



JUnit Selenium

Cauê Guerra
Cecilia Fernandes
Mauricio Aniche
Mauricio De Diana

Windows

A fatal exception 0E has occurred at 0028:C0011E36 in UXD UMM(01) + 00010E36. The current application will be terminated.

- * Press any key to terminate the current application.
- * Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue _

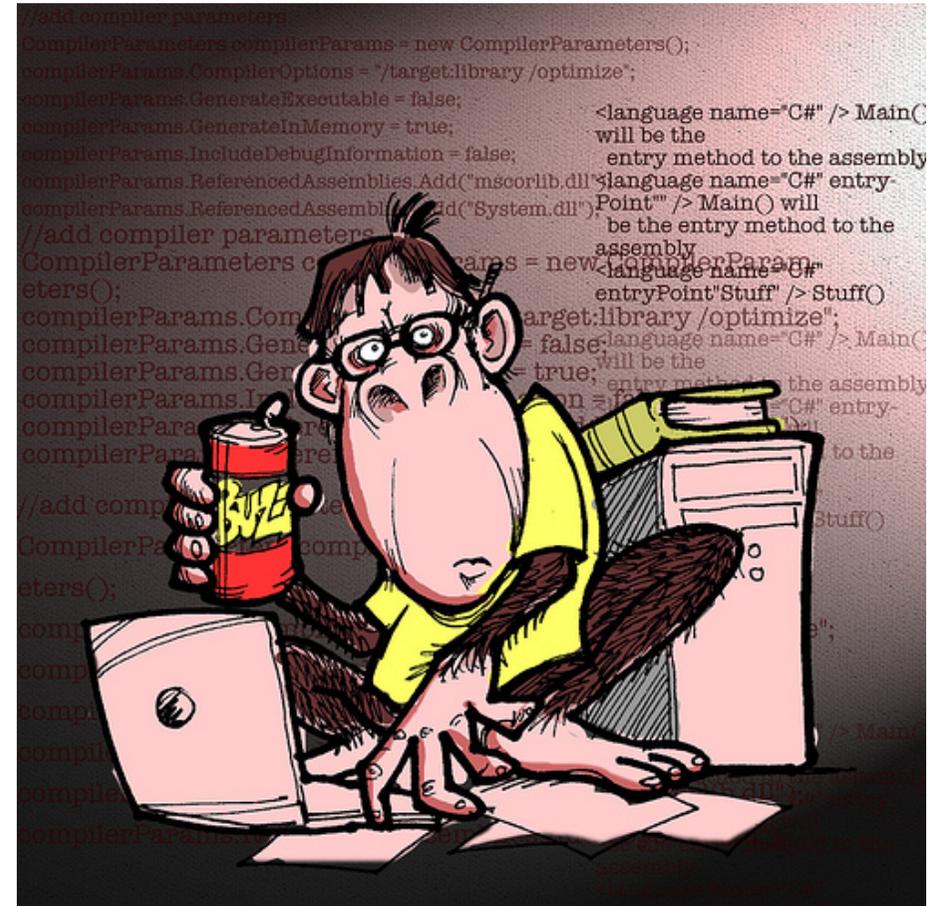
Por quê testar?

Prejuízos de aproximadamente \$59.5 bilhões na economia dos EUA (Fonte: NIST/2002)



