



Instituto Nacional de Ciência  
e Tecnologia para Engenharia  
de Software

# Rede de Laboratórios de Produtividade de Software

# Testes em aplicações WEB

## Uma Visão Geral

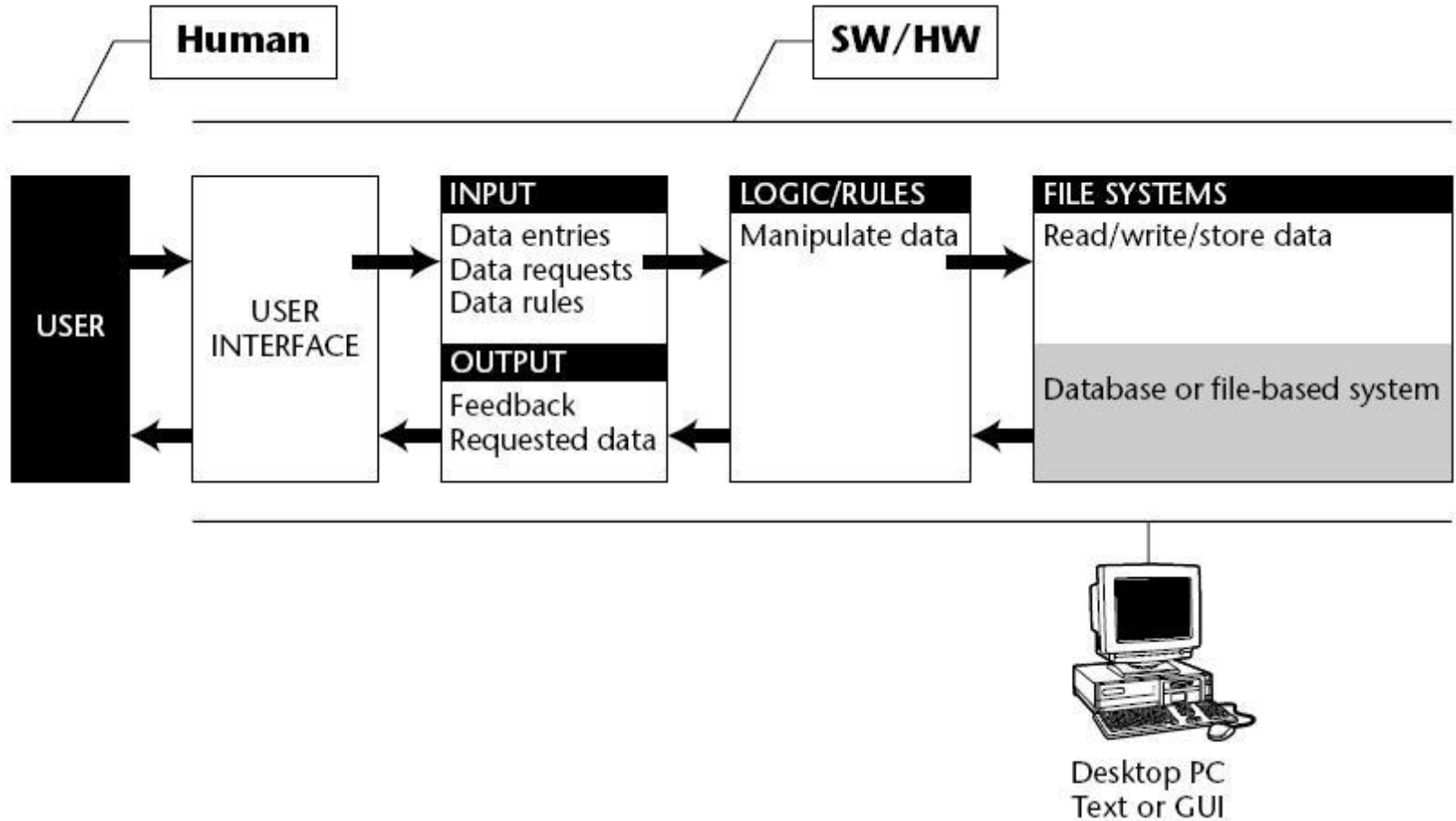
Programa de Capacitação  
em Testes de Software



