

# Banco de Dados

Jordana S. Salamon

[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

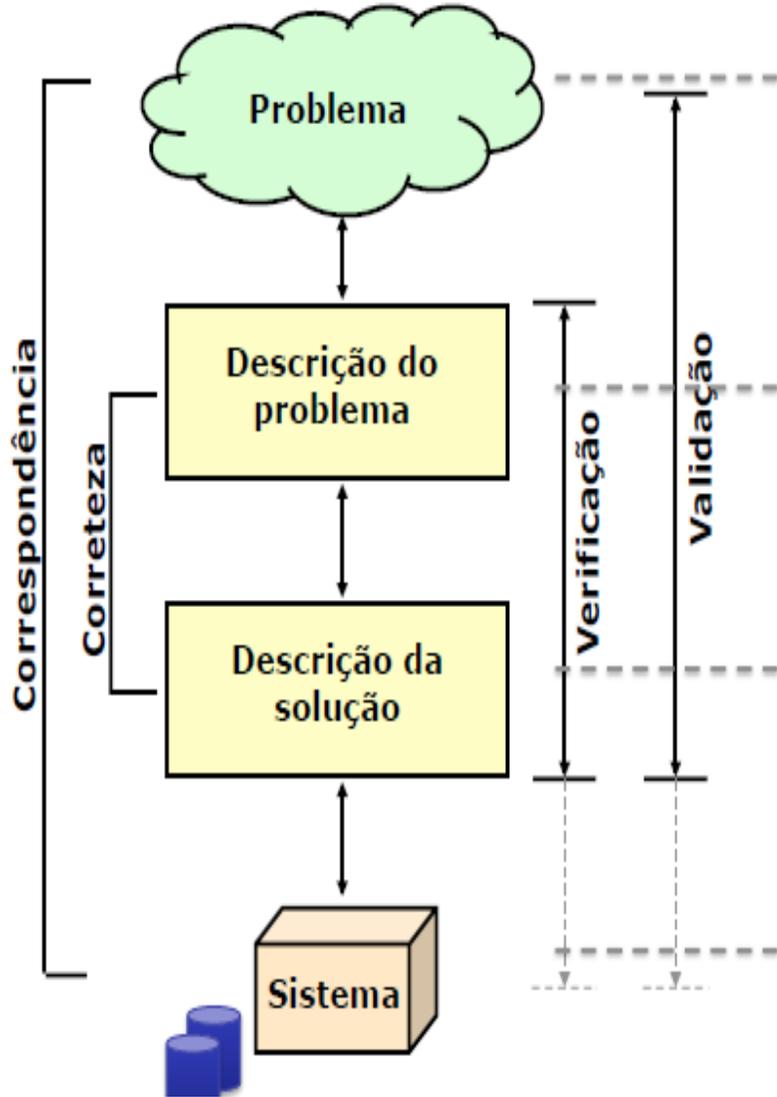
[jordanasalamon@gmail.com](mailto:jordanasalamon@gmail.com)

DEPARTAMENTO DE INFORMÁTICA

CENTRO TECNOLÓGICO

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

# Desenvolvimento de sistemas



Gestão de serviços de uma biblioteca.

Toda solicitação de empréstimo deve ser atendida.

Processo de check-out de livros da biblioteca.

Sistema de check-out de livros da biblioteca.

# Separação entre problema e solução

- ▶ O problema mais óbvio pode não ser o problema que deve ser resolvido;
- ▶ Deve-se discutir o problema com os **stakeholders** para um melhor entendimento;
- ▶ Uma descrição precisa do problema:
  - ▶ Auxilia nas escolhas de projeto;
  - ▶ Permite a criação de bons casos de teste;
  - ▶ Auxilia na comunicação entre a equipe de desenvolvimento.

**Stakeholders** são pessoas que são afetadas pelo problema e, portanto, tem algo a dizer sobre sua solução. Ex.: clientes, usuários, etc.

# Engenharia de requisitos



- ▶ **Propriedades do domínio:**
  - ▶ Coisas que são verdadeiras independente de construirmos ou não um sistema;
- ▶ **Requisitos:**
  - ▶ Coisas que queremos que o sistema atenda;
- ▶ **Especificação:**
  - ▶ Uma descrição do comportamento que o sistema deve ter para atender os requisitos.

# Completude e correção

- ▶ Critérios de correção:
  - ▶ Um determinado **sistema** satisfaz a **especificação**;
  - ▶ A **especificação**, dadas as **propriedades** de domínio, satisfaz os **requisitos**.
- ▶ Critérios de completude:
  - ▶ Descobrimos todos os **requisitos** importantes;
  - ▶ Descobrimos todas as **propriedades** de domínio relevantes.



