

# Teste de Software 3

# Teste de Software

## **Objetivo:**

**Executar software para revelar erros/falhas ainda não descobertos**

**Pode gastar 40% do esforço de desenvolvimento**

# Teste de Software

**Defeito (fault, defects)**

# Teste de Software

**Defeito (fault, defects)**



algoritmo|mecânico

# Teste de Software

**Defeito (fault, defects) >>**



algoritmo|mecânico >>

# Teste de Software

**Defeito (fault, defects) >> Erro (Bugs)**



algoritmo|mecânico

>>

estado inconsistente

# Teste de Software

**Defeito (fault, defects) >> Erro (Bugs) >>**



algoritmo|mecânico

>>

estado inconsistente >>

# Teste de Software

**Defeito (fault, defects) >> Erro (Bugs) >> Falha (Failure)**



algoritmo|mecânico

>>

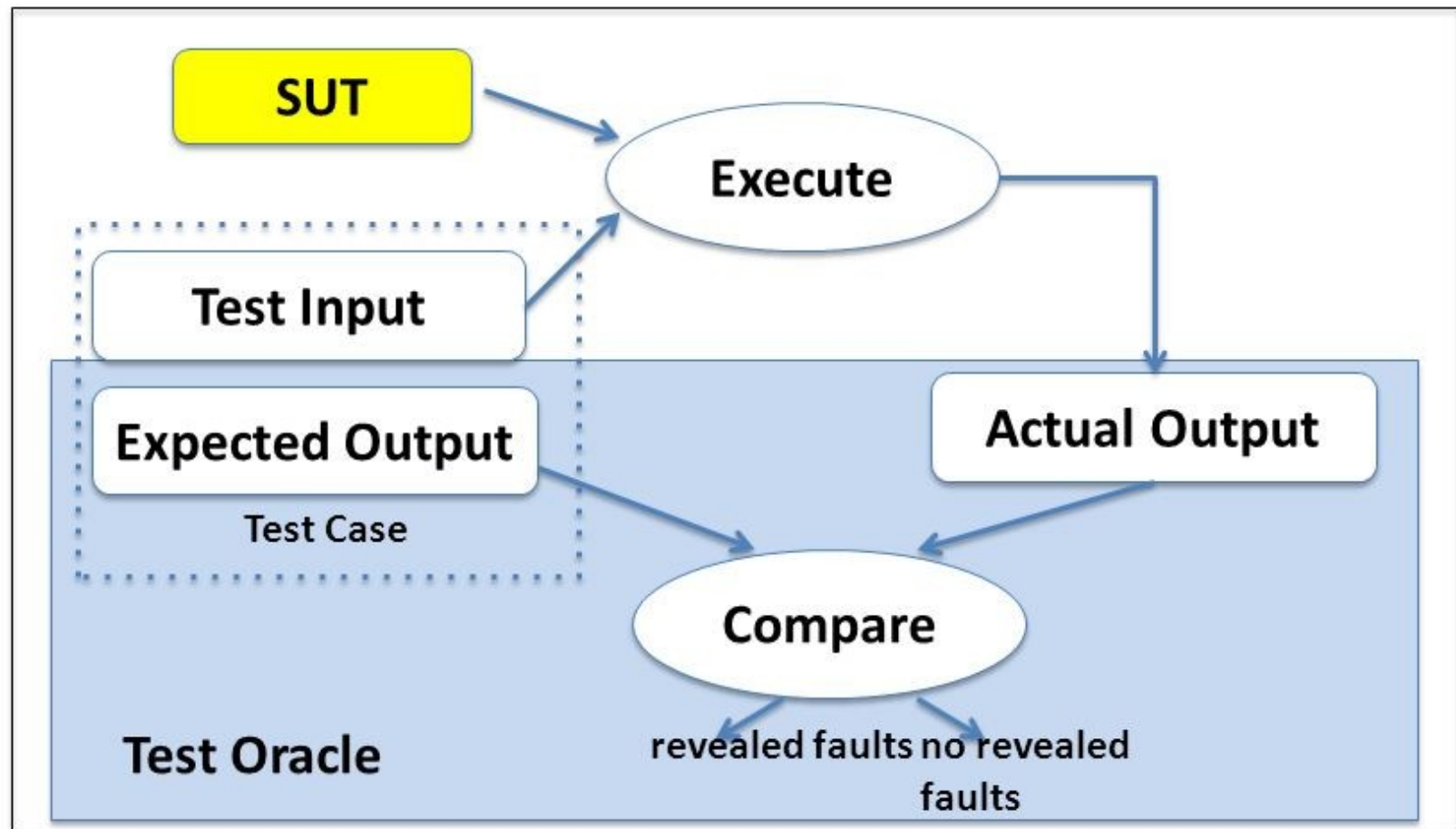
estado inconsistente

>>

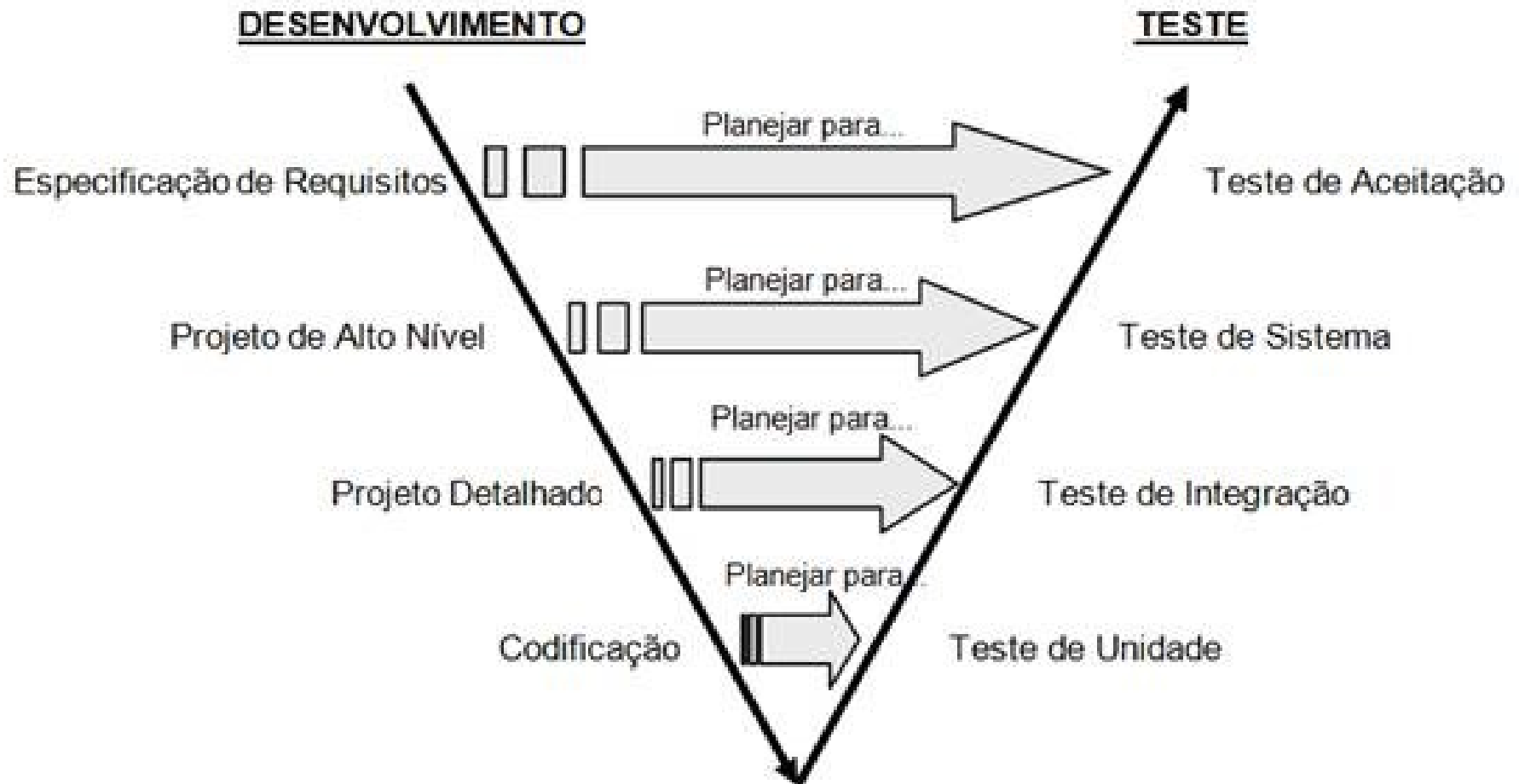
notado



# Simplified Software-Testing Process



# Estratégias de Teste



# Estratégias (fases) de Teste

## Teste de Unidade

- ✓ Identificar erros de lógica e de implementação em cada módulo do software, separadamente
- ✓ Driver → B ; A → Stub

# Estratégias (fases) de Teste

## Teste de Unidade

- ✓ Identificar erros de lógica e de implementação em cada módulo do software, separadamente
