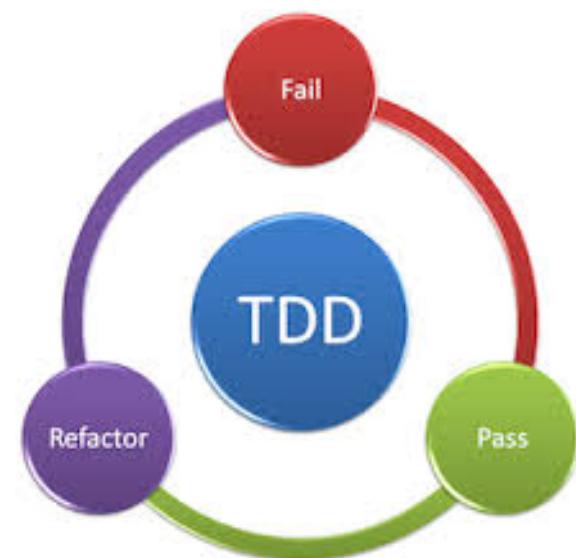
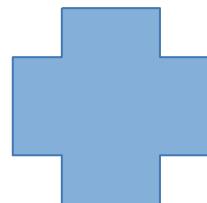


TDD e PHP: Em busca de qualidade



Coisas que quero falar... Se puder



O que é TDD?

Por que usar TDD?

**Aplicar TDD para garantir que as coisas estão
funcionando bem**

**Por que Testar Unidade, cobertura de código, sniffer de
código são importantes?**

Ferramentas TDD e QA para PHP

Como automatizamos o teste?

TDD

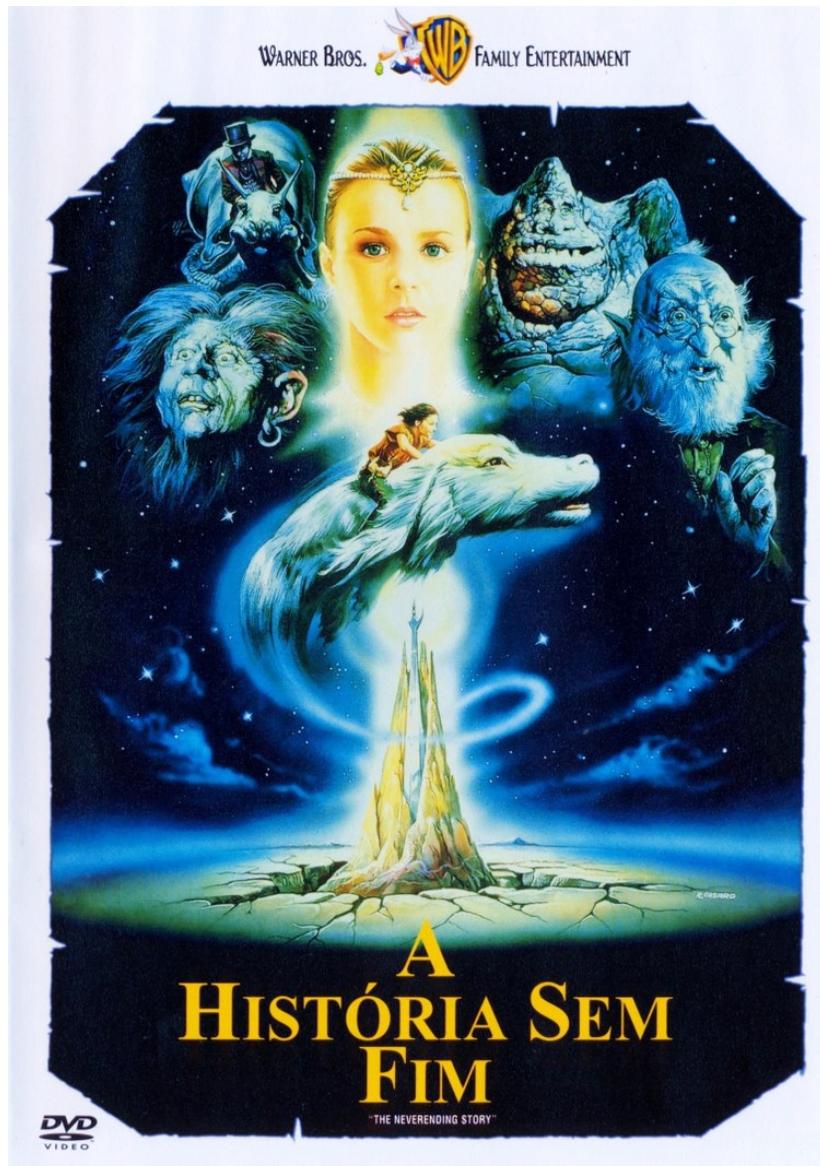


TDD é uma técnica de design não de teste, apesar de falarmos muito de teste, o foco final é o design do sistema...

