



Tools for Java test automation

Daniel Wildt
Software Developer
dwildt@gmail.com
<http://danielwildt.blogspot.com>

Agenda

- Test is a form of art
- Agile Methodologies and testing
- White box testing techniques (Unit Testing)
- Black box testing techniques (Functional)
- Ensuring Quality in the test process with Test Coverage
- Applying Test Driven Development and Continuous Integration
- Final words
- References
- Links

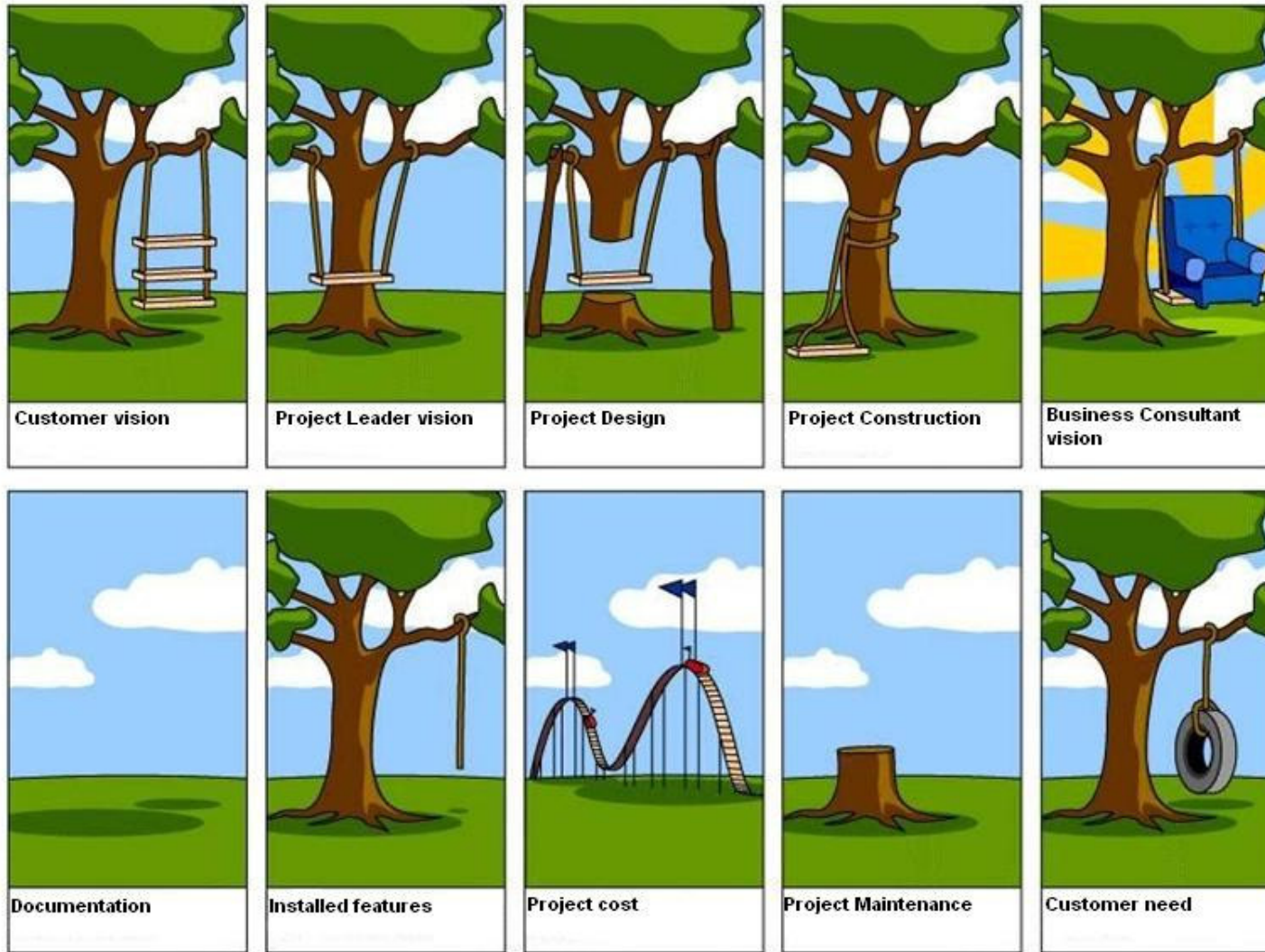
Test is a form of art

- You need to verify development code without seeing it (some times)
- You need to ensure that requirements are implemented in the same way they were defined
- You need to embrace change (focus on customer)
- You need to test faster to deliver faster (short cycles)
- You need to apply quality while testing
- You need to have a good coverage of source/requirements while testing
- You need to be creative while finding spots to test

