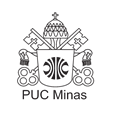
**Marco Willy Azevedo Gomes**

**Professora: Simone**

Engenharia de Software – Noite – Praça da Liberdade

**1 - Defina teste caixa branca e teste caixa preta.**

**Teste de caixa preta**: O testador deve realiza-lo utilizando a experiência do usuário, ou seja, através da interface do produto, o mesmo não precisa ter conhecimento tecnico, precisa de conhecimento do negocio.

**Teste de caixa branca**: O testador deve possuir conhecimento tecnico e acesso à estrutura interna da aplicação, seu foco é garantir que os componentes do software estejam concisos.

**2 - Quais são os tipos de teste? Explique cada um deles.**

**1. Teste funcional**

Essa técnica testa os requisitos funcionais, as funções e os casos de uso (User Stories). É possível realizar testes funcionais manualmente, sem ferramentas, ou com o auxílio de ferramentas e frameworks. Um dos frameworks mais utilizados é o Selenium WebDriver.

**2. Teste regressivo**

Tem como objetivo rodar testes de funcionalidades que já funcionavam anteriormente, relacionando-as com as atualizações realizadas no software. A ideia é garantir que o que já funcionava continua funcionando após alguma alteração e confirmar que as mudanças não causaram nenhum impacto.

**3. Teste exploratório**

Esse tipo de teste de software exige experiência do testador / analista de testes. É a técnica que busca empregar todo o conhecimento prévio do testador em busca de novas falhas.

**4. Teste de segurança**

O teste de segurança avalia se o sistema e os dados são acessados de maneira segura e apenas pelo autor das ações. O objetivo é evitar que algum fraudador intercepte a informação que está sendo trafegada ou que colete dados sensíveis.

**5. Teste de performance**

O teste de performance avalia o software sob as condições normais de uso. Alguns exemplos das análises realizadas: tempo de resposta, número de transações por minuto, usuários simultâneos, etc.

**6. Teste de integração**

Esse teste avalia se um ou mais componentes combinados funcionam de maneira satisfatória, trafegando corretamente as informações. Há quem diga que o teste de integração é composto por vários testes de unidade.

**7. Teste unitário**

Esse tipo de teste de software analise um componente isolado ou classe do sistema. Na maioria das vezes, esse teste fica a cargo do desenvolvedor que está trabalhando no código, mas nada impede que um profissional de Garantia de Qualidade (Quality Assurance – QA) possa fazê-lo.

**8. Teste de usabilidade**

Essa técnica avalia a experiência do usuário, a consistência da interface, o layout, se as funcionalidades estão acessíveis, entre outras variáveis. Ao garantir uma boa usabilidade e uma experiência fluida de navegação, temos grandes chances de fidelizar o usuário na nossa plataforma, pois teremos pessoas satisfeitas com a solução desenvolvida.

**9. Teste de aceitação**

O teste de aceitação é realizado antes de sua disponibilização ao usuário. É o teste que tem como objetivo verificar se os requisitos foram implementados e se as necessidades do usuário estão sendo atendidas.

**10. Teste de acessibilidade**

Por fim, uma técnica bem delicada e que poucos sistemas passam por ela. O teste de acessibilidade verifica se o sistema está adaptado para utilização de usuários com deficiência visual parcial ou completa.

**3 - Qual a diferença entre validação e verificação?**

**Validação:** Esta atividade se resume em responder a esta pergunta. A verificação tem o objetivo de avaliar se o que foi planejado realmente foi realizado. Ou seja, se os requisitos e funcionalidades documentados foram implementados, além disso a verificação também pode ser realizada para especificação de sistemas, para avaliar se os requisitos estão sendo documentados como deveriam e ainda prever falhas ou inconsistências entre requisitos.

**Verificação:** A validação tem o objetivo de avaliar se o que foi entregue atende as expectativas do cliente. Ou seja, se os requisitos, independente do que foi planejado, estão sendo implementados para atender a regra de negócio do cliente, se o sistema é realmente aquilo que o cliente quer e está pagando para ter. A validação final do sistema é realizada pelo próprio cliente ou usuário.

**4 - O que deve conter em um caso de teste?**

Para a elaboração de um caso de teste deve ser informado um título, objetivo detalhado, pré-condição para que a execução ocorra conforme definida, os passos a serem seguidos e o resultado esperado que é a resposta do sistema para as ações executadas.

**5 - O que os testes podem mostrar?**

Os testes tem podem mostrar se as funcionalidades e requisitos solicitados pelo cliente foram atendidos.

**6 - É correto afirmar que os testes garantem ausência de erros? Por quê?**

Não, porque pode acontecer algo não pensado que impacta o resultado final.

**7 - Para os problemas abaixo, preencha a tabela construindo casos de testes funcionais (testes caixa preta) considerando as classes de equivalência.**

* 1. Um programa calcula a área de um triângulo. As entradas são três conjuntos de coordenadas x,y.

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **CLASSE DE EQUIVALÊNCIA - VÁLIDA** | **CLASSE DE EQUIVALÊNCIA - INVÁLIDA** |
| 3 coordenadas (x,y) | Três conjuntos de coordenadas válidas  Caso de teste1: (1,1)(1,1)(1,1)  Valor da Área esperada1:\_\_\_\_\_3\_\_\_\_  Caso de teste2: (2,2)(2,2)(2,2)  Valor da Área esperada2:\_\_\_\_\_6\_\_\_\_  Caso de teste3: (3,3)(3,3)(3,3)  Valor da Área esperada3:\_\_\_\_9\_\_\_\_\_ | Três conjuntos de  Coordenadas inválidas  Caso de teste4: (\_0\_,\_0\_)(-1,-8)(-12,7)  Valor da Área esperada4:\_\_\_\_\_Erro\_\_\_\_  Caso de teste4: (\_0\_,\_0\_)(-1,-8)(-16,7)  Valor da Área esperada4:\_\_\_\_\_Erro\_\_\_\_ |

* 1. Um programa calcula o número de minutos existentes entre duas horas (em formato 12 horas) que são fornecidas como entrada.

