



igti

—

Devops

Capítulo 4. Qualidade

Prof. Thiago Chierici



Aula 4.1. As estratégias para falhar e outras para obter sucesso



- ☐ A importância da estratégia de testes.
- ☐ O que fazer para a sua estratégia falhar.
- ☐ O que fazer para a sua estratégia dar certo.



Testes manuais exploratórios resolvem?



- É comum associarmos Devops às ferramentas e automação.
- Sem uma estratégia bem definida, nossos testes não seriam eficientes.
- Não basta pegar o que algum 'case' utilizou e tentar aplicar.

```
const DevOps = require('./testing.strategy');
```

O que pode fazer dar errado

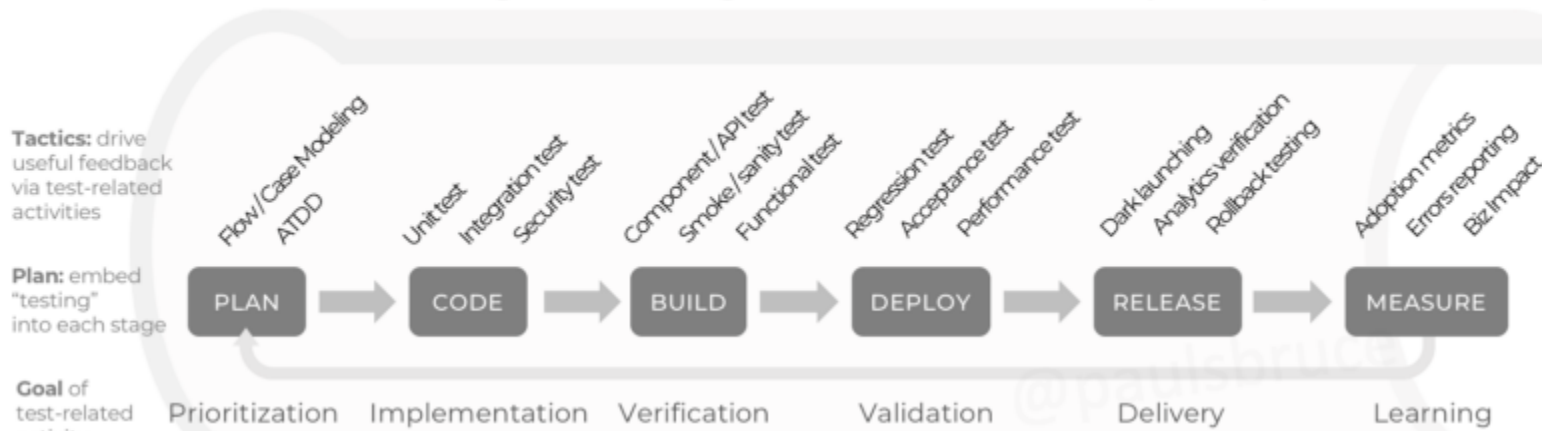
- Testes manuais: não resolvem com Devops.
- Precisamos de bastante esforço para implementar uma boa estratégia de testes automatizados.
- E o custo de não fazer a automação?
 - Pode ser difícil de calcular todos os impactos, mas é preciso ter a noção.



O que ajuda a fazer dar certo

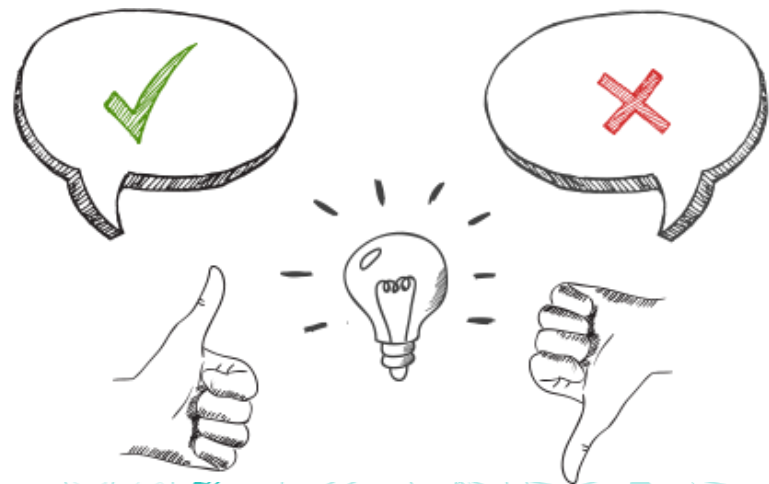
- Alinhar o planejamento dos testes com as prioridades de negócio.
- Montar uma visão prática dos testes, quando e como são feitos e executados.

