

Engenharia de Requisitos de Software

Jordana S. Salamon

jssalamon@inf.ufes.br

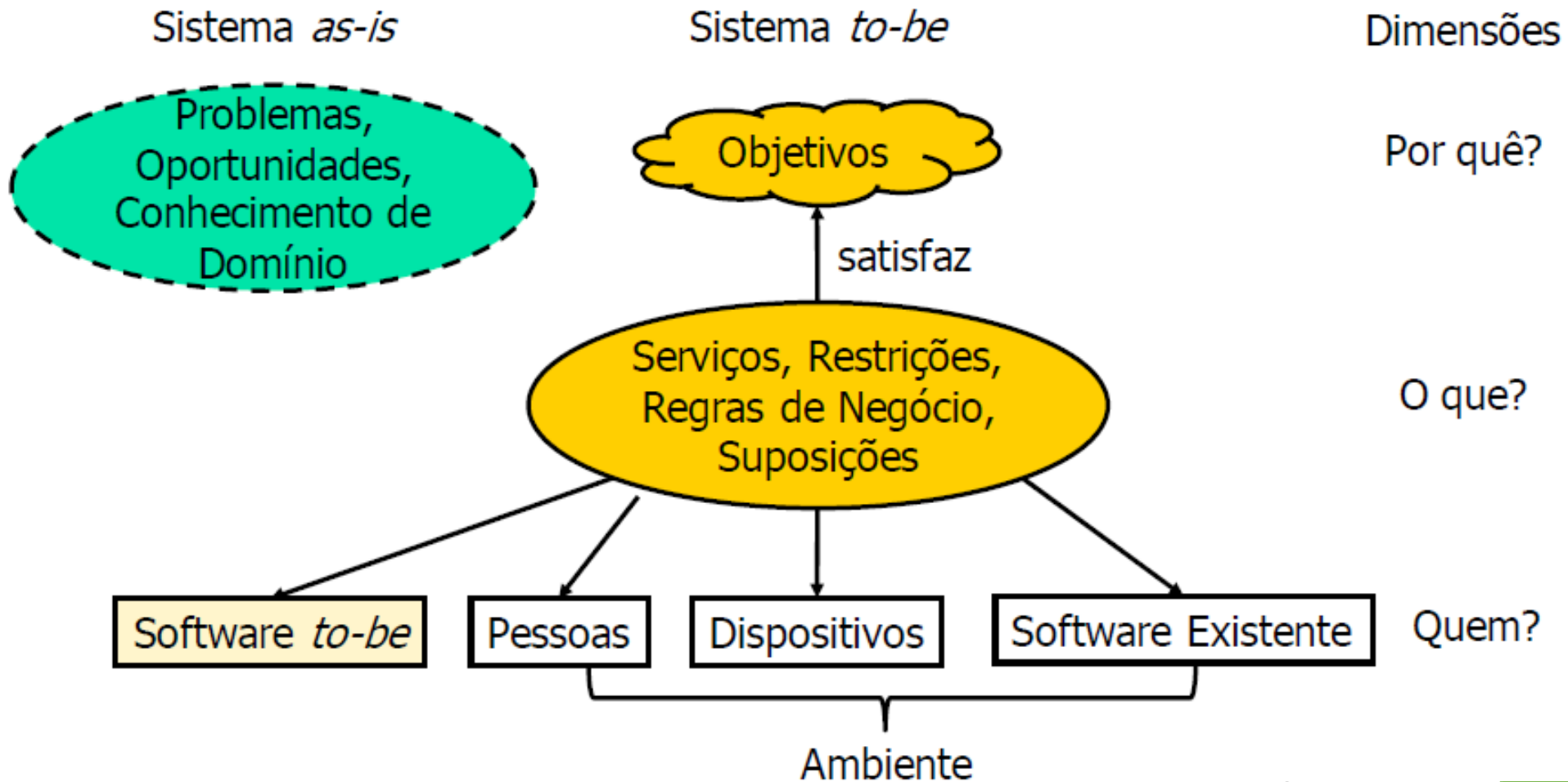
jordanasalamon@gmail.com

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Análise de Requisitos

- ▶ A investigação do domínio do problema e do negócio a ser apoiado nos leva a perceber que o desenvolvimento de um novo sistema de informação é um agente de mudanças no negócio e na prática dos profissionais envolvidos.
- ▶ Podemos pensar que estamos diante de duas versões de um sistema: o sistema como ele é (as-is) e o sistema que se pretende obter (to-be).
- ▶ O sistema as-is apresenta problemas, limitações e deficiências. O sistema to-be tem a intenção de tratar esses problemas usando oportunidades oferecidas pela tecnologia.
- ▶ Assim, para que o sistema *to-be* seja bem-sucedido, o sistema de software a ser desenvolvido (software *to-be*) precisa cooperar efetivamente com seu ambiente (pessoas, dispositivos e outros sistemas de software existentes).

Análise de Requisitos



Análise de Requisitos

- ▶ • **Por quê?** As razões para se desenvolver um novo sistema computacional (software to-be) devem ser explicitadas em termos de objetivos a serem satisfeitos.
- ▶ • **O que?** Esta dimensão preocupa-se com os serviços (requisitos funcionais) que o novo sistema computacional deve prover para satisfazer os objetivos. Esses serviços devem satisfazer restrições (requisitos não funcionais), garantir regras de negócio e apoiam-se em suposições.
- ▶ **Quem?** Alguns serviços serão implementados pelo software to-be, enquanto outros serão realizados por procedimentos manuais, operações de dispositivos ou por outros produtos de software já existentes.



Análise de Requisitos

- ▶ Para responder a essas perguntas, é necessário um entendimento mais profundo do domínio do problema, do negócio a ser apoiado, das limitações do sistema atual e dos requisitos para o novo sistema.
- ▶ Requisitos de cliente são insuficientes para isso. Eles resultam diretamente da atividade de levantamento de requisitos, sendo tipicamente descritos em linguagem natural, em um baixo nível de detalhes.
- ▶ Assim, uma vez identificados os requisitos de cliente, é necessário detalhá-los, colocando-os no nível de descrição de requisitos de sistema.
- ▶ Este é o propósito da Análise de Requisitos.

Análise de Requisitos