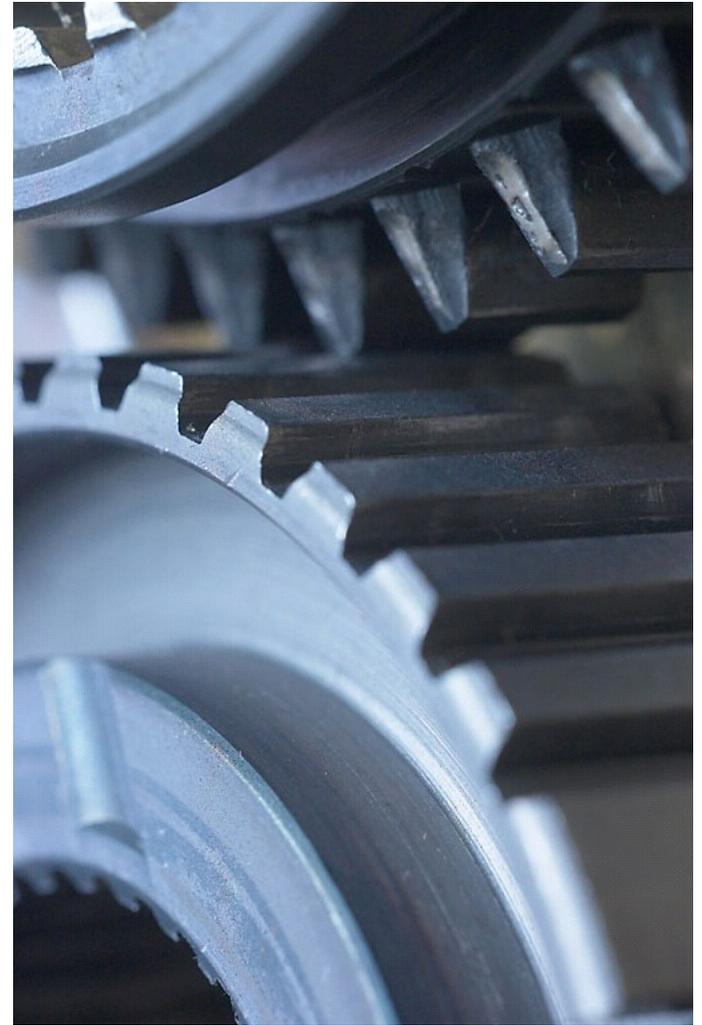
Teste Manual

- Difícil
- Demorado e cansativo
- Executado poucas vezes
- Cobre poucos casos
- Sem documentação



Testes automatizados

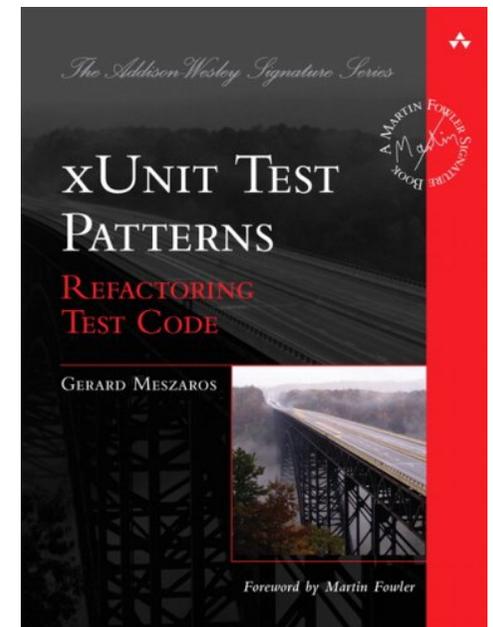
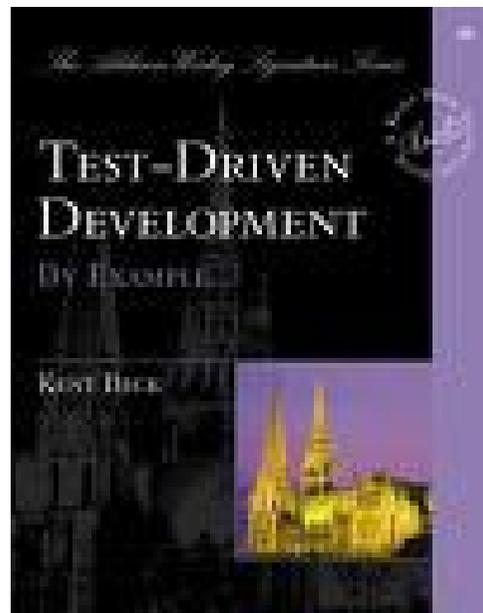
- Testes rodam rápido
- Cobre muitos casos
- Segurança na manutenção
- Ajuda na documentação