Agile Methodologies and Testing

- Agile Methodologies are based on disciplines and practices that focus on principles like these (from Agile Manifesto principles):
 - Satisfy the customer as highest priority, through early and continuous delivery of valuable software.
 - Business people and developers working together daily throughout the project
 - Measure the progress of a project based on working software, its primary measure
 - Self organizing teams, motivated individuals, communication and simplicity as strong values
 - Technical excellence is wanted to enhance agility.
 - Continuous improvement, to tune and adjust teams to become more effective.

Agile Methodologies and Testing



Agile Methodologies and Testing

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

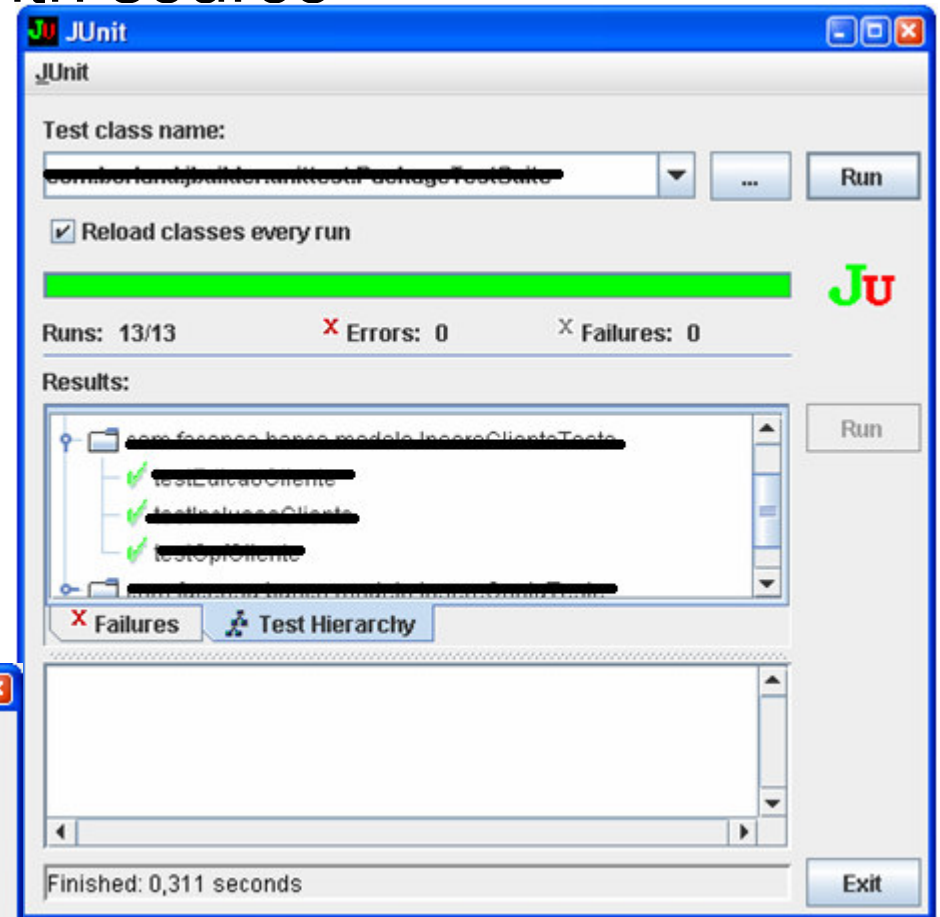
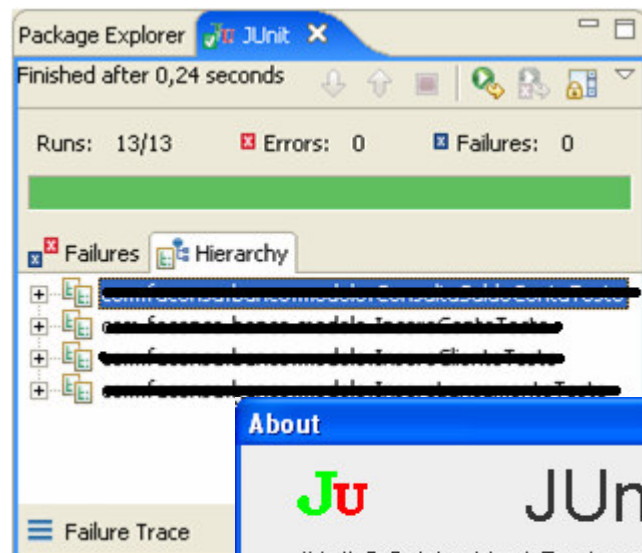
That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