Strategic Testing Across the DevOps Pipeline



Fonte: <https://medium.com/capital-one-developers/no-testing-strategy-no-devops-915287e1b4fd>

- Quebrar grandes testes em pacotes menores.
- Separar os testes rápidos dos demorados.
- Paralelize a execução dos testes.
- Desabilite testes de baixo risco.



- ☑ Vimos que sem uma estratégia bem definida, os testes não são eficientes e nossa qualidade pode ser comprometida.
- ☑ Listamos algumas diretivas que podem contribuir e outras que podem prejudicar seus projetos.



- ❑ Vamos equalizar o entendimento da pirâmide dos testes e o que e como deve ser automatizado.





Aula 4.2. Pirâmide de testes e automação



Nesta aula



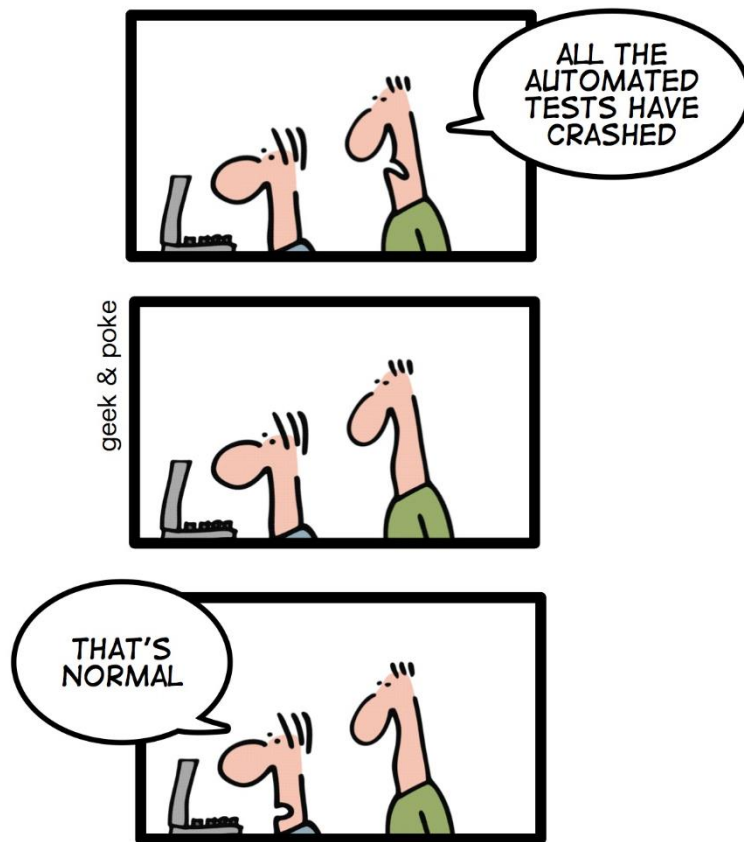
- ❑ Pirâmide de testes.
- ❑ Automação dos testes.



