



# ENGENHARIA DE SOFTWARE

UNIDADE 2 – Modelos de Processos de Software  
(Aula 3 – Principais Tipologias)

Prof. Ivan Nunes da Silva

## *1. Introdução*

---

- Princípios são importantes para o sucesso no desenvolvimento de softwares, mas não são suficientes para obter produtos de qualidade.
- Engenharia de Software:
  - Alia **MÉTODOS, TÉCNICAS** e **FERRAMENTAS** apropriadas que auxiliam no processo de produção de softwares.



## 2. Princípios de Eng. de Software

### ● Rigor e Formalidade

- O processo de software é uma atividade criativa tendendo naturalmente à imprecisão.
- O rigor é a abordagem que produz produtos mais confiáveis pelo controle das variáveis envolvidas.

### ● Separação de Conceitos (Abstração)

- Separar conceitos permite-nos trabalhar com aspectos individuais e diferentes de um mesmo problema.
- Esta separação facilita o entendimento, focando a atenção em certas características mais significativas.

3

## 2. Princípios de Eng. de Software

### ● Modularidade

- Consiste na divisão de sistemas complexos em partes menores e mais simples (módulos) com características desejáveis (coesão e acoplamento).
- **Decomposição** é o ato de dividir um problema original em subproblemas recursivamente.
- **Composição** é o ato de juntar os elementos componentes de um problema até chegar ao sistema completo. Ajuda na manutenção do sistema.

### ● Antecipação de Mudanças

- Sistemas de softwares são desenvolvidos enquanto seus requisitos ainda não estão totalmente claros.
- Quando o sistema é finalmente liberado, novos requisitos podem ser descobertos e velhos requisitos atualizados através do “feedback” do cliente.

4

## 2. Princípios de Eng. de Software

### ● Incrementabilidade

- Caracteriza o processo em modo passo a passo, incrementalmente. O objetivo desejado é atingido por aproximações sucessivas.
- Útil quando os requisitos iniciais não foram todos obtidos antes do início do desenvolvimento da aplicação.

### ● Generalidade/Especialidade

- Soluções genéricas tendem a ser mais caras em termos de recursos e em tempo de desenvolvimento, ao contrário das soluções específicas.
- No processo de produção de software estas questões devem ser cuidadosamente analisadas.



5

## 3. Modelos de Processo de Software

- Um Processo de Software consiste em uma **série de atividades** que garantem, técnica e administrativamente, que o software possa ser desenvolvido de **maneira organizada, disciplinada e previsível**.
- Os modelos de **Processo de Software** determinam a forma como o sistema deverá ser implementado.
- É a **Primeira Metodologia** escolhida pelo engenheiro de software.
- Será usada **ao longo de todo o processo** de software.



6

### 3. Modelos de Processo de Software

#### Fase de Definição do Processo de Software:

- Focaliza “o **quê**” será desenvolvido:
  - Que informação vai ser processada?
  - Que função e desempenho são desejados?
  - Que comportamento pode ser esperado do sistema?
  - Que interfaces vão ser estabelecidas?
  - Que restrições de projeto existem?
  - Que critérios de validação são exigidos para definir um sistema bem sucedido?

7

### 3. Modelos de Processo de Software

#### Fase de Definição do Processo de Software:

- Focaliza “o *quê*” será desenvolvido:

Três tarefas principais ocorrem de alguma forma:

*Engenharia de Sistemas*

*Planejamento do Projeto de Software*

*Análise de Requisitos*

8

### 3. Modelos de Processo de Software

#### Fase de Definição do Processo de Software:

- Focaliza “**como**” o software será desenvolvido:
  - Como os dados vão ser estruturados?
  - Como a função vai ser implementada como uma arquitetura de software?
  - Como os detalhes procedimentais vão ser implementados?
  - Como as interfaces vão ser caracterizadas?
  - Como o projeto será traduzido em uma linguagem de programação?
  - Como os testes serão efetuados?