- ▶ A correta derivação de requisitos de sistema a partir de requisitos de cliente é fundamental para assegurar que equívocos não serão arbitrariamente introduzidos pelos engenheiros de requisitos na especificação de requisitos.
- ▶ Durante a análise de requisitos, requisitos funcionais e não funcionais devem ser expressos em um nível de detalhes que permita guiar as etapas subsequentes do desenvolvimento de software (projeto, implementação e testes).
- ▶ Além disso, atenção especial deve ser dada às regras de negócio. Elas têm de ser capturadas e incorporadas às funcionalidades do sistema.

Análise de Requisitos

- ▶ Para descrever requisitos funcionais detalhadamente, é necessário produzir **modelos**, cada um deles capturando uma perspectiva diferente do sistema.
- ▶ A linguagem natural ainda é utilizada, mas em uma escala reduzida, circunscrita a descrições de certos modelos ou às definições em glossários ou dicionários de dados.
- ▶ Grande parte das informações tratadas na análise de requisitos funcionais é melhor comunicada por meio de diagramas do que por meio de texto.
- ▶ Assim, a modelagem é uma atividade essencial da análise de requisitos.



Análise de Requisitos

- ▶ A modelagem de objetivos pode ser empregada para tratar a dimensão “Por quê?”.
- ▶ Um modelo de objetivo procura capturar precisamente os objetivos a serem satisfeitos pelo sistema to-be, suas ramificações em termos de subobjetivos, como esses objetivos interagem (conflitos e sinergias) e como eles estão alinhados a objetivos de negócio.
- ▶ A modelagem conceitual visa definir em detalhes as funções requeridas pelo sistema e o conhecimento necessário para realizá-las.
- ▶ O produto principal da modelagem conceitual é o modelo conceitual do sistema.



Análise de Requisitos

- ▶ O esquema ou modelo conceitual de um sistema captura as funções e informações que o sistema deve prover e gerenciar.
- ▶ Ele deve ser concebido com foco no domínio do problema e não no domínio da solução, mas deve tratar tanto uma visão externa do sistema (como o sistema é percebido pelos usuários) quanto uma visão interna do mesmo (como as abstrações do domínio são representadas e relacionadas).
- ▶ O produto de trabalho principal da análise de requisitos é o **Documento de Especificação de Requisitos**. Esse documento deve conter os requisitos funcionais e não funcionais descritos em nível de requisitos de sistema.

