Análise e Desenvolvimento de Sistemas

Qualidade de Software

Objetivos

- Estratégia de Testes

Testes de Software

Conceito

Teste de software é todo e qualquer procedimento que ajuda a determinar se o programa atinge as expectativas para as quais foi criado.

Testar um software consiste em verificar se ele atende às expectativas, se seu funcionamento é limpo, amigável e correto e se ele enquadra no ambiente para o qual foi projetado.

Testes de Software

Figura 1.1 Defeito *versus* erro *versus* falha.

Fonte: adaptada de Dias Neto (2010).

Testes de Software

Erro

- Falha humana: produz resultado incorreto
- É, por exemplo, a falha na escrita de um código específico.

Defeito

- Resultado de um código mal escrito, ou seja, um erro.
- Causa anomalia no funcionamento do sistema. O usuário final normalmente não vê o defeito propriamente dito.
- Também é conhecido como *bug.*

Falha

- Quando um código que apresenta um defeito é executado, temos uma falha.
- Funcionamento inesperado das funções do software.
- É a camada que chega aos olhos do usuário.

Testes de Software

Sendo assim, pode dizer que temos um efeito cascata.

1. Toda falha precisa de um defeito para ocorrer, mas nem todo defeito acarreta uma falha.
2. Todo defeito precisa de um erro para ocorrer, mas nem todo erro acarreta um defeito.
3. Toda falha sempre se origina em um erro.

Testes de Software

Quadro 1.1 Defeito *versus* erro *versus* falha no exemplo do caixa eletrônico.

Conceito	Definição	Onde acontece
Erro	Código (da função "saque") mal escrito. Esse código mal escrito pode ou não acarretar defeitos.	No universo físico, enquanto o programador escreve o programa.
Defeito	Quando o cliente seleciona "saque" e ativa o código mal escrito, que reflete um funcionamento anormal, não esperado pelo sistema.	No universo da informação, ou seja, na troca de dados e chamados entre elementos do programa.
Falha	Ato de entregar cheques, e não notas, decorrente da anomalia da execução de um código com erro. É o resultado que não bate com a expectativa.	No universo do usuário. É o resultado final após a execução das tarefas. O usuário não vê, necessariamente, erros e defeitos. Ele tem acesso apenas às falhas.

Fonte: adaptado de Dias Neto (2010).

Testes de Software

Um programa falhar significa que não alcançou os resultados esperados, isso pode ser causado, entre outros motivos, por:

- Códigos errados e incompletos;
- Limitações de hardware ou software;
- Inexistência de diferenciação entre tipos de usuários;
- Defeito no controle de algoritmos, entre outros.

