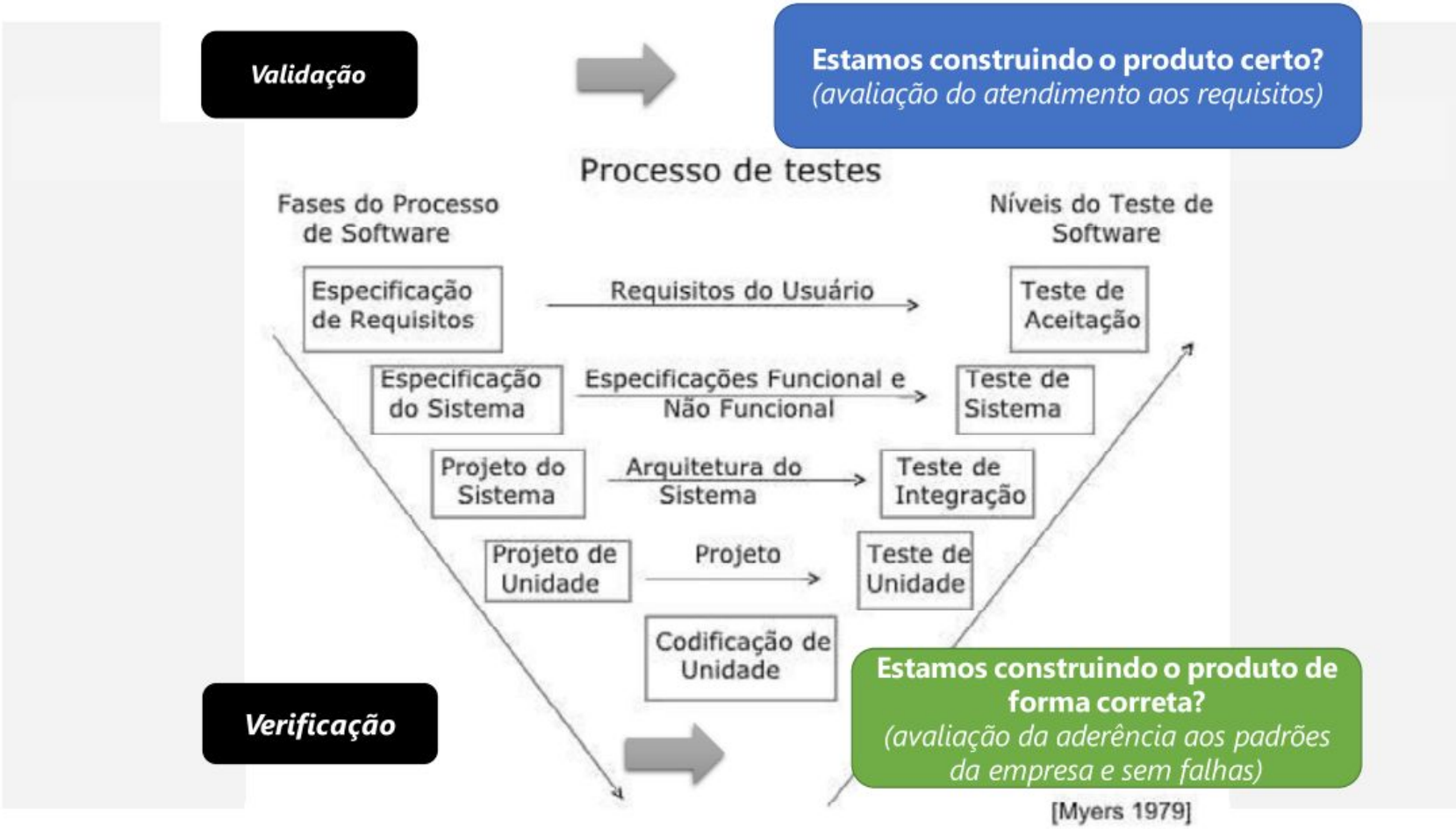
Engenharia de Software I

Testes de software

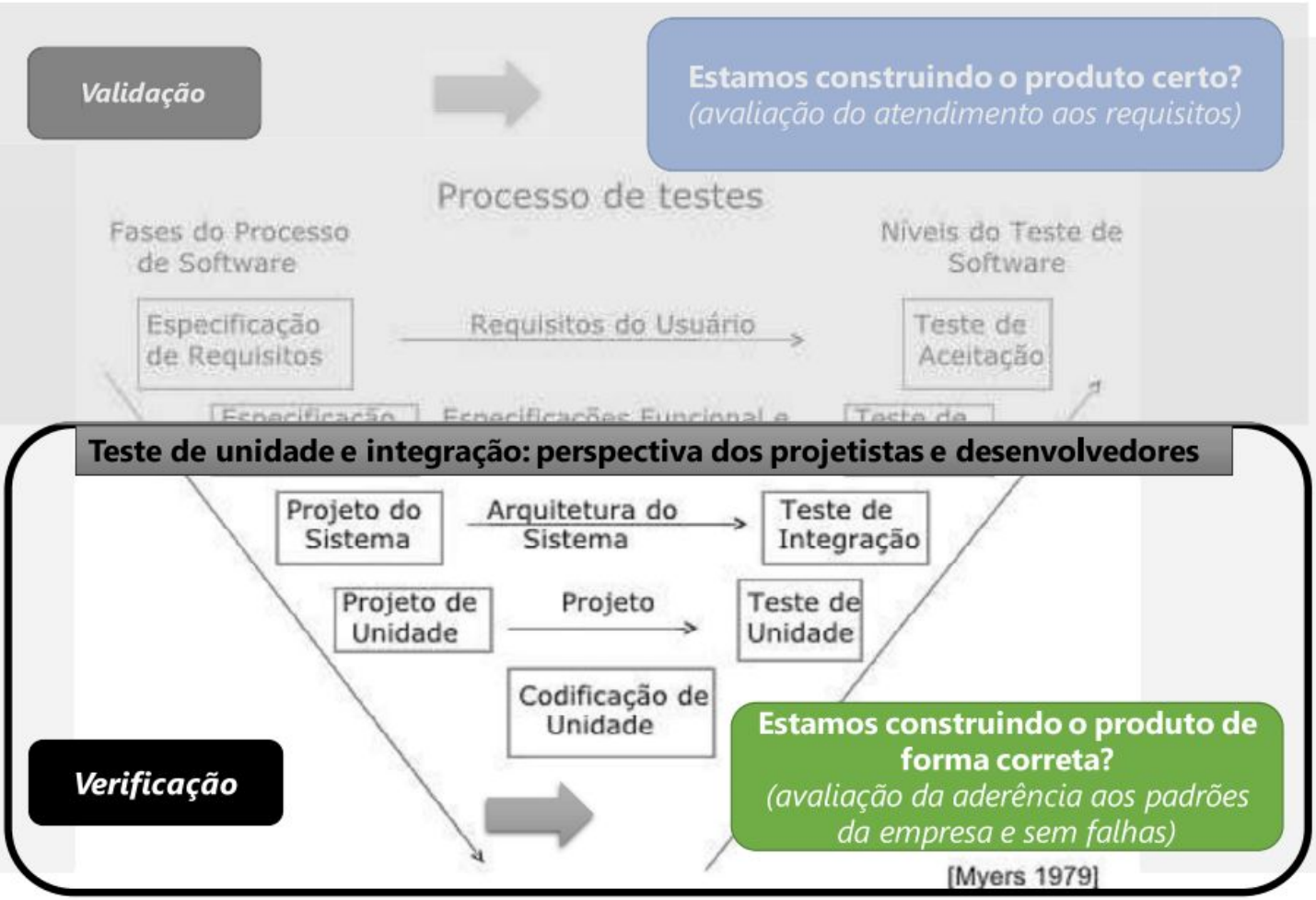
UNIVERSIDADE FEDERAL DA FRONTEIRA SUL

PLANO DE TESTES - Modelo V



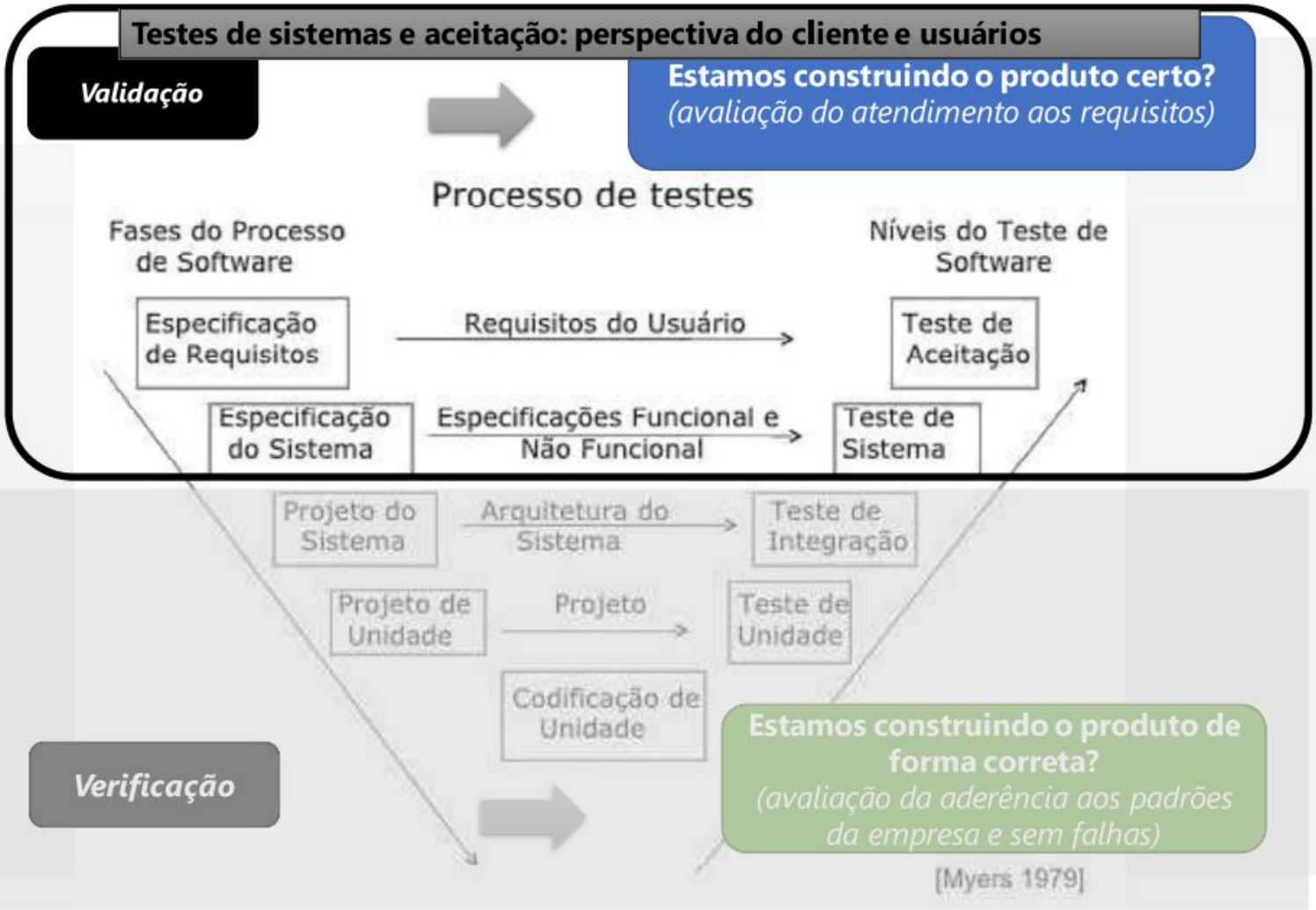
2

PLANO DE TESTES - Modelo V



3

PLANO DE TESTES - Modelo V



4

Testes unitários,
testes automatizados

Testes manuais

- Necessidade casos de testes bem especificados, com definição de informação de objetivo do teste, ação a ser realizada, saída ou comportamento esperado
- Realização de code review: inspeção de código de forma não automatizada
- Code review é uma boa prática para auxiliar na refatoração de código, propriedade coletiva e gestão de conhecimento
- Testado uma vez