Análise de Requisitos

- ▶ Os requisitos funcionais de sistema são descritos por um conjunto de modelos inter-relacionados. Os requisitos não funcionais de sistema detalham os requisitos não funcionais de usuário, adicionando a eles critérios de aceitação.
- ▶ É importante apontar que há uma forte dependência entre os métodos e técnicas usados na modelagem conceitual e o paradigma de desenvolvimento adotado. Assim, para a modelagem conceitual de requisitos funcionais é necessário escolher um paradigma de desenvolvimento a partir do qual os modelos serão construídos.
- ▶ O paradigma orientado a objetos, por exemplo, fornece um conjunto de elementos de modelagem que permite modelar um sistema como sendo composto de objetos organizados em classes que se comunicam entre si por meio de troca de mensagens. Uma classe define as propriedades (atributos, relacionamentos e operações) que todos os objetos dela podem possuir.

Análise de Objetivos

- ▶ Os requisitos funcionais descrevem o que um sistema deve fazer. Os objetivos explicam o porquê.
- ▶ Ao se deixar claros os objetivos subjacentes aos requisitos (tanto funcionais quanto não funcionais), ganha-se um critério de completude/pertinência dos requisitos, abre-se espaço para exploração de alternativas e detecção de conflitos entre stakeholders, garantindo maior estabilidade.
- ▶ Um objetivo é uma declaração de uma intenção que o sistema to-be como um todo deve satisfazer por meio de cooperação entre seus agentes (pessoas desempenhando papéis específicos, dispositivos, outros sistemas de software existentes e o novo software to-be).
- ▶ Seja o caso do seguinte objetivo a ser satisfeito por um sistema de biblioteca: “Exemplares de livros devem ser devolvidos dentro do prazo previsto”. Ele envolve os usuários (que devem devolver os exemplares) e o sistema de informação da biblioteca (que deve acompanhar os empréstimos e emitir lembretes).

Análise de Objetivos

- ▶ Objetivos podem ser descritos em diferentes níveis de abstração.
- ▶ Em níveis mais altos, ditos objetivos estratégicos, os objetivos são mais gerais e relacionados ao negócio ou à organização.
- ▶ Em níveis mais baixos, ditos objetivos táticos ou técnicos, os objetivos são mais específicos e relacionados ao sistema ou a opções de projeto do sistema.
- ▶ A diferença entre níveis de objetivos sugere o uso de um mecanismo de estruturação de objetivos baseado em ligações de contribuição entre objetivos: um objetivo mais geral pode ser refinado em objetivos mais específicos (subobjetivos) contribuindo para a satisfação do primeiro. Esse mecanismo é a base para a modelagem de objetivos.

Análise de Objetivos

- ▶ A análise de objetivos inicia com a identificação dos interessados e seus objetivos.
- ▶ Objetivos estratégicos são obtidos junto aos interessados e refinados em objetivos táticos. Esse refinamento é capturado em um modelo de objetivos.
- ▶ Uma vez produzido um modelo de objetivos, o mesmo pode ser usado como insumo para a descrição da funcionalidade do sistema por meio de modelos de casos de uso.
- ▶ Abordagens de ER que têm como ponto de partida objetivos são ditas abordagens de *Engenharia de Requisitos Orientadas a Objetivos (Goal-Oriented Requirements Engineering - GORE)*.



