



Como é ser uma QA dentro de um time ágil?

Por Loraine B Garutti, QA Concrete

Quem é a Loraine?



O que significa ser QA?



- Ser responsável pela cultura de qualidade;
- Prezar pela qualidade desde o design da arquitetura;
- Ser uma das pessoas que mais conhece o produto;
- Conhecer todos os resultados esperados.

Analista de teste ou QA?

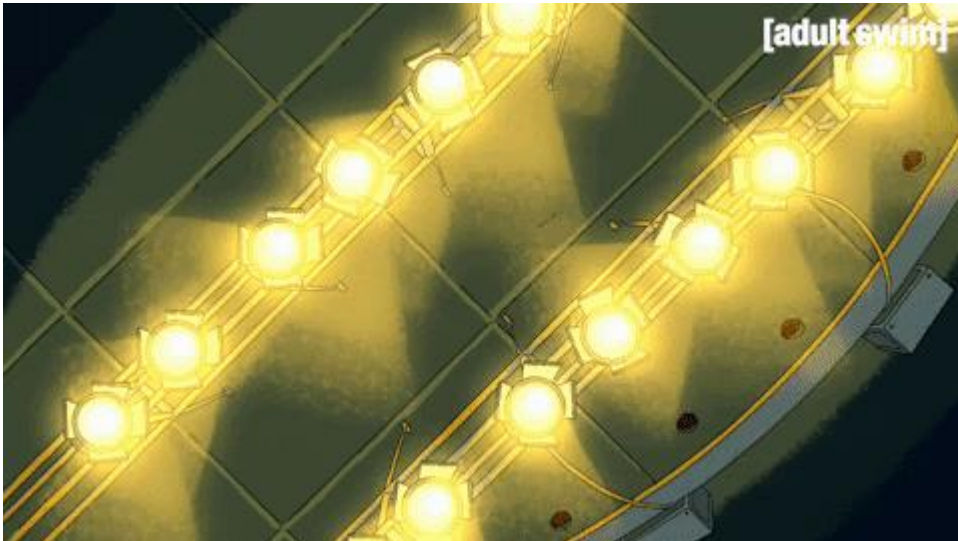
Analista de teste:

- Resistente a mudanças;
- “Bug hunter”;
- Teste no final do desenvolvimento;
- Luta entre analista de teste e desenvolvedores.