Testes unitários,
testes automatizados

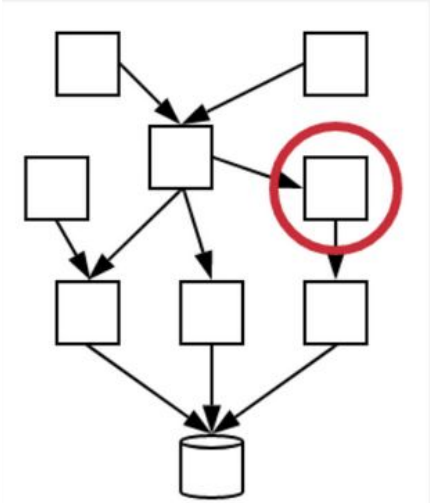
Testes automatizados

Programa/função que são desenvolvidos para testar unidades do seu sistema
- Executa uma rotina que irá testar o retorno do seu código (programa que testa um programa)
- Pode ser executado várias vezes
- Arcabouço de testes automatizados (frameworks/ferramentas que facilitam escrever e executar os testes unitários)
(Exemplo de ferramentas: Python: PyTest, PHP: PHPUnit, Java: Junit, ...)

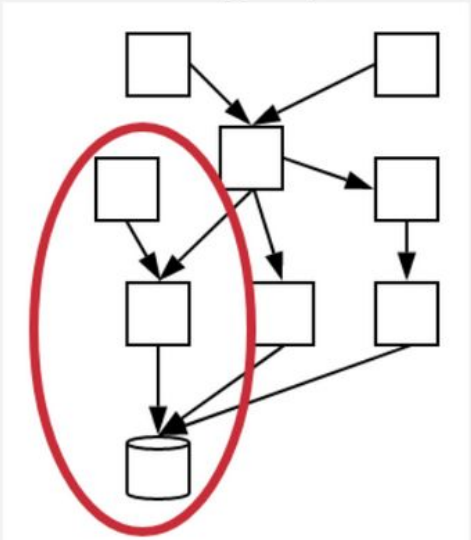
6

Testes realizados pela equipe de desenvolvimento

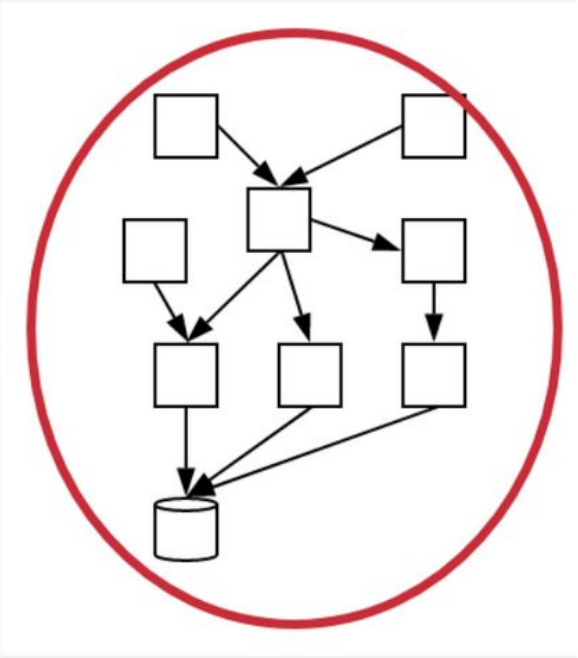
Escopo dos testes unitários



Escopo dos testes de integração



Escopo dos testes de sistema



7

Teste unitário
(sem teste automatizado)

Exemplo – cálculo fatorial

```
def fatorial (n):
    if n < 0:
        return 0
    i = fat = 1
    while i <= n:
        fat = fat * i
        i = i + 1
    return fat
```

O fatorial de um número inteiro e positivo “n”, representado por “n!” é obtido a partir da multiplicação de todos os seus antecessores até o número um

