

A automação de testes é uma área em franca expansão, no entanto, é uma área ainda muito imatura. Muitos dos sucessos nos projetos de automação de testes são decorrentes de processos empíricos de tentativa e erro.

Para piorar a situação, a automação de testes ainda é objeto de mitos que geram percepções errôneas sobre os seus reais benefícios e limitações. Frequentemente, as ferramentas de automação de testes são supervalorizadas gerando falsas expectativas, a infraestrutura requerida é subestimada, entre outros problemas comuns que serão apresentados ao longo deste artigo.

Melhores Práticas na Automação de Testes:

- **A automação de testes não é um processo de testes**
- **Automatize os testes críticos primeiro**
- **Incorpore testabilidade ao aplicativo**
- **As ferramentas de automação de testes também têm defeitos**
- **Demo não é prova de conceito**
- **Dimensione a infraestrutura adequadamente**
- **Encare a automação de testes como um projeto**
- **Alinhe as expectativas e garanta a colaboração de todos os envolvidos**
- **A automação de testes é um investimento de longo prazo**
- **O teste manual é insubstituível**
- **Conclusão**

Nas seções a seguir serão apresentadas algumas boas práticas e conselhos reunidos pelo autor durante os últimos anos (à custa do aprendizado empírico em **projetos de automação de testes**).

A automação de testes não é um processo de testes

Muitas empresas fracassam na implantação da automação de testes em virtude da inversão de prioridades causada pela busca da qualidade a qualquer preço. Neste cenário, as empresas adquirem uma **ferramenta de automação de testes** prematuramente sem, ao menos, possuírem um grau mínimo de maturidade no processo de testes. O processo de testes é informal, as responsabilidades não são bem definidas e os profissionais não possuem o perfil adequado ou não são qualificados.

Segundo Watts S. Humphrey, criador do CMM (Capability Maturity Model), a qualidade do produto é diretamente proporcional à qualidade do processo utilizado no seu ciclo de vida. A implantação da automação de testes depende de testes manuais maduros e consistentes. Os projetos de automação de testes bem-sucedidos são aqueles que se baseiam em processos de testes formais e estruturados. Afinal, como é possível esperar alguma coisa de **testes automatizados** que são baseados em testes manuais errados, ambíguos ou inconsistentes. Não é possível automatizar o caos.

A rigor, o sucesso na implantação da automação de testes depende do entendimento de que a automação de testes ou uma ferramenta de automação, por si só, não vão melhorar ou organizar os testes existentes. É necessário antes, fazer a “faxina” e implantar um processo de testes formal. A necessidade de automatizar os testes virá naturalmente como resultado da evolução da maturidade do processo de testes.

Automatize os testes críticos primeiro

Nove em cada dez projetos de automação de testes cometem o erro de transformar todos os casos de testes manuais em `scripts` de testes automatizados. O leitor deve estar se perguntando: Mas este não seria o objetivo principal da automação de testes? – Sim e Não.

A automação de testes tem o objetivo de reduzir o envolvimento humano em atividades manuais repetitivas. Entretanto, isto não significa que a automação de testes deve se limitar apenas a fazer o trabalho repetitivo e chato. Os casos de testes manuais foram criados num contexto em que os testes seriam executados manualmente. É muito provável que estes casos de testes manuais não sejam muito eficientes em um contexto em que os testes serão executados automaticamente.

Frequentemente, antes de iniciar a automação, os testes devem ser reprojatados a fim de aumentar a probabilidade de revelar um defeito que ainda não tenha sido encontrado. Afinal, o grande **benefício da automação de testes** não é a execução dos testes mais rápida e a qualquer hora do dia ou da noite, mas o aumento da amplitude e profundidade da cobertura dos testes.

Na prática, a automação de 100% dos testes manuais nem sempre é a melhor estratégia. A automação de testes é pouco eficaz quando os testes são complexos e exigem interações Inter sistemas ou validações subjetivas (estes tipos de testes devem permanecer manuais). Além disso, muitas vezes o custo e o tempo para automatizar os testes de um projeto são maiores que o custo e o tempo do próprio projeto de desenvolvimento (o que inviabilizaria a automação de 100% dos testes manuais).