White box testing techniques (Unit Testing)

- Source grow together with source
 - TDD

JUnit



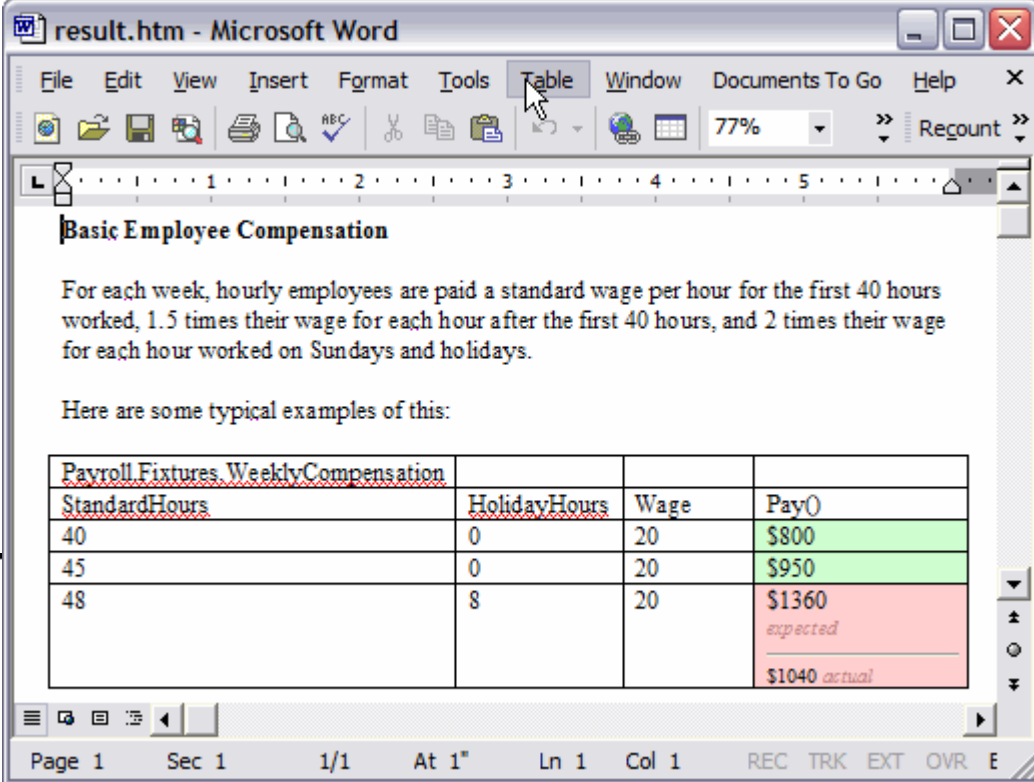
White box testing techniques (Unit Testing)

- Easy integration for Business - FIT



```
public class WeeklyCompensation : ColumnFixture
{
    public int StandardHours;
    public int HolidayHours;
    public Currency Wage;

    public Currency Pay()
    {
        WeeklyTimesheet timesheet = new WeeklyTimesheet(StandardHours, HolidayHours);
        return timesheet.CalculatePay(Wage);
    }
}
```



Basic Employee Compensation

For each week, hourly employees are paid a standard wage per hour for the first 40 hours worked, 1.5 times their wage for each hour after the first 40 hours, and 2 times their wage for each hour worked on Sundays and holidays.

