Testes de software

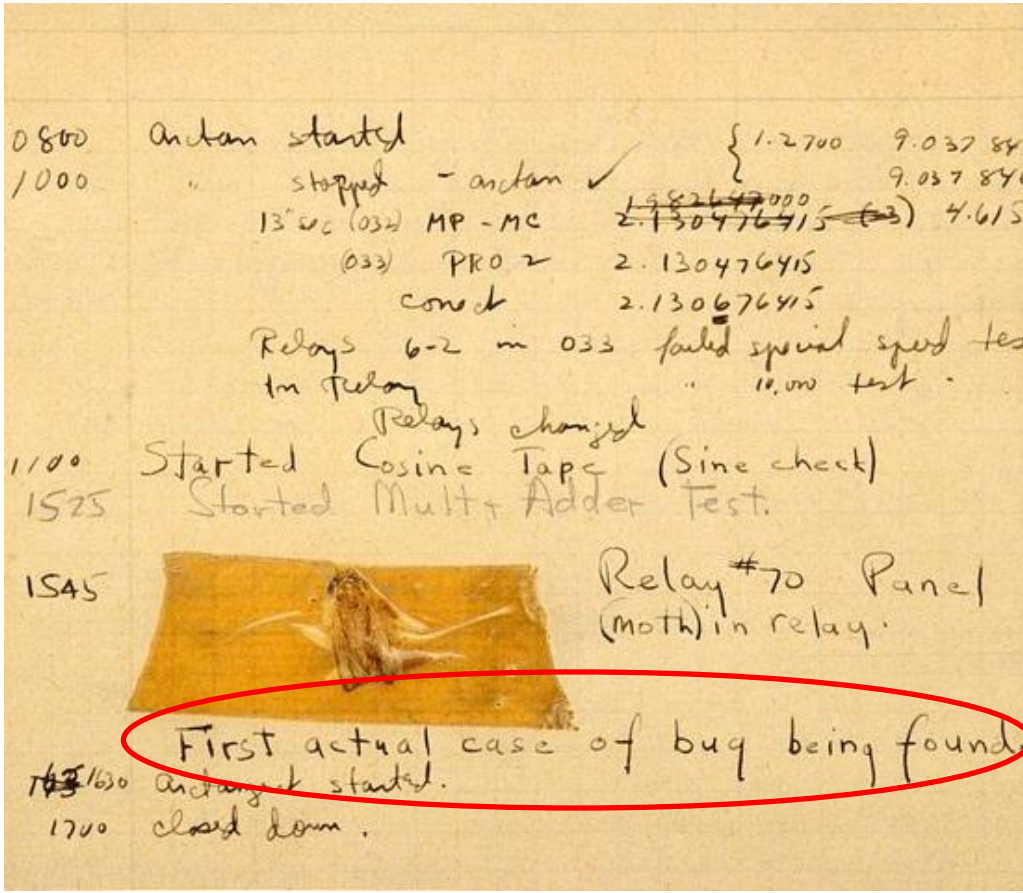
Livro: Pressman

Engenharia de Software I

Bugs famosos

- Primeiro bug: computador Mark II
- Universidade de Harvard em 1945

- O inseto foi descoberto por Grace Hoper ao verificar o motivo da pane no computador. Identificou uma mariposa nos contatos de um relê era a causa do problema.
- O fato ocorreu em 1945 e acredita-se que foi ele que deu a origem do temo “bug” como erro do computador.
- Grace tirou o inseto e colocou em seu caderno de anotações e escreveu: “primeiro caso de bug realmente encontrado”



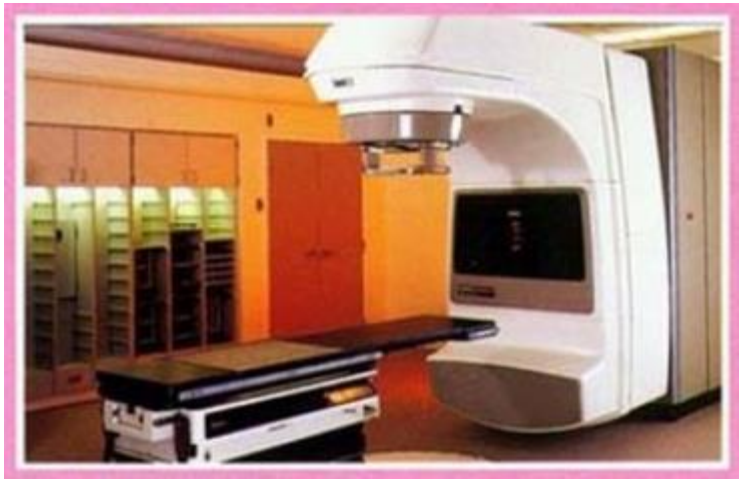
Bugs famosos

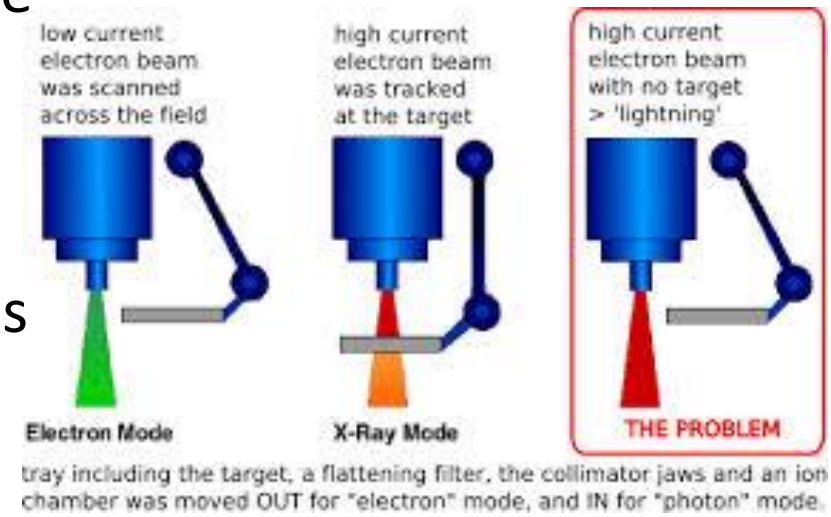
- **Desastre:** Um foguete com uma sonda espacial para Vênus, foi desviado de seu percurso de voo logo após o lançamento. O controle da missão destruiu o foguete 293 segundos após a decolagem (1962).
- **Custo:** 18,5 milhões dólares
- **Causa:** Um programador, ao passar para o computador uma fórmula que haviam lhe entregado escrita manualmente, se esqueceu de uma barra. Sem ela, o software tratava variações normais de velocidade como se fossem sérios problemas, causando falhas por tentativas de correções que acabaram por enviar o foguete fora do curso.

https://pt.wikipedia.org/wiki/Mariner_1

Bugs famosos

- **Desastre:** A máquina de radiação Therac-25 irradiou doses letais em pacientes (1985)
- **Custo:** Três mortos e três seriamente feridos
- **Causa:** O projeto continha travas de hardware para prevenir que o feixe de elétrons de alta intensidade fosse aplicado sem o filtro estar em seu lugar. *Overflows* podiam fazer o software não executar procedimentos de segurança





https://pt.wikipedia.org/wiki/Therac-25

Bugs famosos

- **Desastre:** Durante a primeira Guerra do Golfo, um sistema americano de mísseis na Arábia Saudita falhou ao interceptar um míssil vindo do Iraque. O míssil destruiu acampamentos americanos (1991)
- **Custo:** 28 soldados mortos e mais de 100 feridos.
- **Causa:** Um erro de arredondamento no software calculou incorretamente o tempo, fazendo com que o sistema ignorasse os mísseis de entrada. A cada 100 horas o relógio interno do sistema desviava 1/3 de segundo

Bugs famosos

- **Desastre:** Software utilizado para analisar dados de pesquisa científica. O jornal *The New England Journal of Medicin*e relatou aumento das taxas de suicídio depois de graves desastres naturais. Infelizmente, estes resultados mostraram-se incorretos.

- **Custo:** Credibilidade da ciência

- **Causa:** Um erro no programa mostrava a taxa de suicídios por ano como o dobro do seu valor real, o que foi suficiente para inutilizar toda a pesquisa.

Bugs famosos

- Bug do milênio (ano 2000)
- Datas armazenadas com apenas 2 dígitos
- Foi uma histeria para alterar e testar os sistemas

Garantia da Qualidade X Testes

- A **atividades de garantia da qualidade** de um **produto de software** é o **teste**, para certificar se de sua aderência aos requisitos especificados:
  - Eliminar erros
  - Errar é humano
  - Aumentar a qualidade
  - Reduzir os custos

O que é testar?

- Testar é o processo de **executar um programa** ou sistema com a intenção de **encontrar defeitos** *(Myer, 1979)*
  - *Objetivo:* Demonstrar que o software atende aos requisitos

- Testar é **verificar se o software está fazendo o que deveria fazer**, de acordo com seus requisitos *(Rios e Moreira, 2002)*
  - *Objetivo:* Descobrir situações em que o software se comporta de maneira incorreta, indesejável ou de forma diferente das especificações

Garantia e Controle da Qualidade

Validação → **Estamos construindo o produto certo?** *(avaliação do atendimento aos requisitos)*

Verificação → **Estamos construindo o produto de forma correta?** *(avaliação da aderência aos padrões da empresa e sem falhas)*

Testes → Atividades de V&V dinâmica

Teste de software

- Se **executa** um programa ou modelo utilizando algumas entradas de dados
- Após se **verificar** se o **comportamento** está de acordo com o **esperado**.
- Se os resultados obtidos coincidem com os resultados esperados, então nenhum defeito foi identificado

  →“***O software passou no teste***”

- Se o **resultado** obtido for **diferente do esperado**, então um defeito foi detectado

  → “***O software não passou no teste***”

Teste de software

- A **idéia básica dos testes** é que os **defeitos podem se manifestar** por **meio de falhas observadas durante** a **execução do software**.

- As falhas podem ser resultado de:
  - **uma especificação errada ou falta de requisito,**
  - **de um requisito impossível de implementar considerando o hardware e o software estabelecidos,**
  - **o projeto pode conter defeitos ou**
  - **o código pode estar errado.**
- Assim, **uma falha é o resultado de um ou mais defeitos** (PFLEEGER, 2004).

Importância dos testes

- Investir em **testes** é uma **boa estratégia** para as empresas de desenvolvimento **diminuírem os custos diretos** (manutenção, suporte e retrabalho
- Contribui no **aumento da qualidade** dos produtos
- Melhora a **satisfação dos clientes**

Teste de Software

- Do **ponto de vista psicológico**, o teste de software é uma atividade com um certo **viés destrutivo**, ao contrário de outras atividades do processo de software.

