Introdução ao Teste de Software

Fundamentos de Testes de Software

PUC Minas – São Gabriel

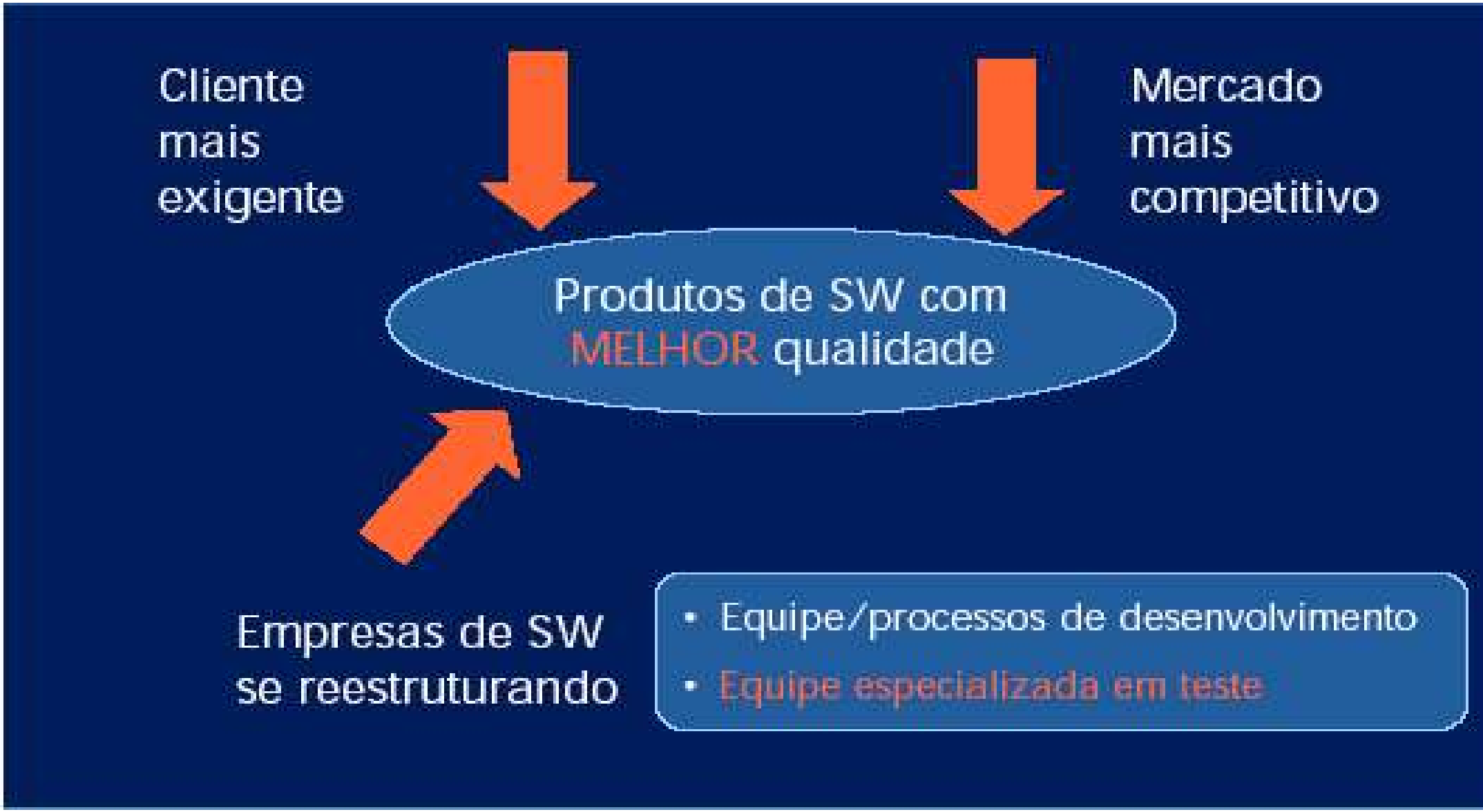
Tópicos

- **Motivação para teste**
  - Por que algumas empresas não testam
- **Finalidades dos Testes**
- **Formando a Equipe de Testes**
  - Usando a Equipe de Desenvolvimento
  - Usando uma Equipe independente
  - Usando uma Equipe de não-especialistas em TI
- **Relacionando as atividades de Testes com as de Desenvolvimento**
- **Processo de Teste**
  - Planejar Testes
  - Especificar Testes
  - Executar Testes
  - Reportar Testes
- **Gerenciamento de Bugs**
- **Ferramentas de Teste**
  - Mantis