“Inspeccionar para prevenir defeitos é bom; Inspeccionar para encontrar defeitos é desperdício” - Shigeo Shingo

Princípios Básicos

- Código dos testes deve ser simples
- Testes podem conter erros
- Devem fazer parte da manutenção
- Não devem exigir intervenção humana



Teste unitário

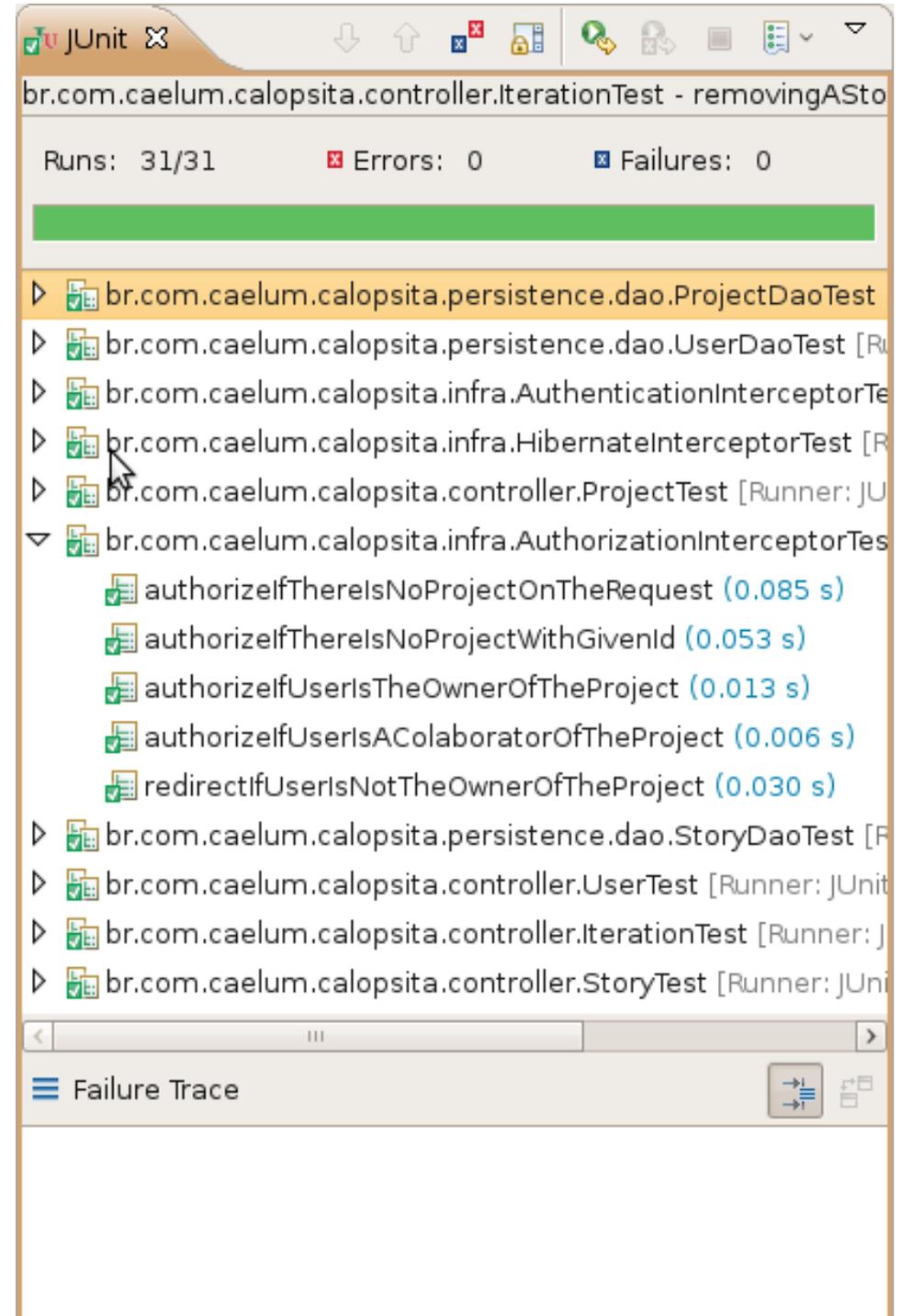
Menor unidade possível

Independente

Teste de regressão

Documentação

Design



Histórico

Criadores: Kent Beck e Erich Gamma

SUnit x JUnit x xUnit



JUnit 3

```
import junit.framework.*;
```

```
public class MultiplicationTest extends TestCase {  
    public void testMultiplication() {  
        assertEquals("Multiplication", 6, 3 * 2);  
    }  
}
```

JUnit 4

```
import org.junit.*;
```

```
public class MultiplicationTest {  
    @Test  
    public void multiplication() {  
        Assert.assertEquals("Multiplication", 6, 3 * 2);  
    }  
}
```

JUnit 3 x JUnit 4

```
import junit.framework.*;
```

JUnit 3

```
public class MultiplicationTest extends TestCase {  
    public void testMultiplication() {  
        assertEquals("Multiplication", 6, 3 * 2);  
    }  
}
```

```
import org.junit.*;
```

JUnit 4

```
public class MultiplicationTest {  
    @Test  
    public void multiplication() {  
        Assert.assertEquals("Multiplication", 6, 3 * 2);  
    }  
}
```

JUnit 3 x JUnit 4

JUnit 3

public void testX()

public void setUp()

public void tearDown()

assertEquals()

JUnit 4

@Test

@Before

@After

Assert.assertEquals()

Assertions