A escolha dos testes automatizados candidatos, ou seja, os mais críticos, deve ser realizada com base no contexto do projeto de automação. No entanto, apesar de não existir uma categorização amplamente difundida, a experiência tem mostrado que os testes candidatos são normalmente agrupados em quatro áreas distintas:

- **Smoke Tests:** Um conjunto mínimo de testes é selecionado com o objetivo de validar um `Build` ou liberação antes do início de um ciclo de testes;
- **Testes de Regressão:** Os testes são selecionados com o objetivo de executar o reteste de uma funcionalidade ou da aplicação inteira;
- **Funcionalidades Críticas:** Os testes são selecionados com o objetivo de validar as funcionalidades críticas que podem trazer riscos ao negócio;
- **Tarefas Repetitivas:** Os testes são selecionados com o objetivo de reduzir o envolvimento dos testadores em atividades manuais repetitivas e suscetíveis a erros, tais como cálculos matemáticos, simulações, processamentos, comparações de arquivos ou dados etc.

Incorpore testabilidade ao aplicativo

Via de regra, os testes são automatizados utilizando a aplicação do jeito que ela é. Ou seja, os testes são criados sob o ponto de vista da interface gráfica e com o único objetivo de automatizar as ações dos usuários finais.

No entanto, testar por meio da interface gráfica nem sempre é a melhor opção para testes que exigem desempenho. Além disso, às vezes a interface gráfica não fornece evidências de que o teste foi realizado com sucesso, afinal, nem sempre a mensagem “Essa operação foi realizada com sucesso” significa que a operação foi realizada com sucesso de verdade.

A experiência tem mostrado que a incorporação de testabilidade na aplicação é um fator chave para o **sucesso de um projeto de automação de testes**. A testabilidade é um atributo que determina a capacidade de uma aplicação ser testada, ou seja, a facilidade de se testar uma aplicação está diretamente ligada ao nível de testabilidade incorporado nesta aplicação.

Normalmente é necessário realizar modificações na aplicação para torná-la mais fácil de testar. Essas modificações têm o objetivo de incorporar um conjunto de mecanismos que facilitam a observação e o controle do estado dos componentes internos da aplicação. Adicionalmente, estes mecanismos expõem ao mundo exterior as funcionalidades da aplicação por meio de APIs, Interfaces de Linha de Comando, Hooks etc.

Isto torna, por sua vez, a aplicação muito mais fácil de se testar, sob o ponto de vista da automação de testes. O objetivo desses mecanismos

(APIs, Interfaces de Linha de Comando, Hooks) é fornecer interfaces para o mundo exterior que não seja dependente da interface gráfica da aplicação. Dessa forma, é possível criar testes especializados para exercitar algumas funcionalidades da aplicação sem que seja necessário utilizar uma interface gráfica.

As ferramentas de automação de testes também têm defeitos

A automação de testes é, em última instância, a execução de um software para testar outros softwares. Dessa forma, podemos afirmar que as ferramentas de automação de testes sofrem dos mesmos tipos de problemas que as aplicações convencionais, ou seja: defeitos. É muito comum ocorrerem atrasos em projetos de automação em virtude de problemas causados pela ferramenta de automação de testes. Dentre os problemas mais comuns, devemos destacar:

- Manuais incompletos e ambíguos;
- Defeitos não reproduzíveis que ocorrem aleatoriamente;
- Vazamento de memória quando a ferramenta é executada durante muitas horas;
- Degradação do desempenho quando a ferramenta é executada durante muitas horas;
- Travamentos durante a gravação ou execução dos testes;
- Relato incorreto de testes executados com sucesso quando ocorrem problemas ou vice-versa;
- Recursos que não funcionam como deveriam;
- Reconhecimento incorreto dos componentes (botões, campos, menus etc.) da aplicação em teste;
- Defeitos de regressão (funcionalidades que não funcionam mais corretamente após um `upgrade` da ferramenta);
- Defeitos críticos que impedem a utilização das principais funcionalidades da ferramenta.

Com base no que foi exposto, a experiência tem mostrado que as seguintes ações devem ser tomadas antes da compra e durante a utilização de uma ferramenta de automação de testes:

- Criar uma prova de conceito antes de comprar a ferramenta para certificar-se que ela não tenha defeitos críticos;
- Utilizar sempre a versão mais atual da ferramenta;

- Não utilizar a versão mais atual da ferramenta (`upgrade`) sem realizar um teste de regressão (para evitar que a correção de um problema crie problemas piores);
- Se não houver solução para os problemas ou limitações da ferramenta, considere a criação de soluções alternativas (`workarounds`) utilizando a própria ferramenta.