Here are some typical examples of this:

Payroll Fixtures	WeeklyCompensation		
StandardHours	HolidayHours	Wage	Pay()
40	0	20	\$800
45	0	20	\$950
48	8	20	\$1360 <i>expected</i> \$1040 <i>actual</i>

White box testing techniques (Unit Testing)

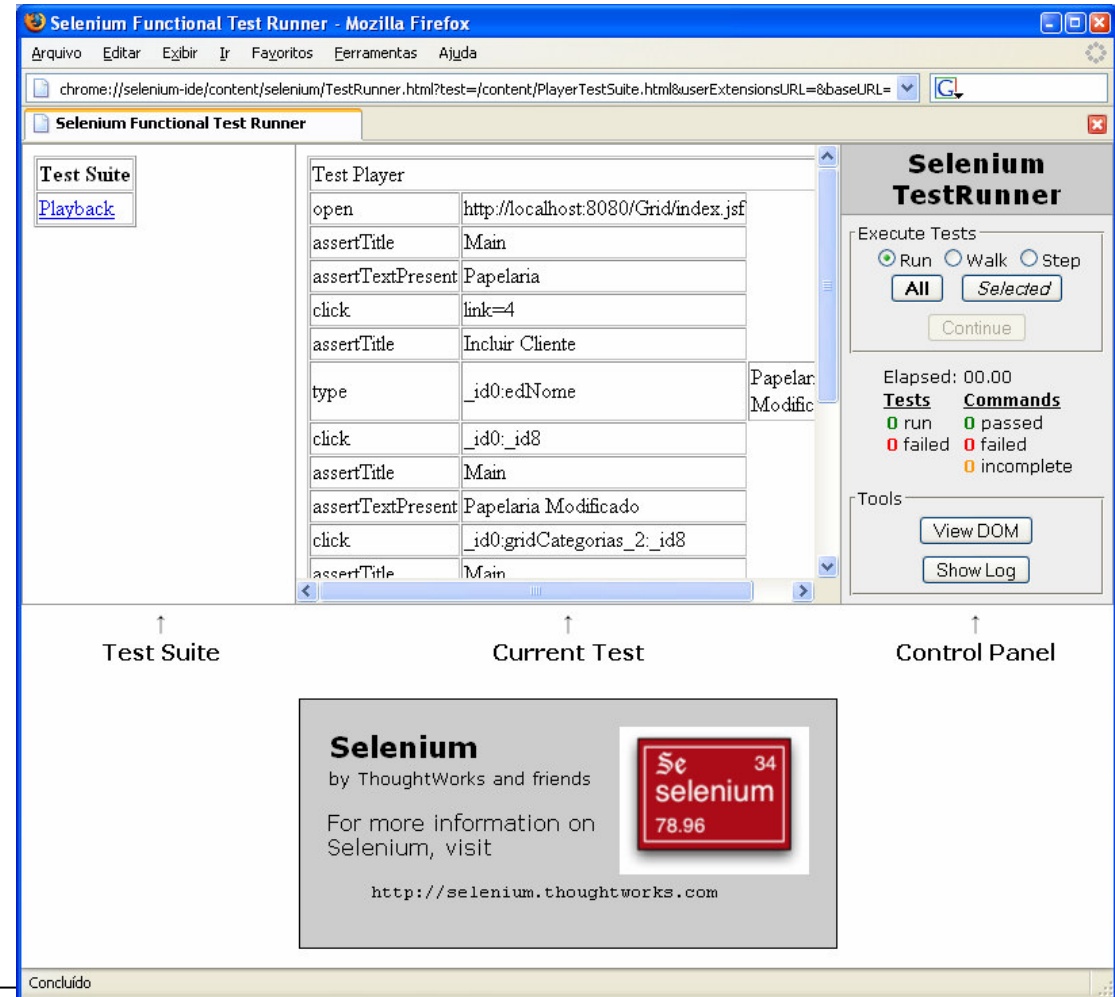
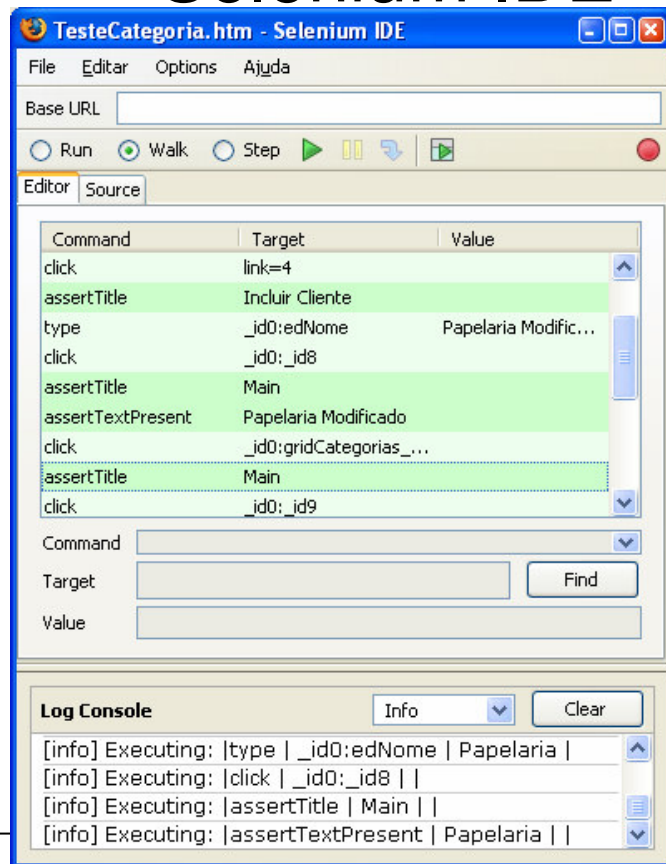
- You don't have it?
Mock!



```
public class ExampleTest {
    @Before
    public void setup() {
        mock = createMock(Collaborator.class);
        classUnderTest = new ClassUnderTest();
        classUnderTest.addListener(mock);
    }
    @Test
    public void addDocument() {
        mock.documentAdded("New Document");
        replay(mock);
        classUnderTest.addDocument("New Document",
            new byte[0]);
        verify(mock);
    }
}
```

