UNIP
UNIVERSIDADE PAULISTA
Interativa

Unidade III

ENGENHARIA DE *SOFTWARE* II

Prof. André Luiz

Verificação e validação – V&V

- As técnicas de verificação e validação são essenciais para o processo de qualidade no desenvolvimento de *software* e são chamadas popularmente de técnicas de V&V.
- Como já visto, garantia da qualidade é executada durante a construção do produto e o controle da qualidade após o produto ter sido construído.
- As técnicas de V&V cobrem ambos os cenários.

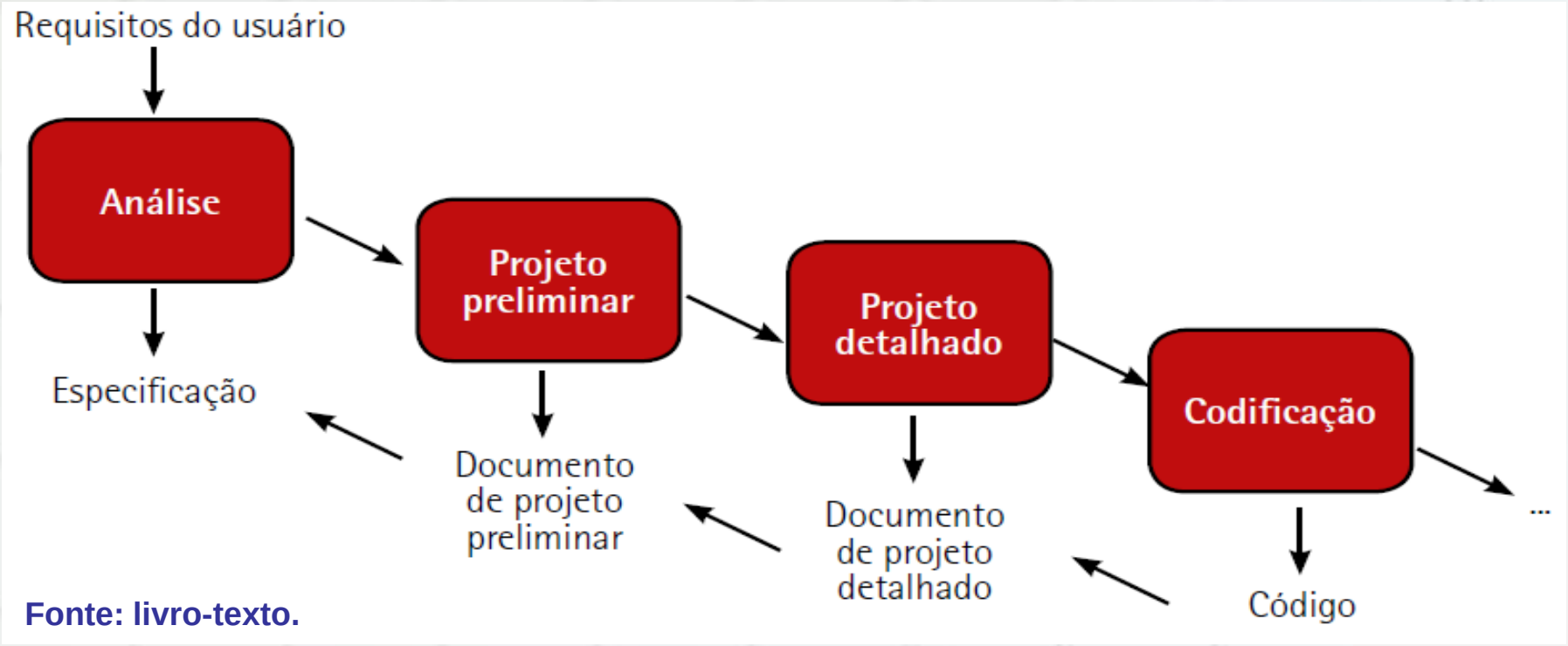
Verificação e validação – benefícios

1. Permite encontrar erros mais cedo
2. Aumentar a integração da equipe
3. Permite o acompanhamento contínuo da qualidade
4. Facilita o gerenciamento
5. Melhora a qualidade
6. Aumenta a interação entre as equipes
7. Avalia se o sistema está apto a ser usado em situação operacional

Fonte: livro-texto.

Verificação

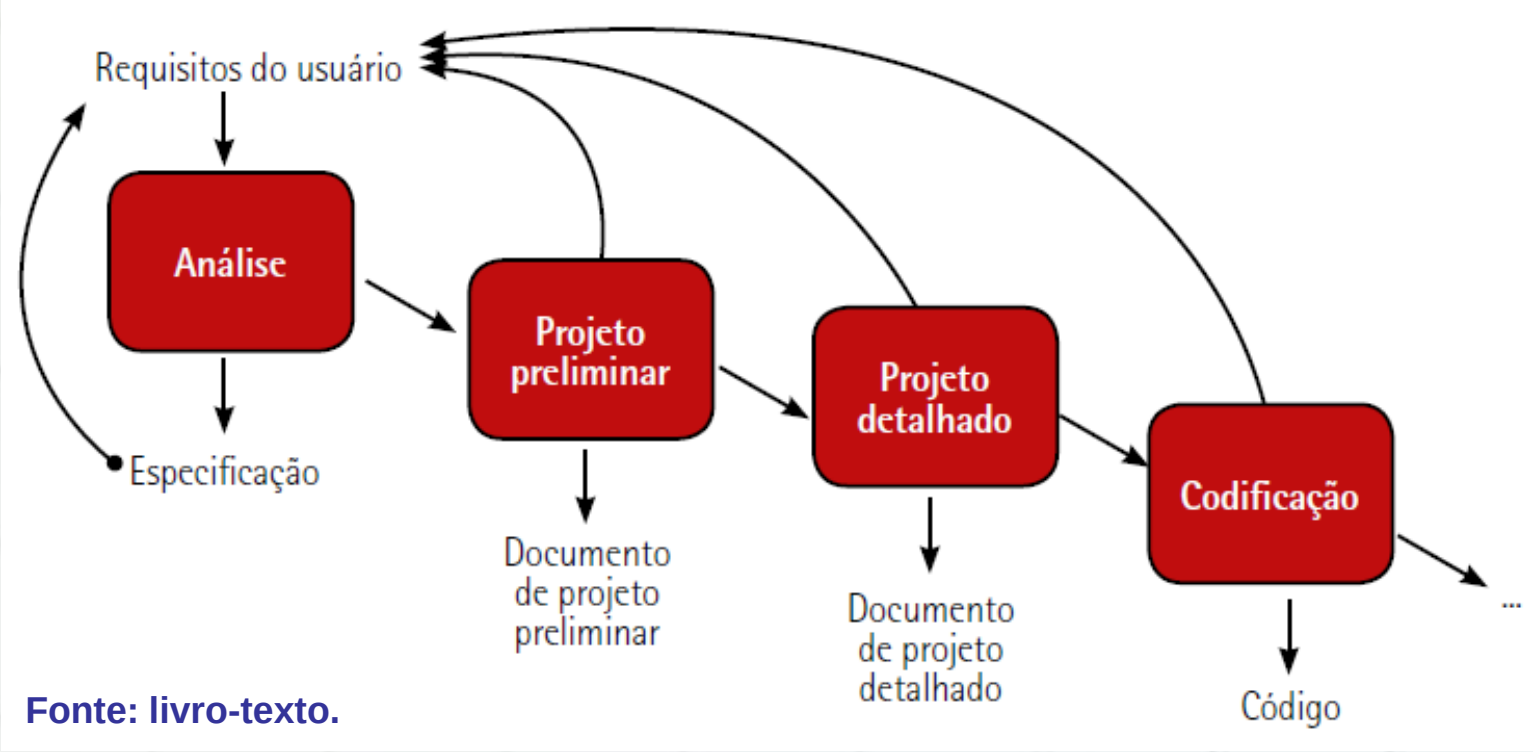
- **Consiste nas ações realizadas ao final de cada fase, interação ou artefato produzido durante o ciclo de desenvolvimento do *software*, com o objetivo de atestar que o produto está sendo desenvolvido corretamente.**



Fonte: livro-texto.