Demo não é prova de conceito

As maravilhas oferecidas pelas ferramentas de testes automatizados normalmente são apresentadas por demonstrações (Demos). Normalmente, essas ferramentas são compradas com base na boa impressão causada durante essas apresentações.

Triste engano, infelizmente. As demonstrações normalmente apresentam cenários de utilização muito simples em um ambiente controlado. Às vezes aquela ferramenta perfeita apresentada na demonstração pode ser um completo desastre para testar a sua aplicação. Neste contexto, a melhor prática é demonstrar os benefícios, limitações e restrições da ferramenta por meio de uma prova de conceito.

Em resumo, você deve escolher os testes mais importantes e os mais complexos que serão automatizados no seu projeto de automação e criar todos os `scripts` utilizando uma versão demo ou `trial` da ferramenta de automação.

Conforme mencionado anteriormente, as demonstrações apresentam as principais funcionalidades da ferramenta de automação por meio de cenários e testes simples. Uma prova de conceito, por sua vez, tem o objetivo de provar se a ferramenta atenderá as reais necessidades do `seu` projeto de automação.

Dentre os aspectos que devem ser analisados numa prova de conceito, devemos destacar:

- Avaliar se as funcionalidades da ferramenta de automação funcionam conforme apresentado na demonstração;
- Avaliar se os manuais são adequados e informativos;
- Avaliar se existem defeitos críticos que impedem a utilização das principais funcionalidades da ferramenta;
- Avaliar se existe degradação do desempenho ou vazamento de memória quando a ferramenta é executada durante muitas horas;

- Avaliar se a ferramenta reconhece os componentes e componentes personalizados (botões, campos, menus, etc) da aplicação que será testada;
- Avaliar se a ferramenta e a linguagem de `script` oferecida é robusta o suficiente para lidar com testes complexos;
- Avaliar se a ferramenta é realmente fácil de usar e se a ferramenta oferece recursos para a criação de bibliotecas de `scripts` a fim de facilitar a reutilização;
- Avaliar se a ferramenta funciona adequadamente na infraestrutura/ambiente existente e avaliar se será necessário algum investimento adicional para adequar a infraestrutura/ambiente aos requerimentos mínimos exigidos pela ferramenta;
- Identificar as limitações e restrições da ferramenta para evitar falsas expectativas.

Dimensione a infraestrutura adequadamente

Muitos projetos de automação de testes fracassam em virtude do subdimensionamento da infraestrutura. As ferramentas de testes automatizados exigem computadores de alto desempenho com processadores rápidos e grandes quantidades de memória. Além disso, estes computadores devem ser dedicados exclusivamente para a automação de testes (durante a execução dos testes o computador não pode ser utilizado para outro fim).

Os computadores onde os testes serão executados devem ser semelhantes ou com maior capacidade de processamento em relação aos computadores utilizados para criar os `scripts` de testes automatizados, caso contrário, ocorrerão problemas de baixa de performance.

A execução dos testes automatizados é muito sensível a influências externas. Devemos reforçar que diferenças sutis no ambiente (sistema operacional, rede, versões dos programas, dados etc.) podem prejudicar ou impedir a execução dos testes automatizados.

Planejar com detalhes a infraestrutura necessária para criar e executar os testes automatizados é um fator chave para o sucesso de um projeto de automação de testes. Dentre os aspectos que devem ser considerados, devemos destacar:

- **Computadores de alto desempenho:** Os computadores que serão usados para criar e executar os testes automatizados deve estar em conformidade com os requerimentos mínimos exigidos pela ferramenta de automação;
- **Computadores dedicados:** Os computadores onde os testes automatizados serão executados devem ser dedicados para essa função, caso contrário os testes podem ser prejudicados e pode ocorrer degradação da performance;
- **Ambiente isolado:** O ambiente deve ser restrito ao time de automação de testes para evitar a desestabilização da integridade do ambiente por meio de influências externas;
- **Ambiente controlado:** Todos os detalhes do ambiente (sistema operacional, rede, versões dos programas, dados etc.) devem ser planejados e controlados, sob pena de prejudicar ou impedir a execução dos testes automatizados;
- **Ambiente similar ao de produção:** O ambiente onde os testes serão executados deve ser igual ou similar ao ambiente de produção, sob pena de encontrar falsos positivos, ou seja, os testes são executados com sucesso no ambiente de testes, mas as funcionalidades apresentam defeitos em produção;
- **Massa de dados consistente:** Os dados utilizados nos testes devem ser conhecidos e representativos, além disso, devem ser armazenados numa linha de base (**baseline**) a fim de permitir que sejam recuperados todas as vezes que os testes automatizados forem executados.

