

Teste de Software: conceitos, técnicas e benefícios

MSc. Alessandro Viola Pizzoleto

IFSP - Instituto Federal de São Paulo

Piracicaba - SP

30 de março de 2017



Sobre Mim

- Mestre em Engenharia de Software (UNESP - 2013);
- Doutorando em Engenharia de Software (Testes de Mutação) (UFSCar);
- Atuei 16 anos na iniciativa privada:
 - Desenvolvedor de Sistemas;
 - Analista de Sistemas;
 - Analista de Teste;
 - Gerente de Projetos;
 - DBA;
 - Gerente de Implementação do Modelo de Qualidade MR-MPS.Br.

Contato: alessandro.pizzoleto@dc.ufscar.br



1 Introdução

- Dados Históricos
- O Antes e o Agora do Teste de Software

2 Conceitos

- O que é Teste de Software?
- Validação, Verificação e Teste de Software
- Fases da Atividade de Teste

3 Técnicas e Critérios de Teste

- Teste Funcional
- Teste Estrutural
- Teste Baseado em Defeitos

4 Execução dos Testes

5 Benefícios

6 Carreira Profissional

7 Perguntas



Sumário

- 1 **Introdução**
 - Dados Históricos
 - O Antes e o Agora do Teste de Software
- 2 **Conceitos**
 - O que é Teste de Software?
 - Validação, Verificação e Teste de Software
 - Fases da Atividade de Teste
- 3 **Técnicas e Critérios de Teste**
 - Teste Funcional
 - Teste Estrutural
 - Teste Baseado em Defeitos
- 4 **Execução dos Testes**
- 5 **Benefícios**
- 6 **Carreira Profissional**
- 7 **Perguntas**



Dados Históricos

Evolução

● 1878

- Primeiro uso do termo “bug” por Thomas Edison, quando teve problemas de leitura em seu fonógrafo devido a um inseto
- Thomas Edison passa a chamar todos os defeitos industriais de “bug”

● 1947

- Encontrado primeiro “bug” em computadores. Harvard Mark I
- Também provocado por um inseto que causava erros nos cálculos
- Primeiro registro de “bug” encontrado em computadores

● 1979

- Publicação do livro “Art of software testing” de Glenford Myers
- Trouxe aos leitores conhecimentos avançados sobre testes



Dados Históricos

Evolução

- **1980**
 - Criação da primeira ferramenta de Testes Funcionais
- **1990**
 - Criação de suítes únicas de ferramentas de Testes Funcionais
 - Usadas também para o gerenciamento dos Testes
- **1995**
 - Criação das primeiras ferramentas para Testes Automatizados de performance
- **2002**
 - Surge a “ALATS” (Associação Latino Americana de Testes de Software)



Dados Históricos

Evolução

- **2006**
 - Primeiras certificações em Testes de Software
- **2007**
 - Primeiro Seminário Brasileiro em Testes de Software
- **Atualmente**
 - Grande quantidade de certificações na área de Testes de Software
 - Vários meios divulgando informações sobre Testes de Software (Blogs, Sites, Lista de Discussões, etc)



Dados Históricos

Antes

Antigamente

- A tarefa de executar testes em software era considerada secundária
- Os testes eram feitos por desenvolvedores
- Os usuários eram envolvidos para aprovar o resultados dos testes e gerar massa de dados



Dados Históricos

Agora

Atualmente

- Empresas têm visto a atividade de testes como prioritária para a maioria de suas aplicações
- Não testar o software pode gerar perdas financeiras e de imagem irreparáveis
- Aplicações cada vez mais complexas, cada vez mais distribuídas



Sumário

- 1 Introdução
 - Dados Históricos
 - O Antes e o Agora do Teste de Software
- 2 **Conceitos**
 - O que é Teste de Software?
 - Validação, Verificação e Teste de Software
 - Fases da Atividade de Teste
- 3 Técnicas e Critérios de Teste
 - Teste Funcional
 - Teste Estrutural
 - Teste Baseado em Defeitos
- 4 Execução dos Testes
- 5 Benefícios
- 6 Carreira Profissional
- 7 Perguntas



Conceitos

O que é Teste de Software?

“Processo de avaliar um software ou um componente de software para verificar se ele satisfaz os requisitos especificados ou identificar diferenças entre resultados esperados e obtidos.”