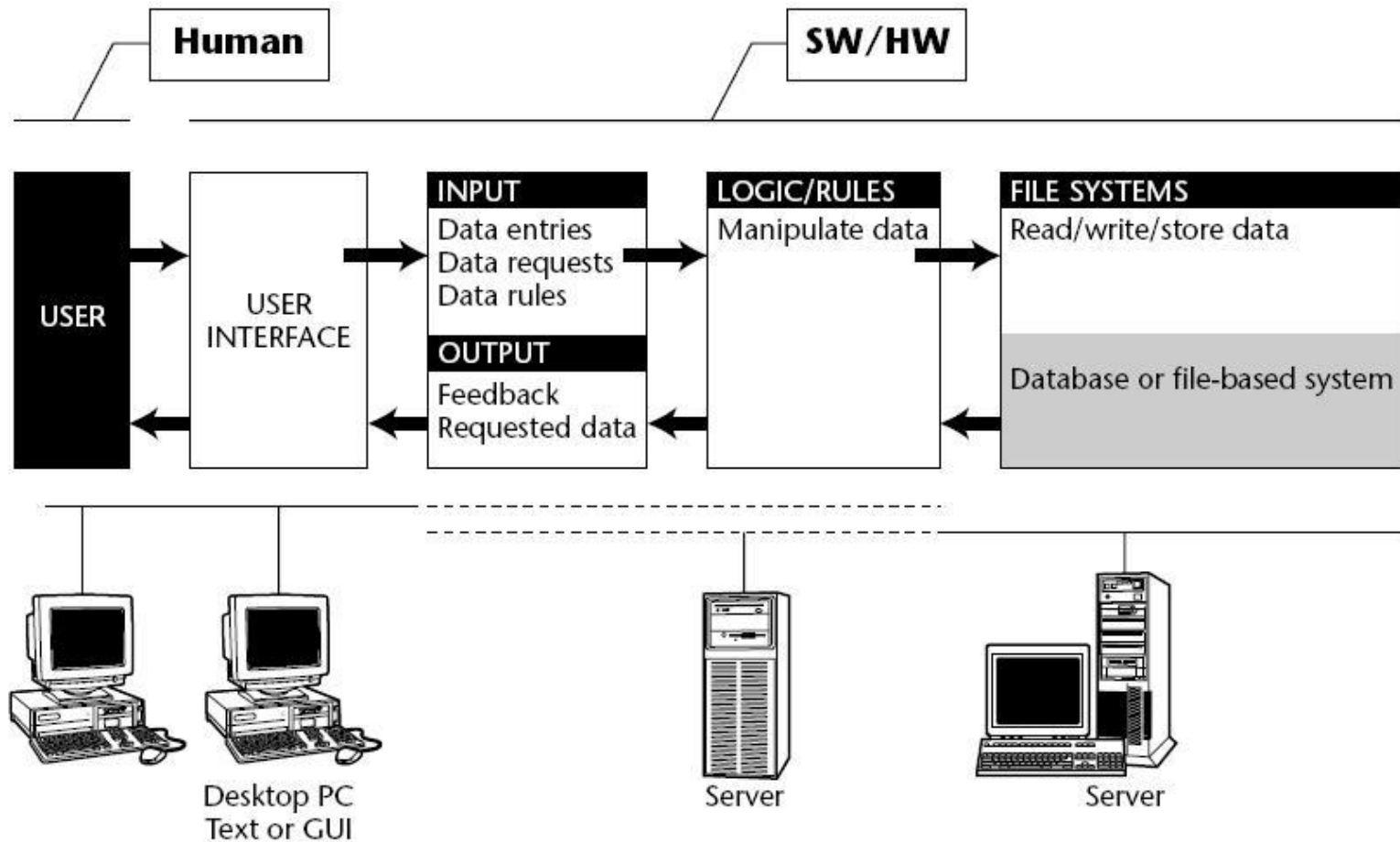
Instituto Nacional de Ciência  
e Tecnologia para Engenharia  
de Software