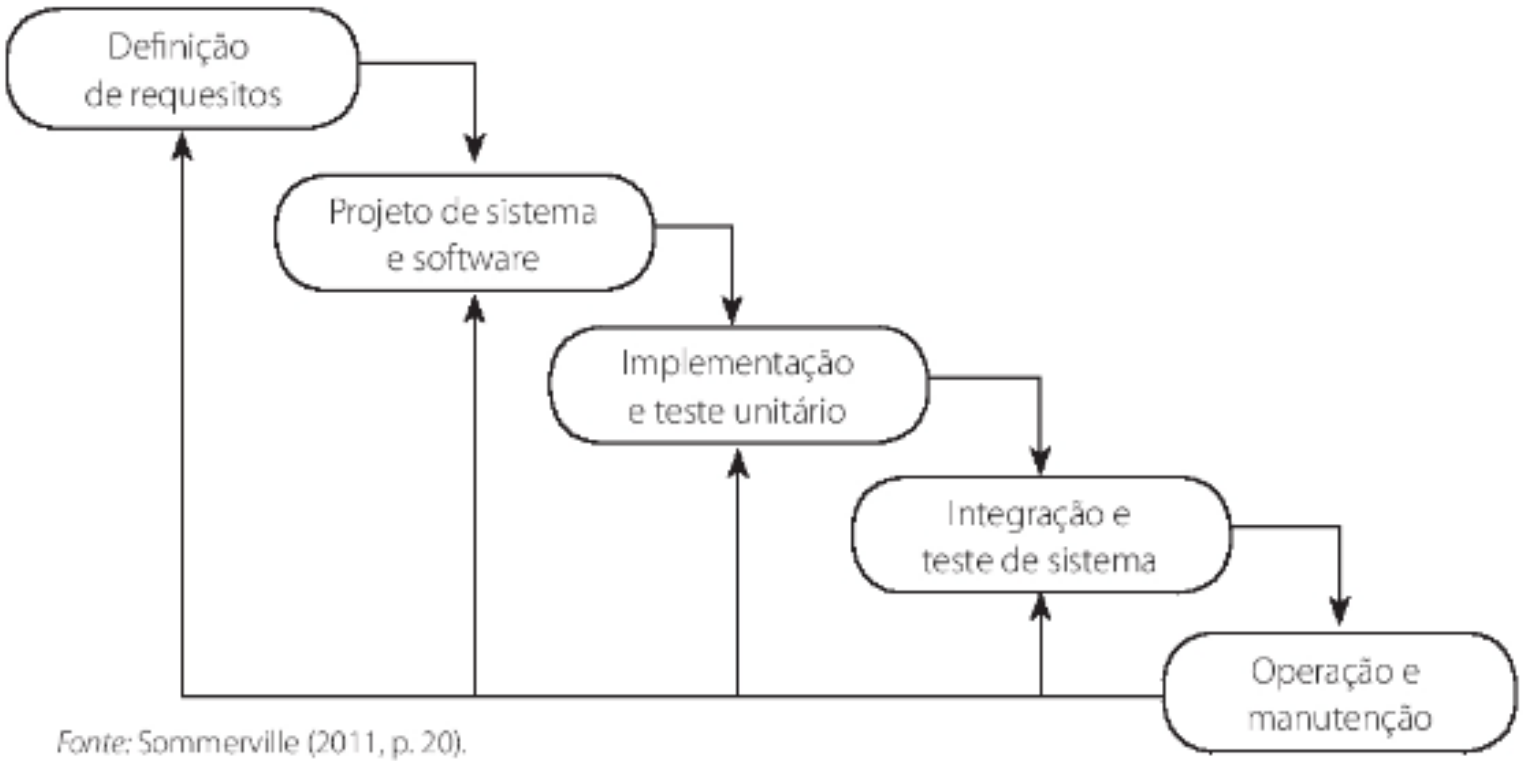
Testes de Software

Consequências práticas de duas grandes causas de erros:

1. Uma equipe muito grande e variada.
2. Diversas modificações ao longo as etapas do ciclo de desenvolvimento de software.

Testes de Software

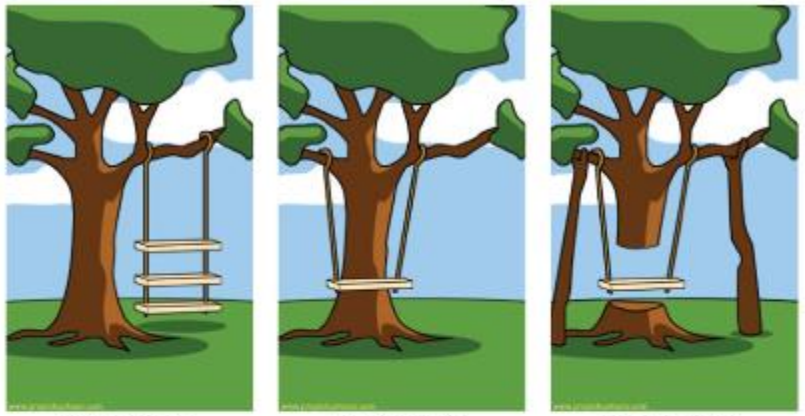
Figura 1.2 O modelo em cascata.



Fonte: Sommerville (2011, p. 20).

Testes de Software

Figura 1.3 Como os projetos funcionam de verdade.



Fonte: adaptada de Flow Projects... (2007).

