

Engenharia de Software

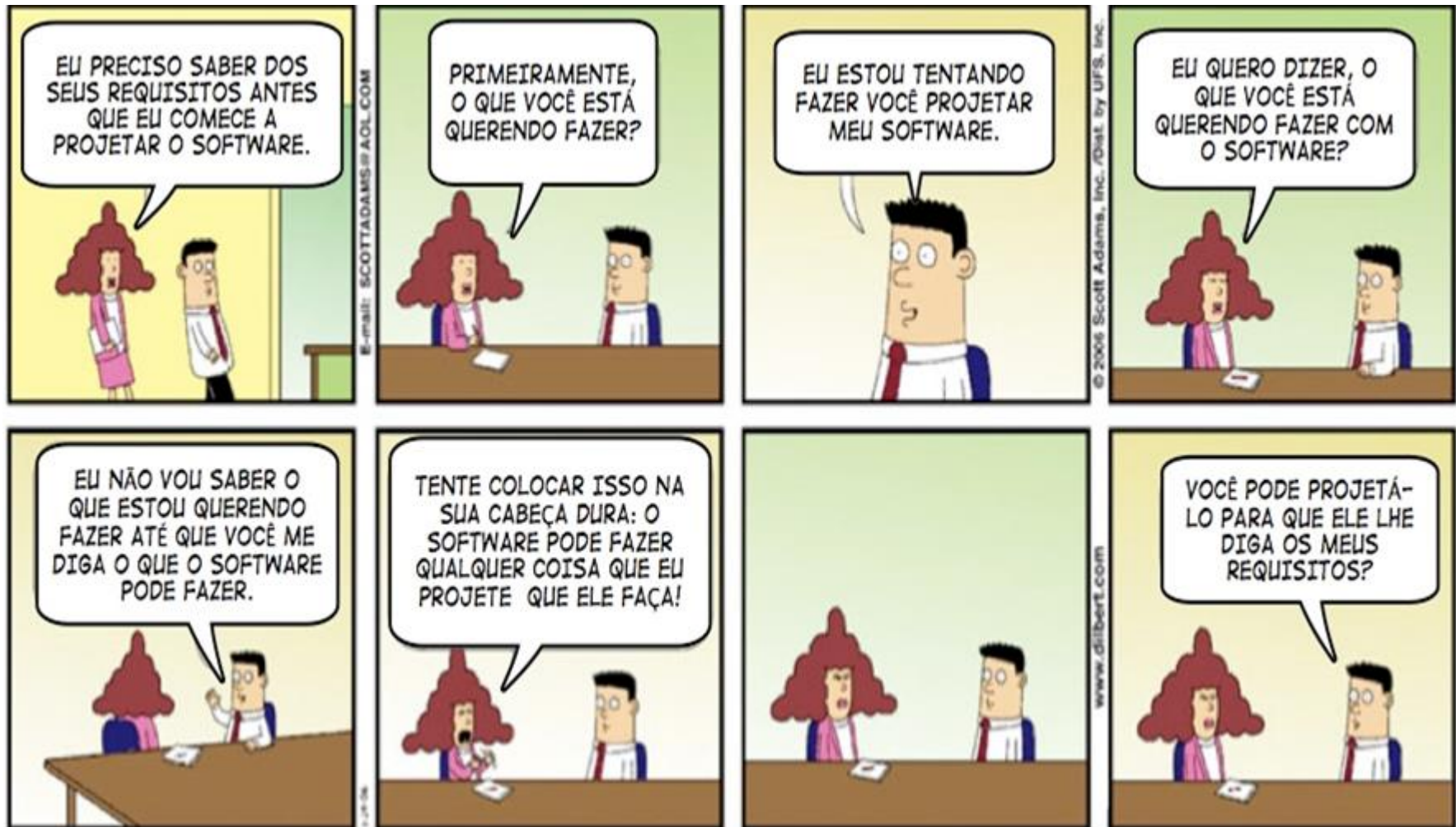
Engenharia de Requisitos

Prof^a Raquel Mini
raquelmini@ufmg.br
DEE / UFMG

Unidade III

- ❑ Engenharia de Requisitos
 - Requisitos
 - Classificação para os requisitos
 - O processo da engenharia de requisitos
 - Desenvolvimento de requisitos
 - Gerência de requisitos

Motivação



© Scott Adams, Inc./Dist. by UFS, Inc.

Cenário atual de desenvolvimento

- ❑ Gasta-se cada vez mais na manutenção e teste de sistemas
- ❑ 85% dos erros são causados por defeitos inseridos durante a análise de requisitos e projeto do sistema

Cenário atual de desenvolvimento

	% do Custo de Desenvolvimento	% dos defeitos introduzidos	% dos defeitos encontrados	Custo relativo de correção
Análise de Requisitos	5	55	18	1
Projeto	25	30	10	1 - 1.5
Códificação e teste de unidade	50	10		
Teste	10		50	1 - 5
Validação e Documentação	10			
Manutenção		5	22	10 - 100

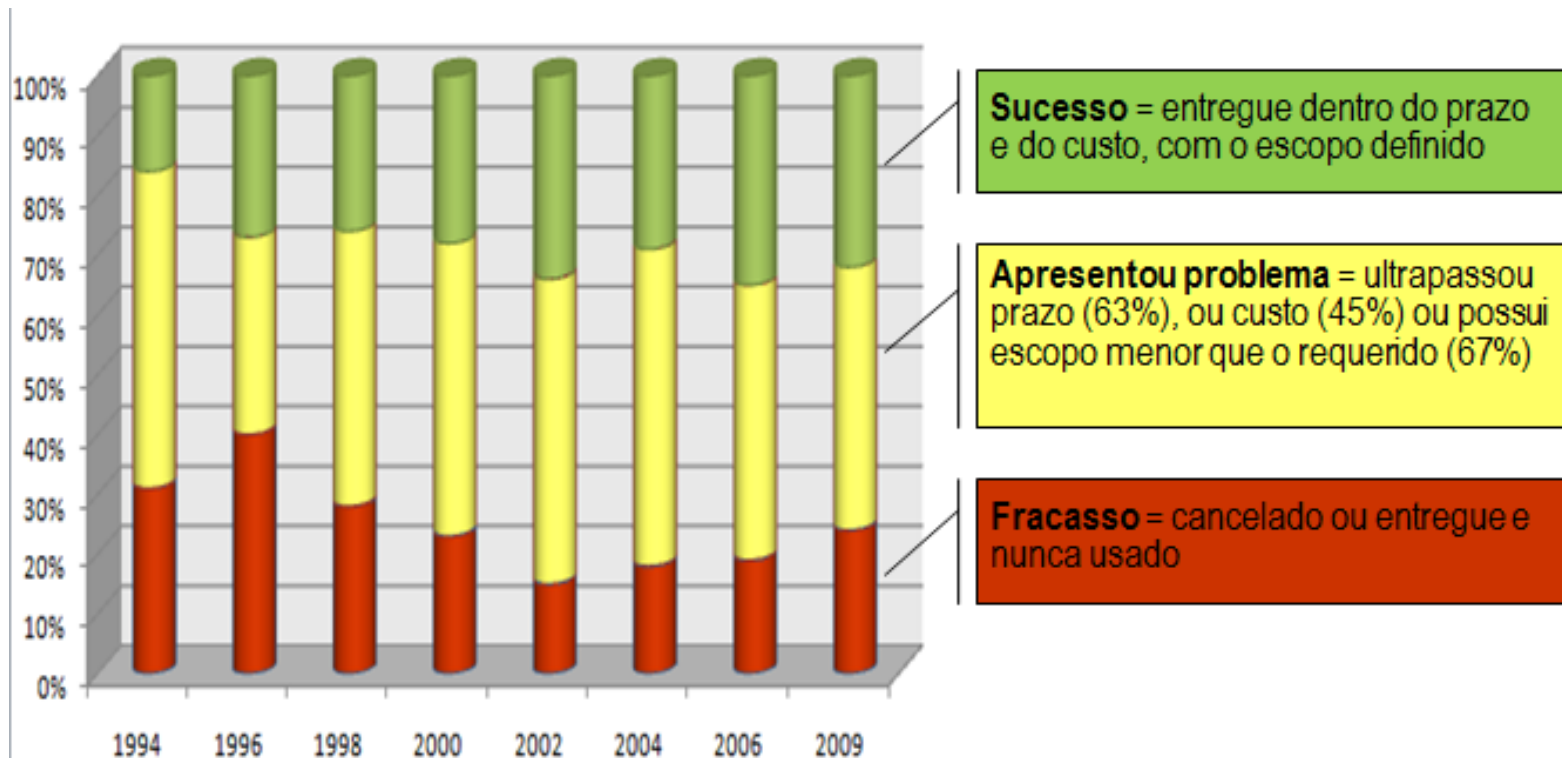
- ❑ Os erros mais caros são aqueles cometidos no processo de requisitos e descobertos pelo usuário!

Requisitos

- ❑ Descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento
- ❑ O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado de **engenharia de requisitos**

Importância dos requisitos

- ❑ Estudo feito pelo Standish Group (2009) com 350 empresas e 8.000 projetos de software



Importância dos requisitos

- ❑ Estudo feito pelo Standish Group (2009) com 350 empresas e 8.000 projetos de software
 - Sucesso (32%)
 - Cobre todas as funcionalidades em tempo e dentro do custo previsto
 - Problemático (44%)
 - Não cobre todas as funcionalidades exigidas, custo aumentado e está atrasado
 - Fracasso (24%)
 - Cancelado durante o desenvolvimento

Importância dos requisitos

❑ Por que os projetos falham?

Fatores de Projetos Críticos	% Resp.
1. Requisitos Incompletos	13,1%
2. Falta de Envolvimento do Usuário	12,4%
3. Falta de Recursos	10,6%
4. Expectativas Irreais	9,9%
5. Falta de Apoio Executivo	9,3%
6. Mudança de Requisitos e Especificações	8,7%
7. Falta de Planejamento	8,1%
8. Sistema não mais necessário	7,5%

❑ Requisitos são constantemente mencionados!

Maiores desafios no desenvolvimento de sistemas

- ❑ Compreensão do domínio do problema
- ❑ Comunicação efetiva com reais usuários do sistema
- ❑ Evolução contínua dos requisitos do sistema

Requisitos de software

- ❑ Descrição e especificação de um software ou sistema
 - O que o software deve fazer / ter?
 - O que o cliente deseja?
 - Quais as restrições?
 - Quais os componentes?
 - Quem vai utilizar o software?
- ❑ Descrevem uma condição ou capacidade com a qual o sistema deve estar em conformidade
[*Unified Process*]



Requisitos de software

- ❑ Podem ser derivados das necessidades dos usuários ou estabelecidos em um contrato, padrão, especificação ou outro documento imposto formalmente [IEEE]
- ❑ É uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir seus objetivos [Pfleeger]
- ❑ É uma propriedade que o software deve exibir para resolver algum problema no mundo real [SWEBOX]

Requisitos de software

❑ Principais problemas com requisitos

▪ Especificação em linguagem natural

- É naturalmente ambígua, o que dificulta seu uso
 - Difícil garantir que todos irão interpretar da mesma forma
 - A mesma coisa pode ser dita de várias formas diferentes, em diversos níveis de abstração
- A busca de maior precisão aumenta o texto e diminui a legibilidade
- Vários requisitos diferentes podem ser expressos juntos
- Estrutura inadequada para gerenciar requisitos

Requisitos de software

❑ Principais problemas com requisitos

▪ Imprecisão dos requisitos

- Requisitos não declarados precisamente ou ambíguos podem ser interpretados de forma diferente por desenvolvedores e usuários
- Ex: O sistema deve fornecer visualizadores apropriados para ler documentos
 - Intenção do usuário: visualizadores de propósito especial para cada tipo de documento diferente
 - Interpretação do desenvolvedor: um visualizador textual que mostra o conteúdo do documento

Requisitos de software

- ❑ Principais problemas com requisitos
 - Dificuldade para gerar uma especificação completa
 - Tempo, esforço, certeza de estar completa
 - Existência de requisitos implícitos
 - A organização pode ter requisitos não incluídos
 - Ex: facilidade de uso e manutenibilidade

Importância dos requisitos

- ❑ Uma especificação de requisitos é importante por que:
 - Estabelece uma base de concordância entre o cliente e o fornecedor sobre o que o software fará
 - Fornece uma referência para a validação do produto final
 - Uma especificação de requisitos de alta qualidade é pré-requisito para um software de alta qualidade
 - Reduz o custo de desenvolvimento

Por que precisamos dos requisitos?

Para entender o que o cliente quer

X

Para entender o problema do negócio

Para documentar o que o cliente quer

X

Para documentar o escopo do projeto e definir suas restrições

Para assegurar qualidade e a satisfação do cliente

X

Para definir critérios de aceitação e gerenciar as expectativas do cliente

Classificação para os requisitos

- ❑ Existem várias classificações para os requisitos de software:
 - Usuário e sistema
 - Funcional e não funcional
 - Essencial, importante, desejável



Requisitos de usuário e de sistema

❑ Requisitos de usuário

- Declarações em linguagem natural com diagramas dos serviços que o sistema deverá fornecer e suas restrições operacionais
- Escrito para os clientes

❑ Requisitos de sistema

- Um documento estruturado estabelecendo descrições detalhadas das funções do sistema, serviços e restrições operacionais
- Define o que deve ser implementado e, portanto, pode ser parte de um contrato entre o cliente e o desenvolvedor

Requisitos de usuário e de sistema

Definição de requisitos de usuário

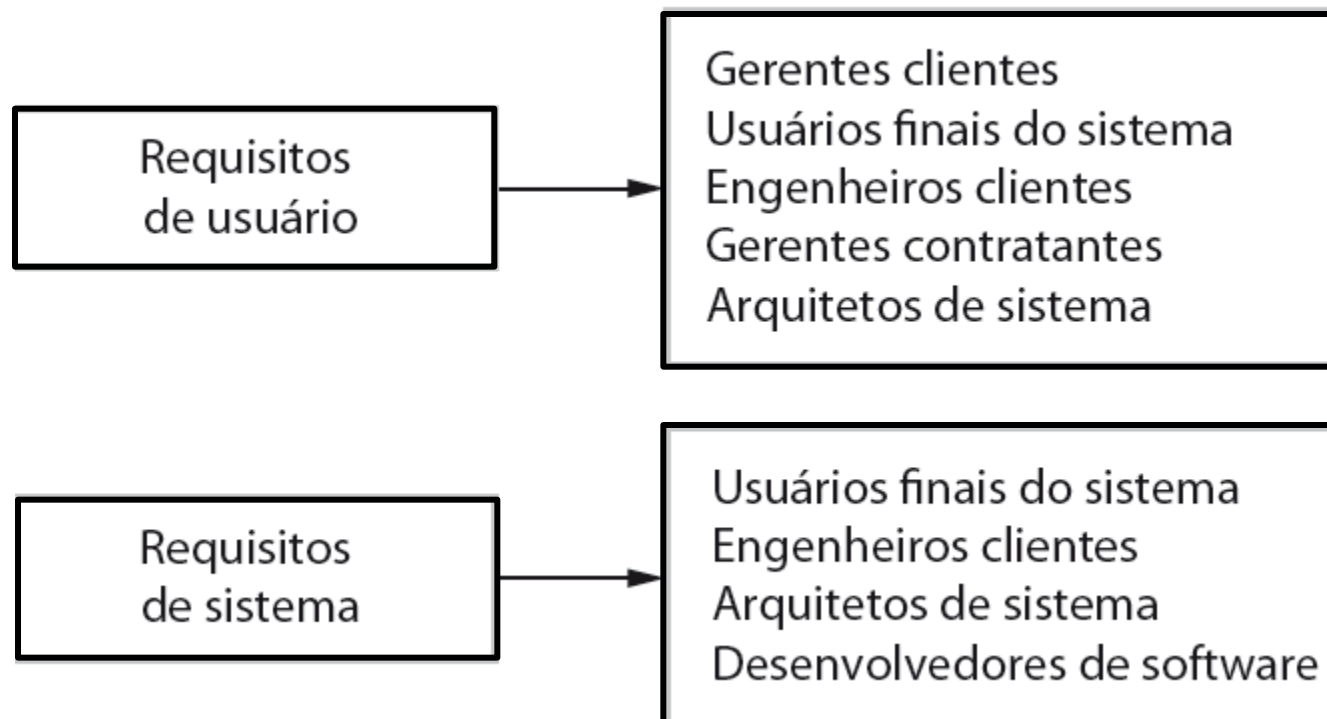
1. O MHC-PMS deve gerar relatórios gerenciais mensais que mostrem o custo dos medicamentos prescritos por cada clínica durante aquele mês.