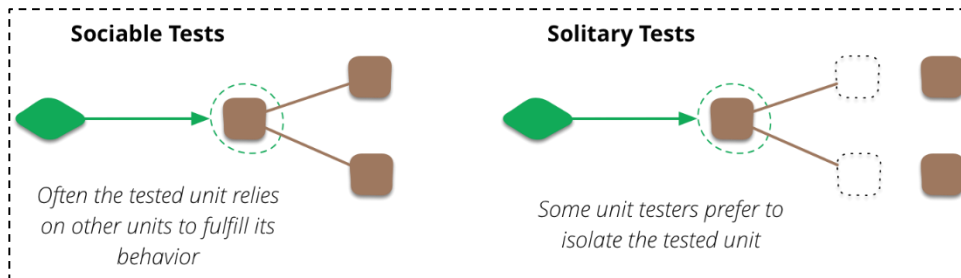
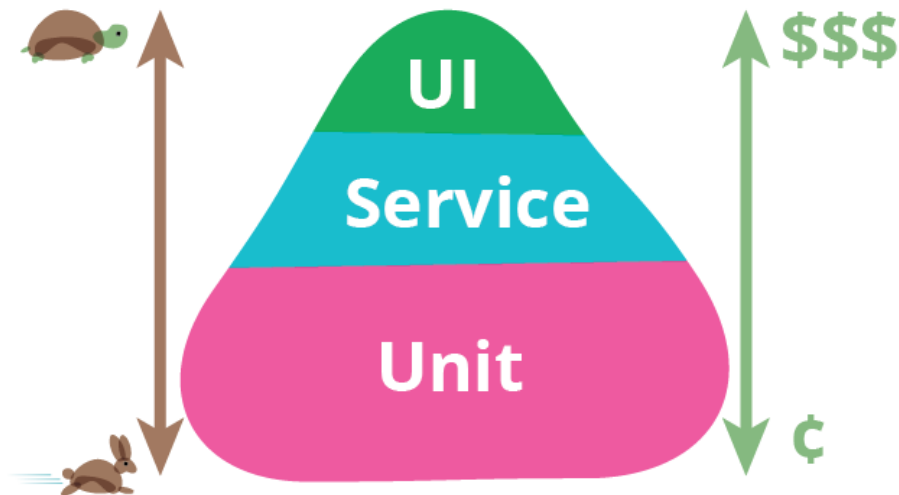
- “Integração contínua serve para que saibamos que a situação está normal, dentro do esperado.”
 - Geek and Poke.



Normal?!

