

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:

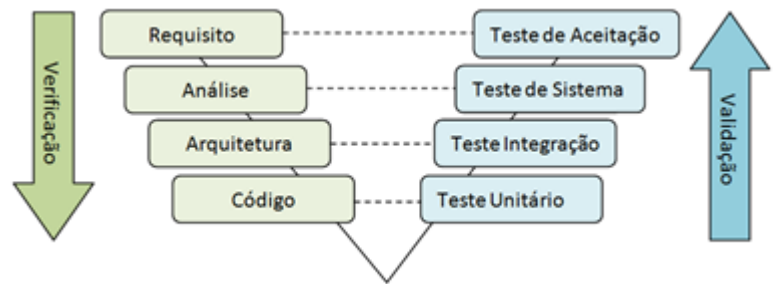


Figura 1 – Modelo V (fases do desenvolvimento x fases dos testes).

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 - Teste de Aceitação - TA

O TA é um tipo de teste de caixa-preta realizado antes da liberação do sistema para os usuários. É do tipo caixa-preta porque os testadores analisam certas funcionalidades do software sem ter acesso às suas estruturas internas.

Objetivo do TA: validar se o sistema cumpre os requisitos especificados pelo cliente no processo de desenvolvimento de software e satisfaz critérios de aceitação, que podem envolver o aval de entidades certificadoras.

Forma de aplicação:

- Geralmente, é realizado por um grupo restrito de usuários finais, num ambiente o mais próximo possível do ambiente de operação. Ele inclui todas as instalações, hardware, softwares, procedimentos e documentação destinados para a operação do software.
- O TA, como todos os casos de teste, não pode ser executado em uma única iteração de teste. Ele deve ser executado utilizando procedimentos predefinidos para dirigir os testadores sobre quais dados utilizarão, os processos passo-a-passo a seguir e o resultado esperado após a execução.

Pré-requisitos: a seguir tem-se alguns pré-requisitos que precisam ser atendidos antes de começar o TA:

- Requisitos de negócio devem estar disponíveis;
- O código da aplicação deve estar plenamente desenvolvido;
- Os testes unitários, de integração e de sistema devem ter sido concluídos;
- Todos os defeitos relatados devem estar corrigidos e testados;
- Somente erros cosméticos são aceitáveis;
- A matriz de rastreabilidade para todos os testes deve estar concluída;
- O ambiente para o teste de aceitação do usuário deve estar pronto.

O teste de aceitação na programação extrema:

- O TA é um termo usado em desenvolvimento ágil de software, particularmente Extreme Programming, referindo-se ao teste funcional de uma história do usuário pela equipe de desenvolvimento de software durante a fase de execução;
- O cliente especifica cenários de testes para uma história, onde cada história pode ter um ou vários TA, o que for preciso para garantir que as peças tenham as devidas funcionalidades. Cada TA representa algum resultado esperado do sistema;
- Os clientes são responsáveis por verificar a exatidão dos TA e rever os resultados dos testes para decidir quais falhas possuem mais prioridade. A história de usuário não é considerada completa até que ela tenha passado pelos TA. Isso significa que novos testes de aceitação devem ser criados para cada iteração ou a equipe de desenvolvimento deverá relatar o progresso zero.

2.1 - Tipos de Testes de Aceitação

Os TA podem ser realizados dentro e fora do ambiente de desenvolvimento e não envolvem apenas a aceitação dos usuários finais do sistema, mas também aspectos legais, governamentais e regras de contrato. A seguir tem-se alguns desses testes.

2.1.1 – Testes Alfa e Beta

Teste Alfa (teste de aceitação interna): **ocorre no ambiente de desenvolvimento** e, normalmente, é feito pela equipe interna - antes do produto ser liberado para testadores ou usuários externos. Ele também pode ser feito por grupos de utilizadores potenciais, mas o importante é que ele ocorra no ambiente de desenvolvimento. O feedback é então usado para corrigir problemas e falhas e melhorar a usabilidade do produto.

O Teste Beta (teste de campo): **ocorre no ambiente do cliente** e envolve o teste por um grupo de usuários em operações cotidianas do sistema. O feedback deverá levar a melhorias do produto que deverão ser corrigidas antes da sua liberação.

