

QA: Garantindo qualidade nos projetos de software



Andrei Rosseti
andrei@facedigital.com.br



Let's talk about . . .

- Definição de qualidade de software
- PDCA
- Falha comum entre os stakeholders nos projetos (cliente, gerente e equipe)
- 4 motivos para melhorar a qualidade nos projetos
- Workflow alfa (cenário atual)
- Workflow omega (visão de futuro)
- Vantagens do workflow omega
- Ferramental
- Hands on

Definição de qualidade de software

“Qualidade de software é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a **padrões de desenvolvimento** claramente **documentados**, e a **características implícitas que são esperadas** de todo software desenvolvido por profissionais”

– Pressman

Definição de qualidade de software

Requisitos

- Funcionais
- Não funcionais

Métodos

Unified Process,
Scrum, XP, FDS.

Processos

Conjunto de
atividades
coordenadas com a
finalidade de obter
produto de software

Técnicas e Padrões de desenvolvimento

Controle de versão,
PSR, Utilização de
frameworks e etc.

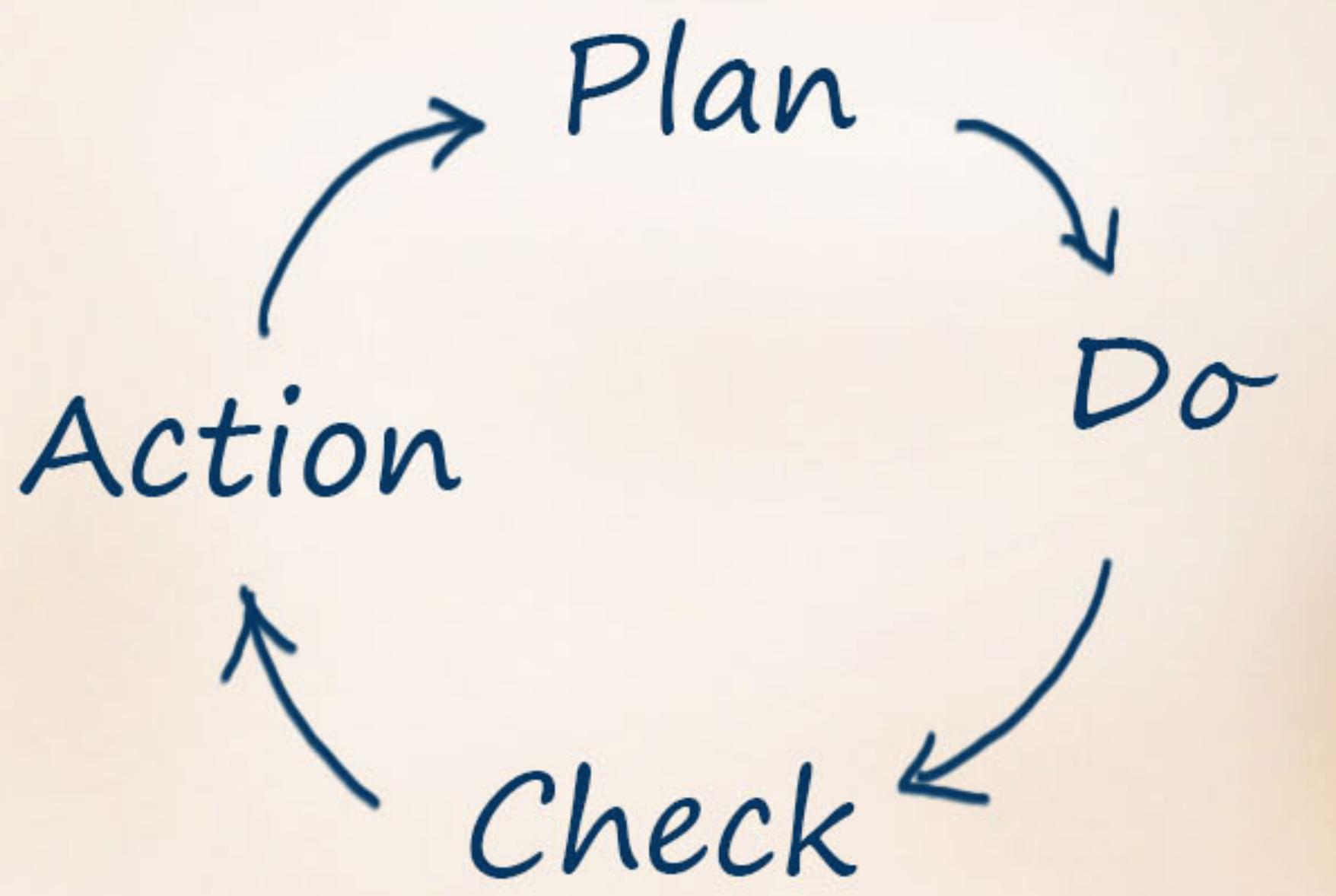
Produto de Software

Garantia de qualidade

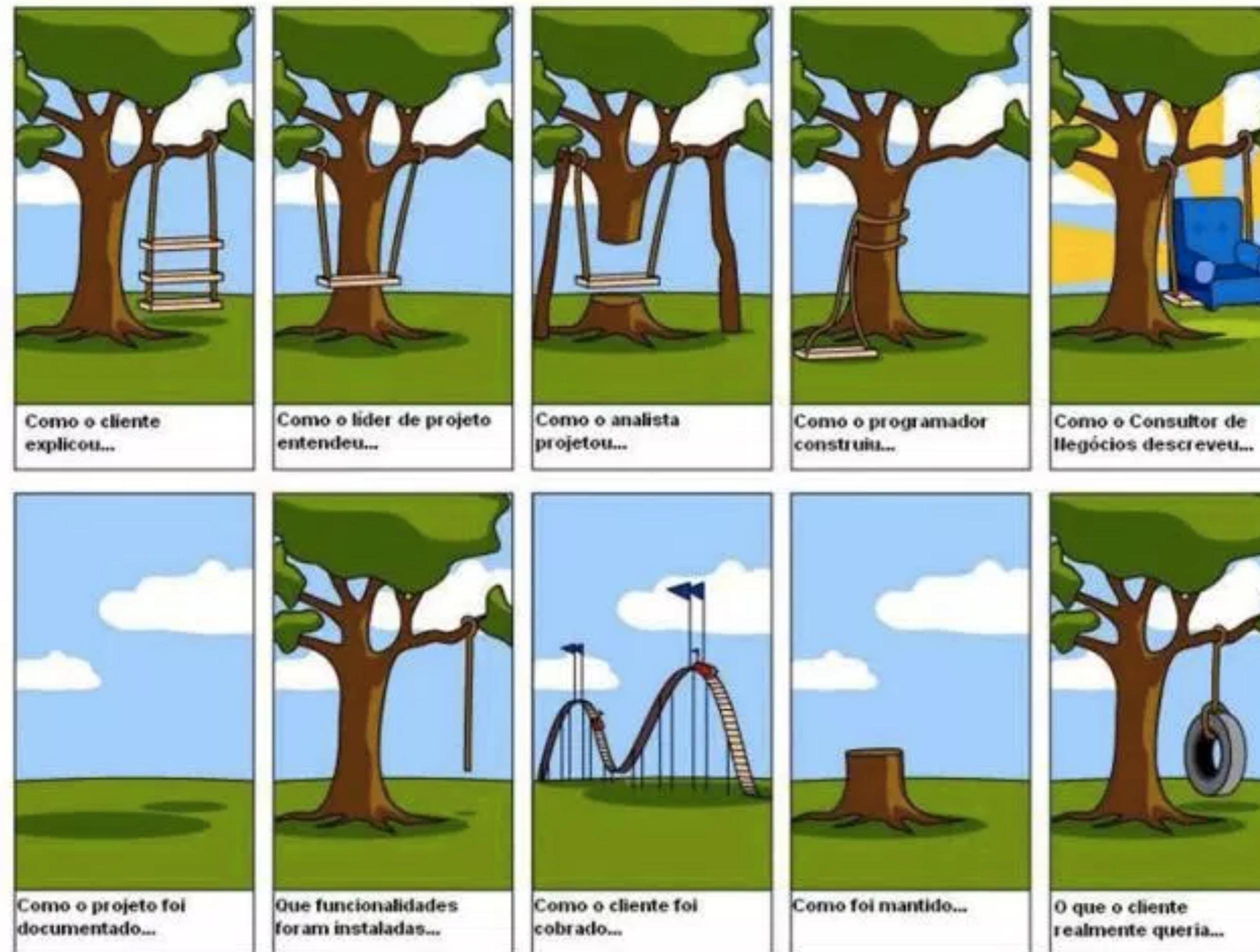
Funções gerenciais de auditar e relatar

Controle de qualidade

Inspeções, revisões e testes usados ao longo do processo de software



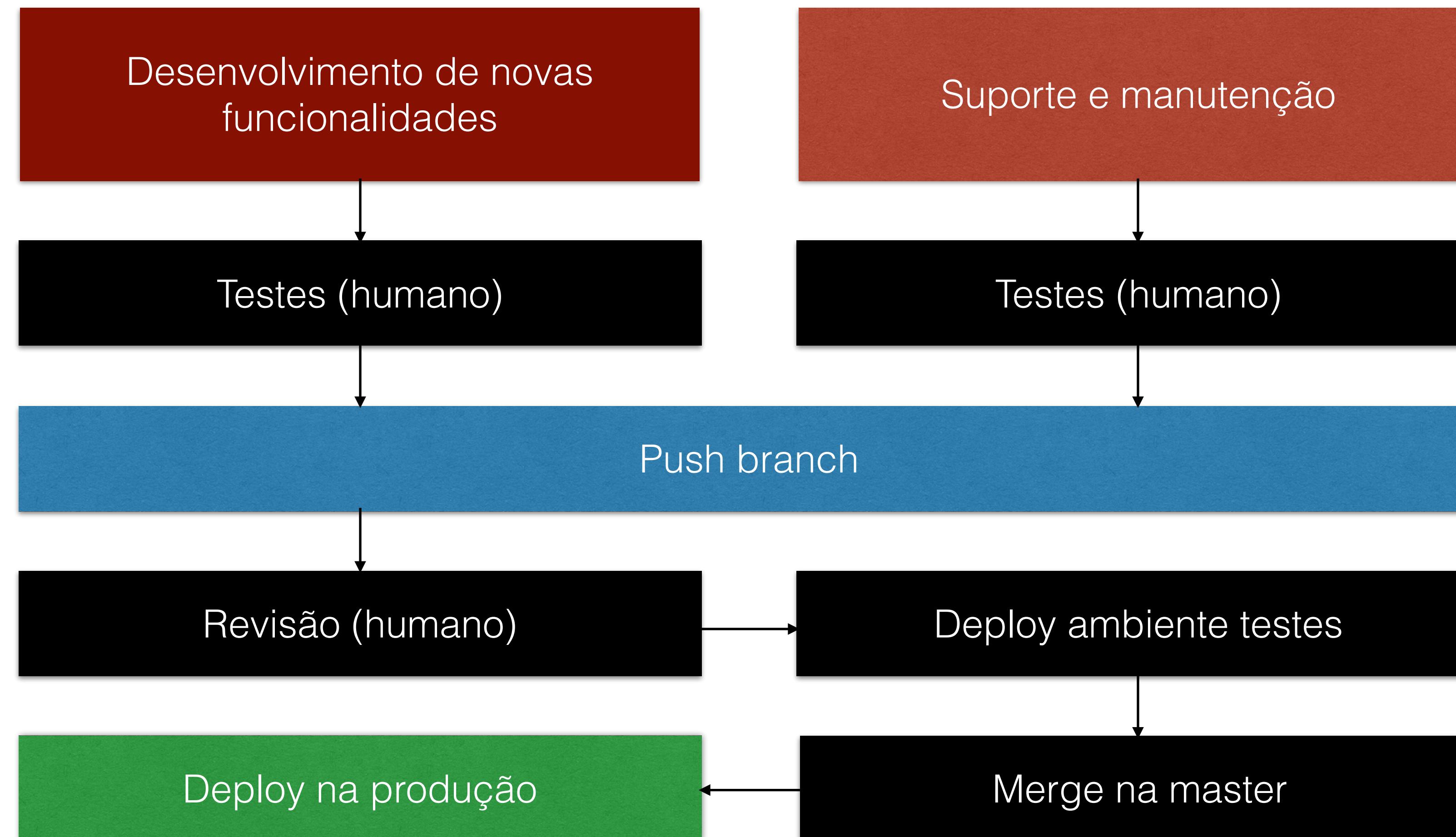
Falha comum entre os stakeholders nos projetos (cliente, gerente e equipe)