# Engenharia de Requisitos

# Engenharia de Requisitos

- ▶ Entender o que os stakeholders querem;
- ▶ Analisar a necessidade;
- ▶ Verificar a factibilidade;
- ▶ Negociar uma solução razoável;
- ▶ Especificar a solução sem ambiguidade;
- ▶ Validar a especificação;
- ▶ Gerenciar mudanças nos requisitos;
- ▶ Etc.



# Engenharia de Requisitos

- ▶ Alguns benefícios que um processo de ER de qualidade pode trazer são:
  - ▶ menor quantidade de defeitos nos requisitos,
  - ▶ redução de retrabalho,
  - ▶ desenvolvimento de menos características desnecessárias,
  - ▶ diminuição de custos,
  - ▶ desenvolvimento mais rápido,
  - ▶ menos problemas de comunicação,
  - ▶ alterações de escopo reduzidas,
  - ▶ estimativas mais confiáveis,
  - ▶ maior satisfação de clientes e desenvolvedores.



# Engenharia de Requisitos de Software



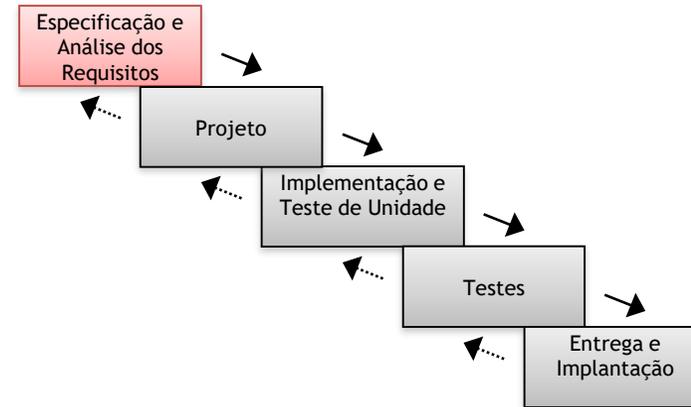
Que requisitos o software deve atender?



- Quem são os envolvidos?
- Quais são suas necessidades em relação ao software?

Mas...

**O que é um requisito?**



# Engenharia de Requisitos de Software

## Requisitos

Requisitos são descrições dos serviços que devem ser providos pelo sistema e de suas restrições operacionais (SOMMERVILLE, 2007).

Um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir seus objetivos (PFLEEGER, 2004).

Um requisito é alguma coisa que o produto tem de fazer ou uma qualidade que ele precisa apresentar (ROBERTSON; ROBERTSON, 2006).



# Engenharia de Requisitos de Software

## Requisitos

Os requisitos de um sistema definem o que o sistema deve fazer e as circunstâncias sob as quais deve operar.

Ou, em outras palavras...

São as funções que um sistema deve incorporar e as restrições que devem ser satisfeitas.



# Engenharia de Requisitos de Software

## Requisitos

Uma das principais medidas do sucesso de um sistema de software é o grau no qual ele atende aos requisitos para os quais foi construído.



# Engenharia de Requisitos de Software

## Tipos de Requisitos

**Funcionais:** são declarações de serviços que o sistema deve prover, descrevendo o que o sistema deve fazer (SOMMERVILLE, 2007). Um requisito funcional descreve uma interação entre o sistema e o seu ambiente (PFLEEGER, 2004), podendo descrever, ainda, como o sistema deve reagir a entradas específicas, como o sistema deve se comportar em situações específicas e o que o sistema não deve fazer (SOMMERVILLE, 2007).

*Ex.: O sistema deve registrar locações, indicando o cliente, os itens locados, a data da locação, a data de devolução e o valor da locação.*



# Engenharia de Requisitos de Software

## Tipos de Requisitos

**Não Funcionais:** descrevem restrições sobre os serviços ou funções oferecidos pelo sistema (SOMMERVILLE, 2007), as quais limitam as opções para criar uma solução para o problema (PFLEEGER, 2004). Neste sentido, os requisitos não funcionais são muito importantes para a fase de projeto (design), servindo como base para a tomada de decisões nessa fase.

Ex.: A consulta ao acervo da locadora deve estar disponível pela Internet, a partir dos principais navegadores disponíveis no mercado. (requisito de portabilidade)



# Engenharia de Requisitos de Software

## Tipos de Requisitos

Além desses requisitos, é importante considerar também *Requisitos de Domínio*.

*Requisitos de domínio (ou regras de negócio)* são provenientes do domínio de aplicação do sistema e refletem características e restrições desse domínio. Eles são derivados do negócio que o sistema se propõe a apoiar e podem restringir requisitos funcionais existentes ou estabelecer como cálculos específicos devem ser realizados, refletindo fundamentos do domínio de aplicação (SOMMERVILLE, 2011).