# Desktop system

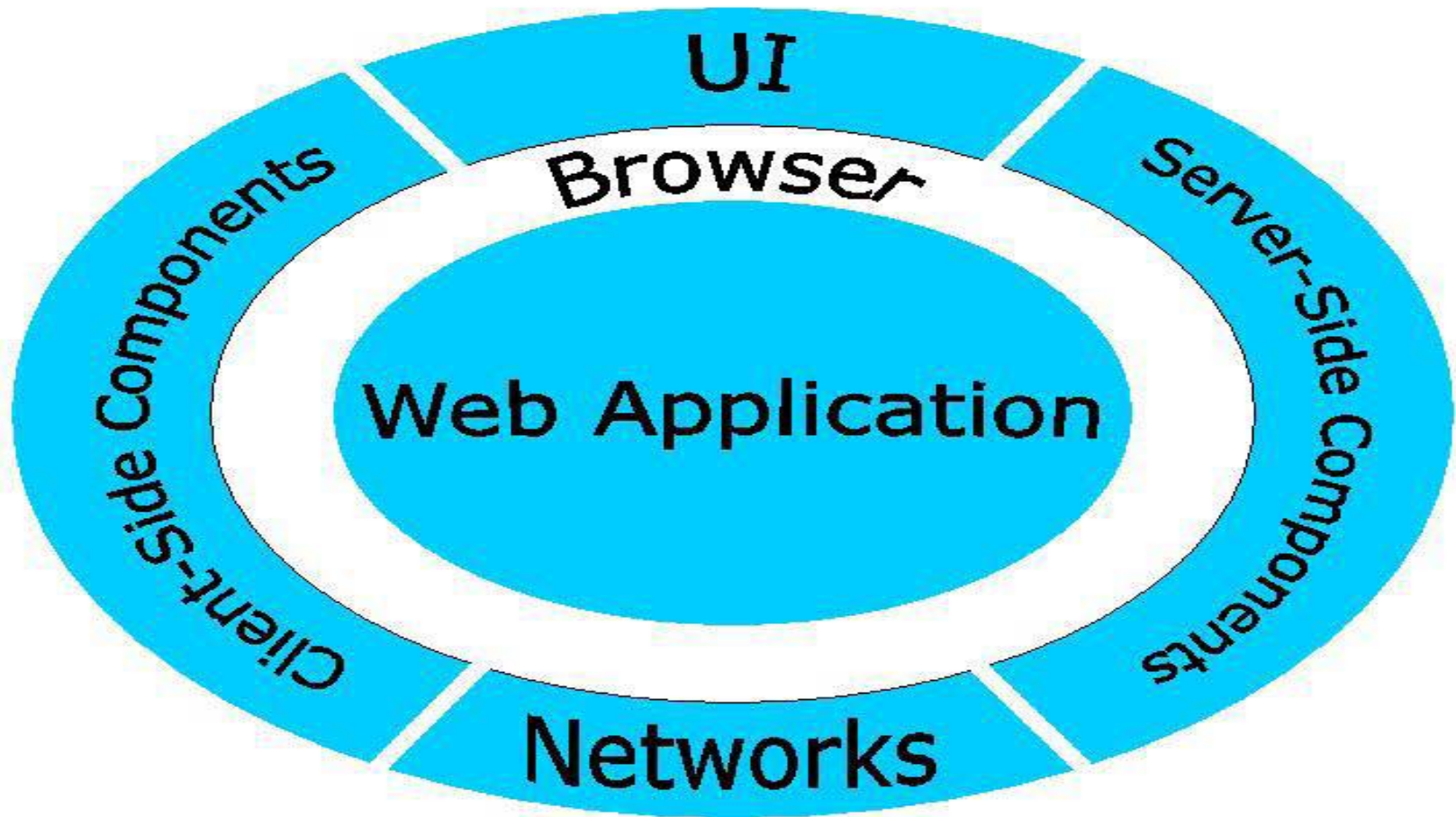


# WEB system

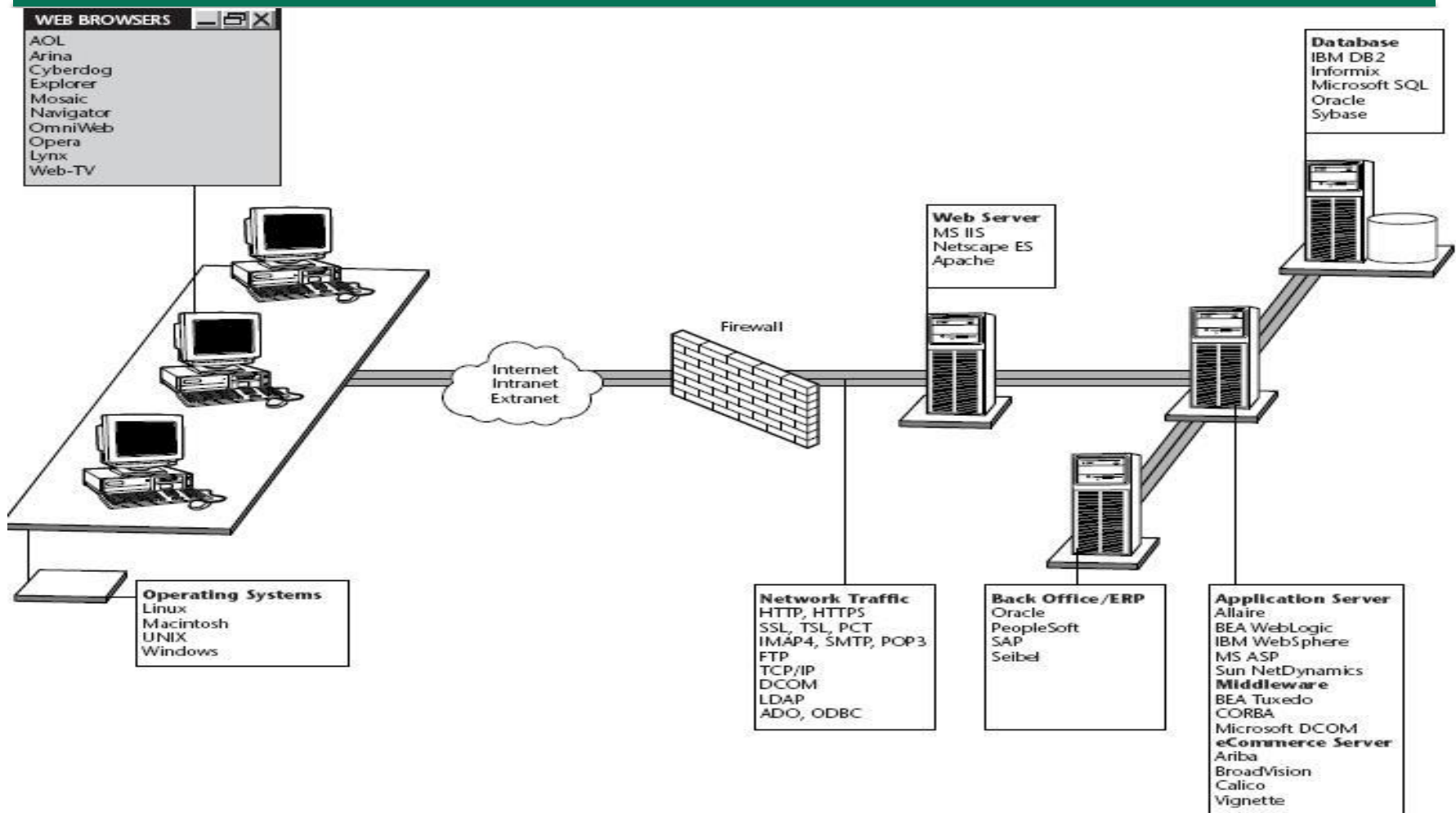


# Ambiente de aplicativo da Web

---



# Ambiente WEB



# Testes de SW X WEB

---

## DESAFIOS

- Base de usuários grande e variada;
