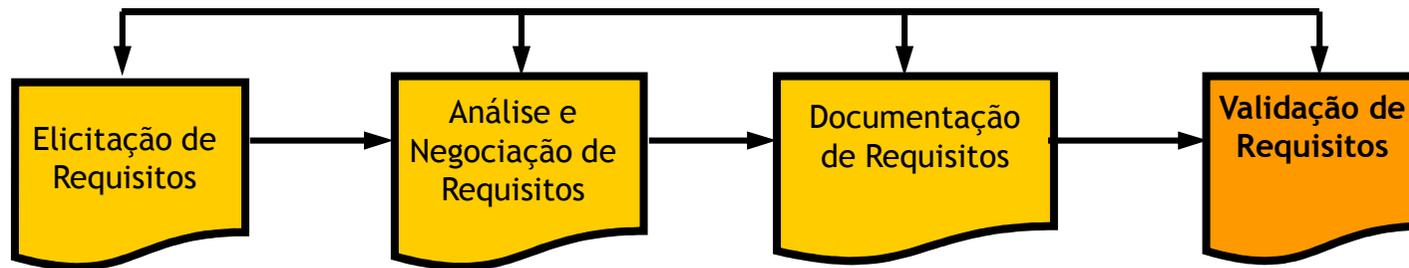
→ Plano de Gerenciamento de Requisitos

- Descreve a documentação de requisitos, os tipos de requisitos e seus respectivos atributos de requisitos, especificando as informações e os mecanismos de controle que devem ser coletados e usados para avaliar, relatar e controlar mudanças nos requisitos

VALIDAÇÃO DE REQUISITOS

Vamos olhar agora especificamente a Validação de Requisitos

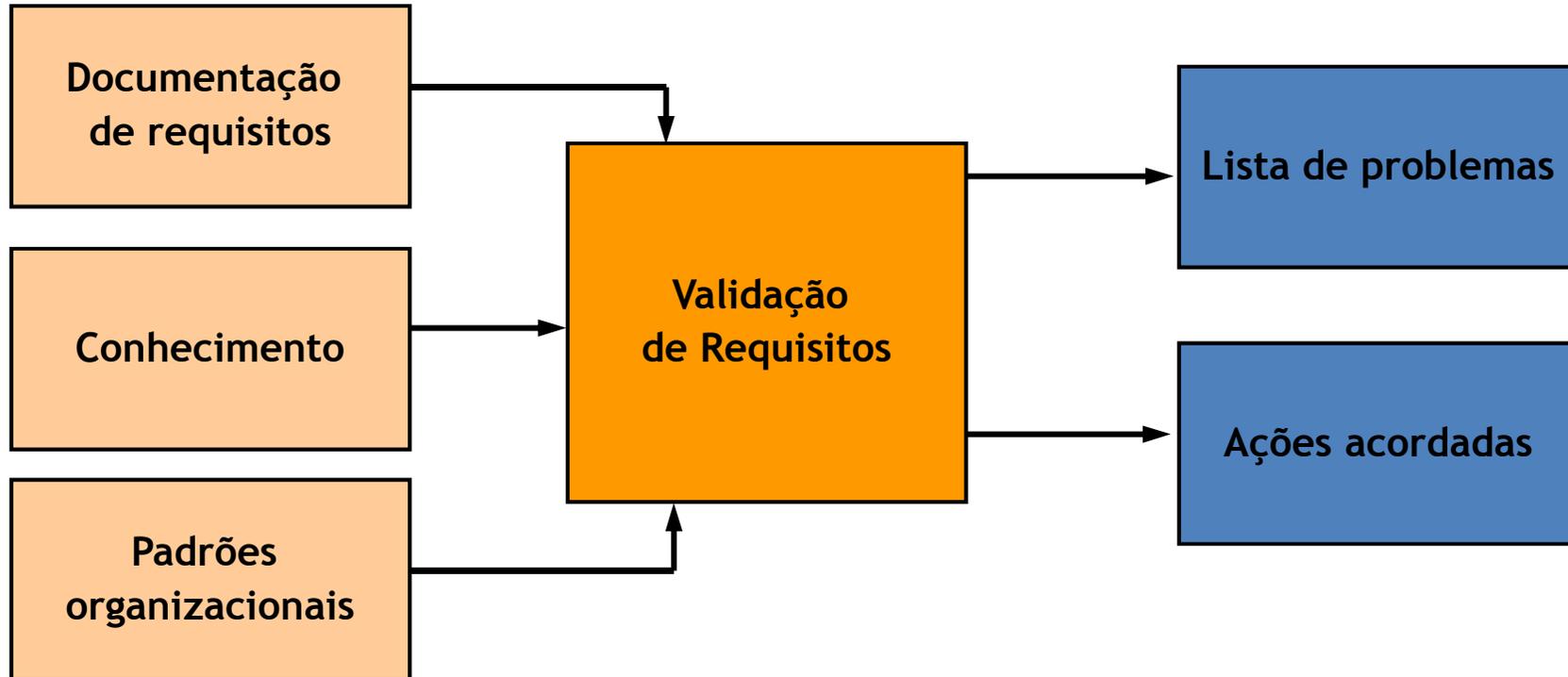
O Processo de Requisitos



VALIDAÇÃO DE REQUISITOS

- A validação de requisitos deve certificar que os requisitos realmente definem o sistema que o usuário quer
- Alguns aspectos dos requisitos devem ser estudados
 - Validade
 - Consistência
 - Completude
 - Realismo
- Validação de requisitos engloba Validação e Verificação
 - **Validação:** A documentação deve ser validada para garantir que os engenheiros entenderam os requisitos, que a vontade do usuário está realmente atendida pelo que foi documentado
 - **Verificação:** A documentação deve ser verificada conforme os padrões organizacionais, e deve estar entendível, consistente e completa

VALIDAÇÃO DE REQUISITOS



VALIDAÇÃO DE REQUISITOS

- **Validação pode ser feita por revisões**
- **Validação pode ser feita por prototipação**
 - Quando já existe um protótipo de elicitação
 - Algo concreto para criticar
- **Validação pode ser feita por teste de requisitos**
 - **Casos de teste**
 - **Qual cenário de uso poderá ser usado para testar um requisito?**
O requisito, sozinho, inclui informação suficiente para a definição de um teste? É possível testar o requisito usando um único teste ou são necessários múltiplos testes?
- **Representantes dos usuários e dos desenvolvedores devem participar das revisões**

VALIDAÇÃO DE REQUISITOS

→ **As revisões devem ser orientadas por algumas atividades**

- **Planejar revisão**
 - Selecionar uma equipe de revisão, hora e local para a reunião de revisão
- **Distribuir documentos**
 - Toda a documentação é distribuída entre os membros da equipe de revisão
- **Realizar revisão propriamente dita**
 - Comentários individuais são expostos ao grupo e um conjunto de ações deve ser proposto
- **Acompanhar acordos fechados**
 - O responsável pelo processo de revisão verifica se todas as ações acertadas foram executadas
- **Revisar documentos**
 - A documentação é revisada para refletir as ações concordadas

VALIDAÇÃO DE REQUISITOS

→ Possíveis ações nas revisões de requisitos

- **Completar informações**
 - O especialista em requisitos que está revisando a documentação deve descobrir a informação que falta com os stakeholders
- **Clarificar requisitos**
 - O requisito pode ter sido mal escrito ou pode ter omitido alguma informação, devendo ser reescrito
- **Resolver conflitos nos requisitos**
 - Os stakeholders envolvidos devem negociar para resolver o conflito
- **Tornar requisito mais realístico**
 - Os stakeholders devem ser consultados para decidir como tornar o requisito mais realístico
- **Tornar requisito testável**
 - Os requisitos devem ser os mais objetivos possíveis

VALIDAÇÃO DE REQUISITOS

→ É comum utilizar *checklists* de revisão

- **Entendimento**
 - Os usuários da documentação realmente entendem o significado dos requisitos?
- **Redundância**
 - Há repetição desnecessária de informação na documentação?
- **Completude**
 - Alguma informação está omissa na descrição dos requisitos?
- **Ambigüidade**
 - Os requisitos estão objetivamente definidos?

VALIDAÇÃO DE REQUISITOS

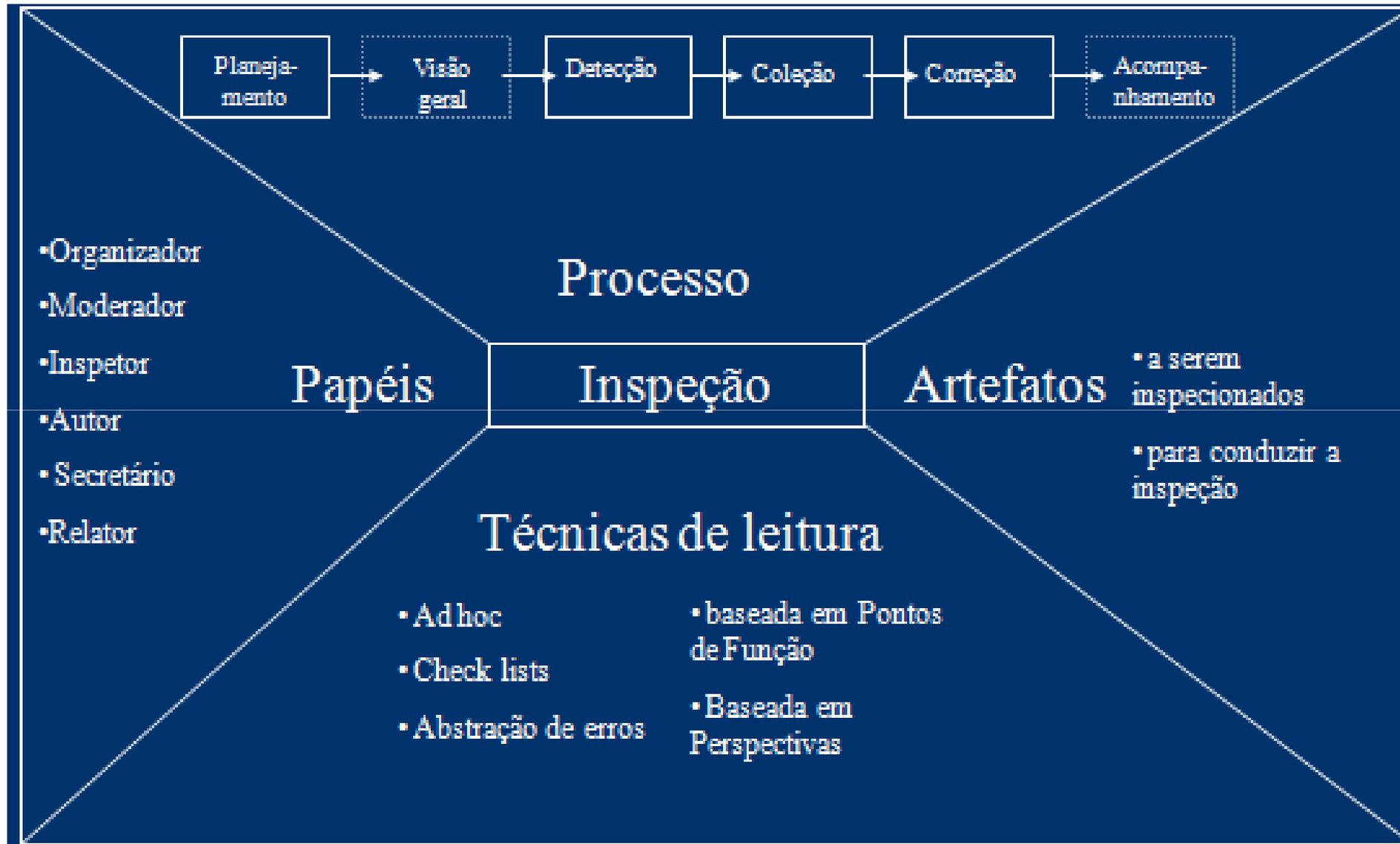
→ É comum utilizar *checklists* de revisão

- **Consistência**
 - Alguns requisitos se mostram contraditórios?
- **Organização**
 - A estruturação da documentação é adequada?
 - Existem pacotes de requisitos?
 - Cada requisito está unicamente identificado?
- **Padronização**
 - Os padrões estão sendo respeitados?
- **Rastreabilidade**
 - É possível acompanhar a vida do requisitos antes e após sua inclusão na documentação?
 - Os requisitos estão relacionados entre si?

Inspeções

- criadas em 1972 por Fagan, na IBM, para melhoria da qualidade de código
- hoje são utilizadas em qualquer tipo de artefato gerado pelo processo de desenvolvimento
- inspeção pode detectar de 30% a 90% dos erros existentes
- técnica de leitura aplicável a um artefato, buscando a localização de erros ou defeitos no mesmo, segundo um critério pré-estabelecido

Inspeções em Requisitos



Inspeções em Requisitos

- Inspeções baseadas em checklists:
 - Inspetores utilizam uma lista com os itens a serem verificados
 - cada artefato tem uma lista específica (Documento de Requisitos, Casos de Uso, Glossários, Léxico Ampliado da Linguagem...)
 - os defeitos são anotados no artefato sendo analisado
 - após a revisão, é realizada uma reunião onde os problemas encontrados são relatados aos desenvolvedores

Checklists de Inspeção

- Pontos a serem avaliados/observados durante o processo de inspeção
- Depende do material a ser inspecionado (casos de uso, cenários, requisitos,...)
- Depende do enfoque da inspeção
- Taxonomia dos defeitos - o que procurar

Exemplo OO

- Checklist OO:
 - Todas as classes são representadas por retângulos com 1,2 ou 3 compartimentos?
 - As classes possuem nomes diferentes?
 - Existem classes sem relacionamentos definidos?
 - Os atributos e os métodos de cada classe são adequados aquela classe?
 - Todo comportamento especificado é possível de ser contemplado através das associações do modelo?

VALIDAÇÃO DE REQUISITOS

Prototipação:

- ✓ Protótipos para validação de requisitos **demonstram** os requisitos e ajudam aos stakeholders descobrirem problemas
- ✓ Protótipos para validação devem ser **completos**, razoavelmente eficientes e robustos.
- ✓ Deverá ser possível usá-los da mesma forma que o sistema
- ✓ Documentação mínima e treinamento devem ser providenciados

Atividades: Prototipação

- ✓ Escolha os testadores do protótipo
 - ✓ Os **melhores testadores** são os **usuários** bem experientes e que tenham **cabeça aberta** sobre o uso do novo sistema
 - ✓
 - ✓ **Usuários finais** que tem **funções diferentes** devem estar envolvidos para que **diferentes áreas** da funcionalidade do sistema possa ser coberta

Atividades: Prototipação

- ✓ Desenvolva os cenários de teste
 - ✓ É necessário um **planejamento detalhado** para preparar um **conjunto de cenários de teste amplo**, que faça cobertura de uma grande quantidade de requisitos

- ✓ Os usuários finais **não** devem apenas **“brincar”** com o sistema, para evitar não exercitar **aspectos críticos** do mesmo

Atividades: Prototipação

✓ Execute cenários

- ✓ Colocar os usuários do sistema, geralmente sozinhos, para testar o sistema através da execução dos cenários planejados

✓ Documente problemas

- ✓ Definir algum formulário de problemas eletrônico (ou papel) para que os usuários possam preencher ao encontrar um problema

Testes de Requisitos

- ✓ Cada requisito deve ser **testável**, isto é deverá ser possível definir teste para checar se o requisito foi ou não **alcançado**
- ✓ Testes de requisitos é uma técnica **efetiva** de validação pois informação ambígua ou incompleta **dificulta** a formulação dos testes
- ✓ Cada requisito funcional deve ter um teste associado

Teste de Requisitos

Definição de casos de teste:

- ✓ Qual cenário de uso poderá ser usado para testar um requisito?
- ✓ O requisito, sozinho, inclui informação suficiente para a definição de um teste?
- ✓ É possível testar o requisito usando um único teste ou são necessários múltiplos testes?
- ✓ O requisito poderá ser re-escrito para tornar os casos de teste mais óbvios?

Teste de Requisitos

Exemplo de teste:

✓ 10.(iv) Quando os usuários acessarem o EDDIS, eles serão apresentados à páginas web com todos os serviços disponíveis

Teste de Requisitos

Exemplo de teste:

✓ 10.(iv) Quando os usuários acessarem o EDDIS, eles serão apresentados à páginas web com todos os serviços disponíveis

Requirements tested: 10.(iv)

Related requirements: 10.(i), 10.(ii), 10.(iii), 10.(vi), 10.(vii)

Test applied: For each class of user, prepare a login script and identify the services expected for that class of user.

The results of the login should be a webpage with a menu of available services.

Requirements problems: We don't know the different classes of EDDIS user and the services which are available to each user class. Apart from the administrator, are all other EDDIS users in the same class?

Recommendations: Explicitly list all user classes and the services which they can access.

PROBLEMAS NA ENGENHARIA DE REQUISITOS

- 1. Requisitos não refletirem as reais necessidades dos stakeholders

- O sistema está de acordo com o que foi especificado, mas aquilo que foi especificado não atende as necessidades dos clientes



- 2. Requisitos serem inconsistentes ou incompletos

- Passam despercebidos na análise e negociação



- 3. Alto custo da mudança de requisitos

- Quando um requisitos é mudado, todas as atividades executadas com base nesses requisitos (escrita de caso de uso, elaboração de diagramas de sequência, classe, implementação, confecção de casos de teste etc) podem estar comprometidas, e muitos artefatos necessitem de alteração



PROBLEMAS NA ENGENHARIA DE REQUISITOS

- 4. Problemas de comunicação entre stakeholders
- 5. Engenharia de requisitos não é um processo sequencial
 - Na prática as atividades podem estar ocorrendo ao mesmo tempo, em conjunto. Enquanto o especialista em requisitos está conversando, entrevistando um stakeholder (elicitação), pode estar anotando em um caderno (documentação) e mostrando para ele perguntando se realmente é aquilo (validação). Dependendo da resposta pode ser necessário chegar a um consenso (negociação).
- 6. Engenharia de requisitos utiliza muita linguagem natural, tornando-se uma área subjetiva
 - A linguagem natural é por si só ambígua e muito da documentação é baseada em linguagem natural



Problemas nos Requisitos

B.C. by Johnny Hart



PROBLEMAS NA ENGENHARIA DE REQUISITOS

- **7. Personalidade e status dos stakeholders**
 - um subordinado pode ficar constrangido e emitir opinião que vá de encontro com a do seu chefe
- **8. Mudanças sempre vão existir**
 - Propostas pelo usuário, evoluções, eventos externos (leis, regulamentos)
- **9. Deixar uma especificação perfeita pode não ser economicamente viável**
 - Custo x benefício
 - Qual é a medida? Como saber se estamos investindo exageradamente ou estamos negligenciando as atividades de requisitos?
- **10. Falta de envolvimento dos stakeholders**
 - Resistência a mudanças, Alguns trabalham com aplicações que estão apenas na sua “mente”,
- **11. Falta de gerenciamento dos requisitos**
 - mudanças nos requisitos acordados, o relacionamento entre requisitos e dependências entre a documentação de requisitos e outros documentos
- **12. Falta de definição de responsabilidades**

PROBLEMAS NA ENGENHARIA DE REQUISITOS

- 13. Baixa qualidade da documentação de requisitos
- 14. O cliente quer ver resultado cedo “forçando” a implementação antecipada
 - A saída do processo de Engenharia de requisitos na melhor das hipóteses possui um protótipo da aplicação. Pode ser que nem isso exista, mas pura e simplesmente documentação.
- 15. Usuários podem não ter uma idéia precisa do sistema por eles requerido
- 16. Usuários têm dificuldades para descreverem seu conhecimento sobre o domínio do problema
- 17. Conflitos entre requisitos não funcionais diferentes são comuns em sistemas complexos
 - Navegabilidade compromete simplicidade
- 18. Falta de investimento na Engenharia de Requisitos

PROBLEMAS NA ELICITAÇÃO

- Problemas de escopo:
 - Requisitos endereçando informações em excesso ou faltando
 - Limites do sistema mal definidos
- Problemas de entendimento:
 - Stakeholders fazem diferentes interpretações da informação
- Problemas de volatilidade:
 - Problemas referentes à natureza mutante dos requisitos

PROBLEMAS NA ELICITAÇÃO: Escopo

- *Escopo:*
 - Conteúdo do produto a ser desenvolvido ou
 - Lista dos requisitos que fazem parte do produto de software.
- Exemplo:
 - Versão inicial do software (após 1 mês de trabalho)
 - Cliente: “cadê o gráfico em pizza e a importação/exportação para Excel?”
 - ??? algo nunca antes mencionado!!!!!!!!!!
 - Ninguém JAMAIS havia mencionado gráficos em pizza ou importação/exportação de planilhas
- Formalizar:
 - Lista do que será entregue, e uma lista do que não será.
 - Define limites do que deve ser feito e uma vez acordado, pode-se estimar prazo e custo.

PROBLEMAS NA ELICITAÇÃO: Escopo

- *Gerenciamento de Escopo:*
 - Processos necessários para assegurar que o projeto contemple todo o trabalho requerido, e nada mais .
- *Etapas*
 - Iniciação:
 - Autorização do projeto
 - Planejamento
 - Desenvolver a declaração do escopo
 - Base para decisões de projeto
 - Detalhamento
 - Dividir o projeto em módulos menores mais gerenciáveis
 - Verificação
 - Formalizar a aprovação do escopo
 - Controle de Mudanças
 - Documentar e controlar mudanças no escopo

PROBLEMAS NA ELICITAÇÃO: Entendimento

- Usuários não sabem das suas necessidades
- Usuários têm pouco entendimento dos recursos e limitações computacionais
- Analistas têm pouco conhecimento do domínio do problema
 - Usuários e analistas falam “línguas” diferentes
- Facilidade de omitir informações **“óbvias”**
- Usuários com visões conflitantes
- Requisitos vagos e não “testáveis”: “... tem que ser amigável”, “robustez”,

PROBLEMAS NA ELICITAÇÃO: Entendimento

- “**good communication** among users, **customers**, and **developers** is very important in solving pragmatic system problems, an issue too often overlooked when system analysis is approached only from a computer-science standpoint” [Michael Deutsch]

BOAS PRÁTICAS

- 1. Trabalhar com estruturas padronizadas nos artefatos de requisitos
 - Uso de templates
- 2. Identificar de forma única cada requisito
- 3. Definir políticas para o gerenciamento de requisitos
- 4. Usar checklists durante a análise de requisitos
- 5. Usar cenários para elicitar requisitos
 - Casos de Uso
- 6. Usar prototipagem
 - Protótipo ajuda na elicitação e validação
- 7. Reusar requisitos

DOCUMENTAÇÃO DE REQUISITOS

- Agora iremos explorar a execução de algumas atividade para confeccionar a documentação de requisitos

1. Desenvolver Visão
2. Capturar Vocabulário Comum
3. Localizar Atores e Casos de Uso
4. Estruturar Modelo de Caso de Uso

DOCUMENTO DE VISÃO

Objetivo: entender a dinâmica da atividade Desenvolver Visão para obtermos o Documento de Visão

Atividades de documentação de requisitos:

1. Desenvolver Visão

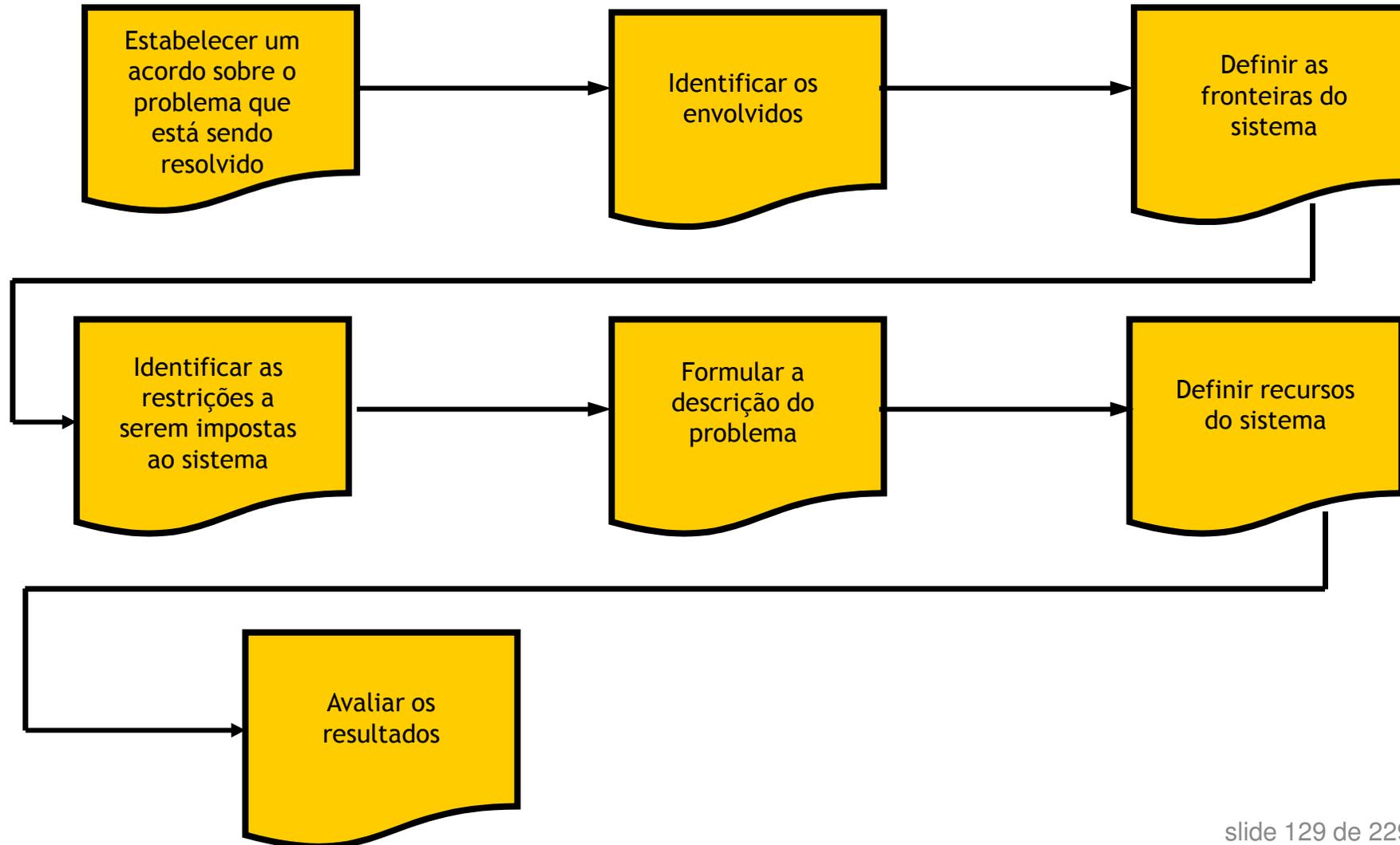
2. Capturar Vocabulário Comum
3. Localizar Atores e Casos de uso
4. Estruturar Modelo de Caso de uso



onde é definido a visão que os stakeholders têm do produto a ser desenvolvido, em termos das necessidades e características mais importantes.