Especificação de requisitos de sistema

- 1.1 No último dia útil de cada mês deve ser gerado um resumo dos medicamentos prescritos, seus custos e as prescrições de cada clínica.
- 1.2 Após 17:30h do último dia útil do mês, o sistema deve gerar automaticamente o relatório para impressão.
- 1.3 Um relatório será criado para cada clínica, listando os nomes dos medicamentos, o número total de prescrições, o número de doses prescritas e o custo total dos medicamentos prescritos.
- 1.4 Se os medicamentos estão disponíveis em diferentes unidades de dosagem (por exemplo, 10 mg, 20 mg), devem ser criados relatórios separados para cada unidade.
- 1.5 O acesso aos relatórios de custos deve ser restrito a usuários autorizados por uma lista de controle de gerenciamento de acesso.

Requisitos de usuário e de sistema

- ❑ Leitores de diferentes tipos de especificação de requisitos



Requisitos funcionais e não funcionais

❑ Requisitos funcionais

- Declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações

❑ Requisitos não-funcionais

- Restrições aos serviços ou funções oferecidos pelo sistema, tais como restrições de tempo e de plataforma de desenvolvimento
- Muitas vezes se aplica ao sistema como um todo ao invés de características individuais ou serviços

Requisitos funcionais

- ❑ Definem os serviços que o sistema deve fornecer, ou seja, a funcionalidade do produto ou serviço do software
- ❑ Diretamente ligados a comportamentos do software
- ❑ Descrevem as funções que o software deve executar
- ❑ Descrevem uma interação entre o sistema e seu ambiente

Requisitos funcionais

❑ Exemplo:

- O software deve permitir que o atendente consulte o relatório com os resultados dos testes clínicos de um paciente
- O sistema deve fornecer visualizadores apropriados para ler documentos
- Para cada pedido deve ser alocado um identificador único

Exemplos de requisitos funcionais



ProxGer
(Sistema PDV)

- ❑ [RF01] O software deve permitir que o caixa efetue a devolução de itens vendidos
- ❑ [RF02] O software deve permitir que o caixa efetue o registro de itens vendidos
- ❑ [RF03] O software deve permitir que o administrador gere um relatório de vendas do mês

Exemplos de requisitos funcionais

- ❑ Sistema de informação de pacientes para cuidados com saúde mental (MHC-PMS)
 - [RF01] Um usuário deve ser capaz de pesquisar as listas de agendamentos para todas as clínicas
 - [RF02] O sistema deve gerar, a cada dia, para cada clínica, uma lista de pacientes esperados para as consultas daquele dia
 - [RF03] Cada membro da equipe que usa o sistema deve ser exclusivamente identificado pelo seu número de funcionário de 8 dígitos

Imprecisão de requisitos

- ❑ Problemas surgem quando os requisitos não são precisamente definidos
- ❑ Requisitos ambíguos podem ser interpretados de maneiras diferentes por desenvolvedores e usuários.
 - Considere o termo “pesquisa” no requisito 1
 - A intenção do usuário: busca pelo nome de um paciente em todas as consultas em todas as clínicas
 - Interpretação do desenvolvedor: busca pelo nome de um paciente em uma clínica, o usuário escolhe a clínica e em seguida pesquisa

Integridade e consistência dos requisitos

- ❑ Em princípio, os requisitos devem ser completos e consistentes
 - Completos
 - Eles devem incluir descrições de todos os serviços necessários
 - Consistentes
 - Não devem haver conflitos ou contradições nas descrições dos recursos do sistema

Exercícios

1. Avalie os requisitos seguintes, de diferentes sistemas, quanto à qualidade. Caso algum deles possua deficiências, especifique o tipo de deficiência e o reescreva fazendo as correções necessárias
 - a) Eu preciso de um software que imprima 4 conjuntos de números, sendo que cada conjunto deverá ter 5 números no intervalo 1 a 60.
 - b) O produto deve fornecer mensagens de status em intervalos regulares não maiores que 60 segundos.
 - c) O produto deve permitir esconder instantaneamente caracteres que não podem ser impressos em relatórios.

Exercícios

2. Pense em três requisitos funcionais para sistemas a serem desenvolvidos para os seguintes domínios:
 - Vídeo locadora
 - Sistema de caixa de auto atendimento de um sistema bancário
 - Apoio inteligente à análise de risco para bolsa de valores

Vídeo locadora

- ❑ [RF01] O software deve permitir que o administrador efetue o cadastro de clientes
- ❑ [RF02] O software deve permitir que o administrador efetue o cadastro de DVDs
- ❑ [RF03] O software deve permitir que o atendente efetue o registro de DVDs alocados

Sistema de caixa de auto atendimento de um sistema bancário

- ❑ [RF01] O software deve permitir que o cliente consulte seu extrato
- ❑ [RF02] O software deve permitir que o cliente efetue saque
- ❑ [RF03] O software deve permitir que o cliente efetue o pagamento da fatura do cartão de crédito

Apoio inteligente à análise de risco para bolsa de valores



- ❑ O domínio da aplicação pode dificultar (e muito) o trabalho de definição e especificação dos requisitos

Exercícios

3. Descubra ambiguidades ou omissões nas seguintes declarações de requisitos para parte de um sistema de emissão de bilhetes:

Um sistema automatizado para emitir bilhetes vende bilhetes de trem. Os usuários selecionam seu destino e inserem um cartão de crédito e um número de identificação pessoal. O bilhete é emitido, e sua conta de cartão de crédito, cobrada. Quando o usuário pressiona o botão de início, é ativado um *display* de menu de destinos possíveis, junto com uma mensagem ao usuário para selecionar um destino. Uma vez que o destino tenha sido selecionado, os usuários são convidados a inserir seu cartão de crédito. Sua validade é verificada e, em seguida, é solicitada ao usuário a entrada de um identificador pessoal. Quando a operação de crédito é validade, o bilhete é emitido.

Requisitos não funcionais

- ❑ Definem as propriedades e as restrições do sistema
 - Por exemplo: confiabilidade, tempo de resposta e ocupação de área
- ❑ As restrições são capacidades de dispositivos de E/S, as representações do sistema, etc.
- ❑ Podem ser mais críticos do que os requisitos funcionais
 - Se esses não forem atendidos, o sistema pode ser inútil

Requisitos não funcionais

- ❑ Restringem a solução ou o processo de desenvolvimento
- ❑ Não confundir com regra de negócio (requisito de domínio)
 - Ex: o limite de empréstimo na biblioteca é de 5 livros

Classificação dos requisitos não funcionais

❑ Requisitos de produto

- Especificam que o produto entregue tem que se comportar de um modo em particular
 - Ex: velocidade de execução, confiabilidade, usabilidade

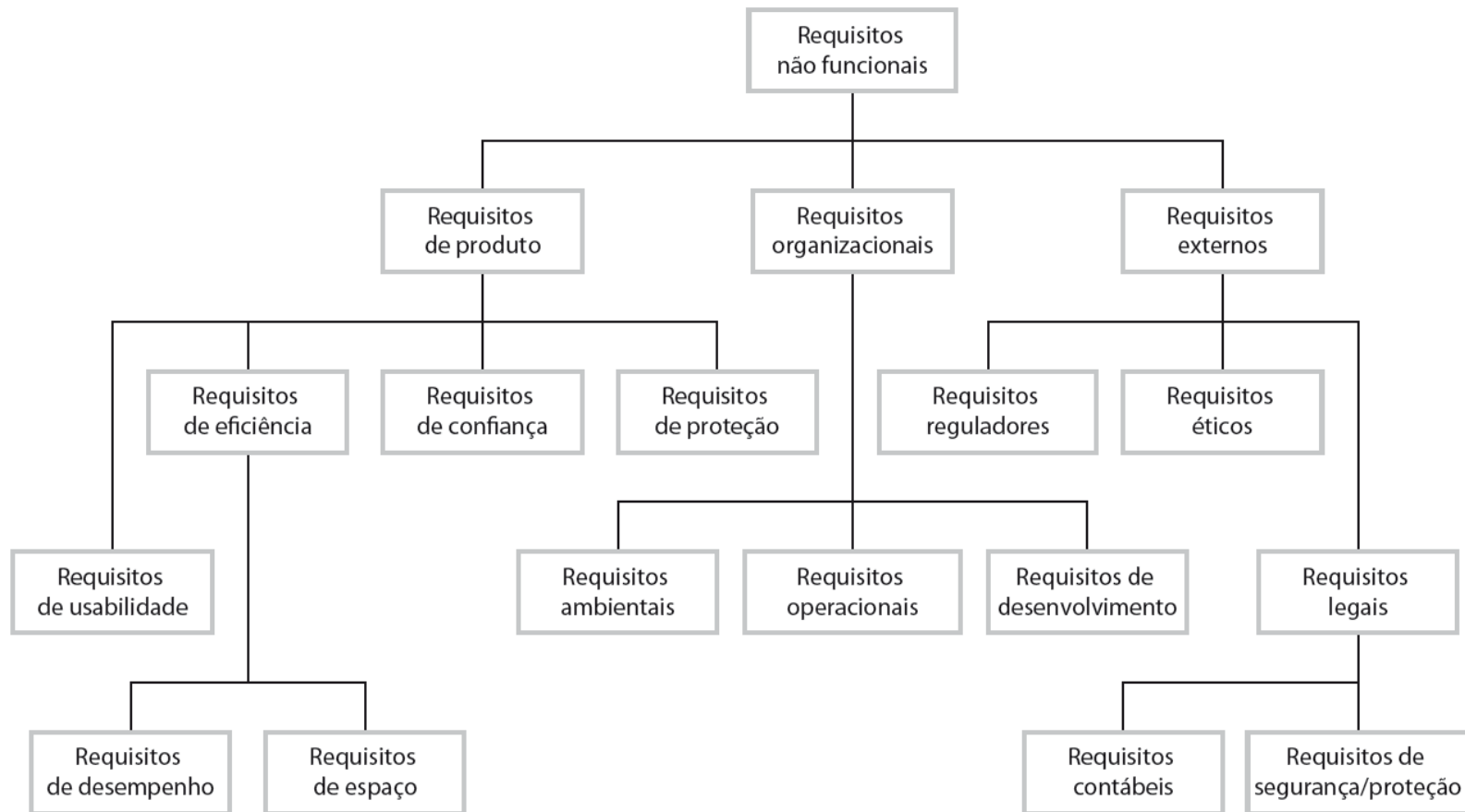
❑ Requisitos organizacionais

- Requisitos que são uma consequência de políticas e procedimentos organizacionais
 - Ex: padrões de processos usados, requisitos de implementação

❑ Requisitos externos

- Surgem de fatores que são externos ao sistema e ao seu processo de desenvolvimento
 - Ex: exigências de interoperabilidade, requisitos legislativos

Requisitos não funcionais



Requisitos não funcionais são os que mais impactam na arquitetura de uma solução e devem ser considerados desde o início !!

Implementação de requisitos não funcionais

- ❑ Requisitos não funcionais podem afetar a arquitetura geral de um sistema, em vez de componentes individuais
 - Por exemplo, para assegurar que os requisitos de desempenho sejam cumpridos, você pode ter que organizar o sistema para minimizar a comunicação entre os componentes
- ❑ Um único requisito não funcional, como um requisito de proteção, pode gerar uma série de requisitos funcionais relacionados que definem os serviços do sistema que são necessários
- ❑ Ele também pode gerar requisitos que restringem os requisitos existentes

Exemplo de requisitos não funcionais

❑ Sistema de informação de pacientes para cuidados com saúde mental (MHC-PMS)

Requisito de produto

O MHC-PMS deve estar disponível para todas as clínicas durante as horas normais de trabalho (segunda a sexta-feira, 8h30 às 17h30). Períodos de não operação dentro do horário normal de trabalho não podem exceder cinco segundos em um dia.

Requisito organizacional

Usuários do sistema MHC-PMS devem se autenticar com seus cartões de identificação da autoridade da saúde.

Requisito externo

O sistema deve implementar as disposições de privacidade dos pacientes, tal como estabelecido no HStan-03-2006-priv.

Metas e requisitos

- ❑ Requisitos não-funcionais podem ser muito difíceis de se definir precisamente e requisitos imprecisos podem ser difíceis de se verificar
- ❑ Metas
 - A intenção geral do usuário, facilmente usável
- ❑ Requisito não funcional mensurável
 - Uma declaração usando alguma métrica que pode ser objetivamente testada
- ❑ Metas são úteis para desenvolvedores quando exprimem as intenções dos usuários do sistema

Requisitos de Usabilidade

❑ Não testável:

- O sistema deve ser de fácil uso pelo pessoal médico e deve ser organizado de tal forma que os erros dos usuários sejam minimizados

❑ Testável:

- A equipe médica deve ser capaz de usar todas as funções do sistema depois de quatro horas de treinamento
- Após esse treinamento, o número médio de erros cometidos pelos usuários experientes não deve exceder dois por hora de uso do sistema

Métricas para especificar requisitos não funcionais

Propriedade	Medida
Velocidade	Transações processadas/segundo Tempo de resposta de usuário/evento Tempo de atualização de tela
Tamanho	Megabytes Número de chips de memória ROM
Facilidade de uso	Tempo de treinamento Número de <i>frames</i> de ajuda
Confiabilidade	Tempo médio para falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício após falha Percentual de eventos que causam falhas Probabilidade de corrupção de dados em caso de falha
Portabilidade	Percentual de declarações dependentes do sistema-alvo Número de sistemas-alvo

Exercícios

4. Para cada caso identifique a classificação correta dos requisitos. No caso de ser um requisito não funcional identifique o seu tipo.

Sistema de Gerenciamento de Locadora

- a) O sistema deve cadastrar o cliente.
- b) O sistema deve emitir um recibo para o cliente.
- c) O sistema deve transformar uma fita disponível em fita emprestada, quando a fita for alugada pelo cliente.
- d) O sistema deve cadastrar o cliente rapidamente, em menos de 2 minutos.
- e) O sistema deve emitir um recibo para o cliente, com o tempo máximo de 8 segundos após a transação.
- f) O sistema deve atender as normas do padrão IEEE.
- g) O sistema não pode perder dados do cliente.

Exercícios

5. Escreva um conjunto de requisitos não funcionais para o sistema de emissão de bilhetes, definindo sua confiabilidade e tempo de resposta esperados.

Engenharia de requisitos

- ❑ Processo de descobrir, analisar, documentar e verificar as funções e restrições do sistema
- ❑ Termo usado para descrever as atividades relacionadas à investigação e definição de escopo de um sistema de software

Engenharia de requisitos

- ❑ Estabelece serviços requeridos e restrições para o desenvolvimento e a operação do sistema
 - Particularmente importante no processo de software
 - Erros neste estágio geram problemas posteriores no projeto e implementação do sistema
 - Resultado: especificação do sistema
 - Usuários finais e cliente → em mais alto nível
 - Desenvolvedores → em um nível mais detalhado

Engenharia de requisitos

- ❑ Identificar objetivos de negócio
 - Por que desenvolver algo?
- ❑ Identificar os interessados (*stakeholders*)
 - Quem está envolvido?
- ❑ Obter diferentes pontos de vista
 - Com que os interessados estão preocupados?
 - Existem conflitos?
- ❑ Resolver conflitos (caso existam)
- ❑ Identificar cenários
 - Quais resultados as pessoas desejam?