- O ambiente de negócios;
- Localidades;
- Ambientes de Teste;
- Segurança;
- Desempenho;
- Disponibilidade;
- Conectividade;

# Processo de Teste

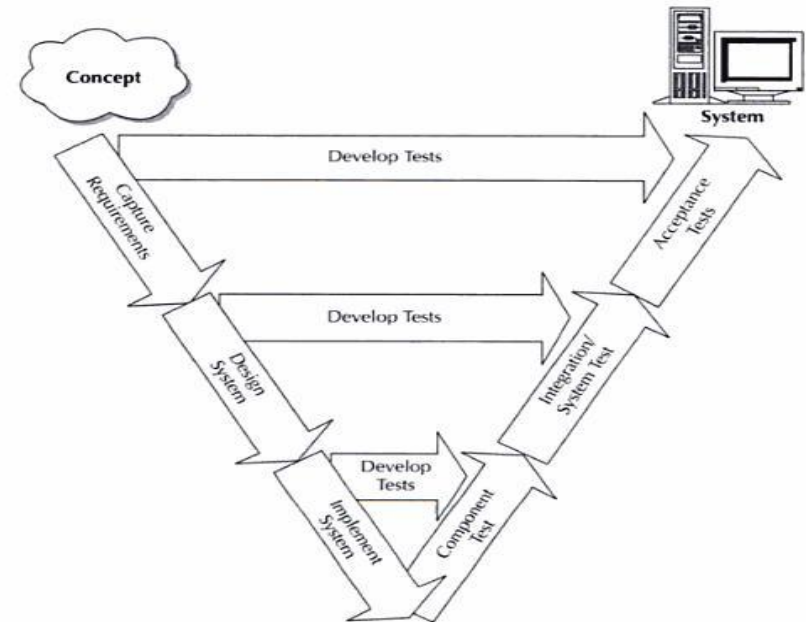
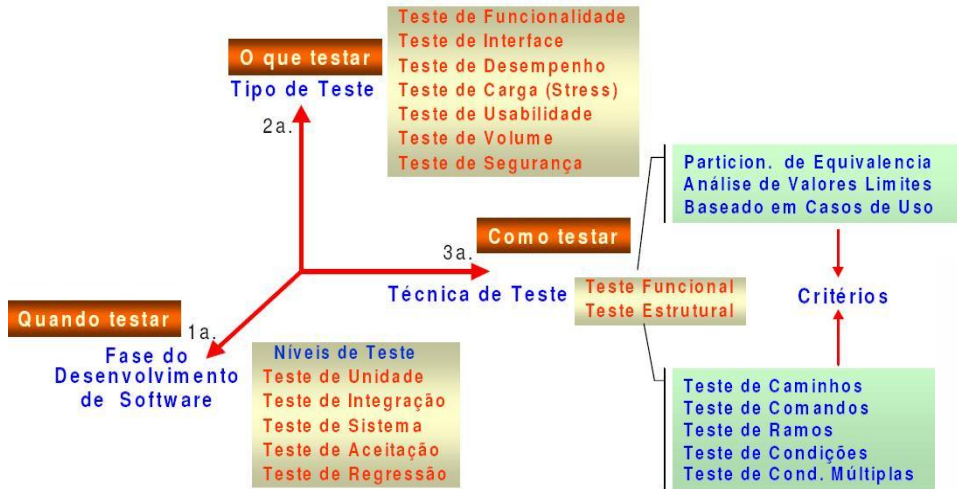


