**ATIVIDADE – TÉCNICAS DE TESTE DE SOFTWARE**

**MATHEUS HENRIQUE BUTKOSKI SILVA**

**TESTE EXPLORATÓRIO**

Uma boa e básica definição de teste exploratório seria a criação e a execução ao mesmo tempo de um teste. Isso significa que não foram realizados estudos anteriores nem mesmo planejamento para sua execução.

É por isso que neste tipo de teste a experiência e o conhecimento do testador é de grande valia, talvez a maior das habilidades esperadas, porque quando ele é realizado, normalmente o testador não tem muitas informações sobre o que vai testar, como vai testar e do próprio software, isto quer dizer que, curiosidade, criatividade, concentração, abstração, organização e experiência são fundamentais para o sucesso dos testes e as principais ferramentas do testador.

Testes exploratórios são do tipo **caixa-preta** e são extremamente úteis para encontrar bugs rapidamente, e quando o projeto não tem uma documentação bem definida ou a mesma encontra-se desatualizada.

Além de descobrir erros o teste exploratório pode servir de base para a criação de testes estruturados convencionais. Ele ganha importância quando, por exemplo, se deseja reproduzir um defeito aleatório não coberto no planejamento inicial ou quando se deseja obter o nível inicial de qualidade de determinada aplicação.

**TESTE FUNCIONAL**

Os testes funcionais permitem que os testes ocorram de uma forma mais eficiente e rápida, possibilitando encontrar as não conformidades do software em relação aos requisitos do sistema.

Geralmente os testes funcionais são aplicados nas fases de teste de Integração, Sistema e Aceitação.

O teste funcional considera o comportamento externo do software (teste de **caixa-preta**).

**TESTE UNITÁRIO**

O teste unitário consiste em verificar o comportamento das menores unidades em sua aplicação, que é focado em partes isoladas de um sistema.

No teste unitário, cada parte do sistema ganha uma atenção devida e detalhada, de modo a otimizar o processo de identificação de erros. O objetivo é ajudar a rastrear os bugs e impedir que eles retornem depois que alterações forem feitas no produto.

Testes unitários devem possuir abordagem automática e geralmente utilizam um framework de testes como o JUnit.

É um teste do tipo **caixa-branca**

**TESTE DE USABILIDADE**

Teste de usabilidade é um teste do tipo **caixa-preta**, que verifica a facilidade que o software possui, de ser claramente compreendido e manipulado pelo usuário.

Seu objetivo é observar usuários reais usando o produto para descobrir problemas e pontos de melhorias.

Em um teste de usabilidade de software são analisados fatores como facilidade na utilização, de memorização dos atalhos e dos comandos, de localização das pastas e diretórios, capacidade autoexplicativa do sistema, entre outros. Tudo isso ajuda a prevenir erros e a indicar soluções para que a interface seja limpa, intuitiva e agradável de usar.

**TESTE DE ESTRESSE**

Teste de stress é realizado para submeter o software a situações extremas. Basicamente, o teste de stress baseia-se em testar os limites do software e avaliar seu comportamento. Assim, avalia-se até quando o software pode ser exigido e quais as falhas (se existirem) decorrentes do teste. Os testes de stress são fundamentais em aplicações em que a eficiência seja uma característica importante. Por exemplo:

**-Servidores de arquivos e servidores web**, que devem atender a solicitações de um grande número de clientes.

**-Aplicações industriais**, tais como o controle de uma refinaria de petróleo.

**-Jogos de computador**, que precisam de um desempenho aceitável para serem viáveis comercialmente.

É um teste do tipo **caixa-branca.**

A grande dificuldade de realizar um teste de stress é configurar adequadamente a plataforma de execução. Por exemplo: se o avaliador está interessado em saber a quantidade mínima de memória disponível para que um programa funcione, retirar fisicamente chips de RAM é uma solução muito trabalhosa. Além de RAM, há diversos outros parâmetros, como disco, CPU e instalações de rede, e seria inviável montar e desmontar um computador com diferentes configurações para todos os testes. Para isso, utilizam-se ferramentas de stress.

Um exemplo de ferramenta que pode ser utilizada para um teste de stress é o **WinStress**, da **Ultra-X**. Trata-se de um programa que permite reduzir artificialmente o desempenho de um computador, de acordo com a configuração desejada pelo avaliador. É possível variar parâmetros como carga de CPU, memória disponível, espaço em disco disponível e carga de rede.

**TESTE DE CARGA**

Tem como objetivo verificar o desempenho de um sistema quando este é submetido a cargas variáveis de usuários ou transações.

