

Engenharia de Software

Aula nº 04

Praticando

Maristela Weinfurter Teixeira

Organização da Aula

Como utilizo tudo que vi?

5-Implementação

6-Testes

7-Implantação

Implementando

Pensando no Software



Implementação

A definição da arquitetura de software traz consigo a plataforma de desenvolvimento.

Boas Práticas de Programação

Um bom código não é só funcional, mas também com baixo custo de manutenção.

Boas Práticas de Programação

Código mal escrito é difícil de ser mantido. Há quem prefira reescrevê-lo.

Boas Práticas de Programação



Boas Práticas de Programação

- Comentários
- Nomes de variáveis significativos
- Declaração e inicialização separadas da lógica

Boas Práticas de Programação

- Comandos claros
- Comandos conhecidos
- Identação
- Não economize inicialização e finalização de blocos

Boas Práticas de Programação

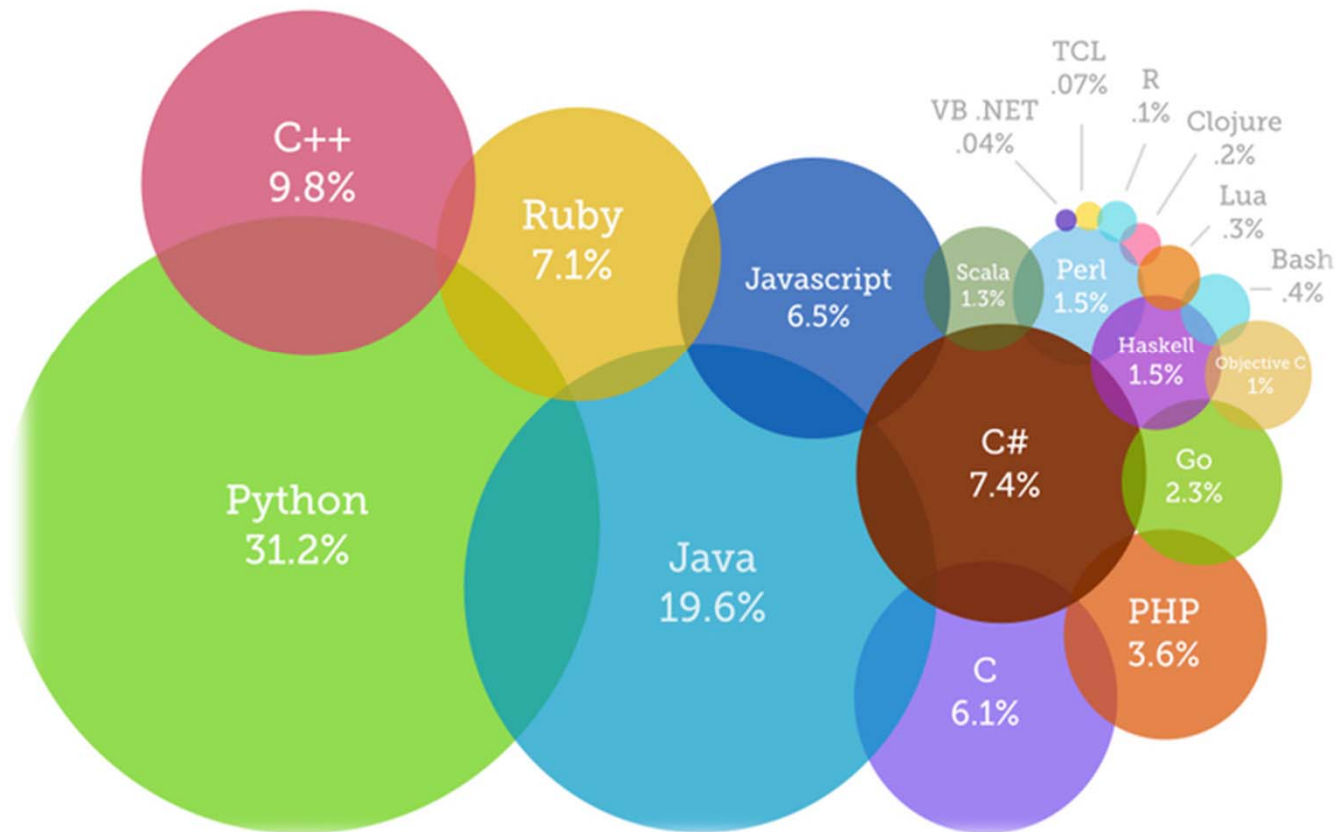
- Programe dentro do paradigma: estruturado, orientado a objetos, etc.
- Modularização e decomposição
- Tipagem forte logicamente