FIGURE 1-3 THE V MODEL OF THE WATERFALL SYSTEM DEVELOPMENT LIFECYCLE



# Dica !!!

---

Testar cada camada independente lhe permite identificar mais facilmente os erros antes da realização dos testes de sistema. Se você confiar apenas no teste do sistema, então você pode ter dificuldade em encontrar os componentes específicos que estão criando o problema.

# Camada de Apresentação

---

Os testes consistem em verificar os elementos da interface de uma página web.

Ex:

Fonte, layout de tela, cores, resolução gráfica, links, gramática, estética global, ortografia, precisão dos campos, valores padrão, etc.

# User Interface Testing

---

- Teste 1: Validar o Código
- Teste 2: Verificar Links Ativos
- Teste 3: Checar texto alternativo
- Teste 4: Verificar acessibilidade
- Teste 5: Verificar Usabilidade
- Teste 6: Verificar formulários
- Teste 7: Verificar entrada de dados inválidos
- Teste 8: Redimensionar o navegador
- Teste 9: Examinar os objetos de página da Web
- Teste 10: Verificar o acesso não autorizado
- Teste 11: Alterar configurações do navegador

# Network e Client-Side

---

## Network Interface Testing

Teste 12: alterar a velocidade de acesso à rede

## Client-Side Component Tests

Teste 13: Conferir Plug-Ins

Teste 14: Verificar o Ambiente

# Camada de Negócios

---

Os testes da camada de negócios tem como objetivo encontrar erros na lógica do negócio. Pode ser utilizadas as técnicas de testes caixa branca e caixa preta. Os testes vão detectar erros nos requisitos de desempenho do aplicativo, aquisição de dados e processamento de transações e autenticação de usuários.

# Server-Side Component Tests

---

- Teste 15: Teste de Carga
- Teste 16: Teste de Stress
- Teste 17: Teste de Performance

# Camada de Acesso a Dados

---

- Os testes da camada de dados procuram por erros da seguinte categoria: Tempo de resposta, integridade dos dados, tolerância a falha e recuperabilidade.
- Teste 18: Teste de Volume

# Tipos de Teste

---

Cada tipo de teste tem foco em um objetivo Particular, ele define o alvo do teste, que pode ser o teste de uma funcionalidade, a ser realizada pelo software ou uma característica não-funcional.

- Teste Funcional
- Teste não Funcional
- Teste Estrutural
- Teste de Confirmação e Regressão



# Principais Tipos de Teste

---

## **Teste Funcional**

✓ Testa as funções, requisitos funcionais e casos de uso.

## **Teste de Segurança**

✓ Se dados e sistemas são acessados apenas pelos possíveis atores.

## **Teste de Configuração**

✓ Se o software funciona no hardware e software propostos.

# Ex: Smart Education

---

“Qualquer navegador compatível com o Internet Explorer 7 (ou superior) ou Mozilla Firefox 2 (ou superior) poderá ser utilizado.

O sistema deverá estar hospedado em um servidor com acesso à internet e com os programas PHP 5.0, ou superior, e MySQL, instalados e configurados adequadamente. No servidor ainda será necessário que esteja instalado e configurado adequadamente o ambiente virtual de aprendizagem Moodle versão 1.9, ou superior.