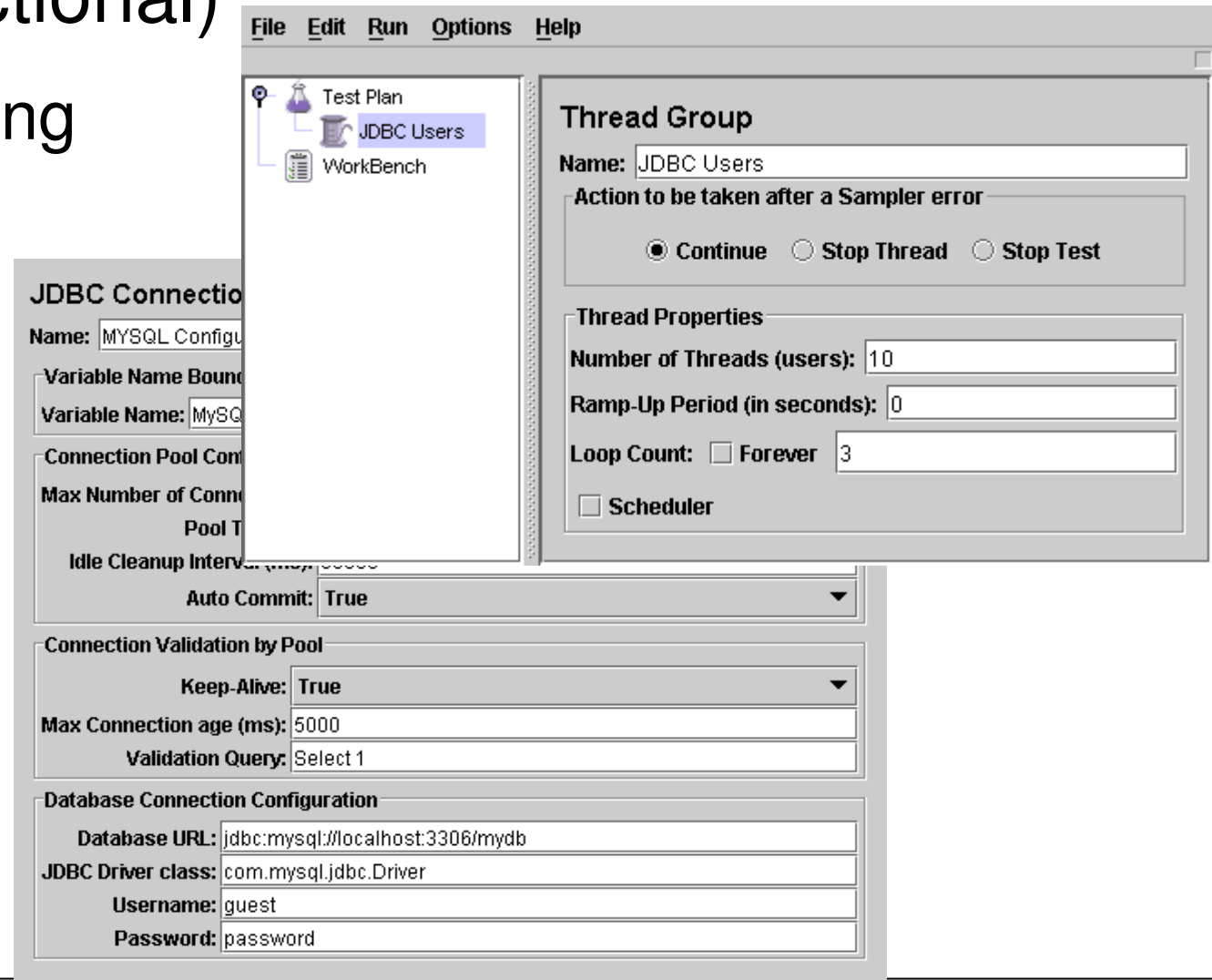
Black box testing techniques (Functional)