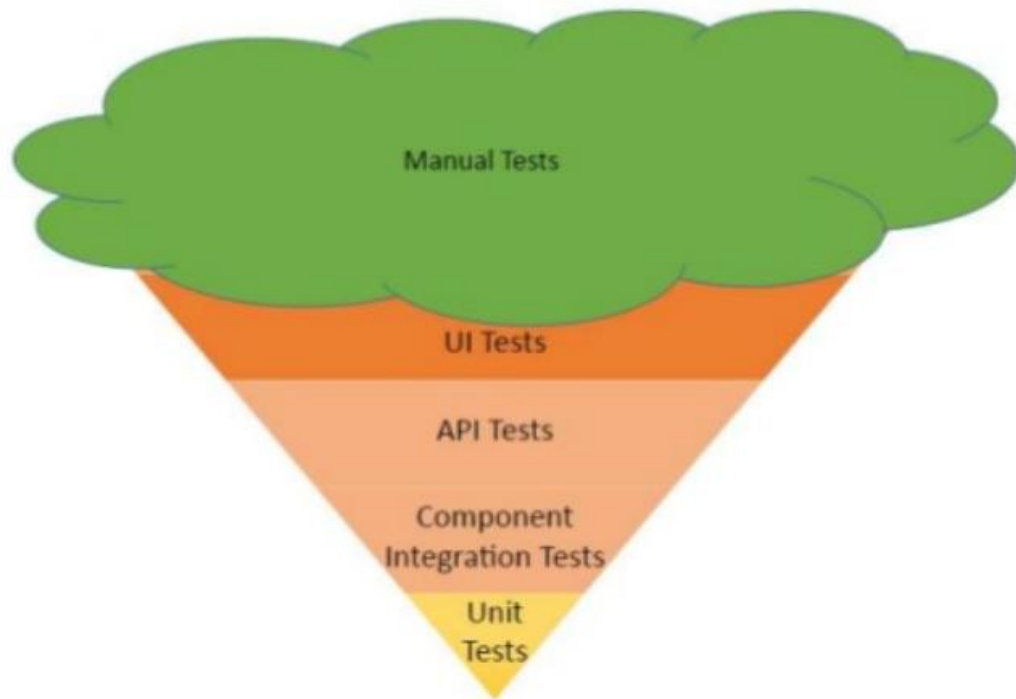


A pirâmide

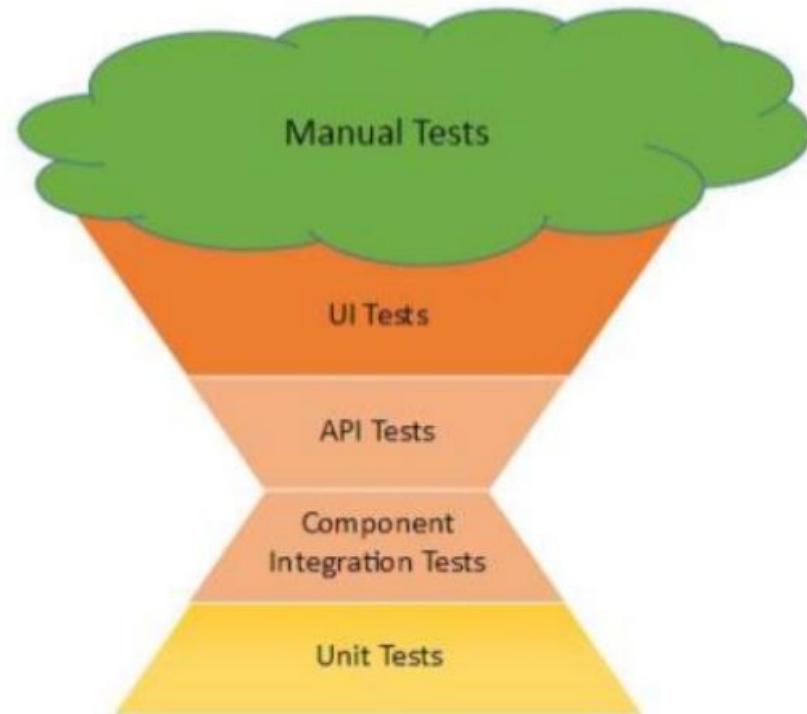


Fonte: <https://martinfowler.com/bliki/TestPyramid.html> <https://martinfowler.com/bliki/UnitTest.html>