Testes de Software

O analista de testes de software

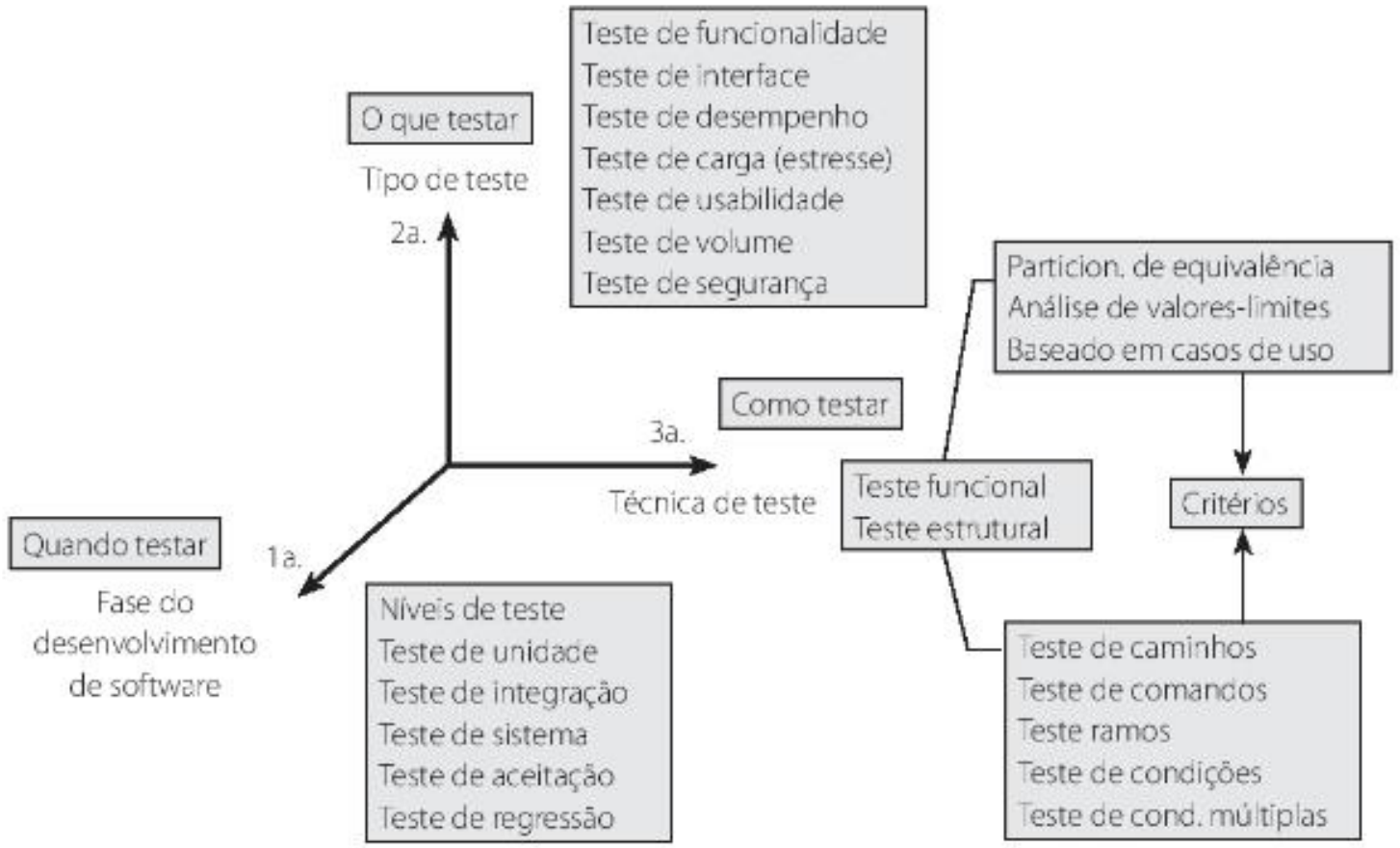
O analista de testes é o membro da equipe que identifica os testes cabíveis ao software e quais são efetivamente necessários, ele monitora a abrangência dos testes e conduz todos os dados, munindo a equipe de ferramentas para obter soluções para os problemas encontrados, alcançando, assim, um resultado satisfatório.

Qualidade de Software

Diz respeito a algo benfeito, não somente no aspecto físico ou necessariamente visual, mas também ao que todo o programa se destina. Os testes existem para verificar se está tudo certo, sendo assim, a qualidade do software é testada nas verificações realizadas antes da entrega para o cliente.