Mitos a serem eliminados

O testador é inimigo do desenvolvedor

A equipe de testes pode ser montada com os desenvolvedores menos qualificados

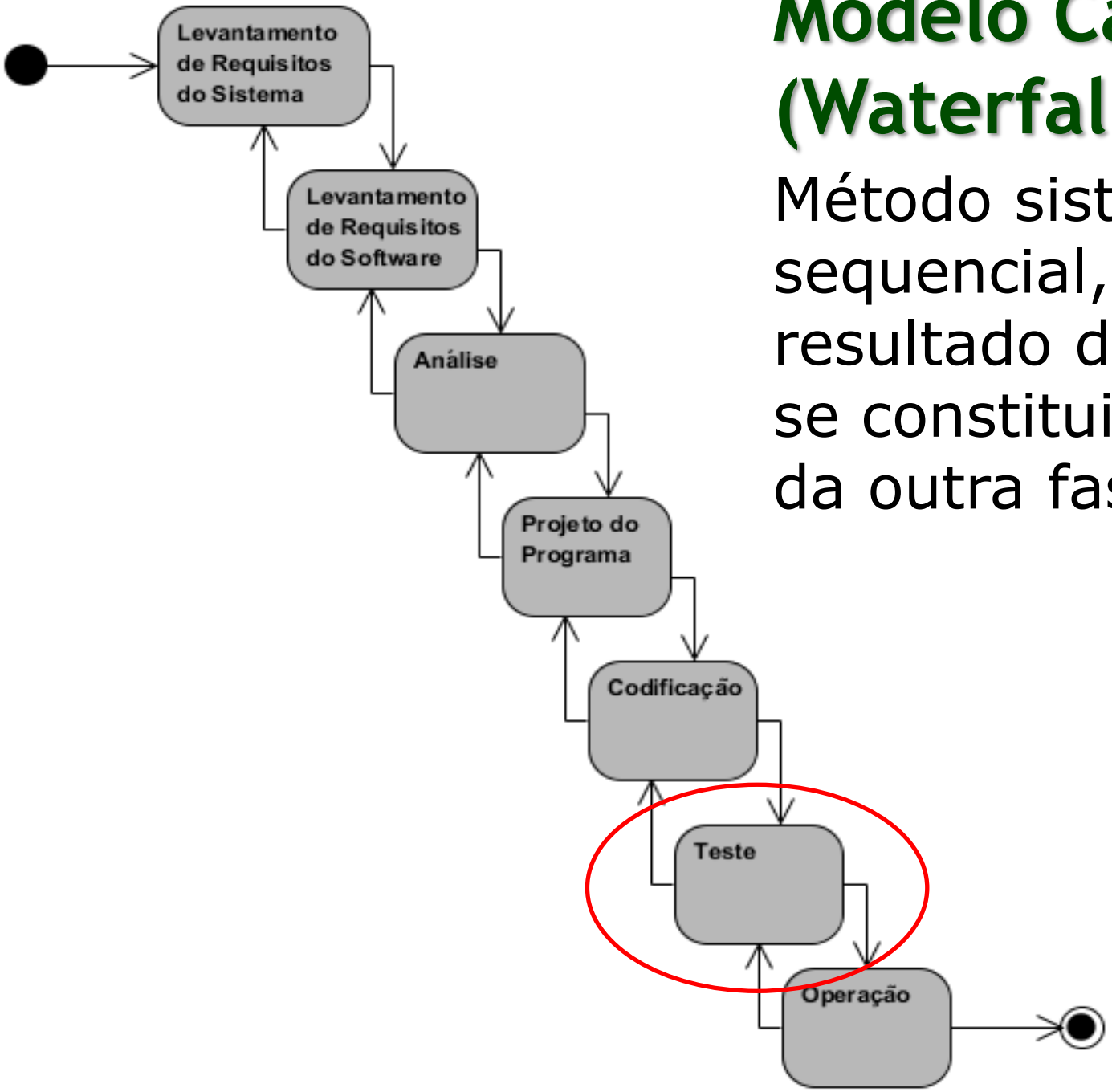
Quando o software estiver pronto deverá ser testado pela equipe de testes

Importante

- O objetivo de um **processo de teste** é **minimizar os riscos** causados por **defeitos** provenientes do **processo de desenvolvimento**.

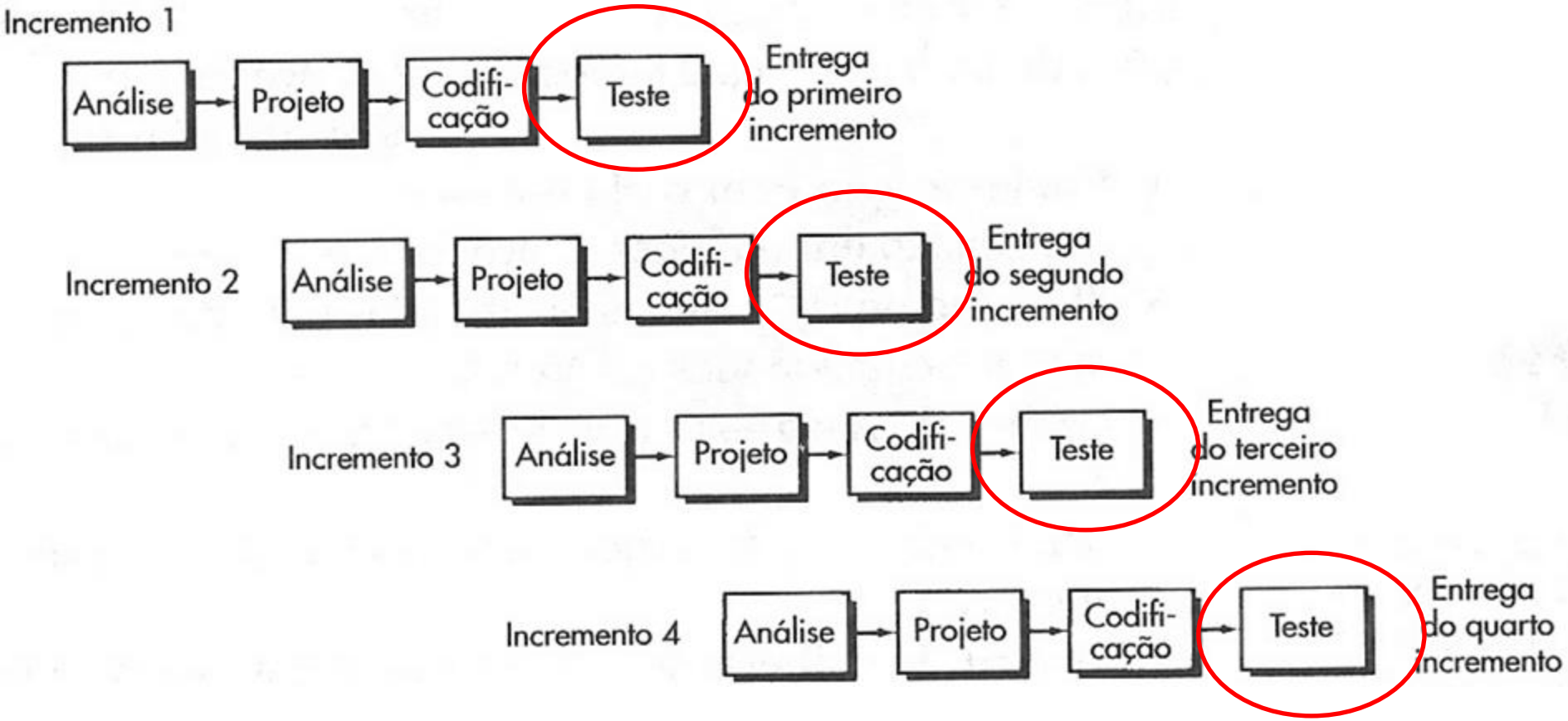
- O planejamento dos testes **deve iniciar com o projeto de construção do software** (parte do plano de projeto)

Modelo Cascata (Waterfall)



Método sistemático e sequencial, em que o resultado de uma fase se constitui na entrada da outra fase.

Modelo incremental e iterativo

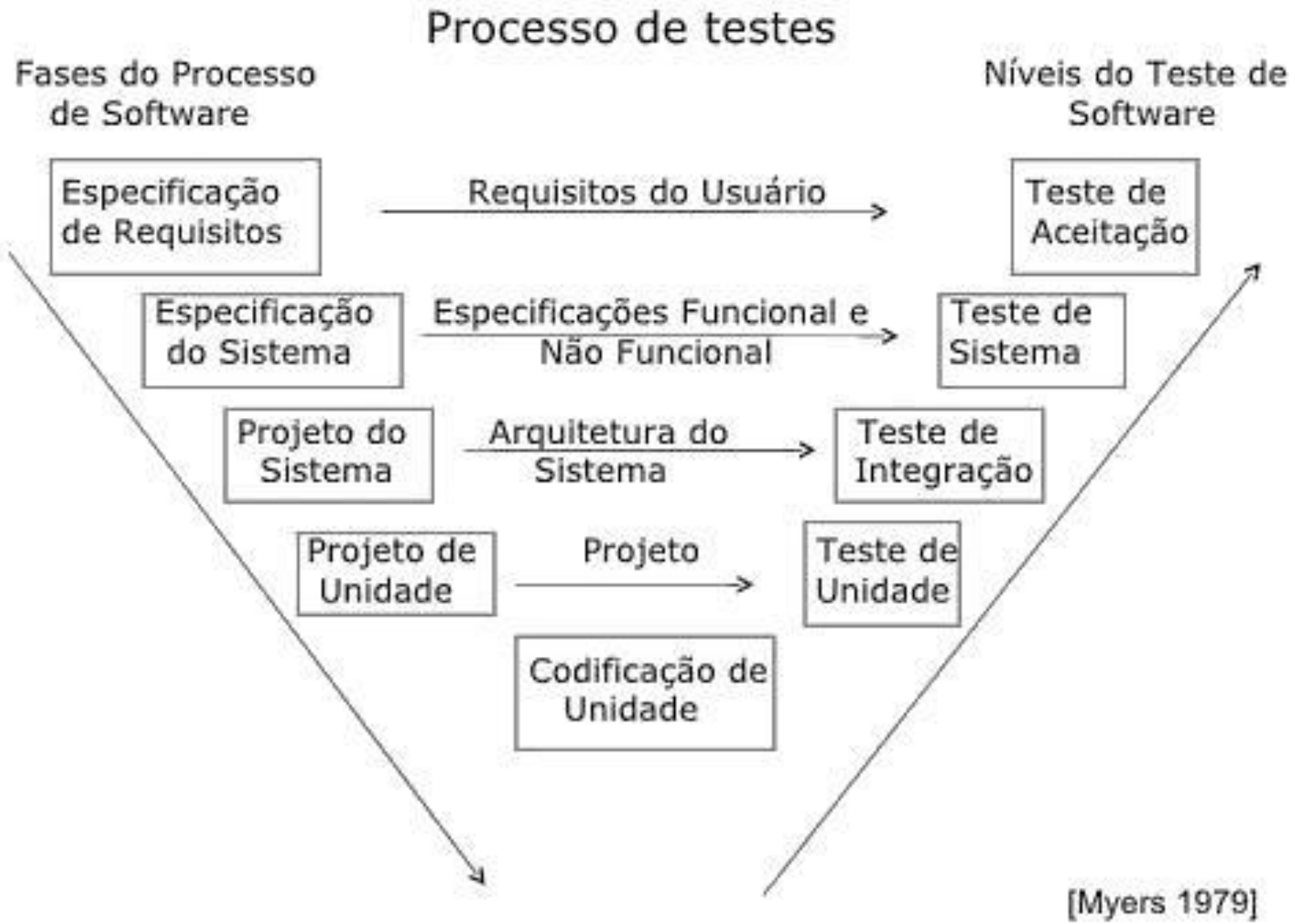


Testar no final de cada iteração é o suficiente?

Desafio

Desafio:

- Como integrar o processo de teste ao longo de todo o ciclo de vida do sistema, não sendo apenas uma atividade a ser desenvolvida após o desenvolvimento?