Modelagem Conceitual

- ▶ Todo sistema incorpora um esquema conceitual. Assim, para que o desenvolvimento de um sistema seja bem-sucedido, é necessário explicitar esse esquema. Esse é o propósito da modelagem conceitual.
- ▶ O esquema conceitual de um sistema de informação é a especificação de seus requisitos funcionais.
- ▶ Um sistema de informação é um sistema projetado para coletar, armazenar, processar e distribuir informação sobre o estado de um domínio. Assim, um sistema dessa natureza tem tipicamente três propósitos ou funções principais:



Modelagem Conceitual

- ▶ • **Função de Memória:** sob esta ótica, o objetivo de um sistema de informação é manter uma representação interna do estado do domínio. O estado do domínio geralmente é alterado com frequência e de muitas maneiras. O sistema precisa acompanhar essas alterações e atualizar sua representação interna adequadamente.
- ▶ • **Função Informativa:** o sistema deve prover aos usuários informações sobre o estado do domínio. A função informativa não altera o estado do domínio; ela apenas provê a informação solicitada pelos usuários.
- ▶ • **Função Ativa:** o sistema pode realizar ações que modificam o estado do domínio. Para realizar essa função, o sistema precisa conhecer as ações que ele pode tomar, quando elas devem ser tomadas e como elas vão afetar o estado do domínio.



Modelagem Conceitual

- ▶ Para ser capaz de realizar suas funções, um sistema tem de ter um conhecimento geral sobre o domínio da aplicação e sobre as funcionalidades que ele deve executar. Esse conhecimento é denominado de esquema conceitual.
- ▶ O modelo conceitual de um sistema é tipicamente composto de vários modelos, cada um deles enfocando uma perspectiva diferente, mas guardando alguma relação com outros modelos. Os diferentes tipos de modelos conceituais podem ser agrupados em duas grandes categorias:
 - ▶ Modelos Estruturais
 - ▶ Modelos Comportamentais



Modelagem Conceitual

- ▶ **Modelos Estruturais:** procuram capturar os principais conceitos do domínio, suas relações e propriedades, que são relevantes para o sistema que se está desenvolvendo. Abstração é um mecanismo chave e a definição do que é relevante é definido com base no propósito do sistema. Dentre os tipos de modelos estruturais, destacam-se o Modelo de Entidades e Relacionamentos e o Diagrama de Classes na Orientação a Objetos.

- ▶ **Modelos Comportamentais:** especificam as ações que o sistema pode realizar e as mudanças válidas no estado do domínio. Tipos de modelos comportamentais bastante utilizados no desenvolvimento orientado a objetos incluem Modelos de Casos de Uso e Modelos Dinâmicos, tais como Diagramas de Transição de Estados e Diagramas de Interação.



Linguagem de Modelagem Unificada

- ▶ A Linguagem de Modelagem Unificada (Unified Modeling Language - UML) é uma linguagem gráfica padrão para especificar, visualizar, documentar e construir artefatos de sistemas de software.
- ▶ Ela é independente do processo de software a ser usado, ainda que seja mais adequada a processos de desenvolvimento orientados a objetos.
- ▶ No contexto da Engenharia de Requisitos, a UML provê diversos diagramas que podem ser usados na modelagem de requisitos, tanto segundo a perspectiva estrutural quanto segundo a perspectiva comportamental:
 - ▶ • **Diagrama de Classes:** modela um conjunto de classes e seus relacionamentos. Diagramas de classes proveem uma visão estática da estrutura de um sistema e, portanto, são usados na modelagem conceitual estrutural.

Linguagem de Modelagem Unificada

- ▶ • **Diagrama de Casos de Uso:** mostra um conjunto de casos de uso e atores e seus relacionamentos. Os casos de uso descrevem a funcionalidade do sistema percebida pelos atores externos. Um ator interage com o sistema, podendo ser um usuário humano, dispositivo de hardware ou outro sistema. Diagramas de casos de uso proveem uma visão das funcionalidades do sistema.

- ▶ • **Diagrama de Gráfico de Estados** (ou simplesmente Diagrama de Estados): mostra os estados pelos quais os objetos de uma classe específica podem passar ao longo de suas vidas, em resposta a estímulos recebidos, juntamente com suas ações. Os diagramas de estados proveem uma visão dinâmica dos objetos de uma classe, sendo importantes para modelar o comportamento de objetos de uma classe em resposta à ocorrência de eventos.