*Ex.: Em um sistema de matrícula de uma universidade, uma importante regra de negócio diz que um aluno só pode se matricular em uma turma de uma disciplina se ele tiver cumprido seus pré-requisitos.*

# Engenharia de Requisitos de Software

## *Requisitos Funcionais (o que o sistema deve fazer?)*

*Ex:*

Identificador	Descrição	Prioridade	Requisitos Relacionados
RF01	O sistema deve registrar locações, indicando o cliente e os itens locados, bem como a data e o valor da locação e a data de devolução prevista de cada item.	Alta	RF06,RF09,RN01,RN08,RN11,RN12,RN13, RN14,RN16,RNF03,RNF04
RF02	O sistema deve registrar devoluções, indicando os itens devolvidos e a data de devolução	Alta	RF01,RN02,RN08,RNF03,RNF04
RF03	O sistema deve registrar os pagamentos de locações	Alta	RF01,RF02,RN01,RN08,RN09,RNF04, RNF05,
RF04	O sistema deve registrar a reserva de filmes a clientes, permitindo indicar, ainda, o tipo de mídia desejado	Média	RF06,RF09,RF01,RF02,RN10,RN16,RN17, RNF04
RF05	O sistema deve permitir o cancelamento de uma reserva, tanto pelo usuário, quanto automaticamente pelo sistema, quando expirado o prazo para retirada do item, de acordo com política da empresa.	Média	RF05,RN03,RN15,RNF04

# Engenharia de Requisitos de Software

**Regras de Negócio** (regras que devem ser obedecidas pelo sistema)

**Ex:**

Identificador	Descrição	Prioridade	Requisitos Relacionados
RN01	O sistema deve permitir que sejam dados descontos nas locações, bem como que sejam ampliados os prazos de devolução de itens, em função da política da empresa.	Média	RN14
RN02	O sistema deve cobrar multa para itens devolvidos com atraso, segundo a seguinte fórmula: $M = n * VL$ , onde M é o valor da multa, n é o número de dias de atraso e VL é o valor de locação do item.	Alta	
RN03	Reservas canceladas pelo sistema não deverão ser efetivamente excluídas pelo sistema, mas sim marcadas como expiradas.	Média	
RN04	O sistema deve manter o histórico de locações e, portanto, clientes que tenham feito locações não poderão ser excluídos.	Alta	

# Engenharia de Requisitos de Software

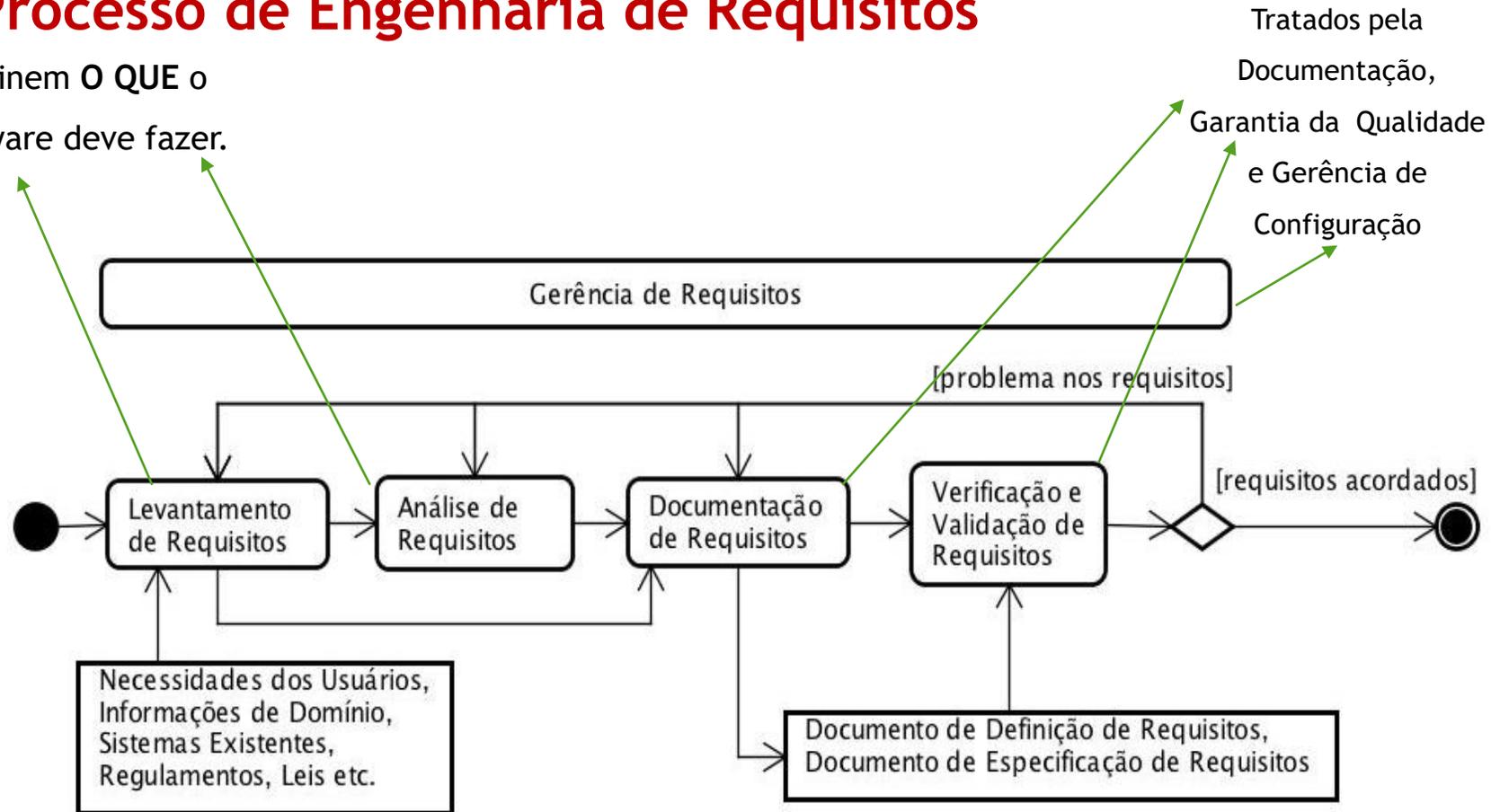
**Requisitos Não Funcionais** (tipicamente, restrições a serem obedecidas em relação a: segurança, portabilidade, desempenho, usabilidade, interoperabilidade,...)