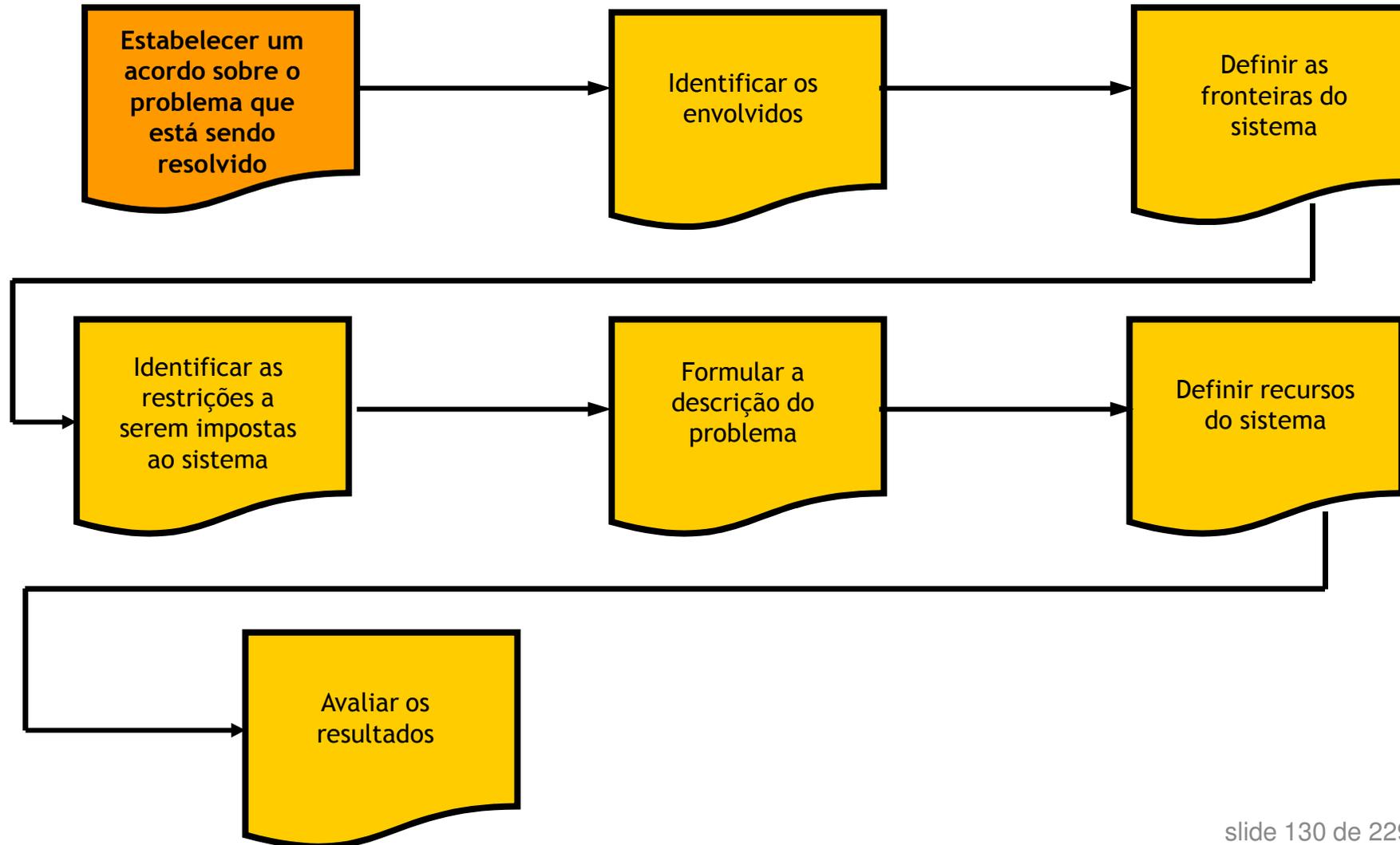
DOCUMENTO DE VISÃO

- Passos da atividade Desenvolver Visão

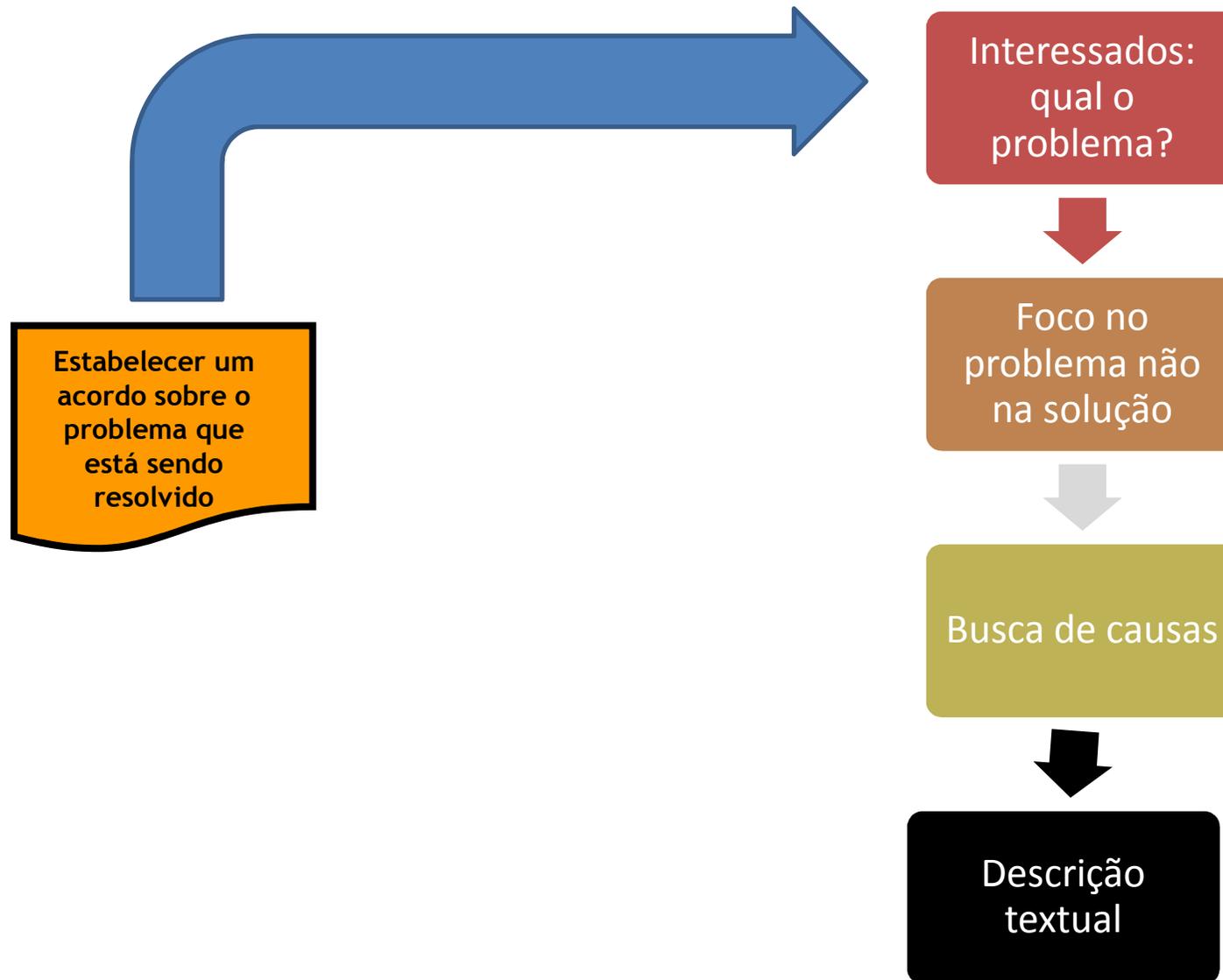


DOCUMENTO DE VISÃO

- Passos da atividade Desenvolver Visão

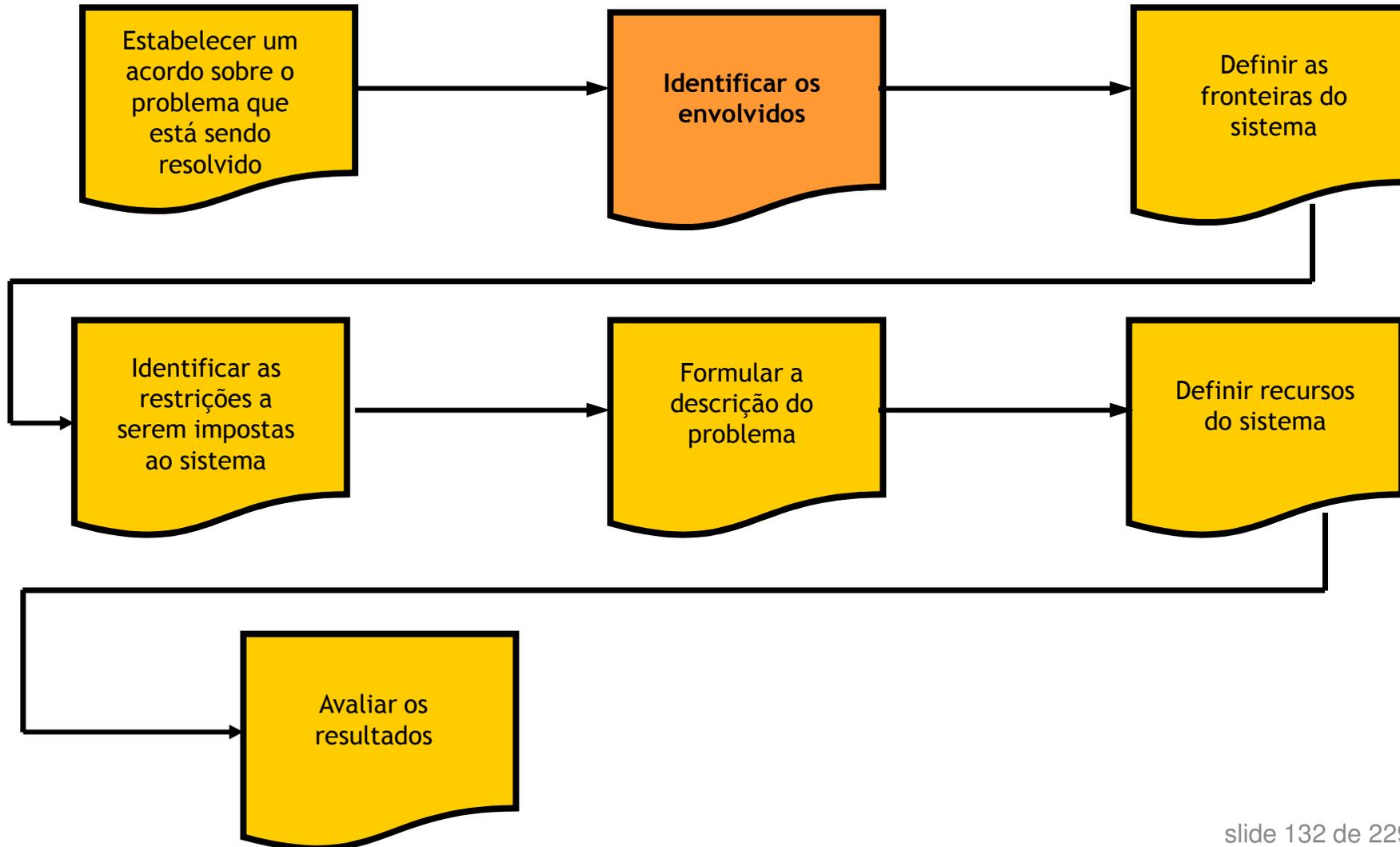


DOCUMENTO DE VISÃO

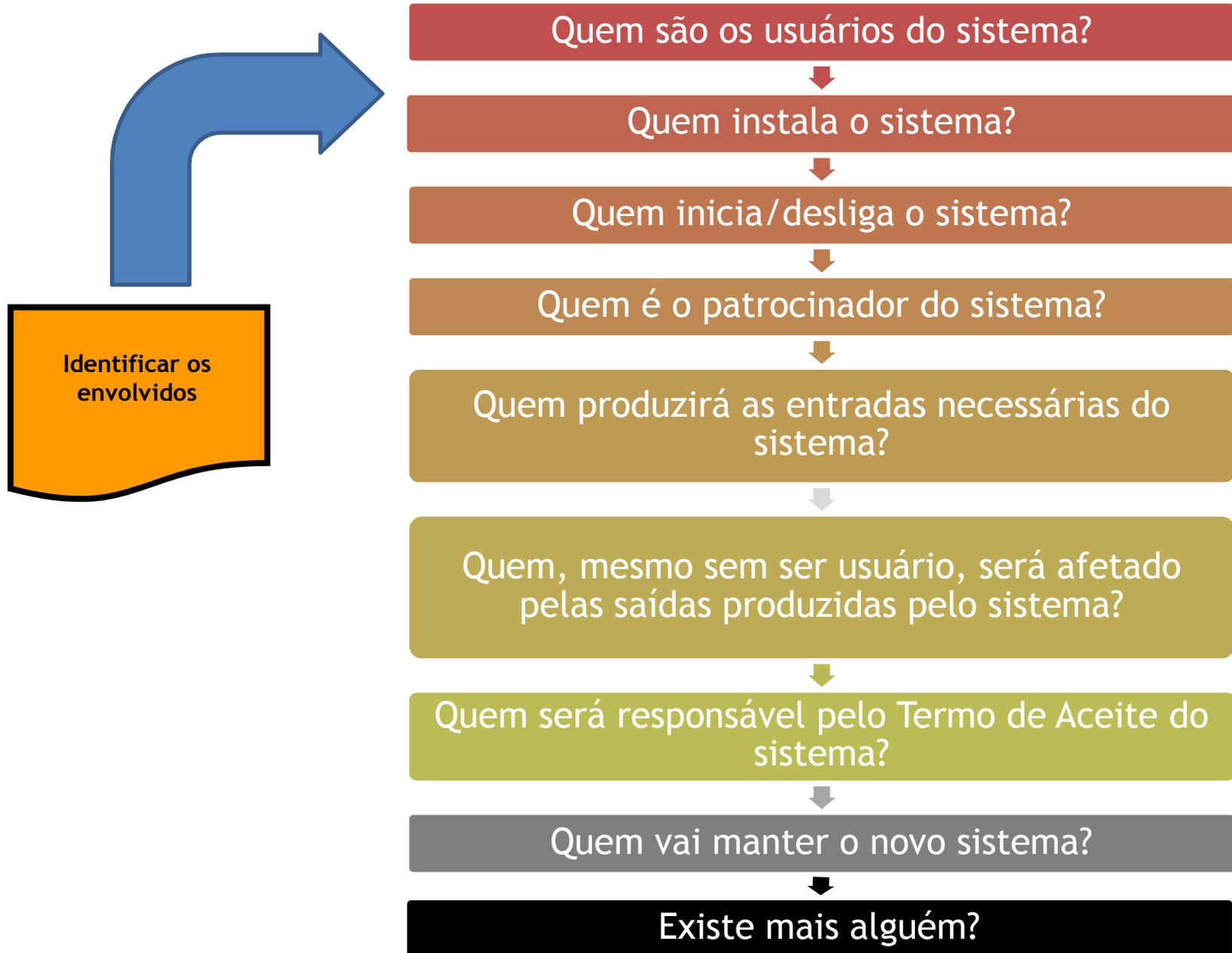


DOCUMENTO DE VISÃO

- Passos da atividade Desenvolver Visão

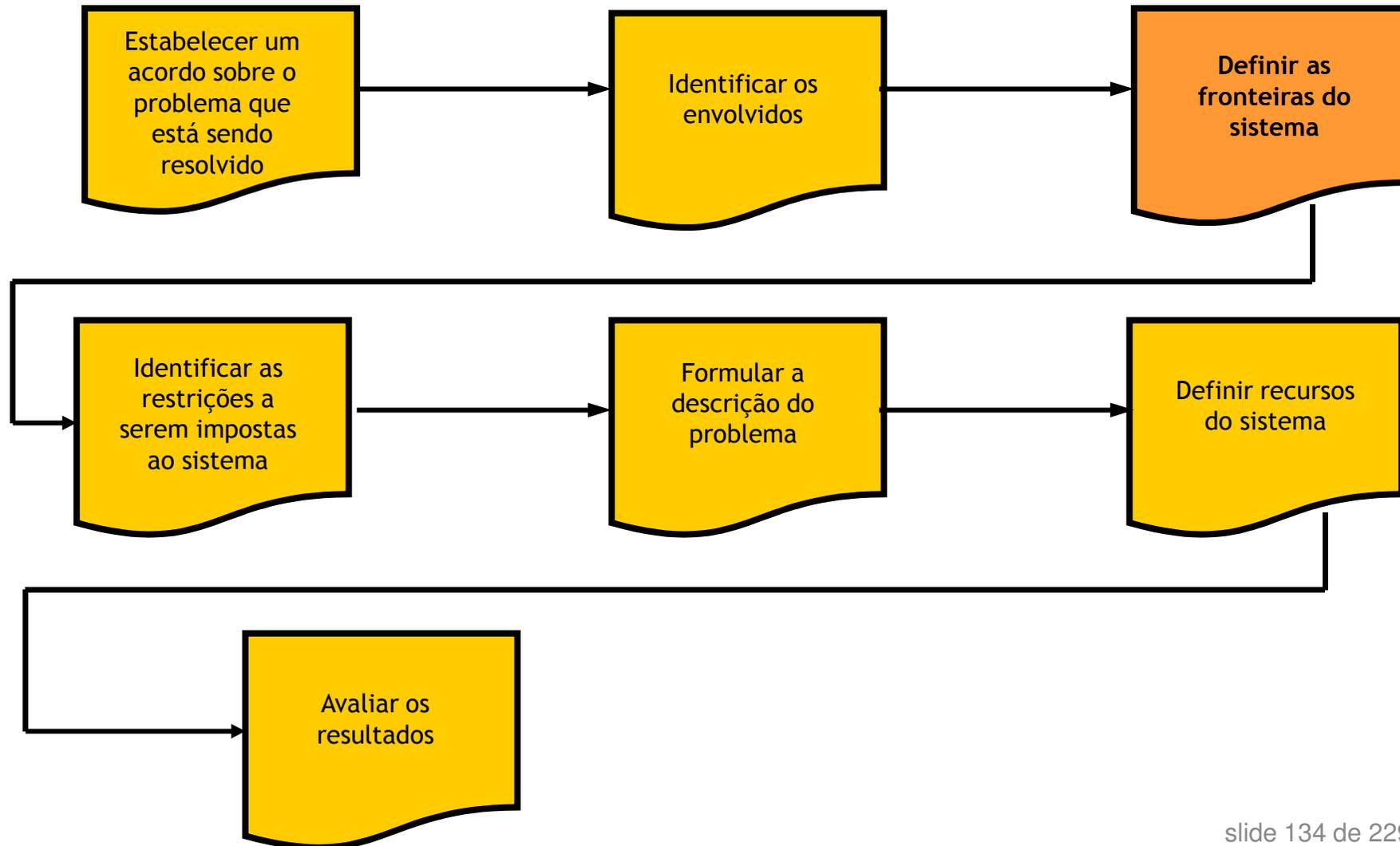


DOCUMENTO DE VISAO



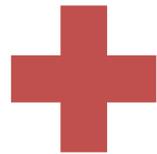
DOCUMENTO DE VISÃO

- Passos da atividade Desenvolver Visão



DOCUMENTO DE VISÃO

Tudo o que está fora da fronteira da aplicação não faz parte dela, mas pode interagir com ela



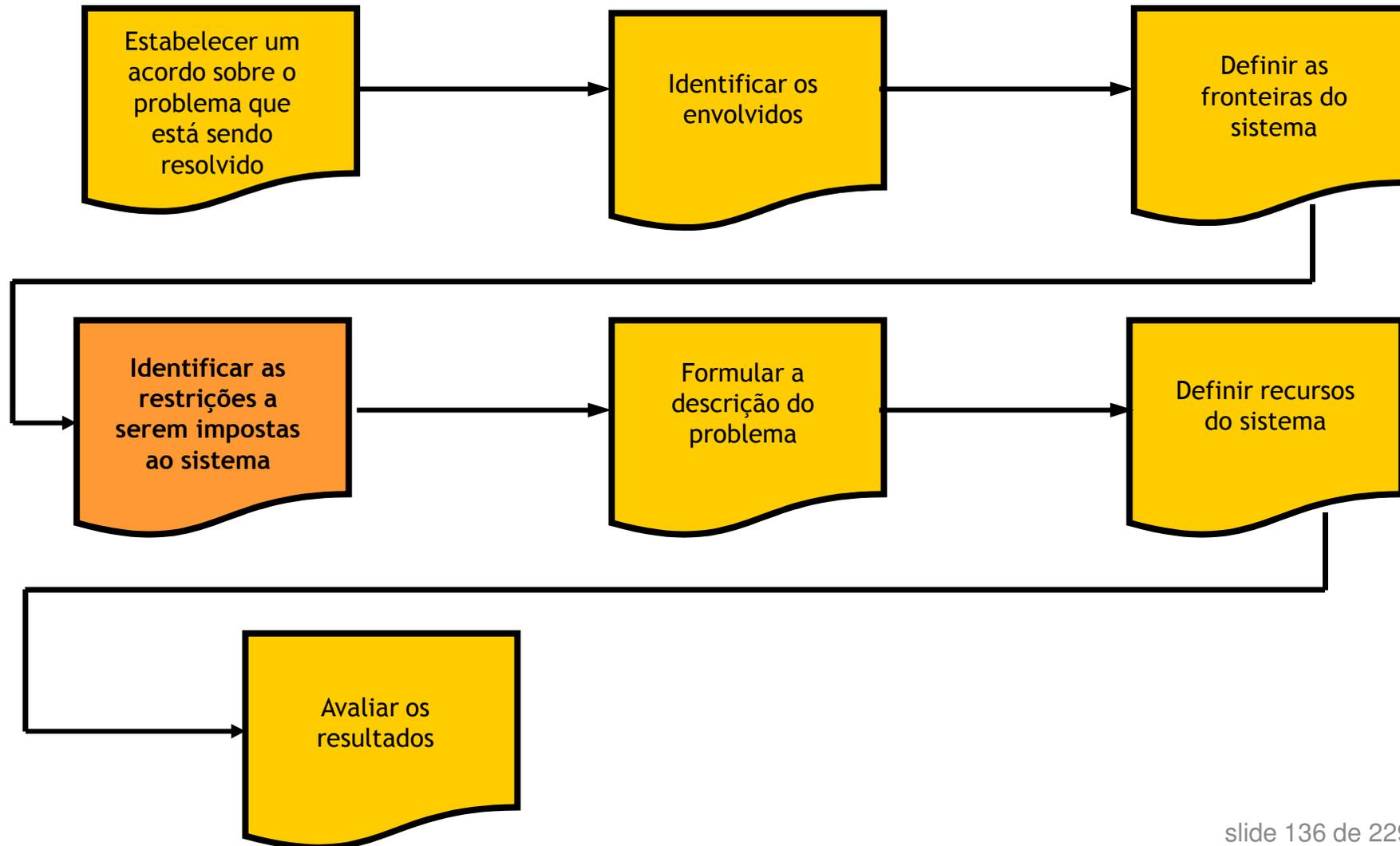
A fronteira da aplicação define o limite entre o sistema e o mundo real que o cerca