Processo de testes
Fases do Processo de Software
Níveis do Teste de Software
Especificação de Requisitos
Requisitos do Usuário
Teste de Aceitação
Especificação do Sistema
Especificações Funcional e Não Funcional
Teste de Sistema
Projeto do Sistema
Arquitetura do Sistema
Teste de Integração
Projeto de Unidade
Projeto
Teste de Unidade
Codificação de Unidade
[Myers 1979]

Validação

Estamos construindo o produto certo?
(avaliação do atendimento aos requisitos)

Processo de testes

Fases do Processo de Software

Níveis do Teste de Software

Especificação de Requisitos

Requisitos do Usuário

Teste de Aceitação

Especificação

Especificações Funcional e

Teste de

Teste de unidade e integração: perspectiva dos projetistas e desenvolvedores

Projeto do Sistema

Arquitetura do Sistema

Teste de Integração

Projeto de Unidade

Projeto

Teste de Unidade

Codificação de Unidade

Verificação

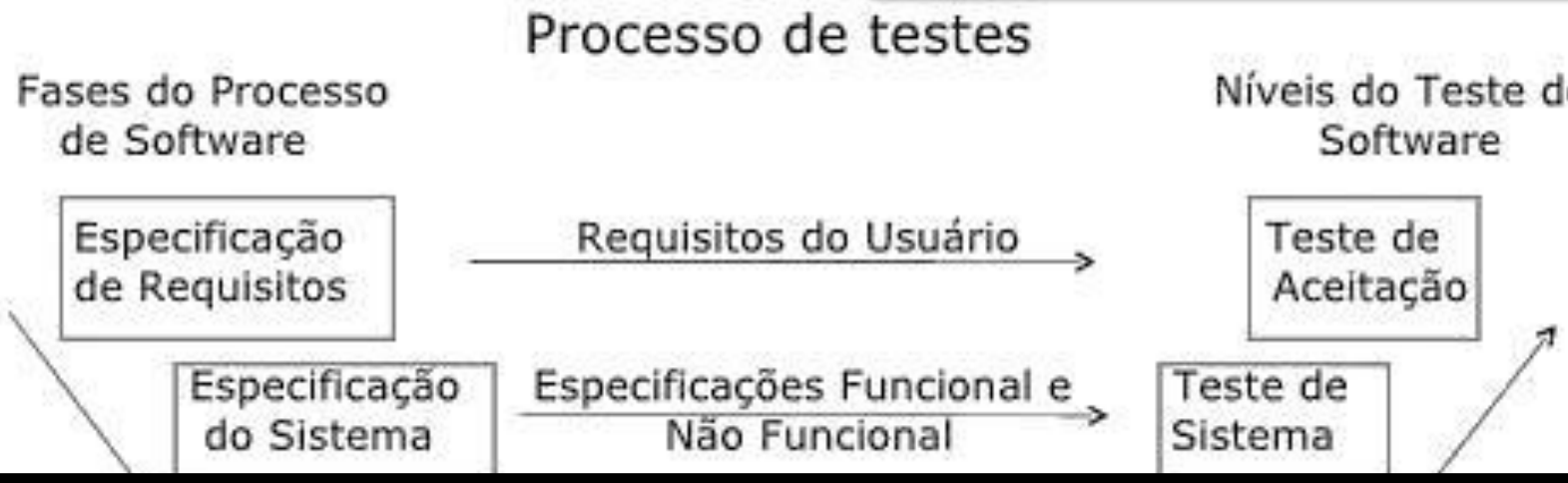
Estamos construindo o produto de forma correta?
(avaliação da aderência aos padrões da empresa e sem falhas)

[Myers 1979]

Testes de sistemas e aceitação: perspectiva do cliente e usuários

Validação

Estamos construindo o produto certo?
(avaliação do atendimento aos requisitos)

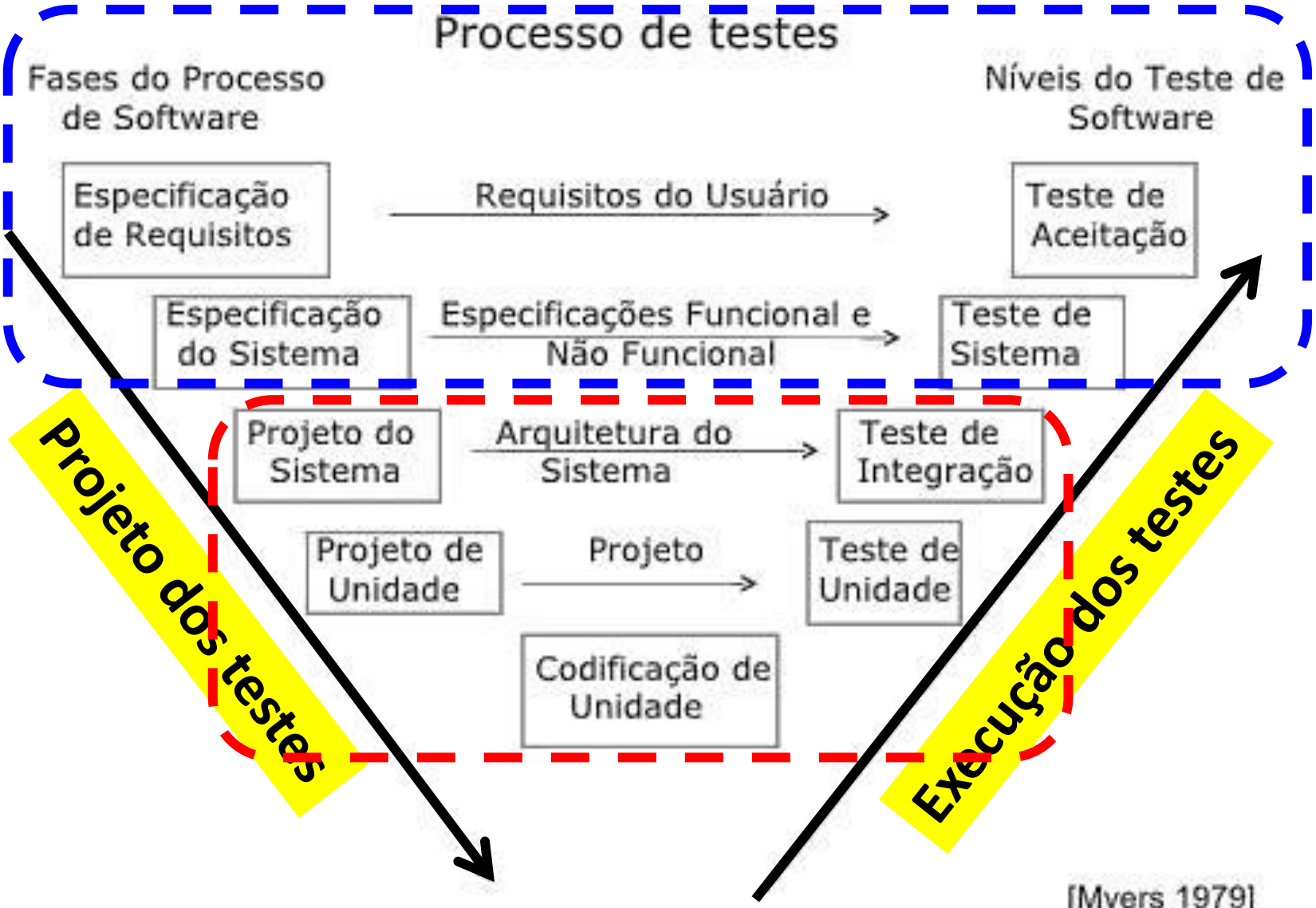


Verificação

Estamos construindo o produto de forma correta?
(avaliação da aderência aos padrões da empresa e sem falhas)

[Myers 1979]

PLANO DE TESTES

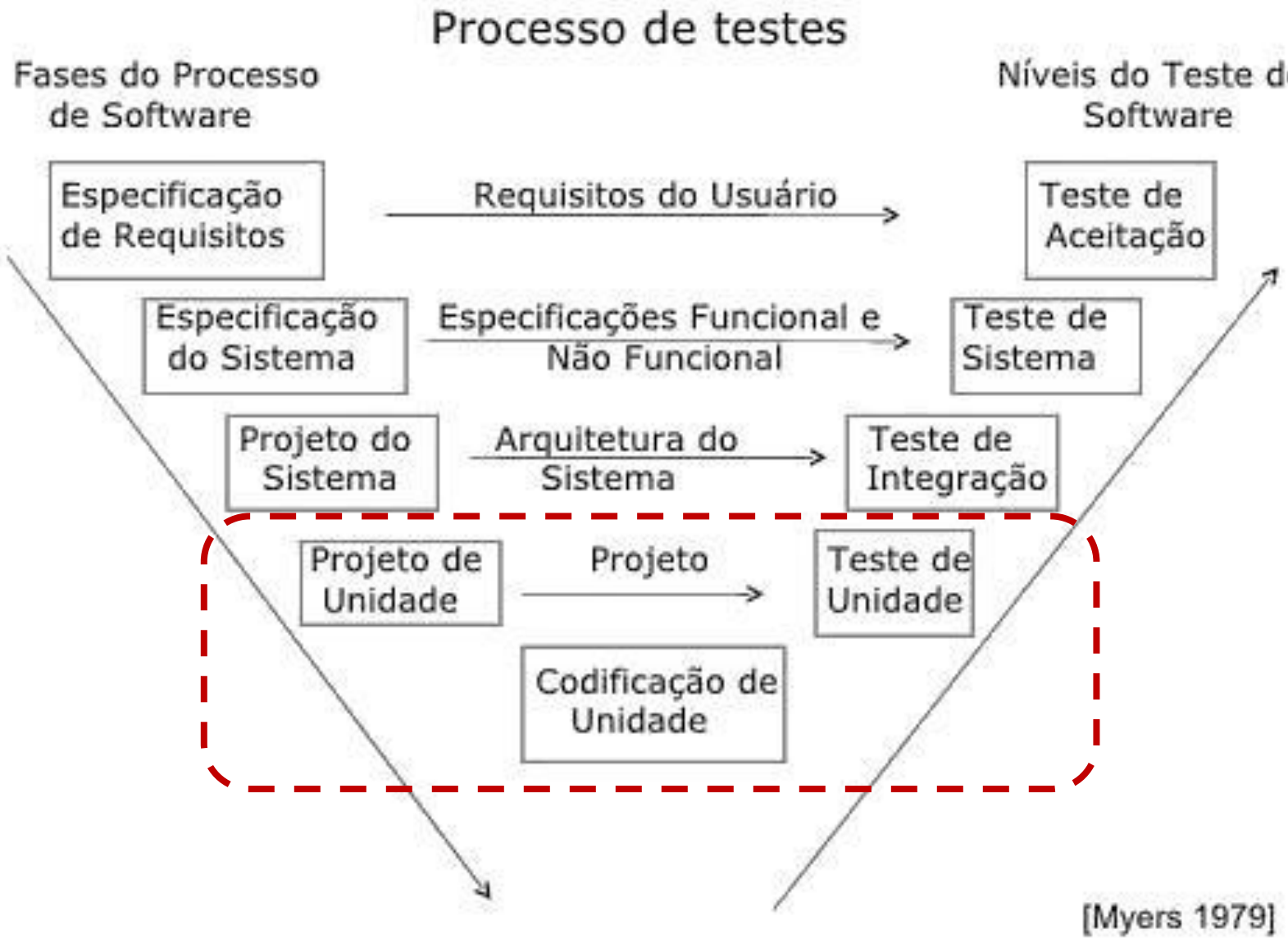


Modelo em V

A importância em estudar o Modelo em V é a **associação dos testes em todas as fases do processo de desenvolvimento**, relacionado as **atividades que devem ser testadas** para **garantir a entrega de um produto de qualidade ao cliente.**

1. NÍVEIS DE TESTES
(Quando testar)

Modelo em V


Processo de testes
Fases do Processo de Software
Níveis do Teste de Software
Especificação de Requisitos
Requisitos do Usuário
Teste de Aceitação
Especificação do Sistema
Especificações Funcional e Não Funcional
Teste de Sistema
Projeto do Sistema
Arquitetura do Sistema
Teste de Integração
Projeto de Unidade
Projeto
Teste de Unidade
Codificação de Unidade
[Myers 1979]