8

Teste unitário
(sem teste automatizado)

Exemplo – cálculo fatorial

```
def fatorial (n):
    if n < 0:
        return 0
    i = fat = 1
    while i <= n:
        fat = fat * i
        i = i + 1
    return fat
```

```
def test_fatorial0():
    assert fatorial(0) == 1

def test_fatorial1():
    assert fatorial(1) == 1

def test_fatorial_negativo():
    assert fatorial(-10) == 0

def test_fatorial4():
    assert fatorial(4) == 24
```

USP - https://bityli.com/rKUfE

https://wiki.python.org/moin/UsingAssertionsEffectively

9

Teste Manuais

PROBLEMAS em vista:

- Extremamente limitados
- Falhos
- Com o tempo é esquecido ou negligenciado

10

Teste unitário
(teste automatizado)

Parametrização

```
import pytest

@pytest.mark.parametrize("entrada,valor_esperado", [
    (0, 0),
    (1, 1),
    (2, 8),
    (-2, -8),
    (10, 1000),

])
```


11

Teste unitário
(teste automatizado)

```python
def fatorial (n):
    if n < 0:
        return 0
    i = fat = 1
    while i <= n:
        fat = fat * i
        i = i + 1
    return fat

import pytest
```

```python
@pytest.mark.parametrize("entrada, esperado", [
    (0, 1),
    (1, 1),
    (-10, 0),
    (4, 24),
    (5, 120)
    ])

def testa_fatorial(entrada, esperado):
    assert fatorial(entrada) == esperado
```


12

Testes unitários – produzindo códigos testáveis

- Em alguns casos, pode ser necessário refatorar o código para que ele se torne testável.
- Quando escrevemos software, devemos pensar em produzir código testável.
- Unidades sem possuir código duplicado.

13

Curiosidade!

- Testes de unidade são fortemente encorajados e amplamente praticados no Google. Todo código de produção deve ter testes de unidade e nossa ferramenta de revisão de código automaticamente destaca código submetido sem os correspondentes testes. Os revisores de código normalmente exigem que qualquer mudança que adiciona novas funcionalidades deve também adicionar os respectivos testes.

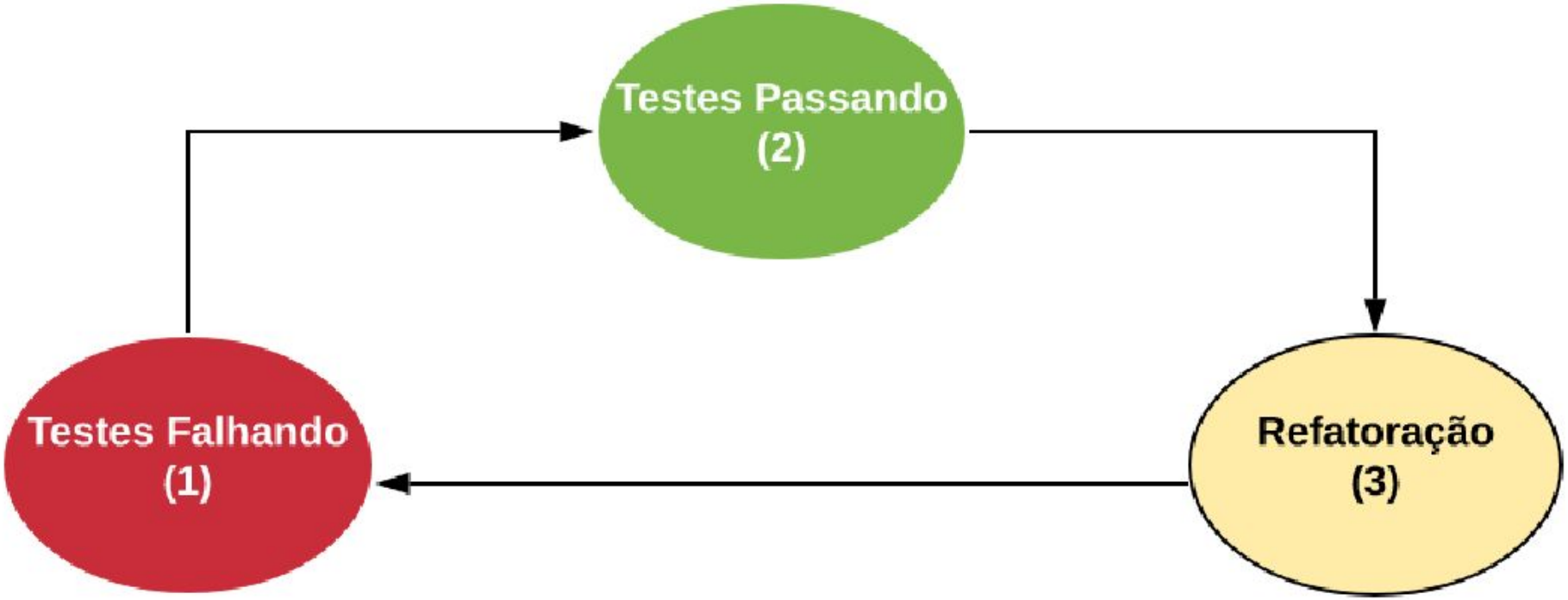
  Fonte: https://doi.org/10.48550/arXiv.1702.01715