Validação

- **Consiste nas ações realizadas ao final ou durante o processo de desenvolvimento do *software*, com o objetivo de avaliar se o produto está de acordo com as especificações de requisitos iniciais fornecidas pelo cliente.**



Fonte: livro-texto.

UNIP Interativa

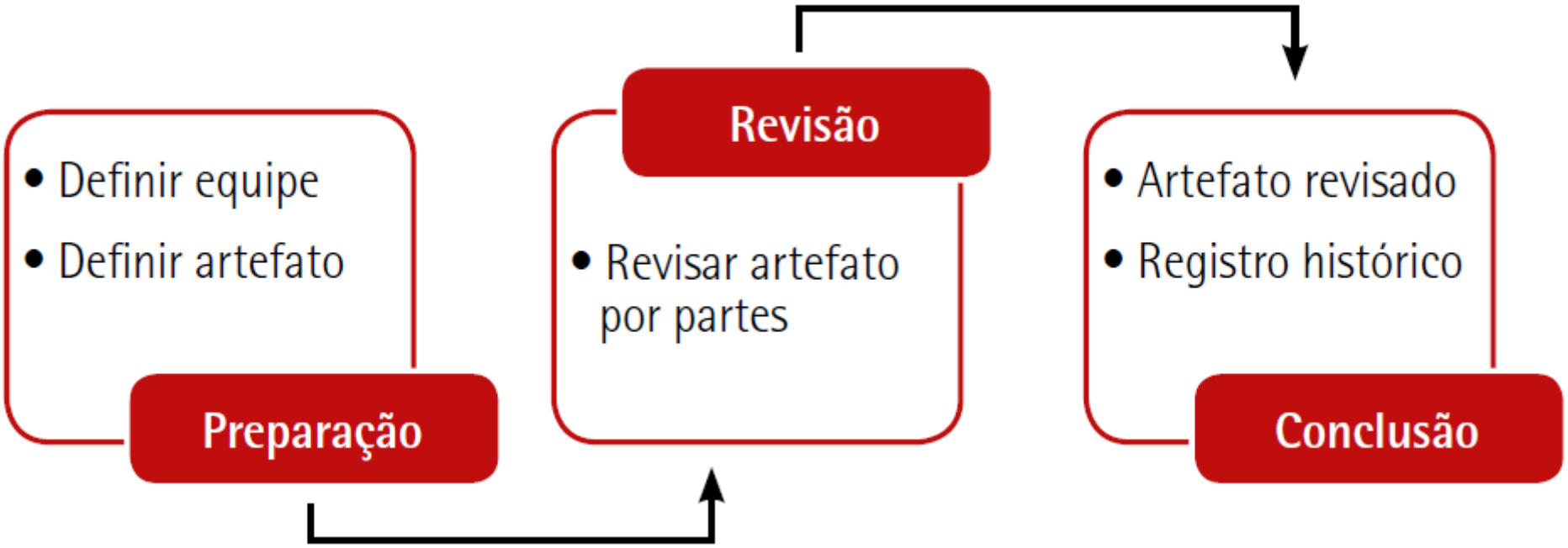
Revisão técnica

- **É uma avaliação crítica de todos os artefatos produzidos durante o desenvolvimento de um *software*, em pontos pré-definidos do ciclo de vida, com o objetivo de encontrar e corrigir eventuais erros inseridos durante o processo.**
- **O fato de encontrar esses erros mais cedo proporciona uma redução de custos no processo de desenvolvimento.**

Revisão técnica – benefícios

- Verifica se o produto de trabalho está em conformidade com os padrões, especificações e requistos definidos.
- Identifica as melhorias necessárias em um produto de trabalho.
- Efetua a documentação e a geração de histórico de erros.
- Busca o consenso entre o cliente e a equipe sobre os produtos de trabalho.
- Aumenta o conhecimento da equipe.
- É uma "via" gerencial para formalmente completar uma tarefa.

Revisão técnica



Fonte: livro-texto.

Revisão técnica – diretrizes

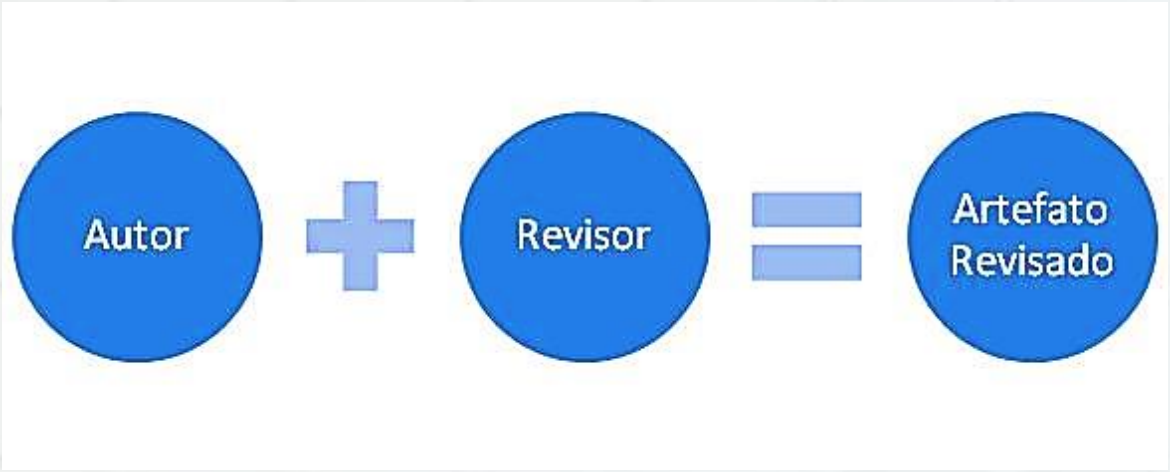
- Fixe e mantenha uma agenda com os participantes.
- Revise o produto, não o autor do artefato.
- Faça anotações por escrito.
- Enuncie os problemas, mas não tente resolver cada problema.
- Limite o debate.
- Realize um treinamento sobre revisões para todos os revisores.
- Reveja suas antigas revisões.

Walkthrough – passeios

- São revisões técnicas informais de um artefato de software, visando a garantia da qualidade.
- Normalmente são chamadas de revisão por pares, mas podem ter até três participantes: autor, revisor e moderador.
- O revisor pode ser um técnico, um cliente ou uma pessoa externa ao projeto que domine o assunto em revisão.
- O moderador preferencialmente não deve ter o mesmo nível hierárquico do autor e do revisor.

Walkthrough – passeios

- Pouca ou nenhuma preparação requerida.
- O objetivo é comunicar ou receber aprovação do artefato.
- Papéis específicos não são estabelecidos.
- O autor guia os presentes através do artefato.
- O revisor lê o documento e faz suas considerações.
- O autor nunca pode ser o leitor.



UNIP
Interativa

Walkthrough – diretrizes

- O autor seleciona os revisores e os convida para a reunião.
- O autor entrega o artefato para os revisores na reunião.
- O foco deve ser o artefato e não o autor.
- O autor deve apresentar o artefato durante a reunião.
- Os revisores apresentam possíveis falhas, comentários e sugestões de melhoria.
- Baseado nas informações apresentadas, o autor faz as devidas correções.