Ice cream cone

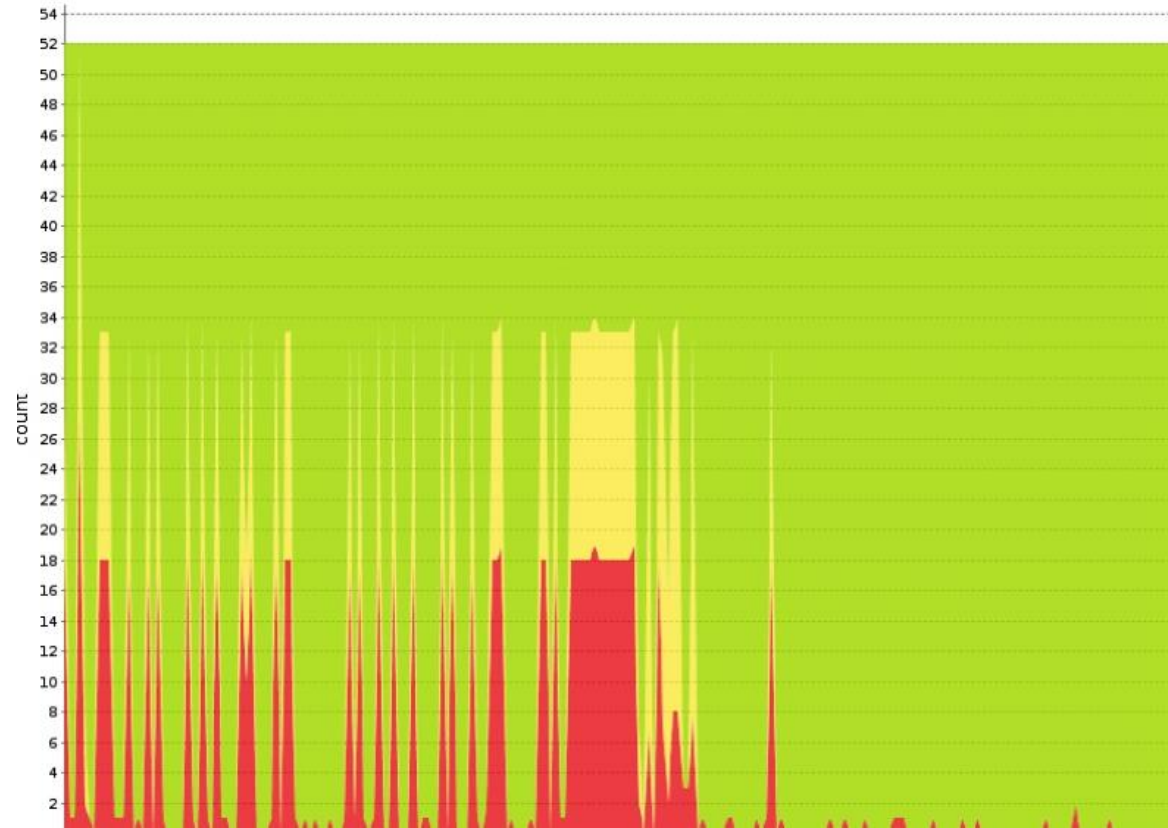


Hour-glass



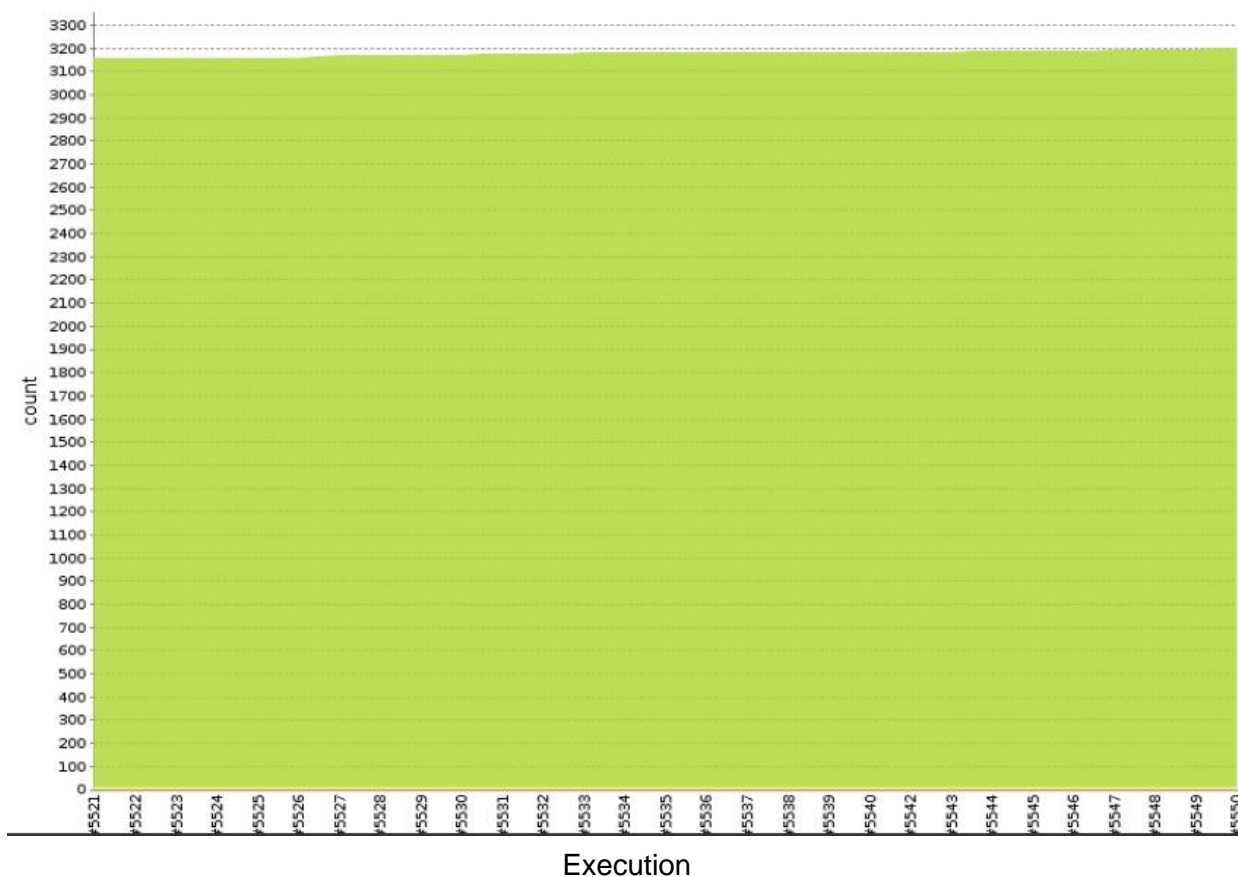
Fonte: <https://pt.slideshare.net/TAlexanderLystad/test-automation-pyramid>

Testes automatizados - Integrados



Fonte: <https://pt.slideshare.net/thiagoghisi/the-test-pyramid>

Testes automatizados - Unitários



- ☑ Vimos a importância de conhecer a pirâmide e buscar uma base maior do que o topo.
- ☑ Vimos como os testes integrados são mais caros e menos eficientes que os unitários.