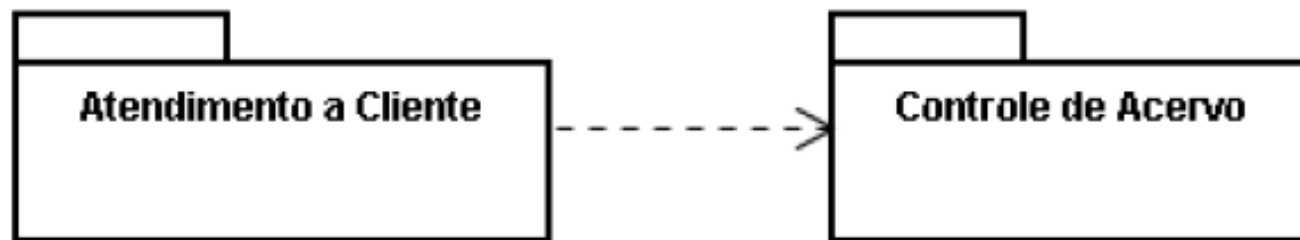


Linguagem de Modelagem Unificada

- ▶ • **Diagrama de Atividades:** mostra a estrutura de um processo. Provê uma visão dinâmica do sistema (ou de uma porção do sistema) e pode ser usado tanto para modelar processos de negócio quanto para modelar funções do sistema. Um diagrama de atividades dá ênfase ao fluxo de controle entre objetos.
- ▶ • **Diagrama de Interação:** mostra um conjunto de objetos interagindo, incluindo as mensagens que podem ser trocadas entre eles. Provê uma visão dinâmica do comportamento de um sistema ou de uma porção do sistema. Há dois tipos de diagramas de interação:
 - ▶ • **Diagrama de Sequência:** dá ênfase à ordenação temporal das mensagens.
 - ▶ • **Diagrama de Comunicação:** tem o mesmo propósito do diagrama de sequência, apresentando, contudo, ênfase na organização estrutural dos objetos que enviam ou recebem mensagens.

Linguagem de Modelagem Unificada

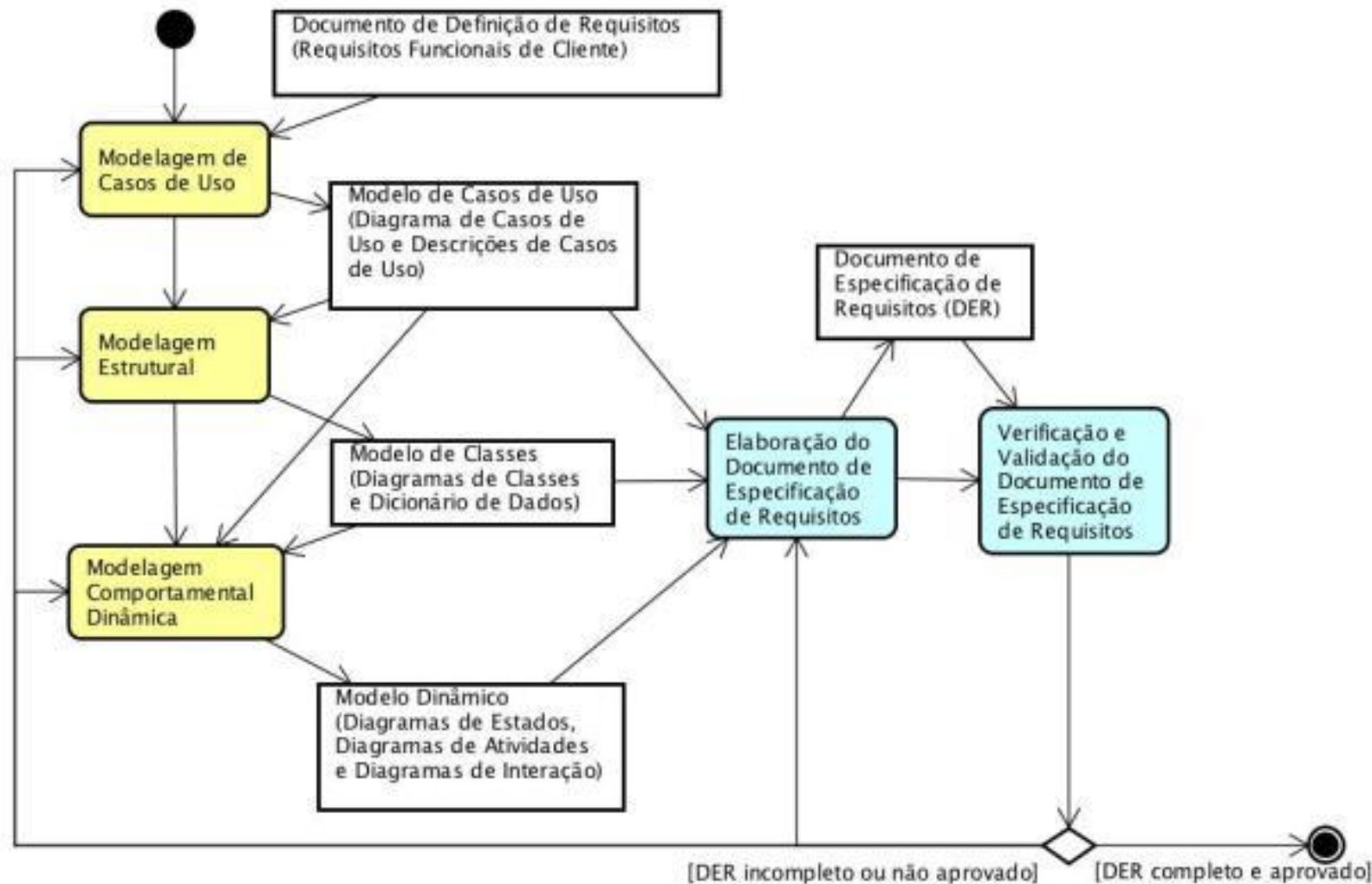
- ▶ No contexto da Engenharia de Requisitos, além dos diagramas anteriormente citados, merecem atenção também os **diagramas de pacotes**.
- ▶ Na UML, pacote é um mecanismo de propósito geral usado para organizar elementos de modelagem em grupos. Assim, um diagrama de pacotes mostra a decomposição de um modelo em unidades menores e suas dependências.