O processo da engenharia de requisitos

- ❑ Os processos da engenharia de requisitos variam muito, dependendo do domínio da aplicação, das pessoas envolvidas e da organização que desenvolve os requisitos
- ❑ No entanto, existe uma série de atividades genéricas comuns a todos os processos:
 - Estudo da viabilidade
 - Elicitação e análise de requisitos
 - Especificação de requisitos
 - Validação de requisitos
 - Gerência de requisitos
- ❑ Na prática, engenharia de requisitos é uma atividade iterativa em que estes processos são intercalados

O processo da engenharia de requisitos

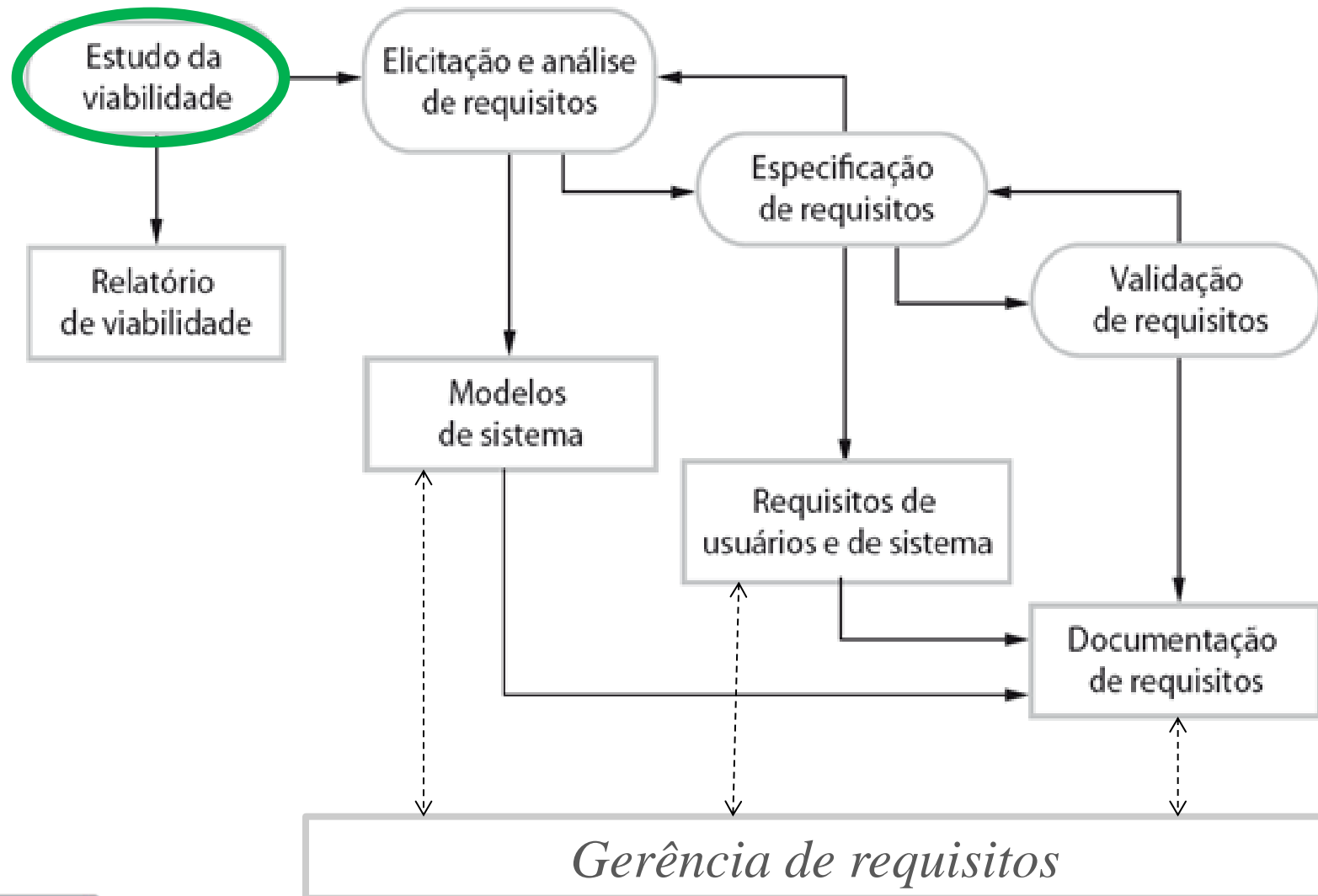
❑ Desenvolvimento de requisitos

1. Estudo de viabilidade
2. Elicitação (levantamento) e análise de requisitos
3. Especificação de requisitos
4. Verificação de requisitos

❑ Gerência de requisitos

1. Entendimento e validação junto a fornecedores de requisitos
2. Manutenção da rastreabilidade (entre requisitos e artefatos)
3. Documentação e controle de mudanças

O processo da engenharia de requisitos



Estudo da viabilidade

❑ Entendimento do problema

- Identificar as necessidades dos *stakeholders*
- Principais dificuldades:
 - Chegam por diversos meios: ferramenta, e-mail e telefone
 - Escritas em linguagem natural
 - Não seguem nenhuma padronização
 - Em geral estão incompletas ou com termos próprios da área
 - Necessitam contato para esclarecimento
 - Separar a identificação do problemas da solução a ser adotada para resolvê-lo
 - Atender a necessidade real dos *stakeholders*
 - Não correr o risco de resolver o problema errado



Estudo da viabilidade

❑ Objetivo

- Oferecer à gerência uma ideia mais clara se o projeto pode (deve) ser feito, se o produto final irá beneficiar seus usuários

❑ É feita uma estimativa acerca da possibilidade de se satisfazerem as necessidades do usuário usando-se tecnologias atuais de software e hardware

❑ Esse estudo deve ser relativamente barato e rápido

Estudo da viabilidade

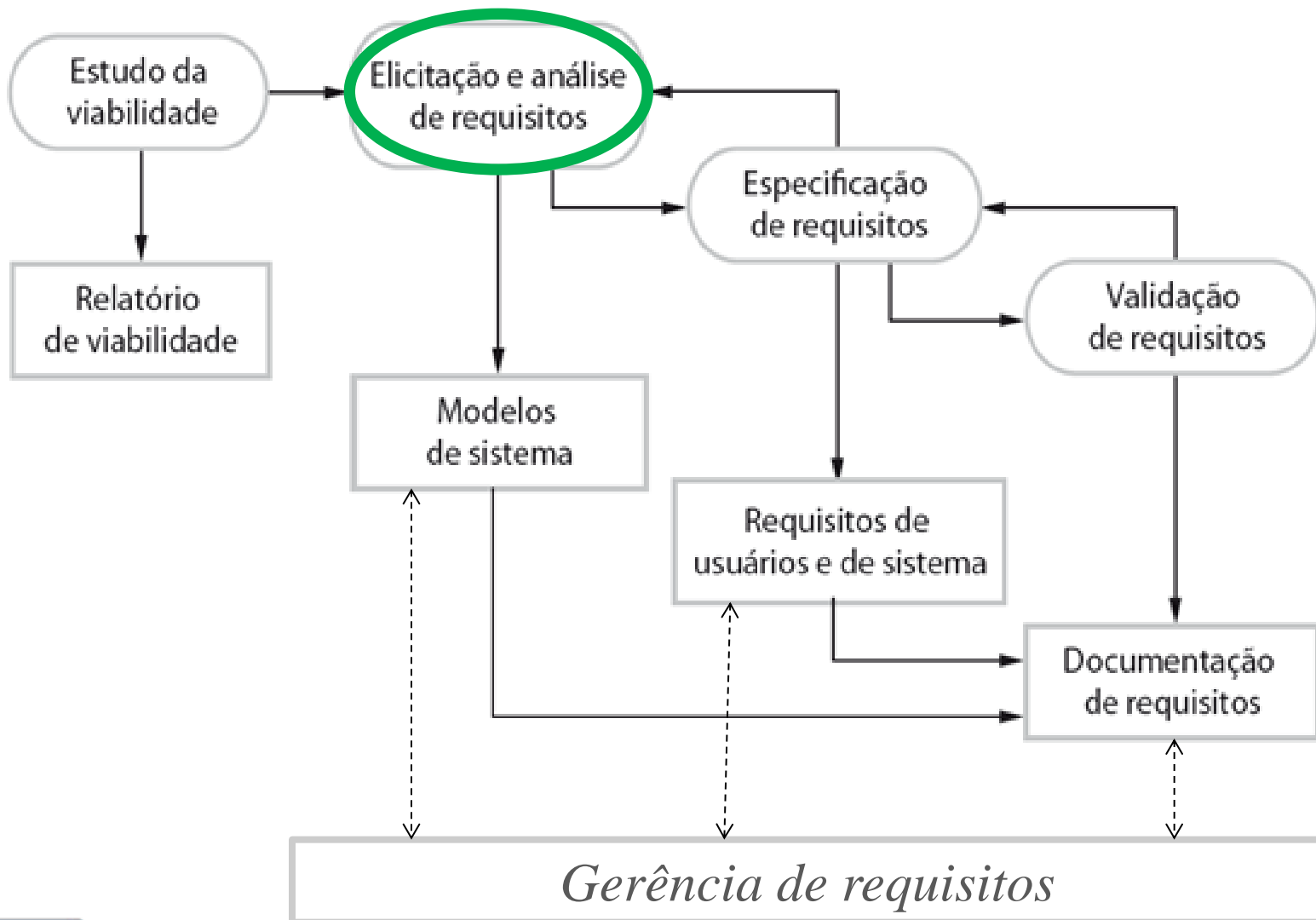
❑ O que “estudar”

- O sistema organizacional
 - Usuários, políticas, funções, objetivos, ...
- Problemas com o sistema apresentado
 - Inconsistências, funcionalidades inadequadas, performance, ...
- Objetivos e outros requisitos para o novo sistema
 - O que precisa mudar?
- Restrições existentes
 - Incluindo requisitos não funcionais (superficialmente)
- Alternativas possíveis para a solução
 - O sistema atual é geralmente uma das alternativas
 - Vantagens e desvantagens das alternativas

❑ O que concluir

- A viabilidade do projeto (decisão de continuar ou não)
- É técnica e financeiramente viável construir o sistema?

O processo da engenharia de requisitos



Elicitação e análise de requisitos

- ❑ Processo de derivação dos requisitos do sistema por meio de discussões com os interessados além de observação dos sistemas existentes
- ❑ Essa parte do processo pode envolver o desenvolvimento de um ou mais modelos de sistemas e protótipos que ajudam a entender o sistema a ser especificado
- ❑ Formação da ideia inicial do sistema

Elicitação e análise de requisitos

❑ Fontes de informação:

- Todos os interessados do sistema
- Reguladores que certificam a aceitabilidade do sistema
- Domínio da aplicação
- Outros sistemas que interagem com o sistema especificado
- Sistemas atuais

Elicitação de requisitos

- ❑ “Elicitar” (em português seria “eliciar”)
 - Extrair, explicitar: obter o máximo de informação para o conhecimento do objeto em questão [SWEBOK]
 - Buscar uma forma inteligível, para o cliente e a equipe
- ❑ Elicitação de requisitos
 - Identificar os fatos relacionados aos requisitos do sistema, para prover o entendimento mais correto e completo do que é demandado do software
 - Significa compreender:
 - O domínio da aplicação e do negócio
 - O problema específico a ser resolvido
 - As necessidades e limitações organizacionais
 - As facilidades específicas necessárias para os envolvidos

Elicitação de requisitos

❑ Problemas:

- Falta de conhecimento do usuário das suas reais necessidades
 - Usuário com vaga noção do que precisa e do que um produto de software pode oferecer
- Falta de conhecimento do analista sobre o domínio do problema
 - Desenvolver sem conhecimento adequado do domínio, o que leva a decisões erradas
- Não existir muito tempo para a elicitação

Elicitação de requisitos

❑ Problemas:

- Domínio do processo de elicitação de requisitos pelos desenvolvedores
 - Desenvolvedor não ouve o que os usuários têm a dizer e força suas próprias visões e interpretações
- Comunicação inadequada entre os desenvolvedores e usuários
 - Usuários incapazes de expressar suas necessidades apropriadamente
 - Significados diferentes a termos comuns

Elicitação de requisitos

❑ Problemas:

- Dificuldade do usuário tomar decisões
 - Falta de entendimento sobre as consequências das decisões ou as alternativas possíveis
- Problemas de comportamento
 - A elicitação de requisitos é um processo social
 - Conflitos e ambiguidades nos papéis que os usuários e desenvolvedores desempenham
- Questões técnicas
 - Complexidade crescente dos sistemas atuais

Elicitação de requisitos

❑ Problemas:

- Usuários com resistência ao novo sistema
 - Não estarem convencidos da sua necessidade
 - Não querer mudar a forma de trabalhar
 - Não participar, fornecer informações errôneas

Técnicas de elicitação de requisitos

- ❑ Entrevistas
- ❑ Observações de cenários e análise social (etnografia)
- ❑ Leitura de documentos e aplicação de questionários
- ❑ Reuso de requisitos, prototipagem do processo
- ❑ *Brainstorm*, entrevistas e reuniões de discussão

Entrevistas

- ❑ Entrevistas formais ou informais com os interessados fazem parte da maioria dos processos de engenharia de requisitos
- ❑ Tipos de entrevista
 - Entrevistas fechadas, nas quais os interessados respondem a um conjunto predefinido de perguntas
 - Entrevistas abertas, nas quais várias questões são exploradas com os interessados
- ❑ Trabalhe com os usuários e não contra eles!

Entrevistas

- ❑ São boas para obter uma compreensão global sobre:
 - O que os interessados fazem
 - Como eles podem interagir com o novo sistema
 - As dificuldades que eles enfrentam com os sistemas atuais

- ❑ Pode ser difícil elicitar conhecimento do domínio por meio de entrevistas
 - O conhecimento do domínio é tão familiar para os interessados que eles acreditam que é tão óbvio que não vale a pena mencionar

Entrevistas

❑ Entrevistar eficazmente

- Ter a mente aberta, evitar ideias pré-concebidas sobre os requisitos e estar disposto a ouvir os interessados
- Induzir os entrevistados a discutir usando uma questão trampolim, uma proposta de requisitos, ou trabalhando em conjunto em um sistema protótipo

Etnografia

- ❑ É uma técnica de observação que pode ser usada para compreender os processos operacionais e ajudar a extrair os requisitos de apoio para esses processos
- ❑ Um analista faz uma imersão no ambiente de trabalho em que o sistema será usado
 - O trabalho do dia a dia é observado e são feitas anotações sobre as tarefas reais em que os participantes estão envolvidos
 - As pessoas não precisam explicar ou articular seu trabalho

Etnografia

- ❑ Ajudam a descobrir requisitos implícitos do sistema que refletem as formas reais com que as pessoas trabalham, em vez de refletir processos formais definidos pela organização
- ❑ Estudos etnográficos têm mostrado que o trabalho geralmente é mais rico e complexo do que o sugerido pelos modelos simples de sistemas

Diretrizes para a elicitação de requisitos

- ❑ Planejar a elicitação
 - Selecionar técnicas a serem utilizadas e estabelecer de que maneira elas serão integradas
 - A escolha das técnicas e seu esquema de integração dependerá do problema e da equipe participante
- ❑ Utilizar uma técnica de modelagem de apoio
 - Representar os fatos obtidos para futuro tratamento
 - Ter conhecimento sobre estas técnicas
 - Identificar onde uma técnica é superior a outra
- ❑ Importante
 - Sempre perguntar: o que? por que(m)? como?
 - Perguntar o óbvio e organizar as respostas
 - Observar e estudar
 - Ser humilde (procurar aprender)

Cuidados na elicitação de requisitos

❑ Lista de Requisitos (em alto nível)