Objetivo

- Apresentar uma abordagem geral sobre o processo de teste de software, abrangendo seus principais fundamentos técnicos e gerenciais. Além disso, serão apresentados os principais conceitos necessários para um bom entendimento sobre as atividades de teste.

Motivação para Teste



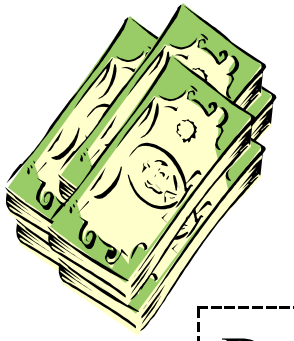
Motivação para Teste

As falhas causam prejuízos financeiros

As falhas causam a perda de confiança do cliente

Por que algumas empresas não testam?

Teste é um processo caro



Desconhecem técnicas de teste adequadas

Dificuldade em implantar um processo de teste

Desconhecem a relação custo/benefício

Só se preocupam com teste na fase final do projeto

Motivação para Teste

- Segundo pesquisas do SEI (*Software Engineering Institute*):
  - 30% dos projetos são cancelados antes de serem finalizados
  - 70% dos projetos falham nas entregas das funcionalidades esperadas;
  - Os custos dos projetos extrapolam mais de 180% dos valores previstos;

Motivação para Teste

- Prazos excedem mais de 220%
- Empresas de nível 1 dedicam cerca de 55% dos esforços para corrigir defeitos
- Esses índices vão sendo gradativamente reduzidos à medida que elas adotam um modelo de qualidade

Finalidade dos Testes

- Verificar se todos os requisitos do sistema foram corretamente implementados
- Assegurar a satisfação do cliente com o produto desenvolvido
- Assegurar, na medida do possível, a qualidade e a corretude do software produzido
- Reduzir custos de manutenção corretiva e retrabalho

Finalidade dos Testes

"Teste é o processo de mostrar que erros estão presentes" (Teste de Defeitos)

"O objetivo do teste é mostrar que um programa executa suas funções corretamente" (Teste de Validação)

"Teste é o processo de criação de confiança de que o programa faz o que ele tem que fazer"

Teste é o processo de executar um programa com a intenção de encontrar defeitos

Formando a Equipe de Testes

Usando a Equipe de Desenvolvimento:

- O Líder do Projeto de Desenvolvimento será também o Líder do Projeto de Testes;

- A Equipe de Teste é a mesma Equipe de Desenvolvimento;

- Os Testes serão executados através de rodízios, onde nunca a pessoa que desenvolveu o módulo executará testes no próprio módulo.

Formando a Equipe de Testes

Desvantagens:

- Diminuição da qualidade do produto final;
- Tendência a não visualizar certos defeitos do projeto (testes de sucesso);
- Tendência a informalidade na execução dos testes;
- Dificuldade de conciliar os cronogramas das equipes de desenvolvimento;
- Falta de conhecimento do negócio da equipe que for executar os testes.

Formando a Equipe de Testes

Usando Equipe Independente:

- Esta é uma prática que está sendo cada vez mais usada no mercado;

- Equipes especializadas em teste produzem resultados, em termos de qualidade do software, muito melhores;

- Essas equipes possuem um treinamento adequado para executar com qualidade os testes e estão bastante familiarizadas com as suas ferramentas e metodologias.

Formando a Equipe de Testes

Desvantagens:

- Custos maiores;
- Aumento no tempo de liberação do software;
- Tendência da equipe de desenvolvimento em relaxar na parte que lhe cabe (teste unitário e de integração);
- Divergências entre as duas equipes.

