Engenharia de Software II

Bem vindos à Engenharia de Software II

Profa. Renata Neves

Engenharia de Software II

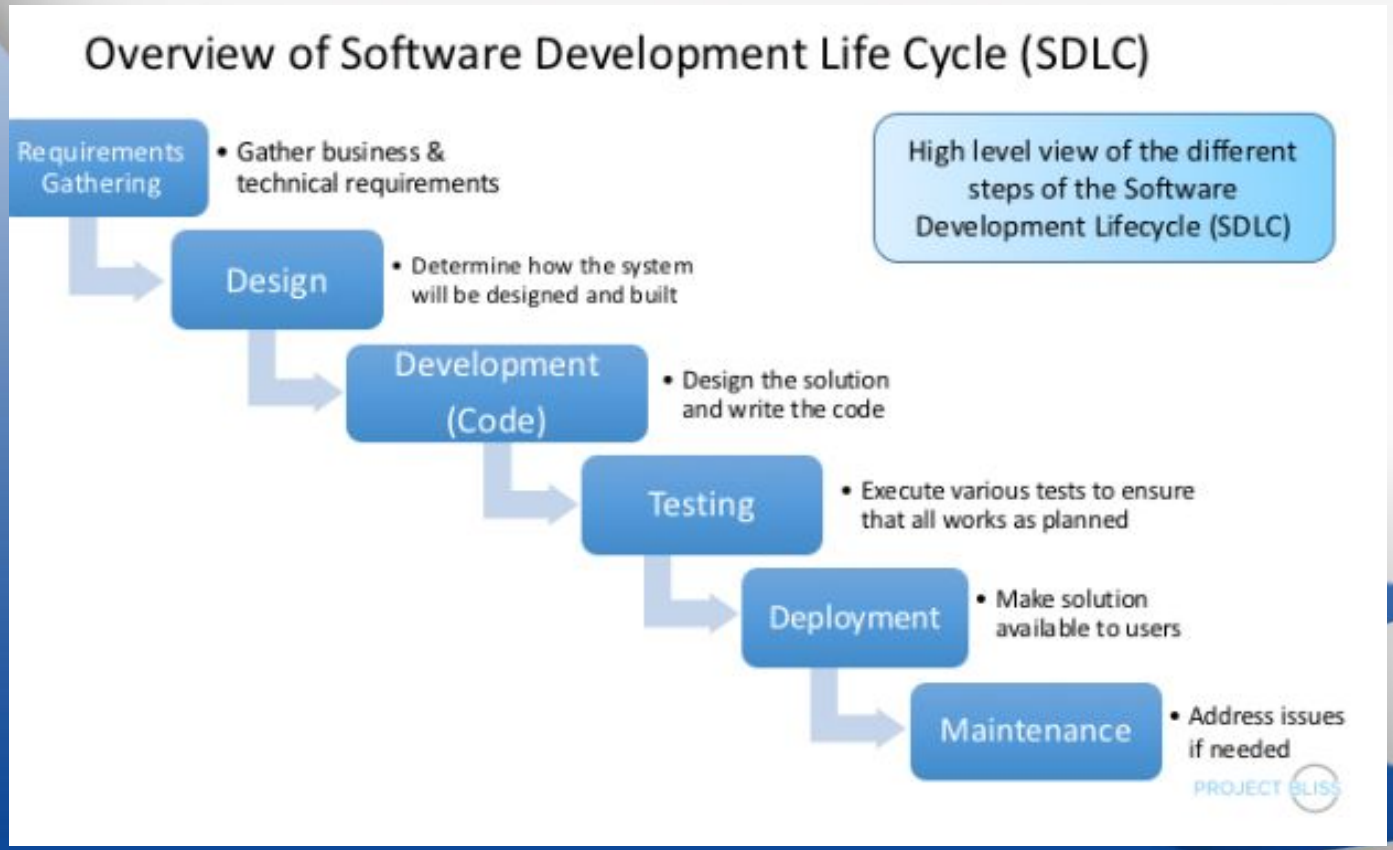
Análise e Modelagem de sistemas

- Ciclo de vida de desenvolvimento de sistemas

  SLDC - Systems Development Life Cycle.
- Aspectos da Modelagem
- Objetivos e filosofia do modelo de Análise
- Regras práticas de Análise

Engenharia de Software II

Ciclo de vida de desenvolvimento de sistemas ***SLDC - Systems Development Life Cycle.***



Engenharia de Software II

Requeriments/Requisitos - Segundo Pfleeger (2004), os requisitos são características de algo que o sistema será capaz de realizar para atingir determinado objetivo.
Para isso, a equipe de desenvolvimento deve questionar o cliente sobre a forma que o sistema deve ser para satisfazer as suas necessidades.

Em seguida a equipe deve registrar esses requisitos em um documento de tal maneira que clientes e desenvolvedores cheguem a um acordo sobre como o sistema deve ser.

Engenharia de Software II

Projeto de Software/Design – Os requisitos são um conjunto de problemas que o sistema deve resolver para o cliente e com base nesses requisitos, a equipe de desenvolvimento deve, antes de codificar, projetar uma solução que satisfaça as necessidades do cliente (PFLEEGER, 2004).