Definir as fronteiras do sistema

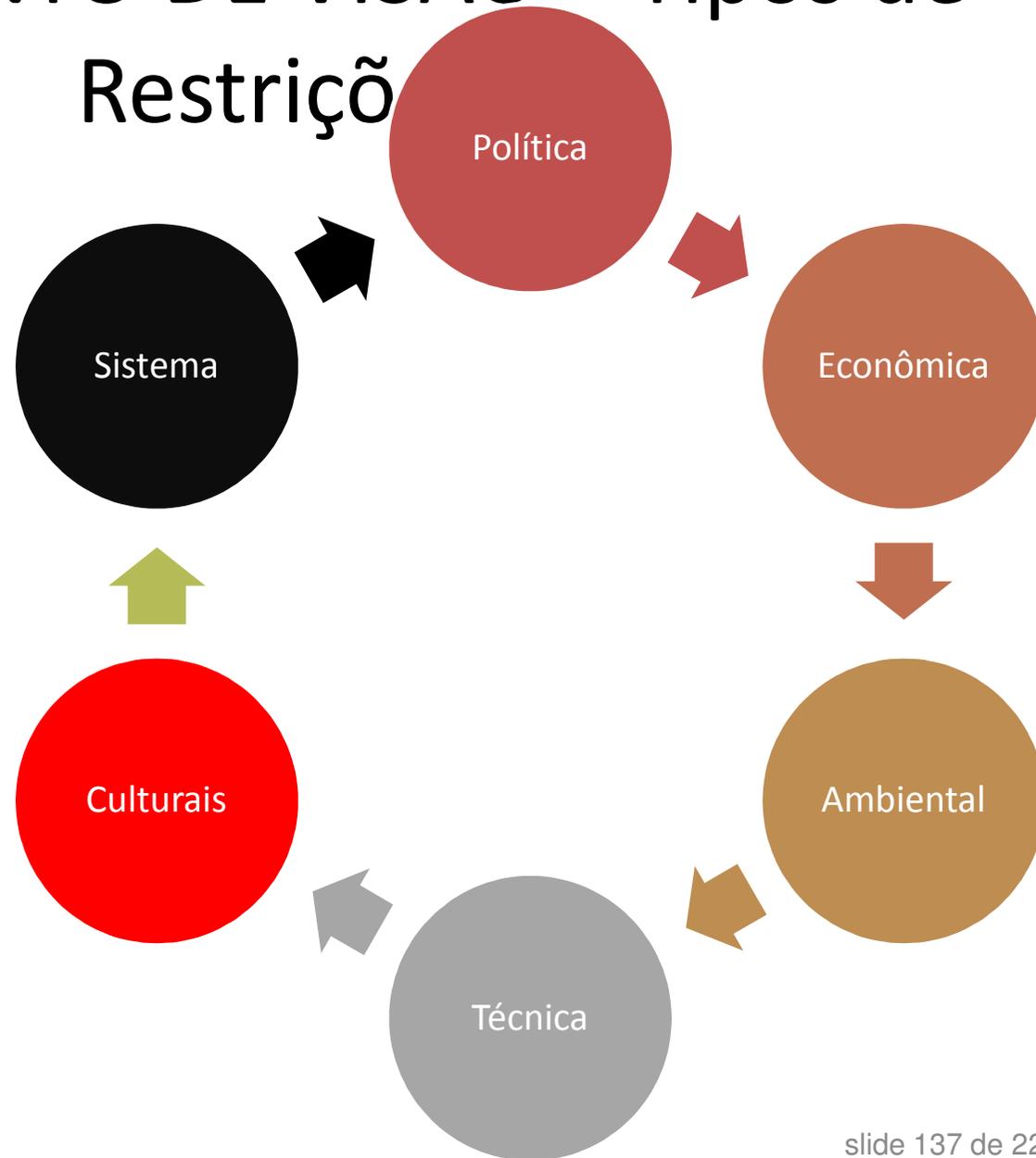
DOCUMENTO DE VISÃO

- Passos da atividade Desenvolver Visão



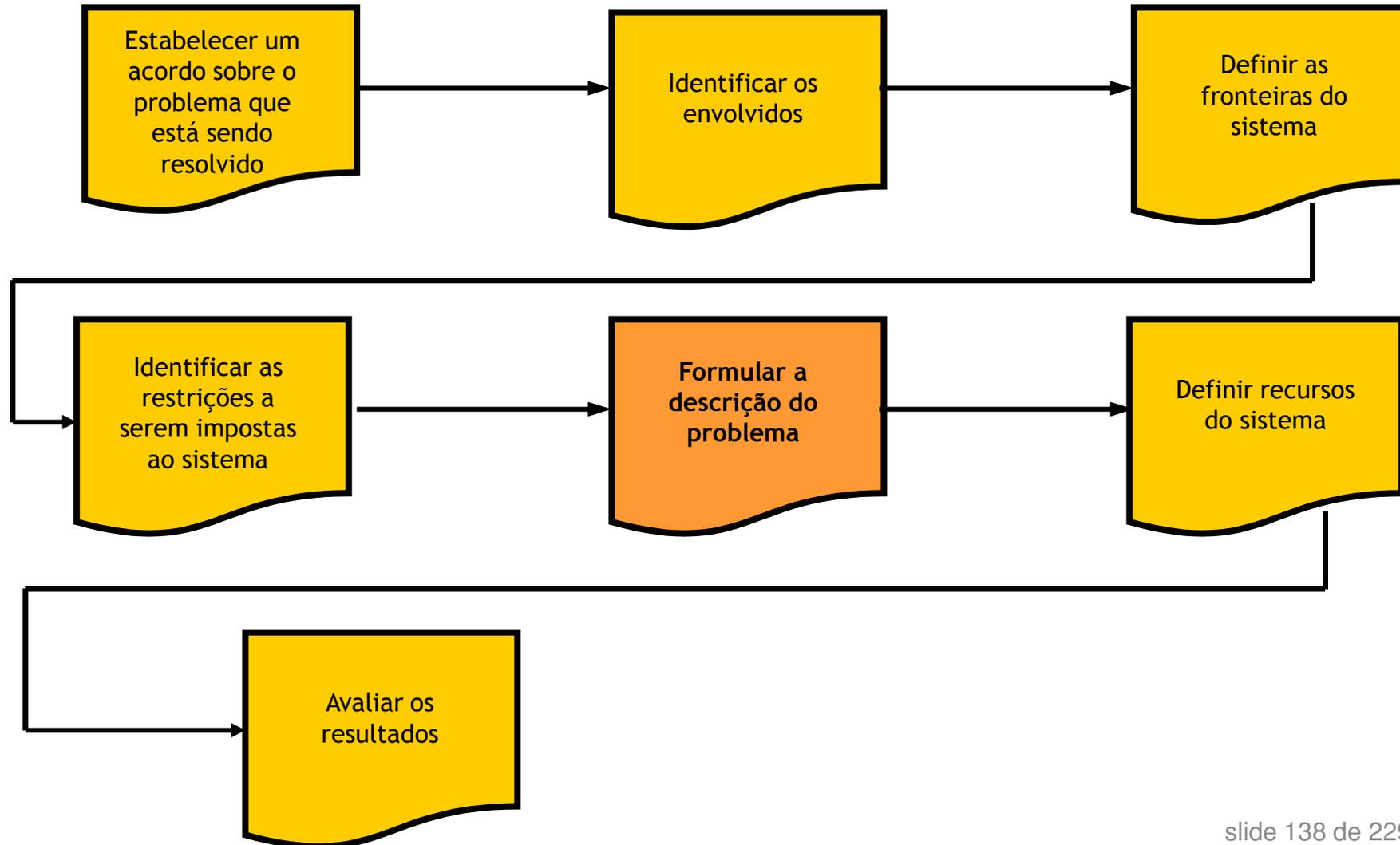
DOCUMENTO DE VISÃO – Tipos de Restrição

Essas restrições irão limitar as possibilidades de solução e precisar ser levadas em consideração desde a concepção do projeto



DOCUMENTO DE VISÃO

- Passos da atividade Desenvolver Visão



DOCUMENTO DE VISÃO

- ✓ A motivação para o desenvolvimento de um sistema é atender as necessidades dos usuários, resolvendo um ou mais problemas
- ✓ **Formulação do problema**
 - **Entender e descrever** qual o problema irá ser resolvido
 - Listar os **afetados** por esse problema, e também o impacto
 - Listar os **benefícios** dos sistema.
- ✓ **Exemplos:**
 - O problema de resolução inoportuna e incorreta de queixas sobre atendimento ao cliente
 - Afeta clientes, a equipe de suporte e os técnicos
 - Como impacto existe a insatisfação do cliente, percepção de falta de qualidade, funcionários insatisfeitos e perda de receita.
 - Uma solução bem-sucedida poderia conceder à equipe de suporte técnico acesso em tempo real a um banco de dados de resolução de problemas e facilitar a designação de técnicos de serviço, de forma oportuna, somente para os locais que realmente precisem de assistência.

Decomposição

Decompor o problema até:

- Cada subproblema esteja no mesmo nível de detalhe
- Cada subproblema possa ser resolvido de modo independente
- As soluções de cada subproblema possam ser combinadas de modo a resolver o problema original

- **Vantagens:**

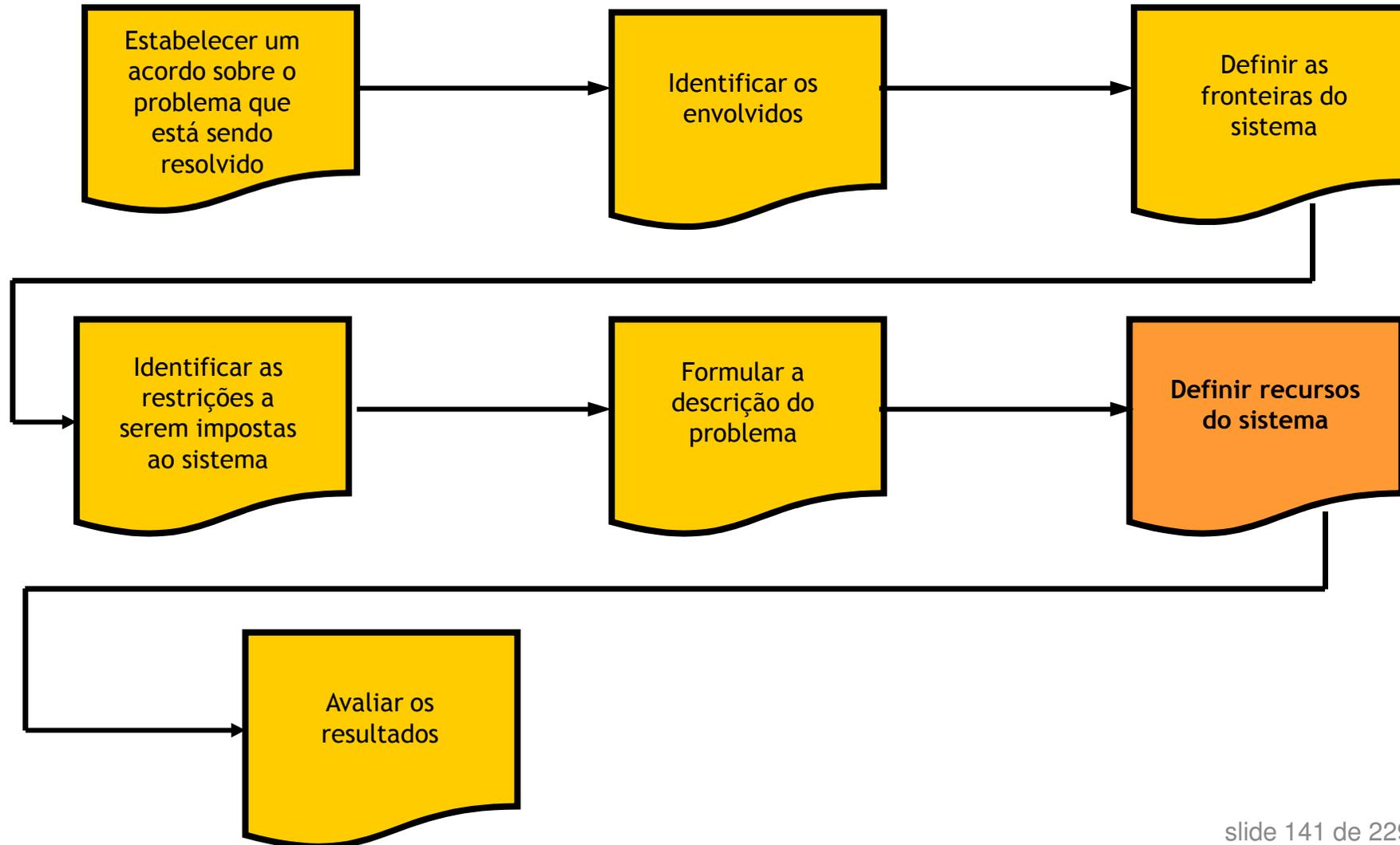
- Pessoas diferentes podem trabalhar nos subproblemas
- Paralelização pode ser possível
- Manutenção é mais fácil

- **Desvantagens**

- As soluções dos subproblemas podem não combinar de modo a resolver o problema original
- Problemas de difícil compreensão são difíceis de decompor
- A estrutura do mundo real NÃO é hierárquica

DOCUMENTO DE VISÃO

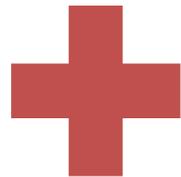
- Passos da atividade Desenvolver Visão



DOCUMENTO DE VISÃO

Definir recursos
do sistema

Descrição
do
problema



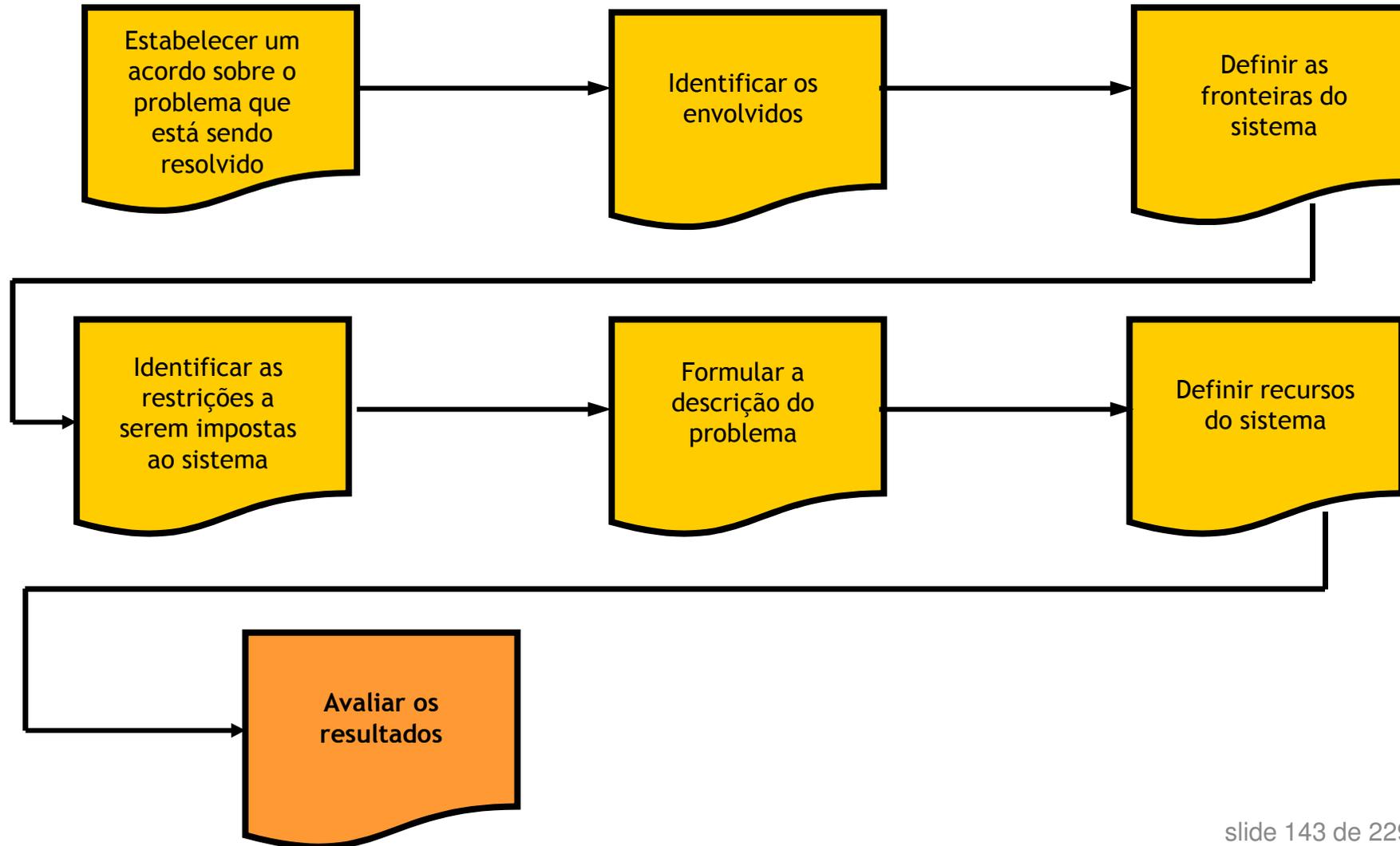
Benefícios
da Solução



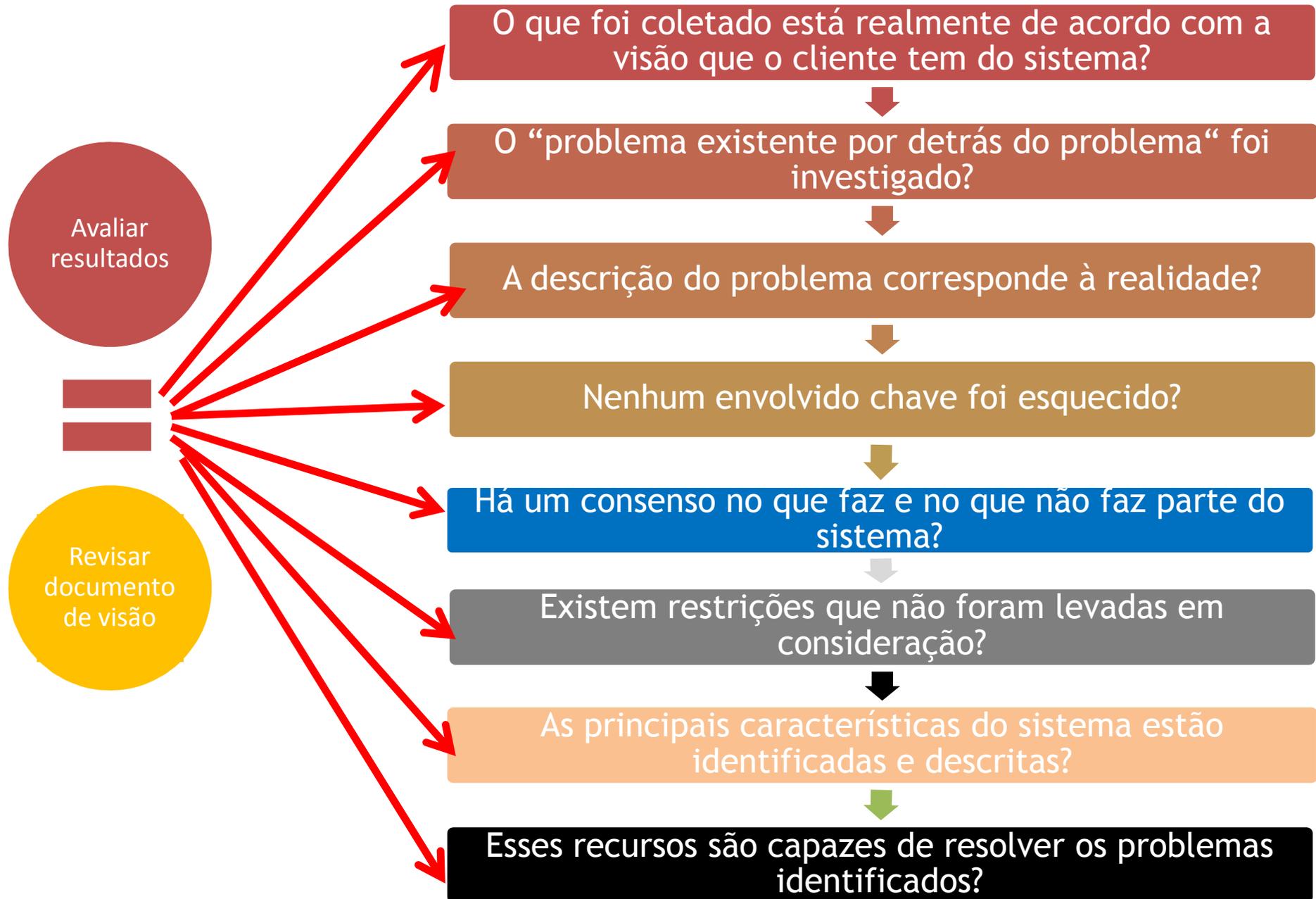
Recursos
necessários
para o
sistema

DOCUMENTO DE VISÃO

- Passos da atividade Desenvolver Visão



DOCUMENTO DE VISÃO



Documento de Visão - Conteúdo

Escopo:

- ✓ Descreve aspectos e funções que devem fazer parte do produto

Gestores:

- ✓ Quem vai gerenciar o documento e as funcionalidades que emergem a partir desse gestor

Levantamento de Necessidades:

- ✓ Formulação do problema
- ✓ Listar as necessidades categorizadas por prioridades

Documento de Visão - Conteúdo

Funcionalidades do Produto:

- ✓ Características funcionais levantadas com o objetivo de satisfazer as necessidades identificadas anteriormente
- ✓ As funcionalidades devem estar ordenadas por prioridade, conforme os critérios do próprio usuário
- ✓ A granularidade destas funcionalidades é maior que a de um caso de uso

Documento de Visão - Conteúdo

Interligação com outros sistemas:

- ✓ Descreve de forma simples, ou através de um diagrama, outros sistemas com os quais este sistema se relacione

Restrições:

- ✓ Requisitos, técnicos ou não, sem os quais o sistema será inviável, ou que limitem as alternativas de solução

DOCUMENTO DE VISÃO

1. Exercício: Em grupos formule a descrição de um problema de duas aplicações: o projeto da fábrica (ou estudo de caso!) e o exercício do sistema de matrícula (moodle)

Problema	<i>[Descrever o problema]</i>
Pessoas Atingidas	<i>[Listar os stakeholders afetados pelo problema]</i>
Impacto	<i>[Descrever os impactos do problema]</i>
Uma solução bem sucedida traria	<i>[Listar os principais benefícios trazidos por uma solução bem sucedida]</i>

GLOSSÁRIO

Objetivo:

- Entender a dinâmica da atividade *Capturar Vocabulário Comum* para obtermos o Glossário

Atividades de documentação de requisitos:

- ✓ Desenvolver Visão

2. Capturar Vocabulário Comum

3. Localizar Atores e Casos de Uso

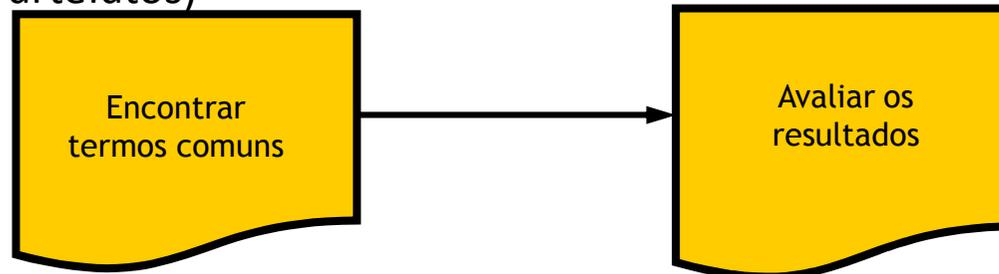
4. Estruturar Modelo de Caso de Uso

GLOSSÁRIO

- Passos da atividade Capturar Vocabulário

Comum:

- Definir um vocabulário comum que possa ser usado em todas as descrições textuais do sistema (documento de visão, nas especificações de casos de uso, ou outros artefatos)



GLOSSÁRIO

Encontrar
termos
comuns

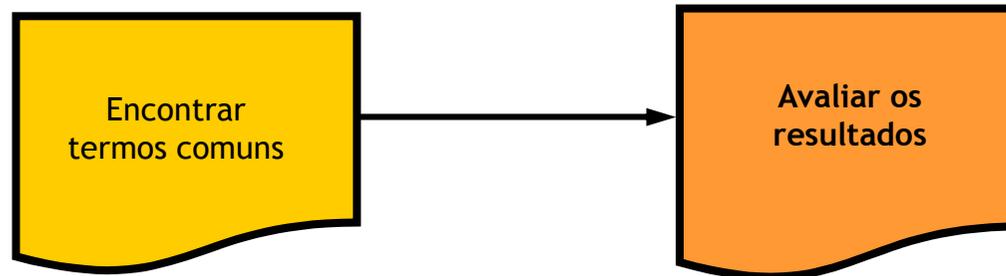
- Elaborar um documento que concentra a definição de termos comumente utilizada

- Um vocabulário comum que utilize os termos e as expressões mais frequentes do domínio de problemas serve como referência para o entendimento de outros documentos.

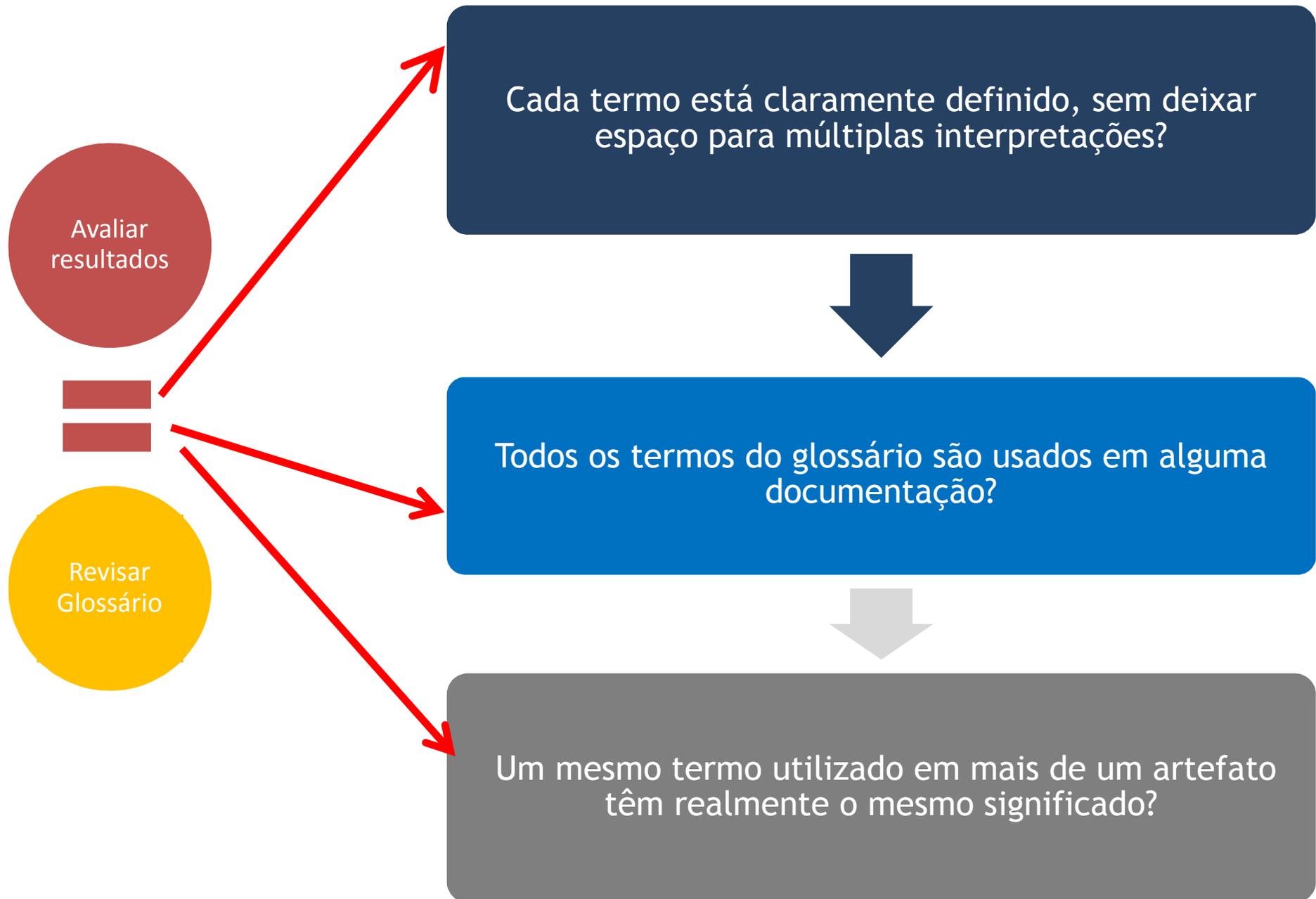
- Deve ser mais voltado para os termos do negócio

GLOSSÁRIO

- Passos da atividade Capturar Vocabulário Comum



GLOSSÁRIO



DOCUMENTO DE VISÃO

1. Exercício: Em grupos, elaborar o glossário de duas aplicações: o projeto da fábrica (ou estudo de caso!) e o exercício do sistema de matrícula (moodle)

Termo	Descrição

MODELO DE CASO DE USO – ATORES E CASOS DE USO

Objetivo:

- Entender a dinâmica da atividade *Localizar Atores e Casos de Uso* para iniciarmos a obtenção do Modelo de Caso de Uso

