Testes e Integração Contínua

Diego Mira David
Eduardo Teruo Katayama

Testes Automatizados

"Qualquer funcionalidade que não possui testes automatizados simplesmente não existe"

Kent Beck

Testes Manuais

- Demorado
- Difícil repetição
- Casos simples
- Ausência de documentação ou documentação obsoleta
- Efeito "bola de neve" devido a ausência de testes de regressão
- Quem gosta de testar manualmente?

Testes Automatizados

- *Scripts* que exercitam as funcionalidades do sistema
- Podem ser facilmente reproduzidos
  - Testes mais frequentes
- Casos complexos
- Regressão com baixo custo

Testes Automatizados

- Favorece a manutenção e refatoração
- Segurança e coragem ao grupo
- Ajuda a encontrar erros mais cedo
- Certificação do que foi testado

Princípios Básicos

- Código dos testes merecem a mesma atenção do código do sistema
- Precisam de Manutenção
- Podem conter erros
- Precisa ser simples
- Mais legível quanto possível

Testes Automatizados

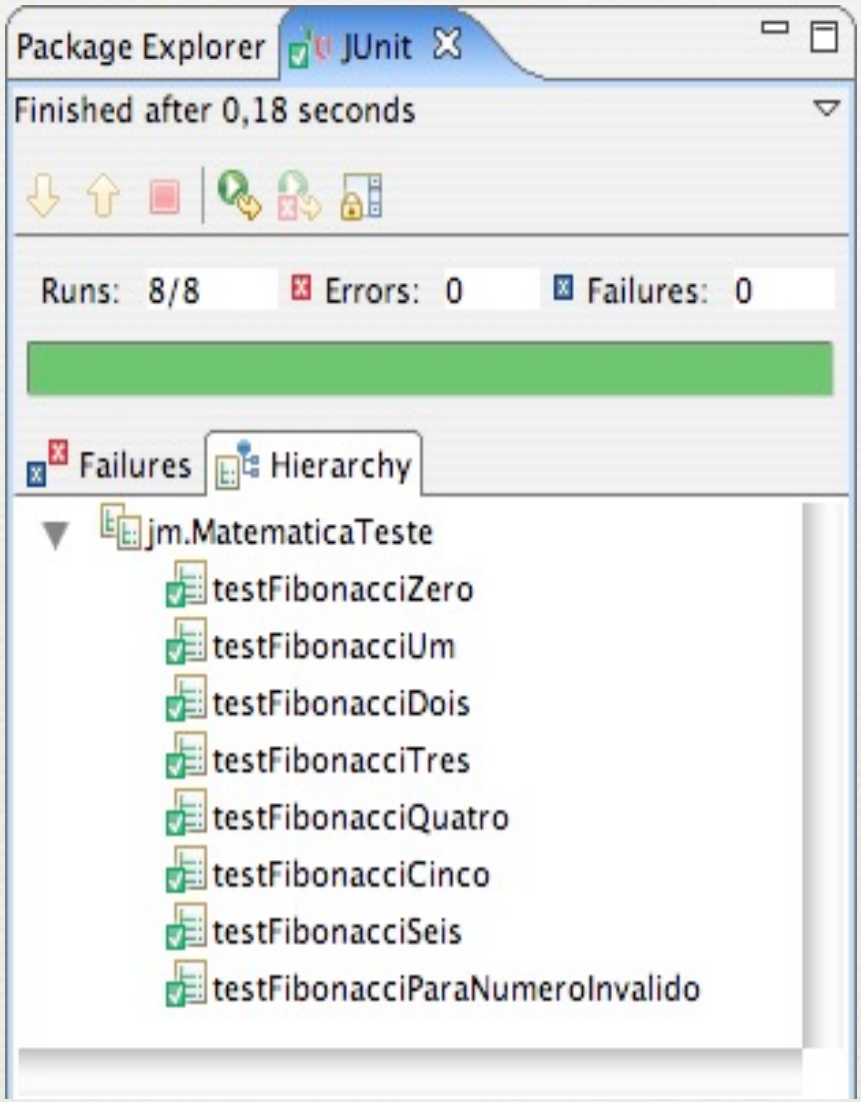
"Testes podem mostrar a presença de erros, não a sua ausência" (Dijkstra)