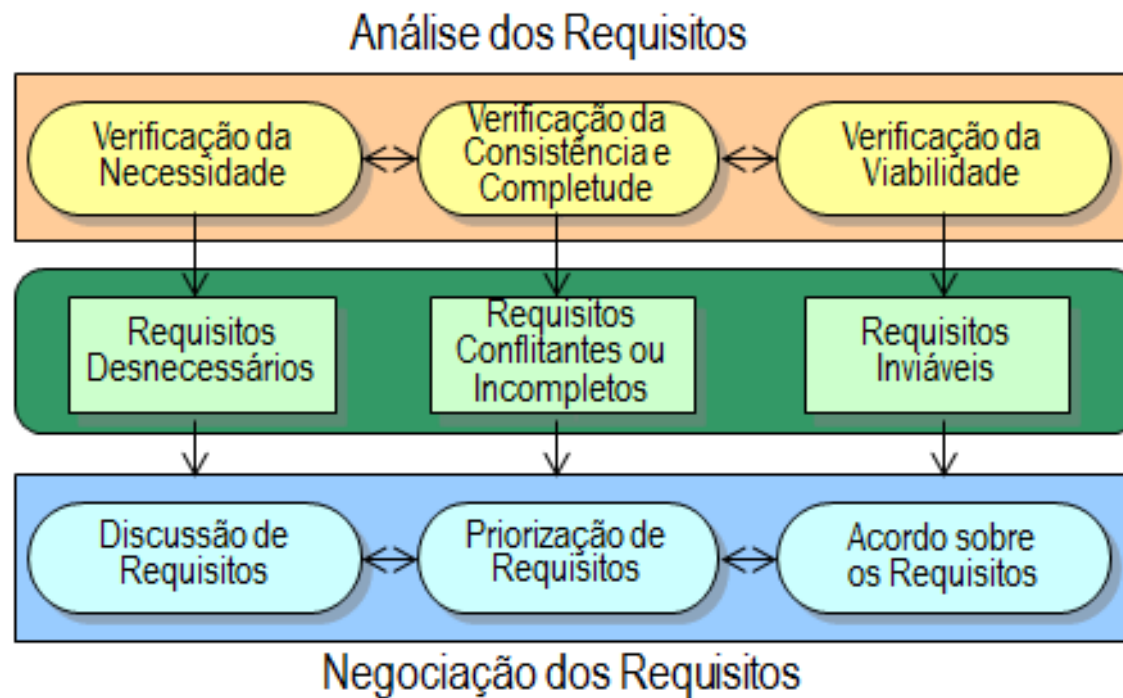
- Padronização da sintaxe {[sujeito + ação + objeto] + qualificação}
 - Exemplos:
 - “O sistema DEVE USAR protocolo TCP-IP”
 - “O sistema DEVE TRANSMITIR todas as mensagens usando o protocolo TCP/IP”
 - “O sistema DEVE EXIGIR que o usuário entre com o sobrenome de novos empregados”
 - “O sistema DEVE PERMITIR que o usuário apague registros que não tenham sido confirmados”

❑ Glossário

- Provê um conjunto consistente de definições que ajudam a evitar interpretações erradas

Análise de requisitos

- ❑ A elicitação, análise e negociação de requisitos é um processo iterativo e intercalado, devendo ser repetido várias vezes



Análise de requisito: verificações

❑ Verificação da necessidade

- O requisito é necessário?
- Contribui (está alinhado) com objetivos de negócio da organização?
- Contribui para o problema específico tratado pelo sistema?

❑ Verificação de viabilidade

- O requisito está apropriado para implementar?
- Pode ser testado (verificado, validado)?
- Os requisitos são viáveis dentro do orçamento e tempo disponíveis?

Análise de requisito: verificações

- ❑ Verificação de consistência e completude
 - A especificação do requisito possui projeto prematuro?
 - O requisito está completo? Está unicamente identificado?
 - Alguma funcionalidade ou restrição necessária foi esquecida?
 - O requisito está ambíguo? Pode ser rastreado?
 - Quais as interações entre requisitos?
 - Existem conflitos / sobreposições nestas interações?
 - Existe algum requisito contraditório?

Elicitação e análise de requisitos

❑ Principais artefatos gerados:

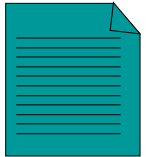
▪ Documento de Visão

- Objetivo

- Analisar e definir as necessidades e desafios do projeto

- Conteúdo

- Descrição do problema e da oportunidade de negócio
- Posicionamento do produto final resultante
- Identificação dos envolvidos (incluindo usuários)
- Resumo das principais necessidades dos envolvidos e usuários
- Alternativas internas e da concorrência
- Visão geral do produto: perspectiva, suposições, dependências, recursos, orçamento, licenciamento, entrega, instalação...
- Lista preliminar de requisitos (funcionalidades e restrições)



Visão

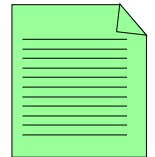
Elicitação e análise de requisitos

❑ Principais artefatos gerados:

▪ Glossário

• Objetivo

- Prover um conjunto consistente de definições para todo o sistema
- Facilitar a comunicação e evitar interpretações erradas pelos envolvidos
 - » Analistas de Sistemas: definir claramente as regras de negócio e assegurar que as especificações de requisitos façam o uso correto e consistente dos termos
 - » Desenvolvedores: projetar e implementar classes, tabelas no banco de dados, interfaces de usuário, ...
 - » Desenvolvedores de treinamento/Redatores: construir materiais de treinamento e documentação usando a terminologia compreendida pelo pessoal de negócio



Glossário

Elicitação e análise de requisitos

❑ Principais artefatos gerados:

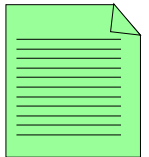
▪ Glossário

• Conteúdo

- Lista dos termos, definições e acrônimos comuns do projeto / domínio

• Verificações necessárias

- Cada termo possui uma definição clara e concisa?
- Os termos são usados de forma consistente nas especificações?
- Cada termo do Glossário aparece nos Casos de Uso?
 - » Se não – pode significar que o Caso de Uso não está completo ou o termo não é necessário

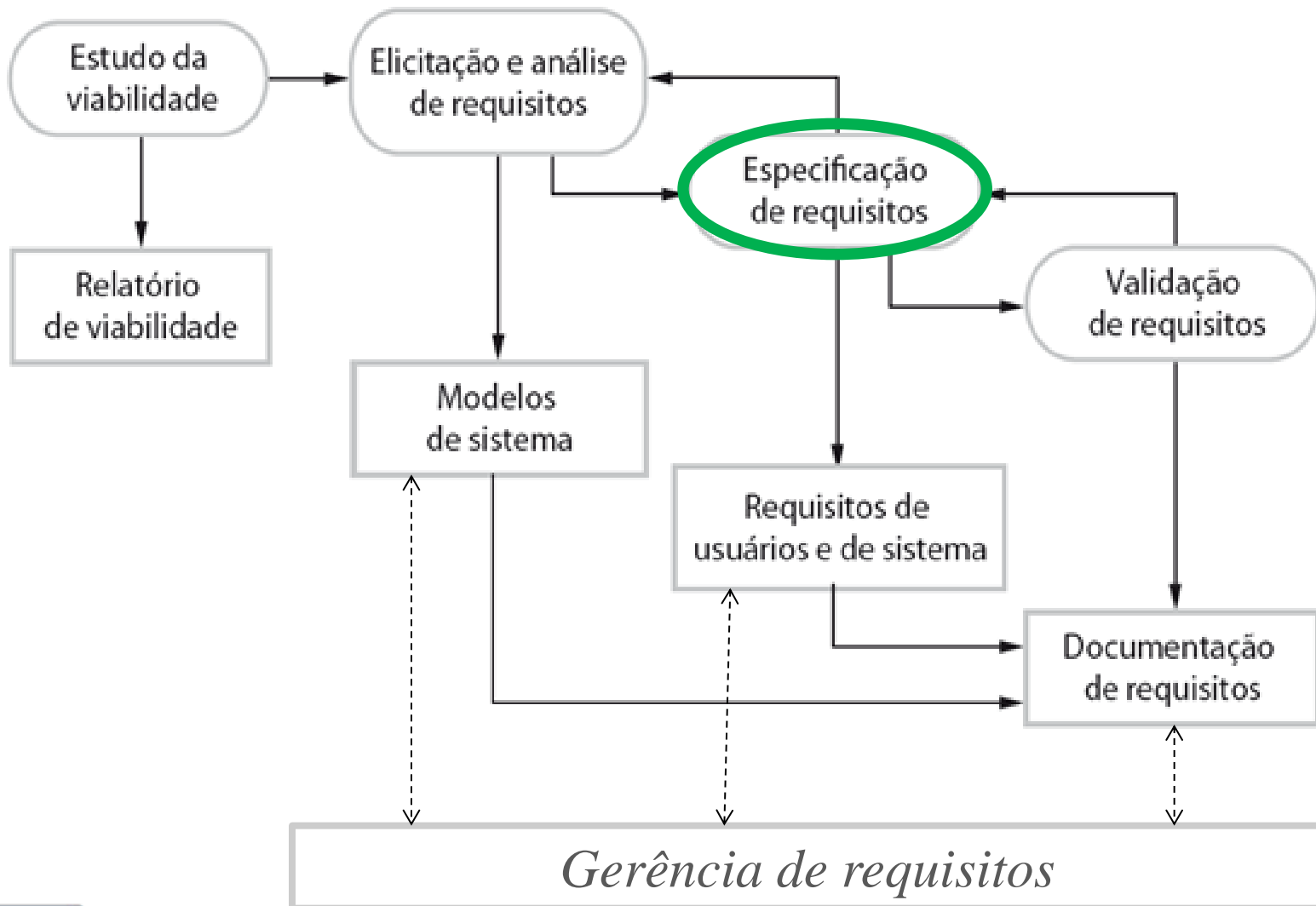


Glossário

Exercícios

6. Sugira como um engenheiro responsável pela elaboração de um sistema de especificação de requisitos pode manter o acompanhamento dos relacionamentos entre requisitos funcionais e não funcionais.

O processo da engenharia de requisitos



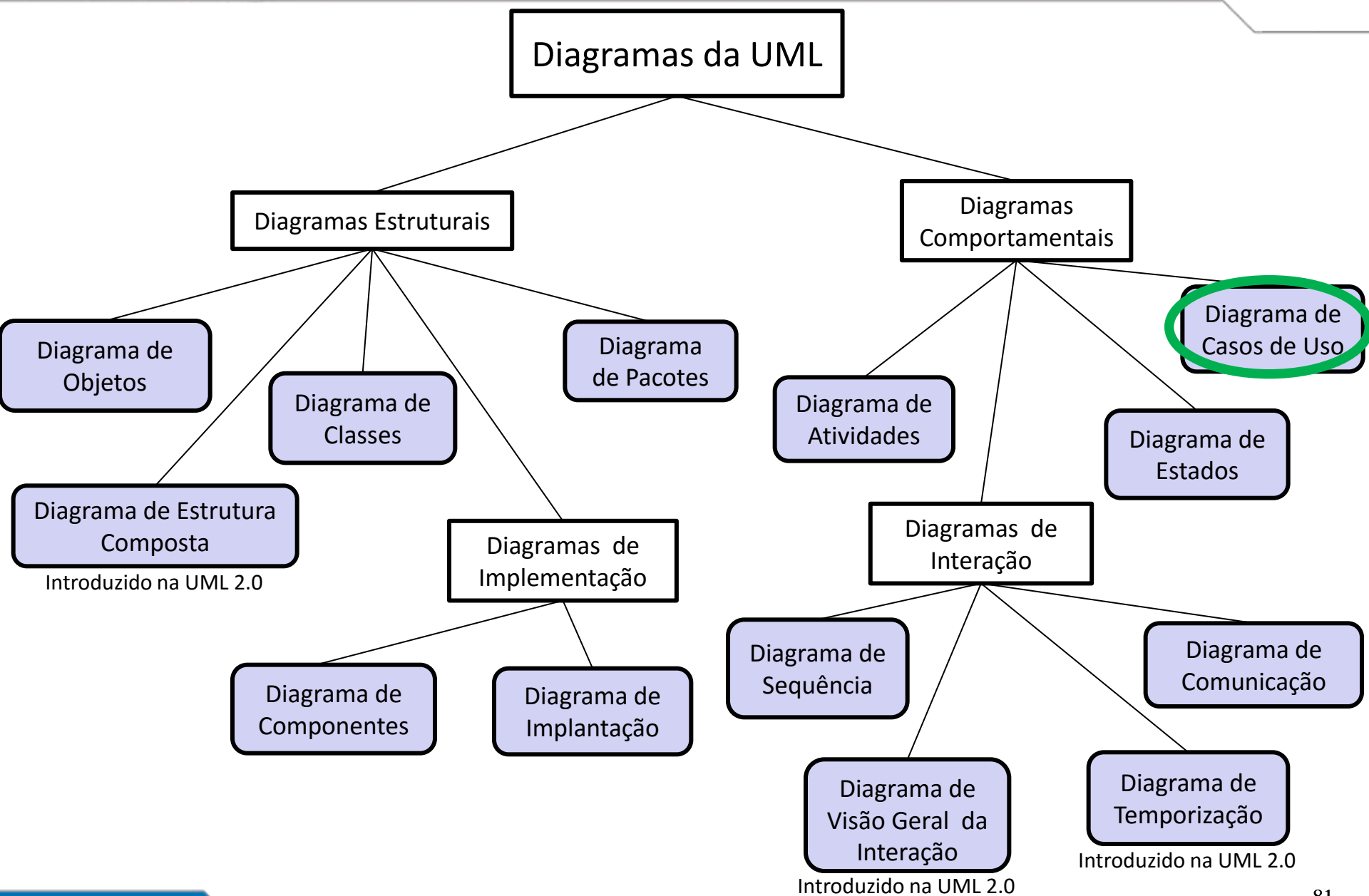
Especificação de requisitos

- ❑ É a atividade de traduzir as informações obtidas durante a elicitação e análise em um documento que defina um conjunto de requisitos
- ❑ Transformar necessidades dos usuários e características do sistema em requisitos do software, de forma clara e explícita

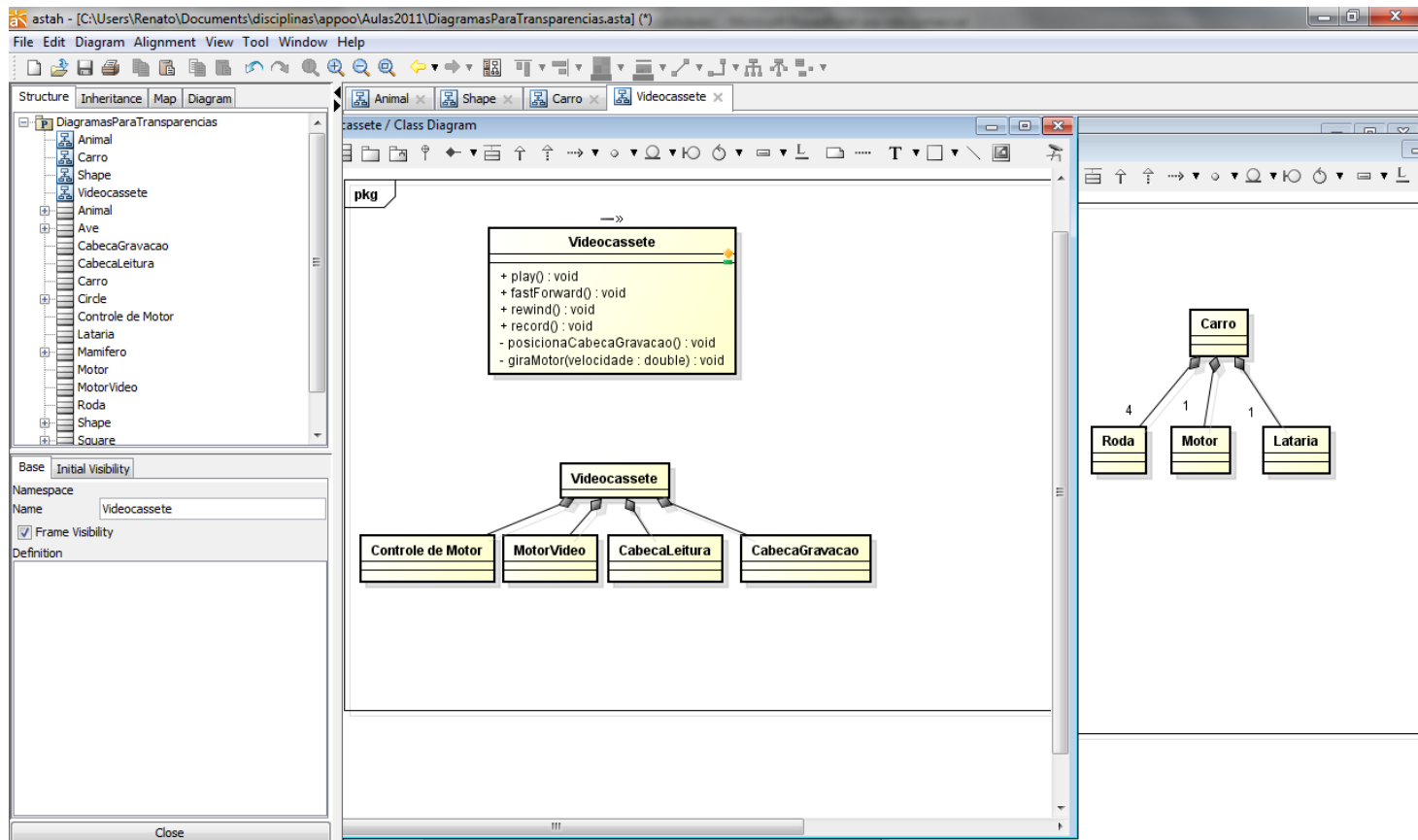
Especificação de requisitos

- ❑ Importância do documento de especificação de requisitos
 - Estabelece uma base de concordância entre o cliente e o fornecedor sobre o que o software fará
 - Clientes podem mudar de ideia \Rightarrow necessário registrar o acordado
 - Nem sempre se tem acesso direto ao cliente
 - Associado a vários outros elementos do projeto
 - Fornece uma referência para a validação do produto final
 - Uma especificação de requisitos de alta qualidade é pré-requisito para um software de alta qualidade
 - Reduz o custo de desenvolvimento