Identificador	Descrição	Categoria	Escopo	Prioridade	Requisitos Relacionados
RNF01	O sistema deve controlar o acesso às funcionalidades. Funcionalidades para controlar o acervo da locadora devem ser restritas a administradores. Funcionalidades de atendimento a clientes devem estar restritas a atendentes. Funcionalidades de consulta ao acervo devem estar disponíveis na Internet.	Segurança de Acesso	Sistema	Alta	
RNF02	A consulta ao acervo deve estar disponível pela Internet, a partir dos principais navegadores disponíveis no mercado.	Portabilidade	Funcionalidade	Média	
RNF03	Os itens devem ser identificados por um código de barras, sendo possível a leitura dos mesmos usando dispositivos de leitores de código de barras.	Facilidade de Operação	Funcionalidade	Alta	

# Engenharia de Requisitos

## Processo de Engenharia de Requisitos

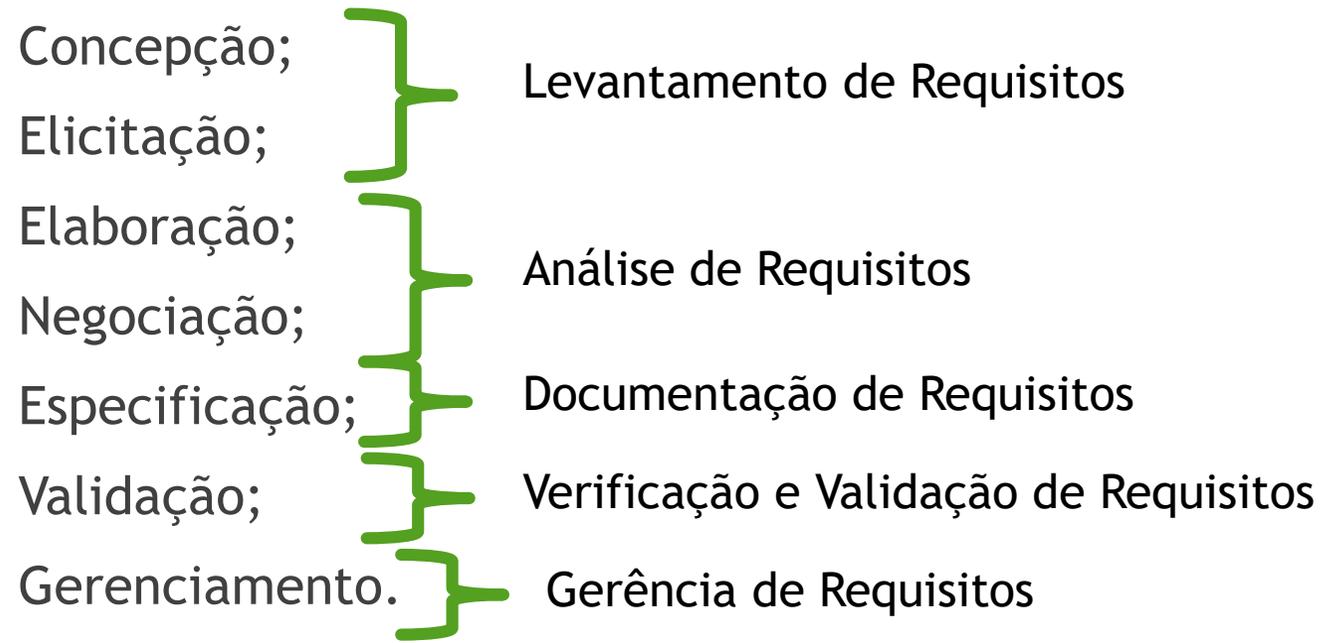
Definem O QUE o software deve fazer.