- ✓ Driver → B ; A → Stub

## Teste de Integração

- ✓ Identificar erros associados às interfaces entre os módulos do software

# Estratégias (fases) de Teste

## Teste de Unidade

- ✓ Identificar erros de lógica e de implementação em cada módulo do software, separadamente
- ✓ Driver → B ; A → Stub

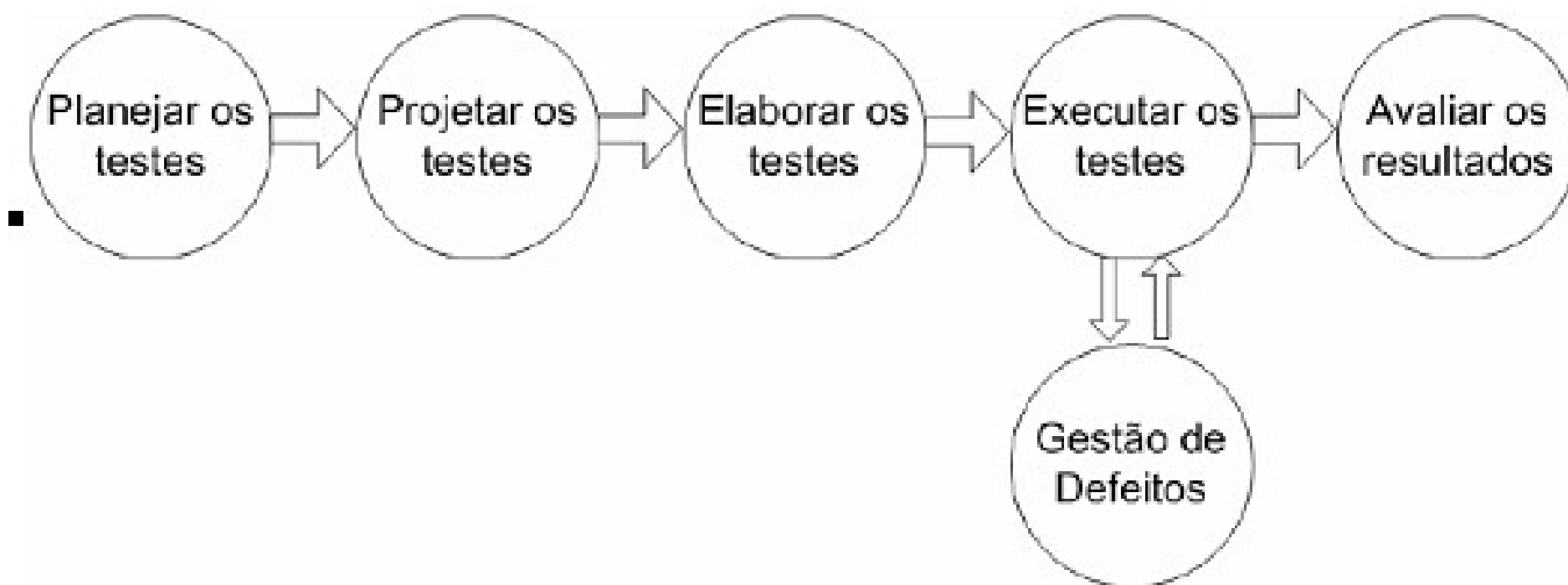
## Teste de Integração

- ✓ Identificar erros associados às interfaces entre os módulos do software

## Teste de Sistema

- ✓ Verificar se as funções estão de acordo com a especificação e se todos os elementos do sistema combinam-se adequadamente