Formando a Equipe de Testes

Usando Equipes de não-especialistas em TI

- Muitas empresas usam grupos de usuários para fazer o chamado trabalho de homologação do software ou o seu teste de aceitação;

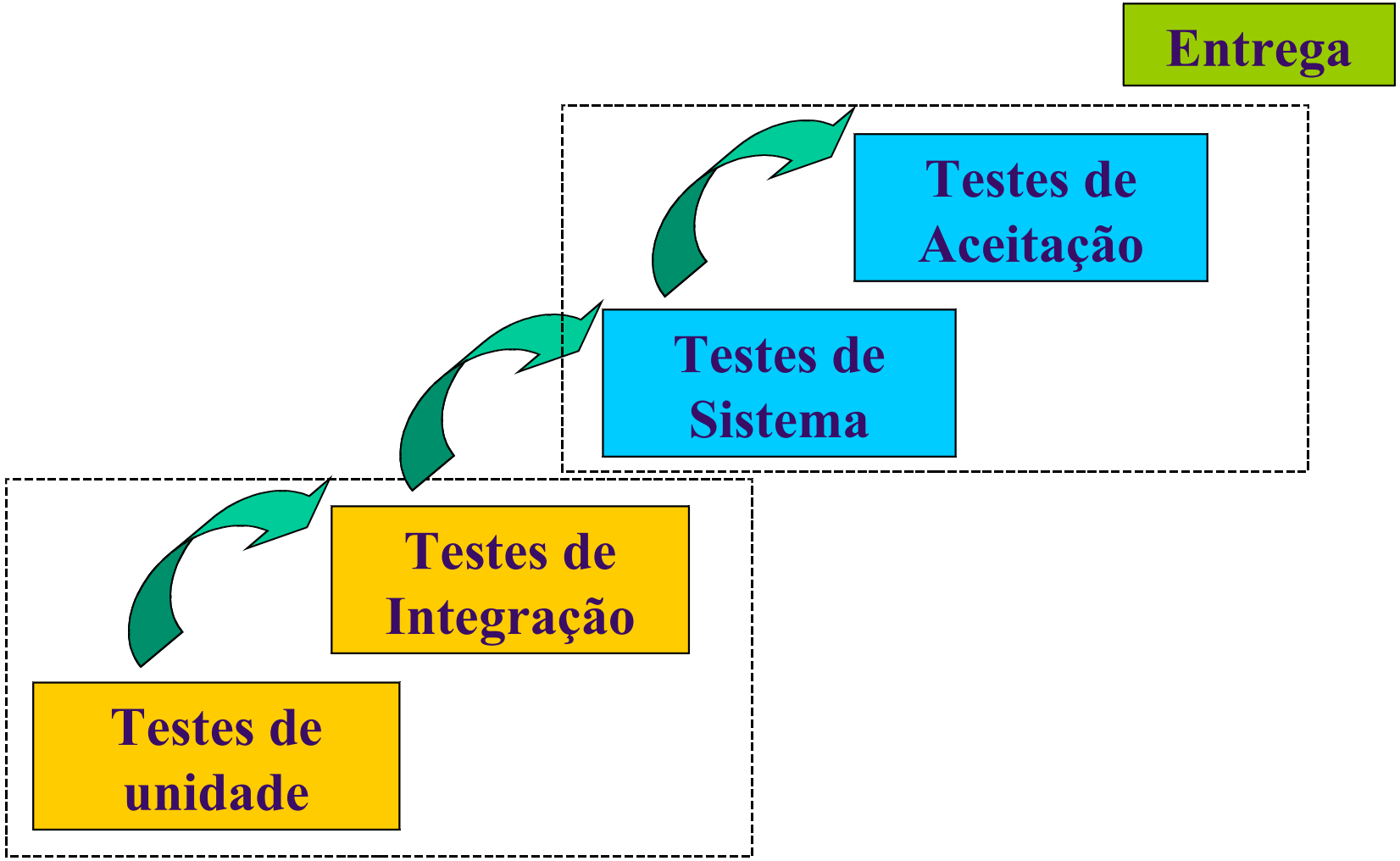
- A perspectiva é sempre a do negócio, ou seja, garantir que o software foi desenvolvido de acordo com os requisitos que foram estabelecidos pelo negócio.