9

### 3. Modelos de Processo de Software

#### Fase de Definição do Processo de Software:

- Focaliza “como” o software será desenvolvido:

*Três tarefas técnicas específicas  
deverá ocorrer sempre:*

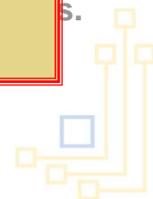
*Projeto do Software*

*Geração do Código*

*Teste do Software*

linguagem de programação.

- Como os testes serão efetuados.



10

### 3. Modelos de Processo de Software

#### Fase de Definição do Processo de Software:

- Focaliza as “**mudanças**” que ocorrerão depois que o software for liberado para uso operacional:

- A fase de manutenção reaplica os passos das fases de definição e desenvolvimento, mas faz isso no contexto de um software existente.

*As mudanças estão associadas com:*

- **correção** de erros.
- **adaptações** exigidas conforme o ambiente do software evolui.
- mudanças devido a **melhoramentos** ocorridos por alterações nos requisitos dos clientes.

11

### 3. Modelos de Processo de Software

#### Atividades de Apoio:

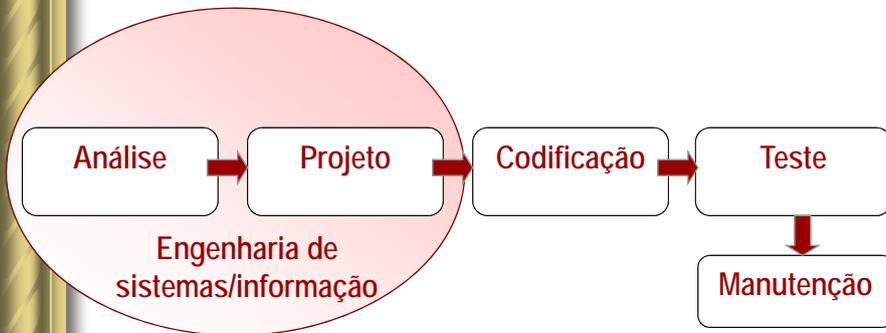
- As três fases genéricas do processo de software são complementadas por uma **série de atividades de apoio**.
- As atividades de apoio são aplicadas durante toda a engenharia do software.

*Atividades de apoio típicas nessa categoria são:*

- *Controle e Acompanhamento do Projeto de Software.*
- *Revisões Técnicas Formais.*
- *Garantia de Qualidade de Software.*
- *Gerenciamento de Configuração de Software.*
- *Preparação e Produção de Documentos.*
- *Gerenciamento de Reusabilidade.*
- *Medidas de Qualidade.*

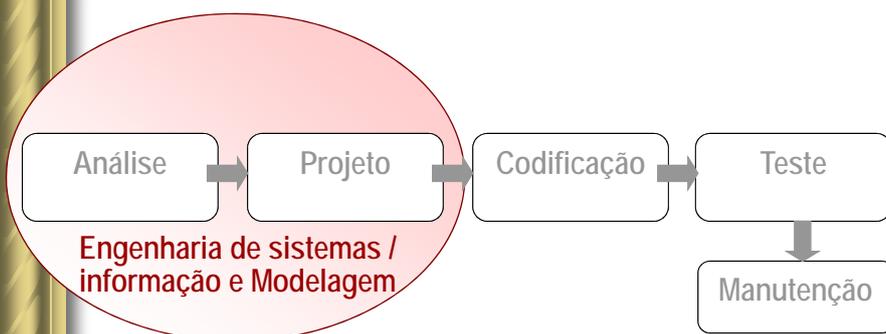
12

## 4. Modelo em Cascata



13

## 4. Modelo em Cascata



14

### Engenharia de Sistemas / Informação e Modelagem

- Envolve a coleta de requisitos em nível do sistema, com uma pequena quantidade de projeto e análise de alto nível.
- Esta visão é essencial quando o software deve fazer interface com outros elementos (hardware, pessoas e banco de dados).