Em relação ao hardware, o equipamento mínimo recomendável é um computador com processador Pentium II 350mhz ou K6-2 500mhz ou equivalente, 128MB RAM, espaço em disco, começando por 1GB e resolução do monitor de, no mínimo, 800 x 600.”

# Principais Tipos de Teste

---

## Teste de Performance

✓ Fluxo de execução, acesso a dados, chamadas ao sistema, tempo de resposta são monitorados para identificação de gargalos na performance e processos ineficientes.

## Teste de Carga

✓ Avalia performance do software sobre condições normais de uso. Exemplo: número de transação por unidade de tempo, tempo de resposta, usuários simultâneos...

## Teste de Volume\*

✓ Como o sistema trabalha com grande volume de dados, transação, usuários, periféricos por um longo período de tempo. Trabalha com o volume máximo requerido.

# Principais Tipos de Teste

---

## Teste de Instalação

✓ Se o software instala conforme planejado em diferentes hardware/software e sobre condições diferentes (pouco espaço, interrupção ...)

## Teste de Integridade

✓ Testa robustez (resistência a falhas), conformidade com linguagem, sintaxe e uso dos recursos;  
✓ Executado sobre unidades e unidades integradas.

## Teste Stress

✓ Testa condições anormais. Utiliza picos excessivos de carga em pequenos períodos.

# Principais Tipos de Teste

## Teste de Usabilidade

✓ Foca em fatores humanos, estética, consistência da interface, help, wizards, documentação do usuário, treinamento.

## Teste de Regressão

✓ Reteste seletivo de um sistema ou componente para verificar se as modificações não causaram efeitos indesejáveis e que o sistema ou componente ainda atende aos requisitos especificados.

## Teste de Sanidade

✓ É um conjunto de caso de teste que estabelece que o sistema é estável e todas as funcionalidades principais estão presentes e trabalham sob condições normais. [Craig and Jaskiel]

# Riscos da não Priorização dos Requisitos não Funcionais

---

- Baixa performance
- Incompatibilidade em ambientes com diversas configurações de HW/SW
- Segurança inadequada
- Usabilidade insuficiente
- Attacks

# Top 10 - Vulnerabilidades Web

---

1. Input inválido
2. Autenticação
3. Gerenciamento de Sessão
4. Cross-site scripting
5. Campos Escondidos
6. Injeção
7. Tratamento de Exceção
8. Criptografia
9. Referências Direta a Objetos
10. Gerenciamento de Navegação



# Ferramenta

---

- Teste de Performance e Carga
  - Principais Funcionalidades
    - Geração de carga no sistema.
    - Medição do tempo para realização de transações específicas.
    - Análise de médias de tempo de resposta
- Exemplos:
  - Jmeter <http://jakarta.apache.org/jmeter/>
  - Jprobe <http://www.quest.com/jprobe/>
  - WebLoad <http://www.webload.org/>
  - Selenium <http://www.openqa.org/selenium/>



# JMeter

---

É uma ferramenta Open Source para testar aplicações Web. O JMeter realiza teste de performance, volume e estresse.

- Entre suas funcionalidades, pode-se destacar:
  - Multiplataforma
  - Suporta teste para diversos protocolos (HTTP, JDBC, FTP, JMS, SOAP, etc)
  - Teste Distribuído