Ao conjunto de atividades relacionadas aos requisitos, dá-se o nome de **Engenharia de Requisitos**.

# Engenharia de Requisitos

► São funções desta fase do processo:



# Concepção - Levantamento Preliminar de Requisitos

- ▶ Geralmente é a primeira fase do processo;
- ▶ Identificação de um problema ou oportunidade;
- ▶ Perguntas genéricas e superficiais;
- ▶ Objetivo é estabelecer:
  - ▶ Um entendimento básico do problema;
  - ▶ Quem são os stakeholders;
  - ▶ A natureza da solução desejada;
  - ▶ A eficácia da comunicação entre o engenheiro de requisitos e os especialistas de domínio.



# Primeiras perguntas (exemplo)

- ▶ Quem está pedindo esta solução?
- ▶ Quem irá usá-la?
- ▶ Qual é seu benefício econômico?
- ▶ De que problema(s) esta solução irá tratar?
- ▶ Em que ambiente de negócio ela está inserida?
- ▶ Existem qualidades fundamentais (desempenho, segurança, etc.) relevantes ao problema?
- ▶ ...



# Elicitação - Levantamento de Requisitos

- ▶ A meta é descobrir informações sobre o problema:
  - ▶ Os objetivos dos stakeholders (problema);
  - ▶ As funções do sistema (solução) a ser construído (o que ele deve fazer);
  - ▶ Como o sistema se encaixa nas necessidades de negócio do cliente;
  - ▶ Como será usado no dia-a-dia.
- ▶ Requer alto nível de organização.



# Técnicas de Elicitação

- ▶ As “primeiras perguntas” darão somente um entendimento básico do problema;
- ▶ Para elicitar os requisitos, devemos utilizar abordagens mais sofisticadas. Algumas são:
  - ▶ Entrevista;
  - ▶ Observação;
  - ▶ Questionário;
  - ▶ Prototipação;
  - ▶ Análise de Documentos;



# Possíveis problemas na elicitação

- ▶ De escopo:
  - ▶ Os limites do sistema não são bem definidos;
  - ▶ O cliente especifica muitos detalhes inúteis.
- ▶ De entendimento:
  - ▶ O cliente não tem certeza do que quer;
  - ▶ Não conhece as capacidades e limitações do ambiente computacional;
  - ▶ Possui problemas de comunicação com os engenheiros de software;
  - ▶ Omite informações consideradas “óbvias”;



# Possíveis problemas na elicitação

- ▶ De entendimento (continua):
  - ▶ Especifica requisitos que conflitam com os de outro cliente;
  - ▶ Especifica requisitos ambíguos.
- ▶ Políticos:
  - ▶ Funcionários não colaboram por acharem que o sistema lhes custará o emprego;
  - ▶ Brigas políticas internas.
- ▶ De volatilidade:
  - ▶ Os requisitos mudam com o tempo.





Como o cliente explicou...



Como o líder de projeto entendeu...



Como o analista projetou...



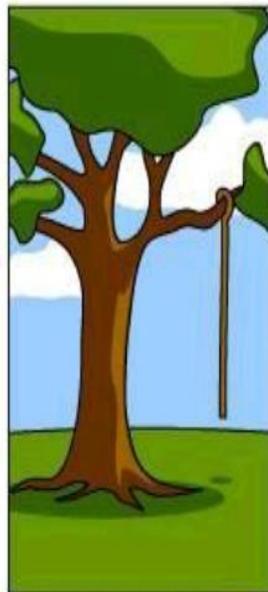
Como o programador construiu...



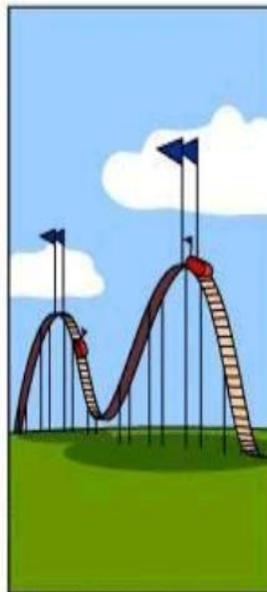
Como o Consultor de Negócios descreveu...



