

TESTES DE SOFTWARE – TIPOS IV

Teste de Sistema

Uma série de diferentes testes cuja finalidade primária é exercitar totalmente o sistema.

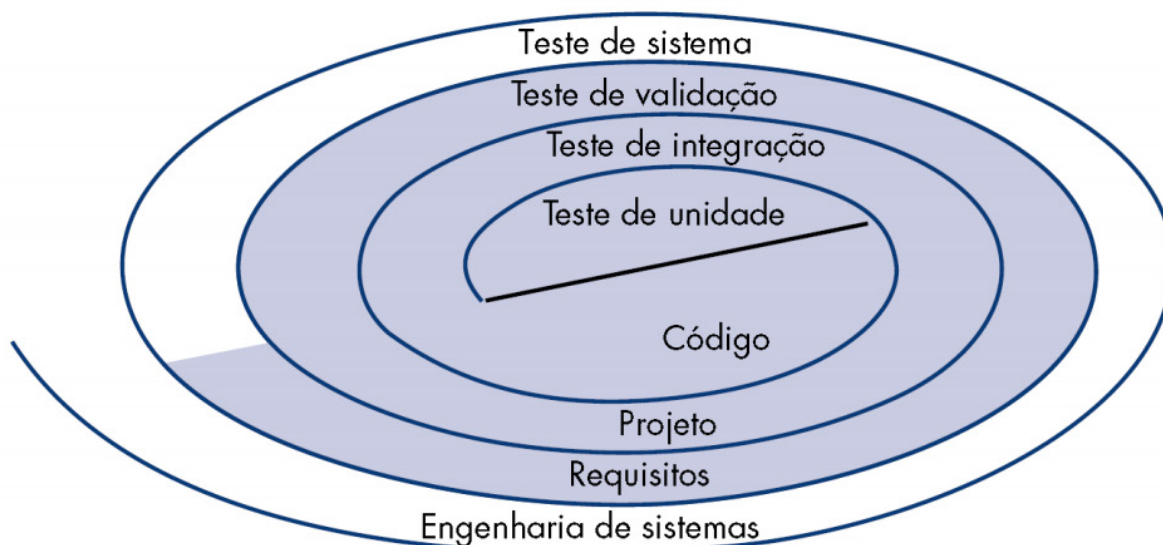
Embora cada um dos testes tenha uma finalidade diferente, todos funcionam no sentido de verificar se os elementos do sistema foram integrados adequadamente e executam as funções a eles alocadas. A tabela apresenta uma abordagem profissional, baseada no Pressman.

Tipos	Descrição
Recuperação	Força o <i>software</i> a falhar de várias formas e verifica se a recuperação é executada corretamente. Se a recuperação for automática (executada pelo próprio sistema), a reinicialização, os mecanismos de verificação, recuperação de dados e reinício são avaliados quanto à correção. Se a recuperação requer intervenção humana, o tempo médio de reparo (<i>mean-time-to-repair-MTTR</i>) é avaliado para determinar se está dentro dos limites aceitáveis.
Segurança	Tenta verificar se os mecanismos de proteção incorporados ao sistema vão de fato protegê-lo contra acesso indevido. O papel do criador do sistema é tornar o custo da invasão maior do que o valor das informações que poderiam ser obtidas.
Esforço (estresse)	Servem para colocar os programas em situações anormais. Essencialmente, o testador que executa teste por esforço pergunta: “Até onde podemos forçar o sistema até que ele falhe?”
Desempenho	Para sistemas em tempo real e embutidos, um <i>software</i> que execute a função necessária, mas não esteja em conformidade com os requisitos de desempenho é inaceitável. Isto é, frequentemente é necessário medir a utilização dos recursos (por exemplo, ciclos de processador) de forma precisa.
Disponibilização (configuração)	Examina todos os procedimentos de instalação e <i>software</i> especializado de instalação (por exemplo, os “instaladores”) que serão usados pelos clientes e toda a documentação que será usada para fornecer o software para os usuários finais.



ANOTAÇÕES

Estratégia de Testes



A estratégia de testes começa na unidade, que testa o código, passando pela integração com a intenção de testar o projeto, depois na validação para validar os requisitos e o teste de sistemas, que é a engenharia de sistema como um todo.

Esse é um modelo do Pressman.



