



Universidade Federal do Acre
Centro de Ciências Exatas e Tecnológicas
Curso de Bacharelado em Sistemas de Informação

Teste de Software

Prof. Daricélio Moreira Soares



Agenda

- Teste de Software: Definição e Conceitos
- Técnicas de Teste
- Fases de Teste
- Processo de Teste
- Automatização do Processo de Teste

Alguma anotação



Teste de Software

- É o processo de executar um programa com o objetivo de encontrar defeitos (Myers, 1979).
- É, portanto, uma atividade de V&V dinâmica.
- 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.



Perspectiva de Teste

- Bons testadores necessitam de um conjunto especial de habilidades. Um testador deve abordar um software com a atitude de questionar tudo sobre ele.
- A perspectiva de teste é, um modo de olhar qualquer produto de desenvolvimento e questionar a sua validade.
- Habilidades requeridas na perspectiva de teste:
 - Querer prova de qualidade,
 - Não fazer suposições,
 - Não deixar passar áreas importantes,
 - Procurar ser reproduzível.



Perspectiva de Teste

- A perspectiva de teste requer que um fragmento de software demonstre não apenas que ele executa de acordo com o especificado, mas que executa apenas o especificado.
- O software faz o que deveria fazer e somente isso?



Teste de Software

- Executa-se um programa ou modelo utilizando algumas entradas em particular e verificar-se se seu comportamento está de acordo com o esperado.
- Caso a execução apresente algum resultado não especificado, um defeito foi identificado.
- Os dados da execução podem servir como fonte para a localização e correção de defeitos, mas teste não é depuração .).



Conceitos

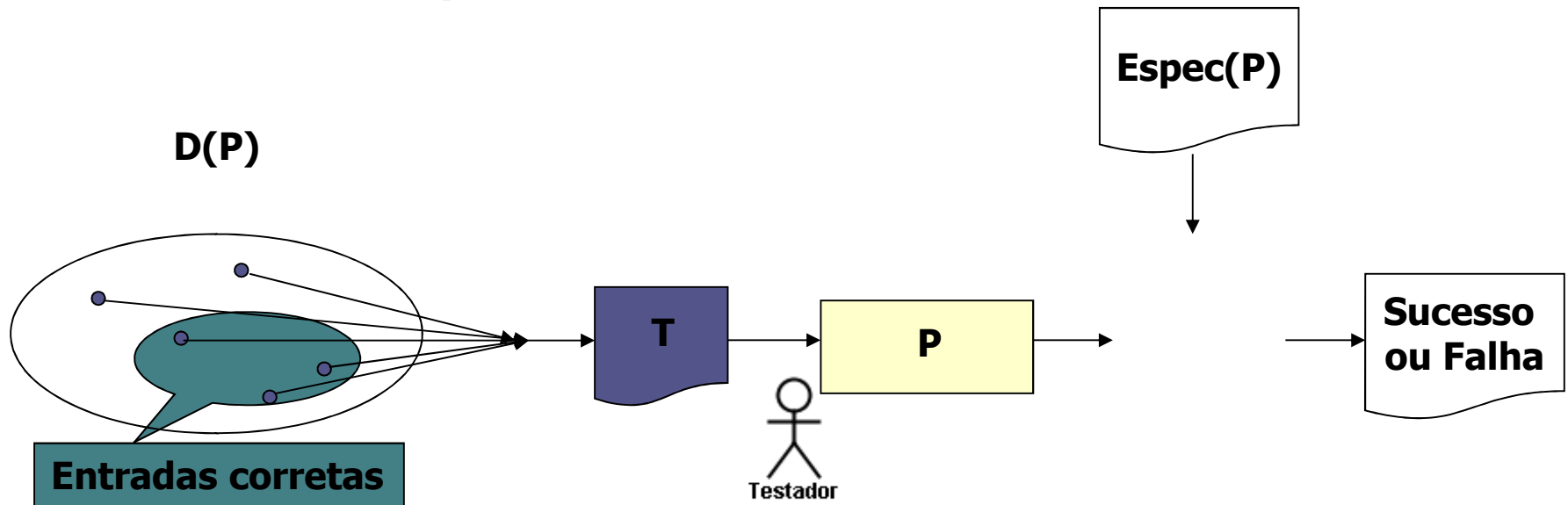
- Seja **P** um programa a ser testado.
- O domínio de entrada de **P** (**D(P)**) é o conjunto de todos os valores possíveis que podem ser utilizados para executar **P**.
- Um dado de teste para **P** é um elemento de **D(P)**.
- O domínio de saída de **P** é o conjunto de todos os possíveis resultados produzidos por **P**.
- Um caso de teste de **P** é um par formado por um dado de teste mais o resultado esperado para a execução de **P** com aquele dado de teste.
- Ao conjunto de todos os casos de teste usados durante uma determinada atividade de teste dá-se o nome de conjunto de casos de teste ou conjunto de teste.