# Ciclo de vida do processo de Teste



# Ciclo de vida do processo de Teste

**Planejar** - selecionar requisitos, artefatos e componentes que serão testados.

# Ciclo de vida do processo de Teste

**Planejar** - selecionar requisitos, artefatos e componentes que serão testados.

**Projetar** – prepara o ambiente onde os testes serão executados



# Ciclo de vida do processo de Teste

**Planejar** - selecionar requisitos, artefatos e componentes que serão testados.

**Projetar** – prepara o ambiente onde os testes serão executados

**Elaborar** – implementar scripts manuais ou automáticos, programas e massas de dados.

# Ciclo de vida do processo de Teste

**Planejar** - selecionar requisitos, artefatos e componentes que serão testados.

**Projetar** – prepara o ambiente onde os testes serão executados

**Elaborar** – implementar scripts manuais ou automáticos, programas e massas de dados.

**Executar** - os testes são executados e relatórios são gerados para posterior avaliação.

# Ciclo de vida do processo de Teste

**Planejar** - selecionar requisitos, artefatos e componentes que serão testados.

**Projetar** – prepara o ambiente onde os testes serão executados

**Elaborar** – implementar scripts manuais ou automáticos, programas e massas de dados.

**Executar** - os testes são executados e relatórios são gerados para posterior avaliação.

**Avaliar resultados** - gera um relatório consolidado dos erros mediante os relatórios gerados. A partir deste o líder toma conhecimento dos erros encontrados ou ocorridos para que possam ser iniciados os acertos necessários.