Processo de depuração

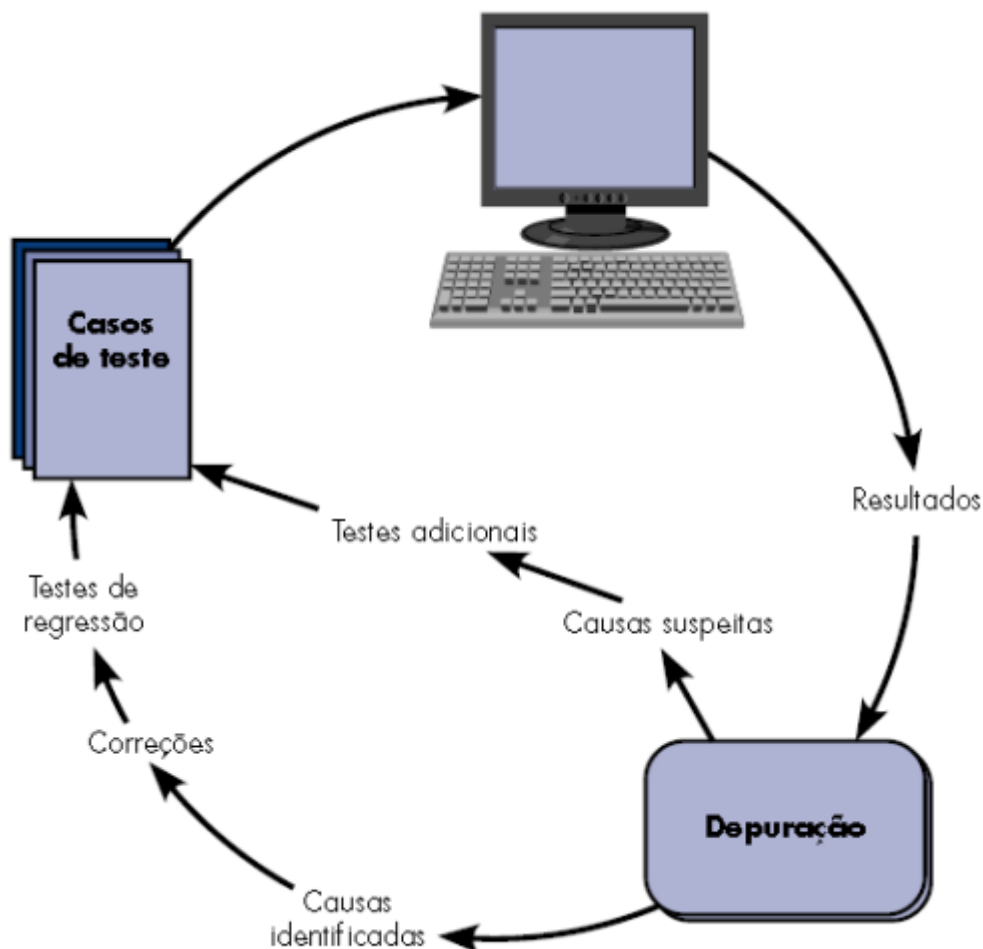
Objetivo primordial é encontrar e corrigir a causa de um erro ou defeito de *software*.

O processo de depuração é a análise do código em pontos específicos para poder identificar onde estão os erros.

Há duas táticas: Força Bruta e Rastreamento. Na primeira, o *software* fica funcionando, ocorrendo todo tipo de erro e depois o log é analisado. Já na segunda, os testes são feitos e os erros identificados, após isso, verifica a parte exata do código onde está ocorrendo o erro e é proposta uma melhoria para eliminar o defeito.

Depuração automática ocorre com a evolução dos Ambientes Integrados de Desenvolvimento (IDE) e proporcionam uma maneira de capturar alguns dos erros predeterminados específicos de linguagem.

ANOTAÇÕES



DEFINIÇÕES

Os termos teste funcional e teste estrutural são às vezes usados em lugar de teste caixa-preta e teste caixa-branca, respectivamente.

O teste funcional está relacionado com o teste caixa-preta, pois tem o interesse de lançar entradas nos casos de testes e avaliar os resultados, não há preocupação com a implementação.

O teste caixa-branca também é chamado de teste estrutural, porque no teste caixa-branca, o código está a vista e é preciso se atentar com a estrutura.