Formando a Equipe de Testes

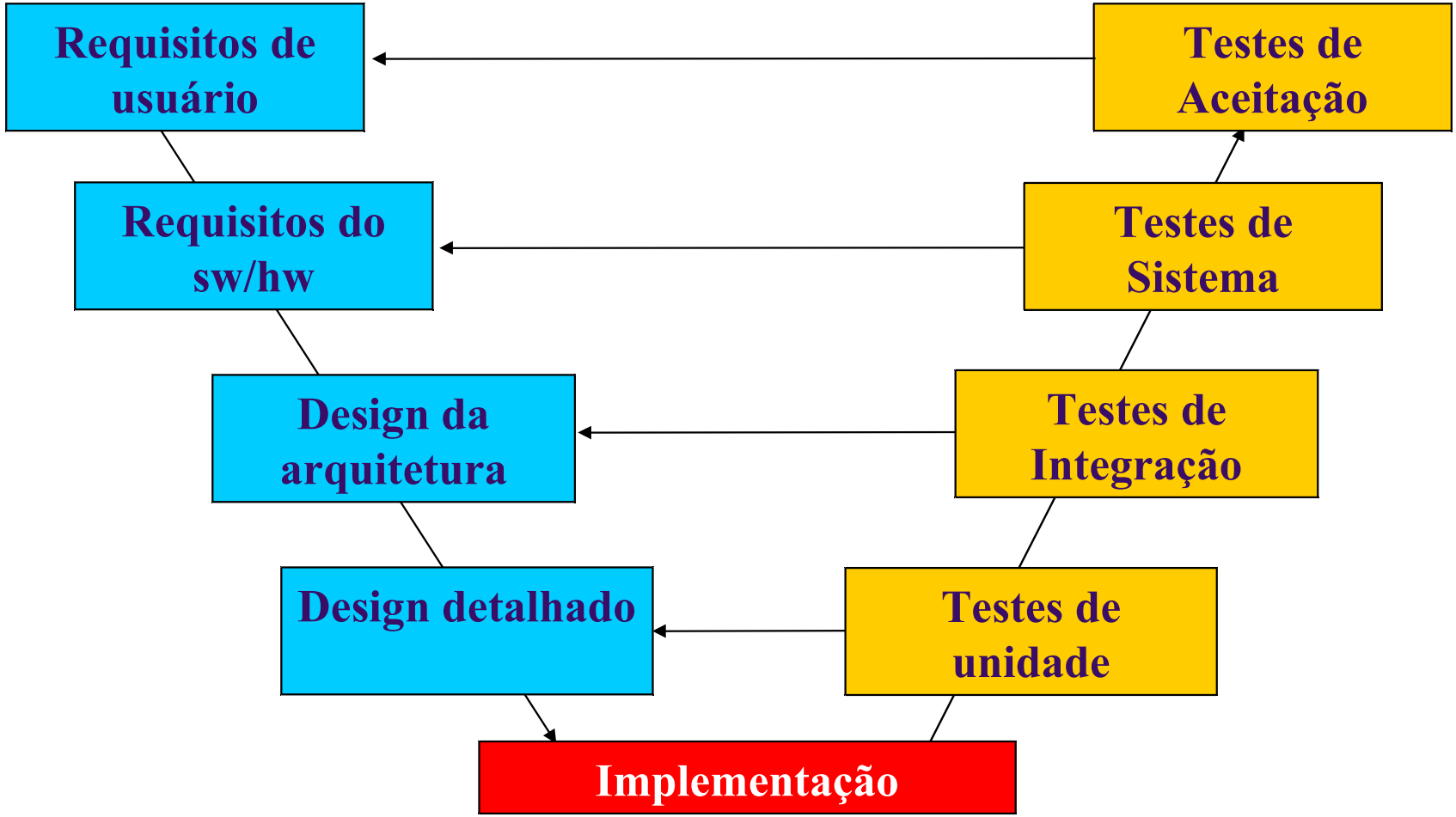
Desvantagens:

- Custos maiores;
- Falta de familiarização com ferramentas;
- Abordagens exclusivas do negócio, esquecendo aspectos técnicos do teste.