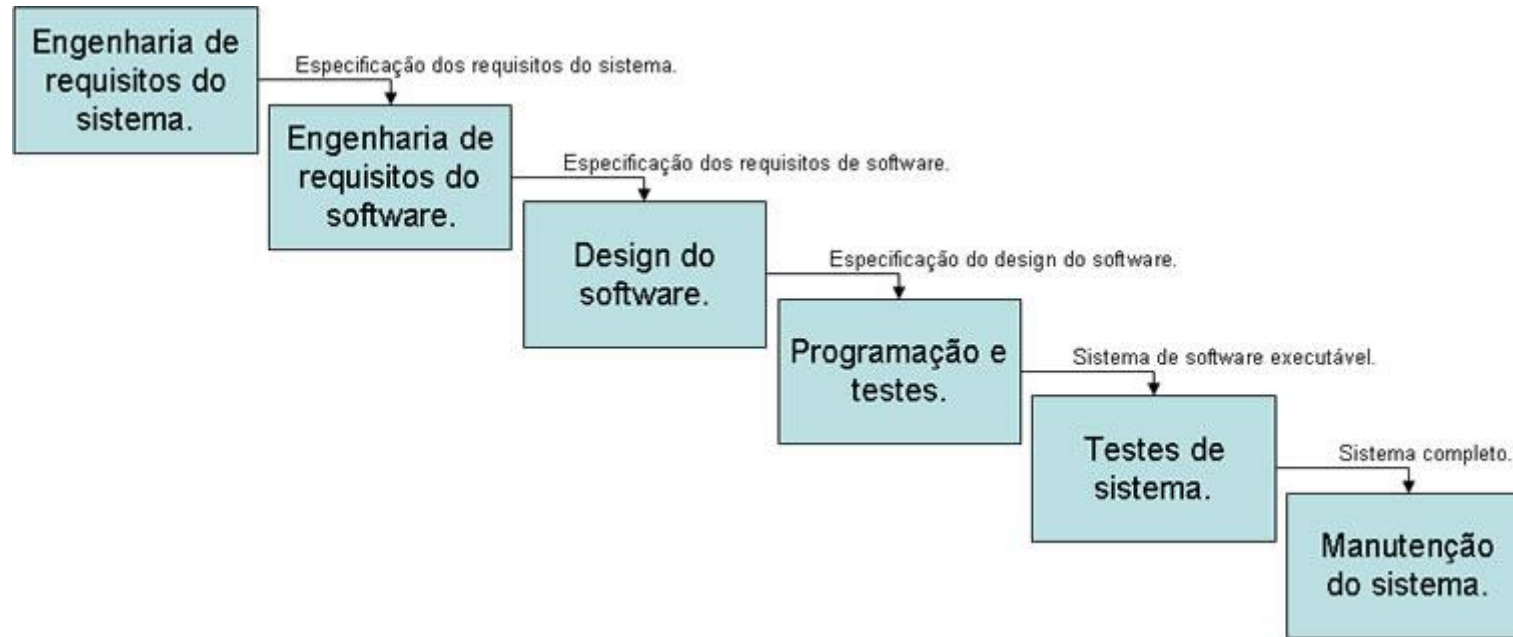
VS

QA:

- Abraça a mudança;
- Prevenção de problemas;
- Teste durante o desenvolvimento;
- Trabalho em conjunto com o time.



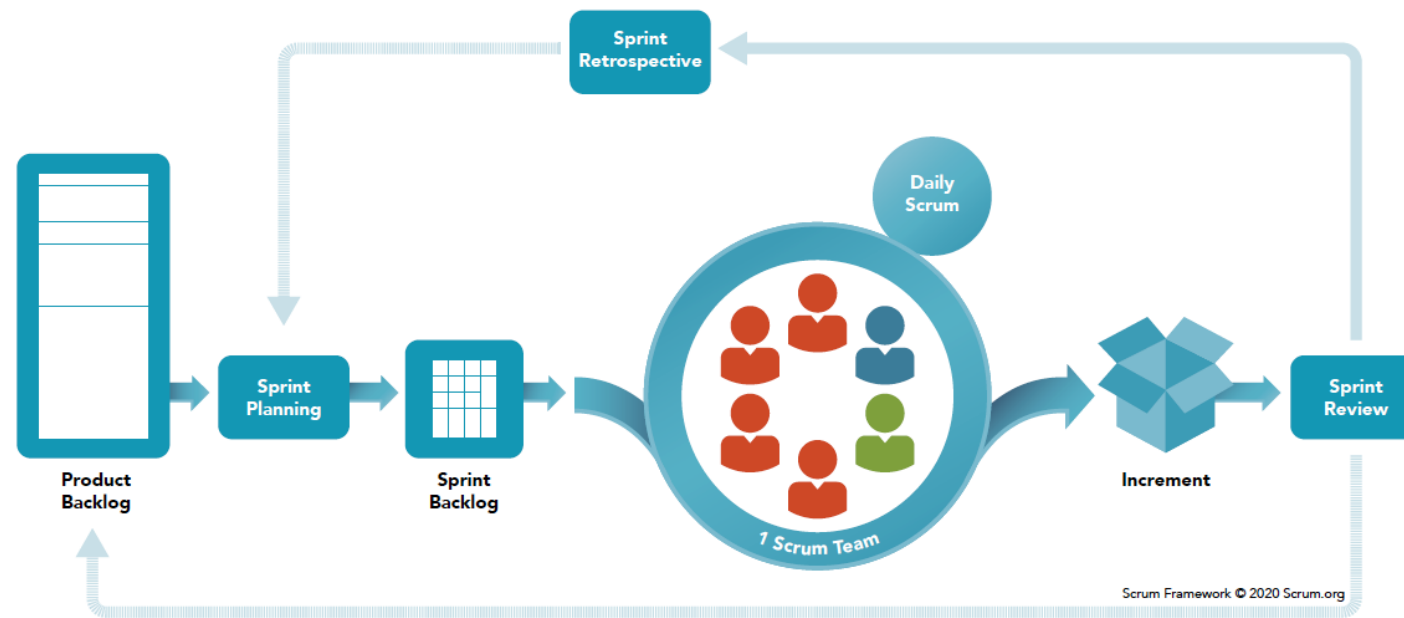
Qual é o fluxo “cascata”?