Testes de Software

Figura 1.4 Relação entre níveis, tipos e técnicas de teste.

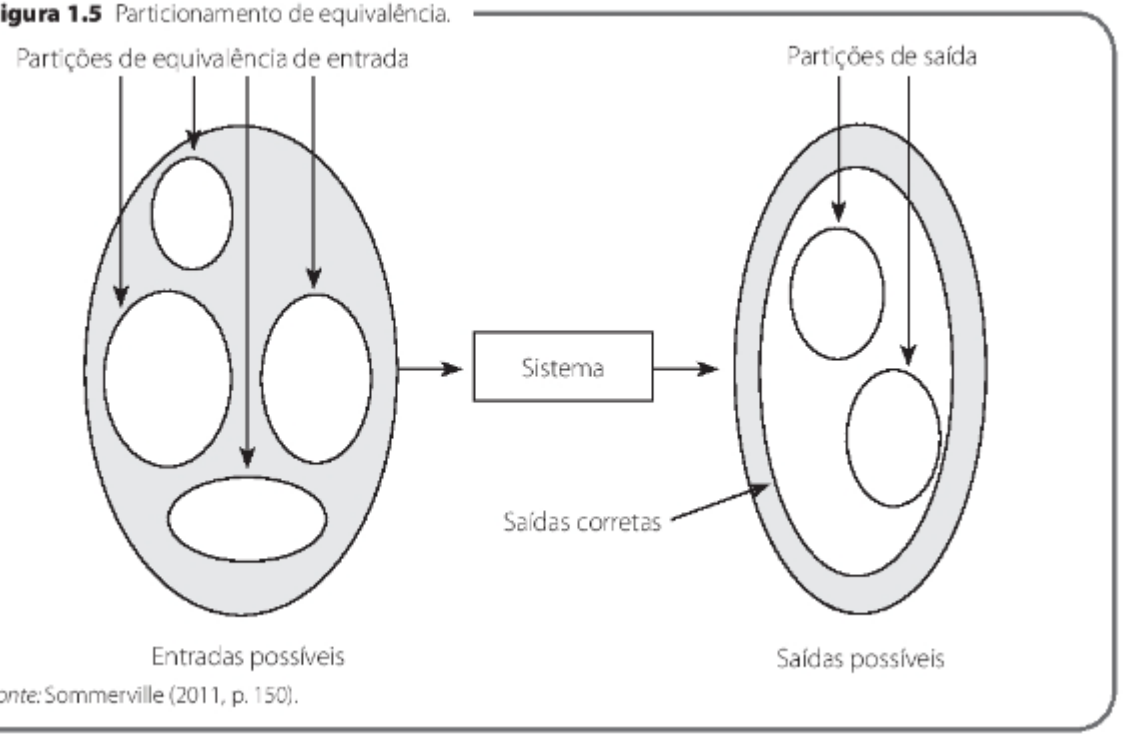


Fonte: Crespo et al. (2004, p. 4).

Testes de Software

Teste unitário – é conhecido como teste de unidade, possibilita que cada módulo do sistema seja analisado de maneira individual, verificando se sua funcionalidade está correta.

Figura 1.5 Particionamento de equivalência.

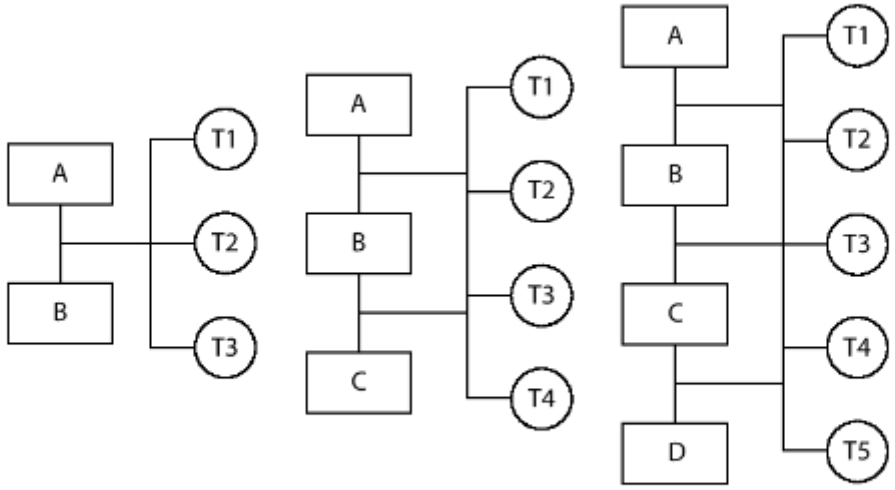


Fonte: Sommerville (2011, p. 150).

