

1 - Estratégias de Testes de Software

Ao longo do processo de desenvolvimento o software precisa ser testado em diferentes fases. A Figura 1 mostra a sequência de realização dos testes e a seguir tem-se o que é testado em cada fase de teste:

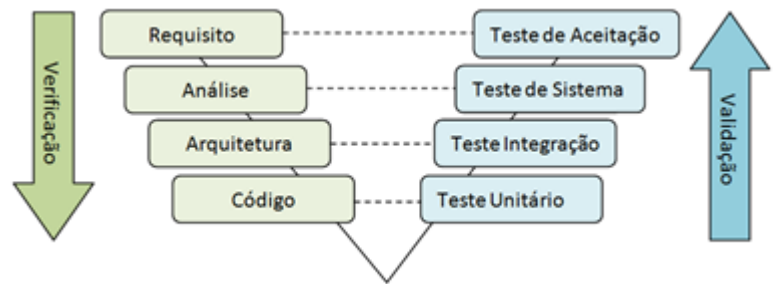


Figura 1 – Modelo V (fases do desenvolvimento x fases dos testes).

Fonte: <http://www.devmedia.com.br/teste-de-integracao-na-pratica/31877>

- Teste de unidade:
 - Tem por objetivo testar individualmente cada unidade programada, como exemplo, uma classe;
 - Os testes de unidade não testam somente as entradas e saídas da unidade, mas também podem testar caminhos específicos na estrutura de controle do código (teste de caixa branca);
 - Esses testes são realizados pelo próprio programador.
- Teste de integração:
 - Enquanto o teste de unidade se preocupa somente com o funcionamento de uma unidade por vez, o teste de integração se preocupa com o funcionamento integrado das unidades;
 - Unidades funcionam bem de modo isolado, porém quando colocadas juntas, situações inesperadas podem ocorrer. Os testes de integração buscam descobrir **erros de interface** entre os módulos/classes e de **dependências entre os componentes** da aplicação;
 - Faz uso de técnicas de projeto de casos de teste que enfocam as entradas e saídas, além de exercitar caminhos específicos (usando diagramas e grafos);
 - Esses testes podem ser realizados pelo próprio desenvolvedor e pela equipe independente de teste.
- Teste de sistema:
 - Testa a combinação do software com outros elementos do sistema, tais como, hardwares, bancos de dados e aplicativos de terceiros;
 - O objetivo é exercitar o sistema por completo;
 - Verifica se a função/desempenho global do sistema é alcançado, ou seja, não se limita a checar somente os requisitos funcionais, mas também os requisitos não funcionais, tais como, uso dos recursos computacionais, tempo de resposta, capacidade de recuperação, estresse, segurança e instalação;
 - Esses testes são realizados pela equipe independente de teste.
- Teste de aceitação:
 - Verifica se o software cumpriu os requisitos especificados;
 - Esses testes são realizados pela equipe independente de teste e podem incluir usuários finais para conferir se os requisitos foram atingidos satisfatoriamente.

2 - Testes de Sistema

O teste de sistema pode ser considerado um tipo de teste de caixa-preta, pois não requer verificar estruturas internas do sistema. São testes aplicados no sistema completo e integrado para avaliar a sua conformidade com os requisitos especificados, sendo basicamente de natureza não-funcionais. Ele pode detectar tanto defeitos de interface do sistema com outros sistemas/hardwares quanto defeitos em componentes internos do próprio sistema.

2.1 - Tipos de Testes de Sistema

A seguir tem-se alguns testes de sistema, esses testes não necessariamente são aplicados a todos os tipos de sistemas, ou seja, cada sistema pode requer um conjunto distinto de tipo de teste. Por exemplo, um sistema de vendas na Web pode precisar de um teste de carga para verificar a capacidade do sistema numa situação de múltiplos acessos, mas, como exemplo, não precisar de um teste de volume, pois não iria precisar processar um conjunto de vendas armazenadas na sua base de dados.

2.1.1 - Teste de Recuperação

É um teste de sistema que força o software a falhar de diversos modos e verifica se a recuperação é adequadamente realizada. Ele basicamente é feito a fim de verificar o quão rápido e melhor a aplicação pode se recuperar contra qualquer tipo de acidente ou falha. O tipo ou extensão da recuperação deve constar nas especificações de requisitos. A recuperação pode ser automática ou com intervenção humana:

- Automática (pelo sistema): a reinicialização, os mecanismos de verificação, a recuperação dos dados e o reinício são avaliados quanto à correção;
- Recuperação com intervenção humana: o tempo médio para reparo é avaliado para determinar se está dentro de limites aceitáveis.

