**QA – Exemplo Prático – Arquivei.com.br**

**1. Metodologia Ágil Scrum**

• O time de QA trabalha junto com o desenvolvimento seguindo a metodologia ágil Scrum;

• A cada 15 dias, há uma reunião de planejamento para decidir novas funcionalidades e estimar o tempo de execução de cada uma;

• As funcionalidades devem ser desenvolvidas nas duas próximas semanas antes da próxima reunião de planejamento.

**2. Escrita de Cenários de Teste**

• Após a reunião de planejamento, o time de QA começa a escrever os cenários de teste utilizando a metodologia BDD e a linguagem Gherkin;

• A linguagem Gherkin é de fácil entendimento tanto para a área de negócio quanto para o time de QA, e permite a portabilidade dos cenários para a automação de testes;

• Para ler os cenários de teste, utiliza-se Python e Selenium, que é um framework bastante utilizado para a automação de testes.

**3. Testes Realizados**

• Quando uma nova funcionalidade é entregue para o time de QA testar, realiza-se primeiro o teste de sanidade automatizado, para garantir que as funcionalidades anteriores não foram afetadas;

• As novas funcionalidades são testadas manualmente, para garantir que elas funcionem como deveriam e não apresentem comportamentos indesejados;

• Além dos testes citados anteriormente, existem três tipos base para a realização dos testes: teste de mutação, testes estruturais e testes funcionais.

**4. Teste de Mutação**

• O teste de mutação é utilizado para validar os testes no código;

• Ele realiza mutações nos testes e, se os testes mutantes passarem nos testes estipulados pela área de QA, significa que o mutante está vivo, ou seja, os testes estão frágeis;

• Se o mutante está morto, significa que o mutante falhou, não foi validado, o que é o resultado esperado para os testes de QA.

**5. Testes Estruturais**

• Os testes estruturais são testes de caixa branca que testam a parte interna do sistema, ou seja, o código-fonte;

• Entre os principais testes estruturais, destacam-se os testes de stress, de carga, de unidade, de comportamento e de integração.

**6. Testes Funcionais**

• Os testes funcionais são testes de caixa preta que utilizam a parte visual do sistema, ou seja, a parte que o usuário utiliza;

• Eles permitem entender como a aplicação está se comportando e como deveria se comportar com base nas especificações e requisitos dados;

• Entre os tipos de testes funcionais destacam-se os testes de regressão, de fumaça, de aceite e os testes automatizados.

**Metodologia ágil:**

A metodologia ágil é uma abordagem para o desenvolvimento de software que enfatiza a entrega de funcionalidades de maneira incremental e interativa.

É baseada em valores e princípios, tais como colaboração, adaptação a mudanças e entrega contínua de valor ao cliente.

É composta por várias metodologias, como Scrum, Kanban, XP, entre outras.

**Scrum**:

O Scrum é uma das metodologias ágeis mais populares e é usado para gerenciar e planejar projetos de desenvolvimento de software.

É composto por três papéis: o Product Owner, o Scrum Master e o Time de Desenvolvimento.

Divide o trabalho em sprints, que são objetivos medidos em tempo, em que as equipes se comprometem a entregar um conjunto de funcionalidades.

Possui diversas cerimônias, como reunião de planejamento, revisão da sprint, retrospectiva e daily stand-up.

**BDD:**

BDD (Behavior-Driven Development) é uma técnica de desenvolvimento de software que visa integrar as perspectivas de negócios e de desenvolvimento de software.

É uma abordagem centrada no comportamento, onde as funcionalidades são escritas em termos de comportamento esperado.

Utiliza uma linguagem de alto nível, como Gherkin, para definir esses comportamentos.

É uma técnica que busca trazer mais clareza e comunicação entre os membros da equipe e os stakeholders.

**Linguagem Gherkin:**

Gherkin é uma linguagem de especificação de funcionalidades que utiliza uma sintaxe simples e legível por humanos normais.

É a linguagem padrão para especificação de cenários de teste no BDD.

