

Estágios dos testes de software

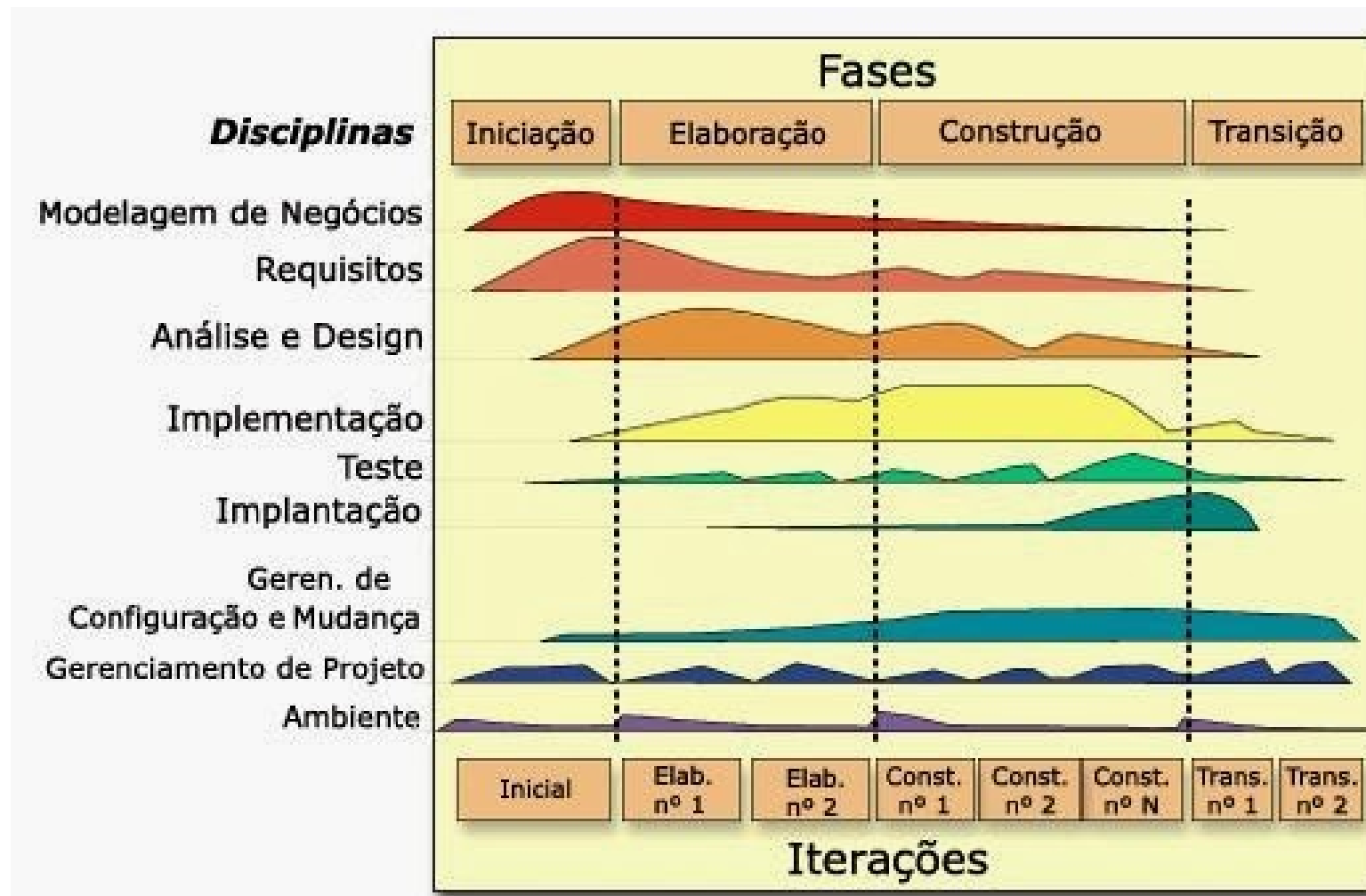
Prof. Dr. Rafael Galvão de Mesquita
rafael.mesquita@garanhuns.ifpe.edu.br