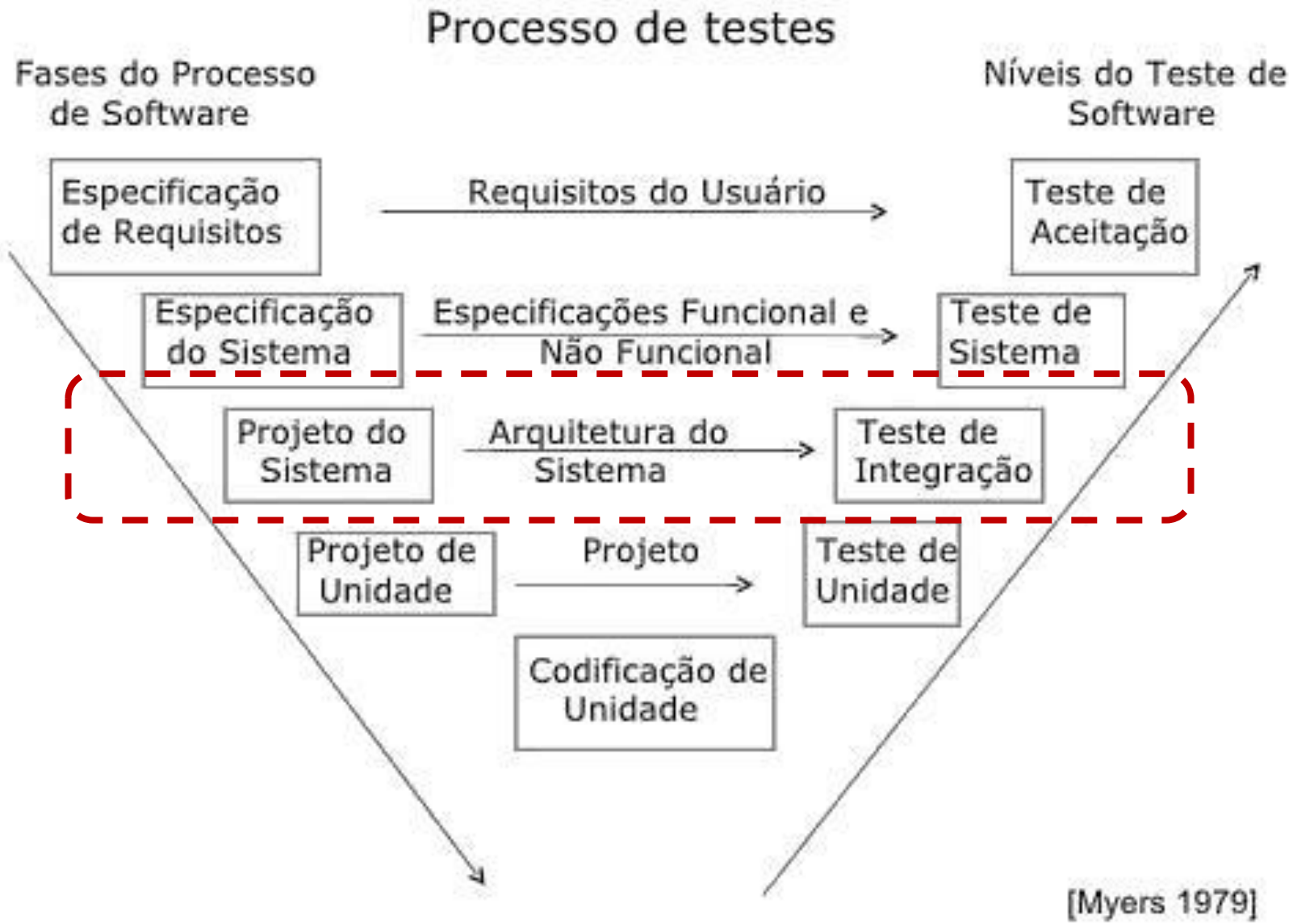
Testes de Unidade

- Também conhecido como testes unitários;
- Objetivo: **explorar a menor unidade do projeto procurando** provocar falhas ocasionadas por **defeitos** de lógica e de implementação em cada módulo, separadamente.
- Alvo do teste: métodos dos objetos ou mesmo pequenos trechos de código;
- São aplicados de maneira individual a cada unidade do sistema;
- Cada unidade do sistema é verificada (testada) de forma isolada;
- Normalmente é realizado pelo próprio programador.

Testes de Unidade

Cuidar para não testar apenas o cenário feliz

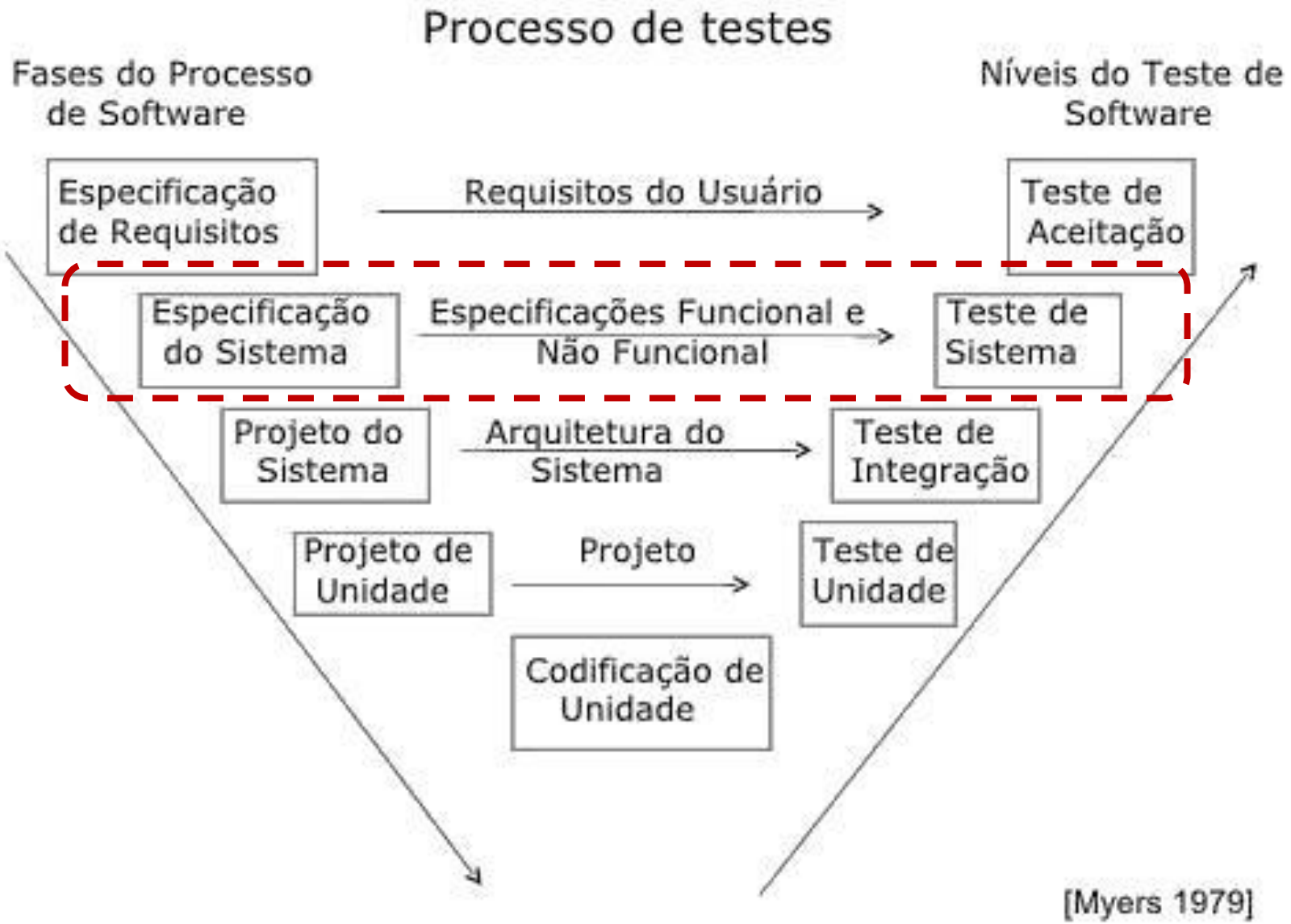
Modelo em V


Processo de testes
Fases do Processo de Software
Níveis do Teste de Software
Especificação de Requisitos
Requisitos do Usuário
Teste de Aceitação
Especificação do Sistema
Especificações Funcional e Não Funcional
Teste de Sistema
Projeto do Sistema
Arquitetura do Sistema
Teste de Integração
Projeto de Unidade
Projeto
Teste de Unidade
Codificação de Unidade
[Myers 1979]

Teste de Integração

- Objetivo é **apontar as falhas decorrentes da integração entre as unidades**.
- É o processo de verificar a interação entre os componentes.
- Visa provocar falhas associadas às interfaces entre os módulos.
- Para que esta fase seja executada, os módulos já devem ter passado pelos testes unitários.

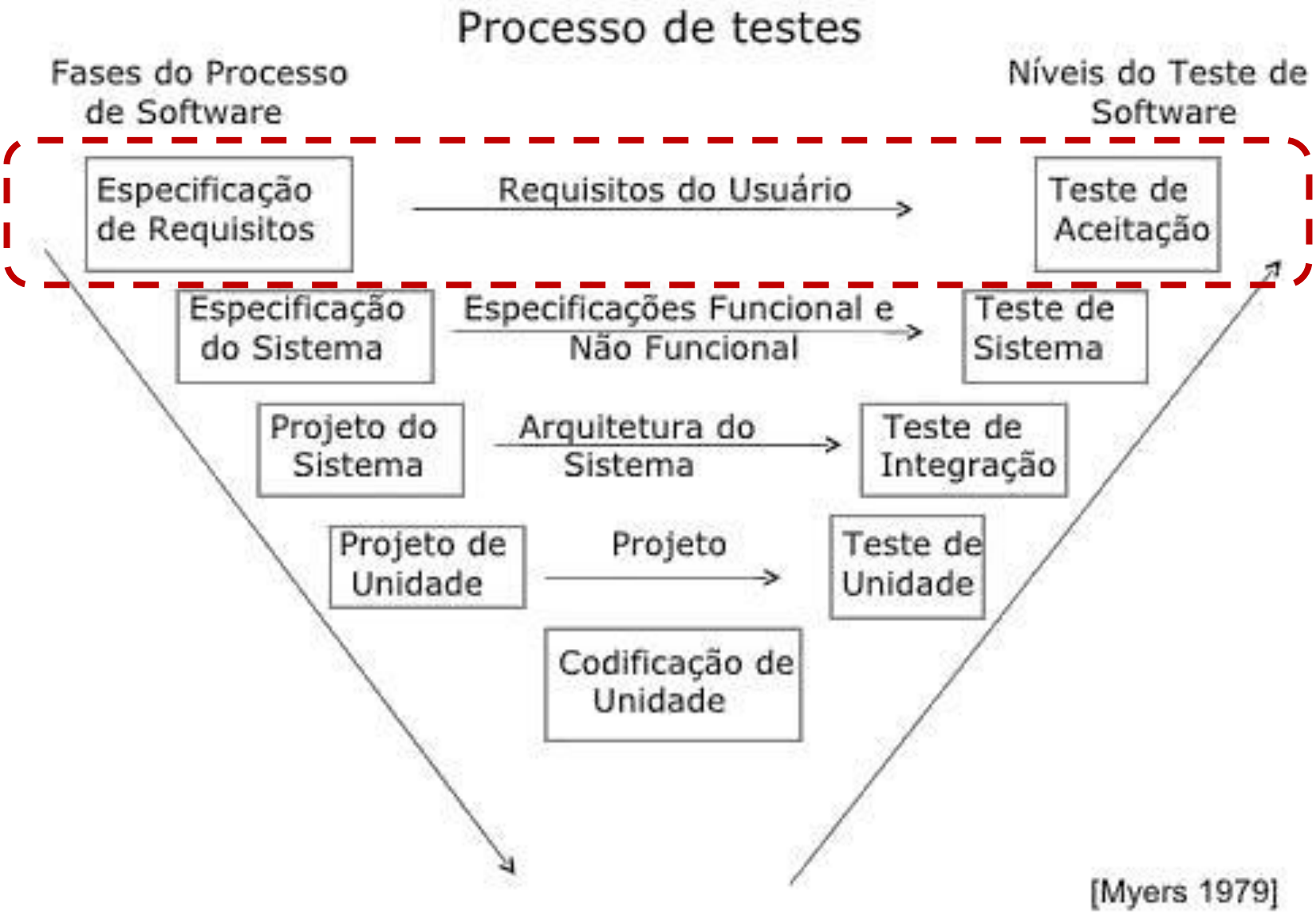
Modelo em V


Processo de testes
Fases do Processo de Software
Níveis do Teste de Software
Especificação de Requisitos
Requisitos do Usuário
Teste de Aceitação
Especificação do Sistema
Especificações Funcional e Não Funcional
Teste de Sistema
Projeto do Sistema
Arquitetura do Sistema
Teste de Integração
Projeto de Unidade
Projeto
Teste de Unidade
Codificação de Unidade
[Myers 1979]

Teste de Sistema

- **Verificação** de cada uma das funcionalidades do sistema (**funcionais e não funcionais**).
- Avalia o software em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final.
- Dessa maneira:
  - **os testes são executados nos mesmos ambientes**
  - **com as mesmas condições**
  - **com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia**
- Nesta etapa o software é testado por completo.