Os testes Alfa e Beta são executados antes do software ser liberado para o cliente.

2.1.2 - Teste de Aceitação do Usuário - TAU

Muitas vezes este teste é conhecido por "teste beta". Ele consiste em um **processo de verificar se a solução funciona para o usuário**, ou seja, garante que o usuário aceita a solução. A palavra chave no TAU é o usuário, porque são eles que irão utilizar o software diariamente.

Este teste é a última fase do processo de teste de software. Nele, os usuários reais testam o software para terem certeza que podem lidar com as tarefas em cenários reais, de acordo com as especificações.

Este teste deve ser realizado por um perito no assunto, de preferência o cliente/proprietário da aplicação em teste, e fornecer um resumo das conclusões. Os usuários realizam testes em linha com aqueles que irão ocorrer na operação do sistema.

Em desenvolvimento ágil de software) são normalmente criados por clientes empresariais e expressa em uma linguagem de domínio do negócio. Estes são testes de alto nível para verificar a integridade de uma história de usuário jogada durante qualquer Sprint/iteração.

A seguir tem-se os passos envolvidos no TAU:

- Planejamento: a estratégia do TAU é delineada durante a etapa de planejamento dos testes;
- Projetar casos de teste: os casos de teste são projetados para cobrir todos os cenários funcionais do software em uso no mundo real. Eles são projetados em uma linguagem simples e de maneira a tornar o processo de teste mais fácil para os testadores;
- Seleção de equipe de testes: a equipe de teste é composta por usuários finais do mundo real;
- Executar casos de teste e documentação: a equipe de teste executa os casos de teste projetados. Às vezes, eles também executam alguns testes aleatórios relevantes. Todos os erros são registrados em um documento de teste com comentários relevantes;
- Correção de falhas: respondendo aos bugs encontrados pela equipe de testes, a equipe de desenvolvimento de software faz ajustes finais para o código para tornar o software livre de falhas;
- Entrega: quando todos os bugs foram corrigidos, a equipe de teste indica a aceitação da aplicação. Isso mostra que o pedido preenche os requisitos dos utilizadores e está pronto para ser lançado no mercado.

2.1.3 - Testes de Aceitação do Contrato e Aceitação da Regulamentação

Em testes de aceitação do contrato, o sistema é testado contra certos critérios e especificações que foram pré-definidos e acordados em um contrato. Os critérios e especificações relevantes para a aceitação devem ser definidos quando o próprio contrato é definido e acordado.

Em testes de aceitação de regulamentação, o sistema é testado para garantir que cumpre as normas governamentais, legais e de segurança.

2.1.4 - Teste de Aceitação Operacional

Também conhecido como teste de prontidão operacional, refere-se à verificação para garantir que os processos e procedimentos estão no local para permitir que o sistema seja utilizado e mantido. Isso pode incluir procedimentos de back-up, procedimentos de recuperação de desastres, treinamento de usuários finais, procedimentos de manutenção e procedimentos de segurança.

São definidos em termos de requisitos funcionais e não funcionais cobrindo os principais atributos de qualidade de estabilidade: [funcional](#), [portabilidade](#) e [confiabilidade](#).

Exercícios

- 1 – Por que o teste de aceitação é um tipo de teste de caixa-preta?
- 2 – Os testadores precisam se basear em documentos para especificar os testes de aceitação. De onde são retirados os itens a serem testados no teste de aceitação?
- 3 – Quais são as pessoas envolvidas em um teste alfa?
- 4 – Quais são as pessoas envolvidas em um teste beta?
- 5 – Onde é realizado um teste alfa?
- 6 – Onde é realizado um teste beta?
- 7 – O teste de aceitação do usuário é considerado um tipo de teste alfa ou beta?
- 8 – Quem são as pessoas envolvidas em um Teste de Aceitação de Contrato?
- 9 – Quem são as pessoas envolvidas em um Teste de Aceitação de Regulamentação?
- 10 – Quantos testes de aceitação do usuário são necessários em um método de desenvolvimento ágil, assim como Extreme Programming?
- 11 – Por que o teste de aceitação operacional é importante?