Permite que as especificações sejam escritas em termos de comportamento esperado, facilitando a comunicação entre membros da equipe e stakeholders.

**Python:**

Python é uma linguagem de programação de alto nível, interpretada e orientada a objetos.

É uma das linguagens mais populares para desenvolvimento de software atualmente.

Possui uma grande variedade de bibliotecas e frameworks, incluindo o Selenium.

**Selenium:**

Selenium é um conjunto de ferramentas de automação de teste para web.

É um dos frameworks mais utilizados para teste de automação, permitindo a simulação de interações do usuário com o navegador.

Pode ser utilizado em conjunto com outras ferramentas, como o Python, para escrever scripts de teste automatizados.

**Teste de mutação:**

O teste de mutação é uma técnica que envolve a criação de mutantes, ou seja, versões modificadas do código original, com o objetivo de testar a eficácia dos testes automatizados.

Consiste em executar os testes automatizados em cada mutante criado para determinar se ele é capaz de detectar as mudanças feitas no código.

Se um teste automatizado não conseguir detectar um mutante, ele é considerado um teste falho.

**Teste estrutural:**

Os testes estruturais são testes que avaliam a estrutura interna do software, incluindo seu código-fonte.

São geralmente testes de ***caixa branca***, ou seja, o testador tem acesso à estrutura interna do software.

Incluem testes de stress, de carga, de unidade, de comportamento e de integração.

**Teste Funcional:**

O teste funcional é uma abordagem para testar se uma aplicação atende aos requisitos do cliente e se está apresentando o comportamento esperado. Ele é realizado a partir da perspectiva do usuário final e pode ser feito manualmente ou de forma automatizada.

**Teste Funcional de Regressão:**

O teste funcional de regressão é usado para garantir que as alterações no software não afetem as funcionalidades existentes. Ele é realizado a partir da execução de testes que já foram validados anteriormente, a fim de verificar se a aplicação ainda funciona conforme esperado.

**Teste de Fumaça:**

O teste de fumaça é usado para garantir que as principais funcionalidades da aplicação ainda estejam funcionando corretamente após uma alteração. Ele é executado após uma atualização de software ou configuração, visando detectar falhas importantes e críticas o mais rápido possível.

**Teste de Aceite:**

O teste de aceite é uma validação do cliente, que verifica se a aplicação atende aos requisitos definidos. Ele é baseado nos critérios estabelecidos pelo cliente, que determinam se a aplicação está pronta para ser liberada ou não.

**Teste Automatizado:**

O teste automatizado é um processo de execução de testes de forma automatizada, usando ferramentas e scripts de teste. Ele é amplamente utilizado para testes funcionais, testes de regressão e testes de carga, a fim de economizar tempo e aumentar a eficiência dos testes.

**Metodologia Kanban:**

A metodologia Kanban é um método ágil de gerenciamento de projetos e fluxos de trabalho.

A ideia central do Kanban é visualizar o fluxo de trabalho em um quadro ou painel.

O Kanban possui algumas regras básicas, como limitar o trabalho em progresso (WIP), fazer com que as equipes trabalhem em colaboração e sempre buscar melhorias contínuas.

O método Kanban é dividido em quatro princípios: visualizar o trabalho, limitar o trabalho em progresso, gerenciar o fluxo e tornar as políticas de processo explícitas.

A utilização do Kanban permite uma visão geral do processo de trabalho, facilita a identificação de gargalos e permite que a equipe ajuste as atividades para melhorar a eficiência.

A metodologia Kanban é bastante flexível e pode ser adaptada para diferentes tipos de projetos e equipes, podendo ser utilizada em empresas de diversos segmentos.

**Metodologia Scrum:**

A metodologia Scrum é uma abordagem ágil para gerenciamento de projetos de software.

O Scrum é baseado em um framework que define papéis, reuniões e artefatos.

Os papéis no Scrum são o Product Owner, o Scrum Master e a equipe de desenvolvimento.

