Test-Driven Development on a Large Scale Project
Speaker

Jan Van Reusel
Speaker

Jan Van Reusel
Speaker
Jan Van Reusel
Speaker
Jan Van Reusel
Speaker

Jan Van Reusel
What keeps me awake at night?
...sometimes...
...more often...
### CruiseControl Status Page

<table>
<thead>
<tr>
<th>Project</th>
<th>Current status</th>
<th>Last build result</th>
<th>Last build time</th>
<th>Last successful build time</th>
<th>Last label</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Total</th>
<th>3</th>
<th>failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed</td>
<td>3</td>
<td>100%</td>
</tr>
</tbody>
</table>

Listing generated at 19:30
...summer 2006...
This Session

What did we do about it?
Context

Ventouris Renewal Program
8 customers

60 team size

100 my

3 year
Why TDD?

- Design
- Documentation
- Scope
- Trust
Test Strategy
Replace code on which SUT depends
http://xunitpatterns.com
Control indirect input
Verify indirect output
http://xunitpatterns.com
Object Mother
Client Presentation Domain Service Infrastructure RDBMS

Test Strategy

Web User Interfaces

Layered Architectures

Test Strategy

Developer Sandbox

Object Mother

Test Helper

Client

Presentation

Service

Domain

Data Source

Infrastructure

Browser

Faces Servlet

JSF Component

Managed Bean

Faces Component

Session Facade

Facade Impl

Business Object

Persistence Manager

Mock Object

Gateway

Mock Object or Test Stub

Humble Object

Business Object

JDO

RDBMS
JUnit Test Process

1. Launches

Selenium Driver

Selenium Browser Bot

AUT

Selenium Server

Selenese Proxy WebApp

Web Server

AUT WebApp

http://www.openqa.org/selenium
JUnit Test Process

1. Launches
2. Requests command

Browser Process

1. Launches Selenium Driver

Selenese Proxy WebApp

AUT WebApp

AUT

Selenium Browser Bot

Web Server

Selenium Server

Selenium Driver

AUT

Selenese Proxy WebApp

http://www.openqa.org/selenium

Developer Sandbox

Test Strategy

Web User Interfaces

Layered Architectures
JUnit Test Process

1. Launches
2. Requests command
3. Waits
4. “Go to AUT” & request result

Selenium Driver

Selenium Browser Bot

AUT

Selenese Proxy WebApp

AUT WebApp

Developer Sandbox

Test Strategy

Web User Interfaces

Layered Architectures

http://www.openqa.org/selenium
JUnit Test Process

1. Launches
2. Requests command
3. Waits
4. “Go to AUT” & request result

Selenium Driver

Selenium Browser Bot

AUT

Selenese Proxy WebApp

AUT WebApp

Web User Interfaces

Layered Architectures

http://www.openqa.org/selenium
JUnit Test Process

1. Launches
2. Requests command
3. Waits
4. “Go to AUT” & request result
5. “Go to AUT”

Web User Interfaces
Layered Architectures

http://www.openqa.org/selenium
JUnit Test Process

1. Launches
2. Requests command
3. Waits
4. “Go to AUT” & request result

Browser Process

1. Launches
2. Requests command
3. Waits
4. “Go to AUT”
5. “Go to AUT”
6. “Go to AUT”
7. HTTP Request

Selenium Driver

Selenium Browser Bot

AUT

Selenium Server

Selenese Proxy WebApp

Web Server

AUT WebApp

http://www.openqa.org/selenium
JUnit Test Process

1. Launches

2. Requests command

3. Waits

4. “Go to AUT” & request result

5. “Go to AUT”

6. “Go to AUT”

7. HTTP Request

8. HTTP Response

9. Gets result

AUT WebApp

Web Server

AUT

Selenium Browser Bot

Selenium Server

Selenium Proxy WebApp

Browser Process

Developer Sandbox

Test Strategy

Web User Interfaces

Layered Architectures

http://www.openqa.org/selenium
JUnit Test Process

1. Launches
2. Requests command
3. Waits
4. "Go to AUT" & request result
5. "Go to AUT"
6. "Go to AUT"
7. HTTP Request
8. HTTP Response
9. Gets result
10. Returns result & requests command

Selenium Driver

Selenium Browser Bot

AUT

Selenese Proxy WebApp

AUT WebApp

Web Server

Selenium Server

Test Strategy

Layered Architectures

Web User Interfaces

http://www.openqa.org/selenium
JUnit Test Process

1. Launches
2. Requests command
3. Waits
4. “Go to AUT” & request result
5. “Go to AUT”
6. “Go to AUT”
7. HTTP Request
8. HTTP Response
9. Gets result
10. Returns result & requests command
11. Waits