Um código com um bom design é testável, o design ruim não é.

Michael C. Feathers.

Pequena história



Problemas encontrados

feedback dos testes no projeto de classes



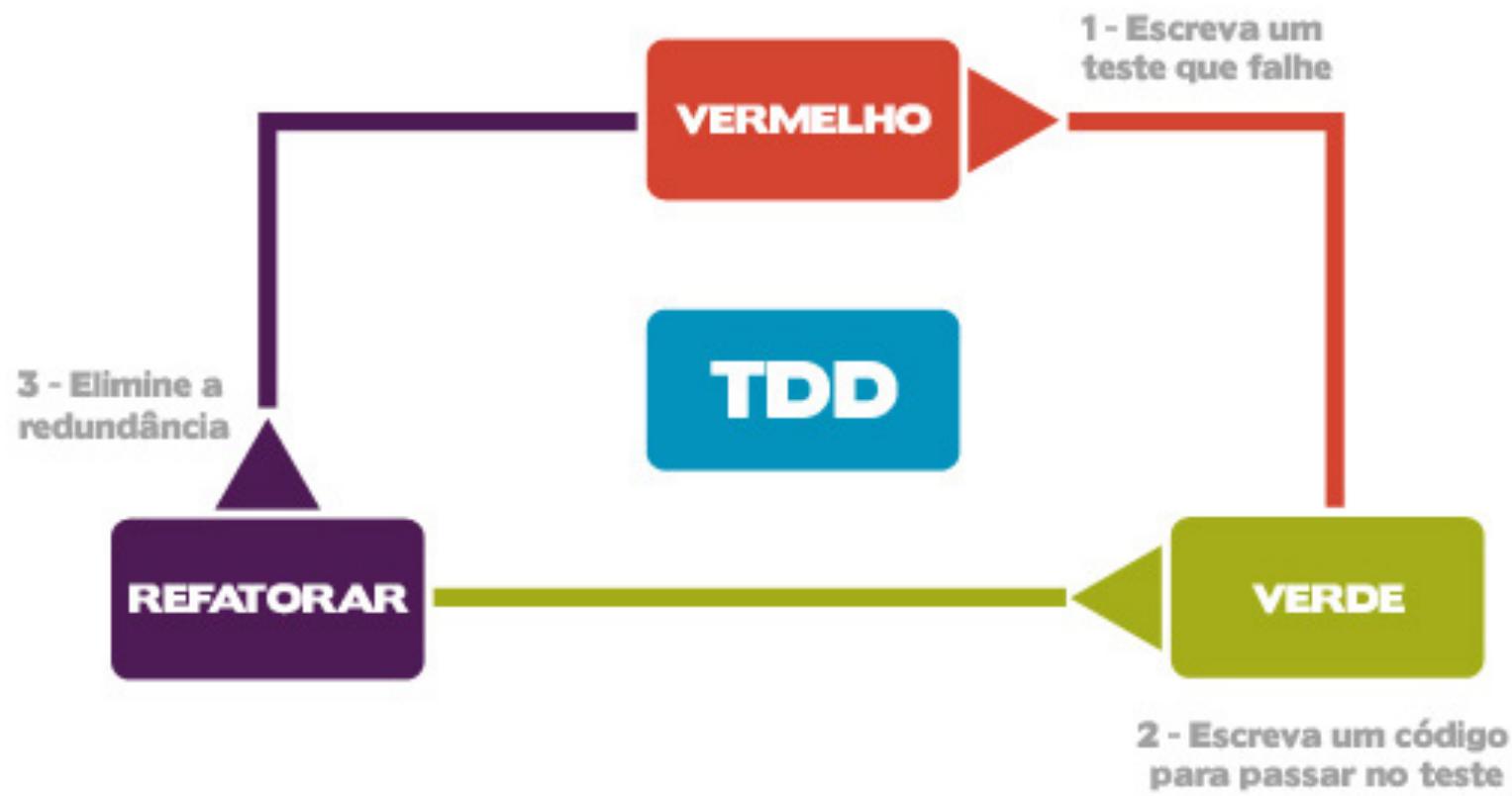
Desenvolvimento
Guiado por
Testes



Abordagem
tradicional

feedback dos testes no projeto de classes

Por que TDD?