O design do projeto cria uma representação ou modelos de software, mas diferente do modelo de análise que enfoca a descrição das funcionalidades, dados e comportamento, o modelo de projeto fornece detalhes da estrutura de dados, arquitetura, interfaces, componentes de software necessários para implementação do sistema. (PRESSMAN, 2006)

Engenharia de Software II

Desenvolvimento – De acordo com Pressman (2006), a atividade de desenvolvimento consiste em transformar, através de técnicas de programação, o projeto de software em um produto executável e operacional.

Testes de sistema – Ainda citando este autor, os testes são um processo de execução de um programa que têm como objetivo revisar a codificação implementada e encontrar erros na estrutura do software.

Engenharia de Software II

Implantação – Implantação compreende a entrega do software para o cliente e a utilização do usuário final. Nesse momento, o cliente tem a oportunidade de utilizar a solução já implantada nas suas áreas de negócio.

Manutenção - Caso haja não conformidades, a equipe deve registrar as modificações necessárias para dar início à manutenção do software (PRESSMAN, 2006).

Engenharia de Software II

Aspectos da Modelagem

O que é ?

A modelagem da análise usa uma combinação de formas textuais e diagramaticas para demostrar os requisitos, funções e comportamento do sistema.
De modo que seja fácil para entendimento e auxilia quanto a correção, completeza e consistência do sistema.

Quem faz ?

O engenheiro de software (analista de sistemas), constrói um modelo utilizando os requisitos que foram explicados pelo cliente. (usuários, cliente e interessados no desenvolvimento do sistema).

Engenharia de Software II

Aspectos da Modelagem

Por que é importante?

Para entendimento da construção, para validação, ela representa os requisitos e funções em várias dimensões, para que o sistema corresponda a necessidade de negócio do cliente.

Para evitar e auxiliar na correção de divergências posteriores.

Para documentação do sistema que serve de base para posteriores adequadas manutenções, diminuindo assim riscos de erros, custos, prazos e aumentando a qualidade dos softwares desenvolvidos.

Ferreira, Renata (2002)

Engenharia de Software II

Aspectos da Modelagem

Quais os passos ?

Os **requisitos** são levantados e apresentados em vários diagramas diferentes.

A **modelagem baseada em cenários** representa o sistema sob o ponto de vista do usuário.

A **modelagem orientada a fluxo** fornece a indicação de como os objetos de dados são transformados em funções de processamento.

Engenharia de Software II

Aspectos da Modelagem

Quais os passos ?

A **modelagem baseada em classes** define objetos, atributos e relacionamentos.

A **modelagem comportamental** mostra os estados do sistemas e suas classes, e o impacto que os eventos tem.

Os **modelos são criados**, e refinados posteriormente quanto sua clareza, completeza e consistência.

O **modelo final de análise** (composto dos anteriores) é então validado pelos interessados.

Engenharia de Software II

Aspectos da Modelagem

Qual o produto do trabalho (artefato) ?

Uma grande variedade de diagramas podem ser escolhidos para o **Modelo de análise** (composto de modelos e diagramas UML)
Cada uma destas **representações fornece uma visão** de um ou mais elementos deste modelo.

Como tenho certeza que fiz corretamente ?
Os produtos da **modelagem da análise** devem ser revisados quanto à correção, completeza e consistência. Eles precisam refletir as necessidades de todos os interessados e estabelecer um fundamento com base no qual o projeto pode ser conduzido.
PRESSMAN (2006).

