

O que é QA?

De forma prática, a Quality Assurance (QA), ou garantia de qualidade, em português, é um processo no desenvolvimento. Ele tenta garantir que o produto seja entregue respeitando as qualidades pretendidas pelo cliente. Isso evita que o usuário receba um 'item' com problemas e erros em sua execução.

Para isso, o profissional da área deve realizar uma série de testes com foco no processo de desenvolvimento. Isso é fundamental para que no final, o produto não chegue nas mãos dos usuários com uma série de erros.

O Analista de Qualidade é o profissional responsável por monitorar cada fase do desenvolvimento de um software, de modo a garantir que os resultados esperados sejam cumpridos. Fica sob a responsabilidade do analista de qualidade criar planos de testes, rastrear bugs, desenvolver padrões de qualidade e identificar potenciais problemas para o usuário.

Qual a importância do Quality Assurance?

Hoje em dia, as pessoas sentem necessidade de medidas urgentes para resolverem seus problemas. Quanto mais um produto ou serviço demora para mostrar resultado, maiores serão as possibilidades das pessoas imediatamente o trocarem pela concorrência. Por isso, a primeira oportunidade que um cliente lhe dar pode ser também a única.

Para comprovar isso, as estatísticas da Compuware, empresa americana de softwares, mostram que 16% dos usuários utilizam novamente um aplicativo que não causou uma primeira boa impressão. 44% dos usuários são mais impacientes e já deletam o aplicativo de imediato, assim que ele não realizou a ação desejada.

Isso é prejudicial não só por perder um cliente, mas também por ter a reputação da marca manchada por um app de má qualidade. Os prejuízos que isso acarreta podem ser irreparáveis. Sendo assim, o quality assurance entra para prevenir esses resultados negativos.

Pensando nisso, todo analista de qualidade deve focar em pelo menos **10 Soft Skills** para se tornar um excelente QA.

Boa comunicação

Não é nenhum segredo que habilidades pessoais são essenciais para se ter e desenvolver. Além disso, são vitais para o sucesso de qualquer projeto. Graças ao progresso tecnológico, hoje temos a capacidade de trabalhar com profissionais de todo o mundo.

Embora muitos de seus colegas de equipe estejam localizados remotamente, você ainda pode se comunicar com eles por meio de bate-papo online, telefone, vídeo chamadas, aplicativos de mensagens instantâneas, e-mails ou redes sociais.

Apesar de uma ampla gama de opções de comunicação disponíveis, ele não melhora as habilidades de comunicação automaticamente. Com isso, você pode e deve melhorar a habilidade suave lendo vários blogs, observando como os especialistas fazem isso, participando de vários eventos e ouvindo os palestrantes.

Portanto, você deve prestar atenção à capacidade de ler, escrever, falar com clareza, de forma que colaboradores técnicos e não técnicos possam entender facilmente seus pensamentos.

Ouvir

Esta habilidade está intimamente ligada à anterior. Então, quando você já tem as habilidades para se comunicar bem com outros membros de sua equipe, também é fundamental aprender a ouvir os outros colegas e entender o que eles querem dizer.

Forte habilidade de escuta combinada com alta perspicácia técnica e conhecimento podem ajudá-lo a detectar e até mesmo prevenir bugs e áreas de alto risco no palco de discussões sobre designs, abordagens de desenvolvimento, recursos.

Adaptabilidade

O mercado e os negócios mudam constantemente. Os requisitos do projeto também podem ser alterados e o cronograma é ajustado. Além disso, há uma situação comum quando uma correção em um teste resulta em uma rodada de testes em tarefas / tickets / casos já fechados.

Sem dizer nada sobre os colegas podem entrar e sair da equipe. É por isso que ter habilidades pessoais ou aprimorá-las ajuda você a se adaptar rapidamente a qualquer situação e se tornar um forte jogador de equipe.

Priorização

Essa habilidade está ligada à já mencionada. As tarefas podem ser infinitas, mas, ainda assim, devem ser realizadas. Se o QA deseja ser proativo, trabalhar com eficiência e obter resultados, é fundamental ser altamente organizado, planejar e definir as prioridades corretamente. Essa é uma forma de economizar horas, dias e semanas de seu tempo e cumprir a meta e o prazo final.