- Functional Test Selenium IDE



Black box testing techniques (Functional)

- Stress Testing
- JMeter



The screenshot shows the JMeter configuration interface. The main window has a menu bar with 'File', 'Edit', 'Run', 'Options', and 'Help'. The left sidebar shows a tree view with 'Test Plan', 'JDBC Users', and 'WorkBench'. The right pane is titled 'Thread Group' and contains the following settings:

- Name:** JDBC Users
- Action to be taken after a Sampler error:** Continue, Stop Thread, Stop Test
- Thread Properties:**
 - Number of Threads (users):** 10
 - Ramp-Up Period (in seconds):** 0
 - Loop Count:** Forever, 3
 - Scheduler

The bottom pane shows the 'JDBC Connection' configuration:

- Name:** MYSQL Configur...
- Variable Name Bound...**
- Variable Name:** MySQ...
- Connection Pool Com...**
- Max Number of Conne...**
- Pool T...**
- Idle Cleanup Interv...**
- Auto Commit:** True
- Connection Validation by Pool:**
 - Keep-Alive:** True
 - Max Connection age (ms):** 5000
 - Validation Query:** Select 1
- Database Connection Configuration:**
 - Database URL:** jdbc:mysql://localhost:3306/mydb
 - JDBC Driver class:** com.mysql.jdbc.Driver
 - Username:** guest
 - Password:** password



Ensuring Quality in the test process with Test Coverage



EMMA Coverage Report (generated Tue May 18 22:20:04 CDT 2004)

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	98% (118/120)	66% (318/483)	81% (15517/19107)	77% (2651.4/3430)

OVERALL STATS SUMMARY

total packages: 1
total executable files: 31
total classes: 120
total methods: 483
total executable lines: 3430

```

172     tf = new JTextField();
173     tf.setText(new Integer(splitPane.getDividerSize()).toString());
174     tf.setColumns(5);
175     tf.getAccessibleContext().setAccessibleName(getString
("SplitPaneDemo.divider_size"));
176     tf.addActionListener(new ActionListener() {
177         public void actionPerformed(ActionEvent e) {
178             String value = ((JTextField)e.getSource()).getText();
179             int newSize;
180
181             try {
182                 newSize = Integer.parseInt(value);
183             } catch (Exception ex) {
184                 newSize = -1;
185             }
186             if(newSize > 0) {
187                 splitPane.setDividerSize(newSize);
188             } else {
189                 JOptionPane.showMessageDialog(splitPane,
getString
("SplitPaneDemo.invalid_divider_size"),
getString("SplitPaneDemo.error"),
JOptionPane.ERROR_MESSAGE);
193             }
194         }
195     });