Cenário Típico da Atividade de Teste

- Definido um conjunto de casos de teste **T**, executa-se **P** com **T** e verificam-se os resultados obtidos.
- Se os resultados obtidos coincidem com os resultados esperados, então nenhum defeito foi identificado (“O software passou no teste”).
- Se, para algum caso de teste, o resultado obtido difere do esperado, então um defeito foi detectado (“O software não passou no teste”).
- De maneira geral, fica por conta do testador, baseado na especificação do programa, decidir sobre a correção da execução (Delamaro et al., 2007).

Cenário Típico da Atividade de Teste



Teste Exaustivo

- Para se poder garantir que **P** não contém defeitos, **P** deveria ser executado com todos os elementos de **D(P)**.
- Seja um programa **exp** com a seguinte especificação: $\text{exp}(\text{int } x, \text{int } y) = x^y$
 - **D(exp)** = todos os pares de números inteiros (x,y) passíveis de representação.
 - Cardinalidade (#) de **D(exp)** = $2^n * 2^n$, onde n = número de bits usado para representar um inteiro.
 - Em uma arquitetura de 32 bits, $\#(\text{D}(\text{exp})) = 2^{64}$
 - Se cada teste puder ser executado em 1 milissegundo, seriam necessários aproximadamente 5,85 milhões de séculos para executar todos os testes.
- Logo, em geral, teste exaustivo não é viável.



Teste de Software

- Ao se testar **P**, devem-se selecionar alguns pontos específicos de **D(P)** para executar.
- Portanto, testes podem mostrar apenas a presença de defeitos, mas não a ausência deles.
- Um aspecto crucial para o sucesso na atividade de teste é a escolha correta dos casos de teste. Um teste bem-sucedido identifica defeitos que ainda não foram detectados.
- Um bom caso de teste é aquele que tem alta probabilidade de encontrar um defeito ainda não descoberto.



Teste de Software

- A escolha de casos de teste passa pela identificação de subdomínios de teste.
- Um subdomínio de teste é um subconjunto de $\mathbf{D(P)}$ que contém dados de teste semelhantes, ou seja, que se comportam do mesmo modo; por isso, basta executar \mathbf{P} com um deles.
- Fazendo-se isso com todos os subdomínios de $\mathbf{D(P)}$, consegue-se um conjunto de teste \mathbf{T} bastante reduzido em relação a $\mathbf{D(P)}$, mas que, de certa maneira, representa cada um de seus elementos (Delamaro et al., 2007).



Teste de Software

- Existem duas maneiras de se selecionar elementos de cada um dos subdomínios de teste (Delamaro et al., 2007):
 - Teste Aleatório: um grande número de casos de teste é selecionado aleatoriamente, de modo que, probabilisticamente, se tenha uma boa chance de ter todos os subdomínios representados em T.
 - Teste de Subdomínios ou Teste de Partição: procura-se estabelecer quais são os subdomínios a serem utilizados e, então, selecionam-se os casos de teste em cada subdomínio.



Teste Funcional ou Caixa-Preta

- Técnica de projeto de casos de teste na qual o programa ou sistema é considerado uma caixa-preta.
- Para testá-lo, são fornecidas entradas e avaliadas as saídas geradas para verificar se estão em conformidade com os objetivos especificados.
- Os detalhes de implementação não são considerados.
- O software é avaliado segundo o ponto de vista do usuário (Delamaro et al., 2007).



Teste Estrutural ou Caixa Branca

- Estabelece os requisitos de teste com base em uma dada implementação, requerendo a execução de partes ou componentes elementares de um programa.
- Baseia-se no conhecimento da estrutura interna do programa.
- Caminhos lógicos são testados, estabelecendo casos de teste que põem à prova condições, laços, definições e usos de variáveis.



Teste Estrutural

- Blocos disjuntos de comandos são considerados nós.
- A execução do primeiro comando do bloco acarreta a execução de todos os outros comandos desse bloco, na ordem dada.
- Todos os comandos em um bloco têm um único predecessor (possivelmente com exceção do primeiro) e um único sucessor (possivelmente com exceção do último) (Delamaro et al., 2007).



Teste Estrutural

- Essa técnica é vista como complementar às demais, uma vez que cobre classes distintas de defeitos.
- As informações obtidas pela aplicação de critérios estruturais são muito úteis para as atividades de depuração e manutenção (Delamaro et al., 2007).



