

Desenvolvimento de Software

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

Atividades de Desenvolvimento

1. Especificação de requisitos
2. Projeto de Software
3. Implementação
4. Validação do software
5. Evolução de software

Atividades de Desenvolvimento

1. **Especificação de requisitos**
2. Projeto de Software
3. Implementação
4. Validação do software
5. Evolução de software

Especificação de Requisitos

- Um sistema de software deve satisfazer as necessidades de seus usuários
 - Tais necessidades são expressas na forma de **requisitos**
- **Requisito** = ação que deve ser executada pelo sistema
 - Ex: registrar as notas dos alunos, calcular a média final, verificar aprovação, etc.

Especificação de Requisitos

- Inclui quatro fases principais
 - Estudo de viabilidade
 - Elicitação (ou análise) de requisitos
 - Especificação de requisitos
 - Validação dos requisitos

Atividades de Desenvolvimento

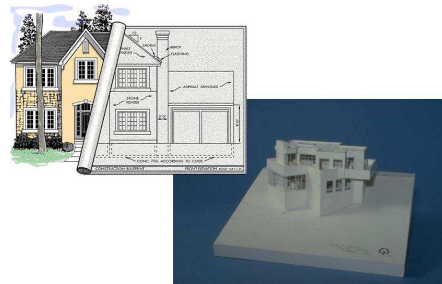
1. Especificação de requisitos
2. **Projeto de Software**
3. Implementação
4. Validação do software
5. Evolução de software

Projeto de Software

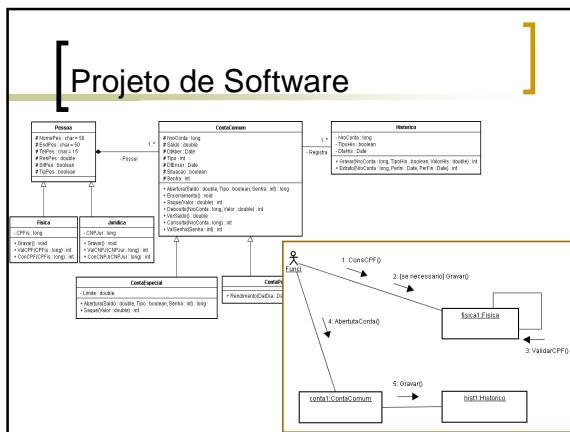
Dividido em duas etapas

- **Projeto Preliminar** define a estrutura modular do software, as interfaces e as estruturas de dados utilizadas
 - Modelo de Arquitetura
- **Projeto Detalhado** descreve detalhadamente cada módulo definido do projeto preliminar
 - Modelo de Projeto

Projeto de uma Casa



Projeto de Software



Atividades de Desenvolvimento

1. Especificação de requisitos
2. Projeto de Software
- 3. Implementação**
4. Validação do software
5. Evolução de software

Implementação

- A implementação segue as definições da fase anterior
 - Transcreve as decisões de projeto arquitetural e detalhado para uma linguagem de programação
- Abordado em outras disciplinas
 - AEDS 1 e 2, POO, Programação de Computadores, Programação Modular, etc.

Atividades de Desenvolvimento

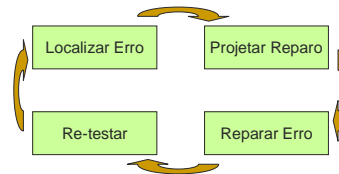
1. Especificação de requisitos
2. Projeto de Software
3. Implementação
4. **Validação do software**
5. Evolução de software

Teste e Validação

- Tem por objetivo garantir que o sistema satisfaça os requisitos
- Consiste da realização de alguns tipos de testes para encontrar erros
- A inexistência de erros não representa a *adequação operacional* do sistema
 - Deve ser feita a **validação** com o cliente

Depuração de Software

- Localizar e corrigir erros



Tipos de Testes

- Teste de Componente (unitário)
 - Garantir que um componente funciona
- Teste de Sistema (integração)
 - Garantir que dois ou mais componentes funcionam juntos
- Teste de Aceitação (validação)
 - Garantir que o sistema faz o que o cliente deseja

Atividades de Desenvolvimento

1. Especificação de requisitos
2. Projeto de Software
3. Implementação
4. Validação do software
5. **Evolução de software**

Manutenção e Evolução

- O custo de manutenção é geralmente muito maior que o custo de desenvolvimento
- Cada vez menos sistemas são desenvolvidos "do zero"
 - Sistemas são desenvolvidos/adaptados a partir de outros sistemas
- Faz sentido considerar desenvolvimento e manutenção como atividades contínuas

Leis de Lehman (Evolução)



M. M. Lehman. **Rules and Tools for Software Evolution Planning and Management**. Annals of Software Engineering, 2001.

[Evolução de Software]

- Objetivo de Manny Lehman
 - Definir uma teoria unificada para evolução de software
- Resultados
 - Um conjunto de oito leis que “governam” a evolução de sistemas

[Lei da Modificação Contínua]

- Sistemas devem ser continuamente adaptados ou eles se tornam progressivamente menos satisfatórios

[Lei da Complexidade Crescente]

- A medida que um sistema evolui, sua complexidade aumenta, a menos que seja realizado esforço para mantê-la ou diminuí-la

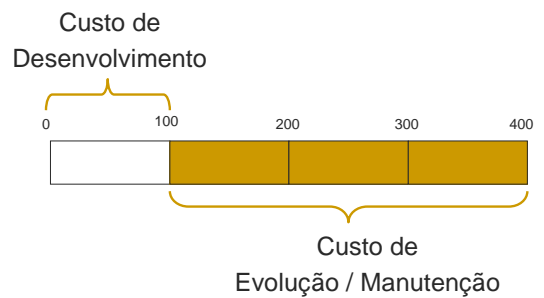
[Lei do Crescimento Contínuo]

- O conteúdo funcional de sistemas devem ser continuamente aumentado para manter a satisfação do usuário

[Lei da Qualidade Declinante]

- A qualidade de sistemas parecerá estar declinando a menos que eles sejam mantidos e adaptados às modificações do ambiente

[Resultado das Leis = Custo]



[Bibliografia]

- Ian Sommerville. Engenharia de Software, 9ª Edição. Pearson Education, 2011.
 - Capítulo 1