

Aula 2

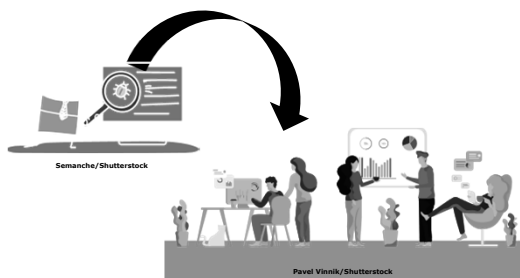
Testes de *Software*

Profª Maristela Weinfurter

Mindset de Testes

Conversa Inicial

Mindset de testes



O profissional de qualidade e testes

O profissional de qualidade e testes

- Ele deve se concentrar em aspectos macro, como testar recursos, integrações entre recursos e fluxos de trabalho de ponta a ponta (E2E)
- Não podemos, no entanto, encerrar os testes somente observando esses aspectos



O profissional de qualidade e testes

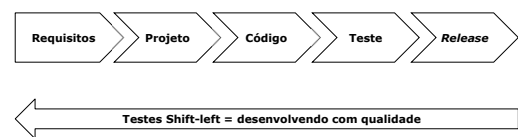
- Testar as características holísticas de qualidade do aplicativo, como segurança, desempenho, acessibilidade, usabilidade entre outros, para atingirmos o objetivo final de fornecer *software* de alta qualidade



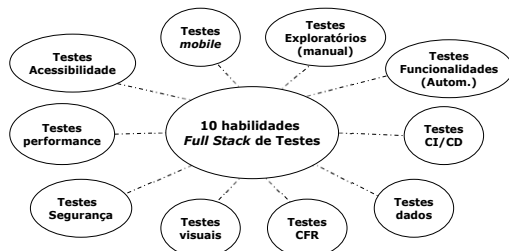
Camada de Stack de testes



Testes Shift-left



Stack de Testes e as 10 habilidades



- O Full Stack QA Engineer/Analyst precisa ter experiência e conhecimento nas 10 habilidades da Full Stack de testes



- O *Full Stack* também deve ter um excelente conhecimento de ciclo de vida do projeto de *software*, metodologias ágeis – como *Kanban* ou *Scrum*, preferencialmente
- Dentro dessas metodologias, ele deve ter experiência nas principais cerimônias: *retrospectiva*, *daily*, *planning* e *sprint*



- O ideal é ter uma certificação ISTQB, mesmo que seja de nível básico
- Conhecimento de uma linguagem de programação também é necessário, pois o conhecimento em codificação auxilia no entendimento mais completo do *software*



- Além do *Full Stack QA Tester*, temos algumas variações de profissionais na área de testes, como:
 - *Software QA Tester*
 - *QA Engineer*
 - *Software Tester*
 - *Mobile App Tester*



- *Game tester*
- *Regression tester*
- *Black box tester*
- *QA automation tester*
- *Web tester*
- *Software performance testing*



- O produto, ao ser lançado, vai ganhando experiência de uso com usuários em tempo real. Estes usuários dão como retorno *feedbacks* valiosos e que geram o aumento da usabilidade do *software*

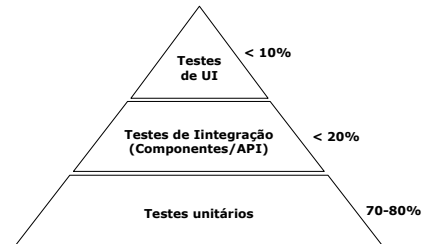


- Tanto o teste Alfa quanto o Beta já são efetuados por usuários reais e ambiente de produção real
- Tais versões facilitam também que o aplicativo seja executado por milhares de tipos de equipamentos, demonstrando também quaisquer problemas em relação a questões de *hardware* e infraestrutura