<http://engenhariadesoftwareuesb.blogspot.com/2012/12/fffrrrrr.html>

Qual é o fluxo de trabalho do Scrum?

SCRUM FRAMEWORK



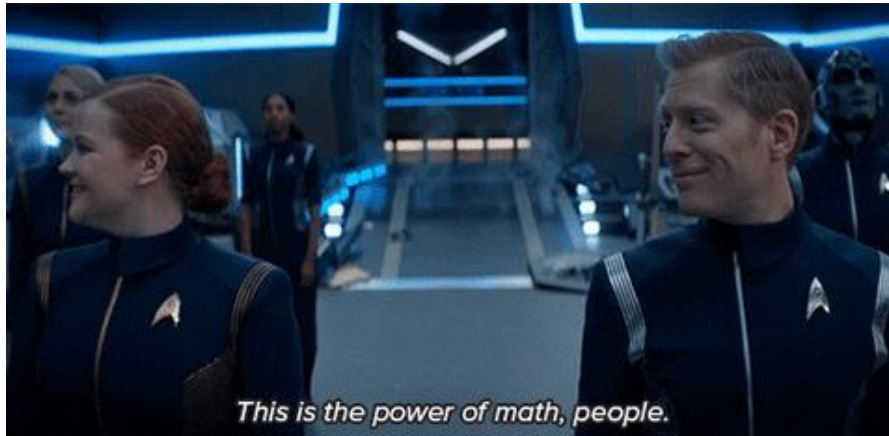
QA é desenvolvedora?

Segundo o Scrum?



QA é desenvolvedora?

E na prática?



Quais atividades fazer durante a sprint?



- Participação em refinamentos técnicos das próximas atividades a serem realizadas (refinements);
- Desenvolvimento de automatizações dos cenários;
- Testes iterativos/incrementais;
- Documentação simples e dinâmica.

O que fazer no final da sprint?



- Garantir que a demanda do usuário está sendo entregue (critérios de aceite) e que as tarefas estejam prontas (Definition of Done);
- Entregar automatizações dos cenários construídos durante a sprint;
- Apresentar métricas de saúde dos testes: testes unitários, testes de mutação, porcentagem de testes E2E automatizados passando, porcentagem de features com pelo menos um cenário automatizado.

Como adaptar ao incremental?



- Dinamismo;
- Errar, refletir, melhorar;
- Adaptação;
- Aprendizado constante.

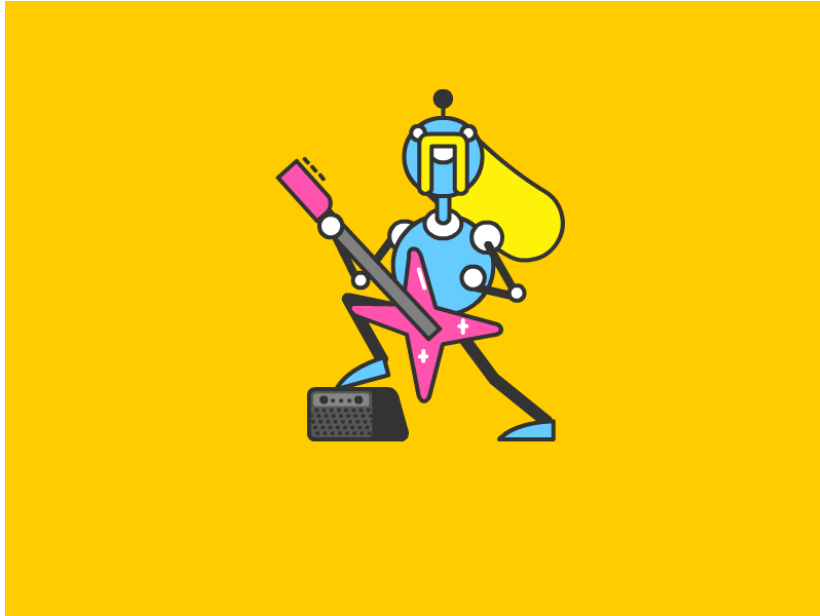


Kahoot



**O que é e por que
automatizar?**

O que é automação de testes?