Web User Interfaces
Layered Architectures

http://www.openqa.org/selenium
JUnit Test Process

1. Launches
2. Requests command
3. Waits

Selenium Driver

4. “Go to AUT” & request result
5. “Go to AUT”
6. “Go to AUT”
7. HTTP Request
8. HTTP Response

Selenium Browser Bot

9. Gets result
10. Returns result & requests command
11. Waits
12. Result

AUT

AUT WebApp

Web User Interfaces

Layered Architectures

Developer Sandbox

http://www.openqa.org/selenium
JUnit Test Process

1. Launches
2. Requests command
3. Waits
4. “Go to AUT” & request result
6. “Go to AUT”
7. HTTP Request
8. HTTP Response
9. Gets result
10. Returns result & requests command
11. Waits
12. Result
13. Asserts on result

Selenium Driver

Selenium Browser Bot

AUT

Selenium Server

Selenese Proxy WebApp

Web Server

AUT WebApp
import com.thoughtworks.selenium.*;
import junit.framework.*;
public class GoogleTest extends TestCase {
    private Selenium sel;
    public void setUp() {
        sel = new DefaultSelenium("localhost",
                               4444, "*firefox", "http://www.google.com");
        sel.start();
    }
    public void testGoogle() {
        sel.open("http://www.google.com/webhp");
        sel.type("q", "hello world");
        sel.click("btnG");
        sel.waitForPageToLoad("5000");
        assertEquals("hello world - Google Search", sel.getTitle());
    }
    public void tearDown() {
        sel.stop();
    }
}
Test Smells

- Buggy Tests
- Slow Tests
- High Test Maintenance Cost
Buggy Tests

- Build master
- Communication
- No Ownership
- Deployment Fails

Buggy Tests
Load factor

High Test Maintenance Cost
High Test Maintenance Cost

- Education
- Too Many Tests
- Hard To Test Code
- Obscure Test
- Fragile Tests
High Test Maintenance Cost
High Test Maintenance Cost

- Education
- Too Many Tests

- Hard To Test Code
- Obscure Test
- Mystery Guest
- Irrelevant Information
- Layers

- Fragile Tests
High Test Maintenance Cost
High Test Maintenance Cost
High Test Maintenance Cost
High Test Maintenance Cost
High Test Maintenance Cost

Study groups

Test coach

Education

High Test Maintenance Cost

Too Many Tests

Hard To Test Code

Obscure Test

Fragile Tests
High Test Maintenance Cost
Load factor

Slow Tests
Slow Tests

- Build machine
- General Fixture
- Too Many Tests
- Slow Component Usage
Slow Tests

- Build machine
- General Fixture
- Minimal Fixture
- Too Many Tests
- Slow Component Usage
Slow Tests

- Build machine
- General Fixture
- Too Many Tests
- Slow Component Usage

Application Server
- Database
- Layers

Slow Tests
Slow Tests

- Build machine
- General Fixture
- Too Many Tests
- Slow Component Usage
- Staged build
- Incremental build
- Run too frequently
- Too much overlap
Slow Tests

- Build machine
- General Fixture
- Minimal Fixture
- Staged build
- Incremental build
- Run too frequently
- Too Many Tests
- Slow Component Usage
- Application Server
- Database
- Layers

Too much overlap
Not Enough

2 HRS

build

time

! test

maintenance
Not Enough

Lost ability to respond to change
Not Enough

Too many tests

too much overlap
Not Enough

Slow component usage
Not Enough

<table>
<thead>
<tr>
<th>Test type</th>
<th>number</th>
<th>%</th>
<th>time</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>predeploy</td>
<td>5020</td>
<td>66</td>
<td>170</td>
<td>3</td>
</tr>
<tr>
<td>postdeploy</td>
<td>2606</td>
<td>34</td>
<td>5072</td>
<td>97</td>
</tr>
<tr>
<td>Test type</td>
<td>number</td>
<td>%</td>
<td>time</td>
<td>%</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>----</td>
<td>------</td>
<td>----</td>
</tr>
<tr>
<td>predeploy</td>
<td>5020</td>
<td>66</td>
<td>170</td>
<td>3</td>
</tr>
<tr>
<td>postdeploy</td>
<td>2606</td>
<td>34</td>
<td>5072</td>
<td>97</td>
</tr>
</tbody>
</table>

Reduce to smoke tests
Result!

10 minute build
Continuous design

Educate your team

Be agile with test strategy

Conclusion
Agile Applied on The Largest J2EE Project in The Benelux
by
Johan Lybaert

Agile Development Workshop
by
Kathleen Cornelis

www.ardatissoftwarefactory.com