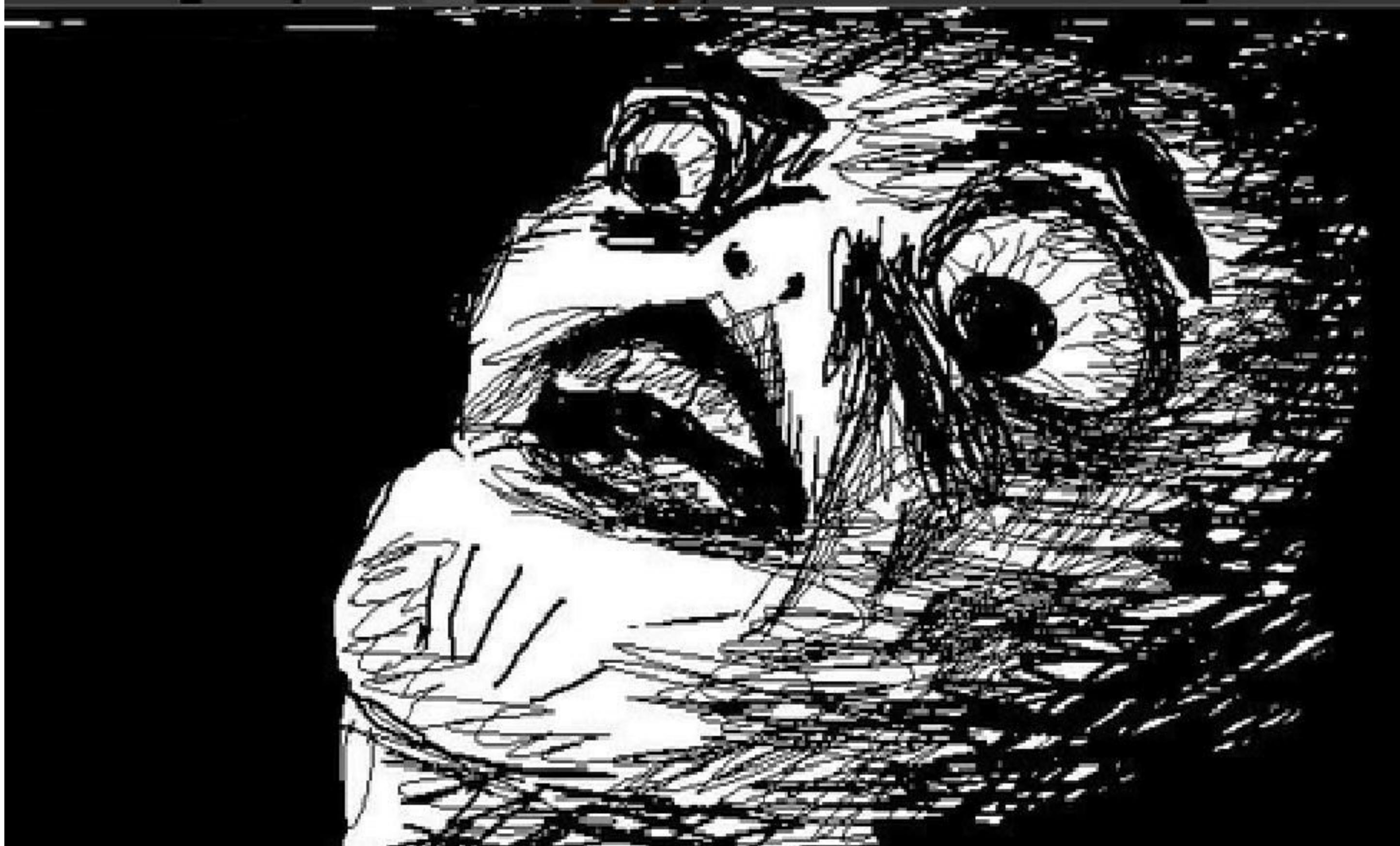
4 motivos para melhorar a qualidade nos projetos

- Proteger nossa reputação como empresa
Veja que para construir a reputação é demorado, para destruir, basta lançar no mercado um produto cheio de problemas (sem ter o cuidado mínimo), reputação cai pra 0.
- Aumentar a satisfação do cliente
Por atender os requisitos com excelência
- Detectar e corrigir problemas através de processos sistêmicos
- Economizar tempo e dinheiro
É o grande sonho de toda organização :-)