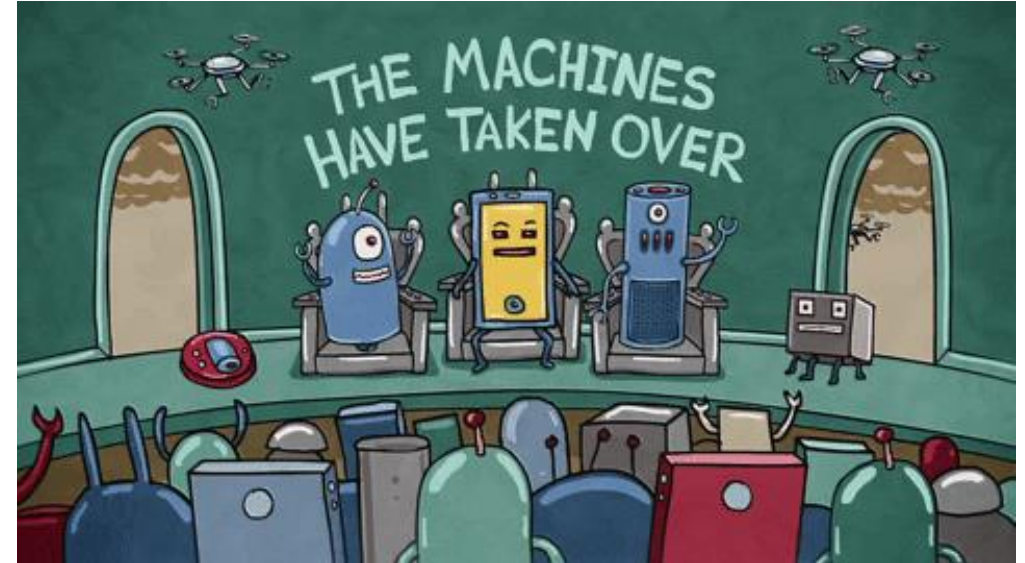
- Programar software para fazer os testes;
- Comparação de resultados esperados com resultados reais;
- Podemos iniciar de um caso de teste manual.

Por que automatizar testes?



- Aqueles casos de teste que temos que repetir todos os dias manualmente (regressivos): por que não automatizar?
- Certificar de que nenhuma funcionalidade crítica que já estava construída sofreu mudanças inesperadas de comportamento;
- Feedback rápido e incremental;
- O vício dos olhos (e do cansaço) humano;
- Redução de trabalhos manuais repetitivos: Podemos focar em processos criativos;
- Documentação viva.

Vamos automatizar tudo?



- Tempo investido na automação X Valor entregue;
- Ambiente desenvolvimento X homologação X produção;
- Maturidade do código do front-end e do fluxo da funcionalidade;
- Menos overengineering, mais praticidade e resultados.

Créditos: <https://medium.com/assertqualityassurance/automatizar-testes-piora-a-qualidade-da-sua-entrega-89e312c4adbc>

Ferramentas que a gente ouve falar



VS



VS



Playwright

VS



Suporta apenas Ruby,
API do Selenium
Webdriver para
controle da maior
parte dos
navegadores.

Suporta apenas Javascript.
API própria para controle de
vários navegadores.

Suporta Javascript, Java, Python,
.NET, C#. API própria para
controle de vários navegadores.

Suporta apenas
Javascript, API própria
para controle apenas do
Chrome.

Quais são as ferramentas que vamos utilizar nos próximos dias?



Cenários de teste em alto nível
(linguagem natural)



Script de teste em Ruby,
utilizando funções exclusivas
do Capybara



Webdriver



Navegadores



Kahoot



**Por onde podemos
começar?**

Pensar em testes antes do código

TDD

(Test Driven Development)

- Casos de teste escritos por desenvolvedores;
- Foco nos resultados esperados;
- Teste por módulo (componente);
- Para testes unitários;
- Escrever teste => executar teste => escrever código => executar teste => mudar código



BDD

(Behavior Driven Development)

- Casos de teste escritos pelos usuários ou pessoas mais envolvidas com o negócio
- Focado nos comportamentos esperados;
- Escritos em linguagem de alto nível;
- Para entender requisitos;
- Escrever comportamento esperado => escrever passo a passo (steps) => rodar os testes e falhar => escrever código para fazer os testes passarem

Vamos construir os cenários?