Modelo em V

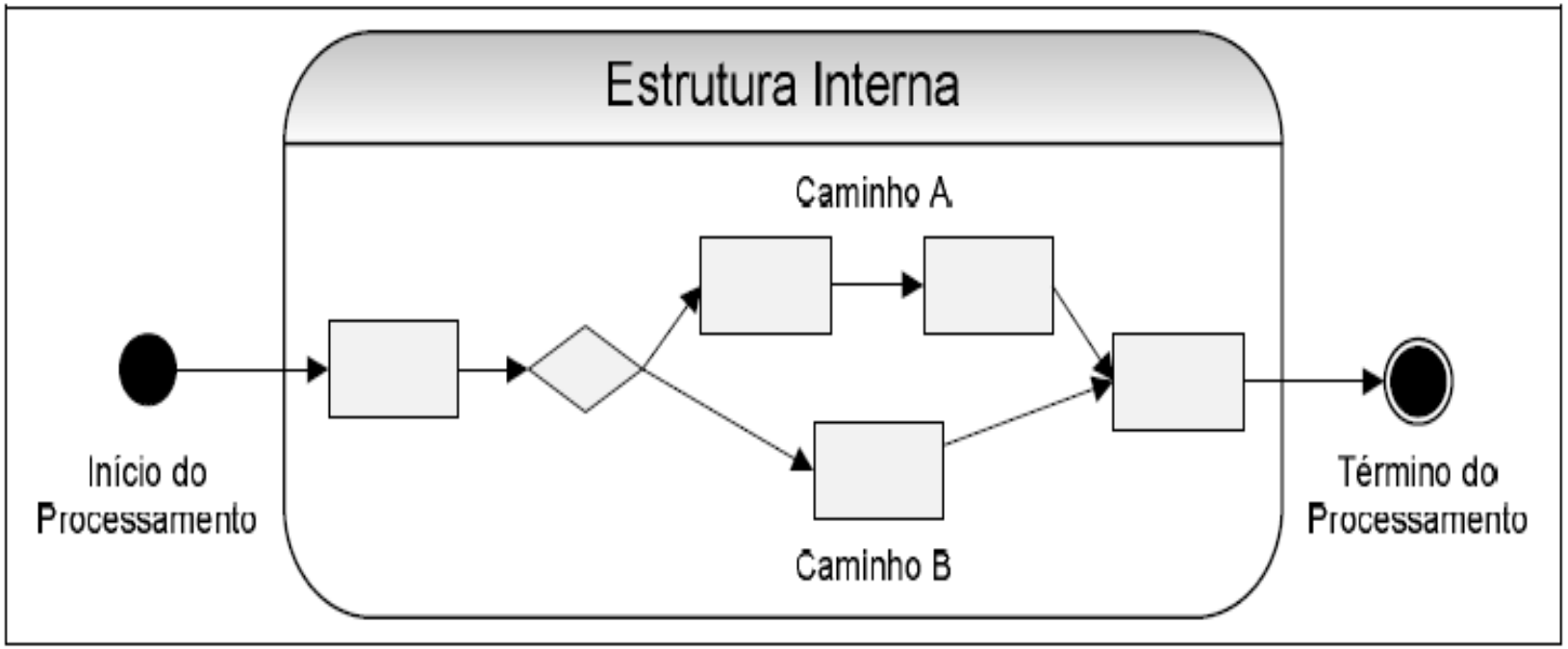

Processo de testes
Fases do Processo de Software
Níveis do Teste de Software
Especificação de Requisitos
Requisitos do Usuário
Teste de Aceitação
Especificação do Sistema
Especificações Funcional e Não Funcional
Teste de Sistema
Projeto do Sistema
Arquitetura do Sistema
Teste de Integração
Projeto de Unidade
Projeto
Teste de Unidade
Codificação de Unidade
[Myers 1979]

Teste de Aceitação

- O software é testado pelo usuário final
- Objetivo demonstrar a conformidade com os requisitos definidos pelo usuário
- Testa as funcionalidades do sistema
- Realizados por um restrito grupo de usuários finais do sistema
- Simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.
- **Nesta fase o cliente confirma se todas as suas necessidades foram atendidas pelo sistema.**

2. TÉCNICAS DE TESTES
(Como testar)

Técnicas de Teste
Caixa Branca



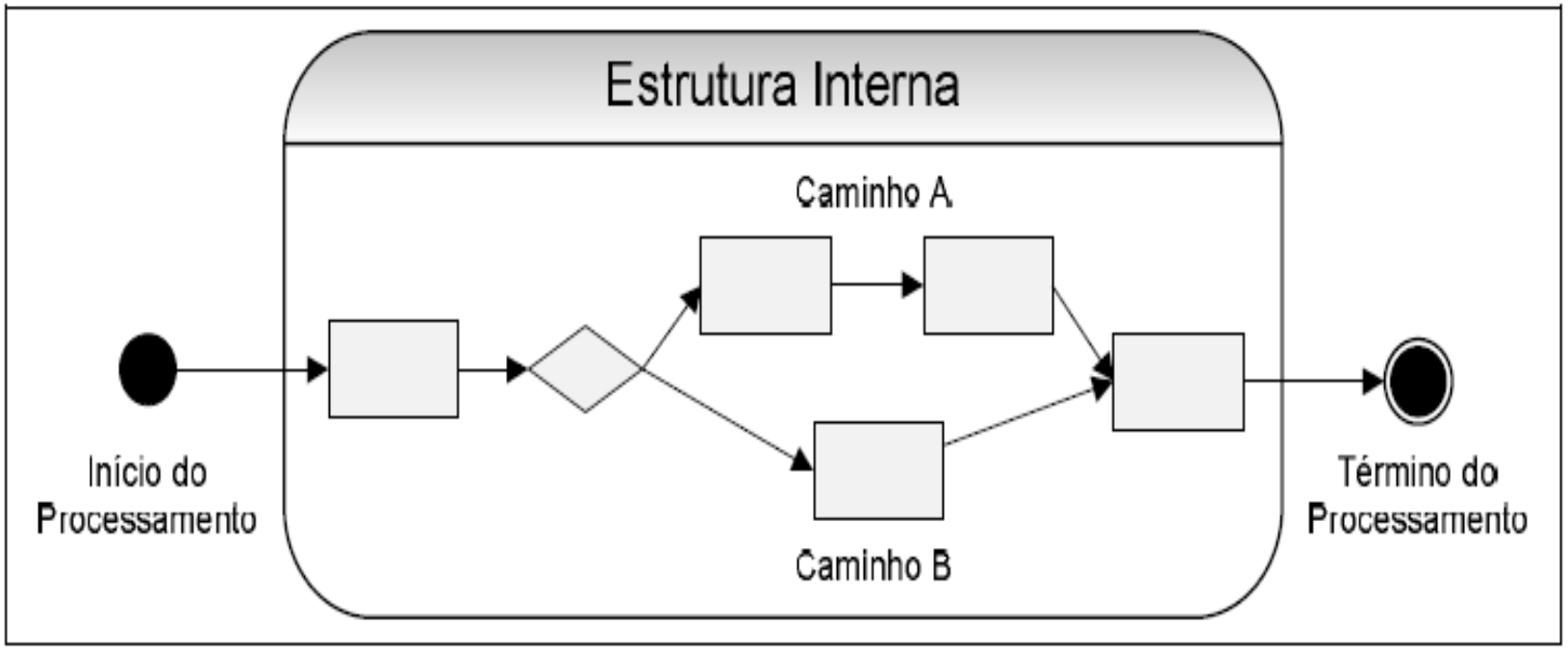
Técnicas de Teste

- As técnicas de teste de software fornecem ao desenvolvedor uma abordagem sistemática de como fazer o teste, ou seja, um conjunto de métodos para ajudar a descobrir a maior quantidade de defeitos possível.
- As técnicas de teste são classificadas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste.
- As técnicas existentes são:
  - Funcionais ou caixa preta
  - Estruturais ou caixa branca

Técnicas de Teste
Caixa Branca

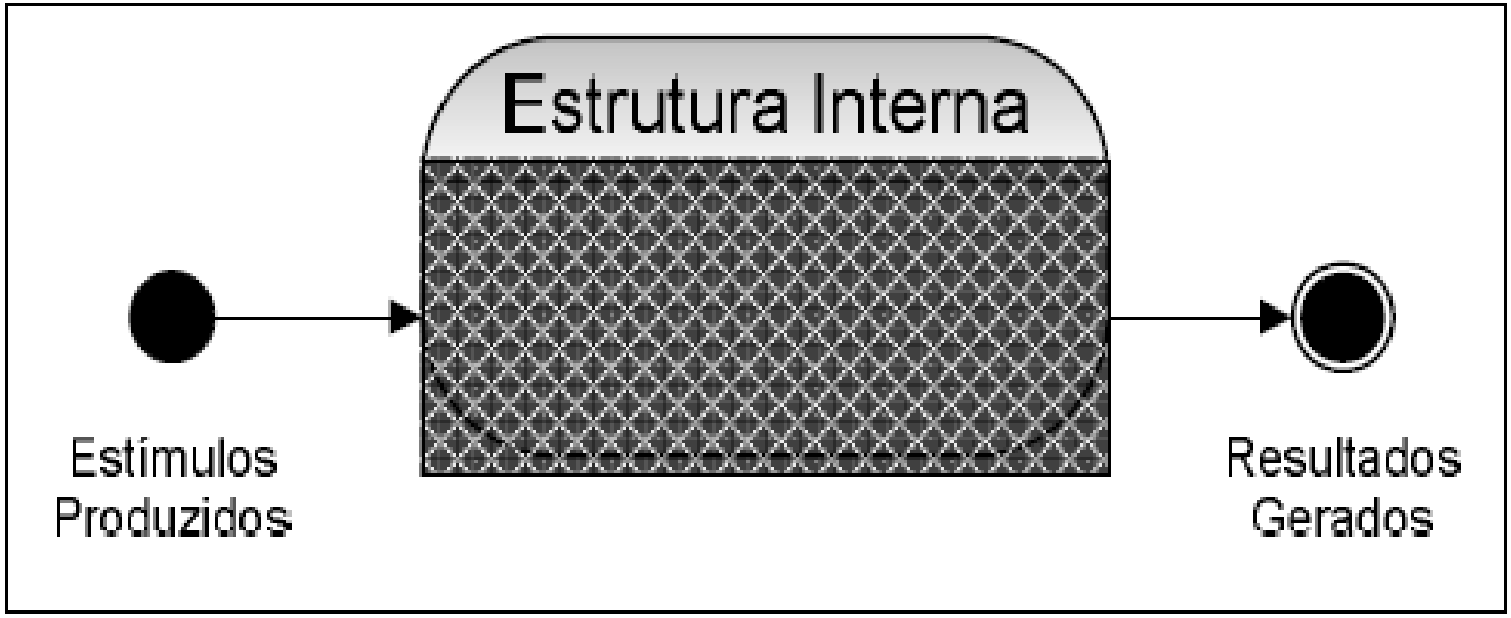
- Avalia o comportamento interno do componente de software
- Trabalha diretamente sobre o **código fonte** do componente de software para avaliar aspectos tais como: condições, fluxo de dados, ciclos ou caminhos lógicos
- O testador tem acesso ao código fonte da aplicação
- O objetivo é **testar todas as estruturas internas** do sistema, de forma a garantir que o software implementado **possui o comportamento desejado**.