## 4. Modelo em Cascata



### Análise de Requisitos de Software

- O processo de coleta dos requisitos é intensificado e concentrado especificamente no software.
- Deve-se compreender o domínio da informação, a função, desempenho e interfaces exigidos.
- Os requisitos (para o sistema e para o software) são documentados e revistos com o cliente.

15

## 4. Modelo em Cascata



### Projeto

- Tradução dos requisitos do software para um conjunto de representações que podem ser avaliadas quanto à qualidade, antes que a codificação se inicie.
- Se concentra em 4 atributos do programa: Estrutura de Dados, Arquitetura de Software, Detalhes Procedimentais e Caracterização de Interfaces.

16

## 4. Modelo em Cascata

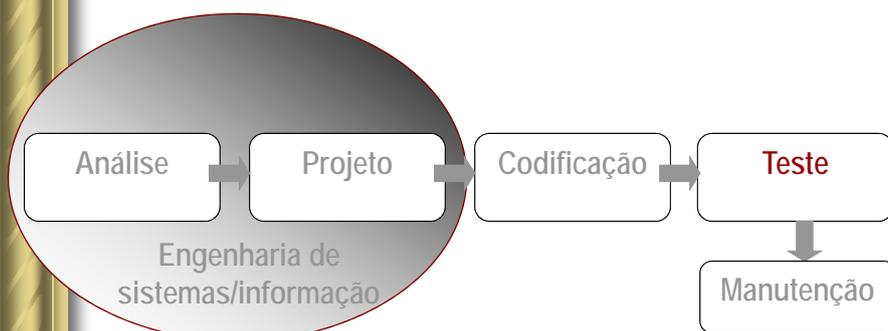


### Codificação

- Tradução das representações do projeto para uma linguagem “artificial” resultando em instruções executáveis pelo computador.

17

## 4. Modelo em Cascata

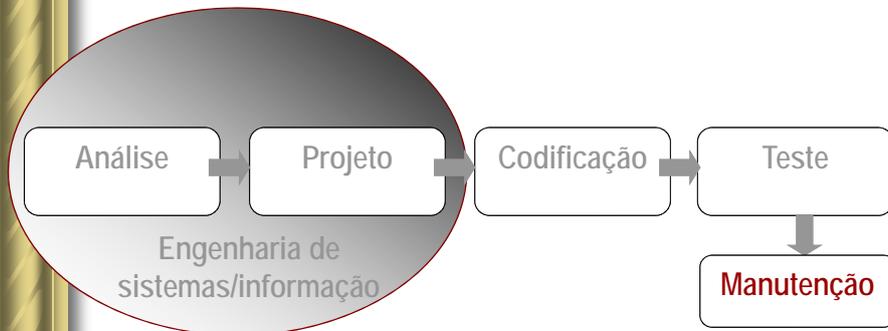


### Testes

- Concentra-se nos aspectos lógicos internos do software, garantindo que todas as instruções tenham sido testadas.
- Concentra-se nos aspectos funcionais externos, para descobrir erros e garantir que a entrada definida produza resultados que concordem com os esperados.

18

## 4. Modelo em Cascata



### Manutenção

- Provavelmente o software deverá sofrer mudanças depois que for entregue ao cliente.
- Causas das mudanças: erros, adaptação do software para acomodar mudanças em seu ambiente externo e exigência do cliente para acréscimos funcionais e de desempenho.

19

## 4. Modelo em Cascata

### Particularidades do Modelo em Cascata:

- É o paradigma mais antigo e o mais amplamente utilizado em engenharia de software.
- Aplicável em softwares de qualquer tamanho e aplicação.
- Ele produz um padrão no qual os métodos para análise, projeto, codificação, testes e manutenção podem ser colocados.
- Logo no início é difícil estabelecer explicitamente todos os requisitos, sendo passível de refinamento.
- Uma versão executável do software só fica disponível numa etapa avançada do desenvolvimento.

20