Curiosidade!

- No Facebook, engenheiros são responsáveis pelos testes de unidade de qualquer código novo que eles desenvolvam. Além disso, esse código deve passar por testes de regressão, os quais são executados automaticamente, como parte dos processos de commit e push.

  Fonte: https://ieeexplore.ieee.org/document/6449236/

TDD - Desenvolvimento Dirigido por Testes (Test Driven Development) - 2003

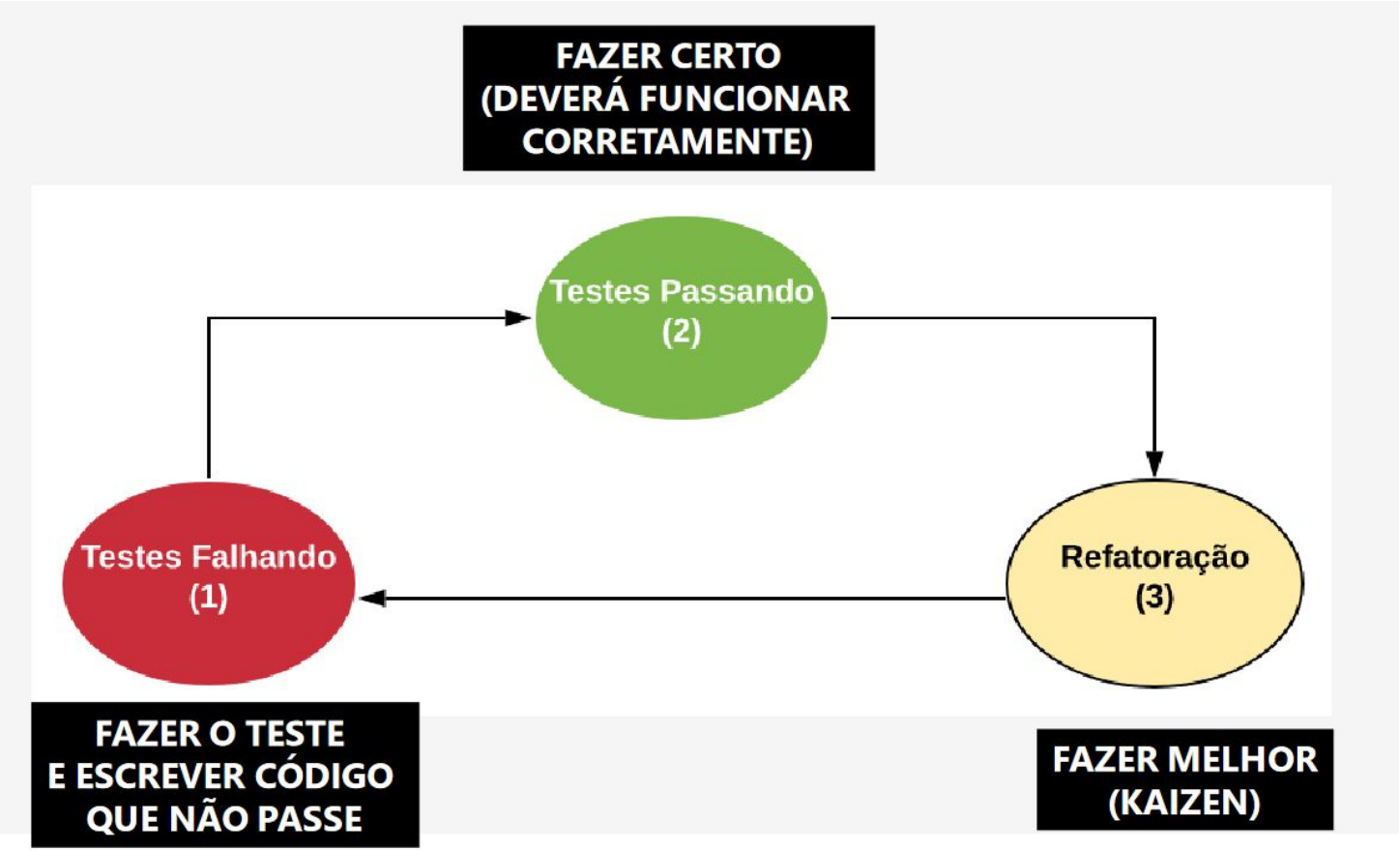


16

TDD

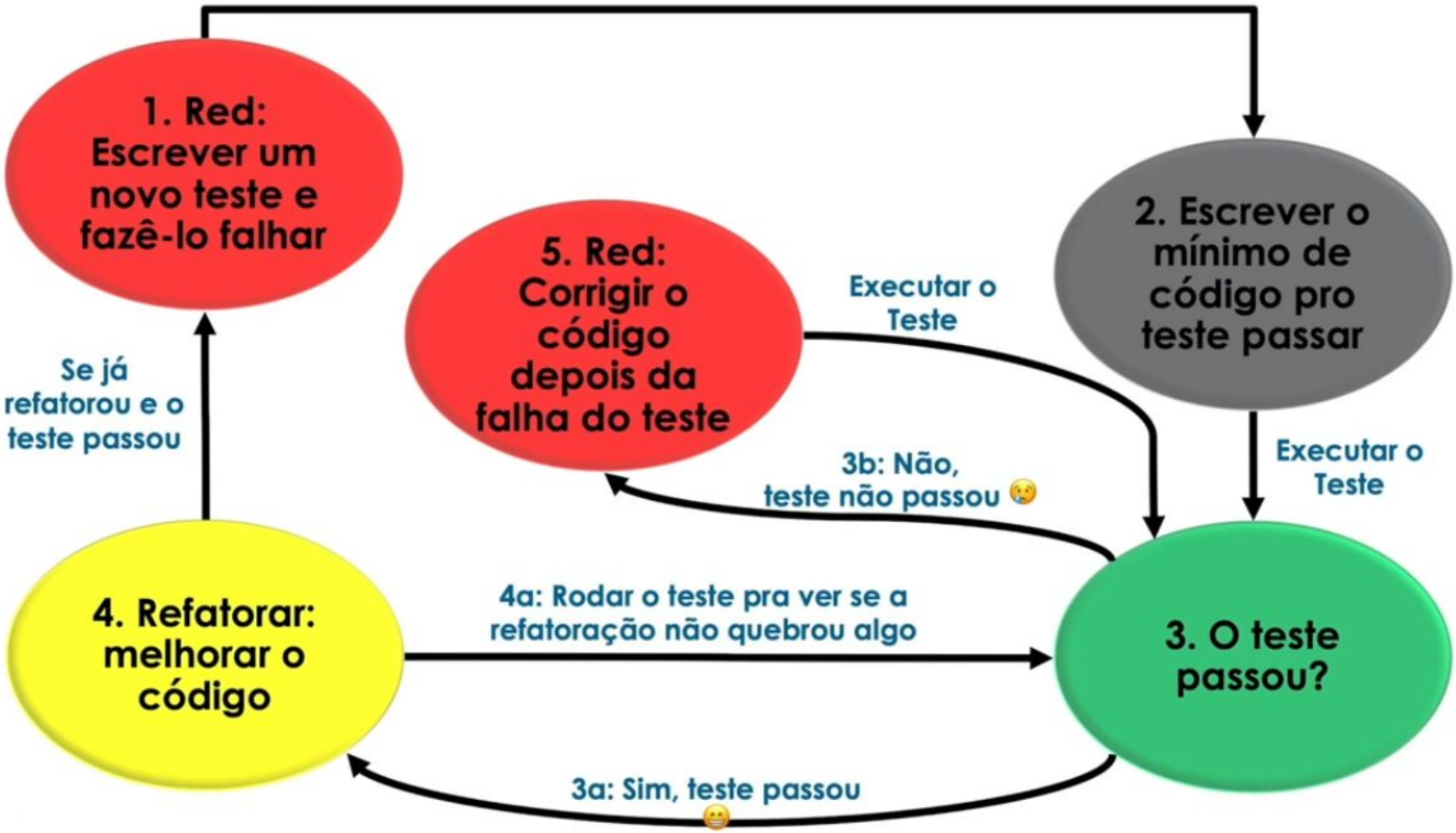
- Técnica utilizada pelo eXtreme Programing – XP
- Os testes devem ser escritos antes do início do desenvolvimento
  Etapas:
  1. Teste que falhe (Red) - Primeiro deve fazer um teste que falhe
  2. Teste que funcione (Green)- Segundo escrever um código que funcione corretamente
  3. Refatorar o código (yellow)- Observar se existe oportunidades de melhorar/refatorar o código, eliminando redundâncias, métodos muito grandes que possam ser quebrado em menores, definição de melhores padrões, etc.