Aplicação prática - utilizando uma ferramenta de modelagem UML: Astah



Ferramenta de Modelagem Astah

❑ Downloads

- Astah Professional
 - Versão completa e com muitos recursos, para uso por equipe
 - UML2.x, *Mind Maps*, ERD, *Flowchart*, DFD, CRUD, ...
- Astah UML
 - Versão para uso pessoal
 - UML2.x e *Mind Map Modeler*
- Astah Community
 - Versão para uso educacional
 - Modelador UML 2.0 gratuito, mais simples e com menos recursos
 - ➔ <http://astah.net/download#community>
- Tutoriais Astah
 - <http://astah.net/tutorials>
 - Primeiros passos
 - <http://astah.net/resources/documents/astah-basic-operation.pdf>

Diagramas de casos de uso

- ❑ Diagrama mais geral da UML
 - Adota uma linguagem simples e acessível ao cliente
- ❑ Mostra
 - Quais usuários realizam quais funcionalidades do sistema
 - Alguns relacionamentos entre essas funcionalidades
- ❑ Objetivo é a compreensão do comportamento externo do sistema por qualquer *stakeholder*
- ❑ Permite dar uma visão global e de alto nível do sistema, sendo fundamental a definição correta da sua fronteira

Diagrama de casos de uso

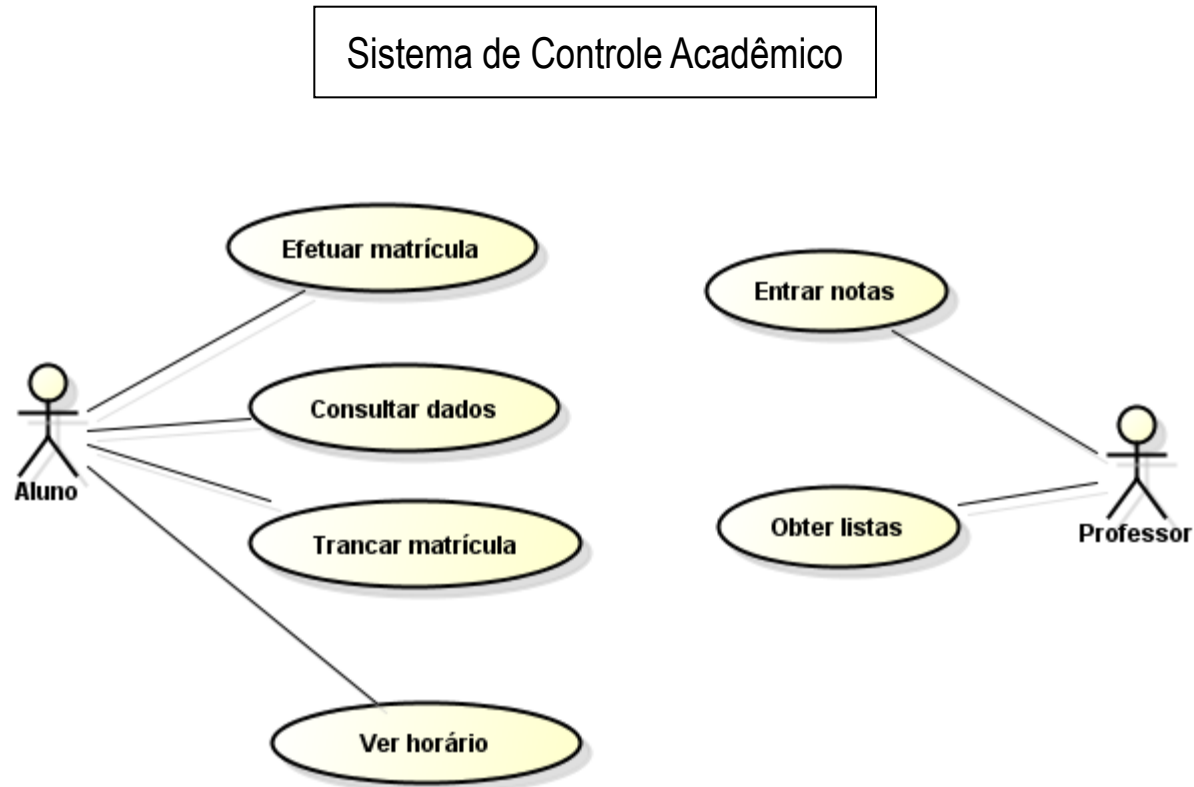
- ❑ Apresenta o sistema através de perspectivas dos usuários
- ❑ Diagrama mais abstrato da UML
 - Portanto, o mais flexível e informal
- ❑ Uma técnica de elicitação de requisitos

Diagrama de casos de uso

- ❑ Apresenta uma visão externa geral das funções e serviços do sistema
 - Define **o que** o sistema faz
 - Não se preocupa em **como** o sistema faz

- ❑ Um caso de uso indica uma funcionalidade que o sistema deve oferecer
 - Abrir conta, Sacar, Verificar Saldo, etc.

Exemplo de diagrama de casos de uso



Elementos do diagrama

- ❑ Atores
- ❑ Casos de uso
- ❑ Relacionamentos
 - Associação
 - Generalização
 - Inclusão
 - Extensão
- ❑ Fronteira do sistema

Elementos do diagrama

❑ Atores

- Quem executa a funcionalidade

❑ Casos de uso

- Qual é a funcionalidade

❑ Relacionamentos

- Como atores e casos de uso se relacionam

Elementos do diagrama

❑ Atores

❑ Casos de uso

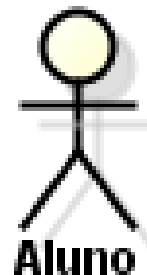
❑ Relacionamentos

- Associação
- Generalização
- Inclusão
- Extensão

❑ Fronteira do sistema

Atores

- ❑ Entidade **externa** que interage com o sistema
 - Pode ser um usuário, um sistema externo ou um dispositivo
 - Pode fazer uma solicitação de serviço ao sistema
 - Pode ser solicitado para prover um serviço e interagir com o sistema por meio de um diálogo (complexo) de requisições de serviço entre o ator e o sistema



Atores

- ❑ Representam os papéis desempenhados pelos diversos usuários
 - Cliente, caixa do banco, gerente, etc.

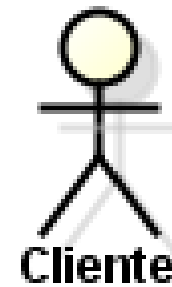
- ❑ Atores podem ser
 - Pessoas que interagem com o sistema
 - Um hardware que dispara uma interação
 - Outro software que comunica com o sistema

Definição de ator

- ❑ O ator é algo (usuário, software ou hardware) que não faz parte do sistema mas que interage com ele em algum momento

- ❑ Representação
 - Homem palito + papel desempenhado

Exemplo de atores



Elementos do diagrama

- ❑ Atores
- ❑ **Casos de uso**
- ❑ Relacionamentos
 - Associação
 - Generalização
 - Inclusão
 - Extensão
- ❑ Fronteira do sistema

Casos de uso

- ❑ Casos de uso descrevem interações entre o sistema e os atores
- ❑ Definem os serviços, tarefas ou funções do sistema
- ❑ Os nomes indicam ação (verbos)
 - Cadastrar venda
 - Sacar
 - Consultar um filme

O que é um caso de uso?

- ❑ Especifica uma sequência de ações que um sistema realiza e que produz um **resultado de valor** para um **ator**
 - É “uma forma de usar o sistema”
 - Pode descrever interação entre atores
 - Pode incluir sequências alternativas ou excepcionais (de erro)



Consultar dados

O que é um caso de uso?

- ❑ Um caso de uso **não é um dos componentes do sistema**, mas **uma funcionalidade oferecida pelo sistema** aos atores que interagem com ele!
 - Banco de dados **não é** caso de uso
 - Interface **não é** caso de uso
 - Efetuar matrícula **é** um caso de uso
 - Pagar boleto **é** um caso de uso



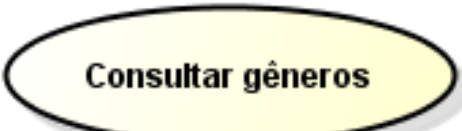
Representação de casos de uso

- ❑ Representados por elipses
 - Um texto dentro descreve a funcionalidade do caso de uso

- ❑ Geralmente a descrição dentro da elipse é curta e direta
 - Verbo [+ Objeto]



Locar filme



Consultar gêneros



Cadastrar cliente

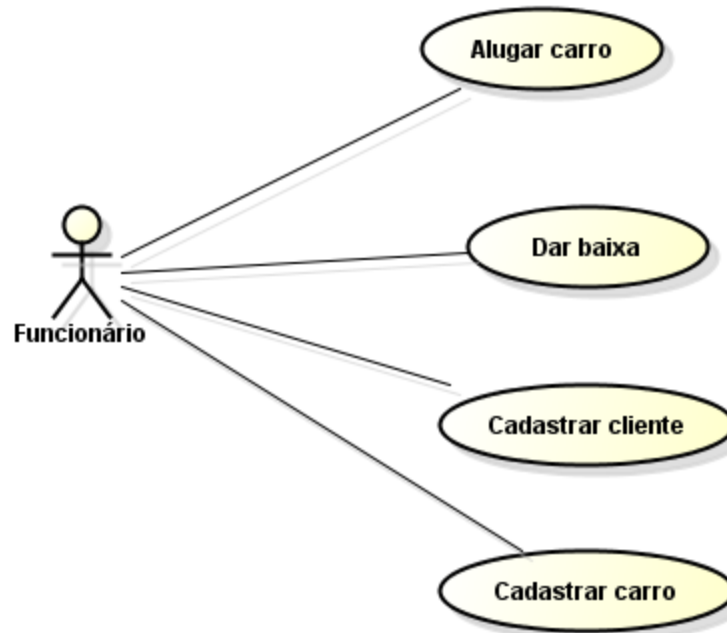
Como projetar um diagrama de casos de uso?

- ❑ Liste as funcionalidades oferecidas pelo sistema para seus atores
 - Tipicamente, cada funcionalidade se transforma em um caso de uso
- ❑ Por exemplo, no sistema de controle acadêmico
 - Um aluno pode efetuar matrícula, consultar seus dados acadêmicos, trancar matrícula, ver horário
 - Um professor pode entrar com as notas, obter listas

Exemplo: sistema de locadora de veículos

❑ Descrição do cenário

▪ Diagrama de caso de uso



▪ Especificação do caso de uso consultar cliente

Casos de uso

❑ Vantagens 😊

- Registra **requisitos funcionais** de forma textual
- É fácil de ler e fácil de rastrear
- Coleta como um objetivo pode ser atendido (sucesso) ou não (falha)
- Pode englobar diversos cenários (básico + alternativos)
 - Cada cenário mostra apenas um caminho (fluxo) específico

❑ Desvantagens 😞

- Casos de uso descrevem apenas os requisitos funcionais
- O projeto não é feito somente de incrementos dos casos de uso, pois é necessário considerar também os requisitos não funcionais

Cuidados na especificação de casos de uso

❑ Bons casos de uso ...

- Possuem textos descritivos, e não apenas descrição de interfaces
- Comunicam em uma linguagem que transcende uma especialidade
- São fáceis de ler e descrevem objetivos do usuário
- Possuem 3 a 9 passos (no máximo) no fluxo principal
- Não especificam formato de dados
- Descrevem o que o sistema deve fazer
- Registram as decisões tomadas
- Permitem checagem quanto à completeza
- Contextualizam requisitos



Cuidados na especificação de casos de uso

- ❑ Bons casos de uso **NÃO** são ...
 - Diagramas UML
 - Descrição de interfaces (GUI)
 - Descrição de formatos de dados
 - Muitas páginas descrevendo um fluxo principal
 - *Features* de um programa
 - Um tutorial sobre o domínio da aplicação



Exemplo: sistema de vendas de passagem aérea

□ Requisitos funcionais

RF1:	Sistema deve permitir o cadastro de usuários.
RF2:	Sistema deve permitir que usuário se identifique.
RF3:	Sistema deve consultar a classe de voo.
RF4:	Sistema deve consultar o trecho da viagem.
RF5:	Sistema deve permitir consulta aos aeroportos.
RF6:	Sistema deve permitir consulta as datas disponíveis de ida e volta.
RF7:	Sistema deve permitir que usuário consulte as formas de pagamento.
RF8:	Sistema deve enviar para os usuários cadastrados e-mails promocionais.
RF9:	Sistema deve permitir que usuário consulte CEP no sistema dos correios.
RF10:	Sistema deve permitir que o usuário solicite a reserva on-line.
RF11:	Sistema deve gerar código de reserva.
RF12:	Sistema deve emitir e-mail para usuário confirmando a reserva com dados.
RF13:	Sistema deve permitir que usuário cancele a reserva.
RF14:	Sistema deve permitir que o Administrador emita relatório de reservas confirmadas.
RF15:	Sistema deve permitir que o Administrador emita relatório de reservas canceladas.
RF16:	Sistema deve validar o pagamento junto com a operadora do cartão.
RF17:	Sistema deve permitir que o Administrador emita relatório de usuários cadastrados.
RF18:	Sistema deve permitir que usuário edite seus dados pessoais.