Interatividade

As técnicas de V&V são essenciais para a garantia da qualidade de um produto de *software*. Entre as alternativas abaixo, qual é a técnica conhecida como revisão por pares?

a) **Verificação.**

b) **Reunião técnica.**

c) **Passeios.**

d) **Revisão técnica.**

e) **Validação.**

Resposta

As técnicas de V&V são essenciais para a garantia da qualidade de um produto de *software*. Entre as alternativas abaixo, qual é a técnica conhecida como revisão por pares?

a) Verificação.

b) Reunião técnica.

c) Passeios.

d) Revisão técnica.

e) Validação.

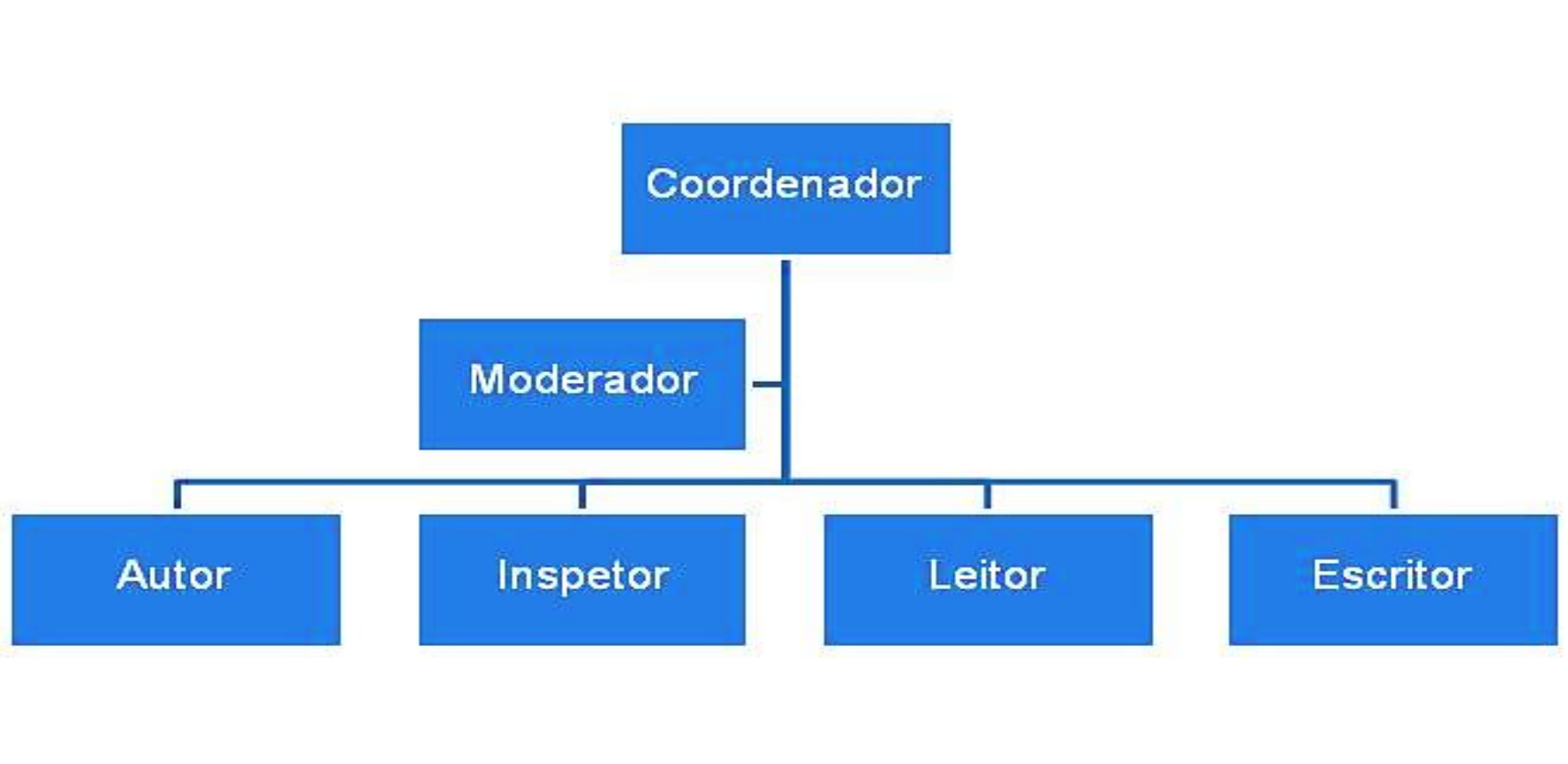
Inspeção

- **É uma técnica de verificação extremamente formal, em que os envolvidos examinam os artefatos produzidos contra uma especificação inicial, com o objetivo de encontrar incoerências, inconsistências e erros.**
- **As inspeções podem ser realizadas em qualquer momento dentro do ciclo de vida de um produto de *software*.**
- **Deve envolver pessoas com conhecimento e domínio do assunto que está sendo inspecionado e possuir uma *checklist* dos pontos que devem ser verificados no artefato.**

Inspeção – processo

Fonte: livro-texto.

Inspeção – papéis



Fonte: livro-texto.

Inspeção – *checklist*

Exemplo:

Artefato	Itens de verificação
: Especificação casos de uso	As pré-condições foram descritas? A especificação está de acordo com a tela? Os fluxos alternativos foram corretamente indicados? As regras de negócio foram apontadas? Na descrição dos casos de uso não há referências de navegação? Os atributos de entrada e saída estão descritos? As pós-condições estão explícitas?

Fonte: livro-texto.