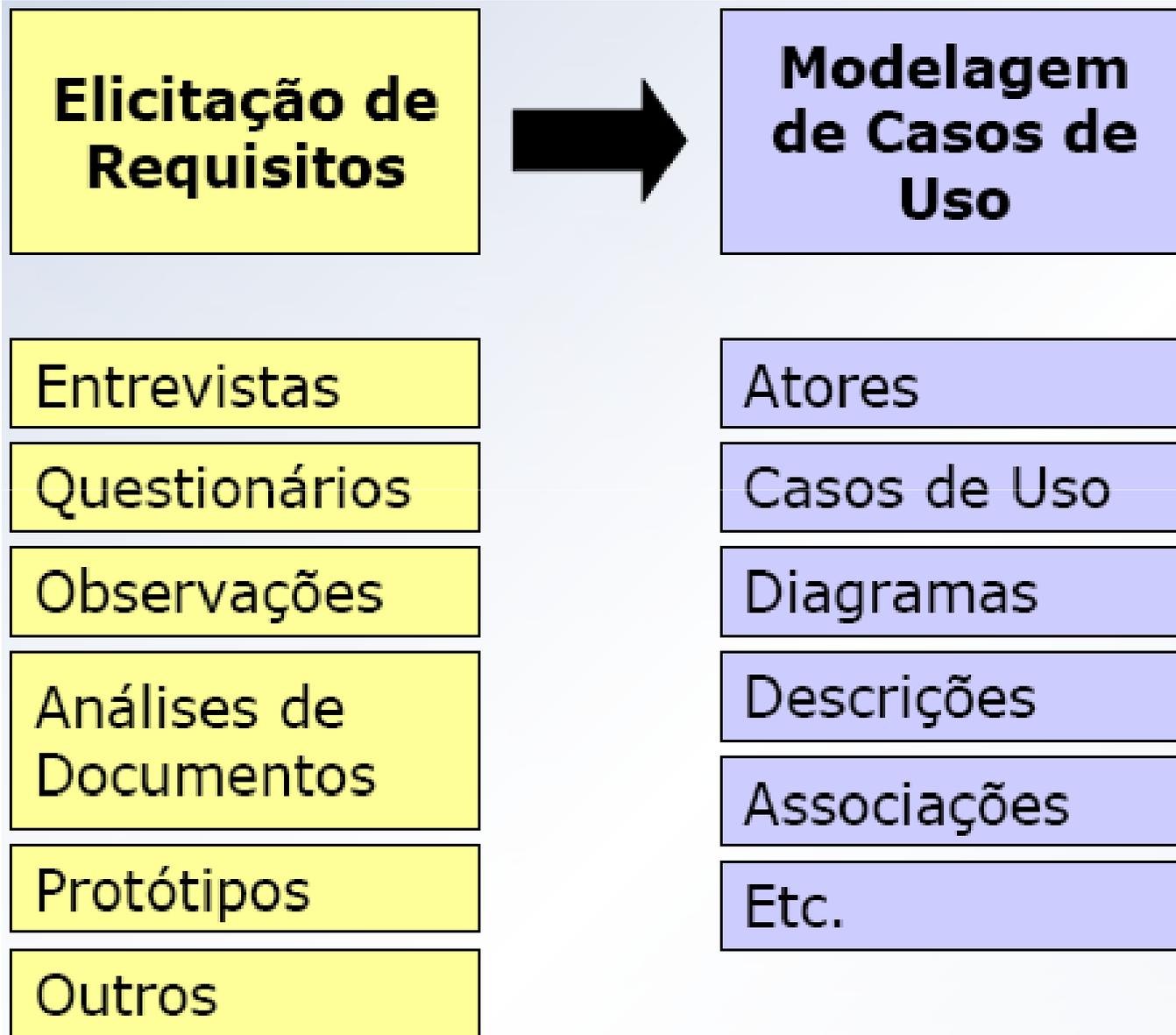
Atividades de documentação de requisitos:

- ✓ Desenvolver Visão
- ✓ Capturar Vocabulário Comum

3. Localizar Atores e Casos de Uso

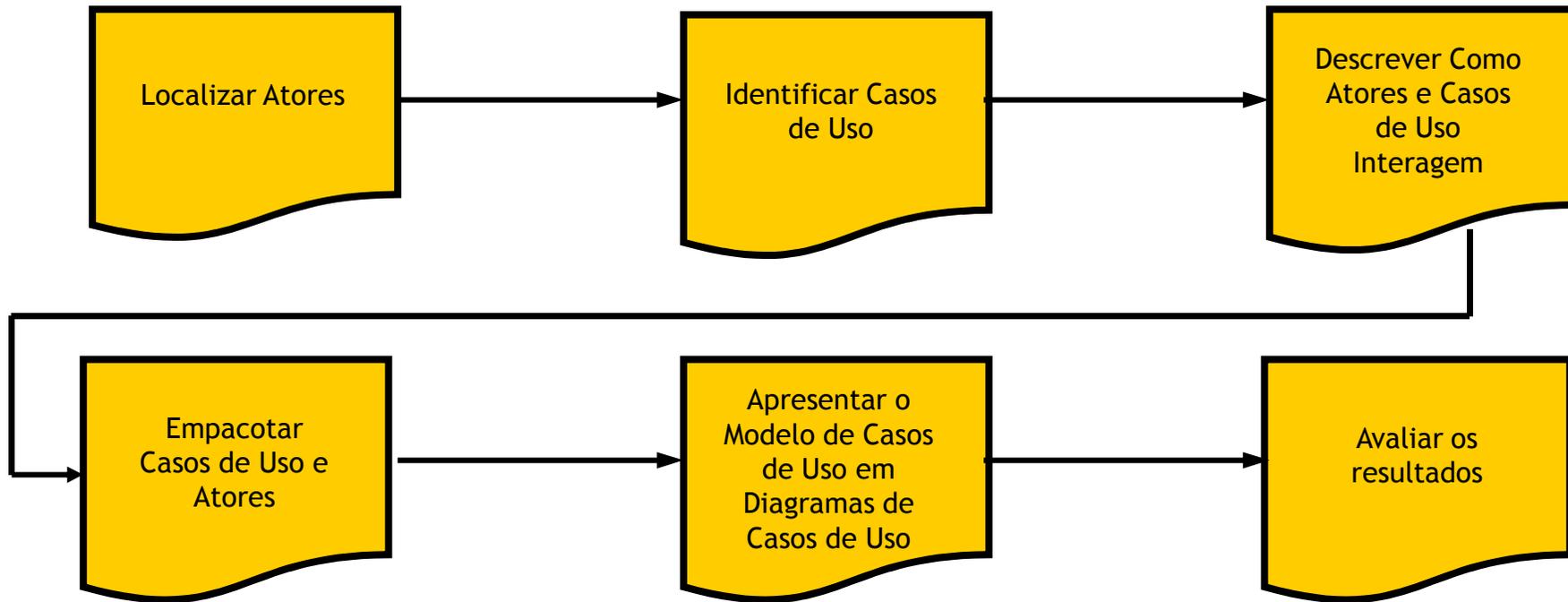
4. Estruturar Modelo de Caso de Uso

Atores e Casos de Uso



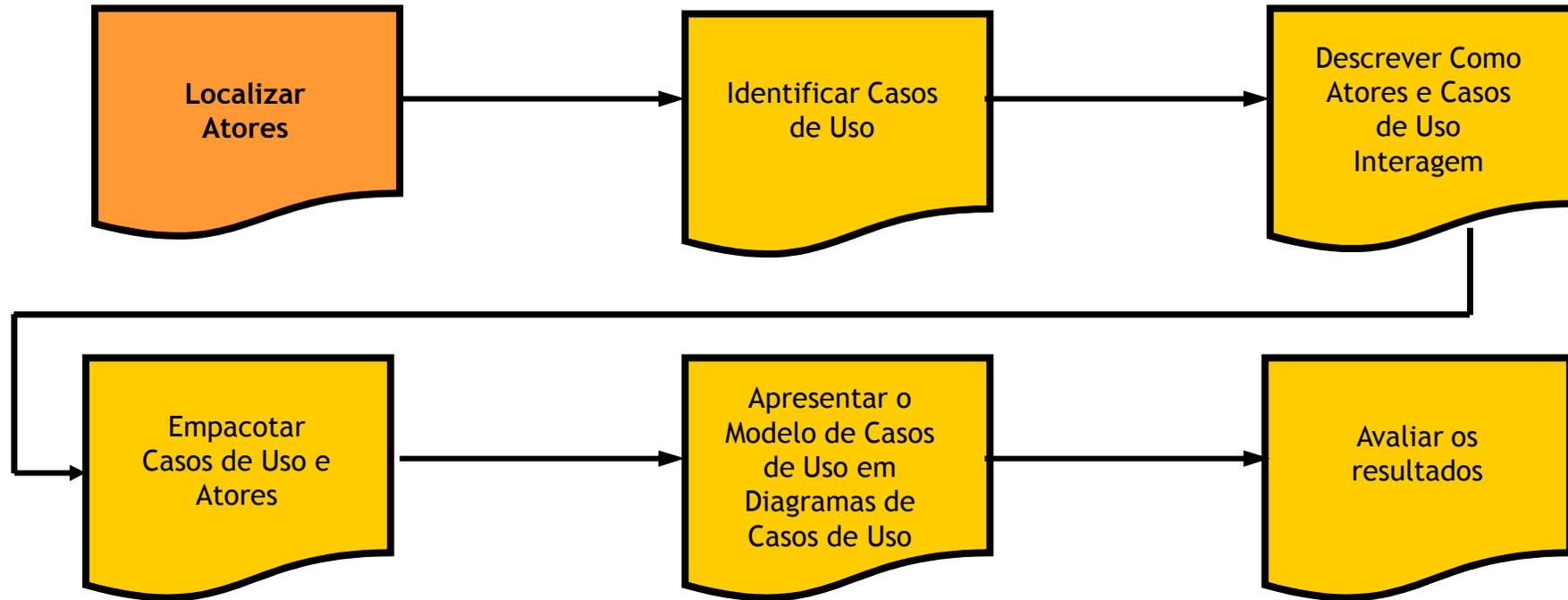
MODELO DE CASO DE USO – ATORES E CASOS DE USO

- Passos da atividade Localizar Atores e Casos de Uso

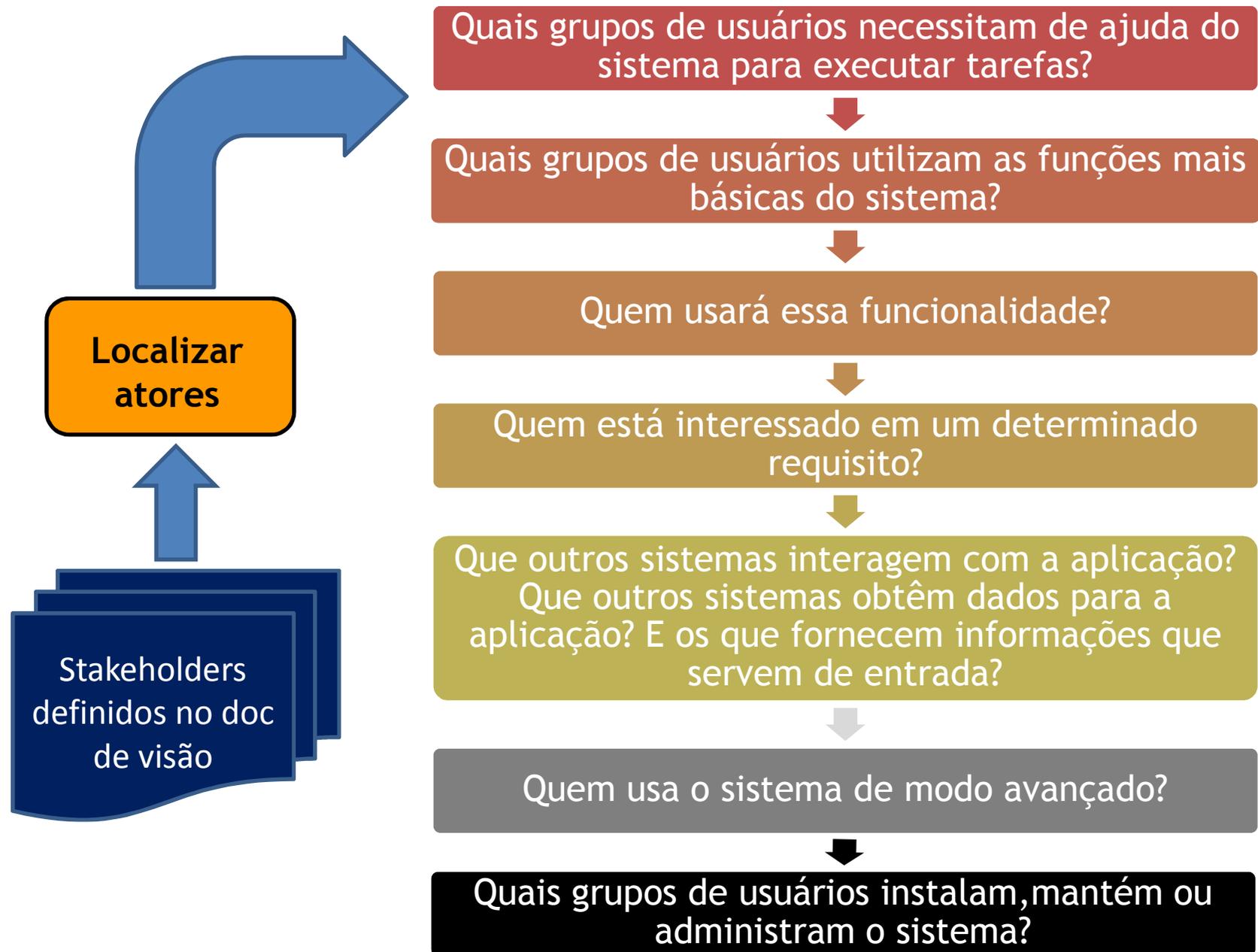


MODELO DE CASO DE USO – ATORES E CASOS DE USO DE USO

- Passos da atividade Localizar Atores e Casos de Uso



MODELO DE CASO DE USO – ATORES E CASOS DE USO



MODELO DE CASO DE USO – ATORES E CASOS DE USO

- ✓ Qualquer desses é um ator **candidato**
- ✓ Uma instância de ator é alguém ou algo **externo** ao sistema que interage com ele
- ✓ Uma classe de ator define um conjunto de instâncias de ator, no qual cada uma desempenha o mesmo papel em relação ao sistema
 - ✓ Exemplo: gerentes de vendas acessando um mesmo relatório gerencial
- ✓ Normalmente um **ator** inicia a seqüência de interações com o sistema

Categorias de Atores

- **Pessoas**

- Empregado, Cliente, Gerente, Almoxarife, Vendedor

- **Organizações**

- Empresa Fornecedora, Agência de Impostos, Administradora de Cartões

- **Outros sistemas**

- Sistema de Cobrança, Sistema de Estoque de Produtos

- **Equipamentos**

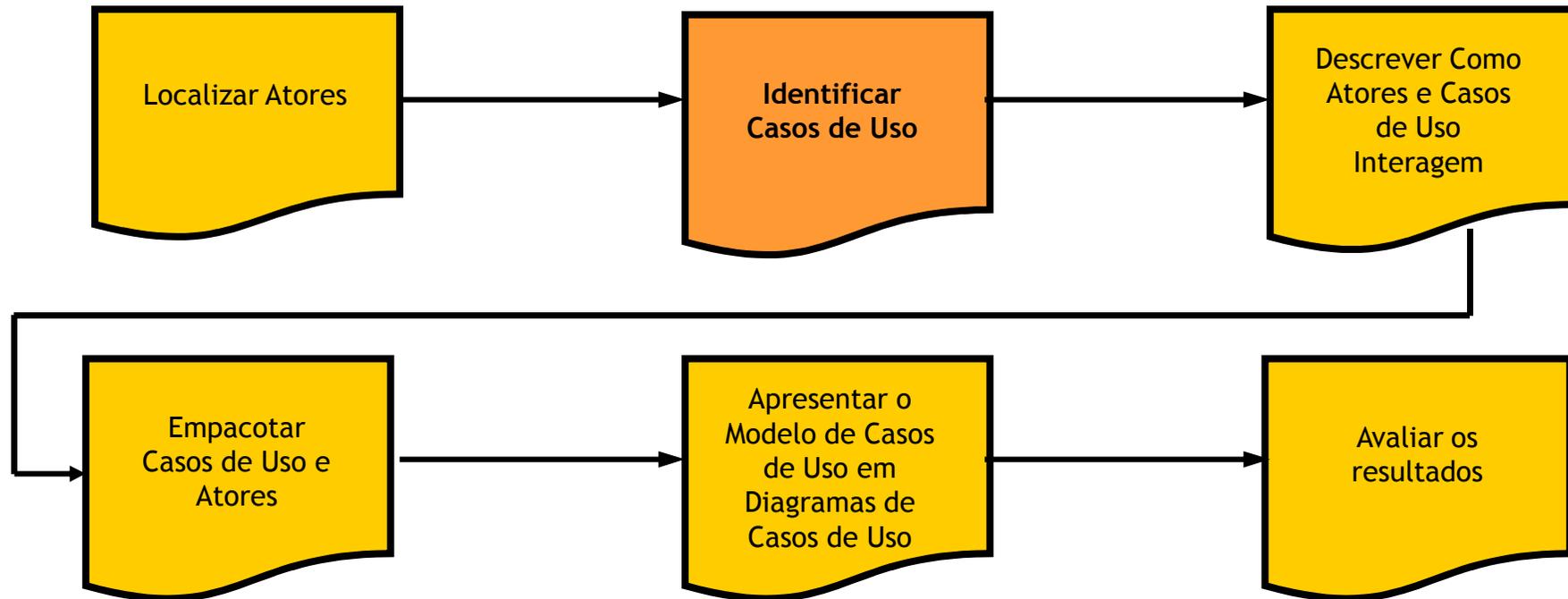
- Leitora de Código de Barras, Sensor

MODELO DE CASO DE USO – ATORES

- Sistema Bancário
 - Cliente, gerente, caixa, diretores...
- Hospital
 - Paciente, atendentes, profissionais de saúde, gerência,...
- Caixa eletrônico
 - Cliente, Banco,
- Compras, vendas e estoque
 - Comprador, fornecedor, almoxarifado, vendedor, cliente, ...

MODELO DE CASO DE USO – ATORES E CASOS DE USO

- Passos da atividade Localizar Atores e Casos de Uso



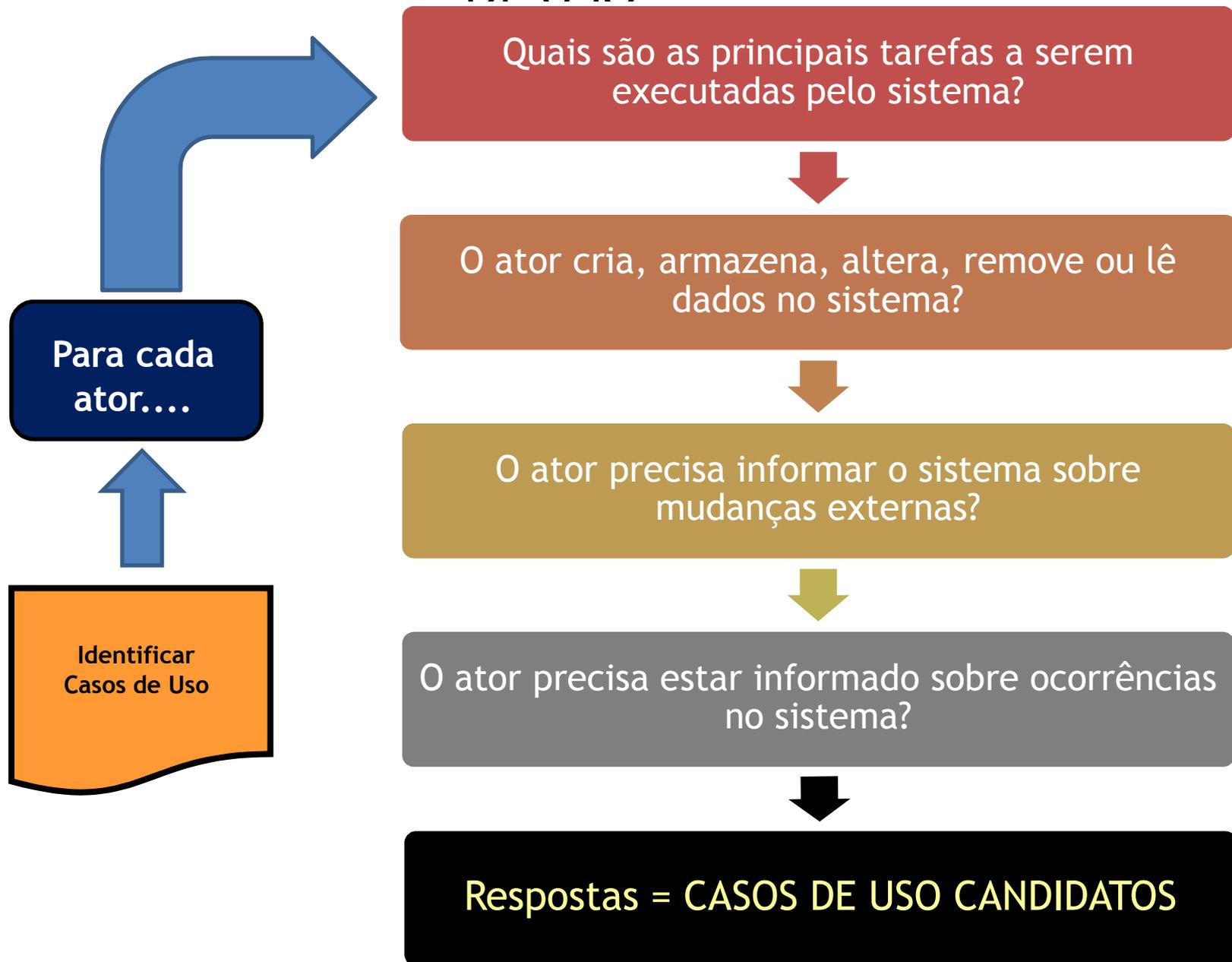
MODELO DE CASO DE USO – ATORES E CASOS DE USO

→ Um caso de uso é uma **seqüência de ações** realizada por um sistema que produz um **resultado** de valor observável para determinado ator

→ Todos os casos de uso juntos devem descrever a **funcionalidade** completa do sistema (requisitos)

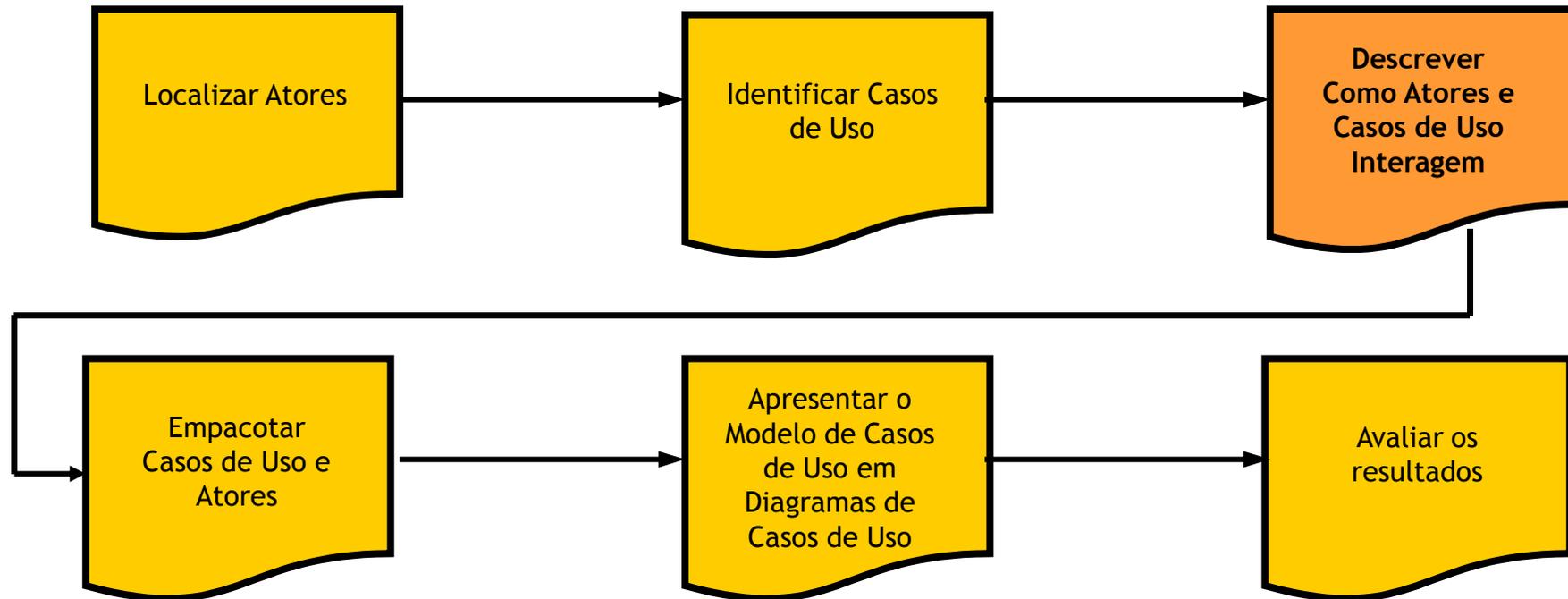
→ **Atores** são fundamentais para a descoberta dos casos de uso