As falhas podem ser causadas de diferentes maneiras, tais como:

- Desligar o computador durante uma execução de algum procedimento;
- Interromper o tráfego de dados durante a execução de algum procedimento.

2.1.2 - Teste de Segurança

Verifica se todos os mecanismos de proteção realmente protegem o software de acessos indevidos, ou seja, busca revelar falhas nos mecanismos de segurança do sistema. Os testes de segurança dependem dos requisitos de segurança do sistema, mas eles podem envolver testes, tais como:

- Confiabilidade: o sistema não divulga dados a usuários não autorizados;
- Integridade: o sistema não permite que dados sejam modificados por usuários não autorizados;
- Autenticação: garantir que o acesso ou dado tenha a origem garantida, ou seja, o usuário obteve autorização do sistema;
- Autorização: o sistema usa o controle de acesso especificado para cada aplicação;
- Disponibilidade: o sistema dá acesso ao serviço ou dado quando solicitado;
- Não repúdio: o usuário não pode negar a autoria de uma ação ou dado, ou seja, o sistema deve manter registros que identifiquem as operações executadas por cada usuário.

2.1.3 - Teste de Estresse

Também conhecido por teste de tortura, é uma forma de teste intensa, usada para determinar a estabilidade do sistema. Ele envolve o teste para **além da capacidade** de operação normal, muitas vezes, a um ponto de ruptura, de modo a observar os resultados. Ele **busca determinar o desempenho máximo possível do sistema em uma determinada tarefa ou função**. As razões podem incluir:

- Determinar pontos de ruptura ou limites de uso seguro;
- Determinar exatamente como o sistema falha;
- Testar o funcionamento fora do uso padrão;
- Confirmar se as especificações pretendidas estão sendo cumpridas, ou seja, o sistema se mantém íntegro até o limite especificado.

Os testes de estresse, geralmente, requerem colocar o hardware sob níveis exagerados de uso, a fim de garantir a estabilidade. Estes podem incluir extremos de carga de trabalho, uso de memória, carga térmica (calor), velocidade do clock ou tensões. Memória e CPU são dois componentes que são comumente testadas desta forma.

Em testes de software, **um teste de estresse do sistema refere-se a testes que colocam maior ênfase na robustez, disponibilidade e manipulação de erro sob uma carga pesada**, e não naquilo que seria considerado comportamento correto em circunstâncias normais. Em particular, os objetivos de tais testes podem ser o de garantir que o software não falhe em condições de recursos computacionais insuficientes, tais como, memória ou espaço em disco.

2.1.4 - Teste de Volume

Esse tipo de teste consiste em oferecer ao sistema uma certa quantidade de dados para verificar o seu comportamento. A quantidade e natureza dos dados depende do tipo de dado que a aplicação irá manusear, como exemplos:

- Para aplicações que manuseiam arquivos textos: submeter arquivos de tamanho incremental e verificar o desempenho do sistema;
- Para aplicações que manuseiam bases de dados: aumentar o número de registros das tabelas do BD e verificar o desempenho do sistema;
- Para aplicações que recebem dados/arquivos pela rede: aumentar o tamanho dos arquivos submetidos e verificar o desempenho do sistema.

2.1.5 - Teste de Carga

É o processo de colocar a demanda em um sistema e medir a sua resposta. O teste de carga é realizado para determinar o comportamento de um sistema sob **condições normais e previsíveis de pico de carga**. Ele ajuda a identificar a capacidade máxima de operação de uma aplicação, bem como os **gargalos** e determinar **qual elemento está causando degradação**. Quando a carga colocada sobre o sistema é aumentada para além dos padrões normais de utilização, esse teste é conhecido como **teste de estresse**. Nesses casos, a carga é geralmente tão grande que as condições de erro são o resultado esperado, mas não há uma fronteira clara quando uma atividade deixa de ser um teste de carga e torna-se um teste de estresse. O termo teste de carga é muitas vezes usado como sinônimo de teste de desempenho, teste de confiabilidade e teste de volume.

O teste de carga pode ser aplicado de várias formas, de acordo com a natureza da aplicação. Mas no geral ele busca modelar o uso esperado do sistema através da simulação de vários usuários acessando o programa ao mesmo tempo, ou seja, ele se caracteriza pelo aumento da demanda.

O teste de carga é especialmente importante se o sistema ou serviço está sujeito a um acordo de nível de serviço - SLA.