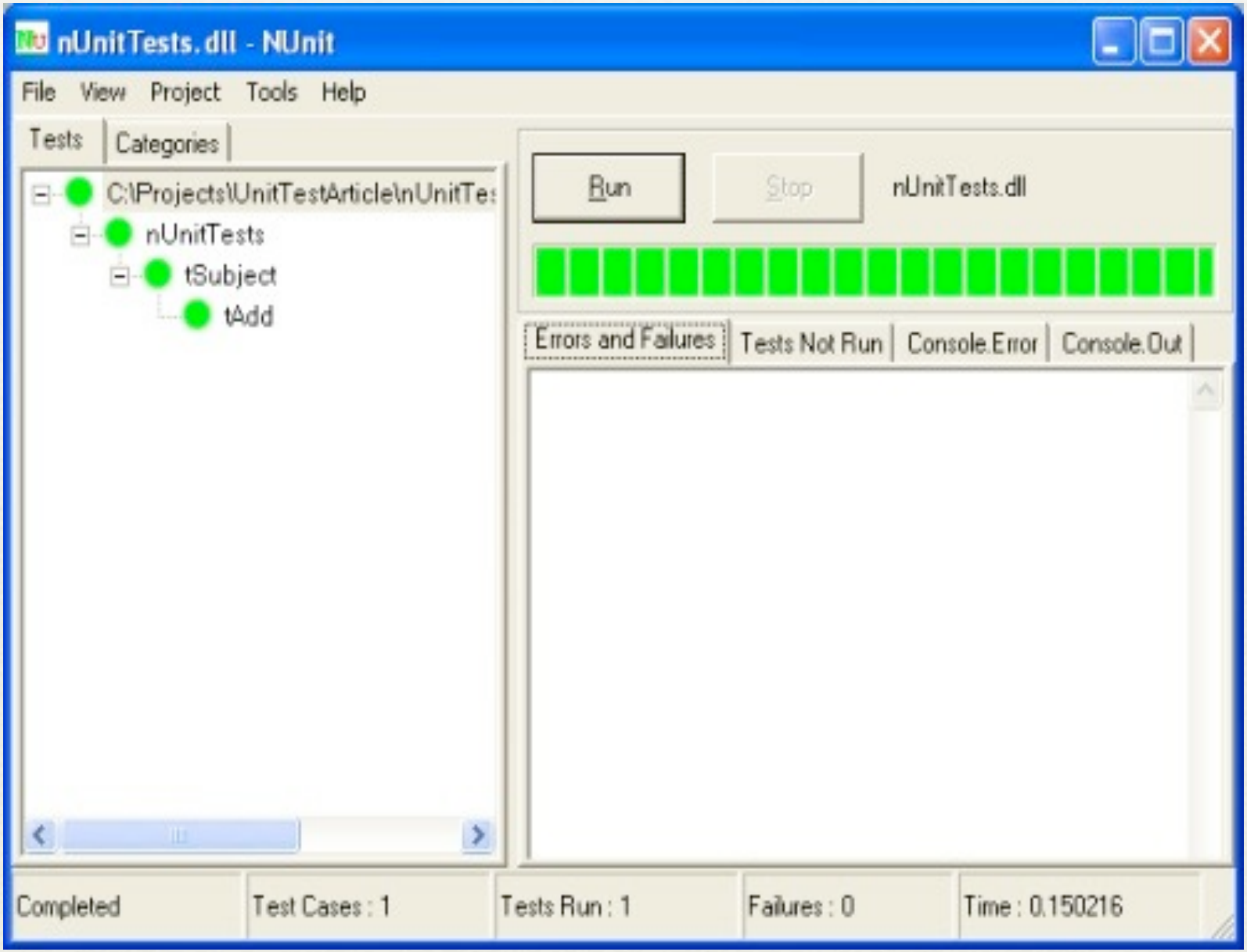
Testes de Unidade

- Testam apenas um componente do sistema
  - Métodos ou classes em programas OO
- Todas as dependências são simuladas (*Mock*)
  - *Arcabouços* (como EasyMock) para criar e gerenciar *mocks*
- Testar caso de sucesso e de erro
  - Exceções esperadas

xUnit

- Família de arcabouços para testes
- Disponível para praticamente qualquer linguagem e ambiente de desenvolvimento
  - SUnit(Smalltalk), JUnit(Java), CppUnit(C++), CUnit(C), NUnit(.NET), pyUnit(Python), etc...


Package Explorer
JUnit
Finished after 0,18 seconds
Runs: 8/8
Errors: 0
Failures: 0
Failures
Hierarchy
jm.MatematicaTeste
testFibonacciZero
testFibonacciUm
testFibonacciDois
testFibonacciTres
testFibonacciQuatro
testFibonacciCinco
testFibonacciSeis
testFibonacciParaNumeroInvalido

