

# Teste de Software

AUTOMATIZAÇÃO DE TESTES



COTI Informática  
Escola de Nerds



# 1. AUTOMAÇÃO NO TESTE DE SOFTWARE.

# 1. AUTOMAÇÃO NO TESTE DE SOFTWARE.

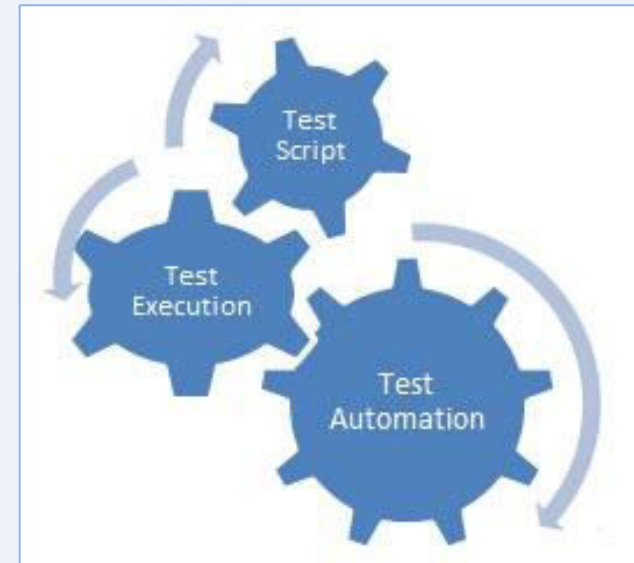
A automação vem aos longos dos anos ganhando um papel importante na área de Teste de *Software*.

E isso se deve a uma série de fatores, dentre os quais podemos destacar:

- Diminuição do uso de mão de obra;
- Diminuição dos custos;
- Aumento na velocidade do processo de Teste de *Software*;
- Maior sustentabilidade da garantia da qualidade, perante o “triângulo da gerência de projeto” (escopo, tempo e dinheiro).

Podemos utilizar a automação em qualquer fase do Teste de *Software*, como especificação de casos de teste, geração de métricas de testes, execução de testes, montagem do ambiente, etc.

Porém, é importante salientar que a automação costuma ser um passo a ser dado, apenas após já termos um processo de Teste de *Software* bem estruturado e uma equipe preparada, pois ela muitas vezes exige um alto conhecimento técnico, principalmente para alguns tipos de testes específicos e é um esforço que precisa ser apoiado por um processo maduro.



# 1. AUTOMAÇÃO NO TESTE DE SOFTWARE.

## Quando automatizar?

Atualmente existem boas ferramentas (tanto livres quanto proprietárias) que podem nos auxiliar em diversas atividades do processo de Teste de *Software*, desde a revisão de documentos, até a execução de testes de desempenho.

Muitas vezes, sente-se a necessidade de ferramentas para automação de testes, quando acabamos despendendo muito tempo, por exemplo: criando ou formatando casos de teste, gerando métricas de testes, etc.

Ao pensar na automação dos testes, não podemos esquecer da automação dos testes unitários e integrados, que geralmente, são feitos pelos desenvolvedores e que já costumam ser executados de forma automatizada.

O problema é que a prática de criação de testes unitários e integrados não é muito comum, no âmbito do desenvolvimento de *software*, mesmo tendo resultados bem expressivos quando implantada.

Neste caso, a automação deve iniciar logo que a primeira linha de código for escrita, ou até mesmo (de preferência), antes mesmo da elaboração da primeira linha de código, como propõe a técnica TDD (*Test-driven development*).

Ao pensar em automatizar os testes de sistema, precisamos estudar a viabilidade da automação, quer dizer, se com a automação conseguiremos obter:

- **Ganho de tempo;**
- **Redução de custos;**
- **Manter a qualidade.**



# 1. AUTOMAÇÃO NO TESTE DE SOFTWARE.

A maior dificuldade está em conseguir medir a viabilidade da automação, pois é necessário analisar vários fatores, dentre eles:

- A maturidade do time e do processo de Teste de *Software*;
- O grau de reutilização dos testes automatizados;
- O nível de capacitação das pessoas para operar o *software*, ou manter o *software*, nos cenários em que a ferramenta ou *script* foi desenvolvido pelo próprio time;
- O conhecimento acerca do comportamento que se espera do sistema a ser testado;
- O tempo disponível para automatização dos testes. Lembre-se que a automação é um investimento a médio e longo prazo;
- O grau de frequência de mudanças das funcionalidades a serem verificadas, pois em alguns casos, pode não ser viável automatizar um teste de uma funcionalidade que irá mudar amanhã (embora isso não seja uma verdade absoluta, pois há exceções);
- A testabilidade do sistema. Em alguns cenários é necessário realizar alterações no código para que seja possível a automação dos testes. Isto ocorre de forma mais frequente, em sistemas legados, nos quais não houve uma preocupação com a testabilidade do sistema e o desenvolvimento não utilizava técnicas que favorecem uma boa testabilidade, como por exemplo, o TDD;
- A garantia de que com a automação do teste, poderemos alcançar a mesma qualidade da execução manual do teste;

# 1. AUTOMAÇÃO NO TESTE DE SOFTWARE.

## Quais são as razões para automação dos testes?

No livro Agile Testing, são citadas as seguintes razões:

- Teste manual é demorado;
- Reduzir a probabilidade de erros das tarefas de teste;
- Liberar tempo para fazer o trabalho da melhor forma;
- Prover uma rede de segurança ( se eu fizer uma mudança no código eu terei os testes, que irão me avisar se eu quebrei algo);
- Prover feedback cedo e frequente;
- Os testes que direcionarão a codificação (utilizando técnicas como o TDD e BDD) podem fazer mais do que isso (ex.: se tornam testes de regressão);
- Os testes provem documentação, aliás, são uma excelente documentação;
- ROI, com o passar do tempo o esforço gasto na automação dos testes se paga.
- Um dos grandes benefícios da automação é prover *feedback* cedo e frequente, e isso é possível, principalmente com a automatização dos testes unitários e integrados (que iremos ver com mais detalhes no próximo tópico), geralmente realizados pelos desenvolvedores. Ou seja, quando falamos de automação de testes, não estamos falando apenas da área de Teste de *Software*, mas sim em práticas que devem ser comum no desenvolvimento de *software* como um todo.



## 2. CONCLUSÃO

A automação pode trazer benefícios significativos para o projeto, quando utilizada adequadamente dentro do mesmo. Precisamos executá-la de forma incremental, sempre analisando quais testes que se automatizados trarão maior benefício à equipe.

Automatizar os testes não é algo simples, e diversas premissas são necessárias para que possamos trazer todos os benefícios da automação, pois caso contrário, ela irá trazer mais problemas.