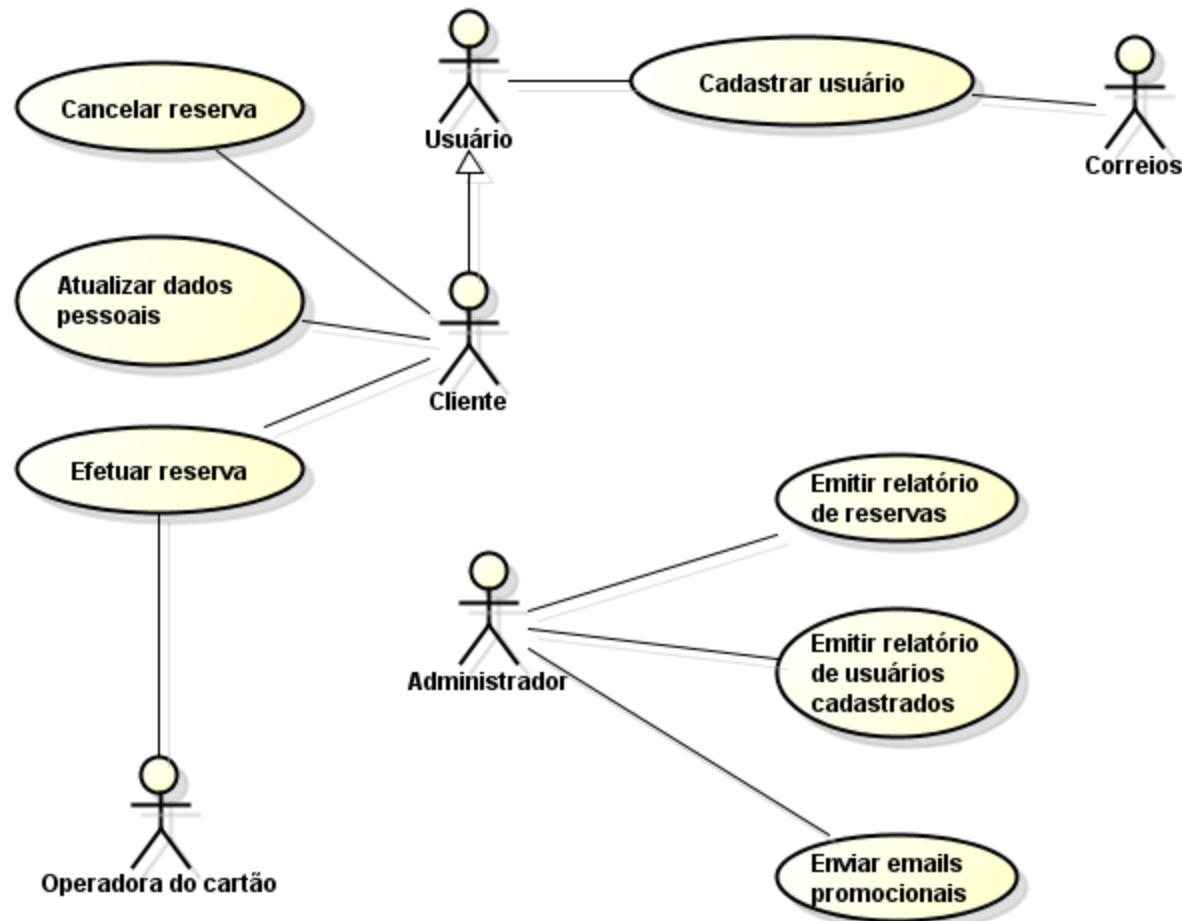
Exemplo: sistema de vendas de passagem aérea

❑ Requisitos funcionais

RF1:	Sistema deve permitir o cadastro de usuários.
RF2:	Sistema deve permitir que usuário se identifique.
RF3:	Sistema deve consultar a classe de voo.
RF4:	Sistema deve consultar o trecho da viagem.
RF5:	Sistema deve permitir consulta aos aeroportos.
RF6:	Sistema deve permitir consultar as datas disponíveis de ida e volta.
RF7:	Sistema deve permitir que usuário consulte as formas de pagamento.
RF8:	Sistema deve enviar para os usuários cadastrados e-mails promocionais.
RF9:	Sistema deve permitir que o usuário consulte CEP no sistema dos correios.
RF10:	Sistema deve permitir que o usuário solicite a reserva on-line.
RF11:	Sistema deve gerar código de reserva.
RF12:	Sistema deve emitir e-mail para usuário confirmando a reserva com dados.
RF13:	Sistema deve permitir que usuário cancele a reserva.
RF14:	Sistema deve permitir que o Administrador emita relatório de reservas confirmadas.
RF15:	Sistema deve permitir que o Administrador emita relatório de reservas canceladas.
RF16:	Sistema deve validar o pagamento junto com a operadora do cartão.
RF17:	Sistema deve permitir que o Administrador emita relatório de usuários cadastrados.
RF18:	Sistema deve permitir que usuário edite seus dados pessoais.

Exemplo: sistema de vendas de passagem aérea

❑ Diagrama de casos de uso



Elementos do diagrama

- ❑ Atores
- ❑ Casos de uso
- ❑ **Relacionamentos**
 - Associação
 - Generalização
 - Inclusão
 - Extensão
- ❑ Fronteira do sistema

Relacionamentos

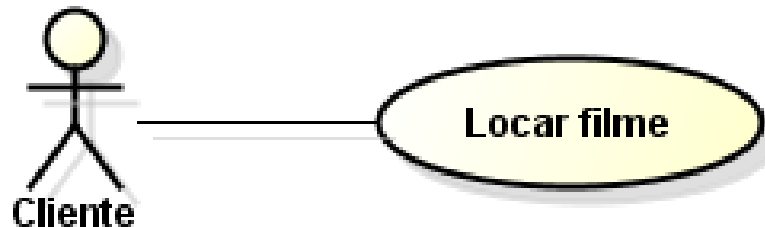
- ❑ Representam as interações ou associações entre
 - Atores e casos de uso
 - Dois ou mais casos de uso
 - Dois ou mais atores
- ❑ Principais tipos de relacionamentos
 - Associação
 - Generalização
 - Inclusão
 - Extensão

Elementos do diagrama

- ❑ Atores
- ❑ Casos de uso
- ❑ Relacionamentos
 - **Associação**
 - Generalização
 - Inclusão
 - Extensão
- ❑ Fronteira do sistema

Associação

- ❑ Demonstra que o ator utiliza a função do sistema representada pelo caso de uso
 - Requisitando a execução da função
 - Recebendo o resultado produzido pela função
- ❑ Representada por uma reta ligando o ator ao caso de uso



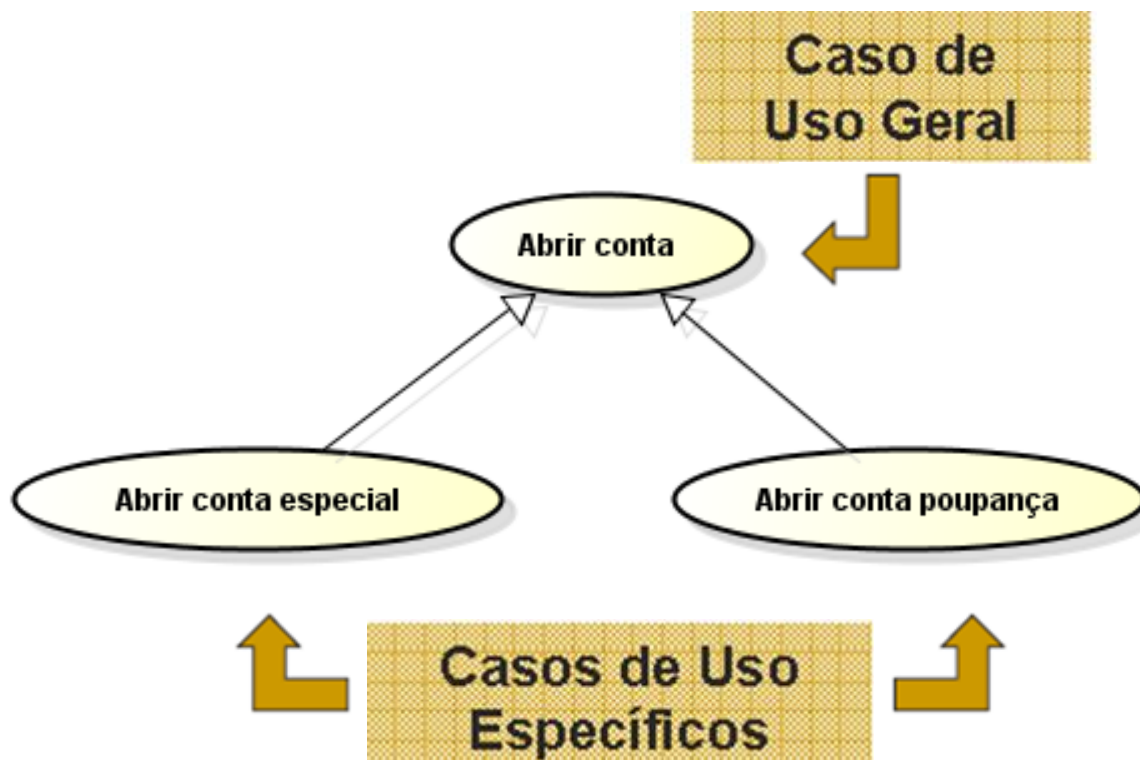
Elementos do diagrama

- ❑ Atores
- ❑ Casos de uso
- ❑ Relacionamentos
 - Associação
 - **Generalização**
 - Inclusão
 - Extensão
- ❑ Fronteira do sistema

Generalização

- ❑ O caso de uso filho herda o comportamento e o significado do caso de uso pai
 - Acontece quando dois ou mais casos de uso possuem características semelhantes
 - Reuso
- ❑ O caso de uso geral descreve as características compartilhadas
- ❑ As especializações definem características específicas

Generalização



Elementos do diagrama

- ❑ Atores
- ❑ Casos de uso
- ❑ Relacionamentos
 - Associação
 - Generalização
 - **Inclusão**
 - Extensão
- ❑ Fronteira do sistema

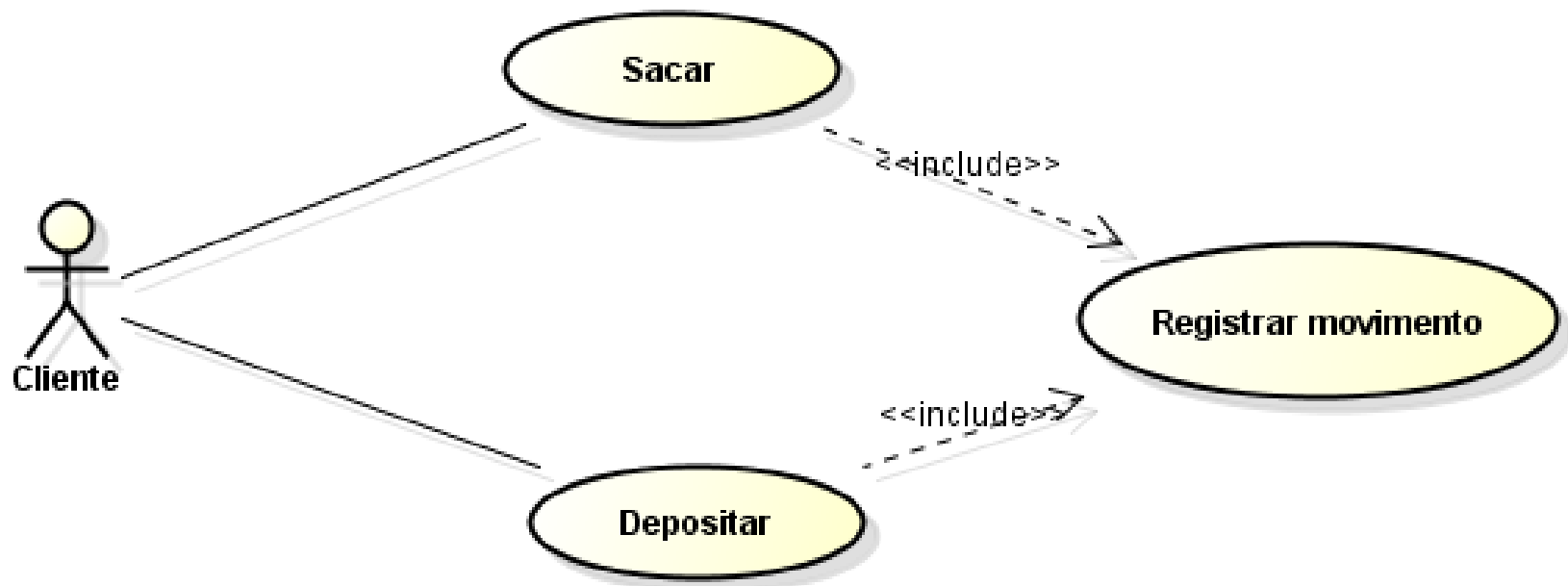
Inclusão <<include>>

- ❑ O caso de uso base incorpora explicitamente o comportamento de outro caso de uso em uma localização especificada na base
- ❑ Usamos inclusão para evitar descrever o mesmo fluxo de eventos várias vezes
- ❑ Os relacionamentos de inclusão indicam obrigatoriedade
 - A execução do primeiro obriga a execução do segundo

Inclusão <<include>>

- ❑ Representada por uma seta tracejada
 - A seta aponta para o caso de uso incluído
- ❑ Possui a palavra “*include*” entre dois sinais de menor (<<) e dois sinais de maior (>>)

Inclusão <<include>>



Elementos do diagrama

- ❑ Atores
- ❑ Casos de uso
- ❑ Relacionamentos
 - Associação
 - Generalização
 - Inclusão
 - **Extensão**
- ❑ Fronteira do sistema

Extensão <<extend>>

- ❑ Geralmente usado em funcionalidades opcionais de um caso de uso
- ❑ Exemplo: cenários que somente acontecerão em uma situação específica
 - Se uma determinada situação for satisfeita
- ❑ Extensão pode necessitar um teste para determinar se o caso de uso será estendido

Extensão

- ❑ Semelhante à inclusão
 - Possuir a palavra “*extend*” entre dois sinais de menor (<<) e dois sinais de maior (>>)

Extensão: exemplo

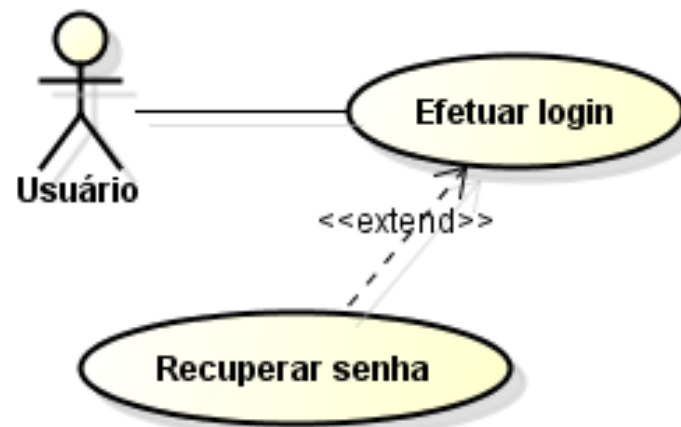
Acesse o Gmail com a sua
Conta do Google

Nome de usuário:

Senha:

☐ Continuar conectado

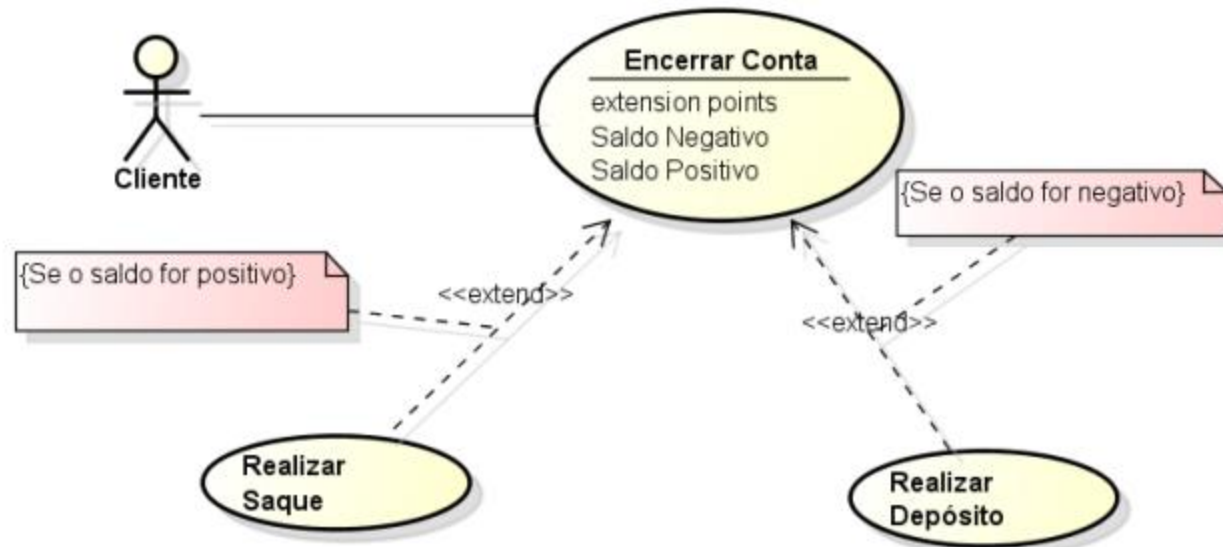
[Não consegue acessar a sua conta?](#)