Encare a automação de testes como um projeto

A automação de testes é uma combinação entre teste e desenvolvimento de software, afinal, a atividade de criação de **scripts** de teste é uma atividade de programação pura. Dessa forma, podemos afirmar que a automação de testes deve ser encarada como um projeto com características próprias, ou seja, exige um planejamento detalhado, assim como, atividades de projeto, desenvolvimento e testes, tal qual o desenvolvimento de um software convencional.

Via de regra, a criação de testes automatizados é realizada por testadores especializados em desenvolvimento, também conhecidos como Testador-Desenvolvedor ou Automatizador. Este profissional deve ter o perfil de desenvolvedor e ser treinado adequadamente para utilizar a ferramenta de automação de testes.

Com base no que foi exposto, a experiência tem mostrado que os seguintes aspectos devem ser considerados em projetos de automação de testes:

- O projeto de automação deve ser encarado como um projeto com características próprias, ou seja, exige um planejamento detalhado, assim como, atividades de projeto, desenvolvimento e testes;
- Os `scripts` de testes podem ter defeitos, logo, recomenda-se a utilização de uma ferramenta de gestão de defeitos (`bugtracker`) para gerenciar estes defeitos;
- Os `scripts` de testes devem ser submetidos a um processo de Gerência de Configuração de Software e controle de versões;
- Durante o desenvolvimento dos `scripts` de testes, deve-se aplicar as melhores práticas utilizadas em projetos de desenvolvimento de software convencionais;
- O Testador-Desenvolvedor deve ter o perfil e interesse em trabalhar com automação de testes, nem sempre um bom testador ou analista de testes se tornará um bom Testador-Desenvolvedor;
- Os recursos humanos que atuarão no projeto de automação de testes devem ser treinados adequadamente. Quanto mais capacitados, maior será a probabilidade de sucesso do projeto.

Alinhe as expectativas e garanta a colaboração de todos os envolvidos

Os projetos de automação de testes costumam fracassar porque as pessoas envolvidas têm percepções diferentes sobre os benefícios e as limitações de um projeto de tal natureza.

Normalmente a alta gerência acredita que a automação de testes é a solução de todos os problemas. Os arquitetos e desenvolvedores desconhecem a necessidade da criação de mecanismos para aumentar a testabilidade durante o projeto e desenvolvimento da aplicação.

O time de testes, por sua vez, costuma acreditar que a automação dos testes é uma tarefa simples que não exige capacitação e é resumida apenas pela transformação dos testes manuais em `scripts` de testes automatizados por meio dos recursos de gravação (`recording`) oferecidos pelas ferramentas de automação de testes.

Essas suposições errôneas são o resultado da falta de conhecimento dos benefícios e limitações de um projeto de automação de testes. As

campanhas publicitárias das ferramentas de automação, por sua vez, pioram esse cenário em virtude de que vendem uma imagem de que a automação de testes é a solução de todos os problemas de qualidade. Dentre as artimanhas de marketing mais comuns usadas pelas empresas fabricantes das ferramentas de automação, devemos destacar:

- Desenvolva software de qualidade e aumente a receita e a lucratividade da sua empresa por meio da ferramenta de testes automatizados XYZ;
- Crie testes sofisticados com treinamento mínimo;
- Utilize o recurso XYZ para gravar os `scripts` e elimine o tempo de codificação dos `scripts`;
- Aumente a produtividade e diminua o tempo gasto em manutenção por meio do compartilhamento de `scripts` e bibliotecas;
- Identifique os problemas facilmente por meio do recurso XYZ utilizado nos nossos relatórios;
- Utilize o recurso XYZ de reconhecimento automático de objetos e garanta a execução dos testes mesmo quando a interface gráfica tenha mudado.