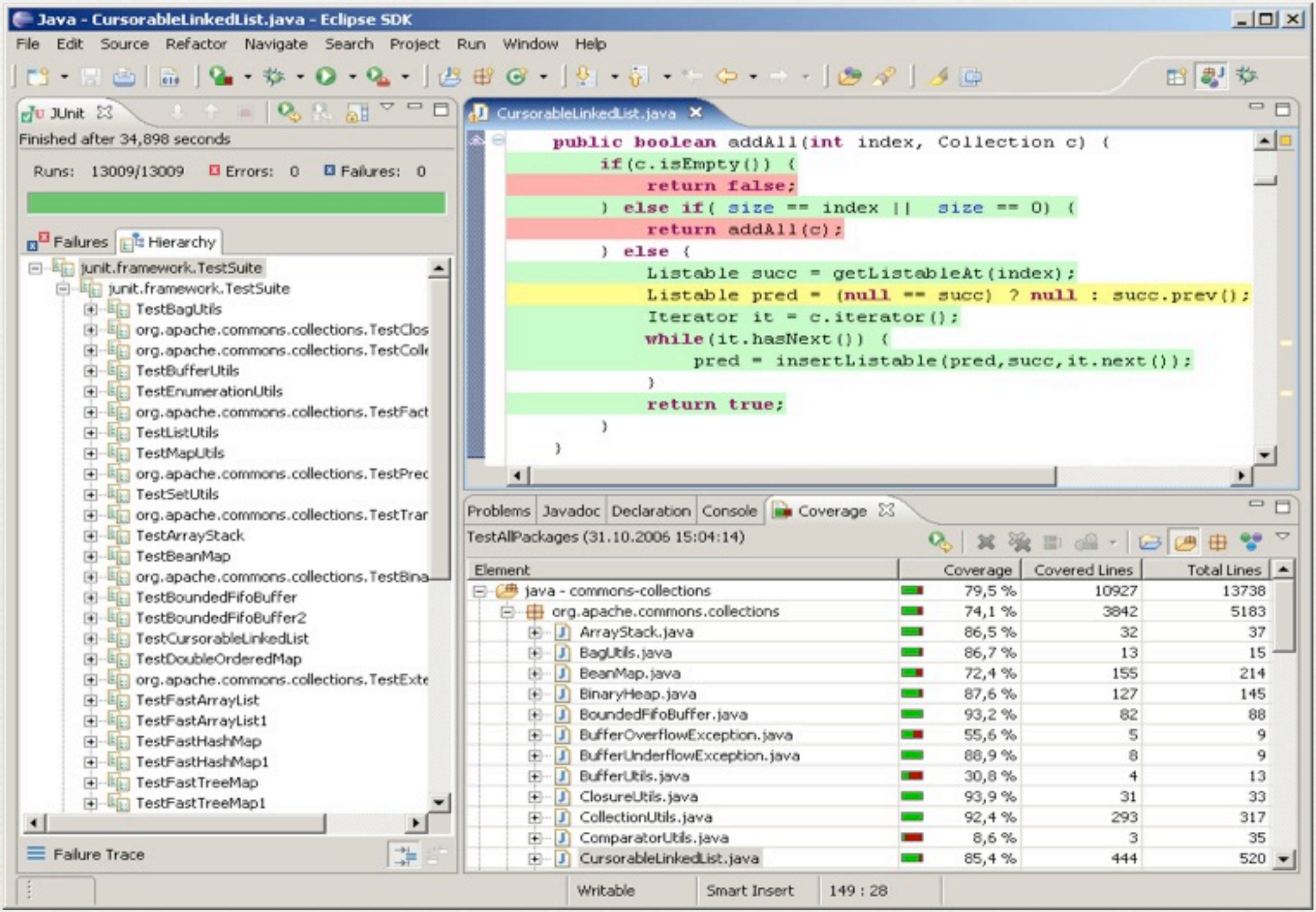

nUnitTests.dll - NUnit
File View Project Tools Help
Tests
Categories
C:\Projects\UnitTestArticle\nUnitTes
nUnitTests
tSubject
tAdd
Run
Stop
nUnitTests.dll
Errors and Failures
Tests Not Run
Console.Error
Console.Out
Completed
Test Cases : 1
Tests Run : 1
Failures : 0
Time : 0.150216

Exemplo de código

```
public class TestFoo extends TestCase {
    public testGetHelloWorld() {
        Foo foo = new Foo();
        assertEquals("Hello World!", foo.getHelloWorld());
    }
}
```

Cobertura dos testes

- Cada trecho do programa deve ser executado por algum teste
- Há ferramentas que indicam a cobertura dos testes
- Indicam as partes do código fonte de uma aplicação que estão sendo exercitadas por testes automatizados


Java - CursorableLinkedList.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
JUnit
Finished after 34,898 seconds
Runs: 13009/13009 Errors: 0 Failures: 0
Failures Hierarchy
junit.framework.TestSuite
junit.framework.TestSuite
TestBagUtils
org.apache.commons.collections.TestClos
org.apache.commons.collections.TestColl
TestBufferUtils
TestEnumerationUtils
org.apache.commons.collections.TestFact
TestListUtils
TestMapUtils
org.apache.commons.collections.TestPrec
TestSetUtils
org.apache.commons.collections.TestTrar
TestArrayStack
TestBeanMap
org.apache.commons.collections.TestBina
TestBoundedFifoBuffer
TestBoundedFifoBuffer2
TestCursorableLinkedList
TestDoubleOrderedMap
org.apache.commons.collections.TestExte
TestFastArrayList
TestFastArrayList1
TestFastHashMap
TestFastHashMap1
TestFastTreeMap
TestFastTreeMap1
Failure Trace
CursorableLinkedList.java
public boolean addAll(int index, Collection c) {
if(c.isEmpty()) {
return false;
} else if(size == index || size == 0) {
return addAll(c);
} else {
Listable succ = getListableAt(index);
Listable pred = (null == succ) ? null : succ.prev();
Iterator it = c.iterator();
while(it.hasNext()) {
pred = insertListable(pred,succ,it.next());
}
return true;
}
}
Problems Javadoc Declaration Console Coverage
TestAllPackages (31.10.2006 15:04:14)
Element Coverage Covered Lines Total Lines
java - commons-collections 79,5 % 10927 13738
org.apache.commons.collections 74,1 % 3842 5183
ArrayStack.java 86,5 % 32 37
BagUtils.java 86,7 % 13 15
BeanMap.java 72,4 % 155 214
BinaryHeap.java 87,6 % 127 145
BoundedFifoBuffer.java 93,2 % 82 88
BufferOverflowException.java 55,6 % 5 9
BufferUnderflowException.java 88,9 % 8 9
BufferUtils.java 30,8 % 4 13
ClosureUtils.java 93,9 % 31 33
CollectionUtils.java 92,4 % 293 317
ComparatorUtils.java 8,6 % 3 35
CursorableLinkedList.java 85,4 % 444 520
Writable
Smart Insert
149 : 28