Estágios de Teste



Ciclo de Vida



Tipos de Teste

Estáticos ou revisões:

- Revisão técnica: Consiste na apresentação do material para uma equipe de revisão onde será feita a análise do produto de trabalho;
- Inspeção: Consiste na verificação, se os produtos do software e processos estão de acordo com os padrões, *guidelines*, especificações e procedimentos;

(Alguns) Tipos de Teste

- Teste Funcional
- Teste de Recuperação de Falhas
- Teste de segurança e controle de acesso
- Teste de desempenho
- Teste de estresse
- Teste de configuração ou portabilidade
- Teste de interface com o usuário
- Teste de regressão

Abordagens de Teste

- Abordagem funcional(“caixa-preta”)

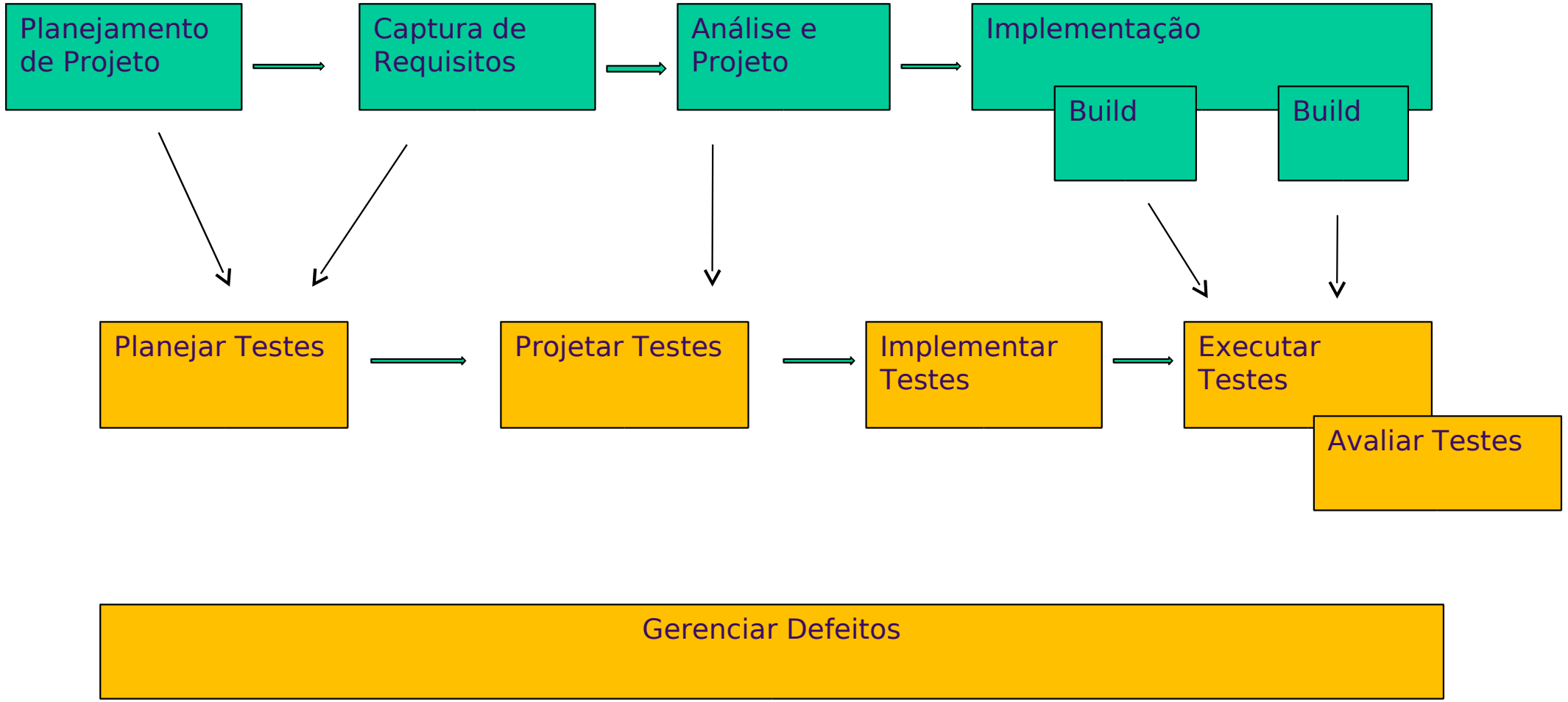
  Os testes são gerados a partir de uma análise dos relacionamentos entre os dados de entrada e de saída