Em projetos de automação de testes, é necessário o alinhamento das expectativas e a colaboração entre a alta gerência, desenvolvedores, arquitetos e testadores. É de suma importância explicar com detalhes os objetivos e o escopo de um projeto de automação de testes e, acima de tudo, garantir que as expectativas de todas as partes interessadas sejam realistas. Dessa forma, conseguimos eliminar o folclore e os mitos criados pelas estratégias de marketing usadas pelas empresas fabricantes das ferramentas de automação.

A estratégia recomendada para garantir o alinhamento das expectativas com relação aos benefícios e limitações de um projeto de automação de testes deve considerar as seguintes atividades:

- Demonstrar os benefícios, limitações e restrições da ferramenta por meio de uma prova de conceito;
- Planejar com detalhes a infraestrutura necessária para criar e executar os testes automatizados para evitar problemas de subdimensionamento;
- Envolver desenvolvedores e arquitetos no projeto de automação de testes com o intuito de incorporar mecanismos para aumentar a testabilidade da aplicação;
- Envolver os testadores para alinhar as expectativas em relação aos objetivos e estratégia da automação de testes, assim como, fomentar o treinamento e a capacitação;

- Envolver a alta gerência para alinhar as expectativas em relação ao retorno de investimento, benefícios, limitações e restrições de um projeto de automação de testes.

A automação de testes é um investimento de longo prazo

A automação de testes, sem dúvida, é uma boa prática em um processo de teste de software. No entanto, a automação de testes não é uma prática que se adota do dia para a noite. Conforme mencionamos anteriormente, a **necessidade de automatizar os testes** virá naturalmente como resultado da evolução da maturidade do processo de testes.

Apesar de todos os benefícios advindos da automação de testes, os investimentos são altos. Rex Black, guru na área de testes de software, no seu famoso artigo chamado “Successful Investing in Software Testing” apresenta um cenário hipotético com os custos requeridos para realizar testes informais, manuais e automatizados e o retorno de investimento (ROI) obtido em cada uma dessas abordagens.

Neste artigo, Rex Black, parte da premissa de que o custo para corrigir um defeito encontrado na fase de desenvolvimento custa cerca de \$10, na fase de teste custa \$100 e em produção custa \$1.000, como pode ser visto na **Tabela 1**.

		Teste Informal	Teste Manual	Teste Automatizado
Investimentos	Pessoal	0	60.000	60.000
	Infraestrutura	0	10.000	10.000

	Ferramentas	0	0	12.500
	TOTAL	0	70.000	82.500
Fase de desenvolvimento	Defeitos encontrados	250	250	250
	Custo para corrigir	2.500	2.500	2.500
Fase de testes	Defeitos encontrados	0	350	500
	Custo para corrigir	0	35.000	50.000
Produção	Defeitos encontrados	750	400	250
	Custo para corrigir	750.000	400.000	250.000

Custo da Qualidade	Conformidade (investimento em prevenção)	0	70.000	82.500
	Não Conformidade (custo para corrigir os defeitos encontrados)	752.500	437.500	302.500
	TOTAL	752.500	507.500	385.000
Retorno de Investimento		N/A	350%	445%

Tabela 1. Retorno do investimento de testes informais, manuais e automatizados

No exemplo proposto por Rex Black, a abordagem utilizando testes automatizados garante um retorno de investimento de 445%. A mensagem é clara: a automação de testes paga o investimento. O grande problema é quanto investimento é necessário e quando esse investimento vai começar a dar retorno.

Você poderá notar que o exemplo de Rex Black considera apenas um investimento de \$12.500 para a compra da ferramenta de automação de testes. Por outro lado, devemos lembrar que uma ferramenta é apenas o meio pelo qual implementamos a automação de testes, mas a automação de testes não se limita apenas à utilização de uma ferramenta.

Como vimos anteriormente, a automação de testes deve ser encarada como um projeto com características próprias, ou seja, exige um

planejamento detalhado, assim como, atividades de projeto, desenvolvimento e testes, tal qual o desenvolvimento de um software convencional. Ou seja, o investimento em automação de testes não se limita apenas aos custos da ferramenta de automação de testes.