# Papeis envolvidos no processo de teste



## Gestor da Qualidade

Responsável pelas ações de controle de qualidade dos produtos



## Líder do projeto de Teste

Responsável por liderar e planejar o projeto de teste



## Arquiteto de Teste

Responsável por montar e disponibilizar o ambiente de teste



## Analista de Teste

Responsável pela modelagem e elaboração dos casos de teste e scripts de teste



## Testador

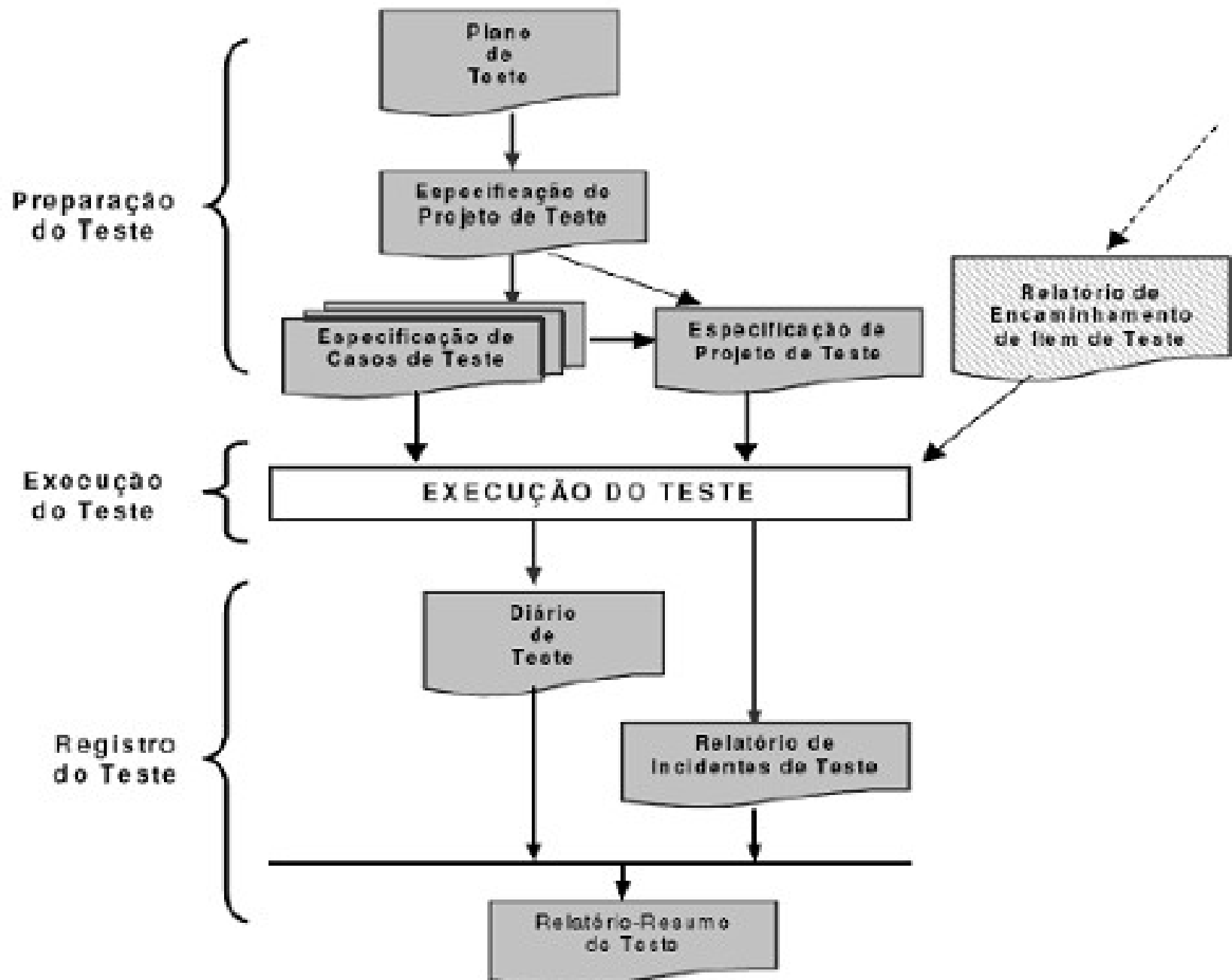
Responsável pela execução dos casos de teste e pela documentação dos resultados



## Usuários

Responsável pela avaliação e aceitação do sistema

# A Norma IEEE 829



# Plano de Teste

**Plano de teste** – apresenta o planejamento para a execução de teste incluindo: abrangência, abordagem, recursos e cronograma. Identifica os itens e as funcionalidades a serem testadas, as tarefas a serem realizadas e os riscos relacionados a atividade de teste. Gera:

**Especificação de teste** – coberta por 3 documentos:

**Especificação do projeto de teste** – refina a abordagem apresentada no plano de teste, identifica as funcionalidades e características a serem testadas pelo projeto e seus testes associados. Também identifica os casos e procedimentos de testes e apresenta critérios de aprovação. Em alguns casos é incluído ou incorporado ao plano de testes.

**Especificação do caso de teste** – define os casos de testes incluindo dados de entrada, resultados esperados, ações e condições gerais para os testes.

**Especificação de procedimento de teste** – especifica os passos para executar os procedimentos de casos de teste.

# Relatórios de testes

Cobertos por 4 documentos:

**Diário de teste** – documenta qualquer evento que ocorra durante a atividade de teste e que requeira análise posterior.

**Relatório Resumo de Teste** – apresenta de forma resumida os conceitos das atividades de teste associados com uma ou mais especificações de projeto de testes e prove avaliações baseadas nesses resultados.

**Relatório de encaminhamento de item de teste** - identifica os itens encaminhados para teste no caso de equipes distintas de desenvolvimento e teste.

**Relatório de incidente de teste** – todos os defeitos encontrados durante o teste são registrados e passados para a equipe de desenvolvimento para as devidas correções.

# Plano de Testes

Descreve o planejamento para execução do teste, incluindo:

- › a estratégia de testes,
- › abrangência,
- › abordagem,
- › recursos e
- › cronograma das atividades de teste.