MODELO DE CASO DE USO – ATORES E CASOS DE USO



MODELO DE CASO DE USO – ATORES E CASOS DE USO

- Passos da atividade Localizar Atores e Casos de Uso



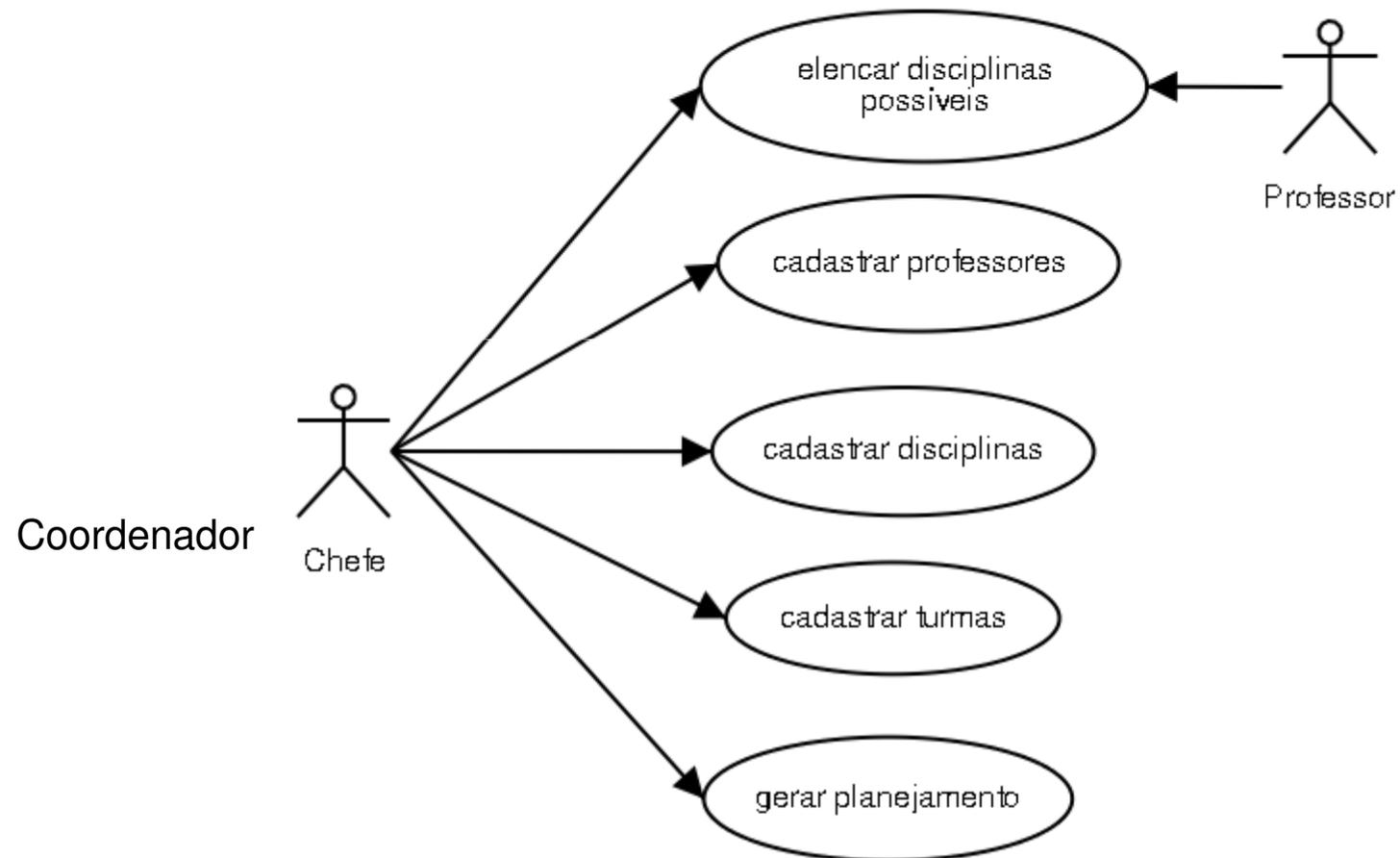
MODELO DE CASO DE USO – ATORES E CASOS DE USO

- **Criar interações**: passo de modelagem dos caso de uso
 - ✓ Diagrama de casos de uso
- Uma **associação de comunicação** é uma associação entre uma classe de **ator** e uma classe de **caso de uso**, que indica haver interação entre suas instâncias
- Um ator se **comunica** com os casos de uso por vários motivos, por exemplo:
 - ✓ Para iniciar um caso de uso
 - ✓ Para solicitar dados do sistema
 - ✓ Para alterar os dados armazenados no sistema

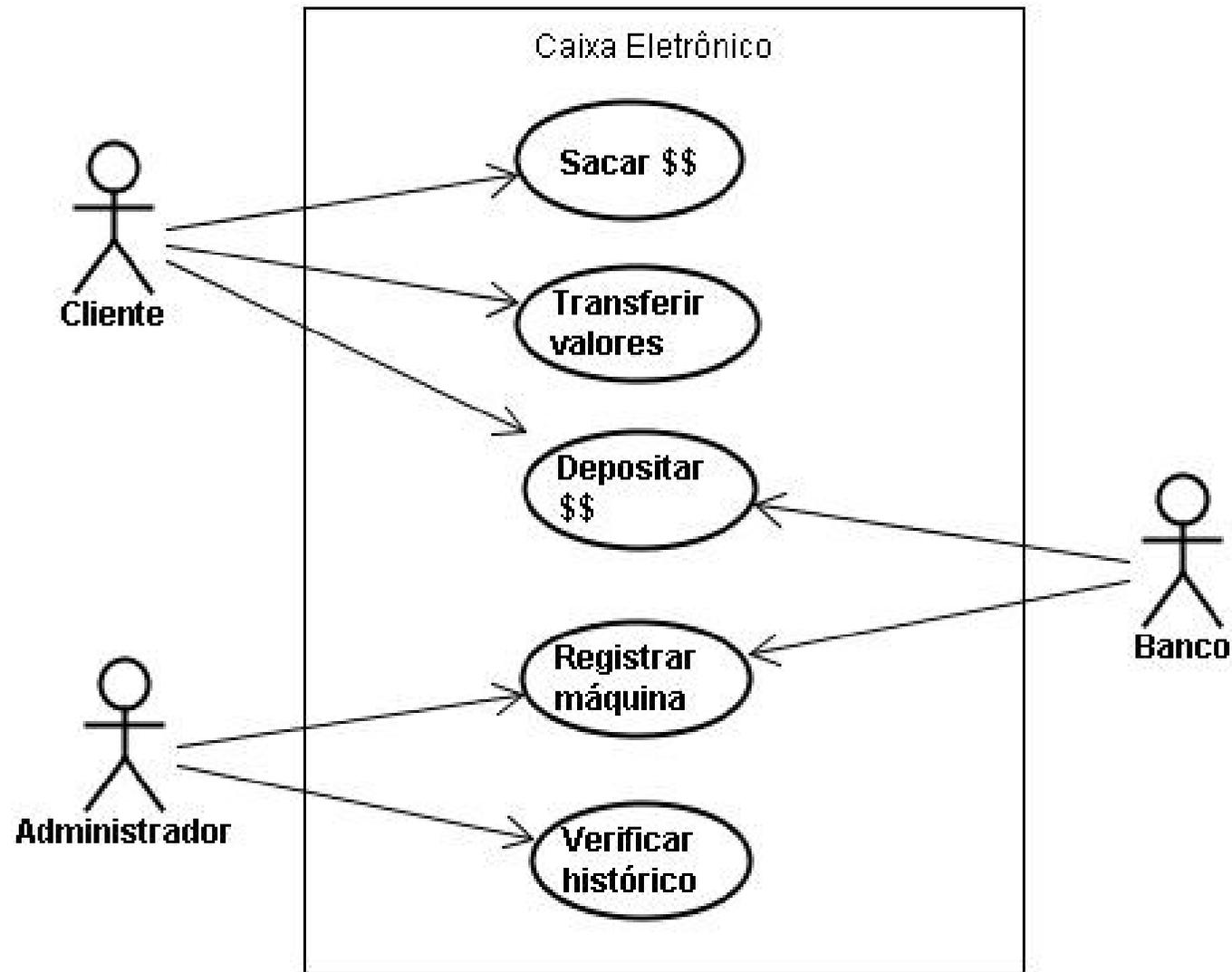
Comunicação

- Representa a informação de **quais atores** estão associados a **quais casos de uso**
- O fato de um ator estar associado a um caso de uso significa que esse **ator interage** (troca informações) **com o sistema**
- Um ator pode se relacionar com **mais de um** caso de uso
- É o mais comum dos relacionamentos

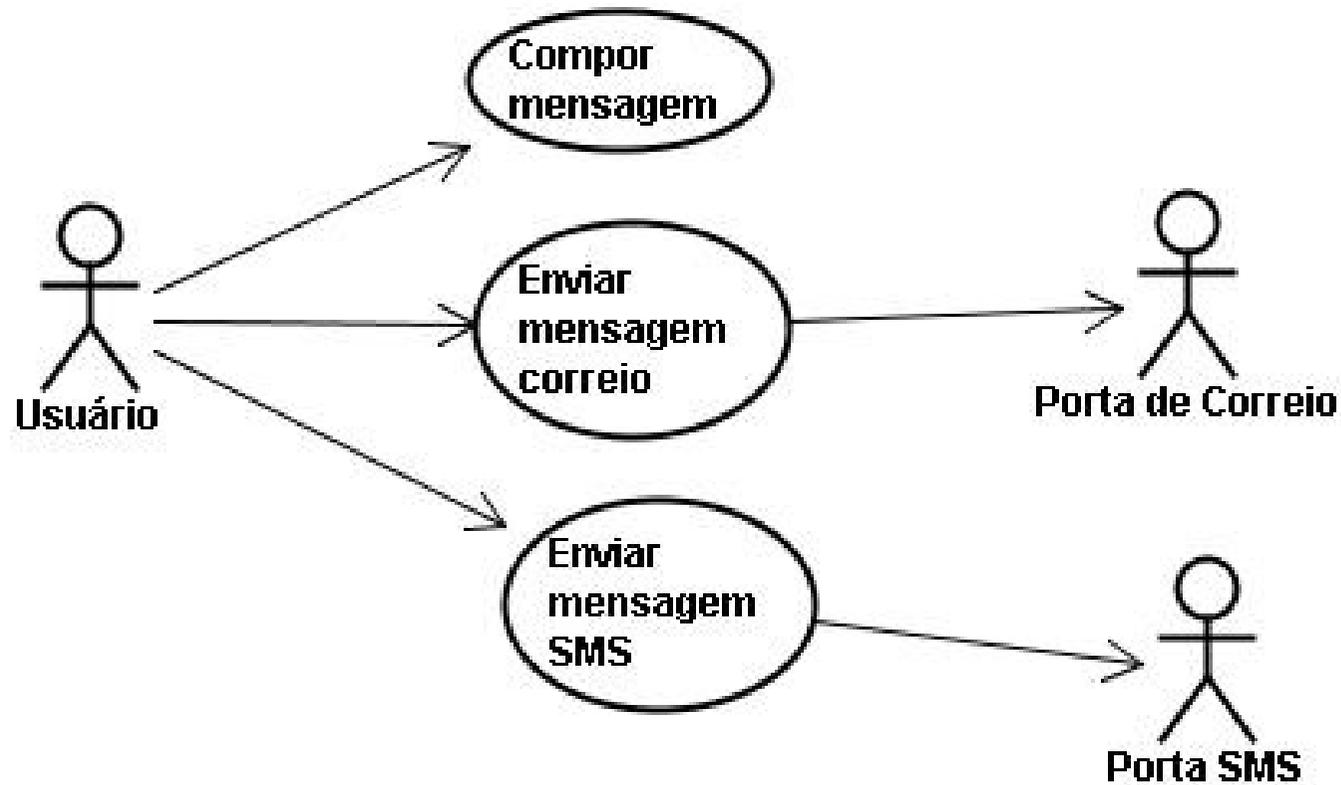
MODELO DE CASO DE USO – ATORES E CASOS DE USO



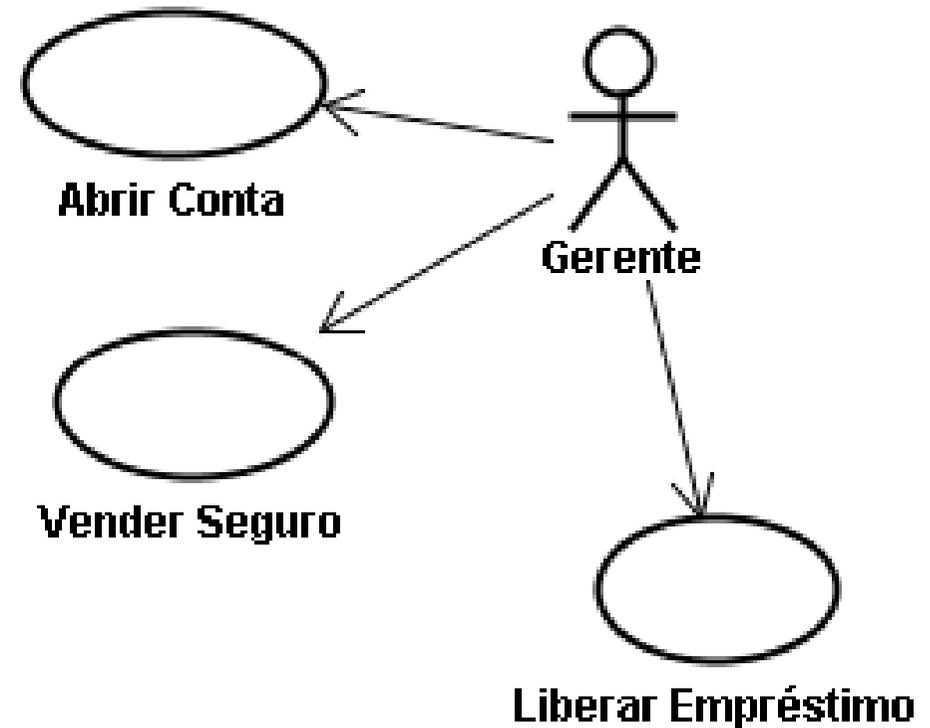
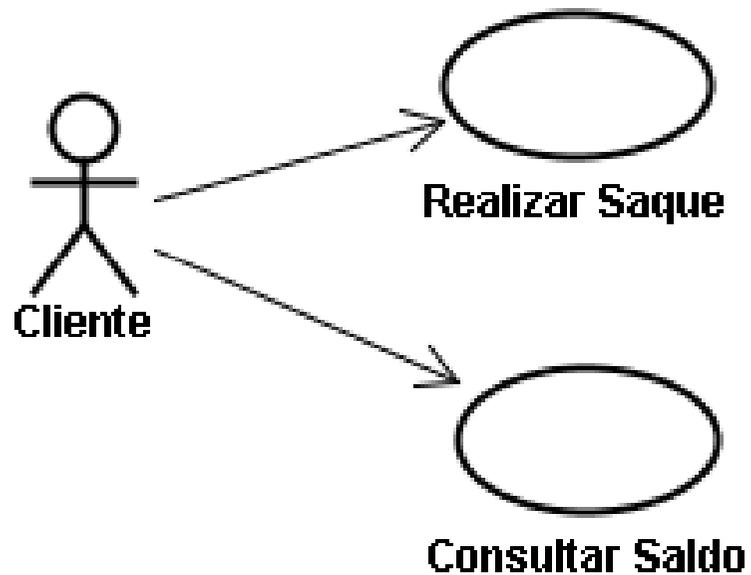
MODELO DE CASO DE USO – ATORES E CASOS DE USO



MODELO DE CASO DE USO – ATORES E CASOS DE USO

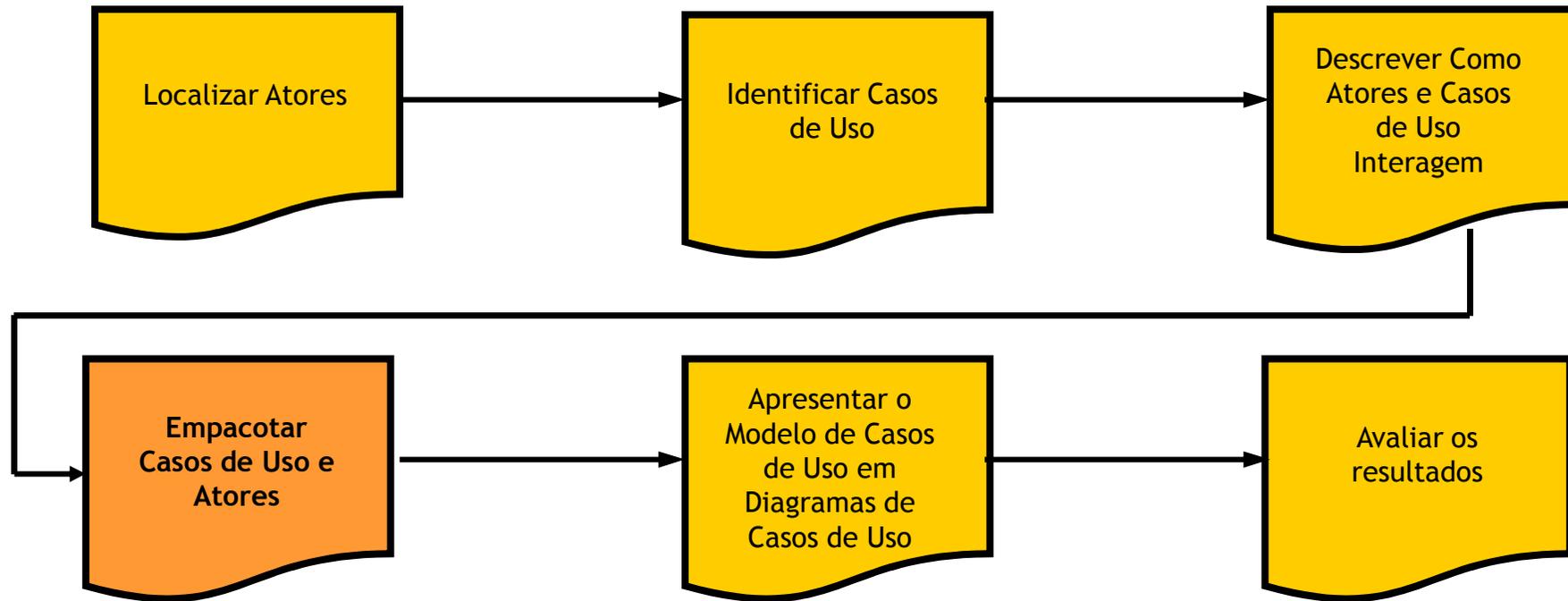


MODELO DE CASO DE USO – ATORES E CASOS DE USO



MODELO DE CASO DE USO – ATORES E CASOS DE USO

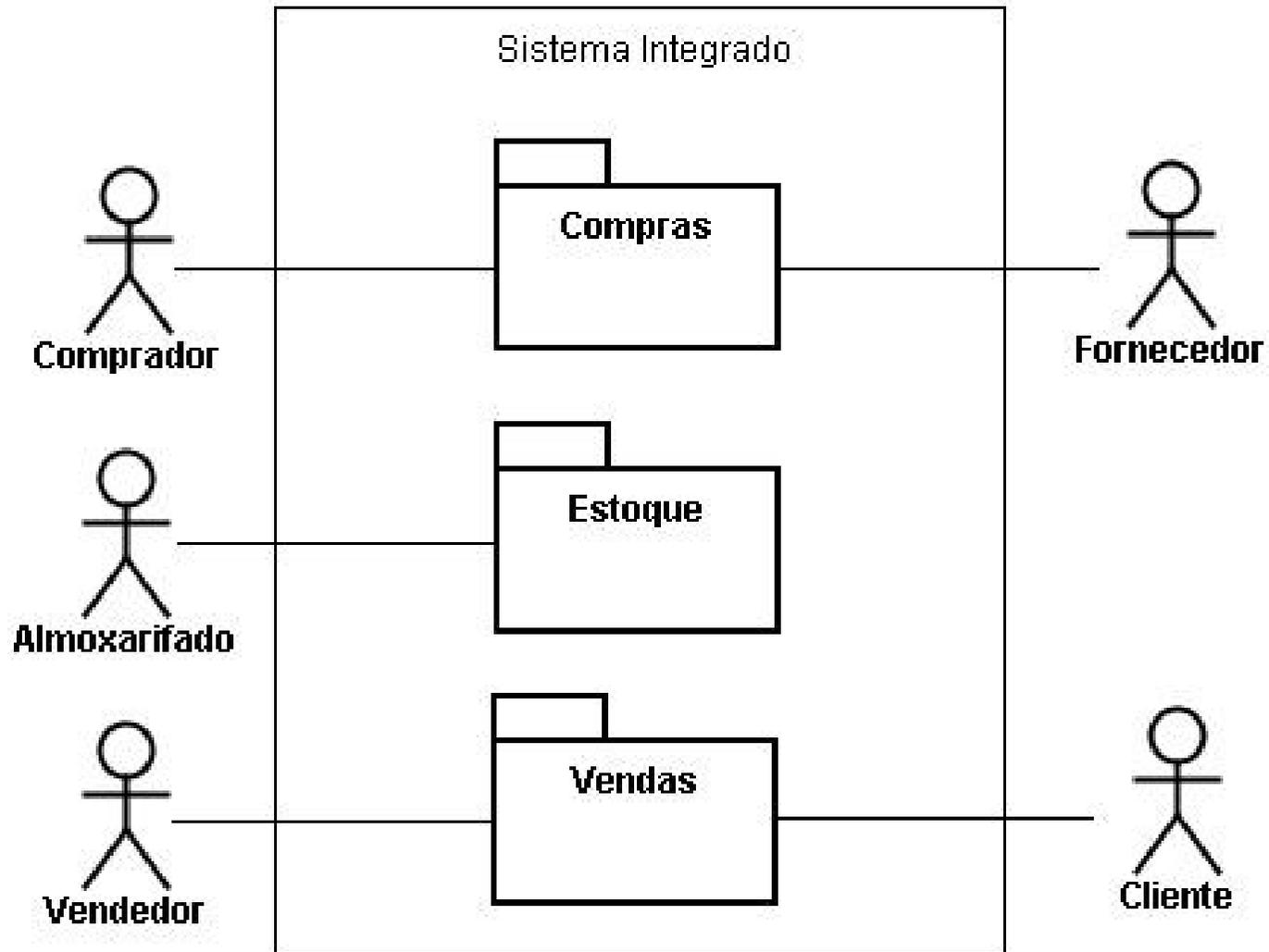
- Passos da atividade Localizar Atores e Casos de Uso



MODELO DE CASO DE USO – ATORES E CASOS DE USO

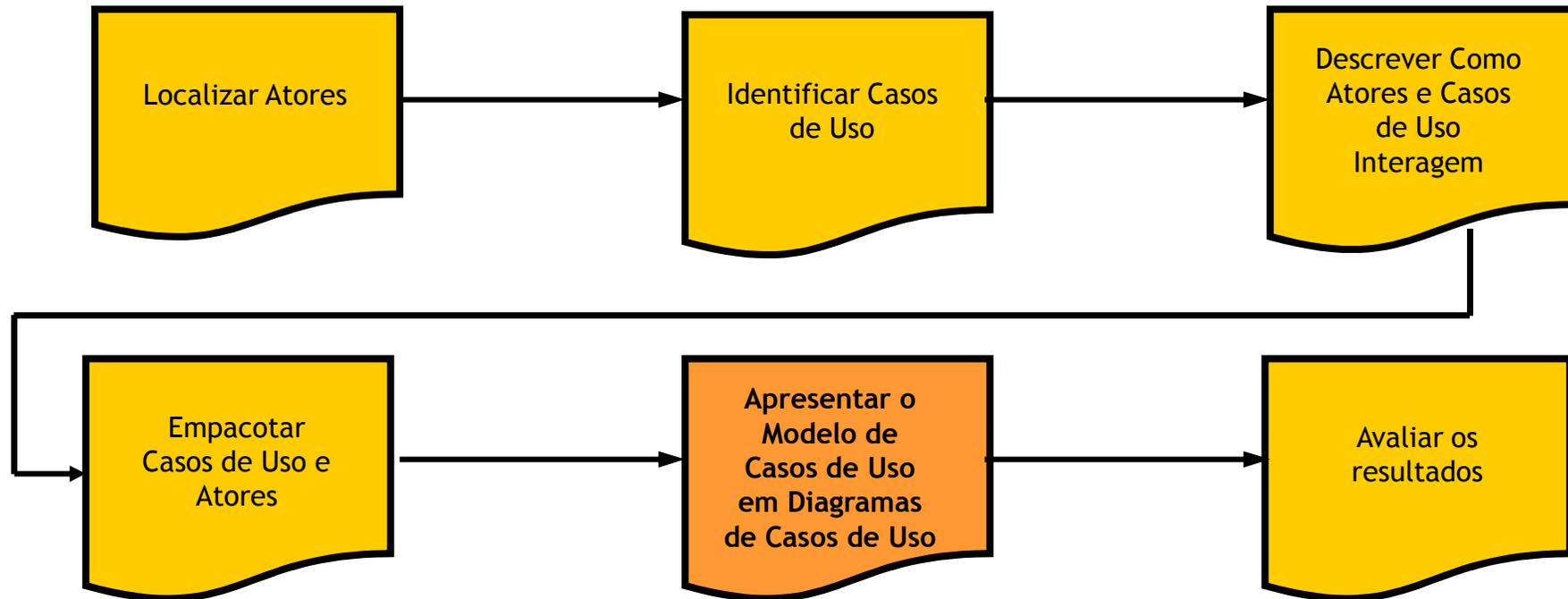
- Organizar casos de uso e atores em pacotes
- Um *pacote de casos de uso* é um conjunto de casos de uso, atores e relacionamentos.
- É usado para **organizar** o modelo de casos de uso dividindo-o em partes menores.
- **Facilita** o entendimento