[ANSI/IEEE Standard 729, 2017(acesso on-line)]

“Processo de executar um programa com objetivo de encontrar erros.” **[Myers, 1979]**



Conceitos

Validação, Verificação e Teste de Software

- Processo de Desenvolvimento de Software
 - é uma tarefa complexa
 - está sujeita a diversos tipos de problemas
- “Erro Humano” como principal fator de causa de problemas
- VV&T tem a finalidade:
 - garantir o modo de construção do software
 - garantir que o software esteja em conformidade com o que foi especificado
 - acompanhar todo o processo de desenvolvimento do software, desde a concepção



Conceitos

Validação, Verificação e Teste de Software

- VV&T divide-se em duas atividades:
 - **Estáticas**
 - não requerem execução
 - não requerem a existência de um programa
 - não requerem um modelo executável
 - **Dinâmicas**
 - requerem que um programa ou modelo seja executado



Conceitos

Fases da Atividade de Teste

A atividade de teste é dividida em fases com objetivos distintos

- **Teste de Unidade**

- tem seu foco nas unidades menores do software: funções, procedimentos, métodos ou classes
- pode ser aplicado à medida que as unidade são desenvolvidas ou sofrem manutenção pelo próprio desenvolvedor
- o software não necessita estar totalmente finalizado

- **Teste de Integração**

- deve ser executado depois do teste de unidade
- tem a finalidade de verificar como as unidades se integram, trabalham juntas
- o testador deve ter grande conhecimento das estruturas internas no software
- preferencialmente deve ser executado pela equipe de desenvolvimento



Conceitos

Fases da Atividade de Teste

- **Teste de Sistema**
 - todo o software é testado
 - o objetivo é verificar se todas as funcionalidades especificadas foram desenvolvidas seguindo o requisito do Sistema
 - além dos requisitos funcionais, os **não funcionais** também são explorados, como: segurança, performance e robustez
 - pode-se usar uma equipe independente para execução deste teste
- **Teste de Regressão**
 - teste que é executado durante a manutenção do software
 - tem o objetivo de validar as novas implementações
 - comprovar que as funcionalidades antigas continuam funcionando



Sumário

- 1 Introdução
 - Dados Históricos
 - O Antes e o Agora do Teste de Software
- 2 Conceitos
 - O que é Teste de Software?
 - Validação, Verificação e Teste de Software
 - Fases da Atividade de Teste
- 3 Técnicas e Critérios de Teste**
 - Teste Funcional
 - Teste Estrutural
 - Teste Baseado em Defeitos
- 4 Execução dos Testes
- 5 Benefícios
- 6 Carreira Profissional
- 7 Perguntas



Técnicas e Critérios de Teste

Introdução

- Maneira sistemática e planejada de elaborar os casos de teste
- Podem ser usados de duas formas:
 - Para seleção dos casos de teste: quando estes são criados para satisfazer os requisitos do critério de teste
 - Para adequação dos casos de teste: quando estes são criados, por exemplo, aleatoriamente e então se verifica se esses casos de teste atendem aos requisitos do critério de teste



Técnicas e Critérios de Teste

Defeito, Erro e Falha

Podem parecer a mesma coisa, mas há diferenças

- **Defeito:** Deficiência algorítmica que, se ativada, pode levar a uma falha

- **Erro:** Estado de execução inconsistente

- **Falha:** Evento observável que mostra que o programa violou suas especificações



Teste Funcional

- Os requisitos de teste são extraídos da especificação do programa
- Aborda o software de um ponto de vista macroscópico. Por isso, é também chamado **Teste Caixa Preta**
- **Problema:**
 - Dificuldade em quantificar a atividade de teste - não se pode garantir que partes essenciais ou críticas do software foram executadas
- **Critérios:**
 - Particionamento de Equivalência
 - Análise do Valor Limite
 - Grafo Causa-Efeito