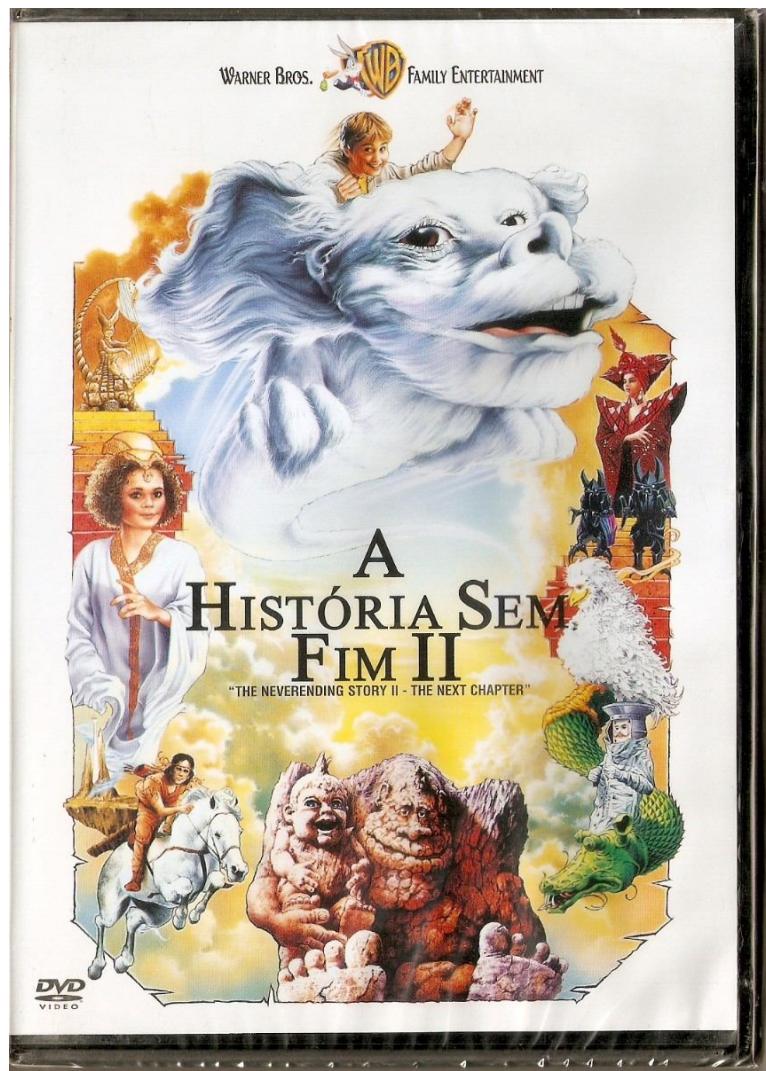
Podemos dizer que...

TDD é uma técnica de desenvolvimento ágil que combina refatoração e desenvolvimento orientado a teste (Teste primeiro).

O conceito de desenvolvimento orientado a teste, criado por Beck centra-se em duas regras básicas:

- Escreva um teste automatizado, antes de escrever qualquer código de produção;**
- Remova duplicações.**

Mais história



O desespero



E agora?

Necessidade de automação dos testes no servidor;

Executar instalação com casos de teste de unidade;

Verificar a cobertura do código e cheirar o seu código.

Framework de teste

Surge então o PHPUnit (2005)

www.phpunit.de

Pear package

Criado por Sebastian Bergmann

baseado no JUnit

Funcionamento

PHPUnit

Utilizar o

extends PHPUnit_Framework_TestCase

Para criar um novo teste

Empregar as assertões para validar

Funcionamento

PHPUnit::Asserções

assertEquals()

assertTrue()

assertFalse()

assertContains()

assertFileEquals()

assertFileExist()

Algumas asserções em PHPUnit

```
<?php
class PrimeiroTestePHPUnit extends
PHPUnit_Framework_TestCase
{
    public function testeIgualdade(){
        $this->assertEquals(2, 1+1, "Valores não
são iguais");
        $this->assertEquals(2.2, 2.1, 'Diferença entre
0.2 não é considerada', 0.2);
        $this->assertEquals(2.2, 2.1, 'Toda
diferença é considerada');
    }
    public function testeContem(){
        $this->assertContains("f", "Fabio", "Dados
não encontrados", true);
    }
    public function testeContagem(){
        $this->assertCount(2, array("fabio", "g"),
"Contagem diferenciada");
    }
}
```

```
public function testeVazio(){
    $this->assertEmpty(array("Teste"), "Não esta
vazio");
}
public function testeFalso(){
    $this->assertFalse(True);
}
public function testeVerdadeiro(){
    $this->assertTrue(False);
}
public function testeNulo(){
    $this->assertNull('Teste');
}
?>
```

Ultima história



Mas o que é um bom teste?