Trabalho em equipe

O trabalho em equipe é necessário para quase todos os setores, desde logística até TI. Empregadores, recrutadores e RHs procuram candidatos com fortes habilidades de trabalho em equipe. Se você deseja trabalhar bem com colegas e companheiros de equipe, o trabalho em equipe é essencial. Quanto melhor você trabalhar com os outros, mais bem-sucedida será sua equipe após tarefas.

Além disso, estar atento aos hobbies, interesses, habilidades dos colegas leva ao estreitamento de relacionamentos e à formação de uma equipe amigável que acelera o cumprimento de metas.

Ética de trabalho

É óbvio que a ética do trabalho é uma das habilidades sociais mais importantes não apenas para o controle de qualidade, mas também para qualquer trabalhador. Esses funcionários chegam ao trabalho pontualmente, mantêm o foco, concluem as tarefas prontamente e se mantêm organizados. Uma boa ética de trabalho não significa ser eleito o “Funcionário do Mês”. Significa aplicar suas habilidades da maneira mais eficaz possível. É por isso que você deve se concentrar em melhorar a habilidade e se tornar mais iniciativa durante reuniões, discussões, e você obterá mais satisfação com seu trabalho.

Pensamento crítico

Todo controle de qualidade deve ter pensamento crítico porque é realmente crucial para o teste de software. Os aspectos da personalidade, como uma mente inquisitiva, bem como o desejo de encontrar a verdadeira raiz do problema, saber que perguntas fazer, etc., são geralmente preferíveis até do que credenciais técnicas.

Grande capacidade de tomada de decisão

Quando você é quase um mestre do pensamento crítico, é mais que provável que seja um bom tomador de decisões. Muitas dúvidas são discutidas com os colegas de equipe, clientes em reuniões, e você escolhe a solução mais eficiente. No entanto, pode ocorrer uma situação em que você não tem tempo para consultar ninguém e deve decidir.

Você está pronto para fazer isso? Não é fácil, mas é real adquirir a habilidade. Às vezes, deixe sua intuição ou “sexto sentido” ser seu guia. De qualquer forma, suas habilidades técnicas com o conhecimento da empresa e das necessidades do projeto devem ser muito fortes.

Aprendizagem rápida e atitude de querer aprender

Como já mencionamos, o negócio e o mercado mudam muito rapidamente. Assim, você se adaptaria rapidamente e é impossível sem uma forte capacidade de aprender rapidamente. Novas tecnologias, métodos, técnicas, abordagens, recursos e muito mais devem ser aprendidos caso você deseje ser um profissional. Claro, nada acontecerá caso você não queira. Com isso, uma atitude de querer aprender é outra habilidade soft crucial que você também precisa ter. Visite conferências, leia blogs, participe de cursos e assim por diante para atualizar suas habilidades técnicas.

Raciocínio lógico

Uma das principais tarefas do Analista de Qualidade é automatizar suas próprias tarefas e testes de sistemas em geral, com isso ele precisa de um bom conhecimento lógico para criação de scripts no seu dia a dia.

Mercado e faixas salarias

Como já conversamos, o QA é um profissional muito importante em qualquer empresa, com isso a demanda por profissionais qualificados aumentou significativamente. Com o aumento dessa demanda, as empresas estão valorizando este profissional cada vez mais.

Fizemos uma pesquisa comparando várias empresas no mercado de pequeno e grande porte, usamos também vagas de empregos em várias cidades e plataformas como <https://www.glassdoor.com.br/> e <https://br.indeed.com/>

Com esta pesquisa chegamos a uma faixa média de mercado para a vaga de Analista de QA.

- Analista QA Júnior - **R\$2.080,00 a R\$ 4.300,00/mês**
- Analista QA Pleno - **R\$ 3.500,00 a R\$ 6.058/mês**
- Analista QA Sênior - **R\$ 4.300,00 a R\$ 7.390/mês**

Estes salários podem variar conforme a experiência e outros.

Estes foram salários médios, mas tem vagas que os salários podem passar dos R\$ 9.000/mês.

O que é um teste de software?

Basicamente, o teste faz parte de um processo no desenvolvimento do programa, podendo ser feito pelos próprios desenvolvedores ou, em alguns casos, feito por profissionais especializados na área. O procedimento pretende antecipar e corrigir falhas e bugs que apareceriam para o usuário final.

