Módulo 06

# Parte 01 – Conceito de modelo

Durante a documentação de requisitos o engenheiro de requisitos documenta 3 diferentes tipos de requisitos:

1. **Metas**

Metas revelam as intenções das partes interessadas.

1. **Casos de uso e cenários**

Casos de uso e cenário descrevem sequências de utilização do sistema.

1. **Requisitos do sistema**

Requisitos de sistema descrevem as funções ou qualidades que o sistema deve possuir.

Requisitos servem como insumo para trabalhos posteriores da engenharia de requisitos.

Documentar requisitos usando modelos

Apesar de serem documentados na maioria das vezes em linguagem natural, os requisitos podem ser documentados utilizando modelos conceituais.

**Modelo é uma abstração da realidade existente ou que ainda será criada.**

Modelo tem 3 propriedades:

1. Representação da realidade

* Leva em consideração apenas os elementos relevantes.
* Pode ser prescritivo ou descritivo.

1. Redução da realidade

* Não considera todos os aspectos da realidade.
* Esta propriedade é suportada pela seleção e compreensão.
* Na seleção apenas certos aspectos são modelados.
* Na compreensão resumem-se os aspectos do sistema.

1. Pragmatismo

* Um modelo é sempre criado comum propósito.
* Um modelo é sempre criado dentro de um contexto.

Linguagens de modelagem são utilizadas para construir modelos conceituais

Constituída de:

Sintaxe são elementos de modelagem que serão utilizados e também as regras de utilização dos mesmos.

Semântica é o estudo do significado de cada elemento.

**Linguagens de modelagem**

**Formais**

* Maior rigidez das definições formais

**Semiformais**

* Certa medida de rigidez das definições formais

**Informais**

* Menor rigidez das definições formais.

Modelos conceituais são chamados de modelos de requisitos.

A UML (Unified Modeling Language) é utilizada para criar modelos de requisitos.

Modelos convencionais são utilizados para documentar soluções escolhidas durante o desenvolvimento do sistema enquanto que modelos de requisitos descrevem aspectos do problema analisado.

Informações são melhor percebidas e memorizadas quando o ser-humano as representa graficamente.

Modelo de requisito tem um enfoque estritamente definido.

O uso de modelos conceituais e linguagem natural permite que o engenheiro de requisitos minimize as deficiências de cada uma das técnicas.

# Parte 2 – Modelo de Metas

Partes interessadas declaram suas intenções ou metas durante a elicitação. Essas metas devem ser documentadas.

Documentar metas: descrevem propriedade do sistema será desenvolvido ou do projeto de desenvolvimento do sistema.

Metas podem ser documentadas através de modelos de metas.

A técnica para criar modelos de metas se chama Árvores E/OU. Com Árvores E/Ou o engenheiro de requisitos documenta decomposições hierárquicas. Com esta técnica representa-se a hierarquia através das ramificações.

Árvores E/OU permitem utilizar dois tipos de relações de decomposições:

Decomposição OU Decomposição E

Na decomposição OU para que super. meta seja atingida basta que pelos menos uma submeta seja atingida.

Na decomposição E para que super meta seja atingidas TODAS as submetas devem ser atingidas.

# Parte 03 – Casos de Uso

Proposto por Jacobson para documentar funcionalidades do sistema.