Teste Funcional

Particionamento de Equivalência

- Divide o domínio de entrada em *classes* ou *partições de equivalência* que podem ser tratadas da mesma maneira
- Observar também a *saída* do programa e verificar se, com base na saída, é possível estabelecer classes no domínio de entrada que permitam avaliar se a saída está sendo produzida corretamente
- As classes podem ser válidas ou inválidas
- Diretrizes para definição das classes:
 - se a condição de entrada especifica um intervalo, são definidas uma classe válida e duas inválidas
 - se a condição de entrada exige um valor específico, são definidas uma classe válida e duas inválidas
 - se a condição de entrada especifica um membro de um conjunto, são definidas uma classe válida e uma inválida
 - se a condição de entrada for booleana, são definidas uma classe válida e uma inválida



Teste Funcional

Particionamento de Equivalência

Observação:

- reduz o tamanho do domínio de entrada
- concentra-se em criar casos de teste baseados unicamente na especificação
- é especialmente adequado para aplicações em que as variáveis de entrada podem ser facilmente identificadas e podem ter valores distintos
- Problemas:
 - embora a especificação possa sugerir que um grupo de dados seja processado de forma idêntica, isso pode não ocorrer
 - a técnica não fornece um guia para a determinação dos dados de teste



Teste Funcional

Análise do Valor Limite

- Complementa o Particionamento de Equivalência
- Coloca sua atenção em uma fonte propícia a defeitos – os limites de uma classe ou partição de equivalência
- Os valores limites das classes é que devem ser selecionados: 0, 1, 20 e 21
- Os caracteres a serem encontrados devem estar na 1ª e na última posição



Teste Estrutural

- Os requisitos de teste são extraídos de uma implementação em particular
- Teste dos detalhes procedimentais. Por isso é também chamado Teste Caixa Branca
- A maioria dos critérios dessa técnica utiliza uma representação de programa conhecida como grafo de programa ou grafo de fluxo de controle
- Permite uma avaliação de cobertura do código
- Problema: caminhos não executáveis



Teste Estrutural

Teste Estrutural

Baseado em
Fluxo de Controle

Critérios:

- **Todos-Nós** (ou Todos-Comandos)
- **Todos-Ramos** (ou Todas-Arestas)
- Todas-Condições
- Todas-Decisões
- **Todos-Caminhos-Básicos**
- Todos-Caminhos
-e outros

Baseado em
Fluxo de Dados

Critérios:

- Todos-Usos
- Todos-p-usos
- Todos-c-usos
-e outros

Teste Estrutural

Grafo de Fluxo de Controle ou Grafo de Programa

- consiste de *nós* conectados por *arcos* com setas que mostram sua direção
- os *nós* representam blocos de comandos
 - **bloco de comando:** é um conjunto de comandos tal que se o primeiro comando for executado, então todos os comandos subsequentes também o serão
- os *arcos* indicam precedência, ou transferência de controle
- essa representação permite que o programa seja examinado independentemente de sua função



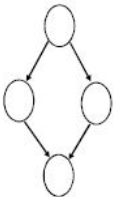
Teste Estrutural

Construções Básicas do Grafo de Fluxo

seqüência



if



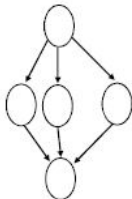
while



repeat



case

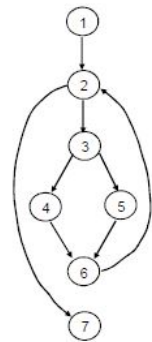


Teste Estrutural

Exemplo de Construção

```

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-enqto
fim } ⑦
    
```



Teste Estrutural

Critérios de Testes

- **Todos-Nós:** Estabelece como requisito de teste que sejam executados todos os comandos do programa ao menos uma vez
- **Todos-Ramos:** Estabelece como requisito de teste que sejam executadas todas as saídas *verdadeiro* e *falso* de todas as decisões



Teste Estrutural

Critérios de Testes

- **Todos-Caminhos:**

- Esse critério determina um conjunto básico de caminhos linearmente independentes, de modo que executando-os se garante a execução de todos os ramos ao menos uma vez
- Um caminho linearmente independente é aquele que contém ao menos um novo nó
- O número de caminhos é determinado pela fórmula da Complexidade Ciclomática de Mc'Cabe

$$V(G) = a - n + 2 \quad \text{ou}$$

$$V(G) = P + 1 \quad \text{ou}$$

$$V(G) = n^{\text{o. de regiões}}$$

sendo:

- G: um grafo direcionado
- a: arestas (ramos)
- n: nós
- P: no. de nós predicativos



Teste Estrutural

Teste de Fluxo de Dados - Descrição

- estabelece requisitos de teste que seguem o modelo de dados usado dentro do programa
- cada ocorrência de uma variável dentro de um programa pode ser classificada como sendo:
 - **def: definição** - quando uma variável é definida através de uma leitura ou quando ela aparece do lado esquerdo de um comando de atribuição, a essa variável é dado um valor
 - **c-use: uso-computacional** - quando a variável é usada na avaliação de uma expressão ou em um comando de saída
 - **p-use: uso-predicativo** - quando a variável ocorre em um predicado e, portanto, afeta o fluxo de controle do programa



Teste Estrutural

Teste de Fluxo de Dados - Aplicação

- seja a aplicação do critério de fluxo de dados: todos-usos (*usos* inclui c-uso e p-uso)
- a ideia é identificar e classificar todas as ocorrências de variáveis no programa e então gerar, para cada variável, dados de teste de modo que todas as definições e usos (denominado par d-u) sejam exercitadas
- as variáveis que precisam ser consideradas são: *x*, *i*, *c*, *achou*, *resposta* e o vetor *a*
- para cada uma dessas variáveis constrói-se uma tabela de pares d-u



Teste Baseado em Defeitos

Os requisitos de teste são estabelecidos com base nos defeitos típicos e comuns cometidos durante o desenvolvimento do software

- **Hipótese do programador competente**

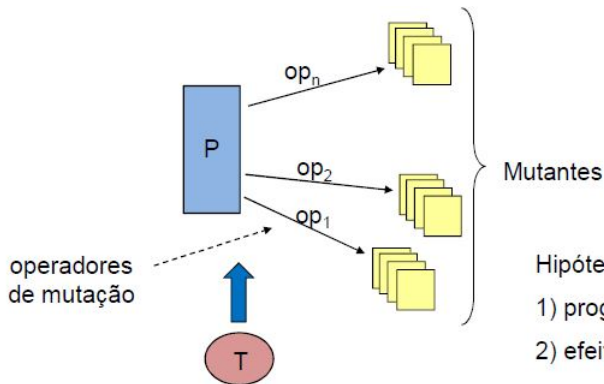
Programadores experientes escrevem programas corretos ou muito próximos do correto

- **Efeito do acoplamento:**

Casos de teste capazes de revelar erros simples são tão sensíveis que, implicitamente, também são capazes de revelar erros mais complexos



Teste Baseado em Defeitos



$$\text{escore de mutação} = \frac{\# \text{ mutantes mortos}}{\# \text{ mutantes gerados não equivalentes}}$$



Teste Baseado em Defeitos

- Os operadores de mutação determinam o tipo de alteração sintática que deve ser feita para a criação dos mutantes
 - Introduzir pequenas alterações semânticas através de pequenas alterações sintáticas que representam defeitos típicos
- Operadores dependem da linguagem alvo
 - FORTRAN (22 operadores) C (71 operadores)



Teste Baseado em Defeitos

Operadores de Mutação

- Retira um comando de cada vez do programa
- Troca um operador relacional por outro tipo de operador relacional
- Troca o comando while por do-while
- Interrompe a execução do laço após duas execuções
- Troca uma constante por outra constante
- Requer valor negativo, positivo e zero para cada referência escalar



Teste Baseado em Defeitos

Análise de Mutantes

- Dados **P** (Programa) e **T** (Teste)
- Passos para a aplicação da Análise de Mutantes
 - **P** é executado com os casos de teste de **T**
 - Mutantes são gerados
 - Mutantes são executados com os casos de teste de **T**
 - Mutantes são analisados



Teste Baseado em Defeitos

Análise de Mutantes - Observação

- Análise de Mutantes consegue mostrar a ausência de defeitos particulares, pois ao matar os mutantes mostra-se que o programa original não possui aquele defeito.
- Esse critério força o testador a analisar cuidadosamente o programa, uma vez que ele precisa criar casos de teste que exponham os defeitos introduzidos.
- **Desvantagem:**
 - é computacionalmente caro devido ao grande número de mutantes gerados, o tempo e recursos usados para compilar e executar todos os eles