MODELO DE CASO DE USO – PACOTES

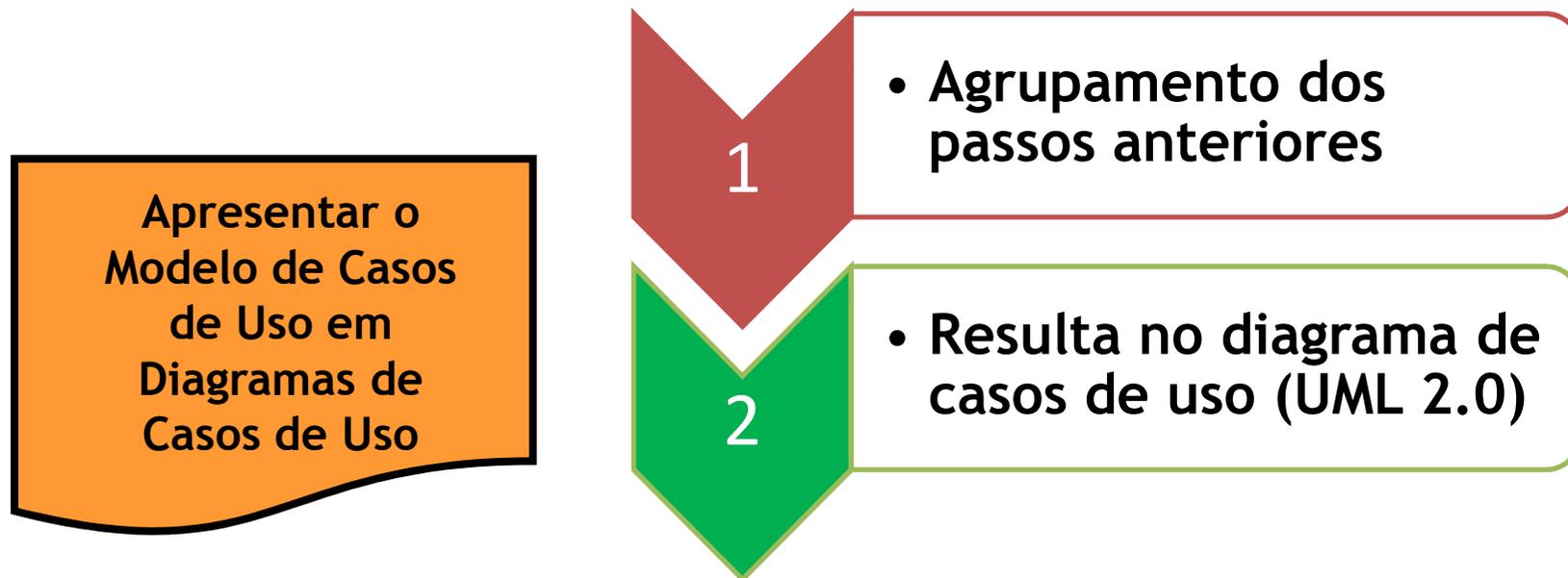


MODELO DE CASO DE USO – ATORES E CASOS DE USO

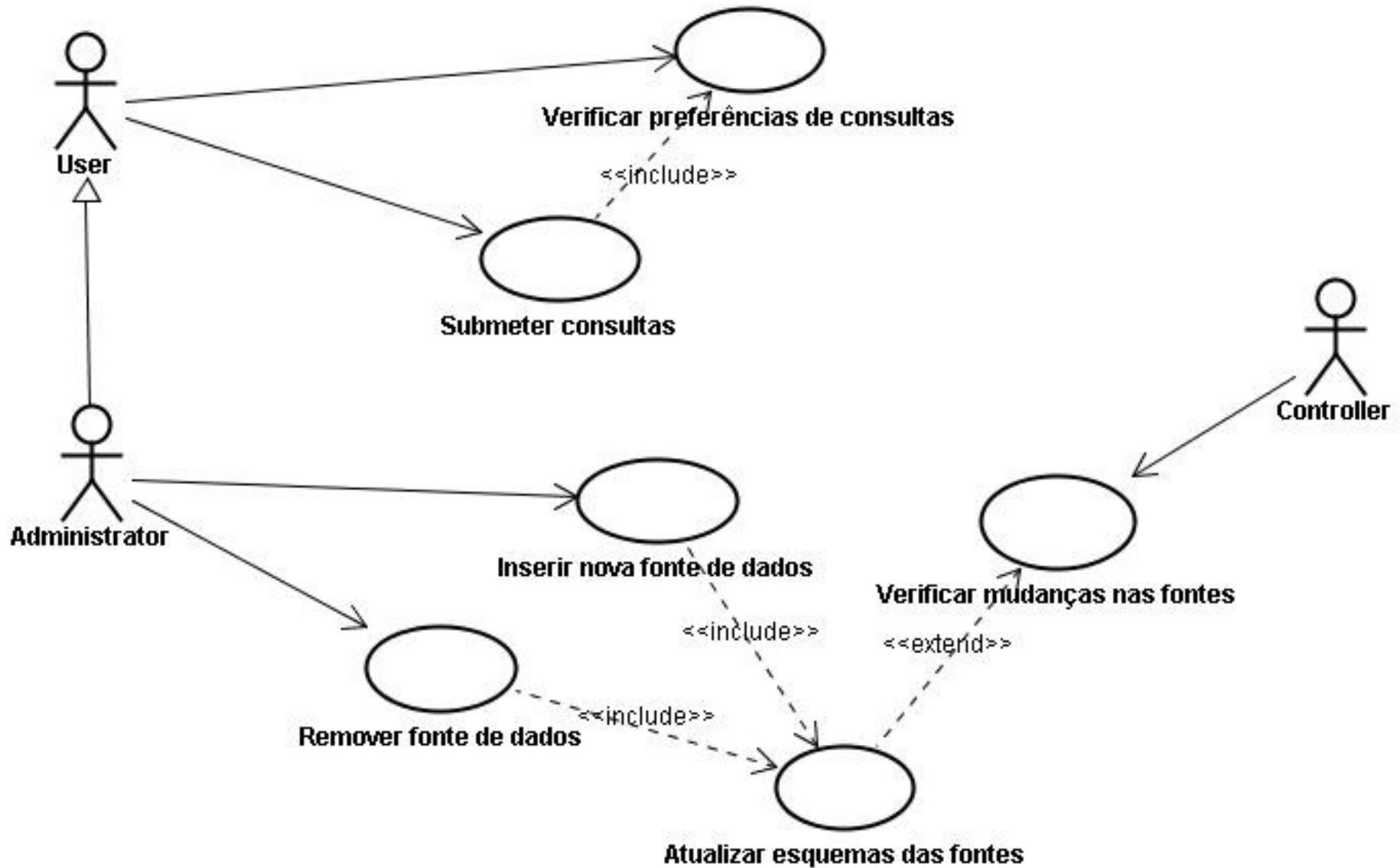
- Passos da atividade Localizar Atores e Casos de Uso



MODELO DE CASO DE USO – ATORES E CASOS DE USO

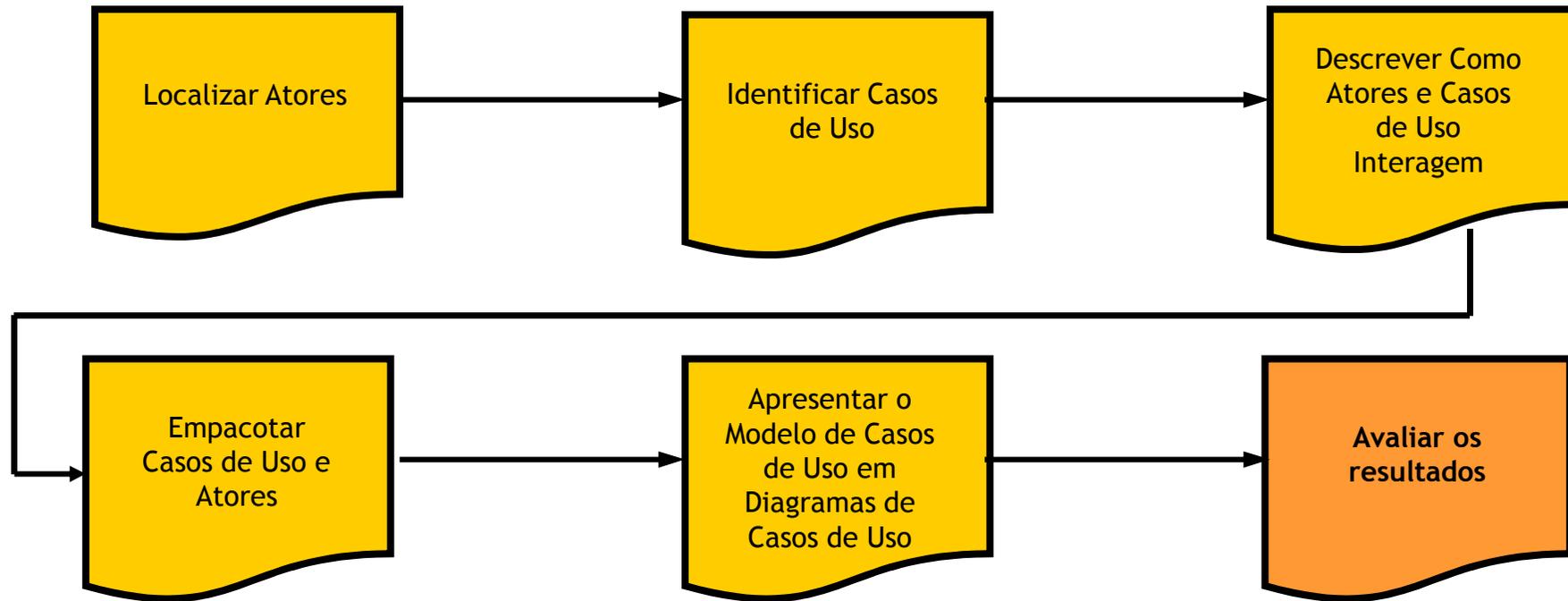


MODELO DE CASO DE USO – ATORES E CASOS DE USO

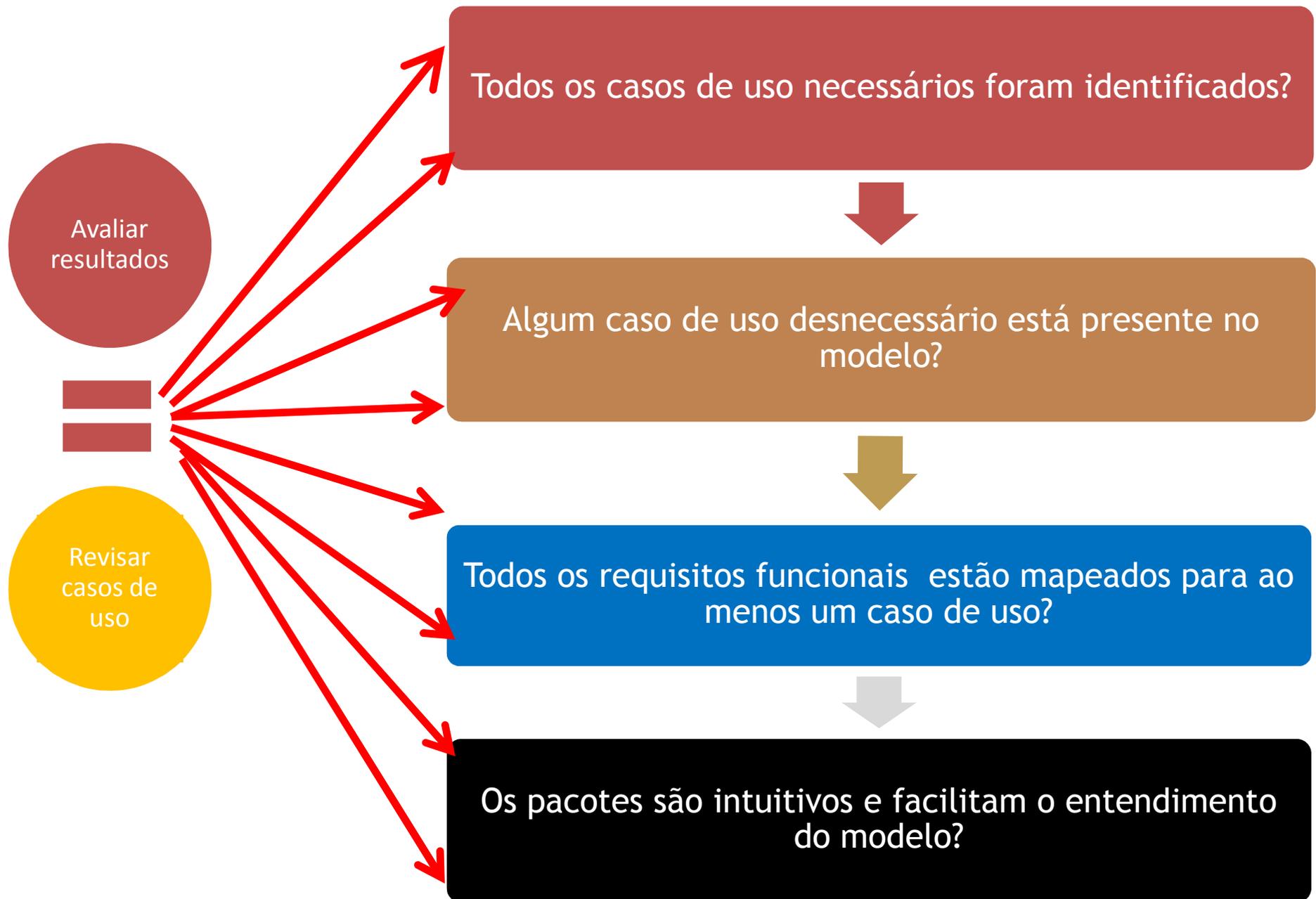


MODELO DE CASO DE USO – ATORES E CASOS DE USO

- Passos da atividade Localizar Atores e Casos de Uso



MODELO DE CASO DE USO – ATORES E CASOS DE USO



EXERCÍCIO

Exercício: Para cada uma das aplicações usadas nos exercícios anteriores (o projeto da fábrica e o exercício do sistema de matrícula que está no moodle), fazer o seguinte :

- 1. Listar os atores envolvidos**
- 2. Listar os casos de uso identificados para cada ator**

MODELO DE CASO DE USO – ESTRUTURAÇÃO

Objetivo:

- Entender a dinâmica da atividade
Estruturar Modelo de Caso de Uso para obtermos o Modelo de Caso de Uso

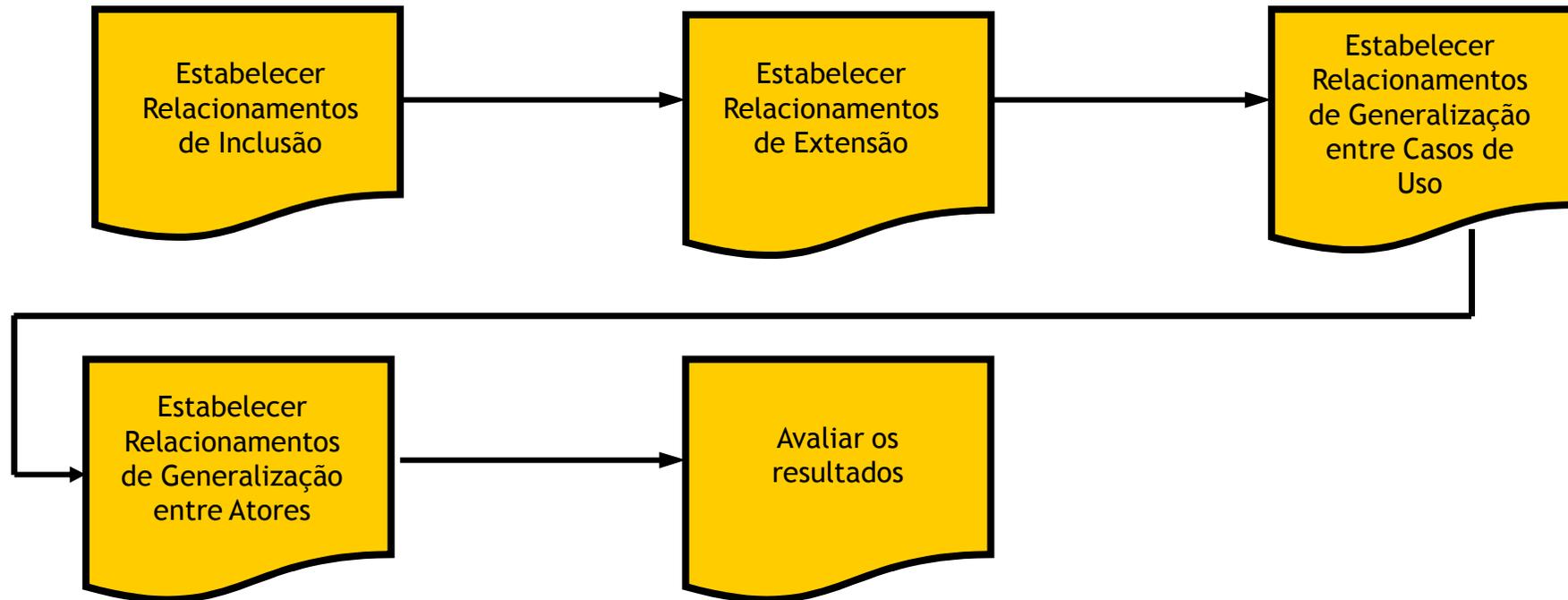
Atividades de documentação de requisitos:

- ✓ Desenvolver Visão
- ✓ Capturar Vocabulário Comum
- ✓ Localizar Atores e Casos de Uso

4. Estruturar Modelo de Caso de Uso

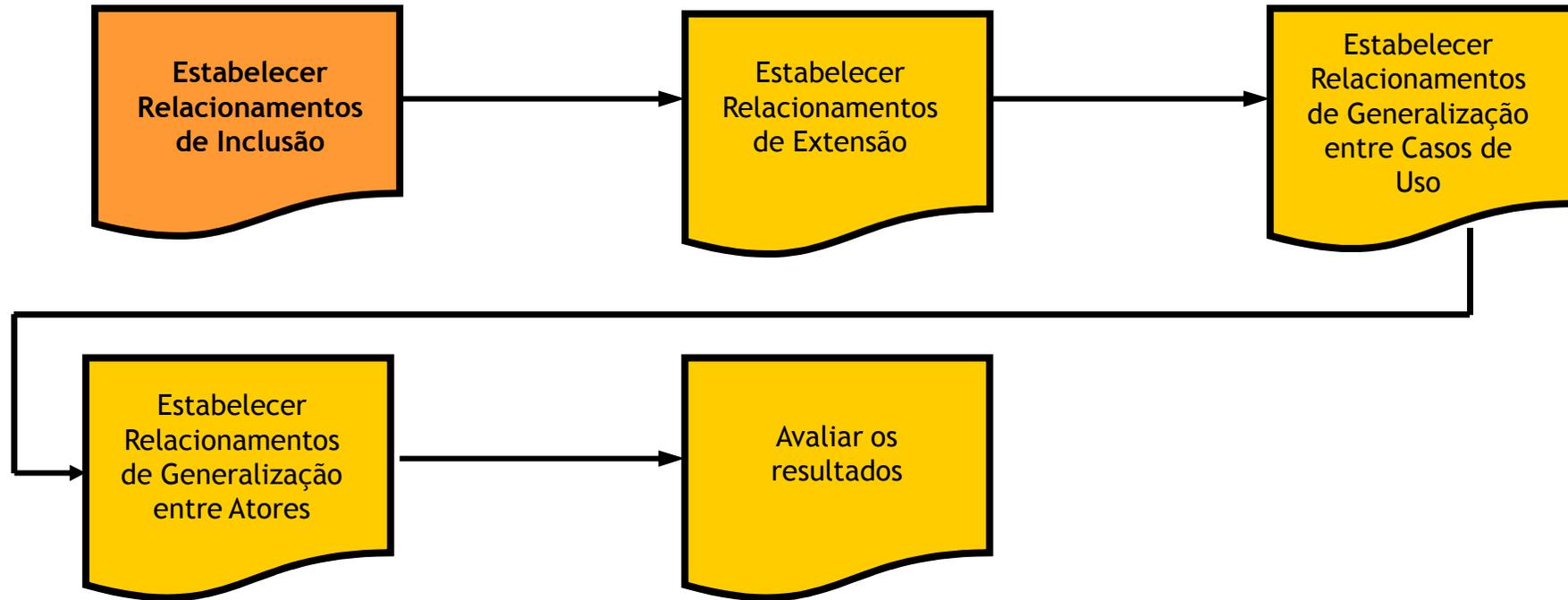
MODELO DE CASO DE USO – ESTRUTURAÇÃO

- Passos da atividade Estruturar o Modelo de Casos de Uso



MODELO DE CASO DE USO – ESTRUTURAÇÃO

- Passos da atividade Estruturar o Modelo de Casos de Uso



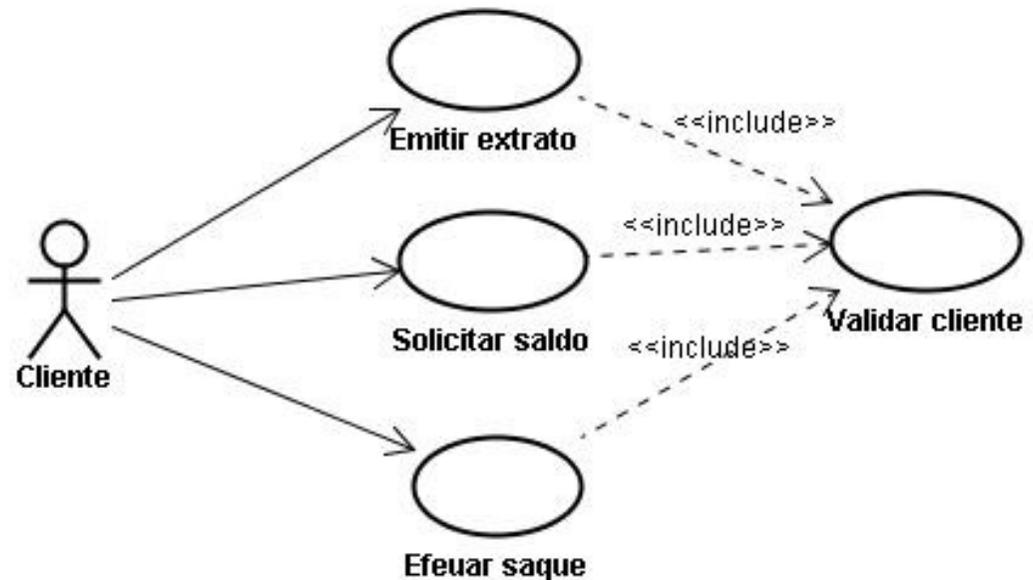
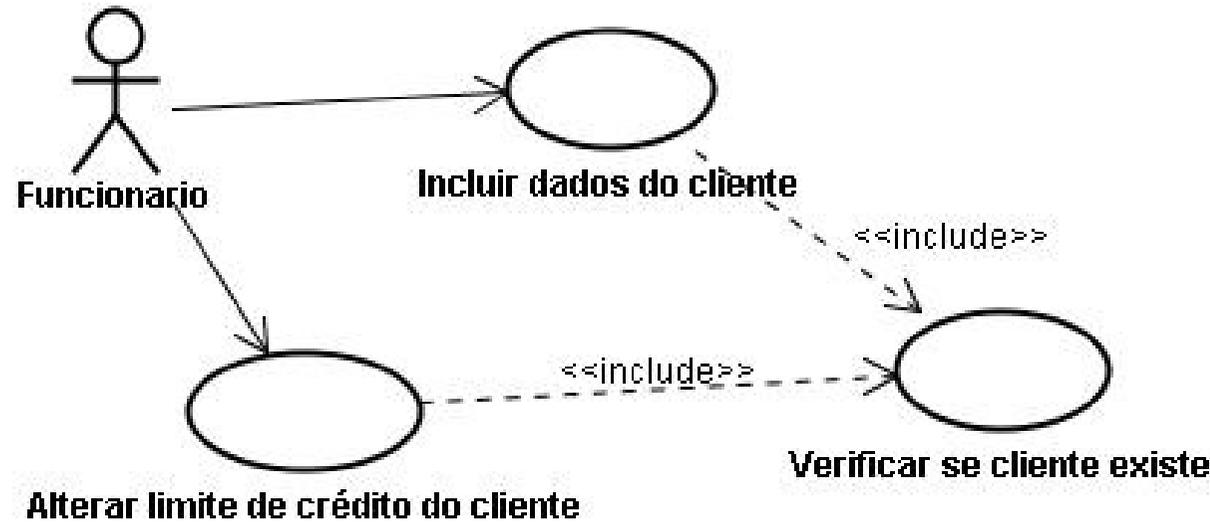
MODELO DE CASO DE USO – ESTRUTURAÇÃO

- Um relacionamento de inclusão é aquele que
 - Associa um caso de uso base a um caso de uso de inclusão
 - O comportamento definido para o caso de uso de inclusão **é inserido** de forma explícita no comportamento definido para o caso de uso base
 - Favorecem reuso
 - Servem para fatorar:
 - separar comportamento complexo do caso de uso ou;
 - comportamento que não faz parte do **propósito principal** do caso de uso a fim de diminuir a complexidade do fluxo de eventos

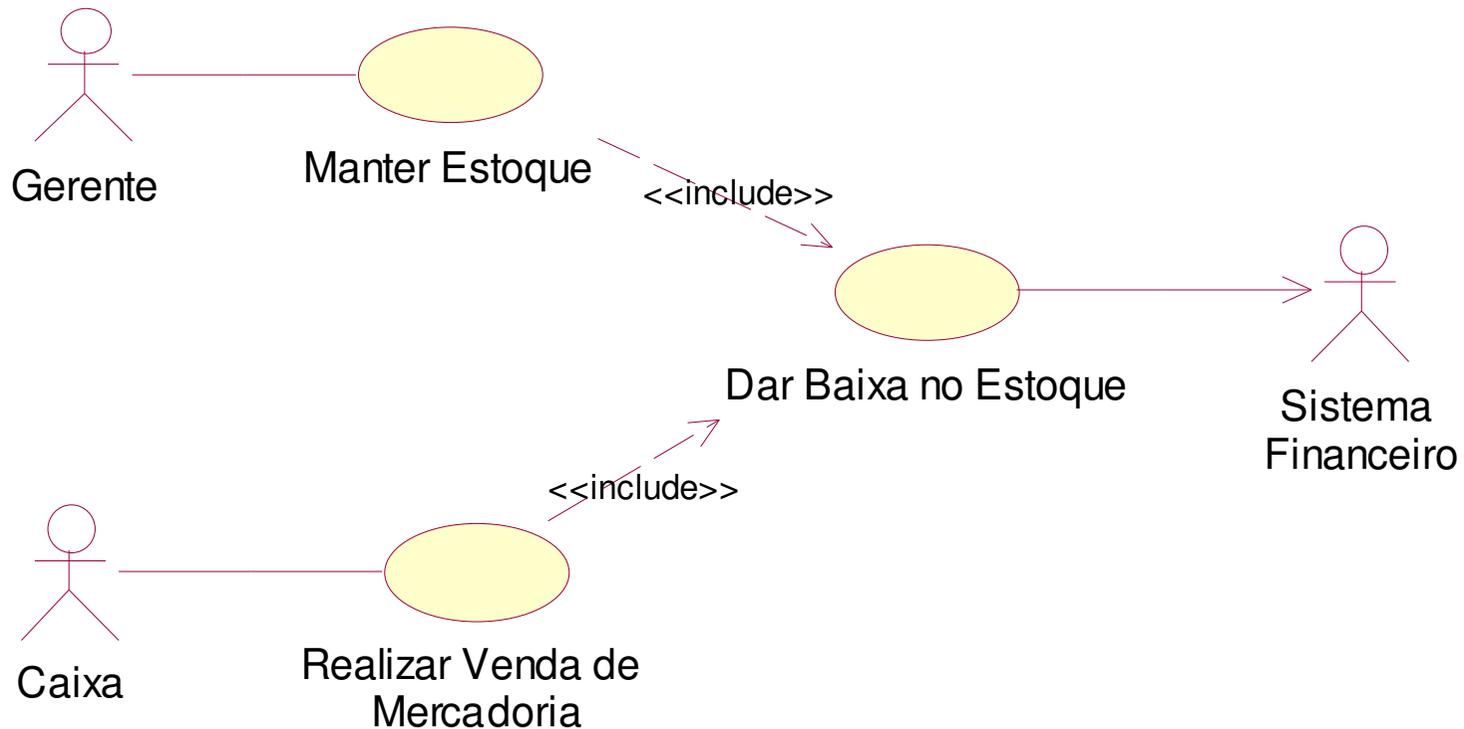
Inclusão

- Existe apenas **entre** casos de uso
- Quando dois ou mais casos de uso incluem uma **seqüência de interações comum**, esta seqüência comum pode ser descrita em um outro caso de uso (**de inclusão**)
- Este caso de uso comum:
 - **Evita** a descrição de uma mesma seqüência **mais de uma vez**