Identifica

- › itens e
- › funcionalidades a serem testadas,
- › tarefas a serem realizadas e
- › riscos associados com a atividade de teste.



| Modelo PMI         | Modelo IEEE 829   | Modelo QAI   |
|--------------------|---|--|
| • Escopo           | <ul style="list-style-type: none"> <li>• Identificação do Plano de Testes</li> <li>• Referências</li> <li>• Introdução</li> <li>• Funcionalidades a serem testadas</li> <li>• Funções testadas da perspectiva do usuário</li> <li>• Funções não testadas da perspectiva do usuário</li> </ul> | <ul style="list-style-type: none"> <li>• Escopo do Teste</li> <li>• Objetivos de Teste</li> <li>• Premissas</li> <li>• Estrutura do Teste</li> <li>• Ferramentas de Teste</li> </ul> |
| • Custo            | -   | -  |
| • Tempo            | • Cronograma  | <ul style="list-style-type: none"> <li>• Recursos</li> <li>• Cronograma de Testes</li> </ul>   |
| • Qualidade        | <ul style="list-style-type: none"> <li>• Abordagem dos testes</li> <li>• Gerência de Dados de Teste</li> <li>• Critérios de conclusão dos testes</li> <li>• Critérios para interrupção e retomada de testes</li> </ul>  | • Gerência de Dados de Teste   |
| • Integração       | • Ambiente de Testes  | • Ambiente de Testes   |
| • Recursos Humanos | • Pessoal e Responsabilidades   | • Funções / Responsabilidades  |
| • Comunicação      | • Entregáveis   | • Comunicação  |
| • Riscos           | <ul style="list-style-type: none"> <li>• Risco do Processo de Testes</li> <li>• Plano de Riscos e Contingência</li> </ul>   | • Análise de Riscos  |
| • Suprimentos      |   | • Ferramentas de Teste   |

Fonte: Base de Conhecimento em Teste de Software.

# Projeto de Casos de Teste

Visa **projetar dados de entradas e saídas esperadas** que testam o sistema.

**Meta** - criar um conjunto de casos de teste eficazes para descobrir defeitos do programa e demonstrar que o sistema atende aos requisitos.

# Projeto de Caso de Teste

## Como fazer?

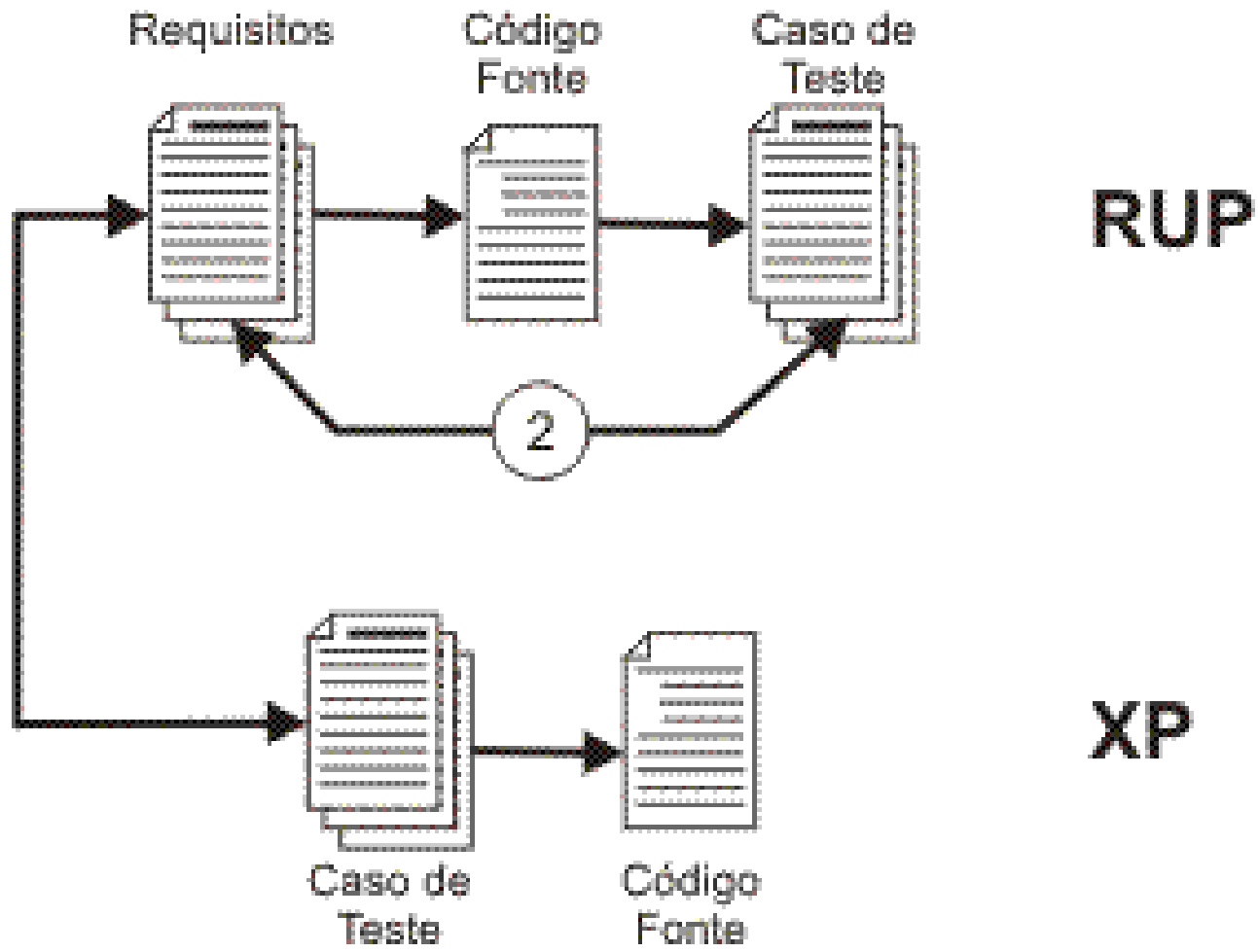
1) **Selecione uma característica** do sistema ou do componente que você vai testar → Critérios de Teste