Estrutura de testes

Pirâmide de testes



Cobertura de Testes em Camada

01	<ul style="list-style-type: none"> Foco em soluções Validação de casos de uso do cliente e E2E Atualize com dados do cliente Teste de caixa preta 	Teste de Cenário do Cliente e E2E	
		Teste de Cenário do Cliente e E2E	Atualização com dados do cliente
02	<ul style="list-style-type: none"> Foco em todo o produto Validação do desempenho/habilidades do produto Teste de caixa preta 	Sistema, Integração e Teste de Habilidades	
		Performance	Usabilidade
03	<ul style="list-style-type: none"> Foco do subsistema Validação de funcionalidades Teste de caixa cinza/preta 	Testes Funcionais de Componentes e APIs	
		Smoke & Sanity	Impacted Regression
04	<ul style="list-style-type: none"> Foco do módulo Verificação de Interfaces Enquanto/teste de caixa cinza 	Processos de Desenvolvimento de Software	
		Testes Unitários	Revisões de Código

Estrutura de testes

- **Evolução de testes**
 - Internet das coisas
 - Big data
 - Móveis
 - Automação de testes
 - API/microserviços
 - IA



Gerenciamento de testes

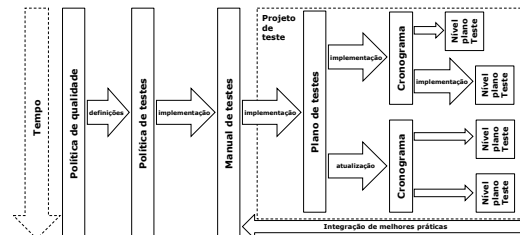
Gerenciamento de Testes

- Planejamento e controle de testes estão associados ao ciclo de vida do software
- A adoção de políticas de testes, sejam eles manuais, sejam automatizados, serve para quaisquer projetos de *software* elaborados dentro da organização

Gerenciamento de testes

- O plano do projeto de desenvolvimento de *software* e plano de testes possuem componentes básicos, bem como a implementação a ser seguida dentro de um ciclo de vida

Políticas de qualidade e níveis de plano de testes



Mentalidade do testador

Mentalidade do testador

- O *mindset* dos antigos analistas de testes buscava efetuar a tradução da documentação em casos de testes, sem contato com o código e aguardando que os desenvolvedores entregassem o sistema para o início dos testes e escrita dos *bugs* detalhadamente, pois não havia contato entre analistas de testes e desenvolvedores

Mentalidade do testador

- Com a abordagem ágil, o analista de testes muda seu foco
- Agora, ele analisa, testa e automatiza casos de testes, e também conhece e usa testes exploratórios, ajudando na identificação de riscos da aplicação ao escrever testes de aceitação
- Porém, as *user stories* ainda são provenientes dos engenheiros de QA

Mentalidade do testador

- O que chamamos hoje de *modelo pós-ágil*, permite que o analista de testes trabalhe imerso com os times de desenvolvimento, demonstrando como testar e incentivando a melhoria da qualidade
- Ao revisar o código, contribui com o desenvolvimento, incluindo na correção dos *bugs*
- Esse perfil de analista de testes também utiliza testes automatizados das várias camadas de cobertura de testes

Automação de testes

Automação de testes

- O volume de testes repetitivos de *software* aumenta sensivelmente em decorrência do número e diversidade de tipos de testes
- A automação de teste é um processo que se utiliza de outro *software* para automatização de ações manuais do usuário executadas no aplicativo ou de testes manuais
- É uma boa ideia validar o *software* com a ajuda de outro *software* no qual o teste manual não é possível, ou é demorado

Automação de testes

- O teste é uma atividade central em qualquer desenvolvimento de solução, e é independente das abordagens do ciclo de vida de desenvolvimento de *software* (SDLC)

Restrições em projetos de *software*



Automação de testes

- A automação de teste pode ser introduzida em diferentes estágios e fases do ciclo de teste, como:
 - Desenvolvimento de produtos ou implementação de soluções
 - Gerenciamento de testes
 - Testes funcionais e de regressão
- (...)

Automação de testes

- (...)
- Suporte
- Geração de testes
- Geração de dados de testes
- Inspeção e avaliação dos resultados dos testes
- Observância

- Automatizar testes de *software* segue processos, assim como o desenvolvimento do *software*, considerando-se projeto, desenvolvimento, execução e outras atividades, além da preocupação com a facilidade nas revisões de tais automações



Olivier Le Moal/Shutterstock