Apesar desta etapa parecer simples, o seu uso é essencial para evitar o fenômeno conhecido como “**apagar incêndios**”, entre o cliente lidar com instabilidades ou dificuldades em acessar o layout defeituoso. Para evitar as surpresas desagradáveis, você

deve recorrer à diferentes tipos de testes para certificar se o sistema está em funcionamento conforme o planejado.

Quais são os tipos de testes de softwares existentes?

Para identificar e organizar os erros, os testes são separados em diferentes tipos.

Testes de caixa branca

Por ter acesso ao código fonte, o profissional pode observar com mais atenção as determinadas etapas do código. Nesse processo, o usuário analisa por qual caminho ocorre o fluxo de dados e é possível verificar se há a passagem correta em todas as condições esperadas.

Teste de caixa-preta

Sendo o oposto do item anterior, nesse procedimento a pessoa não tem acesso ao código fonte e nem a sua estrutura. Como é baseado nos requisitos funcionais, ele também é chamado de teste funcional.

Quando testamos um código, devemos ficar atentos à maneira que os usuários acessam a aplicação. Por esse motivo, testar todas as combinações possíveis na área de entrada de dados é primordial.

Para que o teste seja feito com sucesso, o resultado deve sair como esperado pelo usuário. Por exemplo, imagine que o cliente deve utilizar o seu RG e data de nascimento para acessar o portal do convênio médico.

Como o programa espera que ambos os campos sejam preenchidos, caso o usuário consiga acessar a sua conta usando apenas um dos campos, significa que há uma falha na consistência de dados de entrada.

O teste pode ser usado para testar casos como:

- consistir a entrada de datas futuras em datas de nascimento;
- consistir entrada de valores negativos em campos de pagamentos;
- verificar o funcionamento dos botões para prosseguir o fluxo de processamento.

Testes de regressão

Durante o desenvolvimento do programa, é comum existirem as seguintes situações: ou a inclusão de uma nova funcionalidade pelo cliente, ou encontrar um erro na lógica do código-fonte.

Independentemente do cenário, o desenvolvedor terá de fazer a alteração na programação. O problema é que, em alguns casos, uma simples mudança pode comprometer toda a lógica já escrita, invalidando quaisquer testes básicos feitos no processo de produção. Para evitar imprevistos, é recomendado que você faça, mesmo que a modificação seja pequena, o teste de regressão.

Teste de usabilidade

Utilizando o ponto de vista do consumidor, o teste procura verificar a experiência do usuário. Dessa forma, o responsável deve checar a organização dos itens disponíveis na tela, observar se o layout está correto e se os botões se comunicam corretamente entre as diferentes páginas do sistema.

A verificação cobre também o desempenho do programa ao executar uma determinada ação. Afinal, não é difícil nos depararmos com um aplicativo que demora anos para ser carregado, não é mesmo? Dessa forma é possível imaginar o processo ter entrado em loop ou ter executado alguma função inesperada.

O processo permite, também, verificar o comportamento da plataforma em diferentes dispositivos. Caso esteja utilizando diferentes navegadores ou dispositivos de tamanhos diferentes, saberemos o quanto o layout é responsivo ou não.

Dito isso, vimos que a execução de teste pode se tornar desgastante em alguns momentos, por esse motivo você deve considerar a automação.

Segurança

Os testes de software evoluíram muito com o tempo e um know-how maior foi adquirido pelos profissionais da área. Dentre os novos recursos utilizados para garantir o pleno funcionamento de um programa de computação estão os testes de segurança.

Essa categoria de teste verifica a segurança do software no que diz respeito à proteção a ataques diversos a que pode estar submetido como hackers e vírus bem como na lida dos dados inseridos pelo usuário.

Integração

Nesse teste, em vez de se atestar funcionalidades do software, se analisa a integração entre as diferentes unidades que formam o sistema. São averiguados aspectos como a interface e a dependência entre os componentes.

Performance

Como o próprio nome sugere, atesta o desempenho do software. Se os comandos dados respondem rapidamente, se os componentes não demoram muito a carregar e se a experiência do usuário é satisfatória no produto testado. É de suma importância, pois por mais bem programado que um software possa ser, sua performance é que determinará o uso satisfatório das funções planejadas.