Casos de Uso de Inclusão

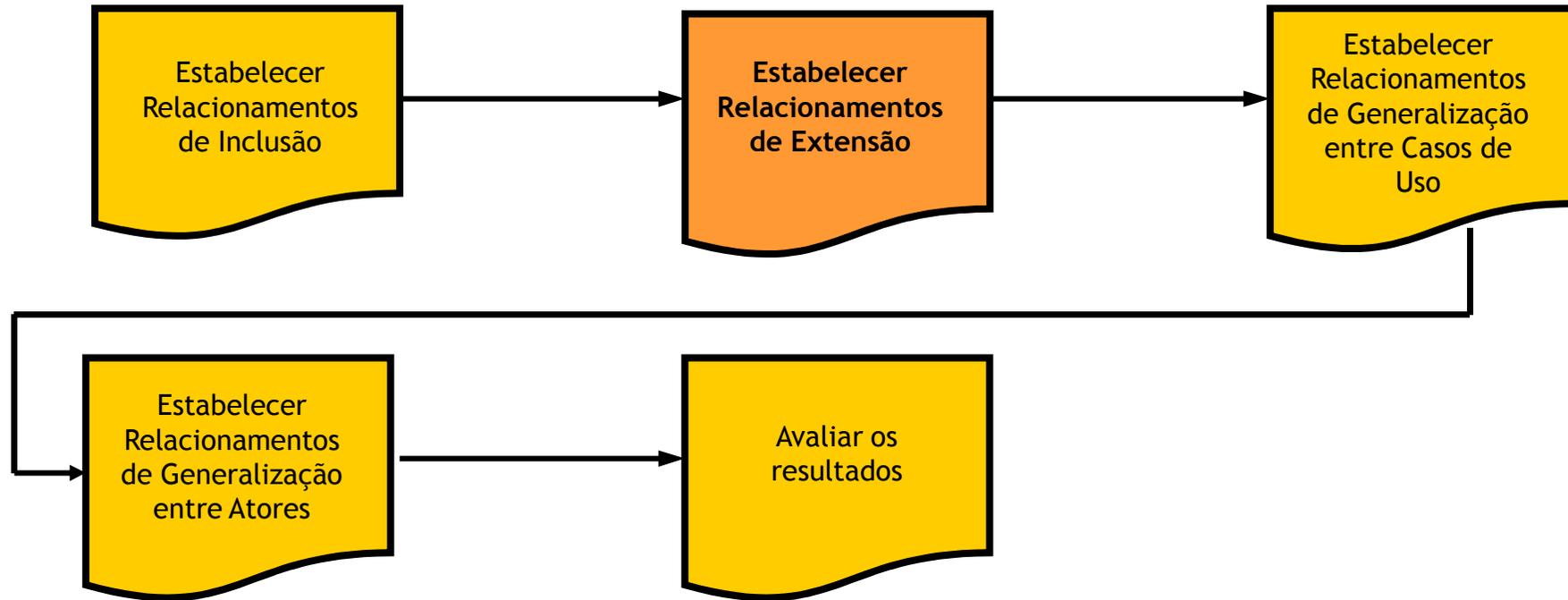


Casos de Uso de Inclusão



MODELO DE CASO DE USO – ESTRUTURAÇÃO

- Passos da atividade Estruturar o Modelo de Casos de Uso



MODELO DE CASO DE USO – ESTRUTURAÇÃO

- ✓ Um relacionamento de extensão
 - ✓ Associa um caso de uso de extensão a um **CASO DE USO BASE**
 - ✓ O comportamento definido para o caso de uso de extensão pode ser inserido no comportamento definido para o caso de uso de base
- ✓ Criados para modelar comportamentos opcionais ou excepcionais
 - ✓ São executados somente face a certas **condições**
 - ✓ **Casos de uso de inclusão são sempre executados**

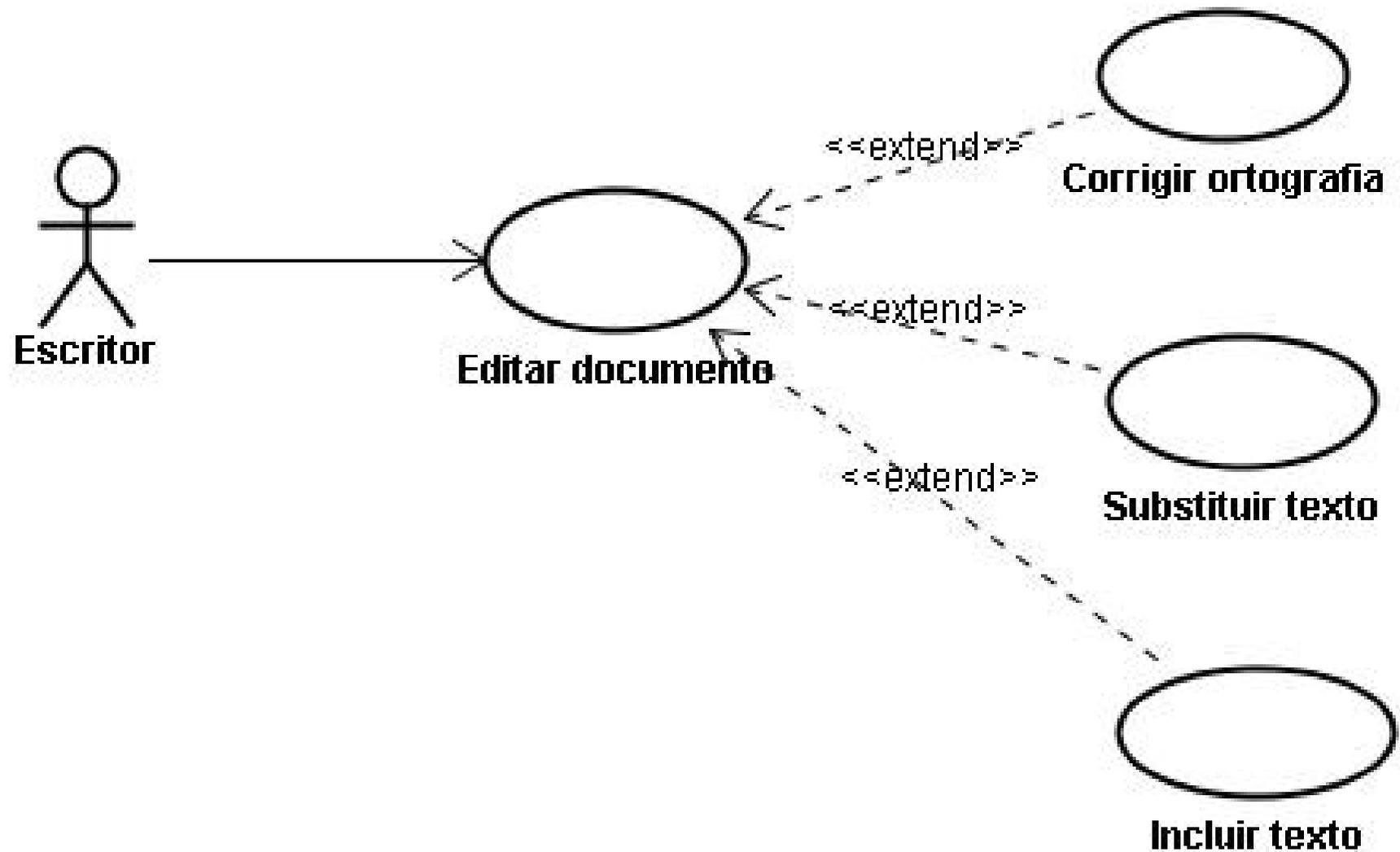
Extensão

- Sejam A e B dois casos de uso.
- Um relacionamento de extensão de B para A indica que um ou mais dos cenários de *A podem* incluir o comportamento especificado por B
- Neste caso, diz-se que *B estende A*
- O caso de uso A é chamado de *estendido* e o caso de uso B de *extensor*

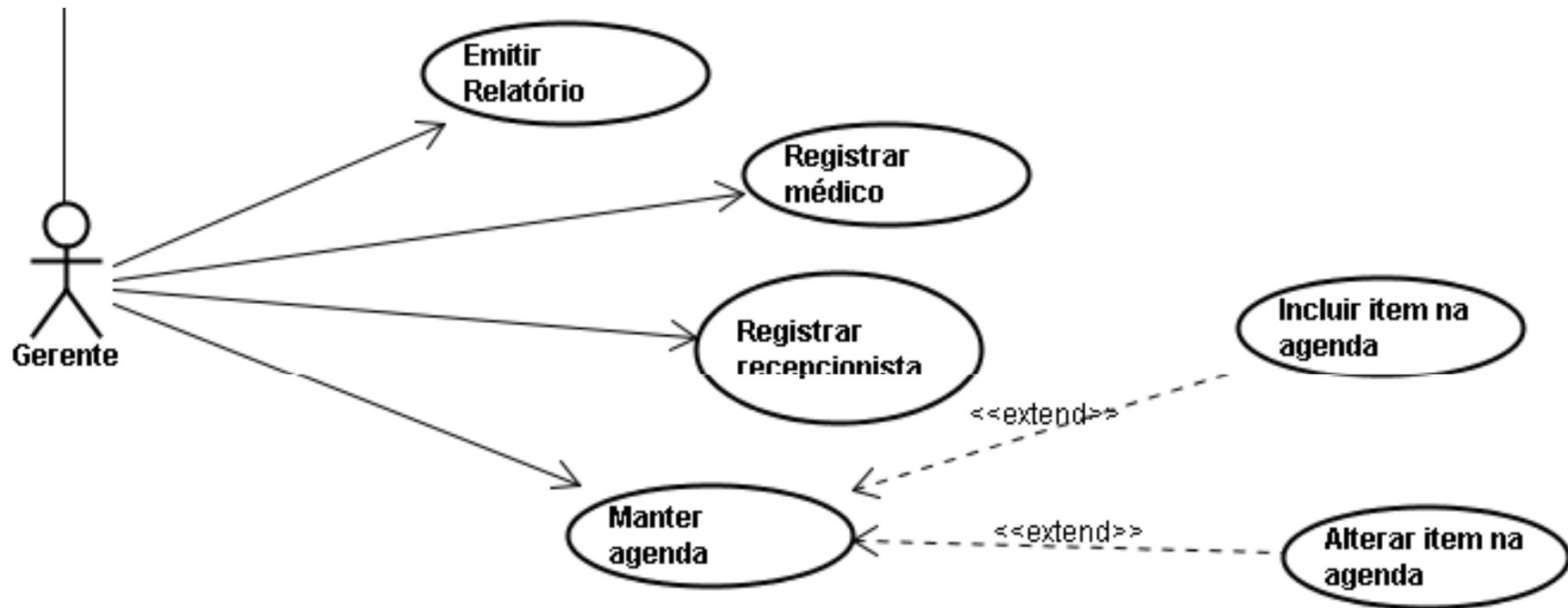
Extensão

- ✓ A execução do caso de uso B (extensor) **depende de escolha do ator**
- ✓ Após a execução de B
 - ✓ O fluxo `volta ao caso de uso principal`, recomeçando **logo após o ponto** em que o extensor foi inserido

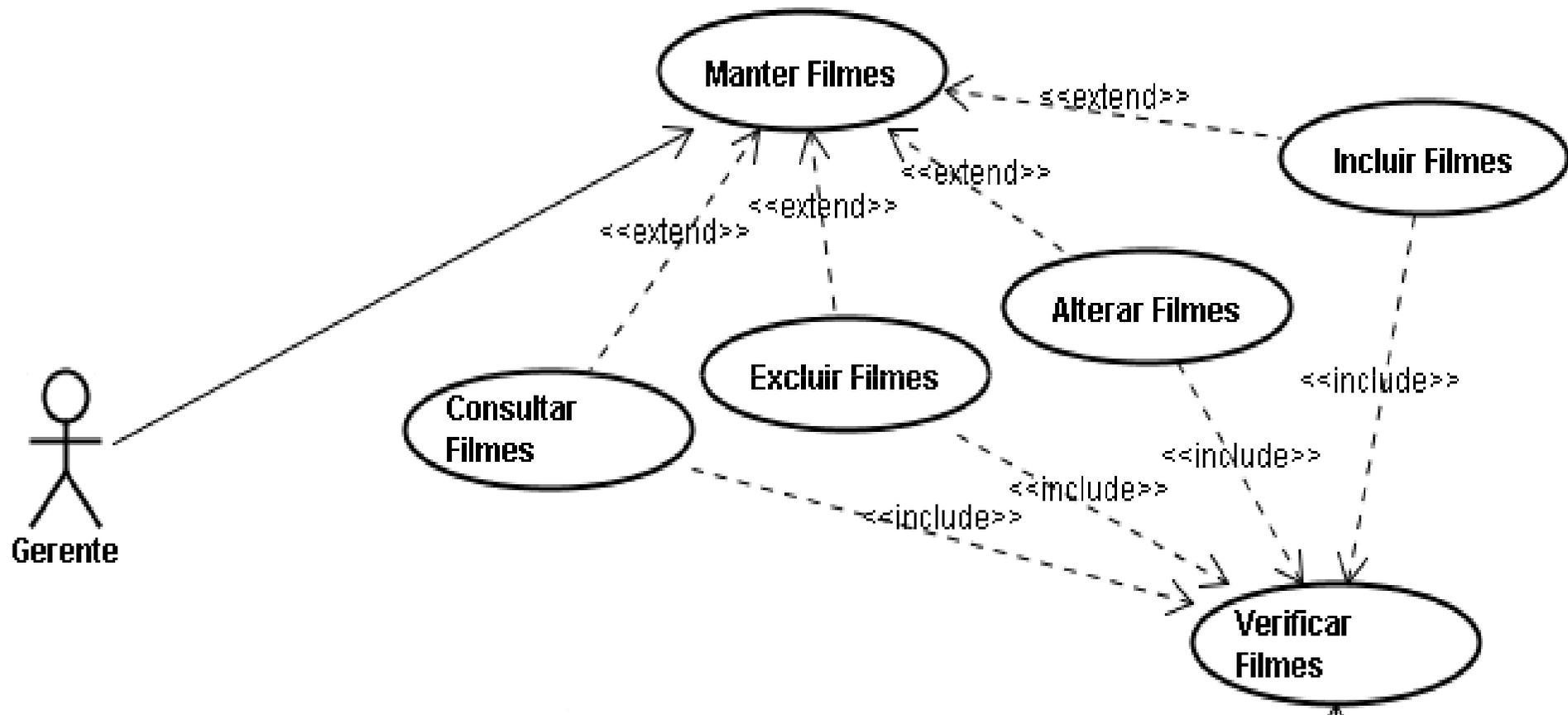
Casos de Uso de Extensão



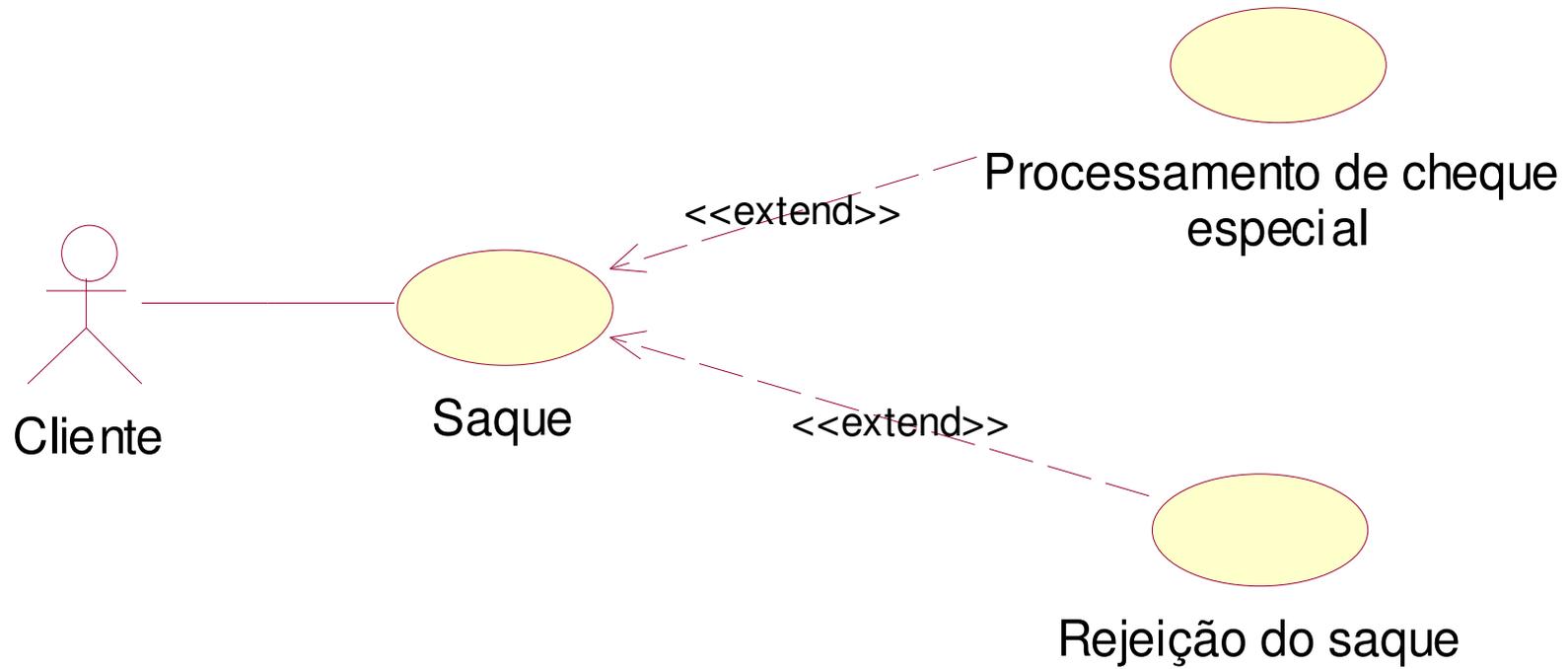
Casos de Uso de Extensão



Casos de Uso de Extensão



Casos de Uso de Extensão



Inclusão x Extensão

Inclusão:

- ✓ O comportamento definido no caso de uso incluído é inserido explicitamente no caso de uso base.
 - ✓ Essa é a principal diferença entre relacionamentos de inclusão e extensão.
- ✓ Em um relacionamento de inclusão, há um momento exato para se invocar o caso de uso incluído

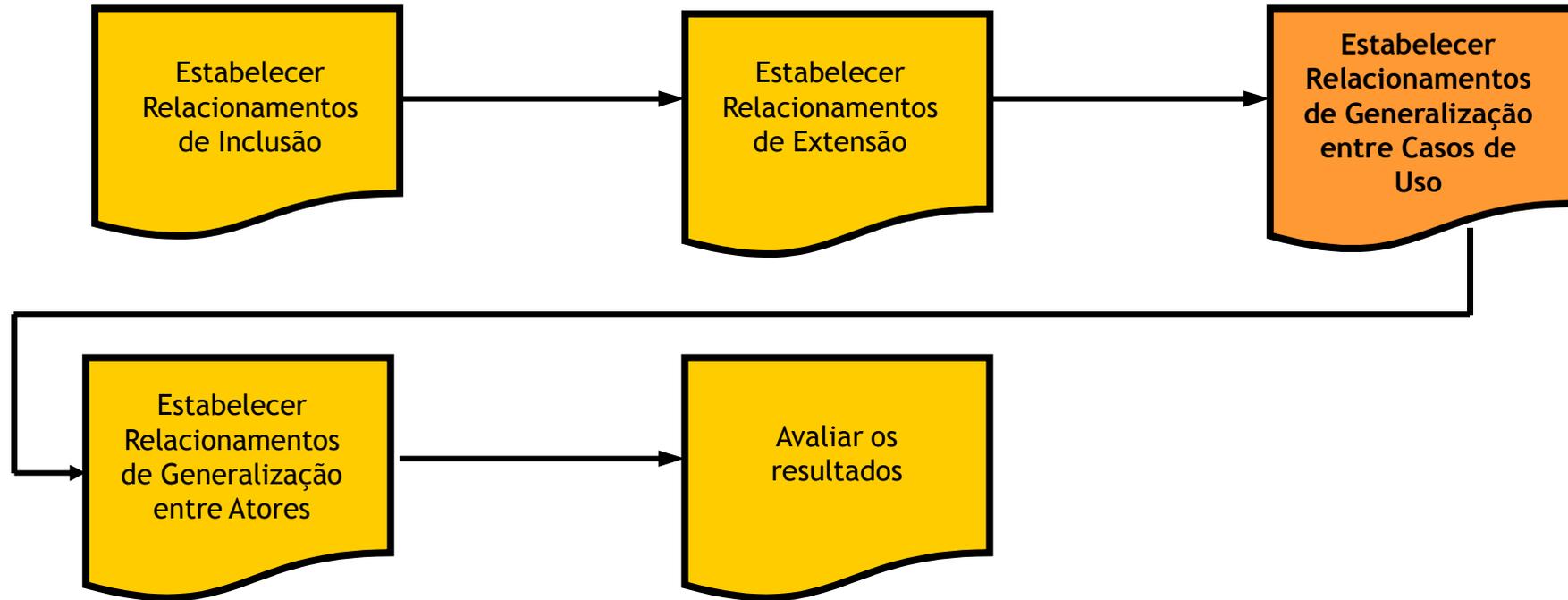
Inclusão x Extensão

É comum identificar Extensão para:

- ✓ **Comportamento Opcional ou Excepcional:**
 - ✓ Representar comportamentos excepcionais ou opcionais executados somente face a certas condições;
- ✓ **Momento indefinido para acontecer:**
 - ✓ Representar comportamentos que não possuem um momento certo para acontecer, ou seja, podem acontecer em qualquer momento do fluxo de eventos do caso de uso base

MODELO DE CASO DE USO – ESTRUTURAÇÃO

- Passos da atividade Estruturar o Modelo de Casos de Uso



MODELO DE CASO DE USO – ESTRUTURAÇÃO

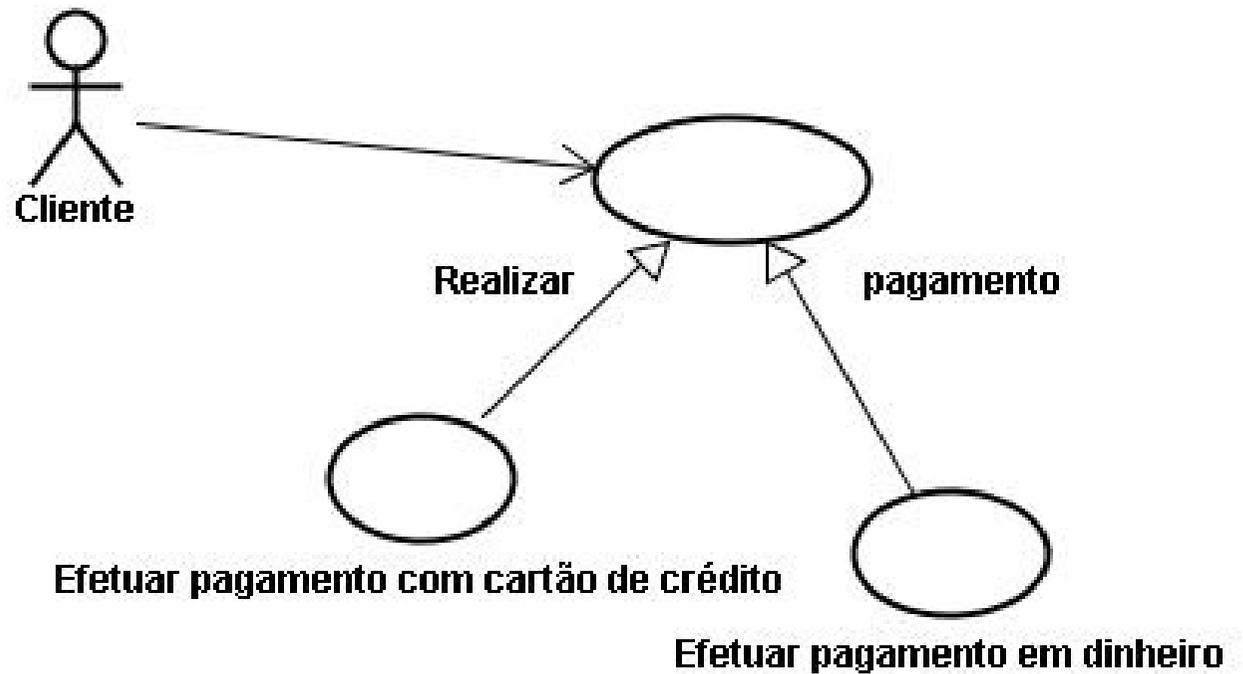
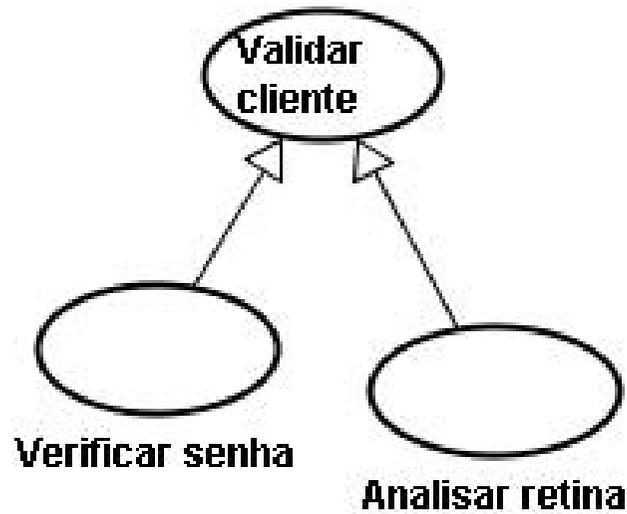
→ Uma generalização de casos de uso

- É um relacionamento de um **caso de uso filho** com um **caso de uso pai**
- O filho **pode** adotar todo o **comportamento** e as **características** descritas para o pai