Técnicas de Teste
Caixa Branca



- Garante que todos os caminhos lógicos e loops foram percorridos pelo menos uma vez
- Exercita estruturas de dados e sua validade
- Testes unitário e de integração

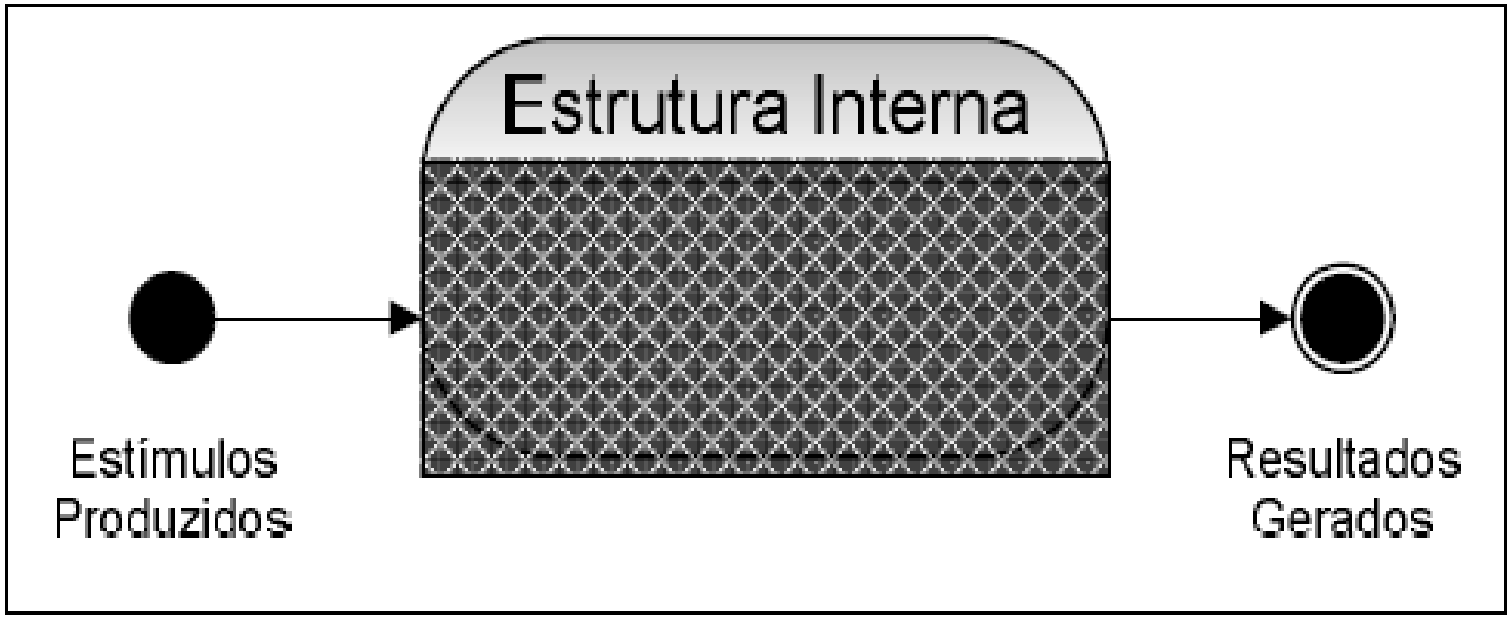
Técnica de Teste
Caixa Preta



Técnica de Teste
Caixa Preta

- Objetivo testar o software numa perspectiva externa.
- Dados de entrada são fornecidos, o teste é executado e o resultado obtido é comparado a um resultado esperado.
- Haverá sucesso no teste se o resultado obtido for igual ao resultado esperado.
- Não exige conhecimento da estrutura interna do desenvolvimento do produto

Técnica de Teste Caixa Preta



- Testa os requisitos funcionais e não funcionais, funções incorretas ou faltantes
- Identifica erros de interface, performance e acesso as estruturas de dados
- Testes de sistema e de aceitação

Equipe SQA
(Software Quality Assurance – Garantia de qualidade de software)

Equipe SQA (Software Quality Assurance – Garantia de qualidade de software)

- A equipe SQA deve identificar um conjunto de atividades que irão filtrar erros dos artefatos antes de serem passados para os clientes
- Para isso é necessário criar um Plano de Garantia de Qualidade
  - Definição de um processo SQA
  - Tarefas específicas a serem realizadas de SQA (plano de SQA)
  - Práticas efetivas
  - Controle dos artefatos testados e mudanças realizadas
  - Mecanismos de medição e relatórios de desempenho

Papéis da Equipe SQA (Software Quality Assurance - Garantia de qualidade de software)

- **Gerente de Testes:**
  - Responsável pelo planejamento e controle dos testes

- **Analista/projetista de testes:**
  - Responsável pela elaboração dos casos de testes e selecionar a técnica de teste mais adequada para cada requisito ou etapa do desenvolvimento
  - Acompanhar o andamento dos testes
  - Monitorar as métricas de testes
  - Conhece os requisitos do sistema

- **Testador:**
  - Responsável pela execução dos testes conforme o planejado visando revelar falhas no produto
  - Registrar os resultados de forma objetiva e baseado em fatos, apontando as falhas que ocorreram durante a execução
  - Documentar as solicitações de mudança
  - Tem experiência de programação
  - É perfeccionista
  - É criativo

Anúncio de vaga de emprego

Oportunidades DBServer

Analista de Testes

Elaboração de casos e cenários de testes, execução de acordo com padrão de documentação, elaboração de relatórios, entre outras atividades pertinentes a função.

Atividades:

- Mapeamento de Cenários de Teste
- Mapeamento de Requisitos x Analise Funcional, documento visão e Técnica
- Escrita de Cenários de Teste e Casos de Testes
- Execução e Reports de Casos de Teste

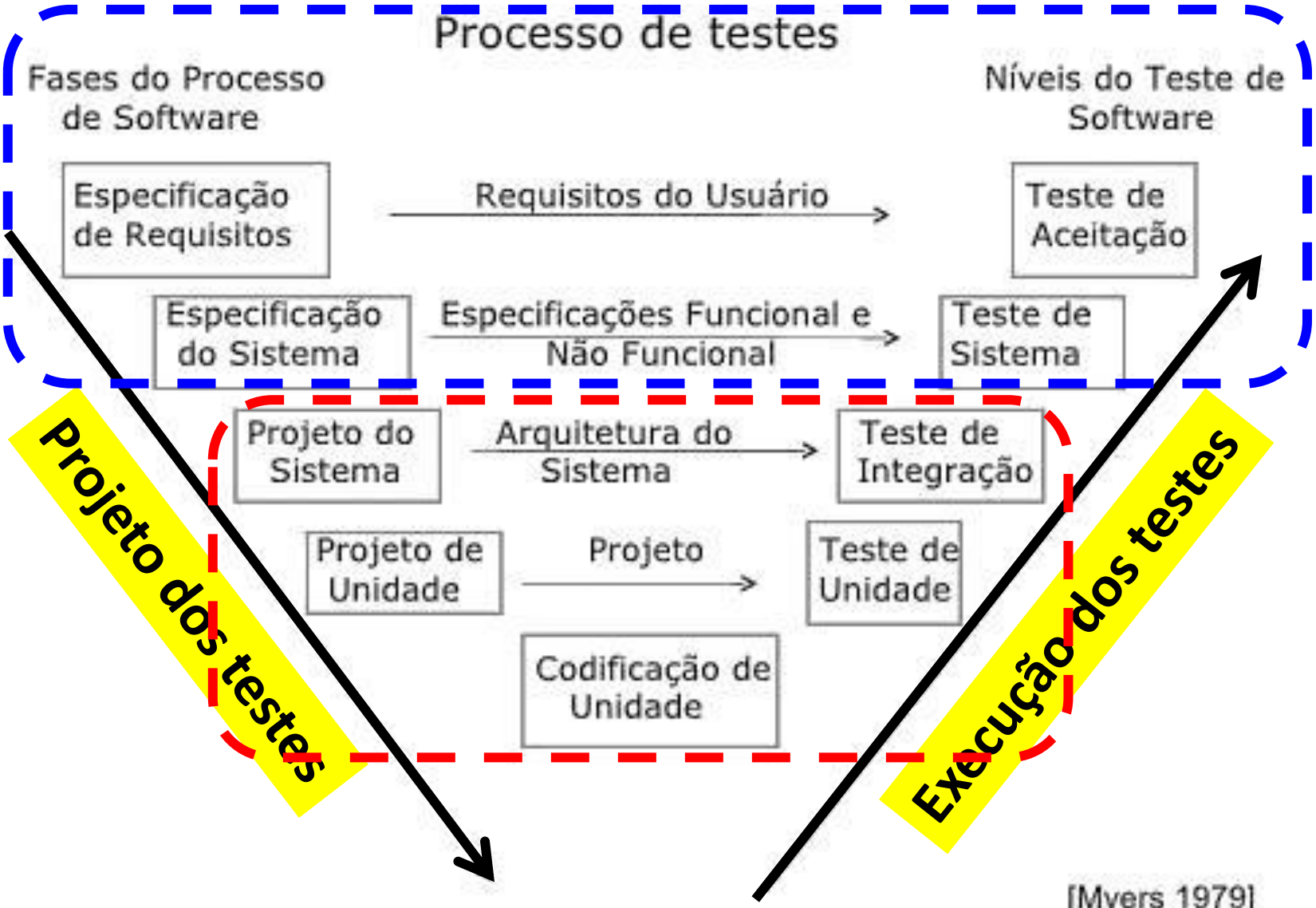
Requisitos:

- Experiência nas atividades listadas.
- Desejável experiência com automação Selenium/C#
- Conhecimentos de SOA e banco de dados (Oracle e SQL)
- Inglês intermediário/avançado para leitura, conversação e escrita, para as oportunidades de clientes internacionais.

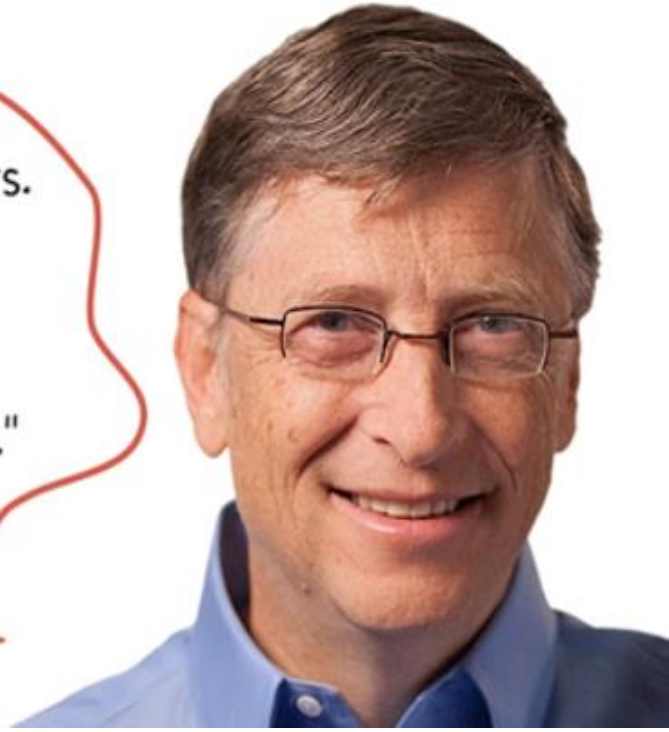
Salário a combinar + benefícios.

Processo de testes

PLANO DE TESTES



"We have as many testers as we have developers.
And testers spend all their time testing,
and developers spend half their time testing.
We're more of a testing, a quality software
organization than we're a software organization."