As reuniões incluem o Sprint Planning, a Daily Scrum, a Sprint Review e a Sprint Retrospective.

Os artefatos no Scrum incluem o Product Backlog, o Sprint Backlog e o Incremento.

O Scrum é baseado em ciclos curtos de desenvolvimento chamados Sprints.

O objetivo do Scrum é fornecer um processo iterativo e incremental para entrega de software de qualidade com maior eficiência e eficácia.

**Teste de Regressão:**

O teste de regressão é um tipo de teste que verifica se as mudanças feitas em uma aplicação não afetam as funcionalidades já existentes.

O teste de regressão é executado após a implementação de novas funcionalidades, correção de bugs, atualizações de software ou hardware, etc.

O objetivo do teste de regressão é garantir que as mudanças realizadas não tenham efeitos colaterais em outras áreas da aplicação.

O teste de regressão pode ser feito manualmente ou de forma automatizada.

O teste de regressão deve ser executado periodicamente para garantir que as funcionalidades já existentes continuem a funcionar corretamente após as atualizações.

**Teste de Fumaça:**

O teste de fumaça é um tipo de teste inicial que verifica se a aplicação pode ser instalada e iniciada com sucesso.

O teste de fumaça é realizado para garantir que a aplicação tenha um mínimo de funcionalidades e que não haja erros graves.

O objetivo do teste de fumaça é identificar problemas no início do processo de teste, evitando que erros simples causem atrasos no projeto.

O teste de fumaça pode ser feito manualmente ou de forma automatizada.

O teste de fumaça deve ser executado sempre que houver uma nova build da aplicação.

**Teste de Aceite:**

É um tipo de teste que avalia se a aplicação atende aos requisitos do cliente e às expectativas do usuário final;

Geralmente, é executado após os testes de integração e de sistema, e antes do lançamento oficial da aplicação;

Pode ser realizado manualmente ou por meio de automação de testes;

O objetivo principal é validar se a aplicação está pronta para ser entregue ao cliente;

O teste de aceite é importante para garantir a qualidade da aplicação, a satisfação do cliente e a redução do risco de problemas após o lançamento da aplicação;

Geralmente, é realizado pelo próprio cliente ou por representantes dele.

**Metodologia Kanban:**

É uma metodologia ágil de gestão de projetos que se baseia no uso de quadros visuais (kanban boards) para organizar o trabalho;

O kanban board geralmente é composto por colunas que representam as diferentes etapas do fluxo de trabalho (por exemplo, "A fazer", "Em progresso", "Aguardando revisão", "Concluído");

Cada item de trabalho (geralmente uma tarefa ou uma história de usuário) é representado por um cartão ou um post-it que é movido pelas diferentes colunas do kanban board à medida que o trabalho avança;

O objetivo do kanban é otimizar o fluxo de trabalho, minimizar os gargalos e maximizar a eficiência;

O kanban é amplamente utilizado em equipes de desenvolvimento de software, mas pode ser aplicado em diversas outras áreas.

**Metodologia Scrum:**

É uma metodologia ágil de gestão de projetos que se baseia no trabalho em equipe, na colaboração e na entrega iterativa e incremental de valor;

A equipe do Scrum é composta por um Scrum Master, um Product Owner e a equipe de desenvolvimento;

O Scrum é organizado em sprints (ciclos de trabalho) que duram geralmente de uma a quatro semanas;

Durante cada sprint, a equipe se reúne diariamente (daily stand-up) para avaliar o progresso, identificar impedimentos e planejar as atividades do dia seguinte;

No final de cada sprint, a equipe apresenta o resultado do trabalho ao Product Owner e ao cliente em uma reunião chamada de sprint review;

O objetivo do Scrum é entregar valor de forma iterativa e incremental, garantindo a qualidade do produto, a satisfação do cliente e a motivação da equipe.