Um Método de Análise de Requisitos Funcionais

- ▶ Uma vez que tipicamente diversos modelos do sistema são produzidos, surgem algumas importantes questões: Que modelos produzir? Em que sequência? Quais as relações existentes entre esses modelos? Estas questões podem ser parcialmente respondidas pela adoção de um método de análise.
- ▶ Um método é composto por uma linguagem estabelecendo a notação a ser usada na elaboração dos artefatos a serem produzidos e de um processo descrevendo que artefatos construir e como construí-los.
- ▶ O método utilizado neste semestre adota a Linguagem de Modelagem Unificada (Unified Modeling Language - UML) como linguagem de modelagem e prescreve o seguinte processo:

Um Método de Análise de Requisitos Funcionais



Um Método de Análise de Requisitos Funcionais

- ▶ Se uma abordagem baseada em objetivos (Goal-Oriented Requirements Engineering - GORE) for adotada, uma nova atividade deve ser inserida no início do processo: a Análise de Objetivos.
- ▶ Em uma abordagem convencional, o primeiro modelo a ser construído é o modelo de casos de uso. Sua escolha como primeiro modelo deve-se ao fato do modelo de casos de uso ser muito simples e, portanto, passível de compreensão tanto por desenvolvedores - analistas, projetistas, programadores e testadores - como pela comunidade usuária - clientes, usuários e demais interessados.
- ▶ O modelo de casos de uso é um modelo comportamental, mostrando as funções do sistema de maneira estática. Ele é composto de dois artefatos: os diagramas de casos de uso e as descrições de casos de uso.

Um Método de Análise de Requisitos Funcionais

- ▶ O diagrama de casos de uso descreve graficamente o sistema, seu ambiente e como sistema e ambiente se relacionam. Assim, ele descreve o sistema segundo uma perspectiva externa.
- ▶ As descrições dos casos de uso descrevem o passo a passo para a realização dos casos de uso e são essencialmente textuais. Elas tratam de como os casos de uso são realizados internamente, complementando os diagramas.
- ▶ Tomando por base casos de uso e suas descrições, é possível passar à modelagem conceitual estrutural, quando os conceitos e relacionamentos envolvidos no domínio são capturados em um conjunto de diagramas de classes. Neste momento é importante definir, também, o significado dos conceitos e de suas propriedades, bem como restrições sobre eles. Essas definições são documentadas em um dicionário de dados do projeto.