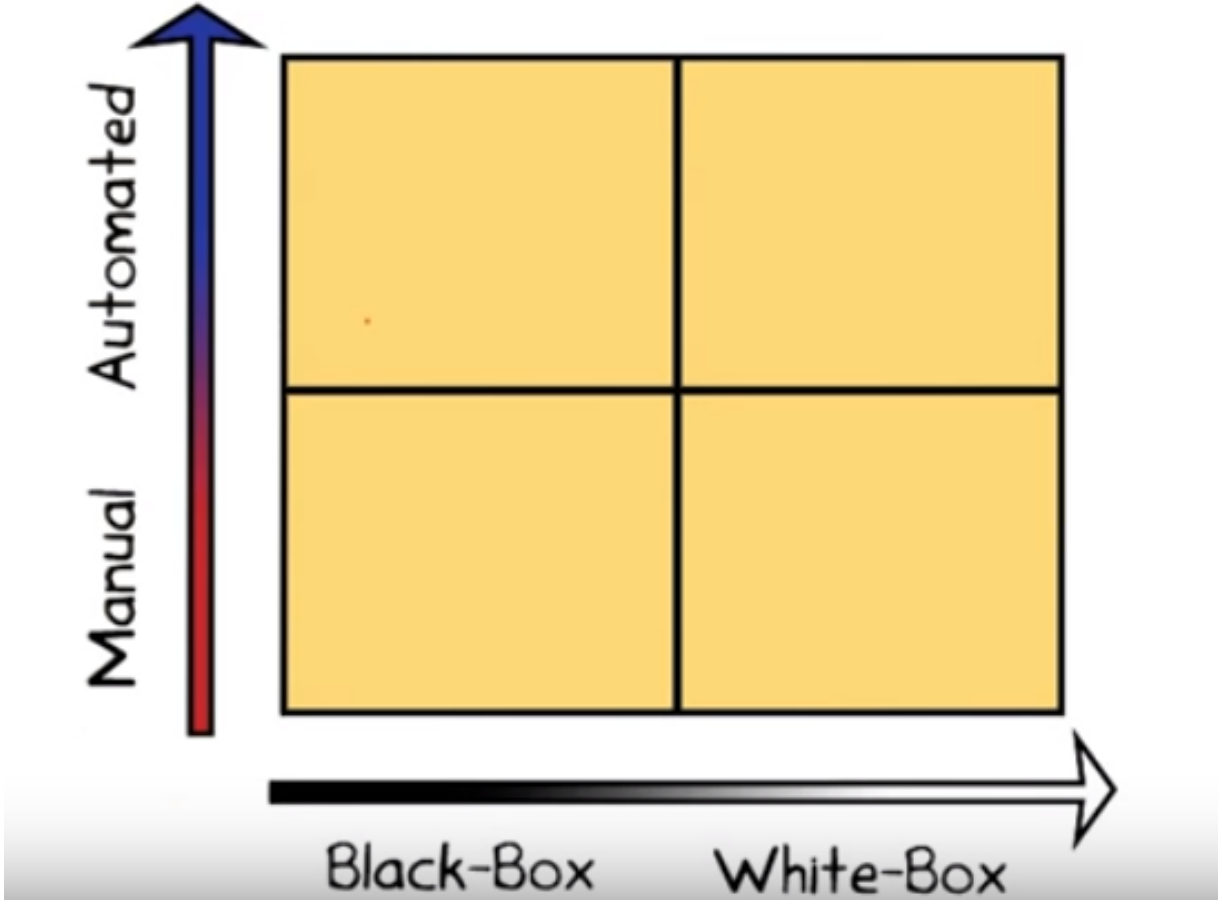
Tipos de vulnerabilidades de um software

- Segurança
- Erro de programação
  - *Ex. Foguete Ariane precisava de uma variável com tamanho maior e por isso ocorreu overflow*
  - Erros não são intencionais, porém as consequências são graves
- Explorar os erros de programas para achar falhas
  - Isso que fazem os hackers
  - Tentam achar qualquer falha que pode comprometer os sistemas
  - Podem ocorrer ataques coordenados

- Testar: O objetivo é **verificar** se a implementação do programa está de acordo com a **especificação** do programa
- **Sem especificação não há nada para testar**
- O teste pode ser visto como a **verificação de consistência** entre a **especificação** e a **implementação do programa**
- Mesmo que não exista um testador é importante o desenvolvedor escrever a especificação
  - Pré-condições
  - Pós-condições

Classificação das abordagens de testes

- Descreve o quanto o ser humano irá participar da execução dos testes



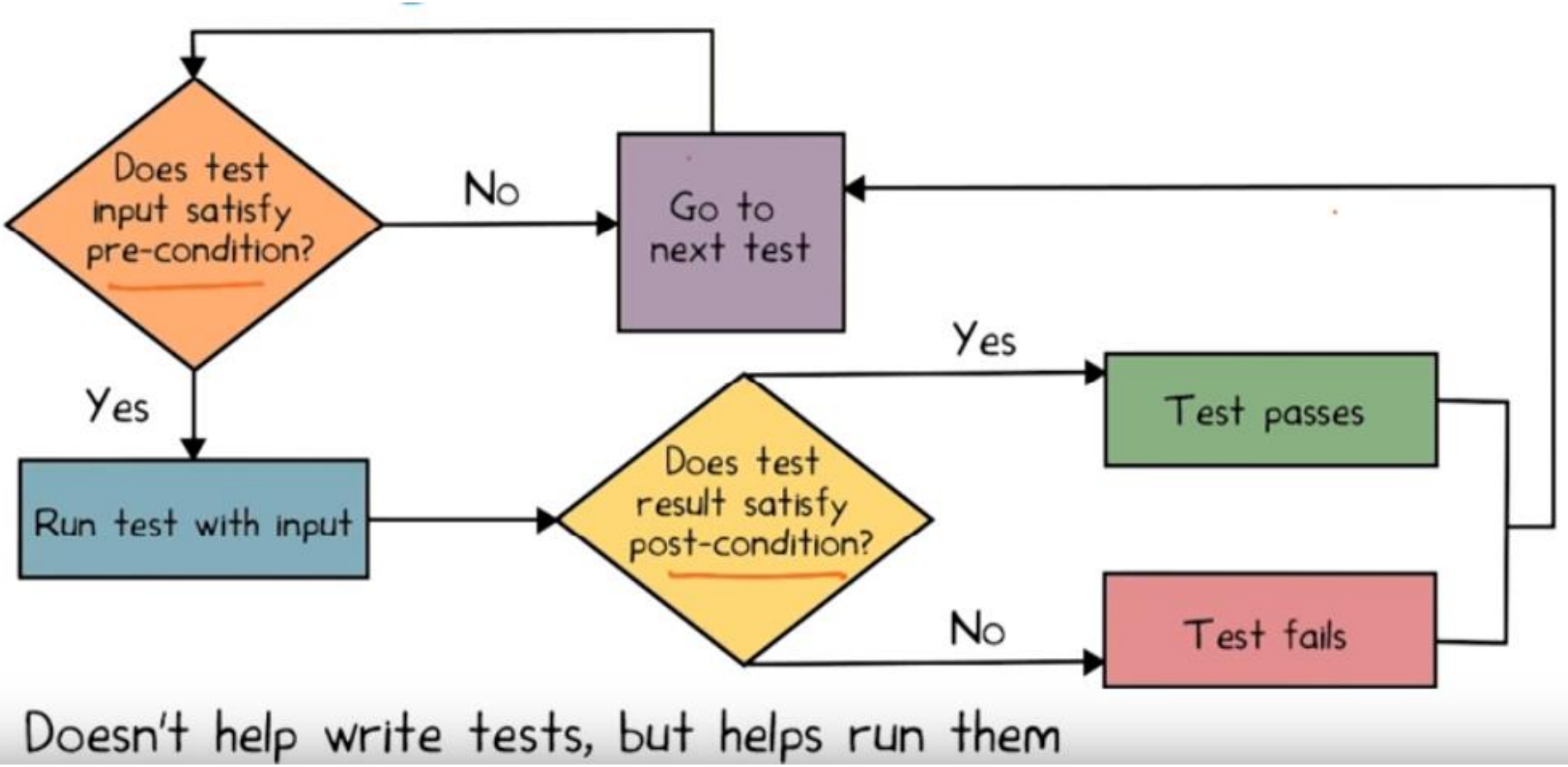
Automatizados X Manual

- Automatizados
  - Necessidade de escrever (codificar) testes que serão executados pelo próprio computador
  - Necessidade de escrever bons testes e com boa cobertura (abrangência)
  - Quanto maior o código mais caminhos lógicos (ramificações a serem testadas)
  - Encontrar erros mais rapidamente
- Manual
  - Comparação entre entradas e saídas
  - Pessoas são muito criativas e podem executar um conjunto de testes muito eficientes
- Automatizados + Manuais: boa alternativa para as empresas

Pré e pós-condições

- Pré-condição: condição assumida antes da execução de alguma função (ex. ter realizado login)
  - Objetivo: garantir que as entradas são corretas
- Pós-condição: o que se espera que a execução faça após a execução do programa que está sendo testado (ex. registrar log de acesso ou bloquear usuário)
  - Objetivo: garantir que as saídas são coerentes as especificações

Usando pré e pós-condição



- Quando as pessoas pensam em teste, normalmente a primeira coisa que vem a mente é a sua execução

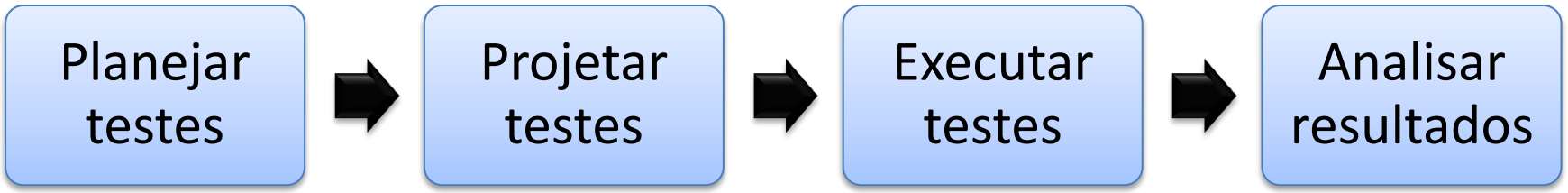
➔ A **Execução dos Teste** é **uma das etapas** do **processo** de testes!

➔Processo de Teste

- Teste é um **processo de avaliar um sistema** ou um **componente** de um sistema **para verificar se ele satisfaz os requisitos especificados** ou **identificar diferenças entre resultados** esperados e obtidos.

- **REQUER PLANEJAMENTO**

Processo básico dos testes



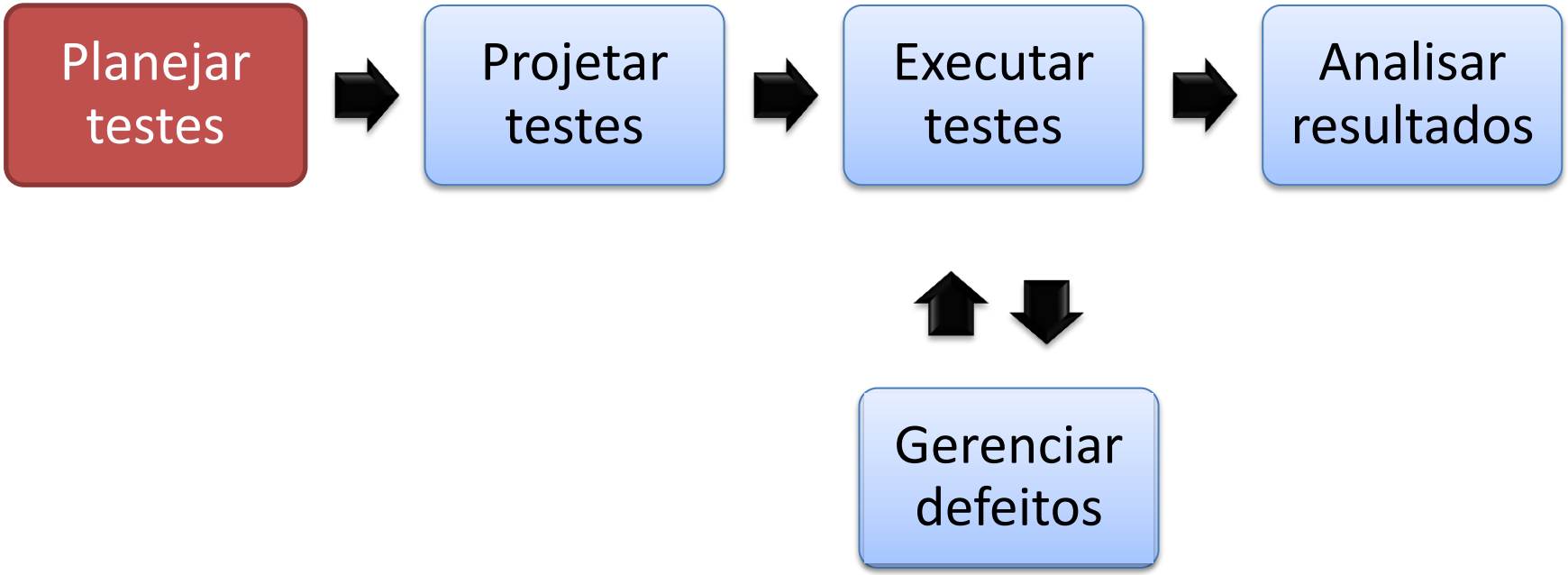
1º) **Planejamento**:
Elaboração do **plano de testes**