Podemos executar um teste de carga em uma aplicação web simples, partindo de 100 usuários simultâneos e adicionar novos usuários a cada 5 minutos, totalizando 1000 usuários ao final do teste.

Dessa maneira, é possível analisar como todo o ecossistema da aplicação se comporta nesse período, se houve lentidão em algum momento do teste, se o uso CPU e memória dos servidores ficou dentro do limite estabelecido, dentre outros indicadores de desempenho.

É um teste do tipo **caixa-branca.**

O teste de carga é parecido com o teste de estresse, porém possui uma diferença primordial. Enquanto o **teste de carga** pretende descobrir qual a capacidade de uso, processamento e acessos de um sistema, o **teste de stress** pretende descobrir em que ponto ele já não aguenta mais funcionar.

**TESTE DE SEGURANÇA**

O Teste de Segurança tem como meta garantir que o funcionamento da aplicação esteja exatamente como especificado. Verifica também se o software se comporta adequadamente mediante as mais diversas tentativas ilegais de acesso, visando possíveis vulnerabilidades. Para isso, testa se todos os mecanismos de proteção embutidos na aplicação de fato a protegerão de acessos indevidos.

É muito comum que as aplicações se tornem alvo de sujeitos que buscam provocar ações que possam prejudicar ou, até mesmo, beneficiar pessoas. Em função de situações como estas, o Teste de Segurança propõe demonstrar se a aplicação faz exatamente o que deve fazer ou se a aplicação não faz o que não deve ser feito.

A execução do Teste de Segurança possibilita que dúvidas sobre prováveis vulnerabilidades do software sejam sanadas. Pode auxiliar também na definição de um plano de contingência, visando determinar qual precaução será tomada contra os possíveis ataques.

É um teste do tipo **caixa-branca.**

Existem diversas ferramentas que dão suporte quando o assunto é testar a segurança de uma aplicação. Entre elas:

Metasploit, Nessus, Nmap, BurpSuite, SqlMap, Nikto, etc…

**ALFA E BETA TESTE**

Último estágio do processo de testes, onde o produto é validado em relação aos requisitos do usuário final. Este passo final ajuda o time a decidir se o produto está pronto para ser lançado ou não.

O objetivo do teste de aceite é estabelecer a confiança no sistema, de uma parte do sistema ou uma característica específica do sistema.

O alfa e o beta test são teste de aceite do tipo **caixa-cinza.**

O **Alfa Teste** usa o sistema de forma não planejada disponibilizando o sistema para um pequeno grupo de pessoas. Essas pessoas participam do projeto dando feedbacks sobre a situação atual em que o sistema se encontra.

O **Alfa Teste** é frequentemente usado para sistemas de prateleira como forma de teste de aceitação interna. O grupo que participa dos testes geralmente é composto por membros da organização e também do cliente.

Esse teste visa a identificação de possíveis erros não identificados sendo analisados por uma quantidade de usuários que desfrutam do sistema. O time de programadores acompanha os testes para coletar dados de falhas a serem corrigidas e melhorias a serem aplicadas.

Já o **Beta Test** é utilizado por um grande número de pessoas e a execução do sistema ocorre de forma não planejada por pessoas desconhecidas, que não possuem nenhuma relação com a equipe ou empresa desenvolvedora.

Frequentemente o teste é utilizado como uma forma de aceitação externa para softwares de prateleira possibilitando avaliar o feedback do mercado.

Os programadores não farão acompanhamento do sistema porque nesse teste são os usuários que reportam os erros encontrados.

Existem hoje no mercado, grandes empresas com aplicativos em Beta para fazer download, informando ao usuário que o sistema em questão está em fase de teste, não garantindo 100% de eficiência. Essa prática garante feedbacks valiosos para o resultado final.

**BIBLIOGRAFIA**

<https://www.devmedia.com.br/testes-exploratorios-teoria-e-pratica/22791>

<https://xtesting.com.br/teste-exploratorio/>

<https://pt.stackoverflow.com/questions/216571/o-que-%C3%A9-um-teste-de-stress>

<https://pt.stackoverflow.com/questions/248318/qual-%C3%A9-a-diferen%C3%A7a-entre-um-teste-de-carga-e-um-teste-de-stress>

<https://medium.com/@ingrid.carvalho.mo/quais-s%C3%A3o-as-fun%C3%A7%C3%B5es-do-teste-alfa-beta-e-regress%C3%A3o-b0db1c3bf5d0>

SLIDES