Testes de Software

Teste de integração – o objetivo desse teste é verificar se a comunicação entre módulos é feita de modo correto.

Figura 1.6 Teste de integração incremental.



Fonte: Sommerville (2004, p. 386).

Testes de Software

Teste de sistema – é um teste no qual é possível encontrar falhas às quais o usuário final pode ter acesso, de forma a serem avaliados do ponto de vista do usuário, como:

- A qualidade da interface gráfica;
- O funcionamento correto das funções do sistemas;
- Se as precondições das operações no projeto são atendidas;

Testes de Software

Teste de aceitação – são parecidos com os testes de sistema, mas essa verificação é muito mais voltada para a relação do usuário final com o programa do que com o sistema propriamente dito, que o usuário não vê.

- Teste de aceitação formal
  - O que será testado.
  - Como será testado.

Testes de Software

Teste de aceitação

- Teste de aceitação formal
  - O que será testado.
  - Como será testado.

Quadro 1.2 Vantagens e desvantagens do teste de aceitação formal.

Vantagens	Os recursos e as funções que serão testados no sistema são conhecidos pelos testadores.
	Os detalhes e a finalidade do teste são conhecidos e, por isso mesmo, mensuráveis.
	Possibilidade de automação do teste, o que permite a realização do teste de regressão, que você verá adiante.
	Possibilidade de monitoração e mensuração do progresso do teste.
	Critérios de aceitabilidade conhecidos pelos testadores.
Desvantagens	Grande planejamento.
	Uso significativo de recursos.
	Pode-se chegar aos mesmos resultados do teste de sistema.
	São procurados apenas defeitos já esperados no uso final.

Fonte: adaptado de UFPE (2006).

Testes de Software

Teste de aceitação

- Teste de aceitação informal
  - O tipo informal deixa o usuário mais “livre” ao testar o programa.

Quadro 1.3 Vantagens e desvantagens do teste de aceitação informal.

Vantagens	Recursos e funções testados também são conhecidos (assim como no teste de aceitação formal).
	Possibilidade de monitoração do progresso do teste.
	Critérios de aceitabilidade conhecidos.
	São apresentados e identificados erros mais subjetivos que no teste de aceitação formal, justamente por depender da interação de cada testador. Isso aumenta a possibilidade de correção de novos erros e, o mais importante, a prevenção de erros futuros.
Desvantagens	Grandes recursos de gerenciamento.
	Não é possível controlar o modo que o sistema será testado.
	Existe a possibilidade de os usuários testadores se adaptarem e gostarem do sistema e, exatamente por isso, não encontrarem possíveis erros presentes.
	Existe, também, a possibilidade de os usuários testadores compararem o sistema que estão testando com uma versão mais antiga. Isso desvia o foco do teste e, consequentemente, atrapalha a detecção de erros reais.
	Os recursos do teste de aceitação informal não são controláveis.

Fonte: adaptado de UFPE (2006).

Testes de Software

Teste de aceitação

- Teste Beta
  - É o menos controlável, seus requisitos são de inteira responsabilidade do testador.

Quadro 1.4 Vantagens e desvantagens do teste beta.

Vantagens	Pode ser realizado diretamente pelos usuários finais.
	Aquele que participa do teste tem sua satisfação elevada.
	Defeitos extremamente subjetivos são revelados.
Desvantagens	Nem todas as funções ou recursos podem ser testados, uma vez que o usuário é livre para usar o programa como deseja.
	A mensuração dos níveis de teste é quase impossível.
	Assim como no teste de aceitação informal, os usuários podem acabar se concentrando na comparação do sistema com uma versão antiga. Além disso, também podem se adaptar e gostar da versão beta. Em ambos os casos, possíveis erros podem não ser identificados.
	Os critérios de aceitabilidade não são conhecidos – são extremamente subjetivos e individuais.