**O que é o teste de aceite:**

O teste de aceite, também conhecido como teste de aceitação ou teste de validação, é uma técnica de teste de software que avalia se o software desenvolvido atende aos requisitos definidos pelo usuário ou cliente. Ele é executado para validar se o software está pronto para ser entregue ou implantado em produção.

**Como o teste de aceite é realizado:**

O teste de aceite é geralmente realizado em um ambiente controlado que simula o ambiente de produção. É conduzido pelo cliente ou usuário final e visa validar se o software cumpre as expectativas do cliente. Os testes podem ser realizados manualmente ou com automação de testes, e os resultados são comparados com os requisitos definidos.

**Tipos de teste de aceite:**

Existem diferentes tipos de teste de aceite, como o teste alfa e o teste beta. O teste alfa é realizado internamente na organização, com usuários selecionados para testar o software em um ambiente controlado. Já o teste beta é realizado com um grupo mais amplo de usuários em um ambiente de produção simulado ou real.

**Benefícios do teste de aceite:**

O teste de aceite ajuda a garantir que o software entregue atenda aos requisitos do cliente, aumentando a satisfação do cliente e melhorando a reputação da empresa. Além disso, o teste de aceite pode ajudar a reduzir os custos de suporte e manutenção, pois erros e problemas são identificados e corrigidos antes da implantação em produção.

**Ferramentas de teste de aceite:**

Existem várias ferramentas de teste de aceite disponíveis, desde ferramentas para gerenciar casos de teste e rastrear defeitos até ferramentas de automação de testes. Algumas das ferramentas mais populares incluem TestRail, Zephyr, JIRA, Selenium e Appium.

**Boas práticas para o teste de aceite:**

Algumas boas práticas para o teste de aceite incluem envolver o cliente ou usuário final desde o início do projeto, definir critérios claros de aceitação, documentar os resultados dos testes e fornecer feedback ao time de desenvolvimento para correção de problemas identificados. Além disso, é importante planejar adequadamente o teste de aceite para garantir que ele seja realizado de maneira eficiente e eficaz.

**O que é o teste de fumaça?**

O teste de fumaça, também conhecido como smoke test, é um tipo de teste funcional que tem como objetivo verificar se as principais funcionalidades do software estão funcionando corretamente logo após a realização de alguma atualização ou mudança.

**Como é realizado o teste de fumaça?**

O teste de fumaça geralmente é executado de forma automatizada, com a criação de um conjunto de testes que verificam as funcionalidades mais importantes do software, como login, cadastro de usuário, visualização de dados, etc. Esses testes são executados de forma rápida e superficial, sem verificar detalhes e funcionalidades menos importantes.

**Para que serve o teste de fumaça?**

O teste de fumaça é utilizado para garantir que as principais funcionalidades do software não foram afetadas após a realização de uma atualização ou mudança, evitando problemas maiores e garantindo que o software continue funcionando de forma adequada.

**Quais são as vantagens do teste de fumaça?**

As principais vantagens do teste de fumaça são a sua rapidez e facilidade de execução, além de possibilitar a identificação rápida de problemas em funcionalidades críticas do software.

**Qual é a diferença entre o teste de fumaça e os demais testes funcionais?**

Enquanto os demais testes funcionais verificam detalhadamente todas as funcionalidades do software, o teste de fumaça tem um foco maior nas funcionalidades mais importantes e é realizado de forma rápida e superficial. Além disso, o teste de fumaça é geralmente executado logo após a realização de uma atualização ou mudança, enquanto os demais testes funcionais são realizados de forma mais ampla e constante ao longo do ciclo de vida do software.

**Conclusão:**

A metodologia ágil Scrum e as ferramentas como BDD, Gherkin, Python e Selenium ajudam as equipes de QA a realizar testes de software de forma eficiente e eficaz. Além disso, os testes de mutação, testes estruturais e testes funcionais são vitais para garantir que o software funcione corretamente e atenda às expectativas do cliente. É importante escolher o tipo de teste adequado para cada situação, a fim de garantir que o software seja entregue com qualidade e sem falhas críticas.