Estágios do teste de software

- Testes de unidade
- Testes de integração
- Teste de aceitação
- Testes de sistema

Testes X Ciclo de desenvolvimento

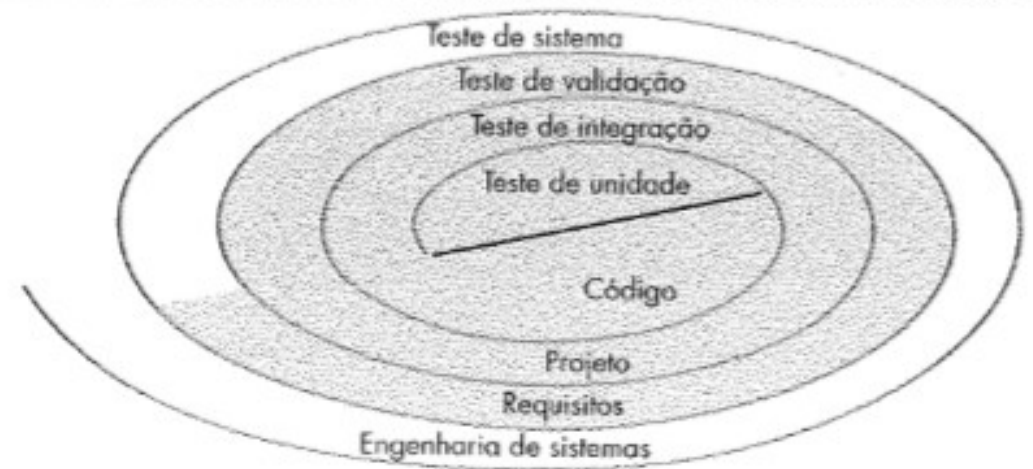
- Testes não se restringem a uma única etapa do ciclo de desenvolvimento



Processo espiral de engenharia de software

- Codificação – teste de unidade
- Projeto – teste de integração
- Requisitos – testes de aceitação (validação)
- Engenharia de sistemas – teste de sistema

Obs: processo realizado de modo incremental!



Processo espiral de engenharia de software

- teste de unidade: testa o código fonte de cada unidade separadamente
- teste de integração: foco no projeto e na construção da arquitetura do software. Integração entre unidades ou componentes
- testes de validação: requisitos do cliente são validados em contraste com o software construído
- teste de sistema: software e outros elementos do sistema são testados como um todo
 - Outros elementos: hardwares, pessoas, BD,...



Testes de unidade

- Focaliza o esforço de verificação em componentes individuais do software
 - Foco: lógica interna de processamento e estruturas de dados de cada componente
- A interface do módulo também é testada de forma a verificar se a informação flui adequadamente para dentro e para fora da unidade que está sendo testada
 - Normalmente é aplicado como um dos primeiros testes
 - Se os dados não entram e saem do módulo adequadamente, os resultados de todos os outros testes são discutíveis



Testes de unidade

- Testes comuns:
 - Caminhos básicos ao longo da estrutura de controle do componente (garantir que todos os comandos sejam executados ao menos uma vez)
 - Situações-limite (valores máximos e mínimos de parâmetros e variáveis)
 - Caminhos de manipulação de erros
 - Fluxo de entrada e saída de dados
- Teste estrutural ou caixa-branca



Testes de unidade

- Normalmente descobre erros relativos a
 - Cálculos errados
 - Precedência aritmética incorreta, inicialização incorreta de variáveis, precisão inadequada,...
 - Comparações incorretas
 - Fluxo de controle inadequado
 - Terminação de ciclo inadequada ou inexistente
 - Variáveis de laços modificadas de modo incorreto
- Comumente realizados pela equipe de desenvolvimento