Boas Práticas de Programação

- Programe para os outros não para você

Boas Práticas de Programação

Most Popular Coding Languages of 2015



Boas Práticas de Programação

Para aprender linguagens de programação online:

Links para aprender programação:

<https://www.codecademy.com/learn>

Links para aprender programação e várias outras áreas do conhecimento:

<https://www.futurelearn.com/courses>

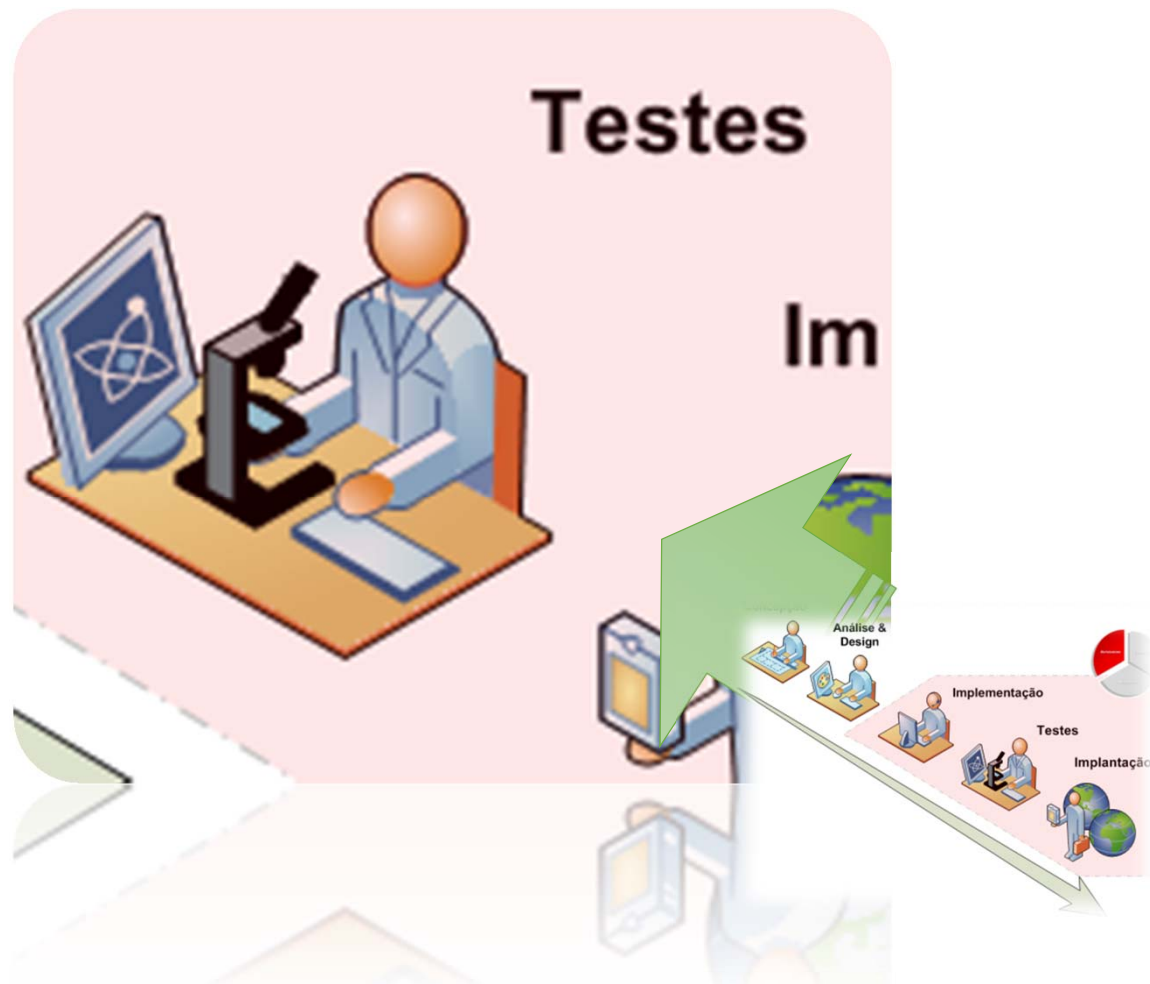
<https://www.coursera.org/>

<https://www.edx.org/>

<https://www.udemy.com>

Testando

Pensando no Software



Testes

IEEE 610.12, 1990:

- Erro: defeito cometido por um indivíduo ao tentar entender uma determinada informação.

Testes

IEEE 610.12, 1990:

- Defeito ou Falta: manifestação concreta de um erro num artefato de software. Um erro pode ser resultado de vários defeitos.

Testes

IEEE 610.12, 1990:

- Falha: comportamento operacional do software diferente do esperado pelo usuário.

Testes

IEEE 830,1998:

- Recomenda práticas para especificação de requisitos de software. A falta de atributos dos requisitos constitui-se num tipo de defeito.

Testes

IEEE 830,1998:

- Omissão
- Ambiguidade
- Inconsistência
- Fato incorreto
- Informação estranha

Testes

Estratégias:

- Baseadas em implementação
- Baseadas em especificação
- Baseadas em modelos

Testes

Estratégias:

- Baseadas em implementação
- Baseadas em especificação
- Baseadas em modelos

Testes

Estratégias:

- Caso de teste
- Procedimento de teste (roteiro)

Testes

Inspeção de software:

- Melhoria da qualidade de artefatos de software através da análise, detectando e removendo defeitos antes que o artefato seja passado para o desenvolvimento.

Testes

Inspeção de software:

- Checklists sobre diferentes níveis de formalidade e de configuração das equipes de inspeções, revisões e validações.

Testes

Testes de Unidade:

- Validação de classes
- Exemplo: framework JUnit

Testes

Casos de Teste:

- Testes eficazes sobre o que foi planejado e especificado através de ferramentas de geração de casos de testes.
- Fluxos e regras.