- ❑ Vamos aprofundar em como testar alguns dos principais requisitos não funcionais.





Aula 4.3. Requisitos não funcionais no Devops



- ❑ Como gerenciar e validar os RnFs em um processo Devops.



- *“Requisitos não-funcionais são os requisitos relacionados ao uso da aplicação em termos de **desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas**. Estes requisitos dizem respeito a como as funcionalidades serão entregues ao usuário do software.”*
 - Vasquez, 2016.

Em uma palavra...

IGTi



- Tipos de inaptidão:
 - Visual.
 - Auditiva.
 - Motora.
 - Cognitiva.
- Principal referência: Web Content Accessibility Guidelines (WCAG).
- 116 ferramentas disponíveis:
 - <https://www.w3.org/WAI/ER/tools/>



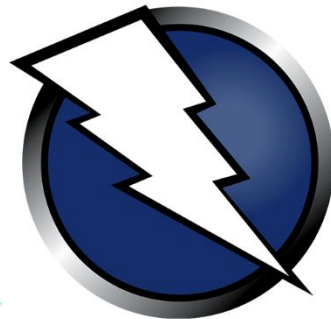
- Um dos mais comuns de se validar de forma automatizada.
- Jmeter:
 - Grátis.
 - Muitos recursos.
 - Fácil de integrar na CI.
 - Conheça também a Blazemeter.
- Outros tipos de ferramenta podem ser úteis:
 - Service mocking, Performance Profiling, etc.



- Disponibilidade:
 - Ao invés de monitorar o código-fonte, monitorea a própria produção.
- Tempo de resposta:
 - Com o mesmo tipo de ferramenta, também podemos observar outras métricas como tempo de resposta.
- Teremos um capítulo mais dedicado ao monitoramento.
- Muitas ferramentas online com essa finalidade.



- Outro RnF validado com frequência em pipelines Devops.
- ZAP – Zed Attack Proxy:
 - Baseada nas recomendações OWASP.
 - Ferramenta gratuita com uma enorme gama de funcionalidades.
 - Também pode ser integrada com facilidade em um pipeline Jenkins ou similar.



- ☑ A automação é crucial para conseguirmos validar continuamente os requisitos não funcionais de um sistema.



- ❑ Especificação executável, ATDD e BDD.





Aula 4.4. Especificação executável



- ☐ Especificação executável – ATDD.
- ☐ Especificação por exemplo.
- ☐ BDD.



- “Nada é permanente, exceto a mudança.”
 - Heráclito.
- Então como documentar?



- Uma documentação errada é pior do que aquela que nem existe.



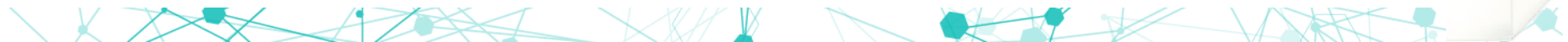
- Uma forma de especificar que pode ser reutilizada de forma automatizada em outras etapas do processo de desenvolvimento.
- Tenta garantir a entrega correta na primeira tentativa.
- Diretamente relacionada ao ATDD.



- Uma forma de descrever os requisitos baseada em cenários reais do negócio, chamados de exemplos.



- Exemplos precisam ser precisos.
- Exemplos precisam ser completos.
- Exemplos precisam ser realistas.
- Exemplos precisam ser fáceis de entender.



- O frete grátis é oferecido aos clientes VIP que compram certa quantidade de livros. O frete grátis não é oferecido aos clientes normais, nem para clientes VIP que comprem outros produtos que não sejam livros.

Examples

Customer type	Cart contents	Delivery
VIP	5 books	Free, Standard
VIP	4 books	Standard
Regular	10 books	Standard
VIP	5 washing machines	Standard
VIP	5 books, 1 washing machine	Standard

Fonte: Adzic (2011)

Exemplo Gherkin (Behat ou Cucumber)



```
# language: pt
```

Funcionalidade: Algum texto descritivo conciso do que é desejado

A fim de realizar um valor de negócio

Como ator explicito do sistema

Eu quero ganhar algum resultado benéfico que promova a meta

Texto adicional...