Instalação

O teste de instalação verifica se sob diferentes condições como pouco espaço de memória, interrupções no sistema e demais entraves que podem comprometer esse processo, o programa consegue ser instalado ou se cede facilmente a essas limitações.

Não é raro, por exemplo, quando alguém está usando um celular e vai instalar um programa novo e o aparelho trava ou a instalação é interrompida por alguns dos fatores citados acima. O melhor é que essas situações não aconteçam e esse processo possa rodar até o fim tranquilamente.

Manutenção

Geralmente, um software não é feito para durar pouco tempo, tampouco uma única versão. Atualizações constantes são necessárias de modo a aprimorar os recursos do programa e a própria experiência do usuário com aquele produto. Os testes de manutenção averíguam se esses aprimoramentos acontecem com sucesso e se são aceitos pelo sistema. Sem isso, corre-se o risco do software tornar-se defasado e até mesmo inoperante pela falta de atualização.

Funcional

O teste funcional abrange tanto o teste de caixa branca quanto o de caixa-preta abordado anteriormente nesse post. Sua importância consiste justamente na capacidade de determinar se o que o software foi programado para fazer está de fato fazendo.

Ele pode ser feito tanto manualmente, quanto automaticamente ou mesmo em um misto dos dois. As várias funções são acessadas e testadas de formas diferentes a fim de encontrar falhas ou mesmo possíveis aprimoramentos no que já está sendo executado. Assim como o teste de performance, o funcional dá uma noção bastante real de como o software se comportará quando chegar até ao usuário. Isso permite que ajustes importantes sejam feitos antes de atender o seu público final.

Aceitação

O teste de aceitação é um tipo de teste aplicado para verificar se o software está pronto e pode ser utilizado pelos usuários, para desempenhar as funções e tarefas para as quais o software foi construído. O teste de aceitação é a ação de teste final antes da implementação do software.

Em muitas empresas, devido à correria e prazos apertados, muitas equipes ignoram as outras etapas de testes e só fazem o **teste de aceitação**, de modo a para garantir que o produto que será entregue realmente está fazendo o que deveria fazer.

Muitas equipes fazem estes testes de forma manual, semelhante ao que o usuário final faria, porém, esse é um trabalho costuma ser muito demorado e plausível de falhas humanas visto que ele é feito manualmente, dependendo do tamanho do recurso liberado, podem ter testes que podem chegar a dias testando.

A maioria das empresas faz os testes de aceitação de forma automatizada, isso é algo que veremos mais a frente.

O que é teste automatizado?

O teste automatizado de software é um processo em que as ferramentas de software executam testes pré-programados em um aplicativo de software antes de serem liberados para produção.

O objetivo do teste automatizado é simplificar o máximo possível do esforço de teste com um conjunto mínimo de scripts. Se o teste unitário consome uma grande porcentagem dos recursos de uma equipe de garantia de qualidade (QA), por exemplo, esse processo pode ser um bom candidato para automação.

As ferramentas de teste automatizadas são capazes de executar testes, reportar resultados e comparar resultados com testes anteriores. Testes realizados com essas ferramentas podem ser executados repetidamente, a qualquer hora do dia.

Existem 3 tipos principais de testes automatizados, eles são::

1. Testes de Unidade Automatizada (Unit Tests)

Testes de unidade automatizada são gravados para teste no nível de código, bugs são identificados nas funções, métodos e rotinas escritas pelos desenvolvedores.

Esses recursos têm acesso ao código-fonte e eles gravam testes de unidade para quebrar o código de produção. Devido à presença de testes de unidade, sempre que o código é compilado, todos os testes de unidade são executados e nos informam se toda a funcionalidade está funcionando. Se algum teste de unidade falhar, significa que há um bug presente no código de produção.

2. Web Service / API Tests

Uma Application Programming Interface (API) possibilita que o software converse com outros aplicativos de software, assim como qualquer outro software, as APIs precisam ser testadas. Nesse tipo de teste, a GUI "Graphical User Interface" geralmente não está envolvida.

O que testamos aqui geralmente são os problemas de funcionalidade, conformidade e segurança. Em aplicativos da web, podemos testar a solicitação e a resposta de nosso aplicativo, quer estejam seguros e criptografados ou não.