Sumário

- 1 Introdução
 - Dados Históricos
 - O Antes e o Agora do Teste de Software
- 2 Conceitos
 - O que é Teste de Software?
 - Validação, Verificação e Teste de Software
 - Fases da Atividade de Teste
- 3 Técnicas e Critérios de Teste
 - Teste Funcional
 - Teste Estrutural
 - Teste Baseado em Defeitos
- 4 Execução dos Testes
- 5 Benefícios
- 6 Carreira Profissional
- 7 Perguntas



Passos para executar um Teste de Software

- O primeiro passo da área de Teste é criar um **Plano de Teste**, documento que contém os seguintes itens:
 - Requisitos que serão testados
 - Técnicas utilizadas
 - Documentos utilizados
 - Cronograma
 - Recursos



Passos para executar um Teste de Software

- O passo após o Plano de Teste é criar o(s) Caso(s) de Teste. É com ele(s) que iremos validar a aplicação, devendo conter:
 - Passos para execução
 - Resultados esperados
 - Ambiente onde será testado
 - Pré condições
 - Massa de dados



Passos para executar um Teste de Software

- Com os Casos de Teste criados, podemos executá-los na aplicação
- Caso ocorra algum erro, abrimos um defeito com as seguintes informações:
 - Sumário
 - Passos para a reprodução
 - Prioridade
 - Severidade
 - Ambiente
- Após a execução de todos os testes e correção dos defeitos, o sistema pode ser entregue ao cliente, junto com o relatório **“Sumário de Teste”** apresentando todo o andamento dos testes



Sumário

- 1 Introdução
 - Dados Históricos
 - O Antes e o Agora do Teste de Software
- 2 Conceitos
 - O que é Teste de Software?
 - Validação, Verificação e Teste de Software
 - Fases da Atividade de Teste
- 3 Técnicas e Critérios de Teste
 - Teste Funcional
 - Teste Estrutural
 - Teste Baseado em Defeitos
- 4 Execução dos Testes
- 5 Benefícios**
- 6 Carreira Profissional
- 7 Perguntas



Benefícios

- Liberação de produtos mais estáveis
- Diminuição do custo de cada falha
- Aumento da satisfação do cliente com produtos que realmente atendam às necessidades



Sumário

- 1 Introdução
 - Dados Históricos
 - O Antes e o Agora do Teste de Software
- 2 Conceitos
 - O que é Teste de Software?
 - Validação, Verificação e Teste de Software
 - Fases da Atividade de Teste
- 3 Técnicas e Critérios de Teste
 - Teste Funcional
 - Teste Estrutural
 - Teste Baseado em Defeitos
- 4 Execução dos Testes
- 5 Benefícios
- 6 Carreira Profissional**
- 7 Perguntas



Profissionais de Teste de Software

Testador

- Execução dos testes funcionais no sistema guiado pelo Caso de Teste

Automatizador de Teste

- Criação dos scripts de automação do sistema guiado pelo Caso de Teste

Analista de Teste

- Criação do Plano de Teste e Casos de Teste

Engenheiro/Arquiteto de Teste

- Criação e execução de Testes Não Funcionais, criação do Ambiente de Teste e Automação

Líder/Coordenador de Teste

- Acompanhamento dos Testes e criação de Relatórios de Teste. Apoio ao time de Teste



Perfil dos Profissionais

- Detalhista
- Criativo
- Organizado
- Sede de conhecimento
- Questionador
- Comunicativo
- Perfeccionista



Carreira em Teste de Software

- Carência de profissionais de teste
- Carência de ambientes estruturados para teste
- Investimento acadêmico em centros de teste
- Visto ainda como novidade
- Muitas empresas querem testar, mas não sabem COMO testar
- Especialização em áreas específicas do teste



Sumário

- 1 Introdução
 - Dados Históricos
 - O Antes e o Agora do Teste de Software
- 2 Conceitos
 - O que é Teste de Software?
 - Validação, Verificação e Teste de Software
 - Fases da Atividade de Teste
- 3 Técnicas e Critérios de Teste
 - Teste Funcional
 - Teste Estrutural
 - Teste Baseado em Defeitos
- 4 Execução dos Testes
- 5 Benefícios
- 6 Carreira Profissional
- 7 Perguntas