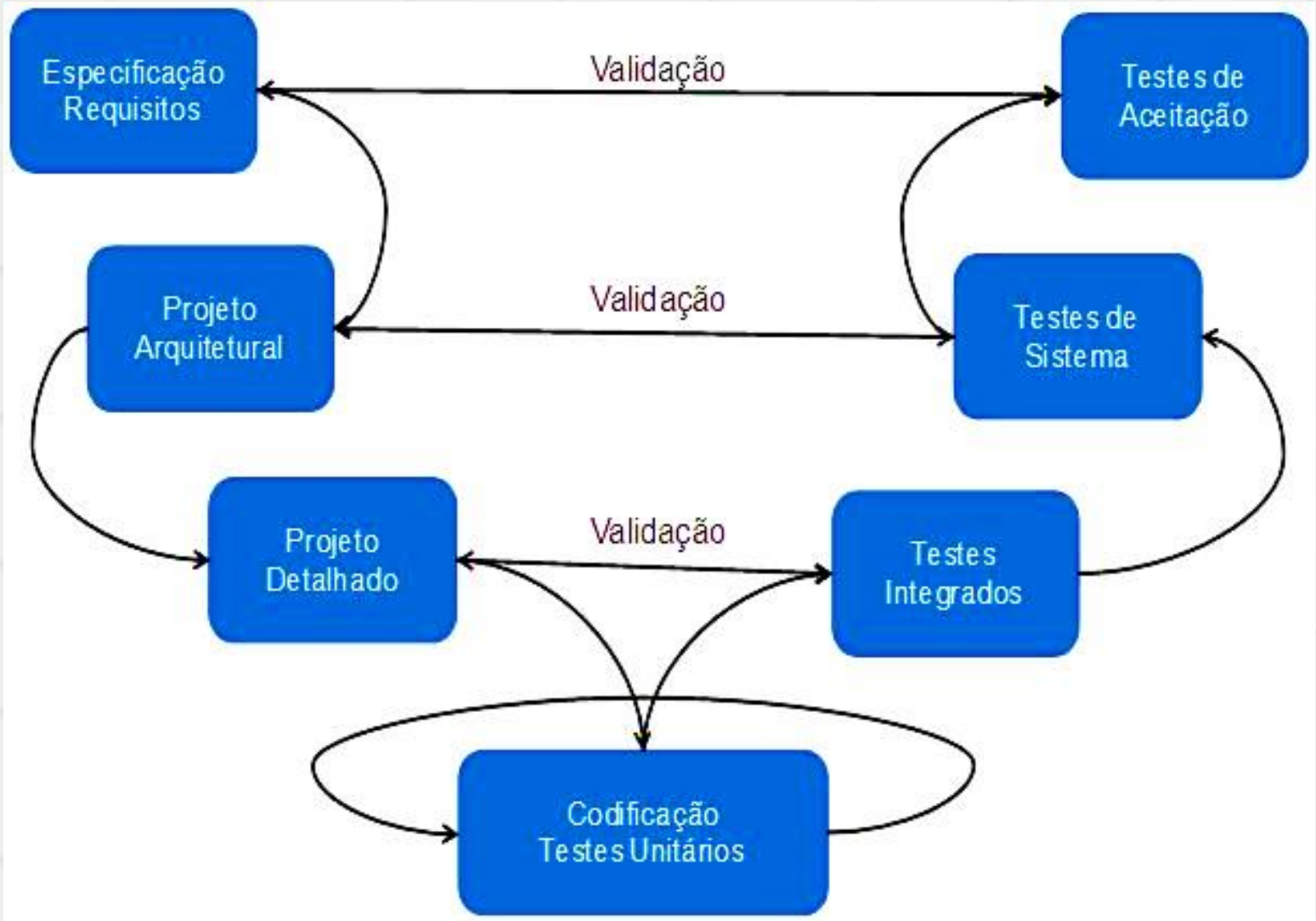
Testes de *software*

- **Segundo Myers (1979), testar um *software* é um processo de executar um programa ou sistema com a intenção de encontrar defeitos.**
- **Para Dijkstra (1985), os testes podem mostrar a presença de falhas em um *software*, mas nunca a sua ausência.**
- **Para o IEEE, testes são um processo de execução de um sistema ou programa sob condições específicas para detectar diferenças entre os resultados obtidos e os esperados.**

Por que devemos testar?

- **Fator determinante de qualidade.**
- **Para satisfazer às expectativas do cliente.**
- **Eliminar retrabalhos.**
- **Redução de custos.**
- **É do conhecimento do mercado de tecnologia da informação que as organizações chegam a gastar até 30% (trinta por cento) do esforço total do desenvolvimento de *software* realizando testes.**

Ciclo de vida de testes – modelo V



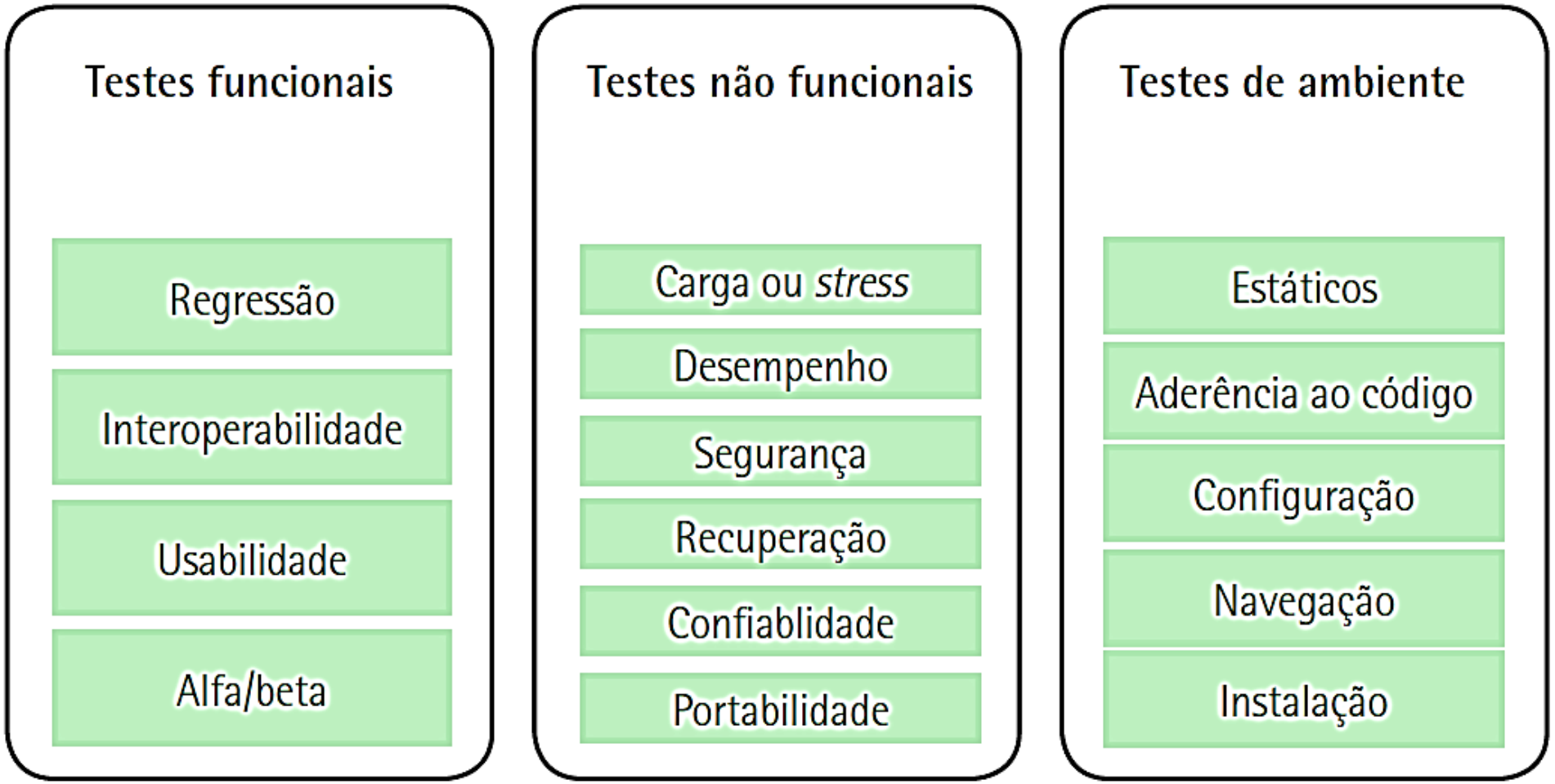
Fonte: livro-texto.