Independente

Rápido

RePLICÁVEL

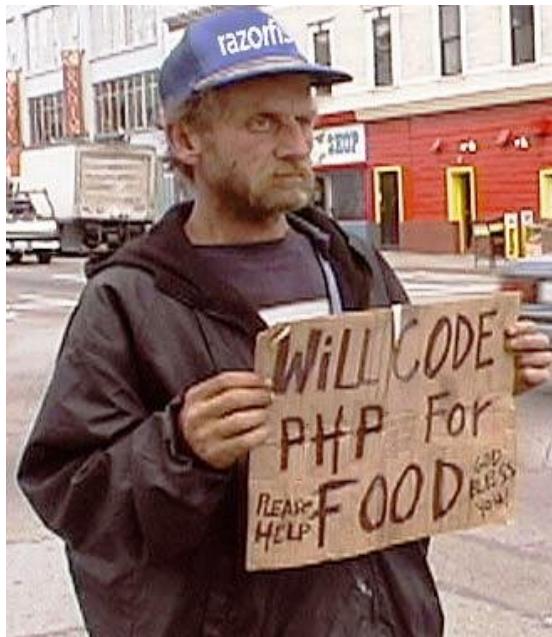
Pequeno

Atualizado

Resiliente

Mas por que testar?

"Nenhum engenheiro de software libera nem mesmo a menor mudança sem testá-la, exceto os muito confiantes e os muito desleixados." Kent Beck



Benefícios do TDD

Alta coesão;

Baixo acoplamento;

Confiança;

Menor tempo de resolução de problemas;

Documentação

O que estão dizendo?

"Precisamos desenvolver nosso software economicamente mais valioso gastando dinheiro mais devagar, obtendo rendimento mais rapidamente e aumentando a provável expectativa de vida produtiva do nosso projeto. Mas o mais importante de tudo é aumentar as opções de decisões de negócios"

John Favaro

Ferramentas para o programador PHP



SimpleTest

Criado por Marcus Backer

Diferenças:

Prove recursos de MOCK, Web Test, teste unitário

Novas assertions:

assertField()

assertWantedPattern()

Exemplo:

\$this->

```
assertWantedPattern('~string.*<span.*background.*#0000  
FF.*blue</span>.*highlight~i',$o->render());
```

Exemplo SimpleTest

```
<?php

require_once('simpletest/autorun.php');
require_once('simpletest/web_tester.php');
class TestContact extends WebTestCase{
    function testContactPage(){
        $this->get('http://gotic.com.br/?page_id=31');
        $this->setField('your-name', 'Fabio');
        $this->setField('your-email', 'gomesrocha@gmail.com');
        $this->setField('your-message', 'Email enviado de teste');
        $this->setField('your-subject', 'Email de teste');
        $this->click('Enviar');
        $this->assertTitle('Novo Contato');
    }
}
?>
```

O que eu aprendi?

- Arquitetura
- Redução de código mal cheiroso
- Melhoria no design do sistema
- Redução das falhas é redução de estresse

Indicação de literatura

PHPUnit Essentials (Zdenek Macheck)
php|arquitect's: Guide to PHP Design Patterns (Jason E. Sweat)
Test-Driven Development: Teste e design no mundo real com PHP (Andre Cardoso e Maurício Aniche)
Programação Extrema (XP) Explicada (Kent Beck)
TDD Desenvolvimento Guiado por Testes (Kent Beck)
Desenvolvimento de Software Orientado a Objetos, Guiado por Testes (Steve Freeman e Nat Pryce)
Métodos ágeis para desenvolvimento de software (Rafael Prikladnicki, Renato Willi e Fabiano Milani)

Obrigado



Contato

<http://www.devmedia.com.br/space/fabio-gomes-rocha>

www.fgrweb.com.br (Não tem nada ainda)

<https://www.youtube.com/user/gomesrocha>

<https://www.linkedin.com/in/fabiorgomesocha/>

www.gpitic.com.br

www.sempescom.org