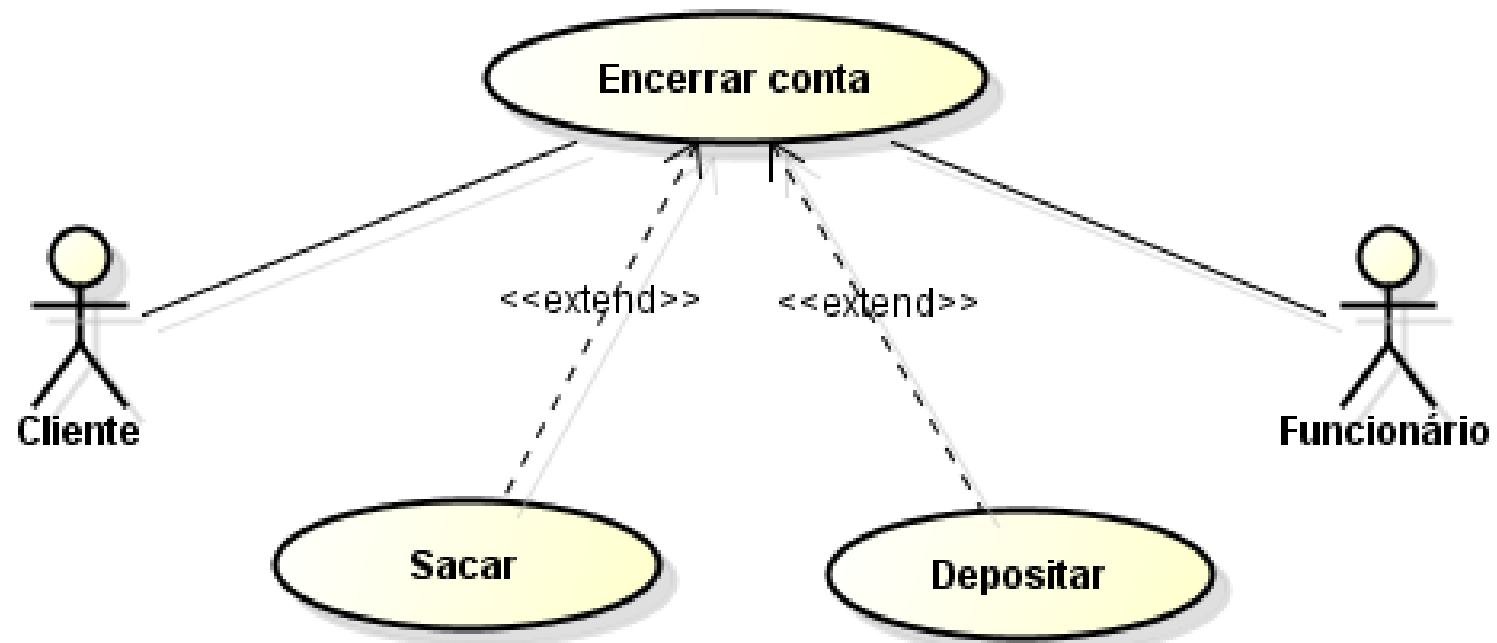
Extensão: exemplo

❑ Adicionando restrições

- Restrições podem ser representadas através de uma nota explicativa que exhibe a condição para que um caso de uso estendido seja executado.



Extensão <<extend>>



Exemplo de diagrama de casos de uso

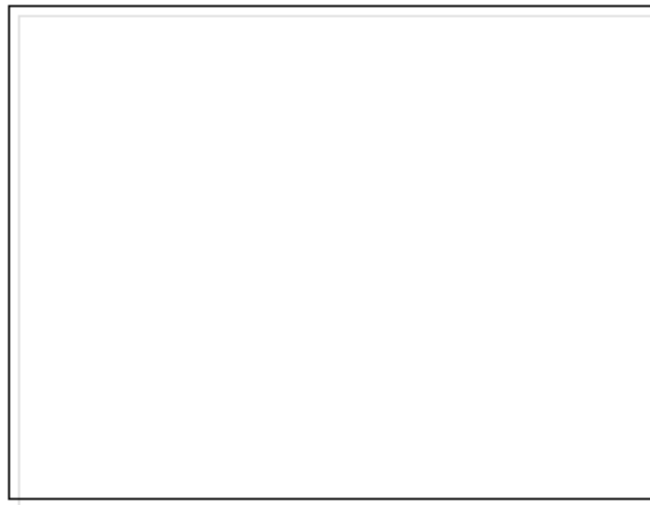


Elementos do diagrama

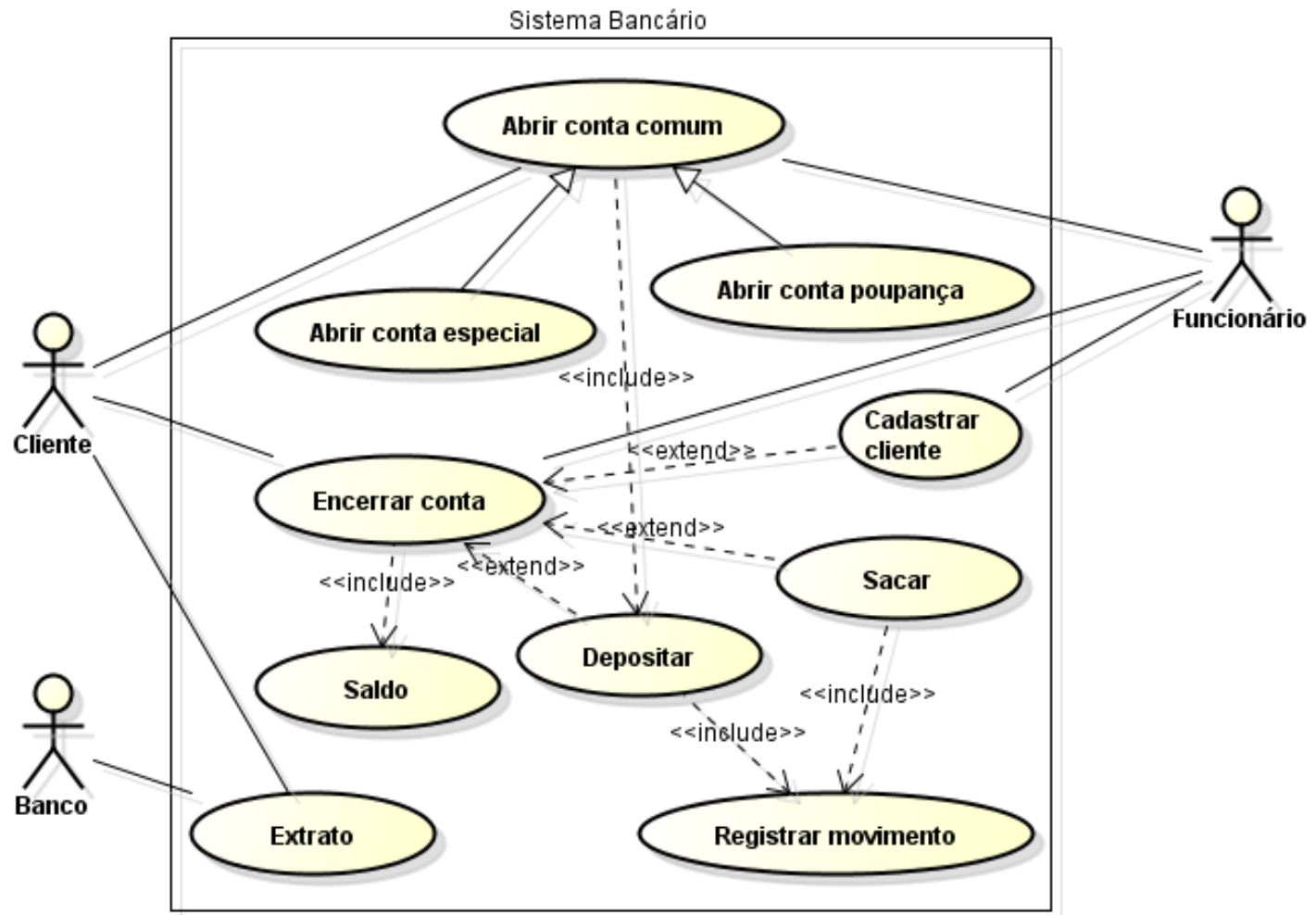
- ❑ Atores
- ❑ Casos de uso
- ❑ Relacionamentos
 - Associação
 - Generalização
 - Inclusão
 - Extensão
- ❑ **Fronteira do sistema**

Fronteira do sistema

- ❑ Elementos opcional
- ❑ Serve para definir a área de atuação do sistema
- ❑ Notação:



Fronteira do sistema



Estrutura de um documento de requisitos

Capítulo	Descrição
Prefácio	Deve definir os possíveis leitores do documento e descrever seu histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das mudanças feitas em cada versão.
Introdução	Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como ele vai funcionar com outros sistemas. Também deve descrever como o sistema atende aos objetivos globais de negócio ou estratégicos da organização que encomendou o software.
Glossário	Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou o conhecimento do leitor.
Definição de requisitos de usuário	Deve descrever os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.
Arquitetura do sistema	Deve apresentar uma visão geral em alto nível da arquitetura do sistema previsto, mostrando a distribuição de funções entre os módulos do sistema. Componentes de arquitetura que são reusados devem ser destacados.

Estrutura de um documento de requisitos

Capítulo	Descrição
Especificação de requisitos do sistema	Deve descrever em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.
Modelos do sistema	Pode incluir modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.
Evolução do sistema	Deve descrever os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, por exemplo. Os requisitos de hardware definem as configurações mínimas ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.
Índice	Vários índices podem ser incluídos no documento. Pode haver, além de um índice alfabético normal, um índice de diagramas, de funções, entre outros pertinentes.

Diretrizes para escrever requisitos

- ❑ Utilize um formato-padrão e garanta que todas as definições de requisitos aderem a esse formato
- ❑ Use uma linguagem consistente para distinguir entre os requisitos obrigatórios e os desejáveis
- ❑ Use uma forma de destacar as partes fundamentais do requisito (negrito, itálico ou cores)
- ❑ Não assuma que os leitores compreendem a linguagem técnica da engenharia de software (ex. arquitetura, módulo)
- ❑ Sempre que possível, tente associar uma lógica a cada um dos requisitos de usuário

Exercícios

Desenhar o diagrama de caso de uso para os cenários descritos nos exercícios 7, 8, 9 e 10.

7. Uma loja de CDs possui discos para venda. Um cliente pode comprar uma quantidade ilimitada de discos para isso ele deve se dirigir à loja. A loja possui um atendente cuja função é atender os clientes durante a venda dos discos. A loja também possui um gerente cuja função é administrar o estoque para que não falem discos. Além disso, é ele quem dá folga ao atendente, ou seja, ele também atende os clientes durante a venda dos discos. As vendas podem ser à vista ou a prazo. Em ambos os casos, o estoque é atualizado e uma nota fiscal é entregue ao consumidor:
 - No caso de uma venda à vista, clientes cadastrados na loja e que comprem mais de 5 CDs de uma só vez ganham um desconto de 1% para cada ano de cadastro.

Exercícios (continuação)

- No caso de uma venda a prazo, ela pode ser parcelada em 2 pagamentos com acréscimo de 20%. As vendas a prazo podem ser pagas no cartão ou no boleto. Para pagamento com boleto, são gerados boletos bancários que são entregues ao cliente e armazenados no sistema e para lançamento posterior no caixa. Para pagamento com cartão, os clientes com mais de 10 anos de cadastro na loja ganham o mesmo desconto das compras à vista.

A partir do cenário descrito, desenhe o diagrama de casos de uso.

Exercícios

8. João confecciona placas por encomenda. Como o volume dos pedidos tem aumentado, ele pediu ao filho que lhe fizesse uma pequena aplicação que controle:

- O cadastro de seus clientes
- As encomendas

Quando recebe uma encomenda, João anota num caderninho o nome do cliente e seu telefone. Para a encomenda, ele registra: o tamanho da placa (altura e largura), a frase a ser escrita, cor da placa (branca ou cinza), cor da frase (azul, vermelho, amarelo, preto ou verde), data da entrega, valor do serviço e valor do sinal.

A aplicação deve obrigar que o valor do sinal seja de, no mínimo, 50%.

Para calcular o valor da placa, as seguintes fórmulas são usadas:

```
área          = altura x largura
custo_material = área x R$ 147,30
custo_desenho  = número_letras x R$ 0,32
valor_placa    = custo_material + custo_desenho
```

Exercícios (continuação)

Para calcular o prazo de entrega, considera-se que ele só consegue produzir seis placas por dia.

João deseja que o sistema controle os pedidos, calcule o preço final das peças e o prazo de entrega. Para cada encomenda cadastrada, deve ser emitido um recibo em duas vias (cliente e empresa), contendo todos os dados da encomenda e do pagamento.

A partir do cenário descrito, desenhe o diagrama de casos de uso. Considere que só existe um ator, que é o Sr. João, dono de sua micro-empresa, que pode ser identificado como “Diretor da empresa”.

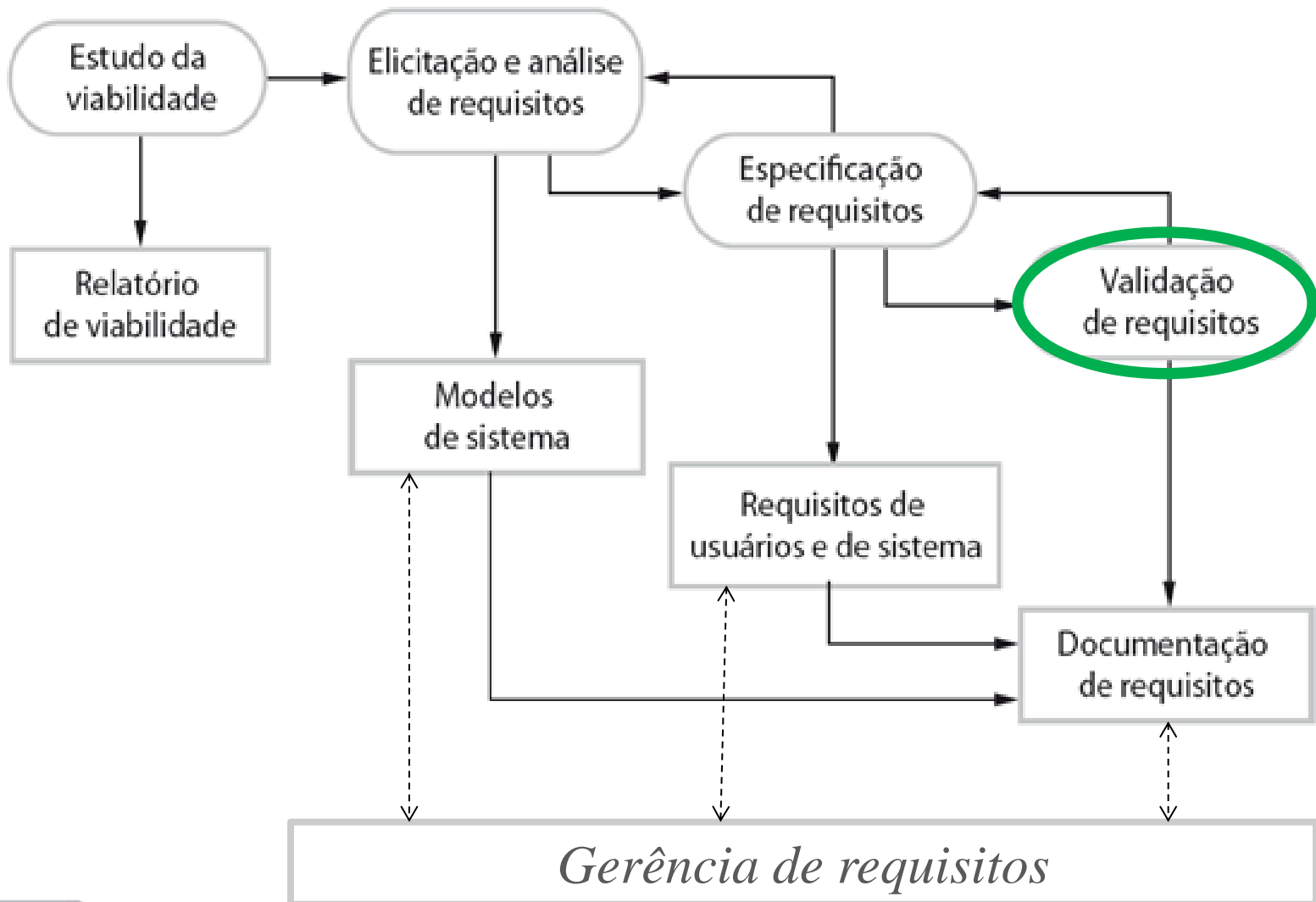
Exercícios

9. Gustavo tem uma coleção grande de revistas em quadrinhos. Por isso, resolveu emprestar para os amigos. Assim foi criado o Clube da Leitura. Mas para não perder nenhuma revista, seu pai lhe fez uma aplicação que cadastra as revistas e controla o empréstimo. Para cada revista, cadastram-se: o tipo de coleção, o número da edição, o ano da revista e a caixa onde está guardada. Cada caixa tem uma cor, uma etiqueta e um número. Para cada empréstimo cadastram-se: o amiguinho que pegou a revista, qual foi a revista, a data do empréstimo e a data de devolução. Cada criança só pode pegar uma revista por empréstimo. O cadastro do amiguinho consiste de: nome, nome do responsável, telefone e de onde é o amigo.