- Primeiro passo: arquivo “.feature”
 - Descrição da funcionalidade com “visão negócio”;
 - Diferentes cenários;
 - Linguagem natural / alto nível;
 - Escritos a partir de demandas do usuário, com auxílio de pessoas do negócio.

```
1  #language: pt
2
3  Funcionalidade: Login de usuarios
4      Eu como programador cadastrado
5      Quero fazer login no site
6      Para que eu possa fazer o upload dos meus projetos
7
8  @smoke
9  Cenario: Login
10      Dado que eu acesse a pagina de login
11      Quando eu submeter o login com username e senha
12      Entao serei redirecionada para minha pagina inicial
--
```

Vamos construir os cenários?

- Primeiro passo: arquivo “.feature”
 - Palavras-chave básicas:
 - “Funcionalidade”;
 - “Cenário”;
 - “Dado”: estado inicial para teste;
 - “Quando”: ação executada;
 - “Então”: resultado esperado.

```
1  #language: pt
2
3  Funcionalidade: Login de usuarios
4      Eu como programador cadastrado
5      Quero fazer login no site
6      Para que eu possa fazer o upload dos meus projetos
7
8  @smoke
9  Cenario: Login
10      Dado que eu acesse a pagina de login
11      Quando eu submeter o login com username e senha
12      Entao serei redirecionada para minha pagina inicial
--
```

Vamos construir os cenários?

- Primeiro passo: arquivo “.feature”
 - Palavras-chave mais avançadas:
 - Esquema do Cenário: reaproveitamento;
 - Exemplos: dados para teste.

```
14 @critical
15 Esquema do Cenário: Tentativa de login
16     Dado que eu acesse a pagina de login
17     Quando eu submeter o login com:
18         | username | <username> |
19         | senha    | <senha>    |
20     Entao devo ver um aviso "Incorrect username or password."
21
22 Exemplos:
23         | username      | senha      |
24         |                | abcde      |
25         | amandanogueira |            |
26         | amandanogueira | abcde      |
27         |                |            |
```

Vamos construir os cenários?

- Primeiro passo: arquivo “.feature”
 - Palavras-chave mais avançadas:
 - “E” e “Mas”: permitem leitura mais fluente do cenário.

```
12      Cenario: Buscar repositório
13          Dado que eu esteja na minha página inicial
14          Quando eu fizer a busca por repositório
15          Então eu verei o repositório correto
16          Mas não verei outro repositório
```

Vamos construir os cenários?

- Segundo passo: boas práticas.
 - Nome dos arquivos: “busca.feature”;
 - Cenários independentes;
 - Evite usar termos ligados a tela (busca de proximidade com as ações do usuário, e não técnicas);
 - Evite muitos “E” e “Mas”: uma linha por step!



Vamos construir os cenários?

- Segundo passo: boas práticas.
 - Imperativos X **declarativos**:

Cenário:

Fazer uma retirada de uma conta com fundos

Dado que minha conta esteja ativa

E que possua R\$100,00 de saldo

E que inseri meu cartão no ATM

E informei minha senha

E pressiono "ok"

Quando faço uma retirada de R\$50,00

Então devo receber R\$50,00

E o meu novo saldo deve ser R\$50,00

vs

Cenário: Fazer uma retirada de uma conta com fundos

Dado que esteja logado na minha conta

E que possua R\$100,00 de saldo

Quando faço uma retirada de R\$50,00

Então devo receber R\$50,00

E o meu novo saldo deve ser R\$50,00



Kahoot

Vamos construir os cenários?

- Terceiro passo: escrever os cenários.



Vamos construir os cenários?

- O que construímos:
 - Feature: Login
 - Cenário 1: Fazer login com conta válida;
 - Cenário 2: Fazer login com dados inválidos;
 - Feature: Carrinho de compras
 - Cenário 1: Adicionar item ao carrinho;
 - Cenário 2: Ser possível escolher opção de entrega;
 - Cenário 3: Ser possível escolher opção de pagamento;
 - Cenário 4: Comprar item.



Vamos construir os cenários?

- O desafio:
 - Feature: Cadastro.



Vamos preparar o ambiente?