Fonte: adaptado de UFPE (2006).

Testes de Software

Teste de regressão – a reaplicação dos testes visto até agora, porém nas novas versões do sistema, para verificar possíveis novos erros após a solução de erros anteriores.

Teste de integridade de dados – consiste no processo de validação da confiabilidade e da integridade dos dados de um determinado sistema. Ele assegurar a robustez o programa, sua capacidade de resistir a falhas.

Testes de Software

Testes não funcionais: são todos os testes que têm foco, obviamente, em requisitos não funcionais. Por requisitos funcionais podemos considerar a interface entre módulos, o funcionamento do sistema como um todo, o processamento de entradas, o fornecimento de saídas e demais atividades que estejam relacionadas estritamente às funções técnica de um software.

Testes de Software

Abordagem de Testes

Caixa branca – a técnica da caixa-branca possibilita que o *loop* seja testado apenas algumas vezes, verificando somente os caminhos lógica relevantes.

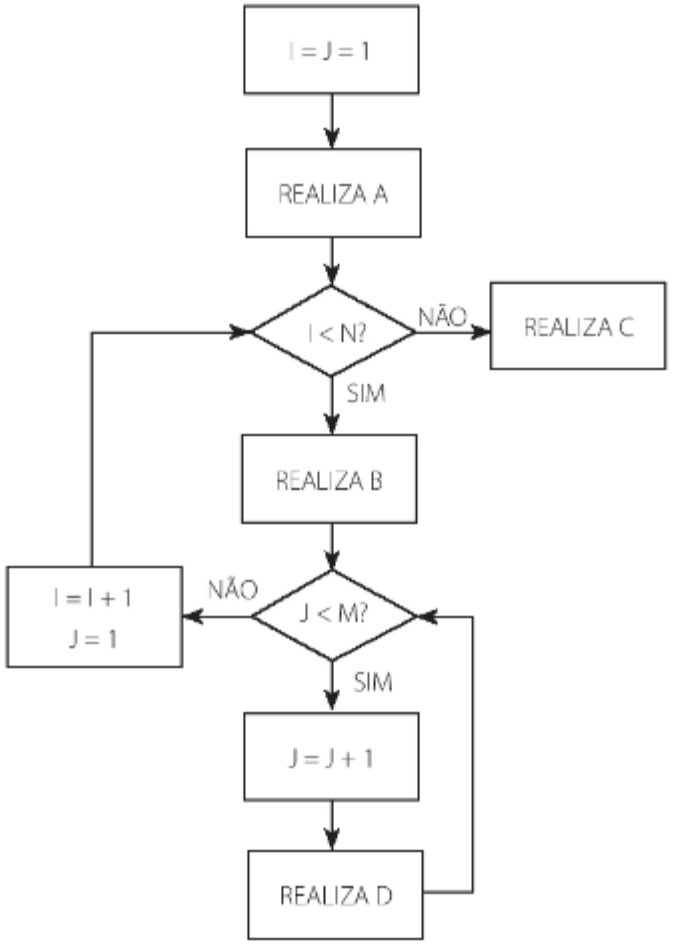
Por exemplo:

- Um valor para I que seja menor que *n*, igual a *n* e maior que *n*.
- Um valor para J que seja menor que *m*, igual a *m* e maior que *m*.
- Combinar os três valores de J com os três valores de I.

Testes de Software

Abordagem de Testes – Caixa branca

Figura 1.7 Exemplo de estrutura lógica.



Fonte: Pfleeger (2004, p. 278).

Testes de Software

Abordagem de Testes

Caixa-preta – você não levará em consideração o comportamento interno do sistema quando utilizar esta abordagem. Isso significa que você terá dados de entrada que serão fornecidos para a realização do teste, por fim, irá comparar as saídas decorrentes dessas entradas com os resultados esperados e previamente conhecidos.

Atividade Teams

- Realizar a tarefa no Teams.

Próxima Aula

- Técnicas de Testes Funcionais

Referência Bibliográfica

Braga, Pedro H. C. (Organizador). Testes de Software [BV:PE]. 1ª Ed.. São Paulo: Pearson, 2016.