Atividades de testes

Fase de desenvolvimento	Fase de testes	Atividade	Ações de qualidade	Participantes
Especificação	Planejamento	Estratégia e preparação do ambiente de testes	Revisão	Equipe de testes
Projeto de arquitetura	Análise	Identificação dos casos de testes	Revisão e inspeção	Equipe de projeto e de testes
Projeto detalhado	Especificação	Elaboração do roteiro de testes	Inspeção	Equipe de testes e usuários
Construção	Execução	Localização de defeitos e realização da correção	Testes unitários e integrados	Equipe de projeto e de testes
Implantação	Homologação	Correção de defeitos	Testes de sistema e aceitação	Equipe de projeto, de testes e usuários

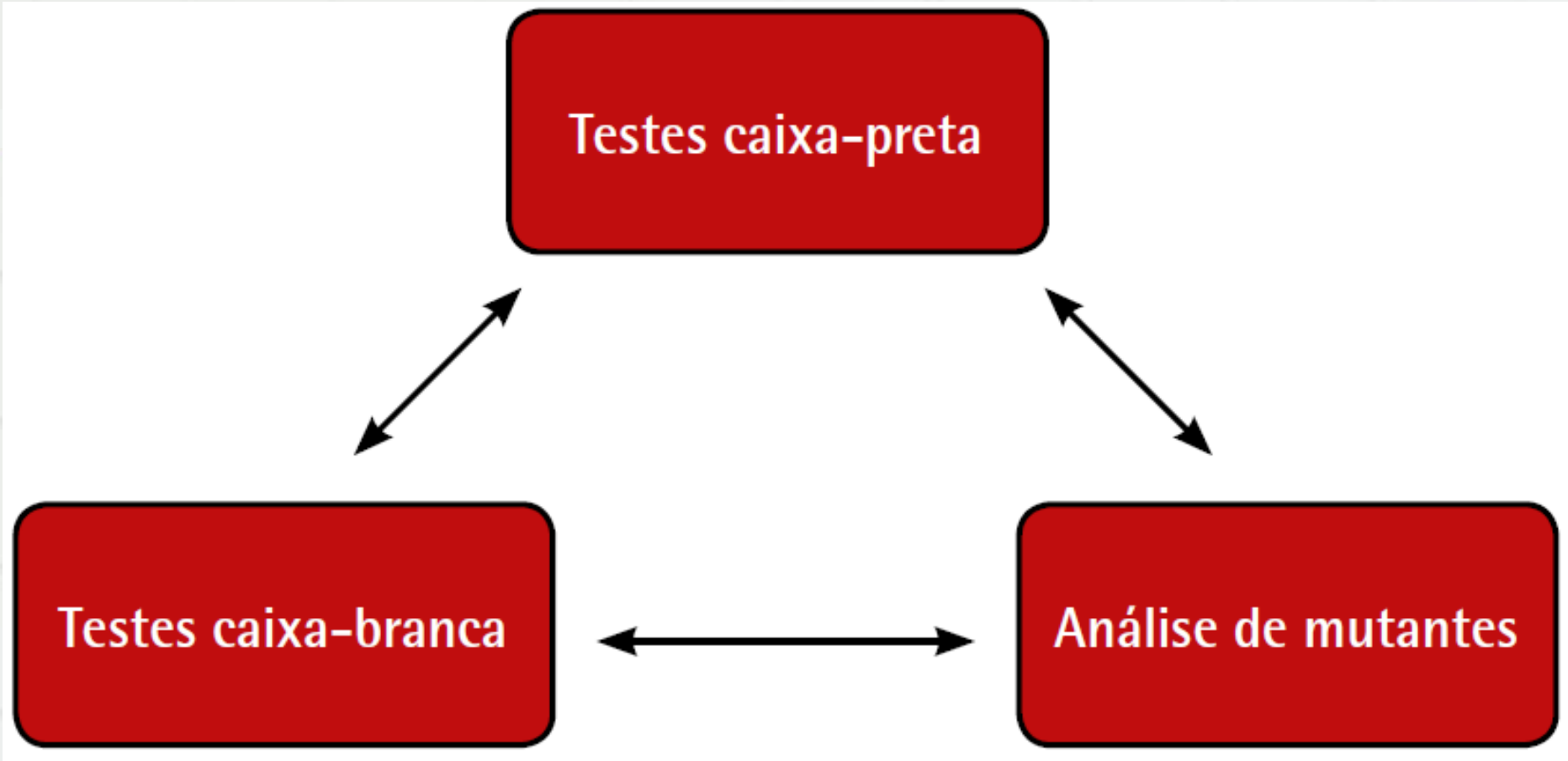
Fonte: livro-texto.

Tipos de testes



Fonte: livro-texto.

Técnicas de testes



Fonte: livro-texto.

Interatividade

O teste de *software* é parte fundamental do processo de controle de qualidade de uma aplicação. Envolve testes dos requisitos funcionais, não funcionais e de ambiente. Assinale a alternativa que contém um tipo de teste funcional.

a) **Usabilidade.**

b) **Desempenho.**

c) **Segurança.**

d) **Navegação.**

e) **Instalação.**

Resposta

O teste de *software* é parte fundamental do processo de controle de qualidade de uma aplicação. Envolve testes dos requisitos funcionais, não funcionais e de ambiente. Assinale a alternativa que contém um tipo de teste funcional.

a) Usabilidade.

b) Desempenho.

c) Segurança.

d) Navegação.

e) Instalação.

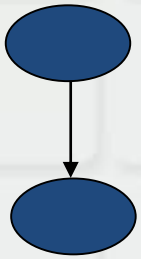
Testes caixa-branca ou estrutural

- **Testes realizados sobre o código escrito pelos desenvolvedores para garantir a qualidade do código produzido.**
- **Foca em avaliar a qualidade do código produzido pelos desenvolvedores, garantindo que toda linha de código escrita seja executada pelo menos uma vez.**
- **Para isso, são identificadas todas as condições de controle do programa (*if*, *while*, *repeat*, *cease, switch*, entre outras) e gerada a massa de testes necessária para a verificação do código-fonte.**

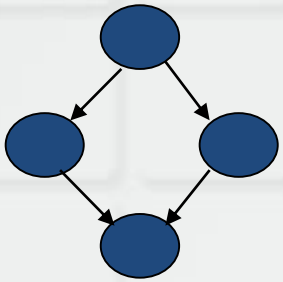
Testes caixa-branca

Representação grafos de controle:

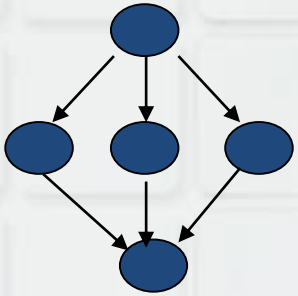
Fluxo



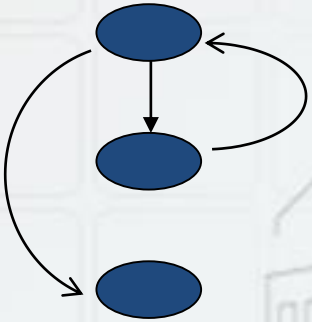
IF



Switch



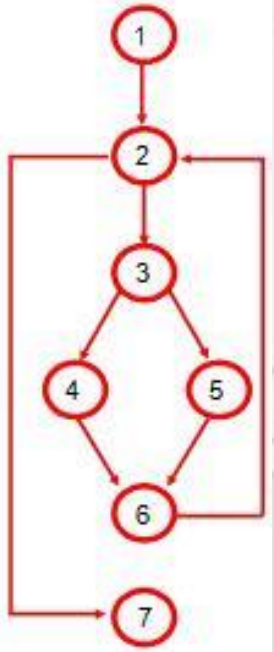
Repetição



Fonte: livro-texto.

Exemplo grafo de controle

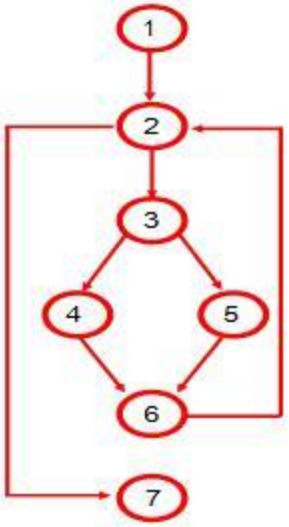
```
início
        leia nro                                    } 1
        enquanto nro ≠ 0                            } 2
            se nro > 0                              } 3
                    raiz = raiz-quadrada(nro)       } 4
                    escreva raiz
            senão
                    escreva mensagem de erro        } 5
            fim-se
            leia nro                                } 6
        fim-enquanto                                } 7
fim
```



Fonte: livro-texto.