Engenharia de Software II

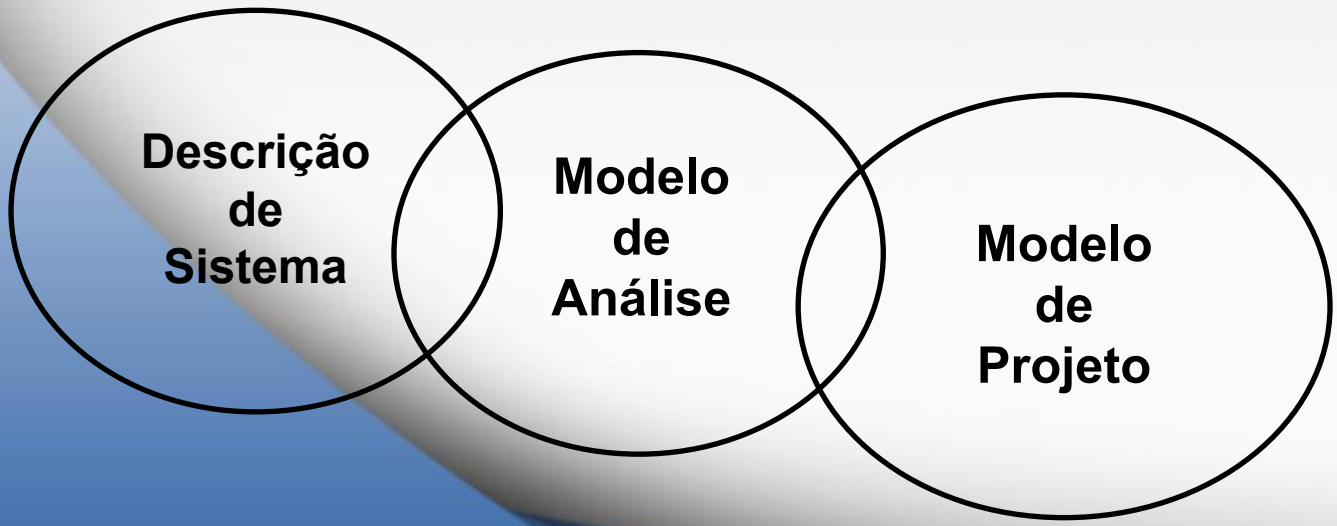
A Filosofia do Modelo de Análise

O **Modelo de análise** vence o espaço entre a **descrição do sistema** que descreve as funcionalidades do sistema, e como ela é conseguida, pela construção do software, hardware, dados, redes, pessoas e outros elementos.

E um **Modelo de projeto de software** descreve a arquitetura, interface do usuário e a estrutura do desenvolvimento técnico , os componentes da aplicação.

Engenharia de Software II

A Filosofia do Modelo de Análise



Engenharia de Software II

A Filosofia do Modelo de Análise

O **modelo de análise** tem que atingir **3 objetivos principais:**

1. Descrever o que o **cliente exige**;
2. Estabelecer a base para a criação de um **projeto** de software;
3. **Definir um conjunto de requisitos** que possam ser **validados** quando o software for construído.

Engenharia de Software II

Regras práticas da análise

Criamos modelos para > ***Obter um melhor entendimento da entidade real a ser construída.***

Forma lógica (software) – é diferente, ele precisa ser capaz de representar os dados que o software transforma, as características que os usuários desejam, e o comportamento do sistema. Resultado: **Modelo de análise**(composto de vários modelos)

Forma física (nós)– podemos construir modelos idênticos em forma e aspecto em escala menor.

Engenharia de Software II

Regras práticas da análise

O Modelo de análise produto da análise de domínio, representa os requisitos do cliente, mostrando o software em três domínios diferentes:

- **DOMÍNIO DA INFORMAÇÃO**

➔

- **DOMÍNIO FUNCIONAL**

➔

- **DOMÍNIO COMPORTAMENTAL**
-

"O PRIMEIRO PROBLEMA DOS ENGENHEIROS DE SOFTWARES EM QUALQUER SITUAÇÃO É DESCOBRIR QUAL É O VERDADEIRO PROBLEMA"

Engenharia de Software II

Regras práticas da análise

Princípios operacionais

Todos os métodos de análise estão relacionados com um conjunto de princípios operacionais:

Princípio 1: O domínio da informação de um problema precisa ser representado e entendido.

Princípio 2: As funções a serem desenvolvidas pelo software devem ser definidas.

Engenharia de Software II

Regras práticas da análise

Princípios operacionais

Princípio 3: O comportamento do software (como conseqüência de eventos externos precisa ser representado)

Princípio 4: Os modelos que mostram informação, função e comportamento devem ser particionados de um modo que revele detalhes em forma de camadas (hierarquias). “Dividir para conquistar”.

Princípio 5: A tarefa de análise deve ir da informação essencial até os detalhes de implementação.

Engenharia de Software II

Regras práticas da análise

Arlow e Neustadt [ARL02] sugerem algumas regras práticas valiosas que devem ser seguidas quando da criação do **Modelo de análise**. São elas:

- O modelo deve focalizar os requisitos que são visíveis no problema ou domínio do negócio. O nível de abstração deve ser relativamente alto. “Não se atole a detalhes**(NESTE MOMENTO)** que tentam explicar como o sistema funcionará.

- Cada elemento do modelo de análise deve contribuir para um entendimento global dos requisitos de software e fornecer uma visão do domínio da informação, função e comportamento do sistema.

Engenharia de Software II

Regras práticas da análise

Adie as considerações de modelos de infra-estrutura e outros modelos não funcionais **(NESTE MOMENTO)**.

- Por exemplo: **um modelo de banco de dados** será necessário, mas somente depois de a análise de domínio do problema ter sido completada e que deve ser feito.

- Minimize o acoplamento ao longo do sistema. É importante representar os relacionamentos entre as funções e classes.

- No entanto se o nível de relacionamentos for alto, deve ser reduzido.

Engenharia de Software II

Regras práticas da análise

- Certifique-se de que o modelo de análise tem valor para os interessados e envolvidos.Por exemplo: interessados no negócio devem avaliar o modelo para validar os requisitos, projetistas devem usar o modelo como base para o projeto. O pessoal da garantia de qualidade (QA) deve usar o modelo para ajudar a planejar os testes de aceitação do sistema.

Engenharia de Software II

Regras práticas da análise

- Mantenha o modelo tão simples quanto puder.
- Não acrescente diagramas que não serão utilizados e não agregue valor ao projeto. Por exemplo: diagramas que não fornecem informação nova, ou formulários notacionais completos, quando uma lista simples pode servir.

Engenharia de Software II