Testes

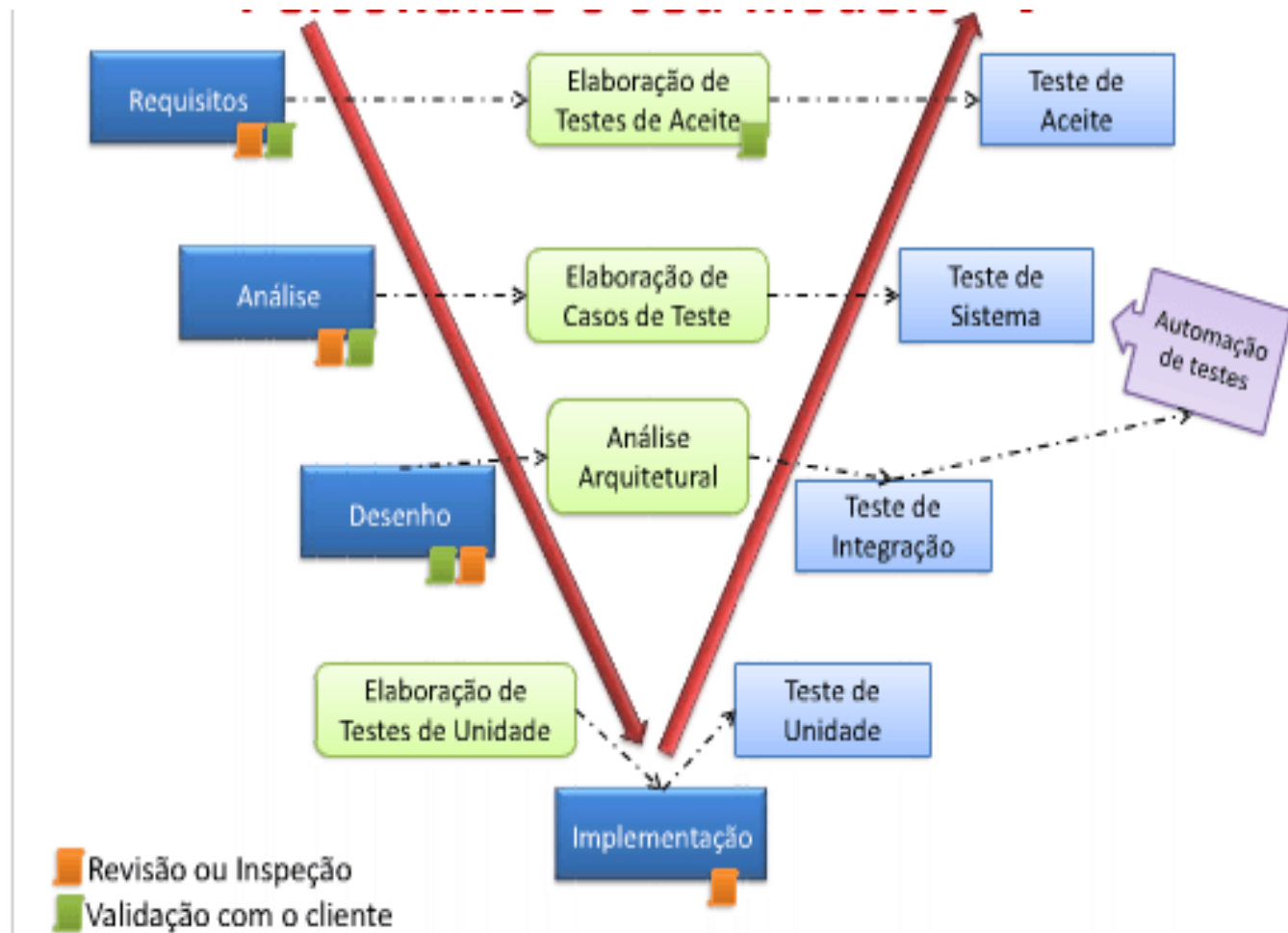
Testes de Sistema

- Testes de aceite
- Abstração de detalhes do software
- Aceite de artefatos
- ISO 9126 – testes de carga, estresse e maturidade

Testes

Mas, quais tipos de testes utilizar?

Testes



Testes

Outros testes:

- Desempenho
 - Carga
 - Estresse
 - Maturidade
- Segurança
 - XSS
 - SQL Injection

Testes

Outros testes:

- Usabilidade:
 - Acessibilidade
 - Facilidade de uso
- Caixa Branca:
 - Cobertura de comandos
 - Cobertura de decisão

Testes

Outros testes:

- Caixa Preta:
 - Transição de estado
 - Tabela de decisão
 - Baseado em histórias do usuário

Implantando

Pensando no Software



Implantação

Finalmente, chegamos no momento da:

- Distribuição e entrega
- Instalação e configuração
- Utilização e
- manutenção

Implantação

Sejam em projetos mais tradicionais ou segundo métodos ágeis, a ideia é que haja uma finalização tranquila do projeto, utilizando-se os métodos de finalização.

Implantação

Esta fase requer que os stakeholders recebam um bom treinamento.

Instalação e configuração recebam a devida atenção.

Implantação

O importante além do aceite oficial dos stakeholders é que o software atinja os requisitos de satisfação na entrega do produto final.

Referências de Apoio

- Sommerville, Ian; Engenharia de Software. 9^a. Ed. São Paulo: Pearson, 2011.
- Presman, Roger; Engenharia de Software. 7^a. Ed. Porto Alegre: Bookman, 2011.

Referências de Apoio

- Pfleeger, Shari L.; Engenharia de software: teoria e prática. 2ª. Ed.. São Paulo, Prentice Hall, 2004.
- Page-Jones, Meilir.; Fundamentos do desenho orientado a objeto com uml. São Paulo, Pearson, 2001.

FIM