Este é um dos exemplos em que podemos usar o Teste de API. As ferramentas mais populares para testes de API são o SoapUI e Rest-assured, que possui versões gratuitas e pagas. Existem outras ferramentas também, que podem ser utilizadas conforme a necessidade do cliente ou da aplicação.

3. Testes de GUI automatizados

O teste de GUI "Graphical User Interface" (Interface Gráfica do Usuário, em português) é um tipo de teste automatizado, é a forma mais difícil de automação porque envolve o teste de uma interface do usuário do aplicativo.

Ele é difícil porque as interfaces gráficas são altamente sujeitas a mudanças. Mas esse tipo de teste também está mais próximo do que os usuários farão na aplicação.

Como o usuário usará o mouse e o teclado, os testes automatizados de GUI também imitam o mesmo comportamento, fazendo uso do mouse e do teclado para clicar ou gravar em objetos presentes na interface do usuário. Devido a isso, podemos encontrar erros precoces e isso pode ser usado em muitos cenários, como testes de regressão ou preenchimento de formulários que levam muito tempo.

Segundo o site Medium, segue o ranking das 10 melhores ferramentas para testes automatizados:

1. Selenium - <http://www.seleniumhq.org/>
2. Katalon Studio - <https://www.katalon.com/>
3. UFT - <https://software.microfocus.com/fr-ca/software/uft>
4. Watir - <http://watir.com/>
5. IBM Rational Functional Tester - <https://www.ibm.com/>
6. TestComplete - <https://smartbear.com/>
7. TestPlant eggPlant - <https://www.testplant.com/>
8. Tricentis Tosca - <https://www.tricentis.com>
9. Ranorex - <https://www.ranorex.com/>
10. Robot framework - <http://www.robotframework.org/>

Além dos testes informados anteriormente, podemos também fazer outros tipos de testes como os de desempenho, estes testes de desempenho podem ser separados em categoria, vejamos:

1. Teste de carregamento

Os testes de carregamento determinam como o software funciona com o aumento da carga de trabalho em um determinado tempo. Esta carga de trabalho pode ser simultânea a usuários, o número de transações, o comportamento do software, etc.

Seu objetivo é observar o tempo de resposta, as taxas de produção, a utilização de recursos, e muito mais. Ao identificar quaisquer gargalos de desempenho nestes atributos, é possível solucionar problemas antes de lançar o aplicativo para garantir uma melhor experiência ao usuário final.

Através de testes de carregamento, os desenvolvedores podem entender quantos usuários simultâneos são capazes de lidar com um aplicativo de software a qualquer momento.

2. Teste de saturação

O teste de saturação também chamado resistência, avalia o desempenho do software durante um período prolongado sob uma carga de trabalho fixa e regular. Em outras palavras, ele determina quanto tempo o software pode suportar uma carga de trabalho constante para proporcionar sustentabilidade a longo prazo.

Durante este teste, as equipes de teste monitoram KPIs como vazamentos de memória, uso de memória, falta de memória e muito mais. Os testes de saturação também analisam os tempos de resposta e o rendimento após uso prolongado para mostrar se essas métricas são consistentes ou não.

3. Testes de resistência

O teste de resistência mede o desempenho do software além dos parâmetros normais de trabalho. Isso significa que o software é colocado sob maiores carregamentos de tráfego (mais usuários, transações, etc.) para ajudar os desenvolvedores a aprender o quão bem o software funciona acima de seus limites de capacidade esperados.

O objetivo deste teste é determinar a estabilidade do software. Ele ajuda a compreender o ponto em que o software falha e como ele se recupera da falha. Ele esforça recursos de hardware como CPU, memória, espaço em disco, etc., para medir o ponto de ruptura do aplicativo com base na utilização dos recursos.

Exemplo: As empresas realizam testes de resistência antes de alguns grandes eventos como Black Friday para verificar o desempenho de aplicativos de comércio eletrônico.

4. Testes de picos

O teste de pico é um tipo de teste de resistência que mede o desempenho do software sob um “pico” significativo e repentino ou aumentando a carga de trabalho como usuários simulados. Ele diz se o software pode lidar com esse aumento abrupto da carregamento de trabalho repetida e rapidamente.