Grafos – caminhos independentes

- Com a construção do grafo de controle é possível determinar os caminhos de testes básicos do programa, chamados de caminhos independentes.
- Caminho independente é aquele que contém pelo menos uma nova aresta no grafo de controle e garante que todo comando será executado pelo menos uma vez.



Caminhos independentes:

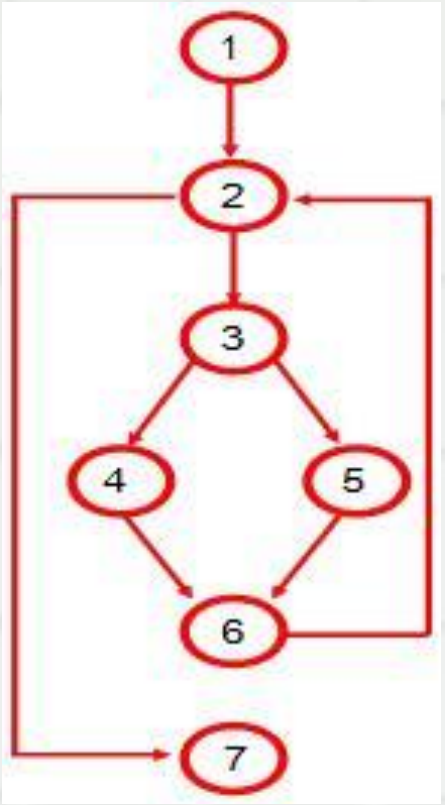
1-2-7
1-2-3-4-6-2
1-2-3-5-6-2

Complexidade ciclomática

- Baseada no grafo de controle. Tem como objetivo medir quantitativamente a complexidade lógica de um programa e fornecer o limite superior para o número de caminhos independentes, que determina a quantidade de testes necessários para garantir que todas as linhas de código sejam executadas pelo menos uma vez.

- V(G) = (E – N)+2, onde: E é o número de arestas.
  N é o número de nós do grafo.
- V(G) = P + 1, onde:
  P é o número de nós predicados do grafo.

Exemplo – complexidade ciclomática



Complexidade Ciclomática:

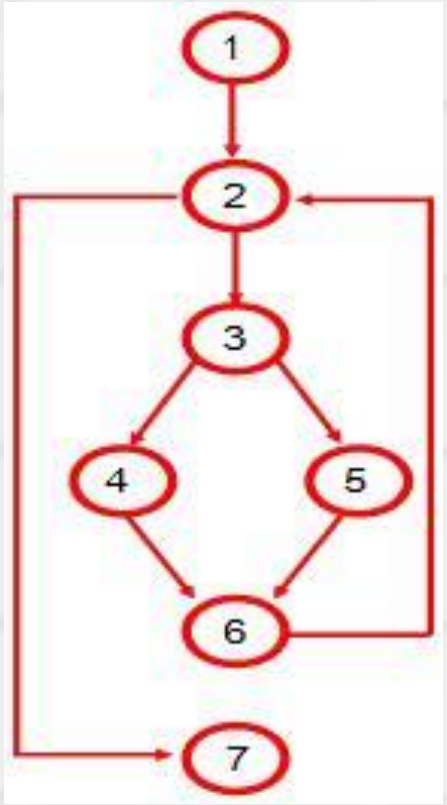
V(G) = (E - N) + 2
V(G) = (8 – 7) + 2 => **V(G) = 3**

Ou

V(G) = P + 1
V(G) = 2 + 1 => **V(G) = 3**

Fonte: livro-texto.

Exemplo de criação de testes caixa-branca



- Determinar a V(G) do grafo de fluxo correspondente:

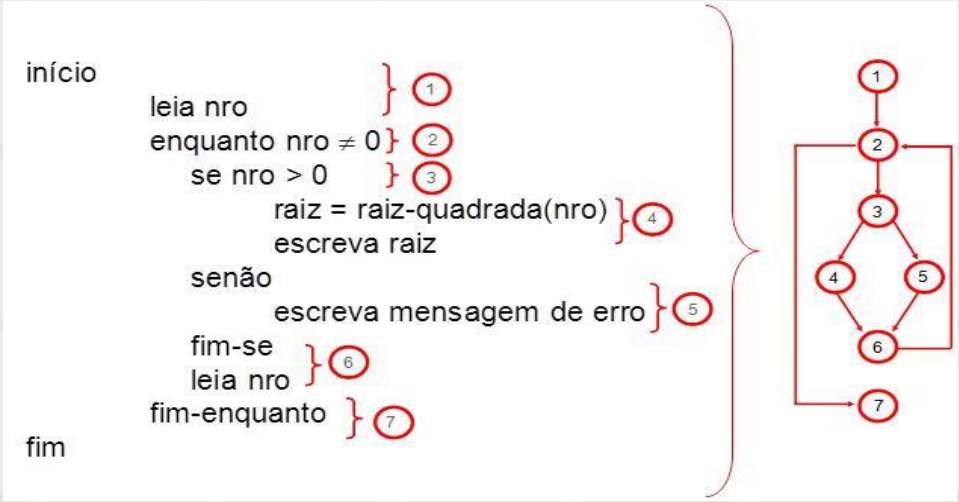
  V(G) = (E - N) + 2

  V(G) = (8 – 7) + 2 => V(G) = 3

- Determinar os caminhos independentes:
  - 1-2-7
  - 1-2-3-4-6-2
  - 1-2-3-5-6-2
- Preparar os casos de teste para executar cada caminho.

Fonte: livro-texto.

CMMI – Representação contínua



Caminho	Dado entrada	Resultado esperado
1-2-7	Zero	Sai do programa sem executar nada.
1-2-3-4-6-2	2 e 4	Calcula a raiz quadrada de 2 e depois de 4.
1-2-3-5-6-2	4 e - 20	Calcula a raiz quadrada de 4 e escreve mensagem de erro para -20.

Fonte: livro-texto.

Interatividade

Os testes caixa-branca são aqueles que o desenvolvedor deve aplicar durante a construção do programa para garantir a execução de cada linha de código produzida. Um programa que possui 7 nós predicados no grafo de controle terá V(G) igual a:

a) 7.

b) 8.

c) 9.

d) 6.

e) 10.

Resposta

Os testes caixa-branca são aqueles que o desenvolvedor deve aplicar durante a construção do programa para garantir a execução de cada linha de código produzida. Um programa que possui 7 nós predicados no grafo de controle terá V(G) igual a:

a) 7.

b) 8.

c) 9.

d) 6.

e) 10.