`assertEquals`

`assertTrue` / `assertFalse`

`assertSame` / `assertNotSame`

`assertNull` / `assertNotNull`

`assertArrayEquals`

Mais testes...

O que importa pro seu usuário?



- a) Seu backend fantástico!
- b) O framework de ponta que sua equipe usa!
- c) Que você usa Scrum com XP no desenvolvimento
- d) A validação em JavaScript que não deixa salvar datas inválidas...

Selenium HQ

by OpenQA (and Thoughtworks?)

Testes de Aceitação

Testa a interface:

- Redirecionamentos
- Alteração de dados em tabelas
- Validações JS
- AJAX
- ...

Selenium HQ

```
public class MyTest extends TestCase {  
  
    private Selenium selenium;  
    @Test  
    public void selecionarSaoPauloAutoCheckEhCapital() {  
        selenium.open("/projeto/pagina.jsp");  
        String[] options = selenium.getSelectOptions("formulario");  
        selenium.select("formulario", options[0]);  
        selenium.fireEvent("cidade", "blur");  
        Assert.assertTrue(selenium  
            .getEval("selenium.page().findElement(\ehCapital\).checked"));  
    }  
}
```

Chato?



BOREDOM

Selenium DSL

```
public class MyTest {
```

```
private Browser browser;
```

```
@Test
```

```
public void selecionarSaoPauloAutoCheckEhCapital() {
```

```
    Page page = browser.open("/projeto/pagina.jsp");
```

```
    Form form = page.form("formulario")
```

```
    form.select("cidade").choose(0).blur();
```

```
    Assert.assertTrue(form.isChecked("ehCapital"));
```

```
}
```

Simplicidade!

Dá pra ser mais fácil?