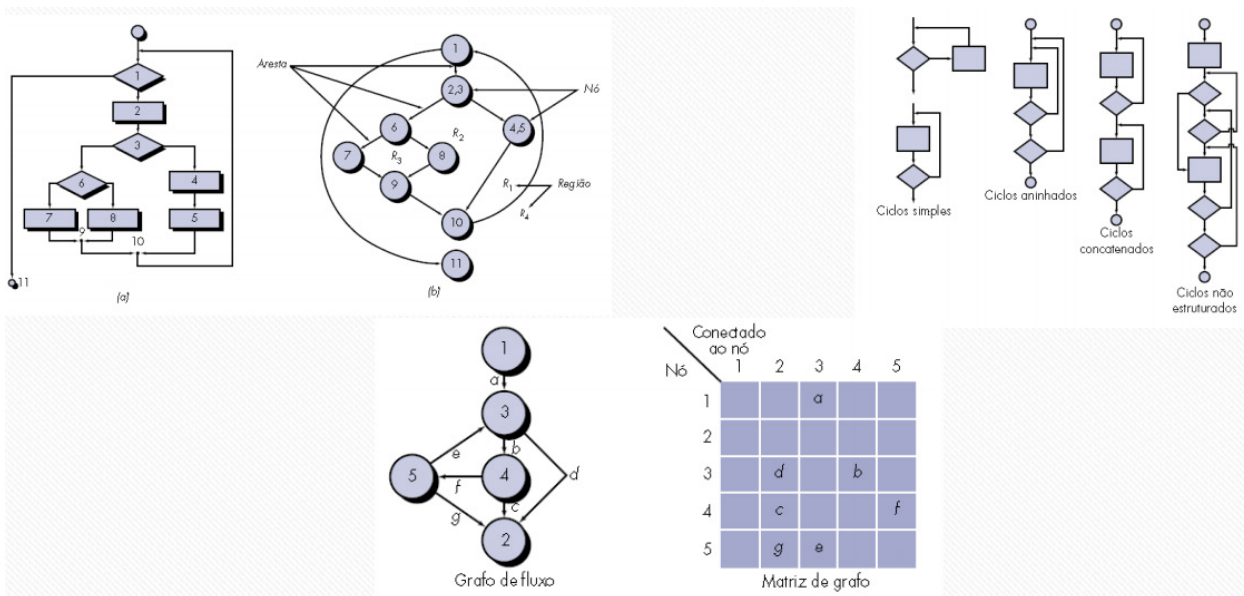


15m

ANOTAÇÕES

Teste Caixa-Branca

Teste	Descrição	
Caminho Básico	Casos de teste criados para exercitar o conjunto básico. Executam com certeza todas as instruções de um programa pelo menos uma vez durante o teste. (Grafo de Fluxo, Caminho Independente, Matriz de Grafo).	
Estrutura de Controle	1 – Condição	Método de projeto de caso de teste que exercita as condições lógicas contidas em um módulo de programa. Uma condição simples é uma variável booleana ou uma expressão relacional.
	2 – Fluxo de Dados	Uma estratégia simples de teste de fluxo de dados é requerer que toda cadeia “Definição-Uso (DU)” seja coberta pelo menos uma vez. É irreal supor que o teste de fluxo de dados será usado extensivamente ao se testar um grande sistema. No entanto, ele pode ser usado especificamente em áreas do <i>software</i> que sejam suspeitas.
	3 – Ciclo	O teste de ciclo é uma técnica de teste caixa-branca que focaliza exclusivamente a validade das construções de ciclo, quando há o <i>loop</i> .



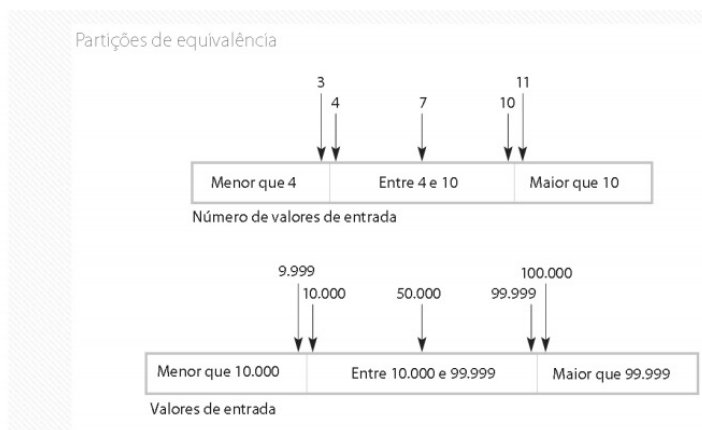
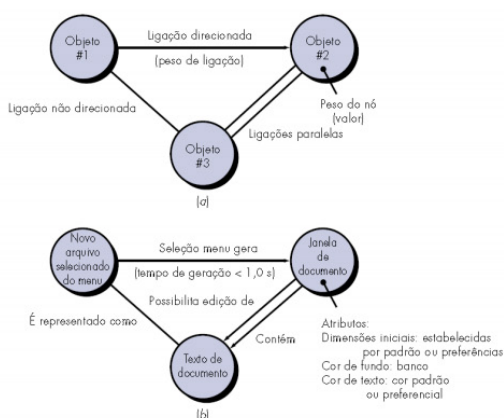
A imagem ao alto e esquerda apresenta a estratégia de fluxograma, ao lado dela, a de grafo. Nos testes de caixa-branca, percorre-se nesses fluxos. No grafo de fluxo, é analisado se o caminho está sendo atendido ou não. Existem estratégias para percorrer e fazer o teste funcionar.