5. Teste de volume

Durante a fase de desenvolvimento, apenas uma pequena quantidade de dados é usada para testar o código. O teste de volume (também chamado de “flood testing”) testa a eficiência do software quando submetido a grandes volumes de dados. Ele verifica a perda de dados, tempo de resposta do sistema, confiabilidade do armazenamento de dados, e muito mais.

6. Teste de escalabilidade

Os testes de escalabilidade medem a eficácia do software no manuseio de uma quantidade crescente de trabalho. O teste dirá seu comportamento quando você aumentar ou diminuir os atributos de desempenho do software.

Você pode realizar testes de escalabilidade adicionando volume de dados ou usuários gradualmente enquanto monitora o desempenho do software.

Diferenças entre testes Automatizados e Manuais

Teste manual significa testar um aplicativo manualmente por um ser humano. Um especialista em garantia de qualidade (testador) que executa testes manuais garante que um aplicativo esteja funcionando corretamente seguindo as condições descritas nos casos de teste. O testador avalia o design, a funcionalidade e o desempenho do aplicativo verificando vários elementos.

O teste manual é útil quando o teste automatizado não é possível. Por exemplo, para testar um aplicativo móvel durante condições imprevisíveis, como quando o telefone está no bolso do usuário e há muitos toques não intencionais. Os testadores simulam as condições para verificar se isso levará ao travamento do aplicativo.

Nos testes automatizados, existem testes pré-programados executados automaticamente. Os testes são executados para comparar os resultados reais com os resultados esperados. Os testes automatizados ajudam a determinar se o aplicativo tem o desempenho esperado ou não.

Testes automatizados são úteis ao executar testes repetitivos e testes de regressão para garantir que um aplicativo funcione corretamente após a implementação de novas alterações. Os testes automatizados são executados com a ajuda de ferramentas, scripts e softwares para executar ações predefinidas escritas nos casos de teste.

Vejamos algumas diferenças:

O teste manual é propenso a erros humanos. Portanto, a precisão e a confiabilidade dos resultados do teste são mais baixas. Já o teste automatizado, por outro lado, é mais confiável. Ferramentas e / ou scripts realizam o teste automatizado de maneira mais precisa, evitando esquecimentos e outros.

O custo do teste manual depende dos recursos humanos utilizados no teste. O custo do teste automatizado depende das ferramentas de teste implementadas para a realização dos testes.

O tempo necessário para o teste manual é maior à medida que os recursos humanos o realizam. O tempo necessário para o teste automatizado é menor quando as ferramentas de software executam os testes.

O teste manual é adequado quando os casos de teste são executados uma ou duas vezes. Portanto, não há repetição frequente de casos de teste. O teste automatizado é adequado quando os casos de teste precisam ser executados repetidamente por um longo período de tempo.

O teste manual ajuda a descobrir a facilidade de uso de um aplicativo. Envolve observação humana para descobrir quaisquer falhas. Portanto, o teste manual ajuda a melhorar a experiência do cliente. O teste automatizado não garante a facilidade de uso do aplicativo. Como não há observação humana envolvida, não há garantia de experiência positiva do cliente.

Quanto usar o teste manual então?

O teste manual é mais adequado para o seguinte:

1. TESTES EXPLORATÓRIOS

Os testes manuais são úteis ao trabalhar com documentação de especificação mal escrita e há escassez de tempo. Nesse tipo de teste, é necessário um Analista de Qualidade experiente, com bom conhecimento do domínio das técnicas, além de

fortes habilidades analíticas, criatividade e intuição. Essa abordagem requer um planejamento mínimo e a máxima execução de teste.

2. TESTES DE USABILIDADE

Esse teste ajuda a descobrir a facilidade de uso de um aplicativo. O objetivo deste teste é descobrir a satisfação do usuário final com o aplicativo, então a observação humana é importante. Portanto, a abordagem manual é imprescindível neste tipo de testes.

3. TESTES AD-HOC

Esse tipo de teste é realizado sem nenhum planejamento e documentação. O resultado desse teste depende do conhecimento e das habilidades do testador.

Quando usar testes automatizados?