Cenário: Uma determinada situação de negócios

Dado uma pré condição

E uma outra pré condição

Quando uma ação é feita pelo ator

E uma outra ação

E outra ação diferente

Então um resultado testável é alcançado

E outra coisa que possamos verificar também acontece

Cenário: Uma situação diferente

...

Fonte: <https://docbehat.readthedocs.io/pt/v3.1/guides/1.gherkin.html>

- Estamos hablando de BDD (Behavior Driven Development)?



- Chef:
 - Ferramentas para testes unitários e integrados.
 - Exemplos:
 - ChefSpec.
 - RuboCop.
 - FoodCritic.
 - ServerSpec.
 - TestKitchen.
 - Cucumber(-).

- ☑ A especificação executável pode funcionar como ponto central de definição, implementação e testes.
- ☑ A especificação por exemplos pode facilitar a compreensão das regras de implementação.
- ☑ A automatização da especificação pode multiplicar o retorno para um esforço que já seria feito.
- ☑ Utilizando ferramentas relacionadas não significa que estamos fazendo BDD.



- ❑ Mais um grande case Devops de mercado, dessa vez com foco na qualidade.





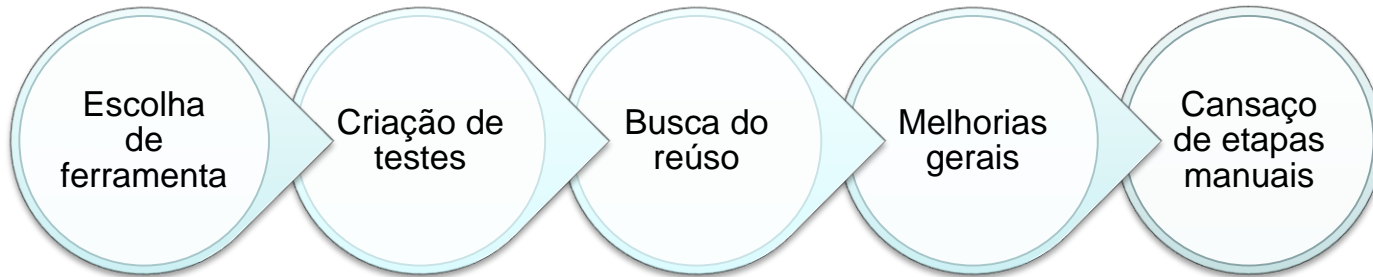
Aula 4.5. Inspiração – IBM: Repensando os testes automatizados no mundo Devops



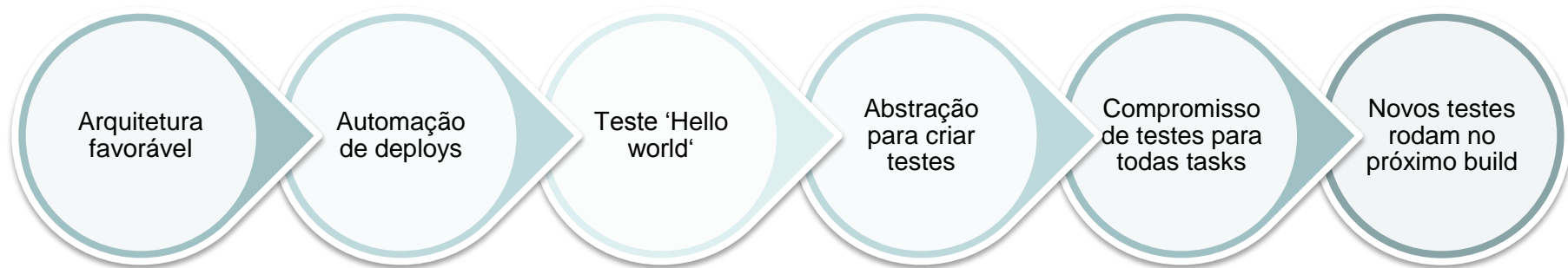
- ❑ Visão da comunidade IBM sobre a melhor forma de evoluir sua abordagem de automação de testes com Devops.



- É possível observar uma tendência (errada) sobre como começar a automatizar testes.



- Foco em processos Devops antes de ferramentas de testes.



- ☑ Precisamos mudar a forma de pensar ao pensar de testes automatizados, focando em ativar um ciclo Devops mínimo antes de escolher uma ferramenta de automação.



Próxima aula



- ❑ Novo capítulo.
- ❑ Ciclos contínuos.