# Ferramenta

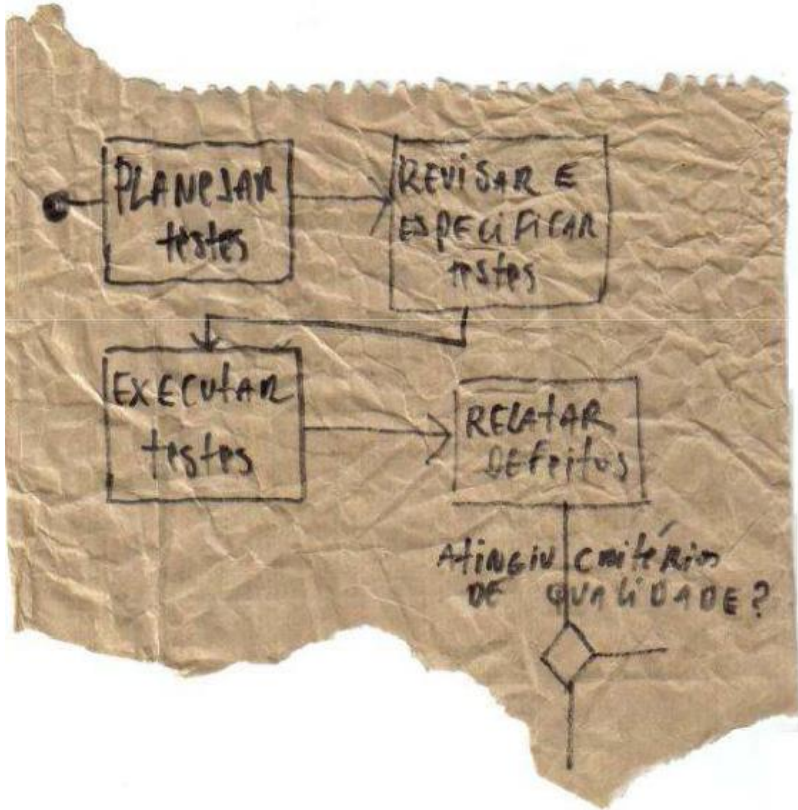
---

- Ferramenta de Monitoração
- Principais Funcionalidades
  - Identificação de problemas e envio de mensagens de alerta.
  - Log de informação real-time e informação histórica.
  - Monitoração do número de usuários na rede.
  - Monitoração de tráfego na rede.

Exemplo: Ethereal, Open Source (<http://www.ethereal.com>)

- Alguns tipos de ferramentas de teste são intrusivos.
- Não são exclusivamente ferramentas de teste.