Workflow alfa



**** Problemas:** Chances de erros são altas, problemas com conflitos, lembrar das regras de negócio...



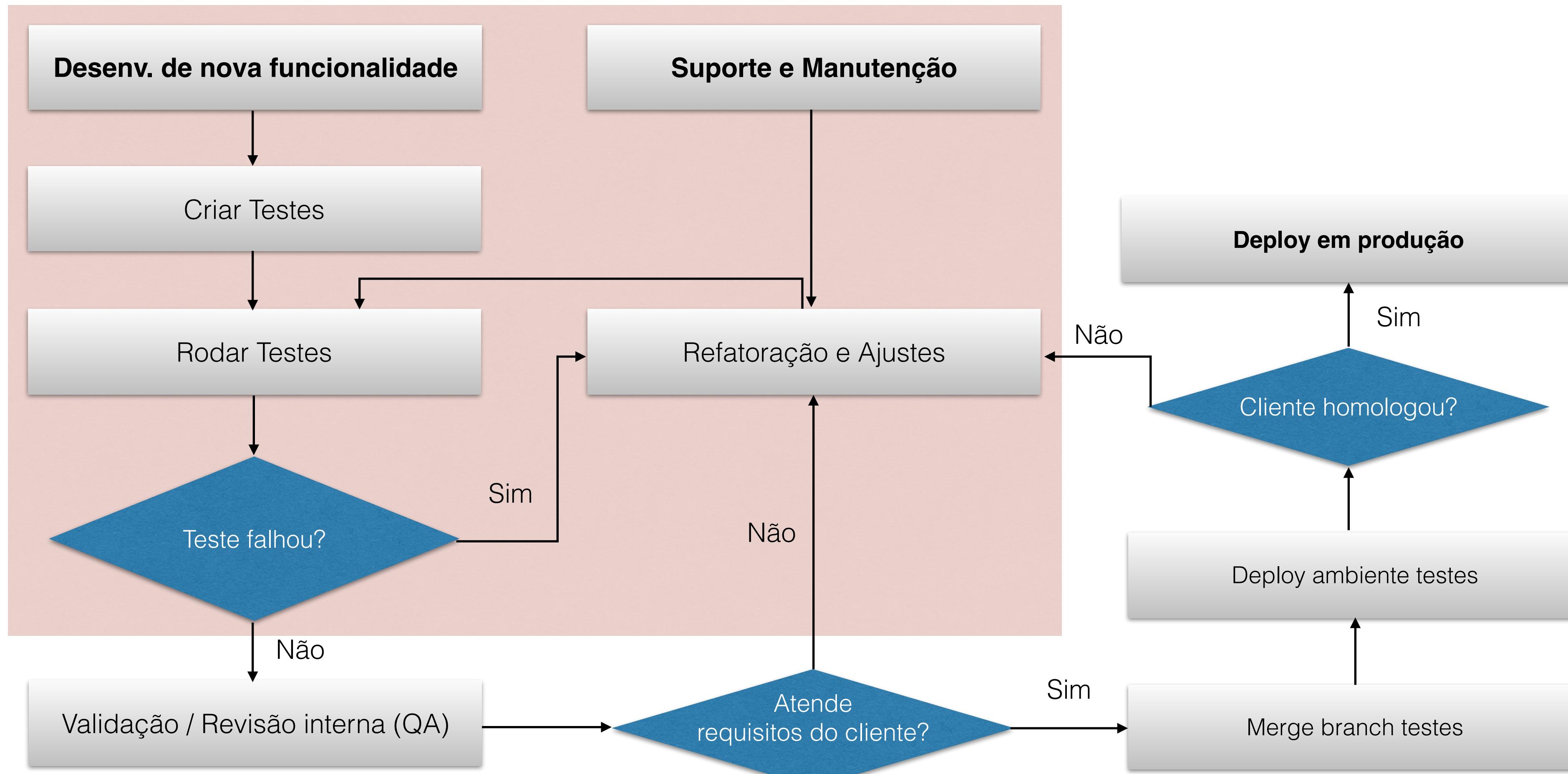
Deploy com git?

Nãããããão!

Quais técnicas podemos utilizar?