Exercícios

10. Quando me aposentar, vou abrir uma loja de bicicletas na Serra do Cipó e quero automatizar a minha loja. A loja venderá, comprará e trocará bicicletas novas e usadas (fazemos qualquer negócio!), peças para bicicletas, além de possuir uma oficina. O programa deverá ser capaz de armazenar os dados de todas as bicicletas do estoque (marca, modelo, último dono, ano de fabricação, número de marchas, tipo de marcha, tipo de freio, de suspensão, de roda e de quadro), gerenciar o estoque de peças e a oficina. Gostaria de ter relatórios diários do total de consertos, das peças usadas nos consertos, das vendas, das compras e o inventário total da loja. Também quero ter acesso aos dados de todos os clientes da loja, além de um relatório mensal mostrando todas as visitas e trocas efetuadas pelos clientes, listadas por dias do mês.

O processo da engenharia de requisitos



Validação de requisitos

- ❑ Processo para verificar se os requisitos definem o sistema que o cliente realmente quer
- ❑ Está preocupada em encontrar problemas com os requisitos
- ❑ Erros em um documento de requisitos podem gerar altos custos quando descobertos durante o desenvolvimento ou após o sistema já estar em serviço

Validação de requisitos

☐ Validade

- O sistema fornece as funções que melhor atendem às necessidades do cliente?

☐ Consistência

- Existe algum conflito de requisitos?

☐ Completude

- Estão incluídas todas as funções e restrições requeridas pelo cliente?

Validação dos requisitos

❑ Realismo

- Os requisitos podem ser implementados com o orçamento e a tecnologia disponíveis?

❑ Verificabilidade

- Os requisitos podem ser verificados?
- É possível escrever um conjunto de testes que demonstrem que o sistema entregue atende a cada requisito especificado?

Técnicas de validação de requisitos

❑ Revisões de requisitos

- Os requisitos são analisados sistematicamente por uma equipe de revisores que verifica erros e inconsistências

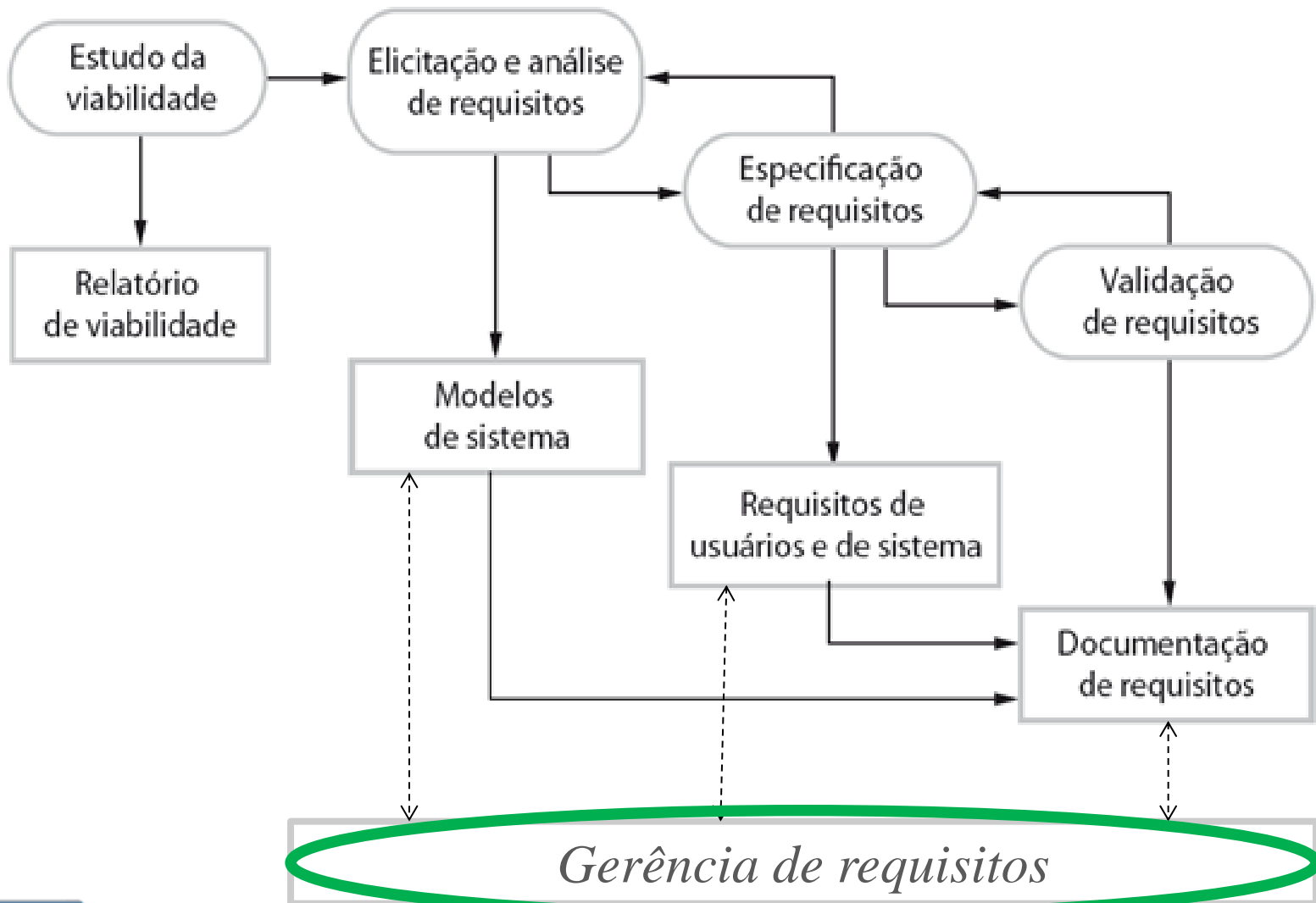
❑ Prototipação

- Um modelo executável do sistema é demonstrado para os usuários finais

❑ Geração de casos de teste

- Os requisitos devem ser testáveis
- Se os testes forem concebidos como parte do processo de validação, isso frequentemente revela problemas de requisitos

O processo da engenharia de requisitos



Gerência de requisitos

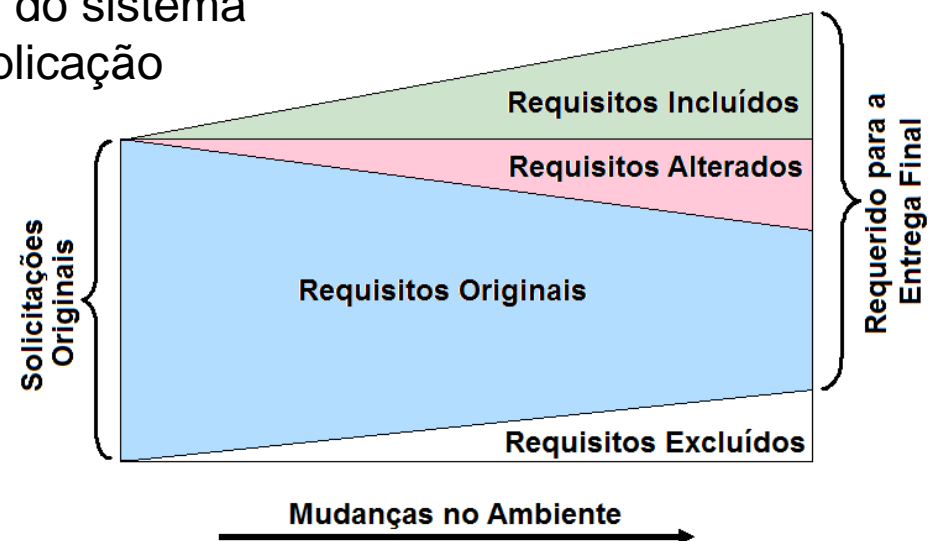
- ❑ Mas o que acontece se...
 - O usuário mudar de ideia em relação a uma funcionalidade?
 - O ambiente mudar?
 - O usuário perceber novas funcionalidades?
 - O engenheiro de requisitos (ou analista) não entendeu corretamente a necessidade do usuário?
- ❑ É preciso gerenciar as mudanças!

Gerência de requisitos

- ❑ Mudanças em requisitos ao longo do processo de desenvolvimento de software fazem parte do processo
- ❑ Alterações em requisitos podem implicar em mudanças em artefatos de projeto, de código, casos de testes, etc.
- ❑ Os requisitos para sistemas de software estão sempre mudando

Gerência de requisitos

- ❑ A necessidade de gerenciar requisitos
 - Mudanças nos requisitos são inevitáveis
 - Qualquer mudança deve ser implementada de forma adequada, sem afetar a qualidade final do produto
 - Requisitos estáveis
 - Não alteram durante um projeto
 - Associados à essência do sistema e ao seu domínio de aplicação
 - Requisitos voláteis
 - Sofrem alteração
 - Em geral específicos de uma instância do sistema para um cliente e para um contexto particular



Por que os requisitos mudam?

❑ Alterações nas prioridades do cliente

- As prioridades do cliente podem mudar durante o desenvolvimento
 - Novos concorrentes no mercado, mudanças de pessoal na organização, ...

❑ Alterações no contexto do sistema

- O meio no qual o sistema vai ser instalado pode mudar, causando alterações nos requisitos do sistema
 - Legislação, tecnologia, ...

❑ Alterações na organização

- A organização pode sofrer alterações na sua estrutura e nos seus processos, criando novos requisitos do sistema

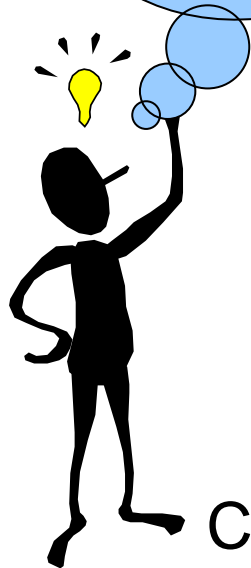
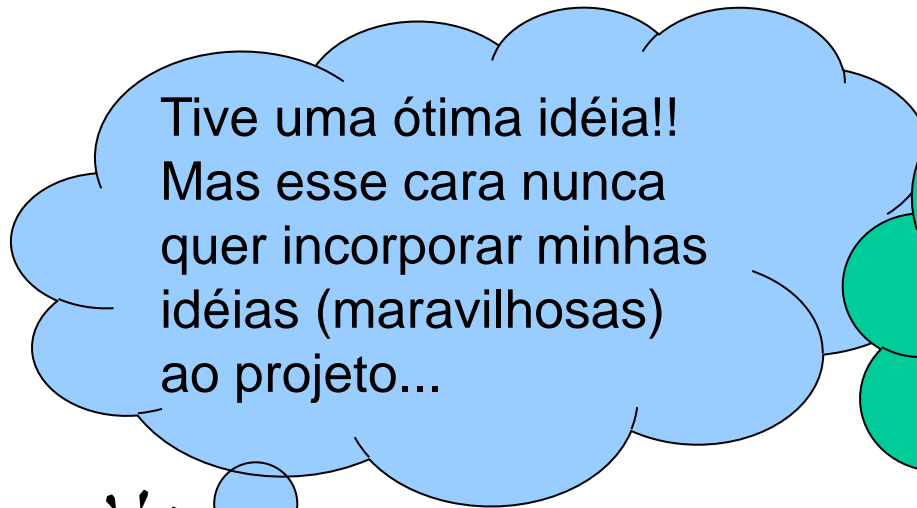
Por que os requisitos mudam?

- ❑ Erros, conflitos e inconsistências nos requisitos
 - Durante a análise e implementação dos requisitos, podem surgir erros
- ❑ Evolução do conhecimento do sistema pelos clientes/usuários
 - Aprendizado a partir da experiência adquirida
- ❑ Problemas técnicos, de planejamento ou de custo
 - Podem surgir problemas durante a implementação de um requisito
 - O custo (ou o tempo) de implementação pode ser muito alto

Por que os requisitos mudam?

- ❑ Uma vez que um sistema tenha sido instalado e seja usado regularmente, inevitavelmente surgirão novos requisitos
- ❑ É difícil para os usuários e clientes do sistema anteciparem os efeitos que o novo sistema terá sobre seus processos de negócio e sobre a forma que o trabalho é realizado
- ❑ Quando os usuários finais tiverem a experiência de um sistema, descobrirão novas necessidades e prioridades

O que fazer quando um requisito muda?



O que fazer quando um requisito muda?

❑ O cliente tem razão ...

- O ambiente de negócios muda
- O cliente muda sua visão a partir da evolução do projeto
- Os requisitos mudam ... e mudarão sempre ...

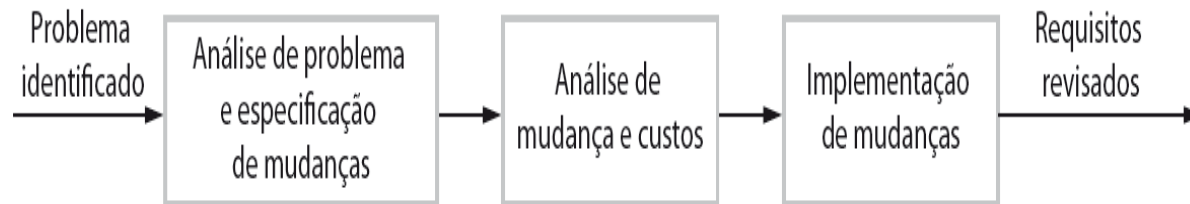
❑ O analista tem razão ...

- Se não parar de mudar, o projeto nunca vai acabar
- Os custos do projeto aumentam
 - Quem paga o re-trabalho?
- Um projeto precisa ter fim... o escopo precisa ser fechado...

Gerência de Requisitos = Mudanças sob Controle !!

Controle de mudança em requisitos

- ❑ O gerenciamento de requisitos é o processo de compreensão e controle das mudanças nos requisitos do sistema
- ❑ Durante esse estágio você deve decidir sobre:
 - Identificação de requisitos
 - Processo de gerenciamento de mudança



- Políticas de rastreabilidade
- Ferramentas de apoio

Processos ágeis e gerência de requisitos

- ❑ Processo ágeis foram concebidos para lidar com requisitos mutáveis
- ❑ Nesses processos, quando um usuário propõe uma mudança nos requisitos, a mudança não passa por um processo formal de gerenciamento de mudanças
- ❑ O usuário tem de priorizar essa mudança e, em caso de alta prioridade, decidir quais recursos do sistema planejados para a próxima iteração devem ser abandonados

Conclusão

❑ Requisitos de Software

- Possuem diferentes visões
 - Definem uma meta para a equipe de desenvolvimento
 - Exprimem critérios de aceitação e validação pelo cliente
- Devem ser capturados de forma que sejam inteligíveis para o cliente e para a equipe de desenvolvimento
 - Devem expressar o problema a resolver e não a solução proposta

Conclusão

❑ Engenharia de Requisitos

- Inclui atividades para elicitar, organizar, documentar e validar funcionalidades e restrições necessárias
- Também cuida dos controles de rastreabilidade e mudança

❑ Especificação de Requisitos

- Documenta, de forma clara e explícita, os requisitos funcionais e não funcionais

Exercícios

11. Quando mudanças emergenciais precisam ser feitas em sistemas, o software do sistema pode precisar ser modificado antes de serem aprovadas as mudanças nos requisitos. Sugira um modelo de um processo para fazer essas modificações de modo a garantir que o documento de requisitos e implementação do sistema não se tornem inconsistentes.