2º) **Especificação (projetar)**:
Elaborar casos de testes

3º) **Execução**:
Executar os testes e registrar os resultados

4º) **Avaliação dos resultados**:
Finalização

Processo básico dos testes



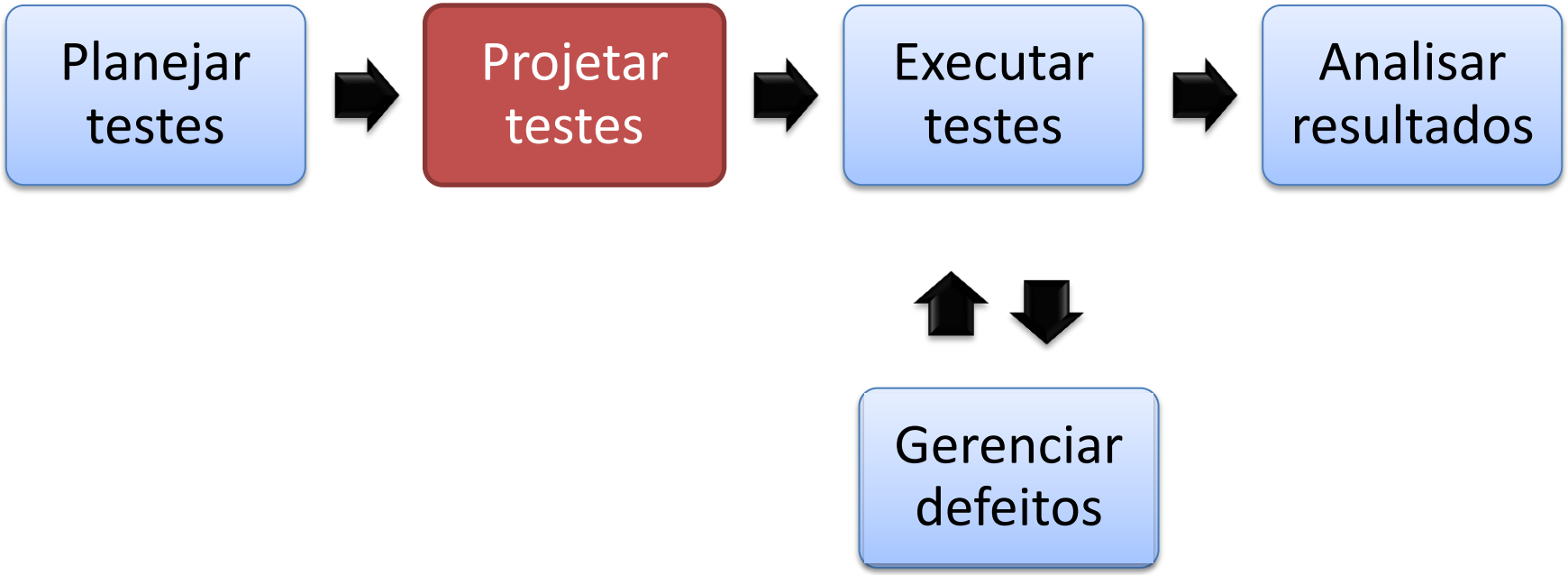
1°) Planejamento: Elaboração do plano de testes

Deve estar relacionada ao planejamento do projeto

➔ O plano de testes é um documento geral do projeto que define:

- Identificação dos itens e funcionalidades que deverão ser testados
- Identificação dos itens e funcionalidades que não serão testados
- Definição das técnicas e tipos de testes a serem utilizados
- Definição dos responsáveis (equipe de testes)
- Definição de métricas a serem coletadas
- Ambiente de testes necessário
- Outros (custos, riscos, etc)

Processo básico dos testes



2°) Especificação: Elaborar casos de testes

➔ Casos de teste:

- **É atividade chave para um teste ser bem sucedido**
- Caso de teste é um conjunto de condições usadas para testar uma condição particular de um software
  - Composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado.
- Pode ser elaborado para:
  - identificar defeitos na estrutura interna do software (verificação)
  - garantir que os requisitos do software que foi construído sejam plenamente atendidos (validação)

Como?

1º) Qual o objetivo (porque testar? o que deve ser avaliado?)

2º) Qual o resultado ou comportamento esperado?

3º) Qual ação de teste a ser executado?

2º) Especificação: Elaborar casos de testes

➔ **Estrutura dos casos de teste:**

- **Resumo/descrição**: Contém a descrição do caso de teste, descrevendo a finalidade ou o objetivo do teste e o escopo
- **Pré-condições de execução**: Condições em que o sistema deve se encontrar antes do início do teste.
- **Entradas de usuário** (dados do teste): Dados que devem ser informados para realização dos testes
- **Ações**: As ações que o analista deverá fazer para realizar o teste
- **Resultados esperados**: Resultado/comportamento esperado pelo sistema após a realização do teste

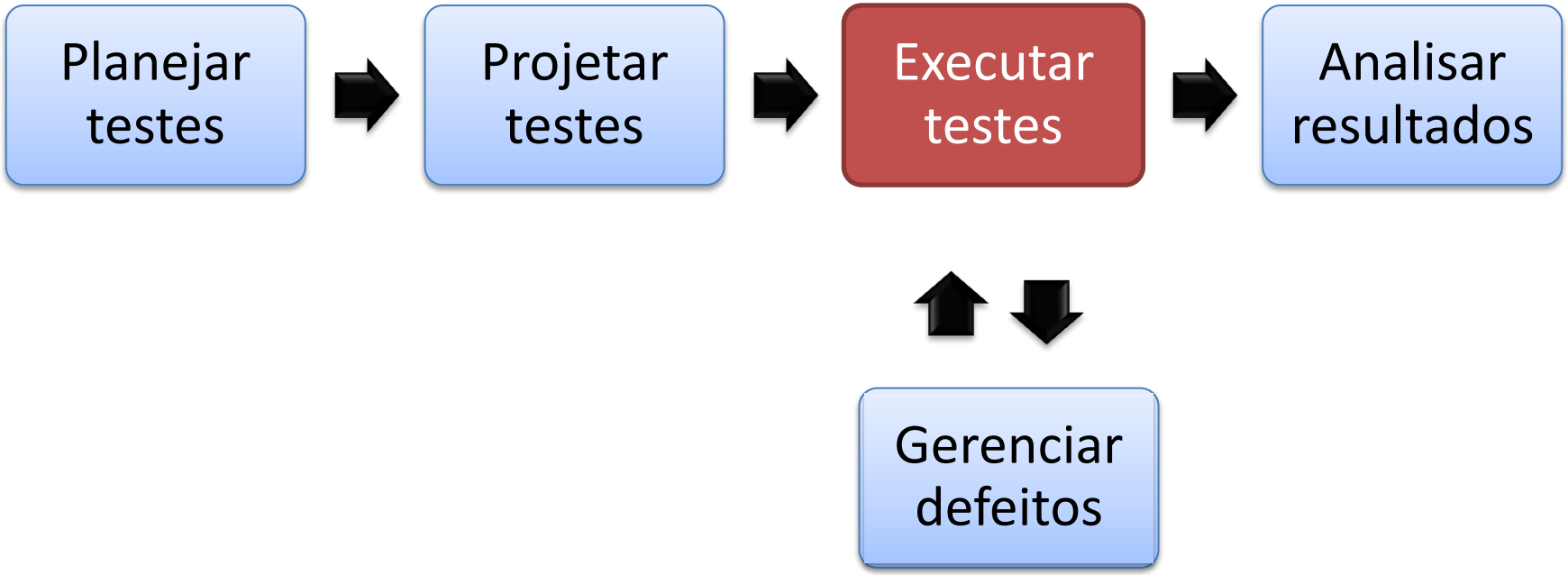
Bons casos de testes são aqueles que detectam defeito ainda não descoberto.

IMPORTANTE:

Exemplo de caso de testes

→ Arquivo disponível no moodle

Processo básico dos testes



3°) Execução dos testes:
Executar os testes e registrar os resultados

➔ Execução dos testes:

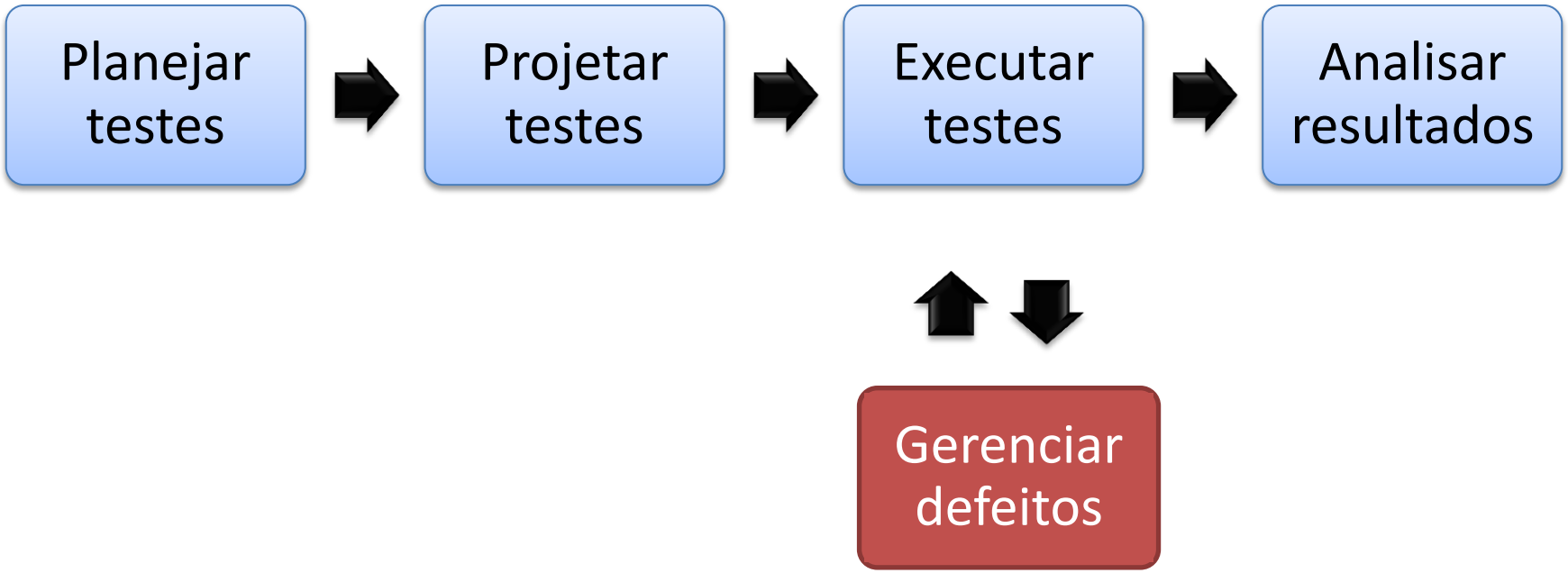
- Consiste na execução dos casos de teste (conforme o planejado)

3°) Execução dos testes:
Executar os testes e registrar os resultados

➔ Registro dos resultados:

- Fornecer um registro cronológico das ocorrências de todo o processo de execução dos testes.
- Principais informações: quem executou, quando executou, resultado obtido com a execução;
- Registrar incidentes: documentar qualquer evento ocorrido durante a execução dos testes que requeira algum tipo de investigação ou correção
- Informações dos incidentes: Descrição do incidente, data, responsável, status, severidade, prioridade e causa do defeito.

Processo básico dos testes



4º) Avaliação dos resultados: Relatório de testes e Finalização

➔Relatório de testes:

- Avaliar os resultados das atividades de teste
- Informações a serem analisadas:
  - Variações ocorridas (prazo, esforço, etc)
  - Quantidade de casos de testes testados
  - Quantidade de incidentes
  - Ocorrências não resolvidas
  - Dados coletados pelas métricas
  - Lições aprendidas