Um Método de Análise de Requisitos Funcionais

- ▶ Algumas classes do modelo estrutural apresentam um comportamento dependente de seu estado.
- ▶ Para essas classes, é útil elaborar diagramas de estados, mostrando os estados pelos quais um objeto da classe pode passar ao longo de sua existência e os eventos que provocam transições de estados.
- ▶ Além disso, uma vez que casos de uso são representados apenas de maneira estática, pode ser útil também mostrar a dinâmica de um caso de uso, representando como objetos do diagrama de classes interagem para realizar um caso de uso. Os diagramas de interação são elaborados com este propósito.

Um Método de Análise de Requisitos Funcionais

- ▶ Finalmente, pode ser útil representar os passos de um caso de uso na forma de um diagrama mostrando objetos gerados e os atores envolvidos em cada passo. Para tal, diagramas de atividades podem ser elaborados.
- ▶ Apesar da figura sugerir que os passos do método são sequenciais, na prática, isso não ocorre.
- ▶ Paralelamente à modelagem de casos de uso, pode-se iniciar a modelagem conceitual estrutural. Os diagramas de atividade podem também ser elaborados em conjunto com a definição dos casos de uso, de maneira a complementar a descrição de casos de uso específicos. Diagramas de estado e de interação requerem a definição preliminar de casos de uso e classes. Contudo, podem ter sido definidos apenas alguns casos de uso e classes (e não todos eles) para já iniciar a elaboração desses diagramas.

Um Método de Análise de Requisitos Funcionais

- ▶ Paralelamente a todas as atividades de modelagem, o documento de especificação de requisitos deve ir sendo elaborado.
- ▶ A verificação e a validação também podem ser feitas por partes e não para o documento como um todo. Por exemplo, é bastante comum validar primeiro somente os casos de uso.
- ▶ Verificações de consistência, tais como as feitas entre casos de uso e classes, ou casos de uso, diagramas de interação e diagramas de classes, podem ser feitas separadamente uma das outras.
- ▶ Assim, as atividades de documentação, verificação e validação são atividades contínuas que ocorrem paralelamente à modelagem conceitual.

Especificação de Requisitos Não Funcionais

- ▶ Assim como os requisitos funcionais precisam ser especificados em detalhes, o mesmo acontece com os requisitos não funcionais.
- ▶ Para os atributos de qualidade considerados prioritários, o analista deve trabalhar no sentido de especificá-los de modo que eles se tornem mensuráveis e, por conseguinte, testáveis.
- ▶ Para cada atributo de qualidade, devem-se definir as medidas a serem usadas, indicando a unidade da medida e sua escala, e os valores mínimo, alvo e máximo.
- ▶ Pode-se, ainda, perguntar aos usuários o que constituiria um valor inaceitável para o atributo e definir testes que tentem forçar o sistema a demonstrar tais características.

Especificação de Requisitos Não Funcionais

- ▶ Ao se estabelecer uma escala de medição e os valores aceitáveis, o requisito é transformado de uma intenção vaga, e até certo ponto ambígua, em um requisito mensurável e bem formado.
- ▶ Estabelecida uma escala, pode-se perguntar ao usuário o que é considerado uma falha em atender ao requisito, de modo a definir o critério de aceitação do mesmo.
- ▶ Assim, na especificação de requisitos de sistema, é importante transformar um requisito de usuário em um requisito mensurável, adicionando a ele um critério de aceitação.

Especificação de Requisitos Não Funcionais

- ▶ A ISO/IEC 25023 - Medidas de Qualidade de Produtos de Software e Sistemas (ISO/IEC, 2016), ou a sua antecessora, a ISO/IEC 9126, pode ser uma boa fonte de medidas.

RNF01 – A funcionalidade “Efetuar Locação de Item” deve ser fácil de aprender.		
Medida: Facilidade de Aprendizagem de função (<i>Ease of function learn</i>) (ISO/IEC, 2003a)	Descrição:	Facilidade de aprender a realizar uma tarefa em uso.
	Propósito:	Quanto tempo o usuário leva para aprender a realizar uma tarefa especificada eficientemente?
	Método de Aplicação:	Observar o comportamento do usuário desde quando ele começa a aprender até quando ele começa a operar eficientemente.
	Medição:	$T =$ soma do tempo de operação do usuário até que ele consiga realizar a tarefa em um tempo especificado (tempo requerido para aprender a operação para realizar a tarefa).
Critério de Aceitação:	$T \leq 15$ minutos, considerando que o usuário está operando o sistema eficientemente quando a tarefa “Efetuar Locação” é realizada em um tempo inferior a 2 minutos.	