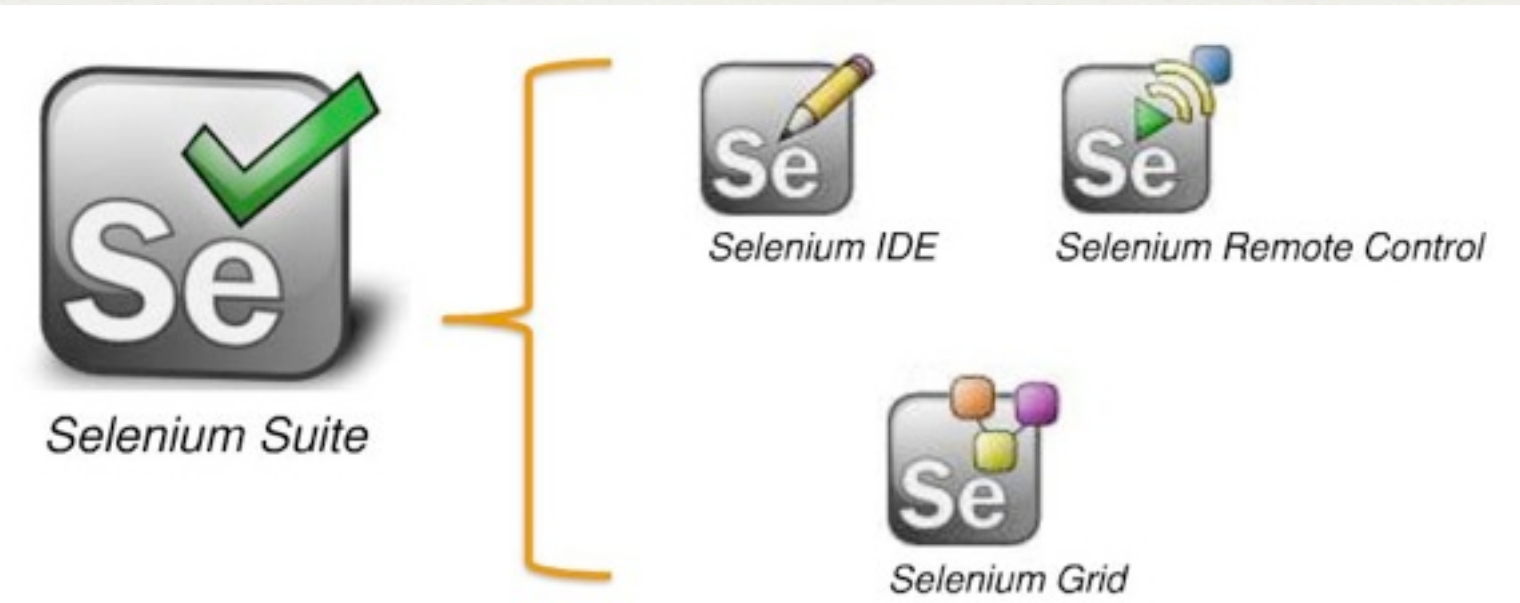
Testes de Integração

- Verificar erros provenientes da integração entre componentes do sistema, que podem funcionar individualmente
- Dependências reais