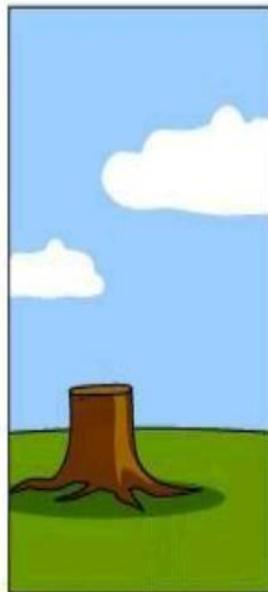
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

# Processo de Engenharia de Requisitos

## Análise de Requisitos

- ▶ Uma vez preliminarmente identificados os requisitos, é possível iniciar a atividade de análise, quando os requisitos levantados devem ser refinados.
- ▶ É uma atividade de construção de **modelos**.
- ▶ Um modelo é uma representação de alguma coisa do mundo real, uma abstração da realidade, e, portanto, representa uma seleção de características do mundo real relevantes para o propósito do sistema em questão.
- ▶ Esses modelos são representações gráficas que descrevem objetivos e processos de negócio, o problema a ser resolvido e o sistema a ser desenvolvido.



# Processo de Engenharia de Requisitos

## Análise de Requisitos

- ▶ No desenvolvimento de sistemas, há duas **perspectivas** principais:
  - *Perspectiva estrutural*: tem por objetivo descrever as informações que o sistema deve representar e gerenciar. Provê uma visão estática das informações que o sistema necessita tratar. Ex.: diagramas de classes e modelos ER.
  - *Perspectiva comportamental*: visa especificar as ações (funcionalidades / serviços) que o sistema deve prover, bem como o comportamento de certas entidades do modelo estrutural em relação a essas ações. Ex.: Diagramas de casos de uso, diagramas de atividades, diagramas de estados e diagramas de interação são usados para modelar essa visão.



# Processo de Engenharia de Requisitos

## Análise de Requisitos

- ▶ Contudo, outras perspectivas podem ser alvo de modelos. A abordagem de Engenharia de Requisitos Baseada em Objetivos (*Goal-Oriented Requirements Engineering* - GORE), p.ex., assume que objetivos são uma perspectiva fundamental, pois estabelecem o "porquê" do sistema (e, portanto, dos elementos identificados em outras perspectivas). Na abordagem GORE, as razões para um novo sistema (ou uma nova versão de um sistema) precisam ser explicitadas em termos de objetivos a serem satisfeitos por ele e, para tal, modelos de objetivos devem ser desenvolvidos.



# Processo de Engenharia de Requisitos

## Análise de Requisitos

- ▶ A análise de requisitos é uma atividade extremamente vinculada ao levantamento de requisitos.
- ▶ Durante o levantamento de requisitos, alguns problemas são identificados e tratados. Entretanto, determinados problemas somente são identificados por meio de uma análise mais detalhada. **A análise de requisitos ajuda a entender e detalhar os requisitos levantados, a descobrir problemas nesses requisitos e a obter a concordância sobre as alterações, de modo a satisfazer a todos os envolvidos.** Seu objetivo é estabelecer um conjunto acordado de requisitos completos, consistentes e sem ambiguidades, que possa ser usado como base para as demais atividades do processo de desenvolvimento de software.



# Processo de Engenharia de Requisitos

## Análise de Requisitos

- ▶ Problemas e conflitos encontrados nos requisitos devem ser listados.
- ▶ Usuários, clientes, especialistas de domínio e engenheiros de requisitos devem discutir os requisitos que apresentam problemas, negociar e chegar a um acordo sobre as modificações a serem feitas.
- ▶ A maior parte do tempo da negociação é utilizada para resolver conflitos de requisitos.



# Elaboração

Modelos são fundamentais no desenvolvimento de sistemas.

Tipicamente eles são construídos para:

- focar os aspectos chave, em detrimento de detalhes irrelevantes;
- possibilitar o estudo do comportamento do sistema;
- facilitar a comunicação entre membros da equipe de desenvolvimento e clientes e usuários;
- possibilitar a discussão de correções e modificações com o usuário;
- servir como base para a tomada de decisão;
- Fornecem uma estrutura para as atividades da ER, sendo a base para a geração de documentos.



# Elaboração



**Mapa Político**



**Mapa Turístico**

Exemplos de modelos como abstrações da realidade

# Negociação - Análise de Requisitos

- ▶ Não é incomum:
  - ▶ Clientes quererem mais do que é possível ser feito;
  - ▶ Stakeholders terem requisitos conflitantes.
- ▶ Deve-se reconhecer os múltiplos pontos de vista e tentar negociar uma solução adequada;
- ▶ Idealmente, deve-se evitar situações em que hajam “vencedores” e “perdedores”.