ANOTAÇÕES

A imagem ao alto e direita apresente o de ciclo. Ele pode ser de ciclo simples, aninhado (onde há dois retornos), concatenado ou não estruturado.

Teste Caixa-Preta

Teste	Descrição
Com base em grafo	Começa criando um grafo de objetos importantes e suas relações e então imaginando uma série de testes que abrangerá o grafo de forma que cada objeto e relação sejam exercitados e os erros sejam descobertos.
Particionamento de equivalência	Método de teste caixa-preta que divide o domínio de entrada de um programa em classes de dados a partir das quais podem ser criados casos de teste.



A primeira imagem é o teste com base em grafo, onde há os objetos e como eles se relacionam com ligações direcionadas ou não direcionadas. Há um grafo para mostrar o caminho, que precisa ser percorrido para o teste e também o que é necessário que ocorre para verificar se o teste vai passar ou não.

A segunda imagem apresenta a partição de equivalência, onde os limites são testados. Pode ter variáveis, que estão na memória e é preciso saber se não vai ocorrer estouro dessas variáveis. Não há preocupação com a implementação, só é feito o teste passando o número de valor de entrada.

ANOTAÇÕES



DIRETO DO CONCURSO

1. (2017/CESPE/TRE-TO/TÉCNICO JUDICIÁRIO/PROGRAMAÇÃO DE SISTEMAS) Em testes de software, a divisão do domínio de entrada de um módulo em um conjunto de estados válidos ou inválidos das condições de entrada é realizada por meio de:
- a. Teste de caminho básico/lógico.
 - b. Particionamento de equivalência.
 - c. Análise de valor limite.
 - d. Teste funcional sistemático.
 - e. Teste de fluxo de dados.



COMENTÁRIO

Particionamento de Equivalências é um dos testes de caixa preta. Técnica baseada em dividir o domínio de entradas de um programa em classe de dados a partir das quais os casos de testes podem ser derivados. Procura definir um caso de teste que descubra classes de erros, assim reduzindo o número total de casos de teste que devem ser resolvidos. É um teste de caixa-preta, porque só há preocupação com as entradas e depois analisa se os limites dos particionamentos. E caso não haja problema na saída, o teste foi aprovado com sucesso.

O foco não é testar todas as entradas disponíveis e sim particionar e testar os limites.



25m

2. (2018/CESGRANRIO/LIQUIGÁS/PROFISSIONAL JÚNIOR/ANALISTA DE SISTEMAS) Um grupo de desenvolvedores elaborou vários casos de teste que selecionam caminhos de teste de acordo com as definições e com o uso de variáveis existentes em um programa. Esse tipo de teste caixa-branca é denominado:
- a. Teste de condição.
 - b. Teste de fluxo de dados.
 - c. Teste de caminho básico.
 - d. Análise de valor-limite.

ANOTAÇÕES

e. Particionamento de equivalência.

COMENTÁRIO

No Testes de Fluxo de Dados é selecionado caminhos de teste de acordo com a localização das definições e dos usos das variáveis no programa. Então eles não visam analisar os aspectos estruturais da aplicação nos trechos de código onde os dados passam, com foco nas estruturas de controle.

Continua:

- Outros Tópicos Relevantes

Referências

PRESSMAN, Roger S.; Bruce R. Maxim. Engenharia de Software, Uma Abordagem Profissional, 8º ed. Porto Alegre: AMGH, 2016. ISBN 978-85-8055- 533-2.

SOMMERVILLE, Ian. Engenharia de Software, 9. ed. São Paulo: Pearson Prentice Hall, 2011. ISBN 978- 85-7936-108-1.

GABARITO

1. b
2. b

Este material foi elaborado pela equipe pedagógica do Gran Cursos Online, de acordo com a aula preparada e ministrada pelo professor Washington Henrique Carvalho Almeida.

A presente gravação tem como objetivo auxiliar no acompanhamento e na revisão do conteúdo ministrado na videoaula. Não recomendamos a substituição do estudo em vídeo pela leitura exclusiva deste material.

ANOTAÇÕES