Testes funcionais ou caixa-preta

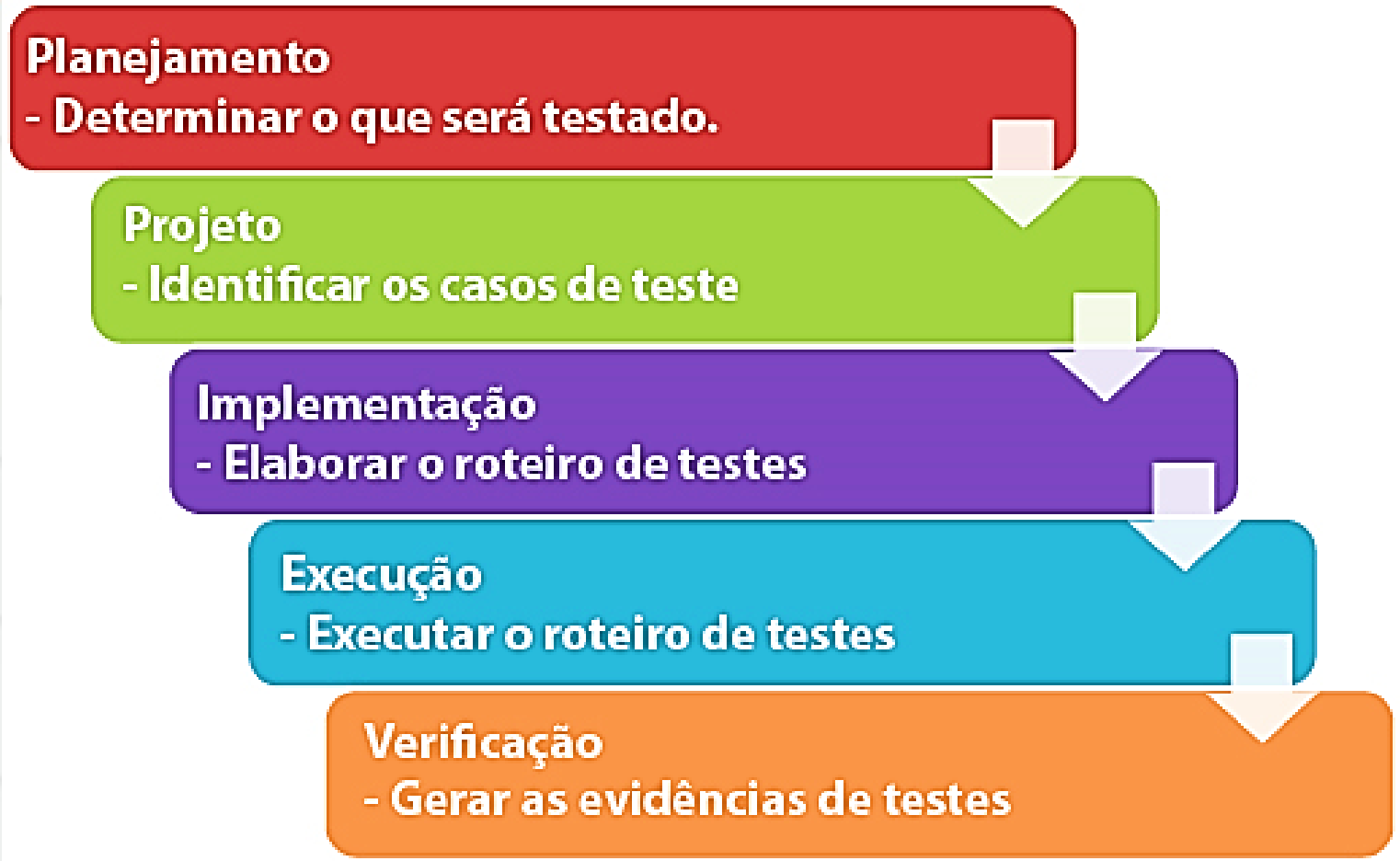
- **Os testes funcionais são os mais utilizados no processo de desenvolvimento de *software*.**
- **Foca nas necessidades ditadas pelos usuários e transformadas em requisitos pelos analistas de sistemas.**
- **As situações de testes criadas devem atestar que o *software* faz exatamente o que foi solicitado e que funciona corretamente.**

Testes caixa-preta

- Como pré-requisitos para se elaborar os testes caixa-preta, temos:
  - especificação funcional;
  - protótipo de telas.

- Na elaboração dos testes funcionais, duas atividades devem ser desenvolvidas e devidamente validadas com os usuários do sistema:
  - especificar os casos de testes ou cenários de testes;
  - elaborar o roteiro de testes.

Elaboração de testes caixa-preta



Fonte: livro-texto.

Testes caixa-preta

Caso de teste

- É uma situação que o sistema apresenta e para a qual se for dada uma informação de entrada será gerada uma saída esperada pelo usuário.
- Exemplo: incluir cliente com sucesso.

Roteiro de teste

- É uma descrição detalhada do passo a passo para a execução do sistema, para verificar cada caso de teste identificado.

Exemplo de um roteiro de testes

Caso de teste: Consultar cliente com sucesso.			
Procedimento inicial: Acessar a URL xxxxx como usuário administrador, acessar o menu cadastros, acessar a opção cliente e clicar em consultar.			
ID	**Passo para execução**	**Dado entrada**	**Resultado esperado**
1	Sistema exibe tela para pesquisa por nome ou CPF	-	Dados exibidos: campos, nome e CPF
2	Usuário informa CPF válido e clica em buscar	111.111.111-11	Sistema exibe dados do cliente: nome: xxxx, endereço: xxxxx, cidade: xxxx, estado: xx e país: xxxxx.
3	Usuário clica em nova busca	-	Tela é limpa e retorna à tela de pesquisa

Fonte: livro-texto.

Testes caixa-preta – interface

- São os casos de testes relativos ao comportamento técnico das telas ou interfaces.
- Esses casos de testes são importantes para garantir que a interface faça as verificações necessárias para tornar o *software* mais robusto e confiável com os dados de entrada.

Os principais casos de testes de interface são:

- reconhecer os atributos de cada campo;
- identificar e obter as regras de validação de cada campo;
- validar a navegação;
- validar as mensagens que serão exibidas.

Testes de interface – atividades

- Identifique os campos e componentes da interface.

Para cada campo identificado:

- coloque o nome, tipo do campo, tamanho, formato, validações e se é obrigatório ou não;
- identifique os eventos que podem ser disparados pelos componentes (*link*, botões, caixas de texto, listas, entre outros);
- valide as ações que serão realizadas para cada evento;
- defina e valide as mensagens de advertência criadas.

Testes caixa preta – interface

Exemplo:

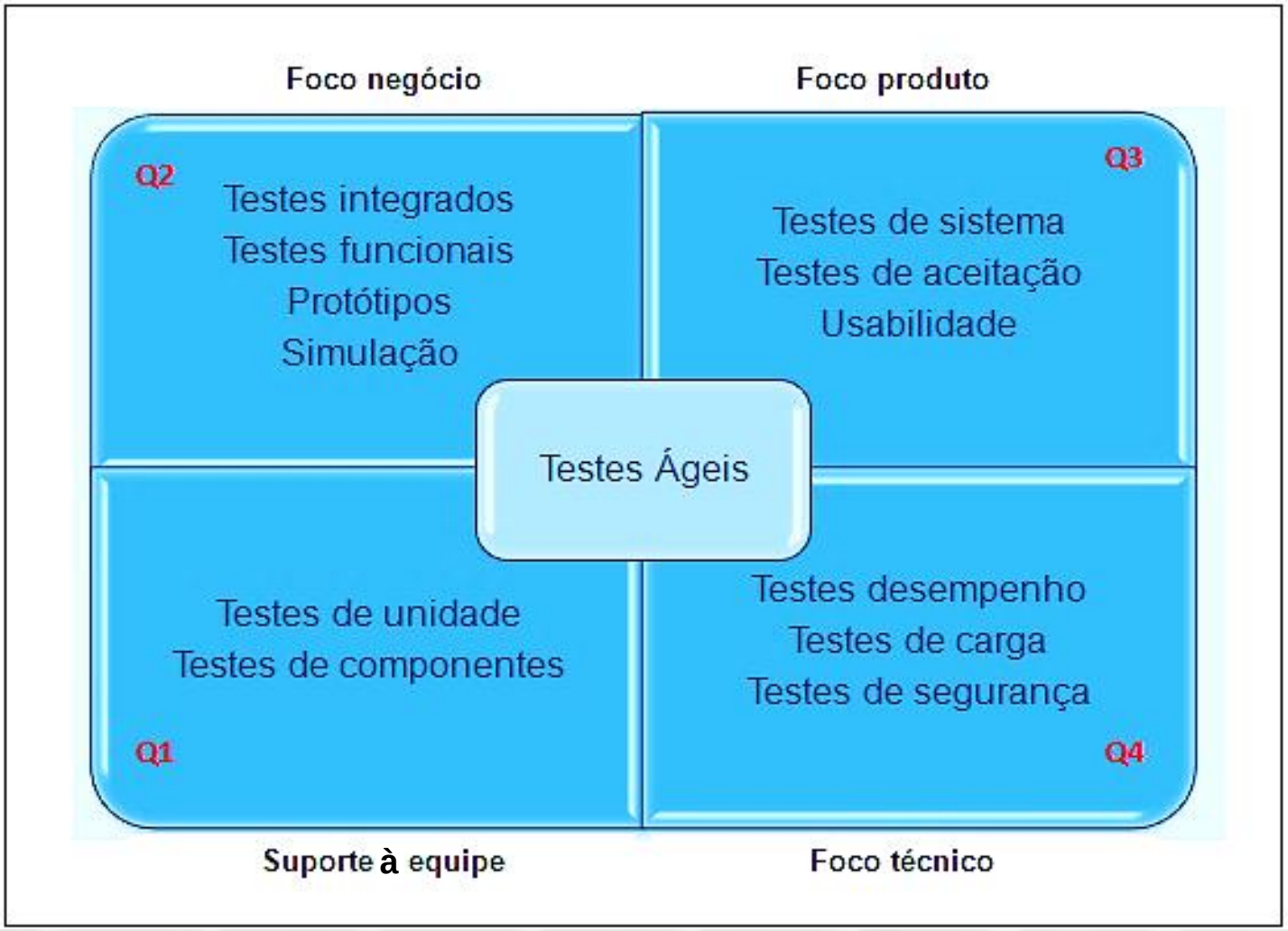
Elemento	Descrição	Tipo/Tamanho	Formato	Validação
Campo	Nome	Alfa (40)	Alinhado à esq	pode estar em branco
Campo	CPF	Numérico (9)	111.111.111-11	CPF deve ser válido
Campo	Endereço	Alfa (40)	Alinhado à esq	-
Campo	Cidade	Alfa (20)	Alinhado à esq	-
Campo	Estado	Alfa (2)	-	UFs válidas do Brasil

Fonte: livro-texto.

Testes em processos ágeis

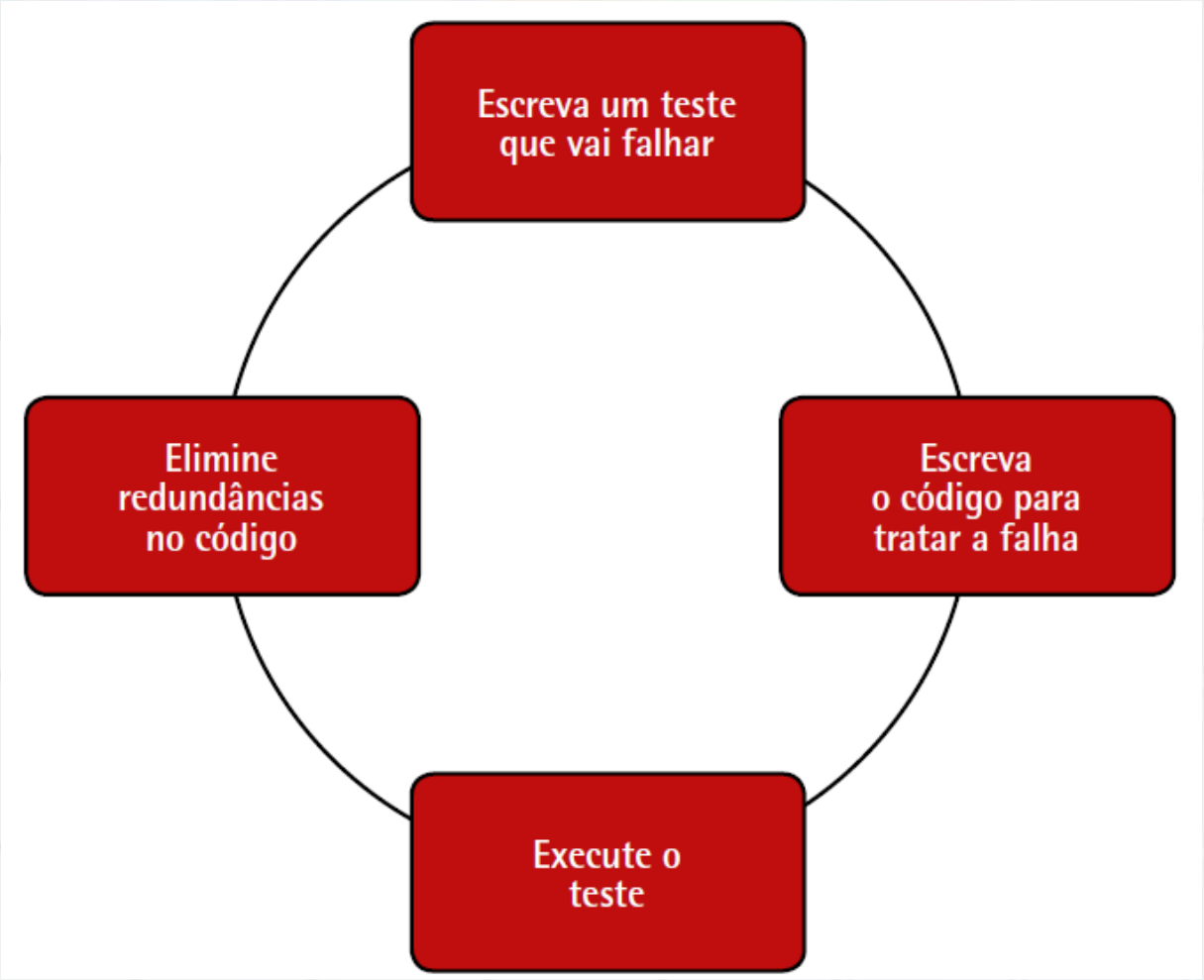
- No processo ágil, o teste é compromisso e responsabilidade de toda a equipe, todos devem possuir habilidade para testar e todos trabalham em conjunto.
- O objetivo nos testes ágeis é colaborar com a solução dos defeitos e não apenas apontá-los, identificar as causas dos defeitos para que não voltem a acontecer.
- No processo ágil não há uma fase de testes específica, os testes são realizados na medida em que a codificação termina e o *feedback* é imediato, ou seja, o defeito é apontado e corrigido na hora.

Testes em processos ágeis



Fonte: livro-texto.

TDD – *Test Driven Development*



Fonte: livro-texto.

Interatividade

Para a elaboração de testes caixa-preta existem alguns passos a serem seguidos. Nesse processo, para que possa ser elaborado um bom roteiro de testes, é necessário que tenham sido identificados os:

a) **requisitos do usuário;**

b) **protótipo de telas;**

c) **programas;**

d) **requisitos de teste;**

e) **casos de testes.**

Resposta

Para a elaboração de testes caixa-preta existem alguns passos a serem seguidos. Nesse processo, para que possa ser elaborado um bom roteiro de testes, é necessário que tenham sido identificados os:

a) requisitos do usuário;

b) protótipo de telas;

c) programas;

d) requisitos de teste;

e) casos de testes.

ATÉ A PRÓXIMA!