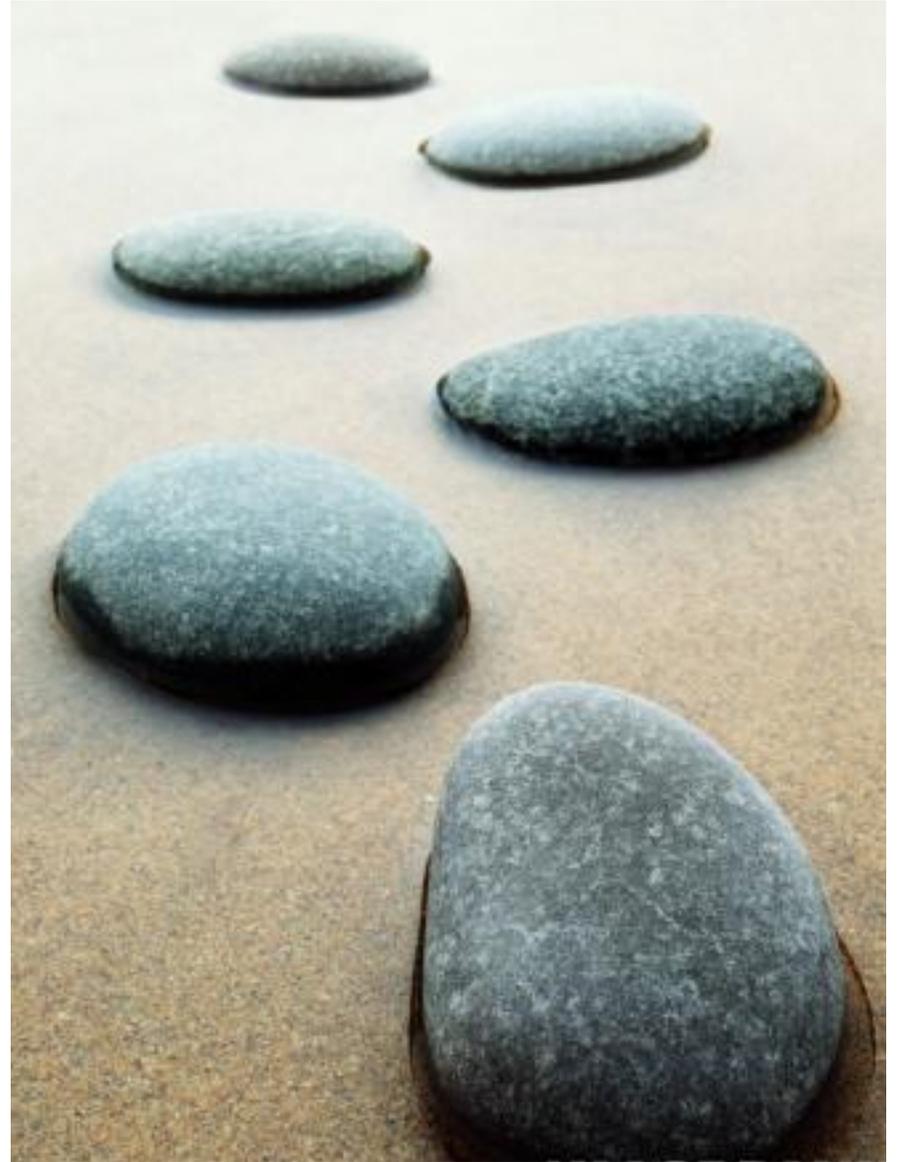
E portátil?

Selenium IDE

Selenium IDE

- Permite gravarmos uma sequencia de passos
- Fácil uso
- Não é preciso conhecer a sintaxe do Selenium

- Difícil manutenção
- Só roda no Firefox (plugin)



Selenium IDE



não é bonito, mas é fácil

Célebre citação...

"The idea of "unit tests" appeals to me only rarely, when I'm feeling my way in a totally unknown environment and need feedback about what works and what doesn't. Otherwise, lots of time is wasted on activities that I simply never need to perform or even think about. Nothing needs to be "mocked up"." *Donald Knuth*

Citações

“Whenever you are tempted to type something into a print statement or a debugger expression, write it as a test instead.”
Martin Fowler.

“Qualquer funcionalidade que não possui testes automatizados simplesmente não existe.” Kent Beck.

"Program testing can be used to show the presence of bugs, but never to show their absence." Edsger W. Dijkstra

Testes...

- Servem como documentação;
- Ajudam você a detectar quebras após qualquer implementação;
- Testes unitários, junto com alguma ferramenta de cobertura de testes, aumentam a qualidade;
- Aumentam a eficiência da sua equipe;
- Aumentam a segurança da equipe em refatorar.

Dúvidas?