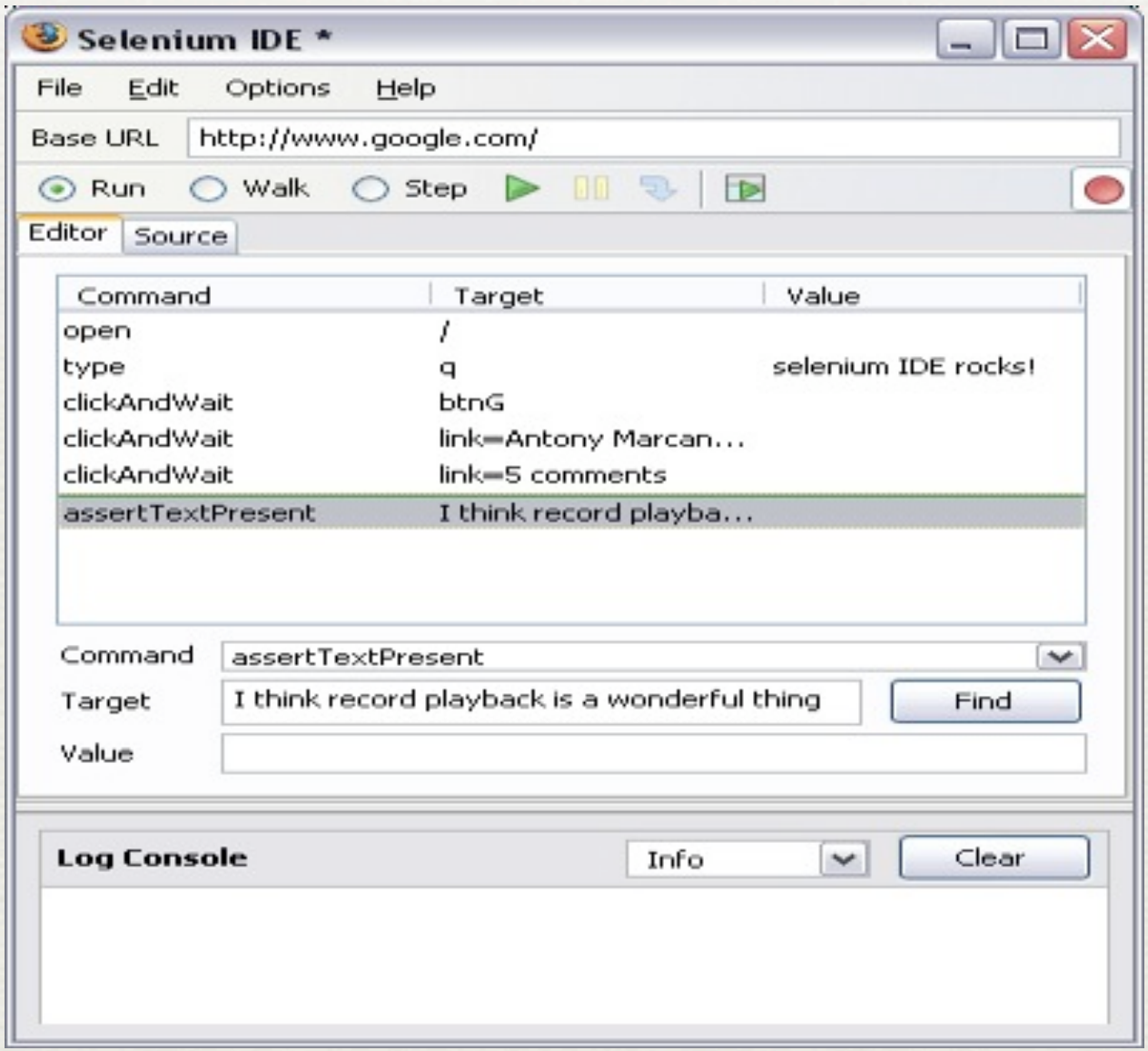
Testes de Aceitação

- Validação do sistema do ponto de vista do cliente
- Testam uma história, funcionalidade ou caso de uso
  - Simulação das ações do usuário
- Envolvem vários componentes do sistema

Selenium


Selenium Suite
Selenium IDE
Selenium Remote Control
Selenium Grid

Selenium IDE



Selenium RC

- Usa o poder de linguagens de programação para criar testes mais complexos
- Permite que o código seja escrito em várias linguagens, como Python, Ruby, C#, Java.

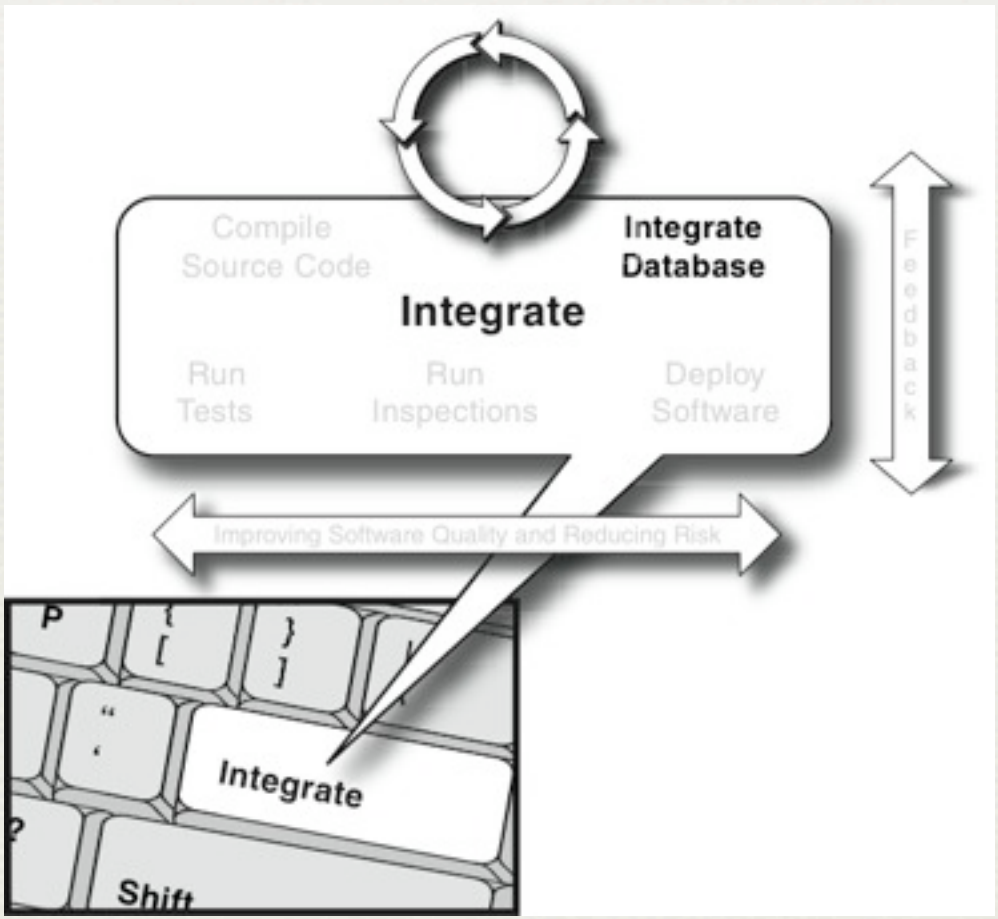
Exemplo

```
from selenium import selenium
import unittest, time, re
class NewTest(unittest.TestCase):
    def setUp(self):
        self.verificationErrors = []
        self.selenium = selenium("localhost", 4444, "*firefox",
                "http://www.google.com/")
        self.selenium.start()
    def test_new(self):
        sel = self.selenium
        sel.open("/")
        sel.type("q", "selenium rc")
        sel.click("btnG")
        sel.wait_for_page_to_load("30000")
        self.failUnless(sel.is_text_present("Results * for selenium rc"))
    def tearDown(self):
        self.selenium.stop()
        self.assertEqual([], self.verificationErrors)
```