- *TDD - Test Driven Development*
Cria funcionalidade, teste falha, ajustes, teste OK
- *Pair Programming*
Essa técnica podemos começar a utilizar para problemas mais complexos, onde o par fica revesando num intervalo de 15 minutos (ou a definir, ciclo pequeno) até chegar na solução.
Piloto e copiloto.

Workflow omega



Vantagens do workflow omega

- Manter as 2 linhas de atualização do projeto sincronizadas (new features & bug fixes)
- Automatização de testes de interface para auxiliar no processo de integração contínua
- Diminuir chances de erros para o cliente
- Possibilitar integração contínua (phPCI)

E se já tivesse TDD não teria isso...

```
< > usuario_model.php ×
191     $res = $this->db->get($this->table_use
192
193     return $res->result();
194 }
195
196 public function get_vendedores()
*-----*
```

OPEN < > prod

```
prod < > produto_model.php ×
203     $data = array_unique(array_merge($data, $gn_produ
204
205     return $data;
206 }
207
208 public function get_vendedores($id_produto = NULL)
*-----*
```

Ferramental

phpci



Integração contínua
Requisito: php + composer

+

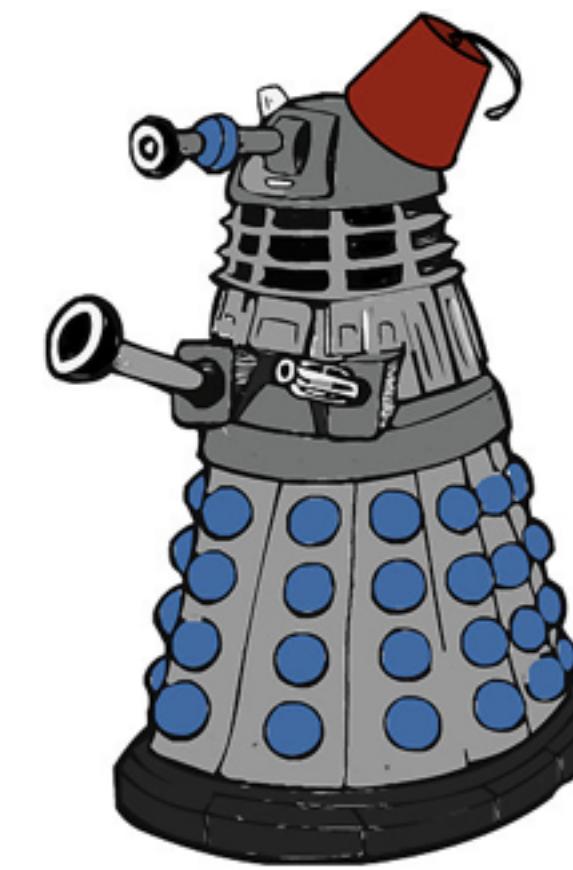


Testes unitários
Requisito: php + phpunit

PHPUnit

DalekJS

Capistrano



+



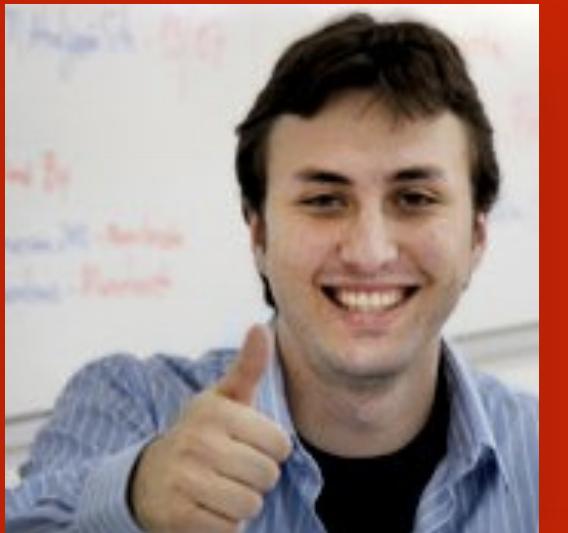
Testes de interface
Requisito: npm

+

Deploy tool
Requisito: rvm ou ruby

Hands on:

<https://github.com/facedigital/techlunch-qa>



Andrei Rosseti

andrei@facedigital.com.br

Referencias

- R. S PRESSMAN. Engenharia de Software, volume 6.ed. São Paulo: McGraw-Hill, 2006.