Testes de unidade

- Teste de limites
 - Uma das estratégias de testes mais importantes, que comumente encontra falhas
 - Exemplo:
 - processamento no n -ésimo elemento de um vetor de tamanho n
 - Remover todos os elementos de uma estrutura de dados e voltar a preenche-la
 - Valores máximos e mínimos de uma variável

Testes de unidade

- Conceitos
 - Testes unitários também podem ser considerados como parte da codificação
 - Projeto de testes pode ser realizado antes da construção do código (abordagem comum em metodologias ágeis) ou após a construção do código

Testes de integração

- Testa se duas ou mais unidades funcionam quando utilizadas em conjunto
- Testes se concentram nas interfaces de comunicação entre as unidades
- Têm como objetivo encontrar falhas de integração entre duas ou mais unidades
- Teste estrutural ou caixa-branca
- Comumente realizados pela equipe de desenvolvimento

Testes de sistema

- Sistema é testado como um todo
 - Testa uma versão do sistema que será entregue ao cliente
- Deve verificar
 - Se os componentes são compatíveis e se interagem corretamente
 - Uso do software como parte de um sistema maior
 - Exemplo: comunicação do software com uma máquina de ponto
 - Acesso às funções dos menus
- É inviável testar exaustivamente todos os casos de teste exaustivamente no teste de sistema
- Comumente realizados pela equipe de testes



Testes de sistema

- Geralmente são baseados em roteiros de teste, criados a partir da especificação de requisitos
- Teste funcional ou caixa-preta
 - Testador normalmente tem acesso apenas à interface do sistema
- Têm como objetivo verificar se o sistema está em conformidade com a especificação de requisitos
 - Verificação
 - Caso a especificação dos requisitos esteja errada, isso não será detectado no teste de sistema
 - Para isso, precisamos do teste de validação



Testes de aceitação

- Testa software no ambiente do cliente
- Teste funcional ou caixa preta
- Validação dos requisitos
- Os testes nessa fase não precisam abranger todos os aspectos e combinações de valores de entrada da funcionalidade testada
 - Concentram-se na validação das regras de negócio
- Comumente realizados pela equipe de testes e pelos clientes



Testes de aceitação

- É uma boa prática automatizar os testes de aceitação
 - Auxilia nos testes de regressão
 - Toda vez que uma nova versão for liberada, ou quando alguma integração/manutenção ocorrer, os testes de aceitação são executados novamente de forma automática
- Porém testes de outros estágios também podem ser automatizados
 - Por exemplo, nesse curso iremos abordar a automação de testes unitários com Junit
- Teste alfa
 - Com a presença de desenvolvedores e um pequeno grupo de usuários
 - Ambiente controlado
 - Desenvolvedor observa usuários utilizando o software
- Teste beta
 - Com grande número de usuários e sem a presença de desenvolvedores
 - Ambiente não controlado



Resumo

Atributos	Nível dos Testes			
	Testes Unitários	Testes de Integração	Testes de Sistema	Testes de Aceitação
Escopo	<i>Unidades</i>	<i>Conjunto de unidades agrupadas</i>	<i>Sistema todo</i>	<i>Sistema todo</i>
Equipe	<i>Desenvolvedores</i>	<i>Desenvolvedores e Analistas de Sistema</i>	<i>Analista de Testes e Testadores</i>	<i>Analista de Testes, Testadores e Usuários</i>
Origem dos dados	<i>Criação manual</i>	<i>Criação manual</i>	<i>Criação automática / dados reais</i>	<i>Dados reais</i>
Volume dos dados	<i>Pequeno</i>	<i>Pequeno</i>	<i>Grande</i>	<i>Grande</i>
Interfaces	<i>Não existem</i>	<i>Não existem</i>	<i>Simuladas / Reais</i>	<i>Reais</i>
Ambientes	<i>Desenvolvimento</i>	<i>Desenvolvimento</i>	<i>Testes</i>	<i>Testes / Produção</i>



Duvidas?