Importância em Software Livre

- Ausência de departamento de qualidade
- Desenvolveres distantes em ambientes heterogêneos
- Maioria dos software de qualidade tem bateria de testes (Firefox, Eclipse, MySql, etc...)

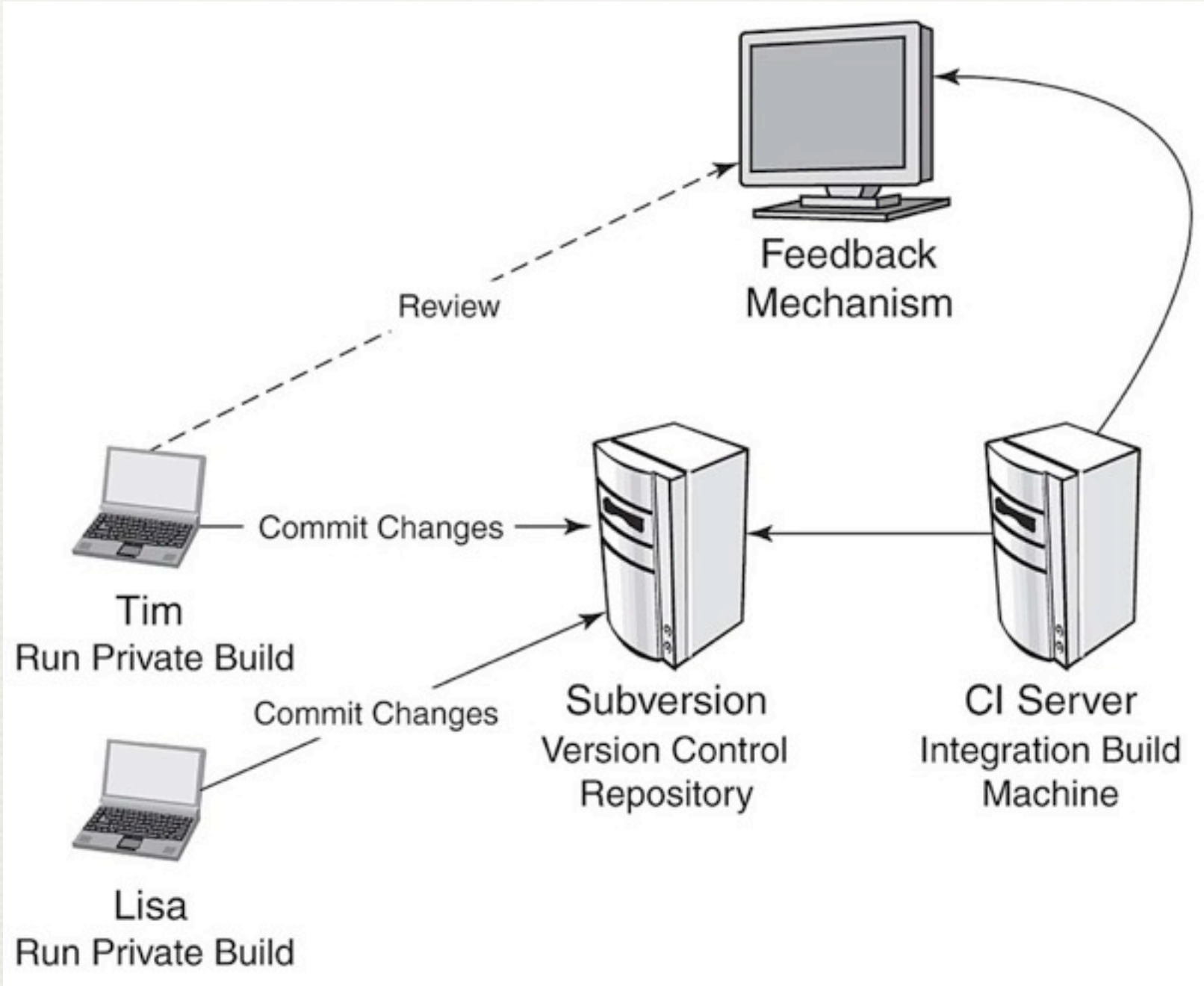
Integração Contínua



Introdução

"Prática de desenvolvimento de software onde os membros de um time integram seu trabalho frequentemente, geralmente cada pessoa integra pelo menos diariamente. Cada integração é verificada por um *build* automatizado (incluindo testes) para detectar erros de integração o mais rápido possível."