“Coloque três stakeholders numa sala e pergunte que tipo de sistema eles querem. Você provavelmente vai obter quatro ou mais opiniões diferentes” (Anônimo)



# Especificação - Documentação de Requisitos

- ▶ Os requisitos e modelos capturados nas etapas anteriores devem ser descritos e apresentados em documentos. A documentação é, portanto, uma atividade de registro e oficialização dos resultados da Engenharia de Requisitos.
- ▶ Uma boa documentação fornece muitos benefícios, tais como:
  - (i) Facilita a comunicação dos requisitos;
  - (ii) Reduz o esforço de desenvolvimento, pois sua preparação força usuários e clientes a considerar os requisitos atentamente, evitando retrabalho nas fases posteriores;
  - (iii) Fornece uma base realística para estimativas;
  - (iv) Fornece uma base para verificação e validação;
  - (v) Serve como base para futuras manutenções ou incremento de novas funcionalidades.



# Engenharia de Requisitos de Software

- ▶ Existem dois níveis de descrição de requisitos:
  - ▶ • **Requisitos de Cliente ou de Usuário:** são declarações em linguagem natural acompanhadas de diagramas intuitivos de quais serviços são esperados do sistema e das restrições sob as quais ele deve operar. **Devem estar em um nível de abstração mais alto, de modo que sejam compreensíveis pelos clientes e usuários do sistema que não possuem conhecimento técnico.**
  - ▶ • **Requisitos de Sistema:** definem detalhadamente as funções, serviços e restrições do sistema. São versões expandidas dos requisitos de cliente usados pelos desenvolvedores para projetar, implementar e testar o sistema. Como requisitos de sistema são mais detalhados, as especificações em linguagem natural são insuficientes e para especificá-los, notações mais especializadas devem ser utilizadas.



# Processo de Engenharia de Requisitos

## Verificação e Validação (V&V) de Requisitos

- ▶ O objetivo da verificação é **assegurar que o software esteja sendo construído de forma correta**. Deve-se verificar se os artefatos produzidos atendem aos requisitos estabelecidos e se os padrões organizacionais (de produto e processo) foram consistentemente aplicados.
- ▶ O objetivo da validação é **assegurar que o software que está sendo desenvolvido é o software correto**, ou seja, assegurar que os requisitos, e o software deles derivado, atendem ao uso proposto.



# Processo de Engenharia de Requisitos

## Verificação e Validação (V&V) de Requisitos

- ▶ No caso de requisitos, a verificação é feita, sobretudo, em relação à consistência entre requisitos e modelos e à conformidade com padrões organizacionais de documentação de requisitos. Já a validação tem de envolver a participação de usuários e clientes, pois somente eles são capazes de dizer se os requisitos atendem aos propósitos do sistema.
- ▶ Nas atividades de V&V de requisitos, examinam-se os documentos de requisitos para assegurar que: (i) todos os requisitos do sistema tenham sido declarados de modo não-ambíguo, (ii) as inconsistências, conflitos, omissões e erros tenham sido detectados e corrigidos, (iii) os documentos estão em conformidade com os padrões estabelecidos e (iv) os requisitos realmente satisfazem às necessidades dos clientes e usuários.



# Processo de Engenharia de Requisitos

## Verificação de Requisitos

- ▶ Em uma revisão, processos, documentos e outros artefatos são revisados por um grupo de pessoas, com o objetivo de avaliar se os mesmos estão em conformidade com os padrões organizacionais estabelecidos e se o propósito de cada um deles está sendo atingido.
- ▶ Assim, o objetivo de uma revisão é detectar erros e inconsistências em artefatos e processos, sejam eles relacionados à forma, sejam eles relacionados ao conteúdo, e apontá-los aos responsáveis pela sua elaboração.



# Processo de Engenharia de Requisitos

## Gerência de Requisitos

- ▶ Mudanças nos requisitos ocorrem ao longo de todo o processo de software, desde o levantamento e análise de requisitos até durante a operação do sistema.
- ▶ Elas são decorrentes de diversos fatores, tais como descoberta de erros, omissões, conflitos e inconsistências nos requisitos, melhor entendimento por parte dos usuários de suas necessidades, problemas técnicos, de cronograma ou de custo, mudança nas prioridades do cliente, mudanças no negócio, aparecimento de novos competidores, mudanças econômicas, mudanças na equipe, mudanças no ambiente onde o software será instalado e mudanças organizacionais ou legais.
- ▶ Para minimizar as dificuldades impostas por essas mudanças, é necessário gerenciar requisitos.

# Processo de Engenharia de Requisitos

## Gerência de Requisitos

- ▶ O processo de gerência de requisitos envolve as atividades que ajudam a equipe de desenvolvimento a identificar, controlar e rastrear requisitos e gerenciar mudanças de requisitos em qualquer momento ao longo do ciclo de vida do software. Os principais objetivos desse processo são:
  - ▶ • Gerenciar alterações nos requisitos acordados.
  - ▶ • Gerenciar relacionamentos entre requisitos.
  - ▶ • Gerenciar dependências entre requisitos e outros documentos produzidos durante o processo de software.