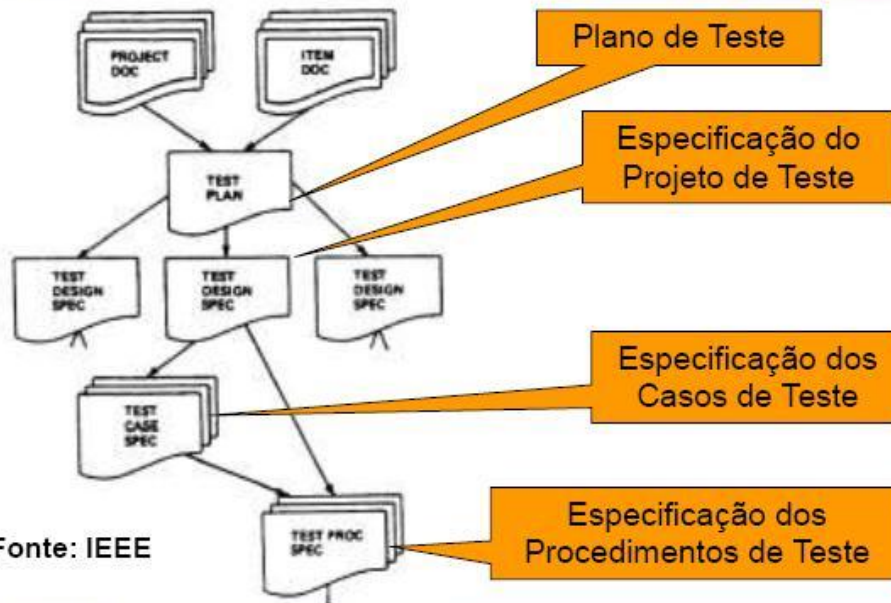
# Elaboração dos Testes



**Cenário de Teste:** É o caminho a ser seguido ou a situação a ser testada.

**Caso de Teste:** É o cenário a ser executado para verificar se o que foi especificado está devidamente implementado.

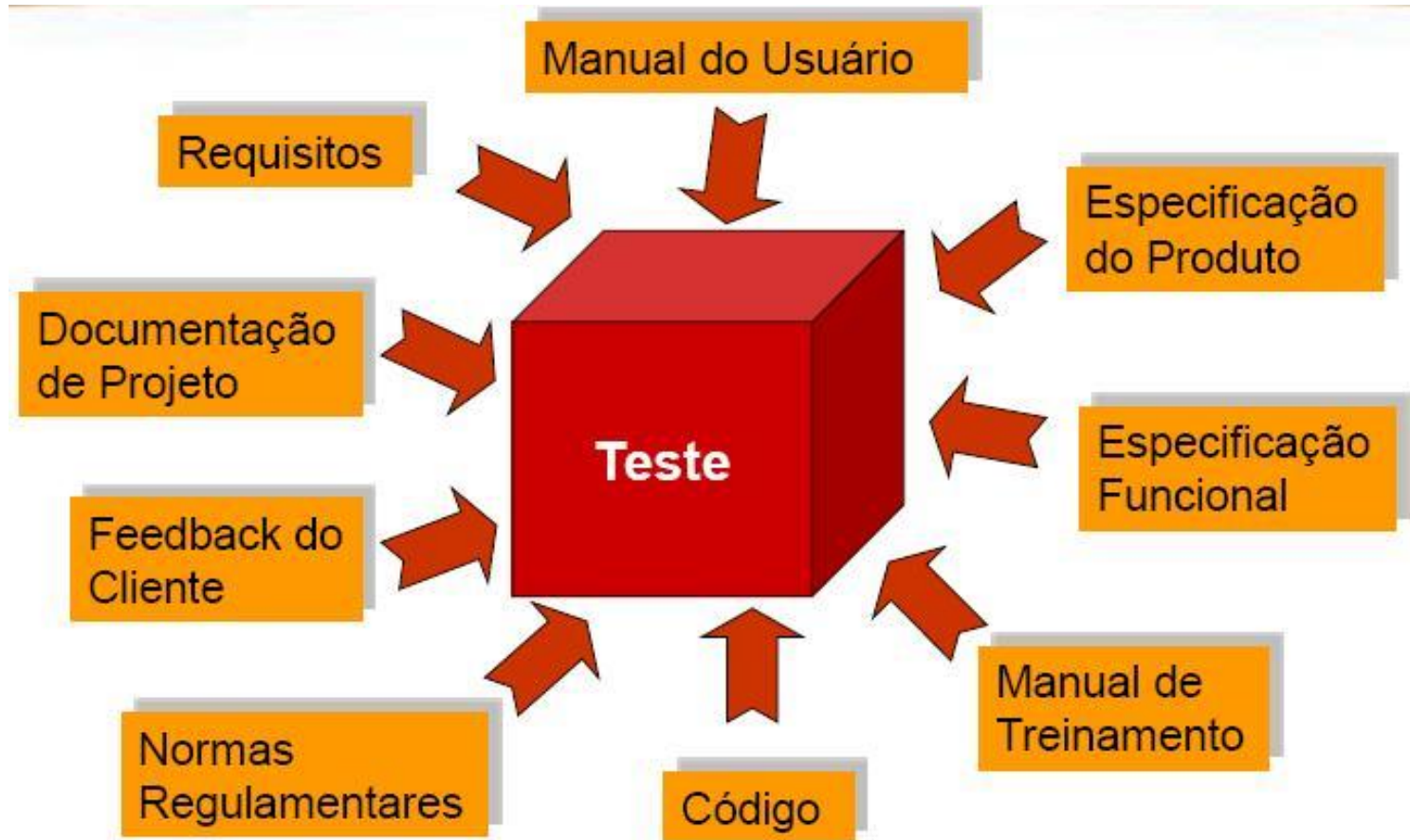
# Artefatos IEEE X RUP



Fonte: IEEE



# Cenários de Teste



# Exemplo de Cenário de Teste

---

- Aplicação Bancária

Requisito:

Transferência Bancária;

Cenário:

Consultar o saldo da conta origem;

Consultar o saldo da conta destino;

Transferir valor da conta origem para conta destino;

Consultar novo saldo das contas;

# Avaliação

---

Baseado nas informações disponibilizadas durante o treinamento, levante os cenários de testes para os requisitos da aplicação WEB (Smart Education).

# Resumo

---

- Contexto

Teste de Software em Aplicações WEB

- Problema

A não priorização dos requisitos não funcionais das aplicações WEB podem aumentar os riscos do fracasso

- Solução

Elaborar uma estratégia de testes voltada a garantir a execução dos tipos de testes mais criticos para as aplicações WEB

- Avaliação da Solução

Satisfação dos usuários

---



# Referências

---

- [IEEE 829] - IEEE Std 829-1998 Standard for Software Test Documentation
- [Syllabus] - ISTQB
- [Aderson Bastos] Base de COhecimento em teste de software
- [Sommerville] Software Engineering
- [Myers] The Art Of Software Testing
- [Ethereal] <http://www.ethereal.com/>
- [JMeter] <http://jakarta.apache.org/jmeter/>

# Sites

---

Wiki do LabPS

<http://labps.cin.ufpe.br>

Twitter do LabPS

[http://twitter.com/Rede\\_LabPS](http://twitter.com/Rede_LabPS)

INES

<http://www.ines.org.br>

Amigos

<http://amigos.ines.org.br>

Cin/UFPE

<http://www.cin.ufpe.br>

# Dúvidas

---



# Obrigado!

# Testes em aplicações WEB

## Uma Visão Geral

Programa de Capacitação  
em Testes de Software



Instituto Nacional de Ciência  
e Tecnologia para Engenharia  
de Software