Martin Fowler



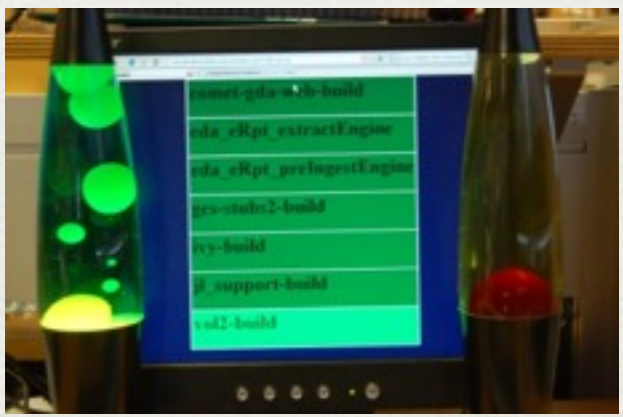
Fonte:www.**javaworld**.com/**javaworld**/jw-06-2007/jw-06-awci.html

O que é necessário

- Uma conexão com um repositório
- Um script para fazer o build
  - Compilação do código fonte
  - Integração do banco de dados
  - Testes, inspeções
  - Implantação
  - Documentação e Feedback

O que é necessário

- Alguns mecanismo de fornecimento de feedback



Fonte: www.hostfrontier.com/.../**agile**.../integrating-**lava-lamps**-to-cruisecontrol.html

- Um processo para integrar as mudanças do código fonte (manual / automático)

Por que utilizar?

- Redução de riscos
  - Defeitos são detectados e corrigidos mais rapidamente;
  - A "saúde" do software é mensurável;
  - Reduz suposições.
- Redução de processos repetitivos

Por que utilizar?

- Geração de software pronto para a implantação em qualquer lugar e em qualquer hora
- Melhoria da visibilidade do projeto
  - Decisões eficientes;
  - Observações de tendências;
- Melhora a confiança da equipe sobre o produto

Principais reclamações

- Aumento do *overhead* na manutenção do sistema de CI;
- Muitas mudanças;
- Muitas falhas no *build*;
- Custo adicional de hardware/software;
- Desenvolvedores deveriam fazer estas atividades;

Como CI complementa outras práticas

- Testes;
- Padronização;
- Refatoração;
- *Small releases*;
- Propriedade coletiva;

Práticas que ajudam

- *Commit* frequente
- Não faça *commit* de código quebrado
- Conserte os *builds* quebrados imediatamente
- Escreva testes automatizados
- Todos os testes e inspeções devem passar
- Execute *builds* privados
- Evite pegar código quebrado

Cruise Control

- Ferramenta livre gratuita
- Consiste de dois componentes
  - Build loop (roda como um serviço)
  - Status dashboard (roda como uma aplicação web)
- Suporta diversos controle de versões;
- Trabalha com qualquer ferramenta de build que pode ser parseada (Ant, Maven)

Cruise Control

- Multi plataforma
- Fácil de configurar utilizando plugins
- Ótimo suporte
  - Manuais, tutoriais, webcast, etc...

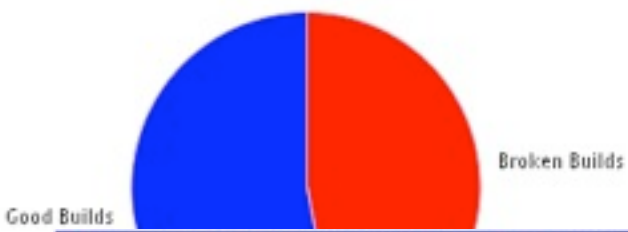
Cruise Control

- Funciona na forma básica:
  - Desenvolvedor faz commit no controle de versão
  - Servidor de IC pega a última versão do repositório
  - Se for detectado alguma mudança:
    - Roda o build, faz o log, cria os artefatos e publica os resultados.