# Processo de Engenharia de Requisitos

## Gerência de Requisitos

- ▶ Se as mudanças não forem controladas, alterações com baixa prioridade podem ser implementadas antes de outras mais importantes e modificações com custo alto que não são realmente necessárias podem ser aprovadas.
- ▶ Ao se detectar a necessidade de alteração em um ou mais requisitos, deve-se registrar uma solicitação de mudança, a qual deve ser avaliada por algum membro da equipe do projeto de software.
- ▶ Nessa avaliação, o impacto da alteração deve ser determinado e valores de custo, esforço, tempo e viabilidade devem ser repassados ao solicitante da mudança.



# Gerenciamento - Gerência de Requisitos

- ▶ Atividades que ajudam no controle e rastreamento de mudanças nos requisitos:
  - ▶ Cada requisitos recebe um identificador;
  - ▶ São montadas tabelas de rastreamento: funcionalidades, dependências, subsistemas, interface, etc.;
  - ▶ Mudanças nos requisitos podem ser mais facilmente propagadas ;
  - ▶ Problemas no software pronto podem ser analisados em termos dos requisitos.



# Técnicas de Elicitação

45

# Processo de Engenharia de Requisitos

## Levantamento de Requisitos

- ▶ Diversas técnicas podem ser utilizadas no levantamento de requisitos, as quais podem possuir diferentes objetos de investigação ou podem ter foco em tipos diferentes de requisitos.
- ▶ Assim, é útil empregar várias dessas técnicas concomitantemente, de modo a se ter um levantamento de requisitos mais eficaz. Dentre as várias técnicas, podem ser citadas:
  - ▶ • **Entrevistas:** técnica amplamente utilizada, que consiste em conversas direcionadas com um propósito específico e com formato “pergunta-resposta”. Seu objetivo é descobrir problemas a serem tratados, levantar procedimentos importantes e saber a opinião e as expectativas do entrevistado sobre o sistema.

# Processo de Engenharia de Requisitos

## Levantamento de Requisitos

- ▶ • **Questionários:** o uso de questionários possibilita ao analista obter informações como postura, crenças, comportamentos e características de várias pessoas que serão afetadas pelo sistema.
- ▶ • **Observação:** consiste em observar o comportamento e o ambiente dos indivíduos de vários níveis organizacionais. Utilizando-se essa técnica, é possível capturar o que realmente é feito e qual tipo de suporte computacional é realmente necessário. Ajuda a confirmar ou refutar informações obtidas com outras técnicas e ajuda a identificar tarefas que podem ser automatizadas e que não foram identificadas pelos interessados.



# Processo de Engenharia de Requisitos

## Levantamento de Requisitos

- ▶ • **Análise de documentos:** pela análise de documentos existentes na organização, analistas capturam informações e detalhes difíceis de conseguir por entrevista e observação. Documentos revelam um histórico da organização e sua direção.
- ▶ • **Cenários:** com o uso desta técnica, um cenário de interação entre o usuário final e o sistema é montado e o usuário simula sua interação com o sistema nesse cenário, explicando ao analista o que ele está fazendo e de que informações ele precisa para realizar a tarefa descrita no cenário. O uso de cenários ajuda a entender requisitos, a expor o leque de possíveis interações e a revelar facilidades requeridas.



# Processo de Engenharia de Requisitos

## Levantamento de Requisitos

- ▶ • **Prototipagem:** um protótipo é uma versão preliminar do sistema, muitas vezes não operacional e descartável, que é apresentada ao usuário para capturar informações específicas sobre seus requisitos de informação, observar reações iniciais e obter sugestões, inovações e informações para estabelecer prioridades e redirecionar planos.



# Processo de Engenharia de Requisitos

## Levantamento de Requisitos

- ▶ • **Dinâmicas de Grupo:** há várias técnicas de levantamento de requisitos que procuram explorar dinâmicas de grupo para a descoberta e o desenvolvimento de requisitos, tais como Brainstorming e JAD (*Joint Application Development*).
- ▶ Na primeira, representantes de diferentes grupos de interessados engajam-se em uma discussão informal para rapidamente gerarem o maior número possível de ideias. Na segunda, interessados e analistas se reúnem para discutir problemas a serem solucionados e soluções possíveis. Com as diversas partes envolvidas representadas, decisões podem ser tomadas e questões podem ser resolvidas mais rapidamente.
- ▶ A principal diferença entre JAD e Brainstorming é que, em JAD, tipicamente os objetivos do sistema já foram estabelecidos antes dos interessados participarem. Além disso, sessões JAD são normalmente bem estruturadas, com passos, ações e papéis de participantes definidos.



That's all Folks!



nemo