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **CLASSE DE EQUIVALÊNCIA - VÁLIDA** | **CLASSE DE EQUIVALÊNCIA - INVÁLIDA** |
| Dois valores de  horário no  formato  Hr:min AM ou PM | Dois valores de horas  Válidas  Caso de teste1:  Entrada da 1ª hora: 12:01 PM  Entrada da 2ª hora: 01:01 PM  Saída esperada: 60min  Caso de teste2: Entrada da 1ª hora: 12:05 AM  Entrada da 2ª hora: 04:05 AM  Saída esperada: 240 min  Caso de teste3:  Entrada da 1ª hora: 01:00 PM  Entrada da 2ª hora: 06:00 PM  Saída esperada: 300 min | Dois valores de  horas inválidas  Caso de teste4:  Entrada da 1ª hora: 05:10 PM  Entrada da 2ª hora: 10:00 PM  Saída esperada: 290 min |

**8 - O que é depuração?**

O debug (ou depuração) é uma das atividades que ajuda a equipe de desenvolvimento a identificar problemas no código-fonte de um aplicativo. Ele pode ser executado durante a criação de sistemas web, aplicativos mobile ou mesmo softwares para notebooks. Em todos os casos, essa atividade reduz as chances de uma pessoa usuária tentar executar alguma atividade e não conseguir em função de um erro que poderia ser solucionado no processo de desenvolvimento.

**9 - Qual a principal diferença entre bibliotecas e frameworks?**

**Biblioteca:** Uma biblioteca se refere a uma coleção de pacotes que fornece funções. Seu objetivo é oferecer um conjunto de funcionalidades prontas para para uso sem se preocupar com outros pacotes.

**Framework:** Framework é um conjunto de bibliotecas. Um framework não oferece apenas funcionalidades, mas também uma arquitetura para o trabalho de desenvolvimento. Em outras palavras, você não inclui uma estrutura, você integra seu código a ele. Um framework é a estrutura de arame de um projeto.

**10 - Quais as vantagens dos componentes sobre as bibliotecas?**

Como dizem alguns programadores, “porque reinventar a roda”. Os componentes estão prontos é so usar... não é necessario ficar recriando.

**11 - O que é uma máquina virtual?**

Uma máquina virtual é um programa que age como um computador completo, com todas as instâncias de um sistema operacional funcional. Em resumo, é um computador funcionando dentro de outro. Utiliza um programa cliente, que age como uma camada de virtualização entre o sistema de um computador e outro que você quer instalar.

**12 - O que é um container de software?**

Container é, basicamente, você pegar tudo o que sua aplicação precisa para ‘viver’ e existir e jogar em um único arquivo. Dessa forma, você garante que a aplicação vai rodar, não importa a versão da tecnologia que você está usando.

**13 - Como os componentes são distribuídos e implantados?**

Em aplicações mobile os componentes são distribuidos e implantados via lojas (App Store e Google Play), em aplicação web são implantados em servidores remotos e distribuidos via dns ou endereço web, em aplicações desktop temos duas formas lojas ou download direto.

**14 - Qual a diferença entre software aplicativo e software de sistema ou software básico?**

**Software de aplicativo:** é um software que permite ao utilizador realizar uma tarefa especifica.

**Software de Sistema:** são programas de computador de baixo nível que interagem com o computador num nível muito básico.

**15 - Cite um exemplo de um software aplicativo e de um software de sistema.**

**Software de aplicativo:** Microsoft Office, Internet Explorer, Adobe Photoshop, navegadores, etc.

**Software de Sistema:** Sistema Operacional, o firmware, controladores de dispositivos e a interface gráfica que permite ao utilizador interagir com o computador.

**16 - Qual a diferença entre software embutido e software que executa em um computador genérico.**

**Software embutido**: é um software de computador , escrito para controlar máquinas ou dispositivos que normalmente não são considerados computadores, comumente conhecidos como sistemas embutidos . Normalmente é especializado para o hardware específico em que é executado e tem restrições de tempo e memória. Este termo às vezes é usado alternadamente com firmware .

**Software de Computador Generico:** é um software instalado em um sistema operacional que tem um determinado fim.

**17 - Posso escrever software embutido e software que executa em um computador generico com a mesma linguagem de programação?**

Depende, se a linguagem usada no desenvolvimento tem a capacidade de converter o que foi escrito em liguagem de maquina, acredito que não havera problemas.

**18 - Qual a diferença entre software compilado e software interpretado?**

**Software compilado:** So havera uma versão do sistema disponivel para uso após compilação com sucesso.

**Software interpretado:** Havera uma versão do sistema após execução com sucesso.

**19 - Posso afirmar que todo software nativo também possui uma versão web?**

Não, pois esse software sendo mobile ou desktop, pode não dispor de uma api(back-end) para seu funcionamento remoto.

**20 - O que é software distribuído? Cite um exemplo de software distribuído.**

Os sistemas distribuídos estão em todo o lugar, ou melhor, acessíveis a partir de qualquer lugar. Segundo Tanenbaum, um sistema distribuído é um conjunto de computadores independentes entre si (e até diferentes), ligados através de uma rede de dados, que se apresentam aos utilizadores como um sistema único e coerente.

**Exemplo:** Sistemas de pesquisas(Google, Bing, Yahoo, etc...)