O Documento de Especificação de Requisitos

- ▶ Os requisitos de sistema, assim como foi o caso dos requisitos de cliente, têm de ser especificados em um documento, de modo a poderem ser verificados e validados e posteriormente usados como base para as atividades subsequentes do desenvolvimento de software.
- ▶ O Documento de Especificação de Requisitos tem como propósito registrar os requisitos escritos a partir da perspectiva do desenvolvedor e, portanto, deve incluir os vários modelos conceituais desenvolvidos, bem como a especificação dos requisitos não funcionais detalhados.
- ▶ Diferentes formatos podem ser propostos para documentos de especificação requisitos, bem como mais de um documento pode ser usado para documentar os requisitos de sistema.

O Documento de Especificação de Requisitos

- ▶ Usaremos aqui um único documento, contendo as seguintes informações:
- ▶ • Introdução: breve introdução ao documento, descrevendo seu propósito e estrutura.
- ▶ • Modelo de Casos de Uso: apresenta o modelo de casos de uso do sistema, incluindo os diagramas de casos de uso e as descrições de casos de uso associadas.
- ▶ • Modelo Estrutural: apresenta o modelo conceitual estrutural do sistema, incluindo os diagramas de classes do sistema.
- ▶ • Modelo Dinâmico: apresenta os modelos comportamentais dinâmicos do sistema, incluindo os diagramas de estados, diagramas de interação e diagramas de atividades.
- ▶ • Dicionário do Projeto: apresenta as definições dos principais conceitos capturados pelos diversos modelos e restrições de integridade a serem consideradas, servindo como um glossário do projeto.
- ▶ • Especificação dos Requisitos Não Funcionais: apresenta os requisitos não funcionais descritos no nível de sistema, o que inclui critérios de aceitação.

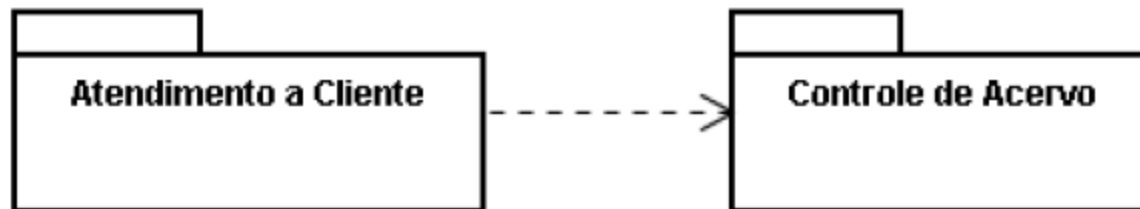


O Documento de Especificação de Requisitos

- ▶ É importante frisar que dificilmente um sistema é simples o bastante para ser modelado como um todo. Quase sempre é útil dividir um sistema em unidades menores, mais fáceis de serem gerenciáveis, ditas **subsistemas**.
- ▶ É útil organizar a especificação de requisitos por subsistemas e, portanto, cada uma das seções propostas acima pode ser subdividida por subsistemas.
- ▶ O agrupamento de elementos de modelo em subsistemas serve basicamente para definir uma representação concisa capaz de orientar o leitor na leitura dos modelos estruturais, podendo ser útil também para a organização de grupos de trabalho em projetos extensos.

O Documento de Especificação de Requisitos

- ▶ A base principal para a identificação de subsistemas é a complexidade do domínio do problema. Através da identificação e agrupamento de elementos de modelo em subsistemas, é possível controlar a visibilidade do leitor e, assim, tornar os modelos mais compreensíveis.
- ▶ A UML provê um tipo principal de item de agrupamento, denominado pacote, que é um mecanismo de propósito geral para a organização de elementos da modelagem em grupos.



That's all Folks!



nemo