Os testes automatizados são adequados para os seguintes cenários:

1. TESTES DE REGRESSÃO

O teste automatizado é adequado quando há alterações frequentes do código e que exigem testes frequentes. Esse teste é realizado para garantir que as novas alterações de código implementadas não afetem as funcionalidades existentes.

2. TESTES DE CARGA

O teste automatizado é preferido para testes de carga ajudando a determinar o desempenho de um sistema em condições de carga da realidade. Esse teste descobre como o sistema se comporta quando vários usuários acessando simultaneamente.

3. TESTES DE PERFORMANCE

O teste de desempenho ajuda a descobrir a velocidade, escalabilidade e estabilidade do aplicativo. O objetivo é eliminar os gargalos de desempenho. Isso requer simulação de milhares de usuários simultâneos. O teste automatizado é o preferido para testes de desempenho.

PRÓS DO TESTE MANUAL:

1. TESTES AO VIVO

- O testador pode testar o aplicativo em condições semelhantes a quando o aplicativo estiver no ar. Quaisquer erros ou falhas que ocorram quando o aplicativo está ativo podem ser rastreados com a ajuda de testes manuais.
- Problemas de interface do usuário e UX
- Ajuda a identificar problemas relacionados à aparência visual do aplicativo. Também ajuda a descobrir os problemas de usabilidade. Os testes automatizados não conseguem identificar essas lacunas.

2. MENOS INVESTIMENTO

- O teste manual requer baixo investimento, pois não requer ferramentas caras ou habilidades de alto nível para ser executado.

3. ADAPTABILIDADE ÀS MUDANÇAS

- O teste manual é adequado quando você faz muitas alterações não planejadas no aplicativo e o aplicativo precisa ser testado após a implementação destas alterações.

CONTRAS DO TESTE MANUAL:

1. POSSIBILIDADE DE ERROS

Como o teste manual depende de humanos, há uma maior possibilidade de erros. Os resultados dos testes serão tão bons quanto, maior for a habilidade do testador que está realizando os testes. Com o teste manual, há chances de o testador não encontrar os mesmos erros ao testar novamente o aplicativo.

2. DEMORADO

Os testes manuais não podem ser reutilizados, o que significa que toda vez que há uma alteração no aplicativo, o testador pode precisar executar todos os casos de testes novamente. Essa pode ser uma atividade demorada.

3. LIMITAÇÕES DE TESTE

O teste manual não é bom para testes de carga e desempenho. Para verificar o desempenho de um aplicativo em testes manuais, é necessário um grande número

de usuários e aparelhos.

PRÓS DOS TESTES AUTOMATIZADOS:

1. RESULTADOS RÁPIDOS

Testes automatizados requerem muito menos tempo para serem executados. Os testes podem ser reutilizados e é fácil executá-los em diferentes máquinas.

2. MENOS ERROS

Os testes automatizados são menos suscetíveis a erros em comparação com a intervenção humana. Portanto, os resultados dos testes automatizados são mais confiáveis.

3. COLABORAÇÃO

Nos testes manuais, apenas o testador tem visibilidade do progresso e dos resultados. Enquanto nos testes automatizados, outros testadores podem entrar na estrutura de testes e visualizar os resultados dos testes progressivamente.

CONTRAS DOS TESTES AUTOMATIZADOS:

1. ALTO CUSTO

O teste automatizado exige que você compre licenças de programas caros de teste de software. Você também pode precisar de uma infraestrutura de hardware adicional para executar testes automatizados, os quais incorrem em custos adicionais.

2. INCAPACIDADE DE DETECTAR PROBLEMAS DE INTERFACE DO USUÁRIO E USABILIDADE

O código do software é incapaz de julgar a estética e a usabilidade de um aplicativo. Os seres humanos podem julgar melhor esses aspectos. Essa é uma grande limitação do teste automatizado.

3. LIMITAÇÃO DE FERRAMENTAS

O desempenho do teste automatizado depende das ferramentas utilizadas para executar o teste. Uma ferramenta pode ter suas limitações e pode ter seus próprios erros e problemas.

CONCLUSÃO

Para otimizar o custo e o tempo necessários para a conclusão bem-sucedida de um projeto, você não pode depender apenas do teste manual ou da abordagem de teste automatizado. Você precisa de uma combinação certa de testes manuais e testes automatizados para obter os melhores resultados.