- Abordagem estrutural(“caixa-branca”)

  Os testes são executados a partir de uma análise dos caminhos lógicos (código) possíveis de serem executados.

Relacionando as atividades de Testes com as atividades de Desenvolvimento

Quando começar a testar?



Processo de Teste

- Planejar Testes

- Especificar Testes

- Executar Testes

- Reportar Testes

Planejar Testes

Entradas

- Documento de Requisitos
- Plano de Projeto
- Modelos de Caso de Uso

Saídas

- Plano de Testes

Plano de Testes

Histórico de Revisões

1. Objetivo

2. Requisitos a serem testados

3. Estágios de Teste

4. Tipos de Teste

5. Abordagens de Teste

6. Critérios de parada/aceitação

7. Recursos

8. Matriz de Responsabilidade

9. Cronograma

Projetar Testes

Entradas

- Documento de Requisitos
- Plano de Testes
- Modelo de Caso de Uso

Saídas

- Projeto de Testes(casos e procedimentos)
- Planilha de Teste

Projeto de Testes

Histórico de Revisões

1. Requisitos a serem testados(prioridade)

2. Identificador do caso de Teste

3. Requisitos Associados

3. Casos de Teste

3. Tipo de Teste

4. Pré-condição

4. Dados de entrada

5. Procedimento

6. Resultado esperado

7. Status do teste

Execução de Testes

Entradas

- Projeto de Testes
- Código executável do sistema

Saídas

- Planilha de Teste

Relatório de Testes

- Registrar resultados

- Avaliar resultados

- Encaminhar ao desenvolvedor responsável

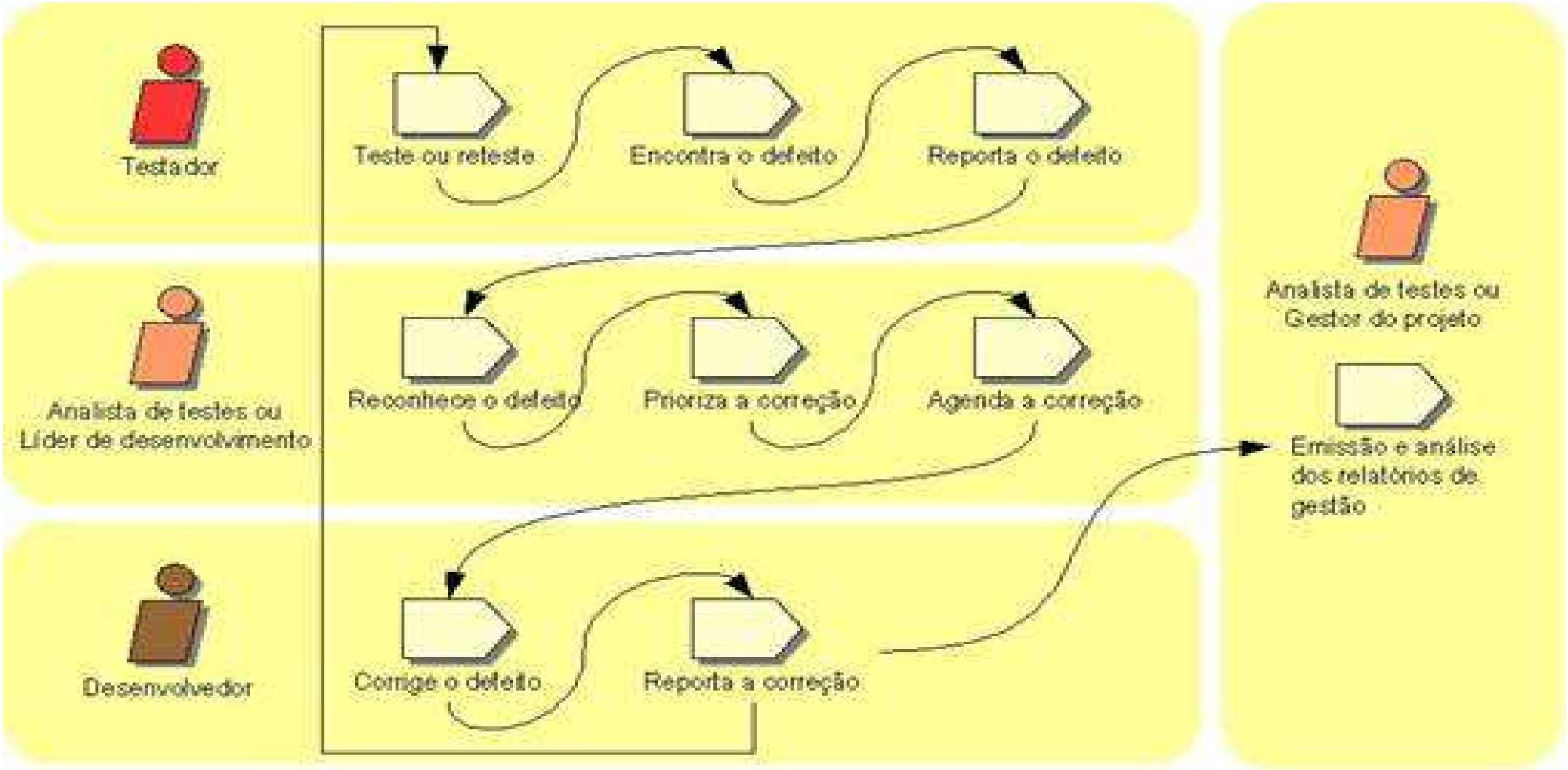
Gerenciamento de bugs

Classificação de defeitos:

1. Faltante: O defeito ocorre em virtude da falta parcial ou total de um requisito;