2A) Depois **selecione um conjunto de dados de entradas** para executar aquelas características → Critérios de Seleção de Dados de Teste

2B) Caso os dados não existam é necessário **gerar o conjunto de dados de teste** para executar as características → Critérios de Geração de Dados de Teste

3) Documente as **saídas esperadas**.

# Projeto de Caso de Teste



# Projeto CT

|                     |   |
|---------------------|---|
| ID                  | CT-001  |
| Caso de Teste:      | Efetuar login no sistema.   |
| Funcionalidade:     | Login   |
| Pré - Condição:     | 1. Possuir usuário válido para efetuar o login.                               |
| Procedimento:       | 1. Acessar o sistema;<br>2. Inserir usuário e senha;<br>3. Clicar em "Login". |
| Resultado Esperado: | 1. Login do usuário efetuado com sucesso.                                     |

|                     |   |
|---------------------|---|
| ID                  | CT-002  |
| Caso de Teste:      | Cadastrar cliente no Sistema  |
| Funcionalidade:     | Cadastro de cliente   |
| Pré - Condição:     | Estar logado no sistema   |
| Procedimento:       | 1. Entrar na tela "Cadastro" -> "Cliente"<br>2. Inserir dados do cliente.<br>3. Clicar em "Salvar". |
| Resultado Esperado: | 1. Cliente é cadastrado com sucesso..   |

# Atividades do Trabalho 3 - para 25/05

## Parte I - V&V

### Para as revisões

b) → Ler Seções 15.4, 15.5 e 15.6 Pressman

b) → Para cada cenário: preparar um plano de revisões para cada fase

c) → **Para cada cenário e para cada artefato construído, aplicar o plano de revisões**

# Atividades do Trabalho 3 - para 25/05

## Parte I - V&V

### Para os testes

b) → Para cada cenário: preparar os planos de teste de cada fase (estratégia) de teste. Considerar *drivers* e *stubs*

**c) 1. escolher um caso de uso de complexidade média (considerada no cálculo PCU)**

**c) 2. gerar um modelo conceitual referente a este caso de uso**

**c) 3. a partir do modelo conceitual, evoluir para o diagrama de classes do Design**

**c) 4. a partir deste diagrama de classes, implementar 5 métodos**