Infelizmente, **muitos projetos de automação de testes costumam fracassar** porque a alta gerência costuma acreditar que o único investimento necessário para a automação de testes é a compra de uma ferramenta. Entretanto, existem muitos outros investimentos necessários, tais como:

- Contratação ou capacitação de pessoal para realizar a gestão do projeto de automação de testes;
- Contratação ou capacitação de pessoal para projetar/criar os casos de testes automatizados;
- Compra ou melhoria da infraestrutura/ambiente para atingir os requerimentos mínimos exigidos pela ferramenta;
- Gastos relacionados à incorporação de testabilidade na aplicação para torná-la mais fácil de testar;
- Gastos relacionados à criação de provas de conceitos ou protótipos.

A automação de testes é um investimento com retorno garantido quando aplicada corretamente. No entanto, o retorno é de longo prazo, ou seja, não existe retorno imediato.

Conforme mencionamos anteriormente, a automação dos testes deve começar a partir de uma prova de conceito e gradualmente crescer até atingir maturidade por meio de um conjunto de `scripts` robustos e estáveis que garantam uma cobertura de testes adequada.

O caminho para a maturidade na automação de testes é tortuoso e demorado. É necessária muita persistência para atingir a maturidade na automação de testes. A melhor prática para aumentar as chances de sucesso é evoluir iterativamente. Dessa forma, os prejuízos advindos de uma decisão errada numa iteração serão baixos e administráveis.

O teste manual é insubstituível

Segundo Cem Kaner, autor do livro “Lessons Learned in Software Testing”, o propósito da automação de testes pode ser resumidamente descrito como a aplicação de estratégias e ferramentas tendo em vista a redução do envolvimento humano em atividades manuais repetitivas.

Devemos reconhecer que a automação de testes não deve ser empregada como um substituto do teste manual. Ela deve ser introduzida como uma técnica adicional cujo objetivo principal é agregar valor, sem, no entanto, invalidar o teste manual existente. Afinal, testes manuais e automatizados são abordagens de testes diferentes que se reforçam mutuamente.

Dessa forma, por meio da automação dos testes, os testadores poderão reduzir o seu envolvimento em atividades manuais repetitivas e focar em abordagens de testes complementares que exigem a aplicação da intuição e julgamento. O teste exploratório é uma estratégia de testes complementar interessante uma vez que faz uso intensivo da intuição e julgamento do testador.

O teste exploratório é, na sua definição mais básica, a criação e a execução ao mesmo tempo de um teste. Quando se realiza um teste exploratório, normalmente o testador não tem informações detalhadas sobre o que vai testar e como vai testar. O testador se baseia na sua experiência, assim como no conhecimento que ele vai adquirindo sobre o aplicativo durante a execução do teste exploratório.

A partir dessa perspectiva, podemos afirmar que o teste exploratório é uma atividade iterativa e empírica de exploração que exige idas e vindas num processo de investigação contínuo onde a intuição, a criatividade e a experiência do testador são indispensáveis para garantir a eficiência do teste.

O grande benefício da utilização de testes exploratórios está ligado à sua natureza empírica. **Os testes automatizados repetem sempre os mesmos testes**, os mesmos caminhos, as mesmas operações. Por meio dos testes exploratórios temos a oportunidade de empiricamente e iterativamente criar novos testes e seguir novos caminhos, que por sua vez, nos leva a descoberta de novos defeitos.

Conclusão

A automação de testes não deve ser empregada como um substituto do teste manual. Ela deve ser introduzida como uma técnica adicional cujo objetivo principal é agregar valor. O enfoque deve ser na melhoria do processo de testes utilizado na sua empresa. A necessidade de automatizar os testes virá naturalmente como resultado da evolução da maturidade do processo de testes.

Finalmente, devemos lembrar que a automação de testes é um investimento cujo retorno é de longo prazo. O caminho para a maturidade na automação de testes é tortuoso e demorado.

Nesta jornada, provavelmente surgirão problemas imprevistos, assunções incorretas, expectativas não alinhadas, defeitos na ferramenta de automação, falta de testabilidade na aplicação, problemas na infraestrutura/ambiente, entre outros aspectos discutidos neste artigo. O importante é aprender com os próprios erros e seguir em frente, afinal, aqueles que não conseguem se lembrar dos erros do passado estão condenados a repeti-los (George Santayana).