1. Diagrama de casos de uso

* São modelos simples para documentar funções do sistema do ponto de vista do usuário.
* Elementos do diagrama de casos de uso:
  1. **Caso de uso:** Utiliza-se elipses para representar o caso de uso. É necessário informar o nome do caso de uso. Um caso de usos representa uma funcionalidade do sistema.
  2. **Ator (Pessoa):** O ator está fora do limite do sistema e representa pessoas ou sistemas que interagem com o sistema modelado.
  3. **Ator (Sistema)**: Se o ator é uma pessoa, então utiliza-se a imagem do boneco. Se o ator é um sistema utiliza-se o boneco ou um retângulo.
  4. **Limite do sistema**: O limite do sistema separa a parte do caso de uso das partes que estão fora do limite do sistema.
  5. **Relacionamento de Extensão (Extend):** Um relacionamento extend de um caso de uso para outro caso de uso indica que o caso de uso origem pode ser acrescentado para descrever o comportamento de outro caos de uso
  6. **Relacionamento de Inclusão (Include):** Um relacionamento de include de um caso de uso A para o caso de uso B indica que B é essencial para o comportamento de A. Pode-se dizer que B é parte de A.
  7. **Relacionamento entre atores e caso de uso:** É o relacionamento que descreve a interação entre atores e caso de uso.
  8. **Generalizações:** É um tipo de relacionamento entre atores e casos de uso. Neste tipo de relacionamento casos de uso ou atores especialistas herdam propriedades de casos de uso ou autores generalistas.

1. Especificação de casos de uso:
   1. Descrevem a interação sistemática entre o caso de uso e ator. Documentam essas interações textualmente através de templates.
2. **Template para documentação textual de caso de uso**
3. **Identificação:** Identificação única do caso de uso.
4. **Nome:** Nome caso de uso.
5. **Autor:** Nome de quem criou o caso de uso.
6. **Prioridade:** Importância do caso de uso.
7. **Criticidade:** O quanto de prejuízo o caso de uso pode causar.
8. **Origem:** Onde (parte interessada, sistema ou documentação) o caso de uso foi elicitado.
9. **Responsável:** Parte interessadas que responde pelo caso de uso.
10. **Descrição:** Breve texto que define o caso de uso.
11. **Gatilho:** Aquilo que dispara a execução do caso de uso.
12. **Atores:** Lista de participantes no caso de uso.
13. **Pré-condições:** Estados do sistema produzidos após a execução do caso de uso.
14. **Pós-condições:** Estados do sistema produzidos após a execução do caso de uso.
15. **Resultado:** Aquilo que é produzido ao fim da execução do caso de uso.
16. **Fluxo/Cenário básico:** Descrição do cenário/fluxo principal do caso de uso
17. **Fluxos/Cenários alternativos:** Descrição dos cenários/ fluxos alternativos do caso de uso.
18. **Fluxos/Cenários de exceção:** Descrição dos cenários/ fluxos de exceção do caso de uso.
19. **Qualidades:** Lista dos requisitos de qualidade relacionadas ao caso de uso.

* Itens 1 e 2 são atributos para identificação única do caso de uso.
* Itens 3 a 7 são atributos de gerenciamento.
* Item 8 é um atributo para descrição do caso de uso.
* Itens 9 a 17 são atributos específicos do caso de uso.

# Parte 04 – Três Perspectivas sobre Requisitos

* **Perspectiva Estrutural ou de dados:** Documenta-se nessa perspectiva as estruturas de dados de entrada e saída, aspectos estático-estruturais das relações de uso e a dependência do sistema no contexto.
* **Perspectiva Funcional:** Documenta-se nessa perspectiva a informação que está no contexto e é manipulada pelo sistema e os dados enviados do sistema para o contexto.
* **Perspectiva Comportamental:** Documenta-se nessa perspectiva a integração do sistema com o contexto baseada nos estados, ou seja, aquilo que acontece com o sistema (mudança de estado) em função do que acontece no seu contexto.

**Exemplos de diagramas para cada perspectivas:**

* **Estrutural ou de dados**: Diagrama de classes de UML (ESTRUTURA DE DADOS)
* **Funcional:** Diagrama de atividades de UML (DADOS OU INFORMAÇÃO)
* **Comportamental:** Diagrama de estados (ESTADOS)

# Parte 05 – Modelagem de requisitos na perspectiva Estrutural

* Modelos:
  + **DER – Diagrama de Entidade e Relacionamento**

**Tipo Entidade:** Define o conjunto de entidades dentro de um universo. Neste universo os objetos possuem a mesmas propriedades.