TDD - Desenvolvimento Dirigido por Testes (Test Driven Development)



18

TDD - Desenvolvimento Dirigido por Testes (Test Driven Development)



19

Como identificar os testes unitários que serão necessários?

- Diagrama de classes/documentação de projeto
- Cenários dos casos de testes
- Cenário de teste X Caso de teste:
  1º Define o que deve ser testado.
  2º É uma forma específica de como deve ser feito o teste validando ou não se a funcionalidade está correta.

20

Cenários dos casos de testes

Senha	Confirmação	Alterado?	Mensagem
1234user	1234user	Não	Senha não pode conter sequencia com mais de três números
User12new	User12new	Sim	Senha alterada com sucesso!
user12	user12	Não	Senha deve conter uma letra maiúscula!
USER123	USER123	Sim	Senha alterada com sucesso!

CENÁRIOS DE TESTE			
No. Passos	**Objetivo** *(o que deve ser validado / por que testar)*	**Ação** *(passos a reproduzir)*	**Resultado ou comportamento esperado** *(qual o resultado ou comportamento que o sistema deve fornecer)*
1	Testar campos obrigatórios	Deixar os campos em branco e clicar no botão salvar	O sistema deve mostrar mensagem avisando que os campos: descrição, valor, data e categoria de recebimentos, são obrigatórios preenchimento.
2	Valor negativo	Informar valor negativo no campo valor	Não salvar e mostrar mensagem ao usuário.
3	Valor muito alto	Informar valor acima de R$ 1.000.000,00	Mostrar mensagem: "Valor muito alto. Tem certeza que desaja continuar?"
4	Identificar erros na digitação de datas	Informar datas com formato e valores inválidos	Não salvar, apresentar mensagem ao usuário
...	...	...	...
5	Salvar registro no banco de ddos	Clicar no botão salvar	Verificar se o registro foi salvo corretamente no banco de dados.
6	Atualizar saldo da conta	Verificar o saldo anterior da conta. Clicar no botão salvar.	O valor da conta deve ser: [saldo anterior] + [valor]
7	Redirecionar para a tela anterior	Após salvar	Deve ter somado o valor recebido ao valor existente do saldo da conta do usuário

21