Teste Baseado em Modelos

- Boa parte do tempo despendido na atividade de teste é gasta buscando-se identificar o que o sistema deveria saber.
- Antes de se perguntar se o resultado está correto, deve-se saber qual seria o resultado correto.
- Um modelo é muito útil para tal, servindo tanto como oráculo (instrumento que decide se a saída obtida coincide com a saída esperada) quanto como base para a definição de casos de teste.
- Existem diversas técnicas baseadas em Máquinas de Transições de Estados (Delamaro et al., 2007).

Teste Baseado em Defeito ou Teste de Mutação

- Defeitos típicos do processo de implementação são utilizados como critérios de teste.
- O programa que está sendo testado é alterado diversas vezes, criando-se um conjunto de programas mutantes, inserindo defeitos no programa original.
- O trabalho do testador é escolher casos de teste que mostrem a diferença de comportamento entre o programa original e os programas mutantes.
- Cada mutante determina um subdomínio de teste relacionado com um defeito específico (Delamaro et al., 2007).



Teste de Mutação

- Envolve os seguintes passos:
 1. Geração dos mutantes;
 2. Execução do programa em teste;
 3. Execução dos mutantes;
 4. Análise dos mutantes.
- É praticamente impossível executar 1, 3 e 4 sem apoio computacional. Assim, na prática, deve-se associar à aplicação dessa técnica uma ferramenta de suporte.
- Na geração de mutantes, operadores de mutação são aplicados. Ex.: eliminação de comandos, troca de operador relacional, troca de variáveis escalares etc (Delamaro et al., 2007).



Fases de Teste

- A atividade de teste é dividida em fases com objetivos distintos. De uma forma geral, pode-se estabelecer como fases (Delamaro et al., 2007):
 - Teste de Unidade
 - Teste de Integração
 - Teste de Sistema
 - Teste de Aceitação
 - Teste de Operação



Teste de Unidade

- Consiste em validar dados válidos e inválidos via I/O sendo aplicado por desenvolvedores ou analistas de teste.
- Uma unidade é a menor parte testável de um programa de computador.
- Domínio, Senha, Valores, Tipos



Teste de Integração

- É a fase do teste de software em que módulos são combinados e testados em grupo.
- Ela sucede o teste de unidade, em que os módulos são testados individualmente, e antecede o teste de sistema.
- O propósito do teste de integração é verificar os requisitos funcionais, de desempenho e de confiabilidade na modelagem do sistema.
- Com ele é possível descobrir erros de interface entre os componentes do sistema.



Teste de Sistema

- Fase do processo de teste de software e de hardware em que o sistema já completamente integrado é verificado quanto a seus requisitos num ambiente de produção.
- É um teste mais limitado em relação aos testes de unidade e de integração, pois se preocupa somente com aspectos gerais do sistema.
- Não se limita a testar somente requisitos funcionais, mas também requisitos não funcionais como a expectativa do cliente.



Teste de Aceitação

- Fase do processo em que um teste de caixa-preta é realizado num sistema antes de sua disponibilização.
- Verifica o sistema em relação aos seus requisitos originais, e às necessidades atuais do usuário.
- É geralmente realizado por um grupo restrito de usuários finais, num ambiente parecido com o deles.



Teste de Operação

- Fase de teste que é conduzido pelos administradores do ambiente final onde o sistema ou software entrará em ambiente produtivo.
- Vale ressaltar que essa fase é aplicável somente a sistemas de informação próprios de uma organização, cujo acesso pode ser feito interna e/ou externamente a essa organização.
- Nessa fase de teste devem ser feitas simulações para garantir que a entrada em produção do sistema será bem sucedida.
- Envolve testes de instalação, simulações com *backup* e *restore* das bases de dados, etc.



Processo de Teste

- O processo de teste pode ser definido como um processo separado, mas intimamente ligado, ao processo de desenvolvimento. Isso porque eles têm metas e medidas de sucesso diferentes.
- Por exemplo, quanto menor a taxa de defeitos (razão entre o n° de casos de teste que falham pelo total de casos de teste), melhor sucedido é considerado o processo de desenvolvimento. Por outro lado, quanto maior a taxa de defeitos, considera-se melhor sucedido o processo de teste (McGregor e Sykes, 2001).



Processo de Teste

- Independentemente da fase de teste, o processo de teste inclui as seguintes atividades:
 - Planejamento
 - Análise de Requisitos de Teste
 - Projeto de Casos de Teste
 - Implementação de Casos de Teste
 - Execução
 - Análise



Automatização do Processo de Teste

- É um ponto importante para o sucesso no teste de um software.
- O processo de teste tende a ser extremamente dispendioso e é muito importante utilizar ferramentas de apoio ao teste para buscar aumentar a produtividade.
- Há diversos tipos de ferramentas de teste. Ainda assim, muito trabalho humano é necessário para criar os casos de teste adequados ao critério utilizado (Delamaro et al., 2007).