Cruise Control

Number of Build Attempts 111
Number of Broken Builds 52
Number of Successful Builds 59

Breakdown of build types



Build Results | Test Results | XML Log File | Metrics | Control Panel

BUILD FAILED

Ant Error Message:	Return code is 1
Date of build:	11/12/2007 09:16:39
Time to build:	0 minute(s) 16 second(s)
Last changed:	11/09/2007 22:16:05
Last log entry:	CDP-1029 CDP-2366 move isVirtualAsset information to

Build Artifacts

Initial Messages

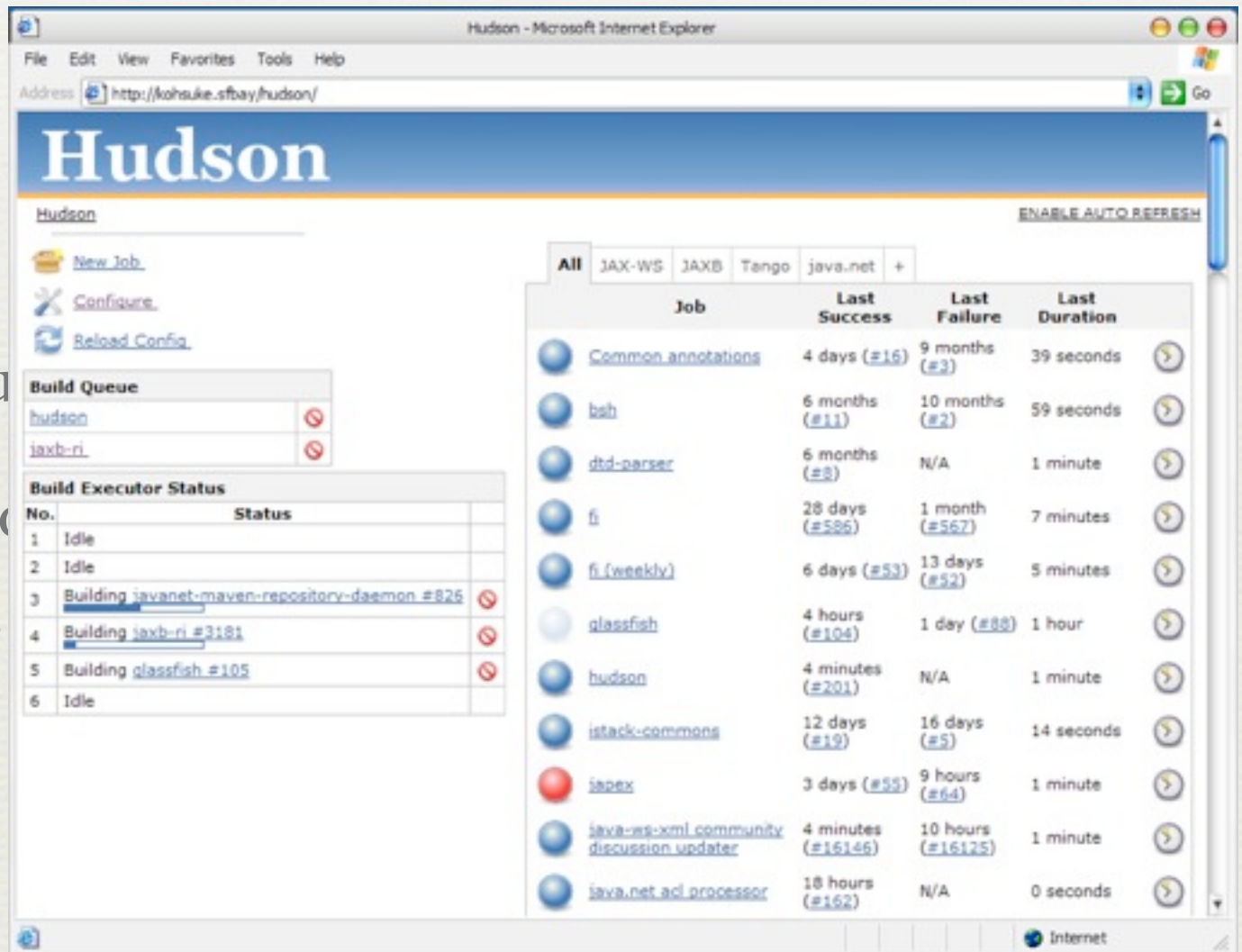
[INFO] Scanning for projects...
[INFO] --
[INFO] Building muse Storefront V3

Modifications since last successful build: (54)

modified	mscharl	pom.xml	11/08/2007 16:24:22	[maven-release-plugin] prepare for next development iteration
modified	mscharl	pom.xml	11/08/2007 16:24:17	[maven-release-plugin] prepare release rosa-v3-xfire-server-8-0-3
modified	mscharl	pom.xml	11/08/2007 16:21:08	moved constants from ContentService to ContentConstants -> fixed tests
modified	mscharl	src/test/java/com/threeunited/content/rosa/v3/impl/ContentServiceTest.java	11/08/2007 16:21:08	moved constants from ContentService to ContentConstants -> fixed tests
modified	mscharl	src/test/java/com/threeunited/content/rosa/v3/impl/ExceptionHandlingTest.java	11/08/2007 16:21:08	moved constants from ContentService to ContentConstants -> fixed tests
modified	mscharl	src/test/java/com/threeunited/content/rosa/v3/impl/ImageServiceTest.java	11/08/2007 16:21:08	moved constants from ContentService to ContentConstants -> fixed tests
modified	mscharl	src/test/java/com/threeunited/content/rosa/v3/impl/PerformanceTest.java	11/08/2007 16:21:08	moved constants from ContentService to ContentConstants -> fixed tests

Hudson

- Cru
- Hu
- ....?



Hudson

- Software Livre gratuito
- Fácil de instalar
- Fácil de configurar
- Monitoramento por RSS / E-mail / IM
- Suporte a *plugin*

Dúvidas?