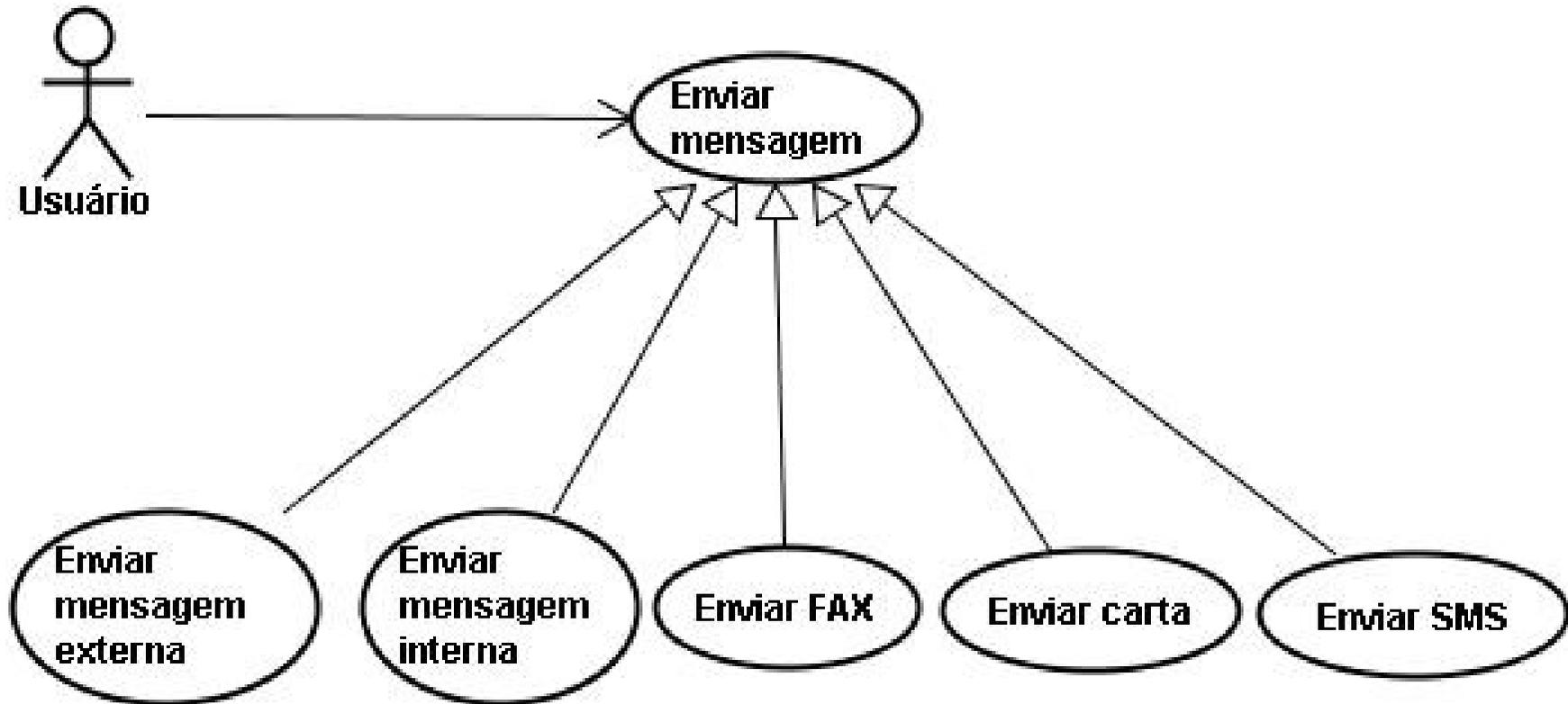
→ Usado para relacionar um caso de uso mais **especializado** com um mais **geral**

→ Este relacionamento complica mais do que facilita e deve ser evitado

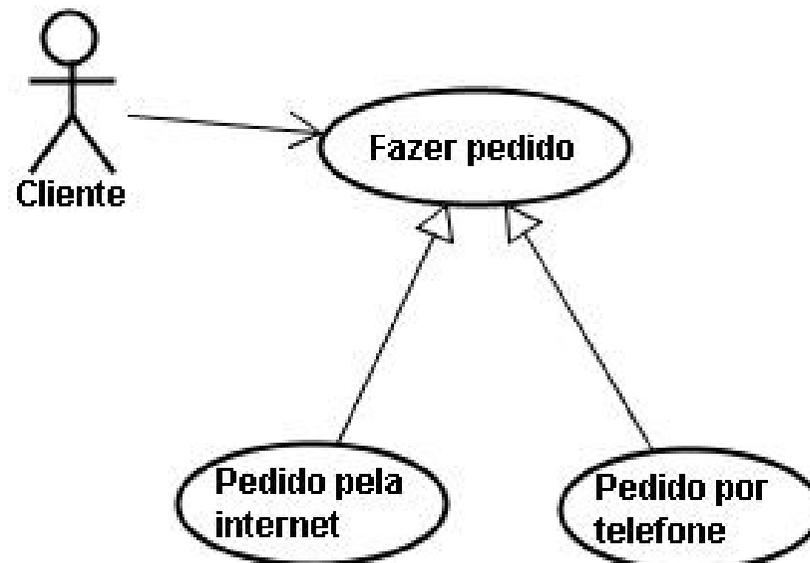
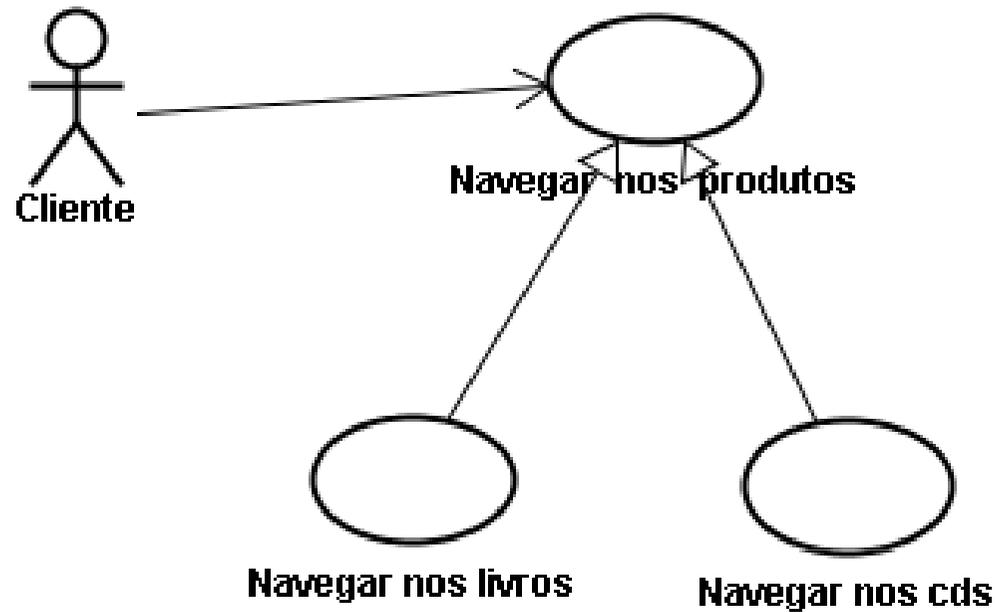
Casos de Uso de Generalização



Casos de Uso de Generalização



Casos de Uso de Generalização



Notação

- Os relacionamentos de **inclusão** e **extensão** são representados por uma seta direcionada de um caso de uso para outro
- A seta (**tracejada**) de um relacionamento de inclusão recebe o estereótipo `<<include>>`
- A seta (**tracejada**) de um relacionamento de extensão recebe o estereótipo `<<extend>>`
- A seta (**sólida**) de um relacionamento de **generalização** não recebe estereótipo

Estudo de Caso

- O Supermercado XXXXXX quer agilizar e modernizar a forma com que seus clientes efetuam as compras
- Para que isso ocorra pretende disponibilizar terminais de consultas aos preços, por todo o supermercado
- Assim, o *cliente poderá, a* qualquer momento, consultar o preço de um produto

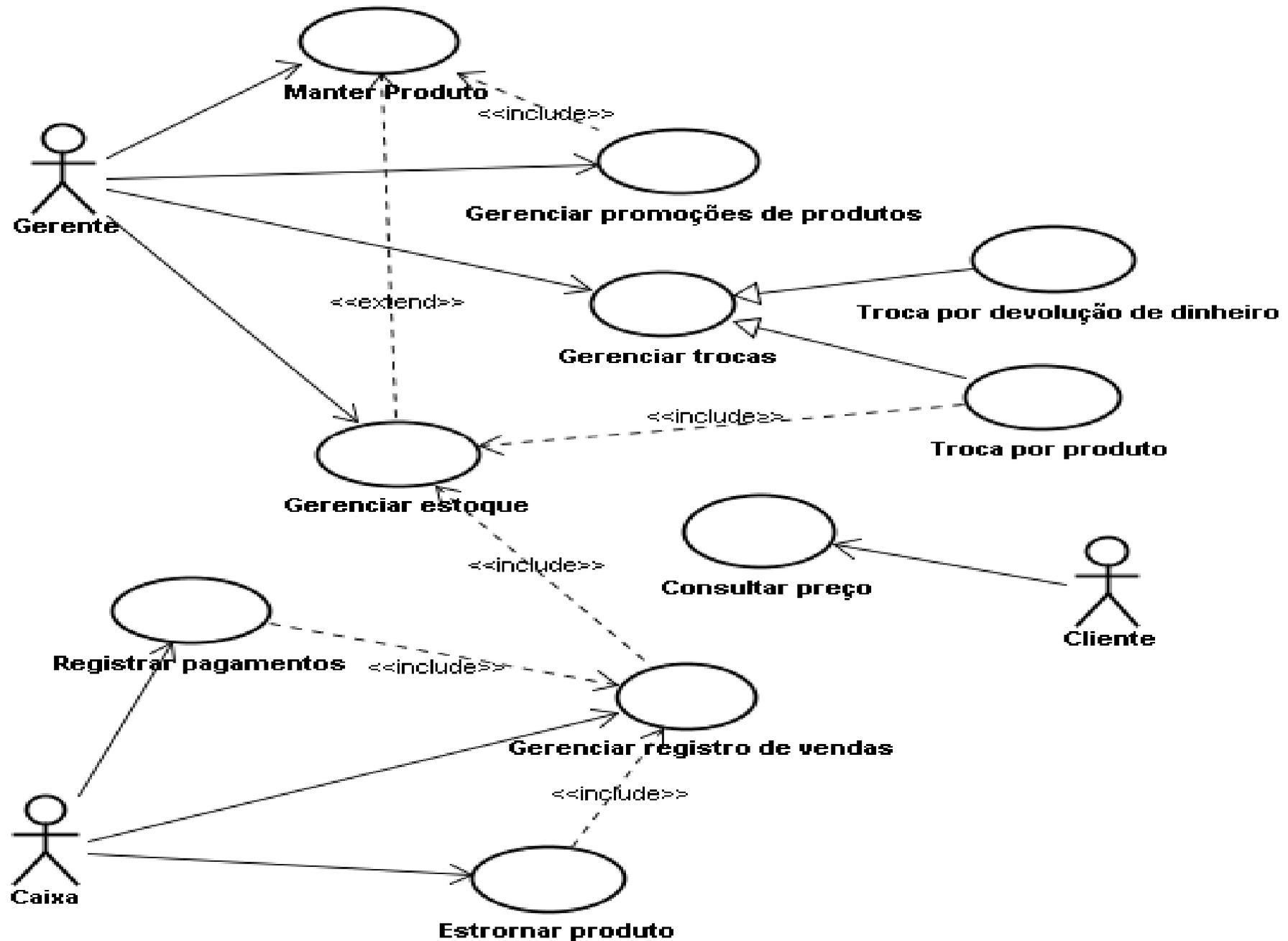
Estudo de Caso

- O gerente do supermercado recebe novos produtos e os cadastra no sistema
- Concluído o cadastro dos produtos, o gerente autoriza os funcionários a colocarem estes produtos nas prateleiras
- Outra tarefa do *gerente é realizar a troca dos produtos* para os clientes.
 - Ao realizar a troca, o *gerente verifica se há* ainda produtos disponíveis para que a mesma possa ser efetuada. Caso não haja produtos disponíveis o gerente efetua a devolução do dinheiro do cliente.

Estudo de Caso

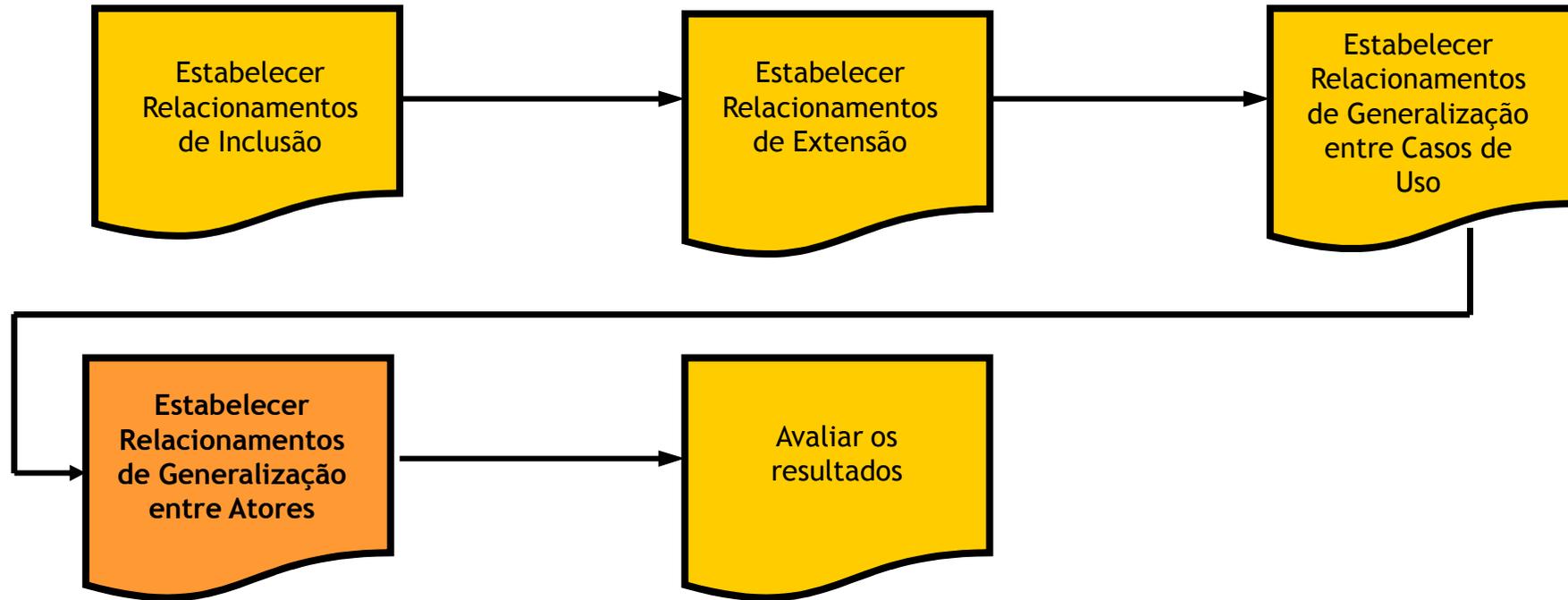
- Todas as compras dos clientes são registradas pelo funcionário caixa. Caso o caixa registre o produto duas vezes ou o cliente desista de levá-lo, então é feito o estorno deste produto
- Concluído o registro de todos os produtos o caixa armazena o pagamento do cliente

MODELO DE CASO DE USO – ESTRUTURAÇÃO



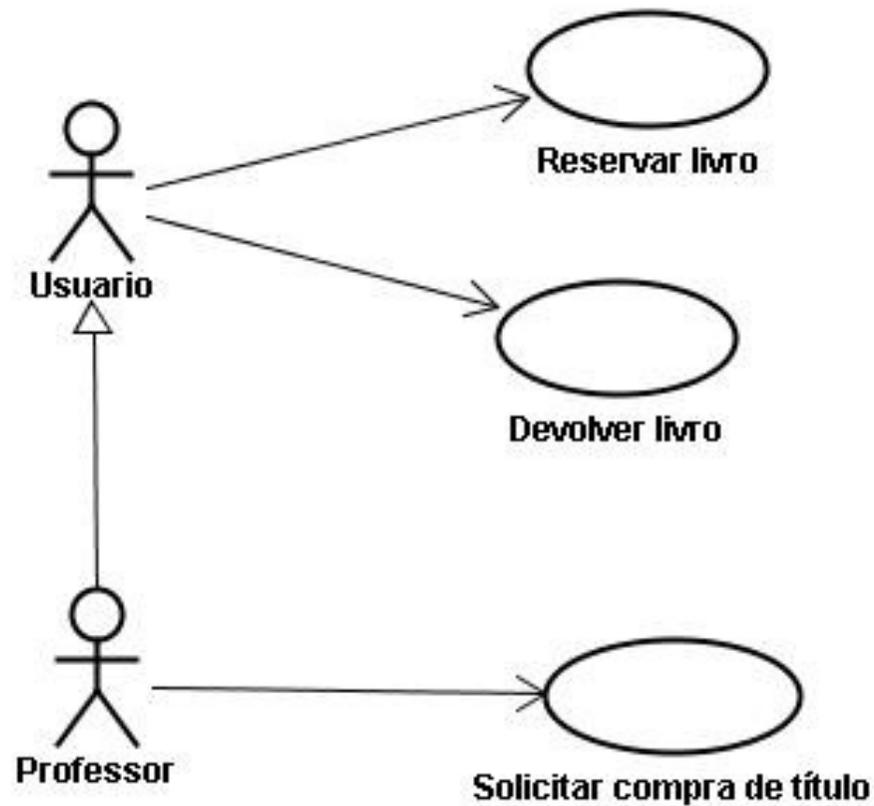
MODELO DE CASO DE USO – ESTRUTURAÇÃO

- Passos da atividade Estruturar o Modelo de Casos de Uso



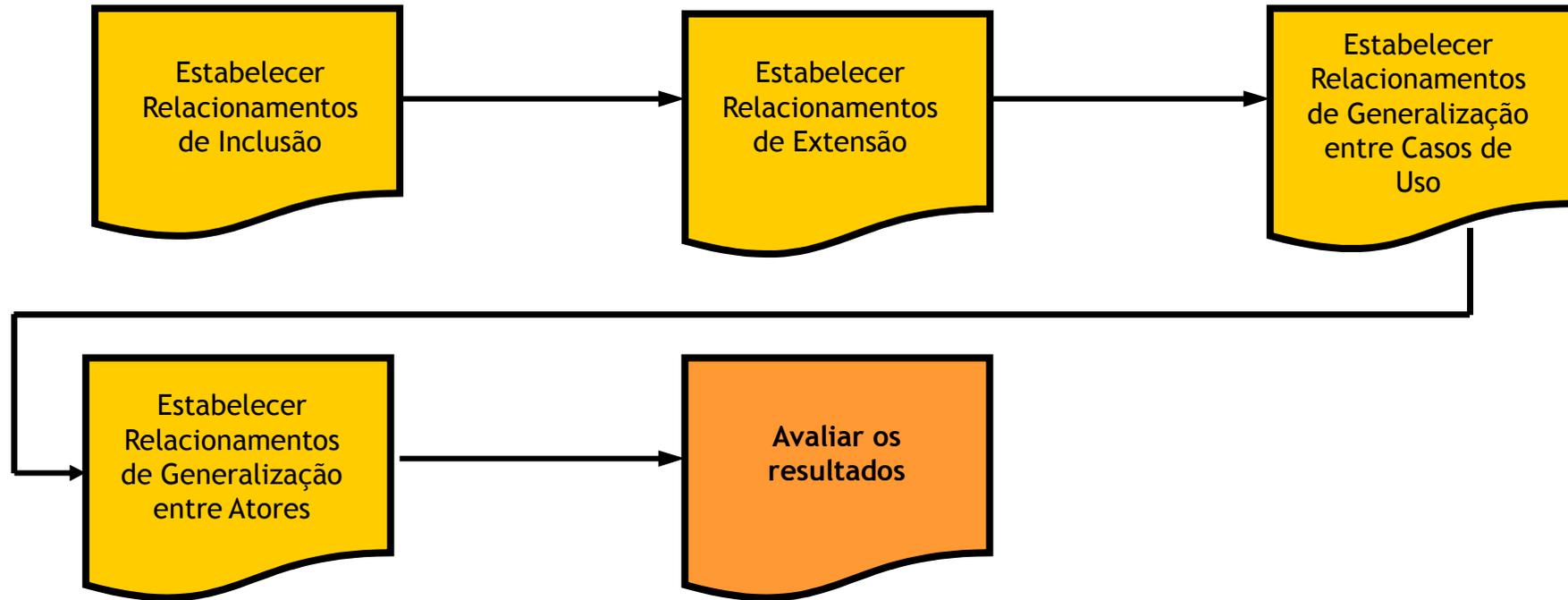
MODELO DE CASO DE USO – ESTRUTURAÇÃO

- Na generalização de ator o filho herda o papel que o pai pode desempenhar em um caso de uso.
- Facilita a leitura do modelo de caso de uso

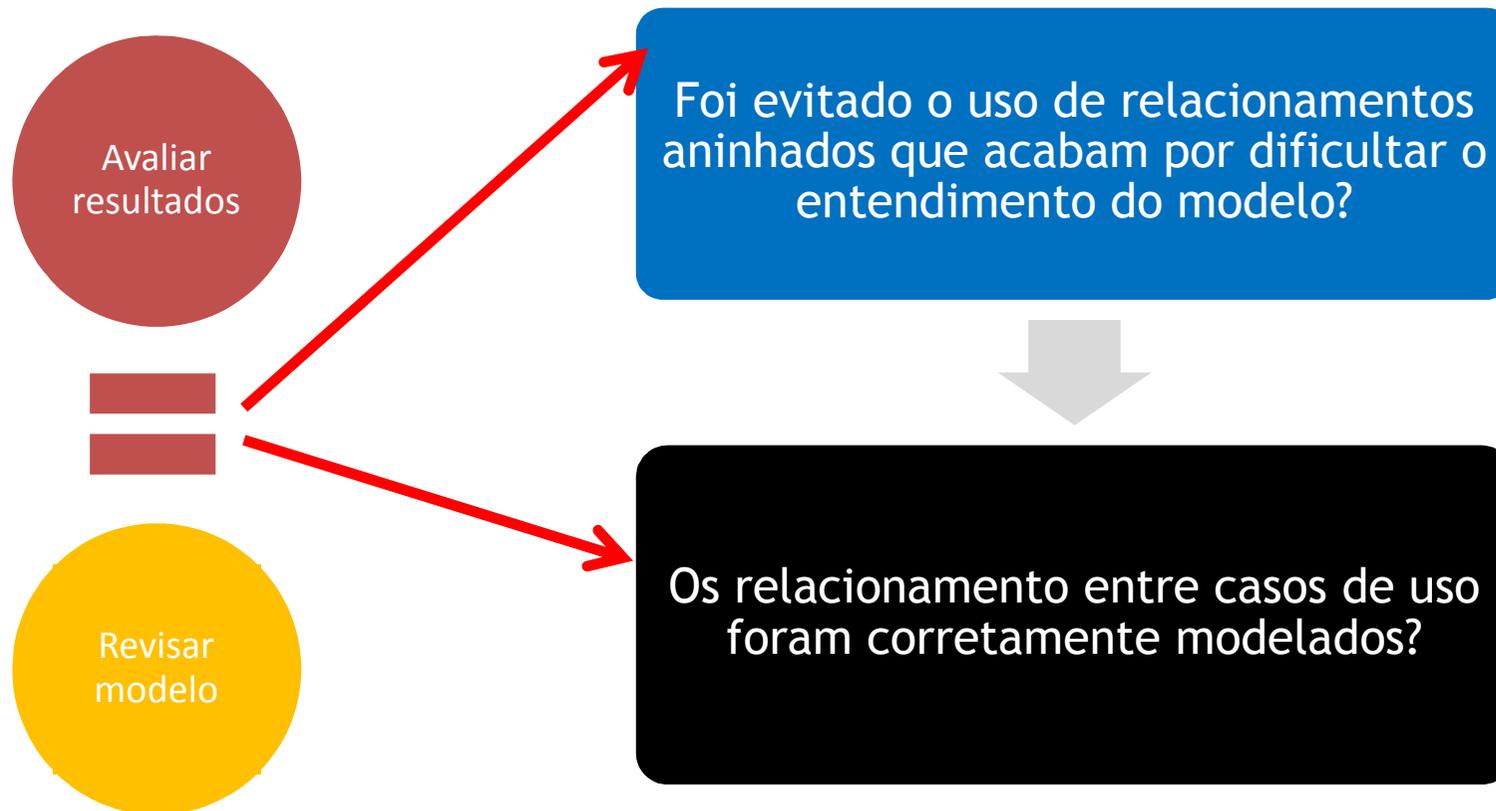


MODELO DE CASO DE USO – ESTRUTURAÇÃO

- Passos da atividade Estruturar o Modelo de Casos de Uso



MODELO DE CASO DE USO – ESTRUTURAÇÃO



Exercício 1

- Em uma empresa, vários projetos são realizados. Os empregados da empresa trabalham em pelos menos um projeto. Há um sistema implantado na empresa com várias funcionalidades para gerenciar os projetos. Uma delas permite aos participantes de um determinado projeto computarem suas horas de trabalho nos projetos. É possível também os gerentes de projeto consultar com quem estão distribuídas as horas trabalhadas em um projeto. Esse sistema também permite ao funcionário administrativo (que também acumula as funcionalidades do gerente), no fim do mês, gerar os relatórios com os totais de horas trabalhadas de cada participante. Para os gerentes, o sistema possui um módulo de manutenção de projeto que pode utilizar as funcionalidades de incluir, alterar e excluir um projeto. Na inclusão, o sistema verifica a existência do projeto.

Exercício 1

- Com base na descrição do slide anterior, fazer o seguinte:
 - Uma lista com os atores que você definiria para esse sistema.
 - Uma lista com os casos de uso que você definiria para esse sistema.
 - Desenhar o diagrama de caso de uso completo com todos os relacionamentos entre os atores e casos de usos identificados.
 - Usar relacionamentos de inclusão, extensão e generalização

Exercício 2

- Estruturar o diagrama de casos de uso do projeto da fábrica (ou estudo de caso!)

Exercício 3

- Estruturar o diagrama de casos de uso do sistema de matrícula (moodle)

REFERÊNCIAS

- [] Applying Use Cases: A Practical Guide, Geri Schneider e Jason P. Winters, Addison-Wesley, 1998.
- [] Chung, L., Nixon, B. A., Yu, E., Mylopoulos, J.: Non-Funcional Requirements in Software Engineering, Kluwer Academic Publishers, 2000, ISBN 0-7923-8666-3.
- [] CMMI, Capability Maturity Model® Integration, disponível em <http://www.sei.cmu.edu/cmmi>
- [] IEEE Computer Society, disponível em <http://www.ieee.org/portal/site>
- [] IFPUG: Function Point Counting Practices Manual, Versão 4.1.1. International Function Point Users Group <http://www.ifpug.org> (2000)
- [] Modelo de Referência Mps.Br (Melhoria do Processo de Software Brasileiro) – Guia Geral
- [] Neighbors J. Software Construction Using Components. Tese Doutorado) - Universidade da Califórnia, Irvine, EUA. 1981.
- [] Requirements Engineering Processes and Techniques, Gerald Kotonya, Ian Sommerville Wiley, 1998.
- [] RUP, Rational Unified Process 2002
- [] SEI, Software Engineering Institute, disponível em <http://www.sei.cmu.edu/>
- [] Software Engineering, Ian Sommerville, Addison-Wesley, 1996.
- [] SWEBOK, Guide to the Software Engineering Body of Knowledge
- [] The Unified Modeling Language Reference Manual, James Rumbaugh, Ivar Jacobson e Grady Booch, Addison-Wesley, 1999
- [] The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh e Ivar Jacobson, Addison-Wesley, 1999.
- [] The Unified Software Development Process, Ivar Jacobson, Grady Booch e James Rumbaugh, Addison-Wesley, 1998.
- [] Unified Modeling Language Superstructure Especification, 2.0.