**Tipo Relação:** Define a abstração a partir de uma característica concreta.

**Atributo:** Define as propriedades do tipo de entidade ou do tipo relação

**Cardinalidades:** Define o número de instâncias de relacionamento dos quais uma entidade faz parte. Se nenhuma cardinalidade foi definida então admite-se que no mínimo zero entidades participam de um relacionamento. A definição de cardinalidades, portanto, limita o número de instâncias em um relacionamento.

* + **Diagrama de Classes UML**

**Classes:** Define uma estrutura que abstrai um conjunto de objetos com características similares. É representado graficamente por um retângulo.

**Relacionamentos**

**Associação:** Define o relacionamento entre classes. É representado por uma linha. Associações pode ter nome e multiplicidades em cada ponta da associação. Multiplicidades são definições de instaciamento de uma classe. Uma multiplicidade define quantas instancias de uma classe pode se associar a instâncias de outra classe.

**Composição:** Define um tipo de relacionamento entre uma parte e seu todo. É um tipo de associação onde uma parte não pode existir sem o todo.

**Agregação:** Define o tipo de relacionamento entre uma parte e seu todo. E um topo de associação menos forte que a composição.

**Generalização:** Define um tipo de relacionamento entre uma classe mais específica, também chamada de subclasse, e uma classe mais geral, também chamada de superclasse. Neste caso a subclasse herda todas as propriedades da superclasse;

**Diagrama de classes**

Maior poder descritivo

* Diagrama de classes

Menor poder descritivo

* DER

# Parte 06 – Modelagem de requisitos na perspectiva Funcional

* Modelos

**DFD -Diagrama de fluxos de dados**

* **Processo:** Define as funções de um sistema que transformam os dados. Um processo recebe dados de entrada processa e gera dados de saída. O como o dado é transformado não é demonstrado no DFD.
* **Depósito de dados:** Define dados persistidos. Processos acessam depósitos de dados para realizar uma leitura ou gravação de dados.
* **Fluxo de dados:** Define a movimentação dos dados entre processos, entidades e depósitos de dados. Em um DFD apenas os fluxos de dados mais importantes são modelados.
* **Entidade externa (fontes Destino de dados):** Define os objetos (pessoas, grupos de pessoas, departamentos, organizações e sistemas) que estão no ambiente e trocam informações com o sistema. Uma entidade pode ser fonte, quando fornece dados ao sistema, ou destino, quando recebe dados do sistema

**Diagrama de Atividades de UML**

* **Ação/ Atividade:** Descreve a execução de uma ação
* **Nó de início:** Descreve um evento que inicia a execução do diagrama de atividades.
* **Nó de término:** Descreve o fim da execução do diagrama de atividades.
* **Fluxo de controle:** Descreve o fluxo de controle.
* **Nó de decisão:** Descreve a condição que desencadeia fluxo alternativos de controle
* **Barra de sincronização:** Descreve a execução concorrente de objetos.

DFD versus Diagrama de Atividades UML

**Mais detalhes**

* Diagrama Atividades UML
* Inclui atividades paralelas, simultâneas e decisões.

**Menos detalhes**

* DFD
* Inclui a transformação e o fluxo de dados.

# Parte 07 – Modelagem de requisitos na perspectiva Comportamental

* Modelos

**Statecharts**

* **Estado:** Define o momento no qual o sistema apresenta um comportamento e espera um evento para executar uma transição definida.
* **Transição:** Define a mudança de um estadão para outro estado. É desencadeada por um evento que ocorre em um determinado estado.
* **Estado inicial:** Define o início da transição de estados.
* **Estado Final:** Define o fim da transição de estados.

**Diagrama de estados UML**

* **Estado:** Define o momento no qual o sistema apresenta um comportamento e espera um evento para executar uma transição definida.
* **Transição:** Define a mudança de um estadão para outro estado. É desencadeada por um evento que ocorre em um determinado estado.
* **Estado inicial:** Define o início da transição de estados.
* **Estado Final:** Define o fim da transição de estados.

**Ponto de entrada:** Descreve o pseudo-estado externamente visível imediatamente associado a um estado interno.

**Ponto de saída:** Descreve o pseudo-estado externamente visível tendo como origem um estado interno.