Exemplo:

- Em um sistema de vendas, milhares de clientes fazem consulta ao mesmo tempo, várias vendas são feitas por minuto e verifica se o tempo de resposta está adequado e se um cliente não está comprando um produto que não existe mais no estoque, mas que ainda não foi atualizado.

2.1.6 - Teste de Desempenho

Tem por objetivo determinar como o sistema executa em termos de capacidade de resposta e estabilidade sob uma determinada carga. Também pode servir para verificar e medir a qualidade de outros atributos do sistema, tais como, escalabilidade, confiabilidade e uso de recursos.

A medida de desempenho do sistema deve trazer um diagnóstico dos componentes do sistema que prejudicam o desempenho do sistema.

É fundamental para o custo de um novo sistema que os esforços de teste de desempenho comecem no início do projeto de desenvolvimento e estenda-se pelas demais fases. Quanto mais tarde um defeito de desempenho é detectado, maior será o custo da recuperação. Geralmente o sincronismo entre desenvolvimento e testes já ocorre para testes funcionais, mas não é comum para testes de desempenho (não funcionais), devido à natureza de ponta a ponta do seu âmbito. Logo, é importante envolver a equipe de teste de desempenho o mais cedo possível, para preparar o ambiente de testes de acordo com os requisitos de desempenho do sistema.

O teste de desempenho pode ser baseado em outros tipos de teste de sistema, tais como:

- **Teste de carga:** é a forma mais simples de testes de desempenho. Um teste de carga é, geralmente, conduzido para compreender o comportamento do sistema sob uma carga previsível. Esta carga pode ser o acesso concorrente de certo número de usuários executando certas operações em um período definido. Este teste vai dar os tempos de resposta das principais operações críticas do sistema. Neste teste, banco de dados, servidores de aplicações, etc. também são monitorados, isso ajuda a identificar gargalos nos componentes da aplicação e hardwares;
- **Teste de estresse:** é usado para entender os limites máximos de capacidade do sistema. Ele é feito para determinar a robustez do sistema em termos de carga extrema e ajuda a determinar o comportamento do sistema para essas cargas;
- **Teste de pico:** é usado para determinar se o desempenho do sistema irá alterar ou falhar com mudanças bruscas na carga de trabalho. Ele é feito aumentando ou diminuindo bruscamente a carga e observando o comportamento do sistema;
- **Teste de resistência:** é usado para determinar se o sistema pode suportar a carga esperada de forma contínua. O objetivo é monitorar o desempenho para assegurar que os tempos de processamento e resposta são mantidos após longo período de atividade sustentada, ou seja, o objetivo é detectar que o sistema não aloca memória/recursos de forma contínua e, desta forma, pode se tornar mais lento ao longo do tempo por escassez desses recursos de hardware.

2.1.7 - Teste de Instalação

Busca identificar o grau de dificuldade que os usuários terão para instalar o software e configurá-lo. O processo pode envolver a instalação completa ou parcial do software, bem como a sua desinstalação. Ele busca quantificar o tempo gasto na instalação e se a documentação (manuais de usuário) contempla as particularidades que podem ocorrer no processo de instalação.

Exercícios

- 1 – Qual é o tipo de teste que busca determinar o comportamento do sistema quando utilizado em situações que estão além da sua capacidade?
- 2 – Qual é o tipo de teste que busca colocar o sistema em uso em situações extremas de memória e processador?
- 3 – Qual é o principal objetivo de um teste de estresse?
- 4 – Qual é o tipo de teste que busca checar a performance do sistema em quantidades de dados muito grande?
- 5 – Qual é o tipo de teste que possui como mecanismo aumentar a quantidade de registros da base de dados para testar o comportamento do sistema?
- 6 – Qual é o tipo de teste que busca checar o comportamento do sistema em quantidades de dados previsíveis?
- 7 – Qual é o tipo de teste que busca identificar a capacidade máxima de operação de uma aplicação?
- 8 – Qual é a diferença entre teste de carga e teste de estresse?
- 9 – Qual é o tipo de teste que busca simular o aumento da demanda do sistema?

10 – Qual é o objetivo de um teste de desempenho?

11 – Qual é o tipo de teste de sistema que é constituído por vários outros tipos de testes de sistemas?

12 – Qual é o tipo de teste de sistema que verifica a capacidade do sistema de se recuperar de uma falha?

13 – Qual é o tipo de teste de sistema que verifica a proteção dos dados do sistema?