2. Errado: O defeito ocorre porque o requisito não foi implementado corretamente;
3. Acréscimo: O defeito ocorre em virtude de um comportamento ou elemento que foi implementado mas não foi especificado no requisito.

Gerenciamento de bugs

Ciclo de vida de um defeito



Ferramentas de Teste

- Automatizam atividades do processo de teste
- Podem nos auxiliar em todas as atividades do processo de teste

Ferramentas de planejamento e projeto de testes:

- Elaborar plano de testes. Ex: Project
- Projetar testes: Excel, TestManager
- Executar testes: Excel, TestManager
- Avaliar testes: Excel, TestManager
- Implementação: Junit(unidade), Jtest e C++Test (Análise estática de código)
- Gerência de defeitos: Bugzilla, Mantis

Referências

- ACKERMAN, A., BUCHWALD, L., LEWSKI, F., 1989, “Software Inspections: An Effective Verification Process”, IEEE Software, vol. 6, no. 3, pp.31-37.
- KALINOWSKI, M., SPÍNOLA, R.O., TRAVASSOS, G.H., Infra-Estrutura Computacional para Apoio ao Processo de Inspeção de Software. No:

  Simpósio Brasileiro de Qualidade de Software, 2004, Brasília.
- BOEHM, B. W., BASILI, V.R., 2001, “Software Defect Reduction Top 10 List.”, IEEE Computer 34 (1): 135-137.
- BOEHM, B.W., ABTS, C., BROWN, A.W., CHULANI, S., CLARK, B.K., HOROWITZ, E., MADACHY, R., REIFER, D., STEECE, B., 2000, Software Cost Estimation with COCOMO II, Prentice Hall. BOEHM, B.W., 1981, Software Engineering Economics, Prentice Hall.
- CIOLKOWSKI, M., LAITENBERGER, O., BIFFL, S., 2003, “Software Reviews: The State of the Practice”, IEEE Software 20 (6): 46-