```

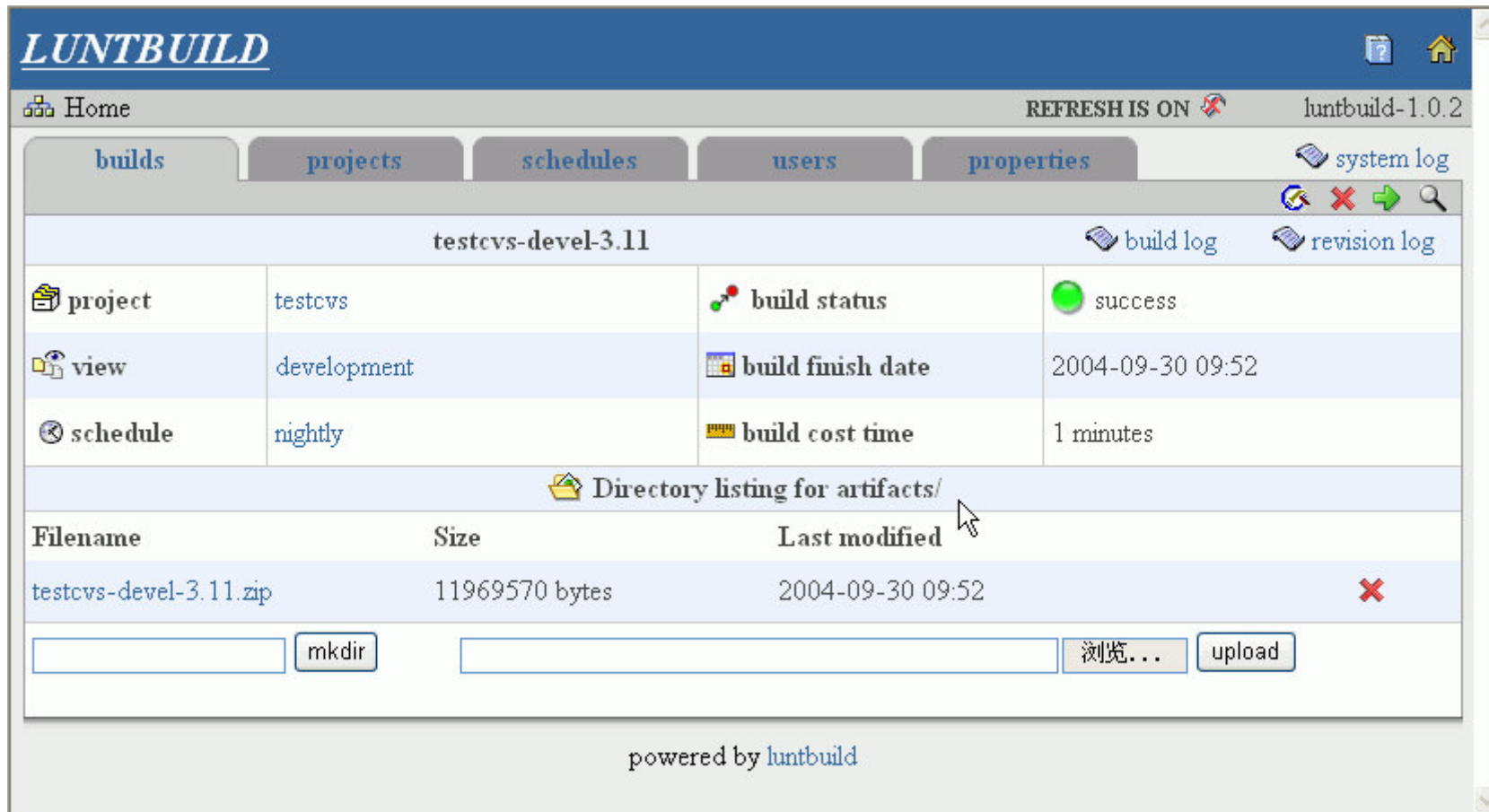
COVERAGE BREAKDOWN BY PACKAGE

name	class, %
default package	98% (118/120)

[all classes]

EMMA 2.0.4015 (stable) (C) Vlad

Applying Test Driven Development and Continuous Integration



The screenshot shows the LUNTBUILD web interface. At the top, there's a navigation bar with tabs for 'builds', 'projects', 'schedules', 'users', and 'properties'. The current view is for a build named 'testcvs-devel-3.11'. Below the tabs, there's a table with build details:

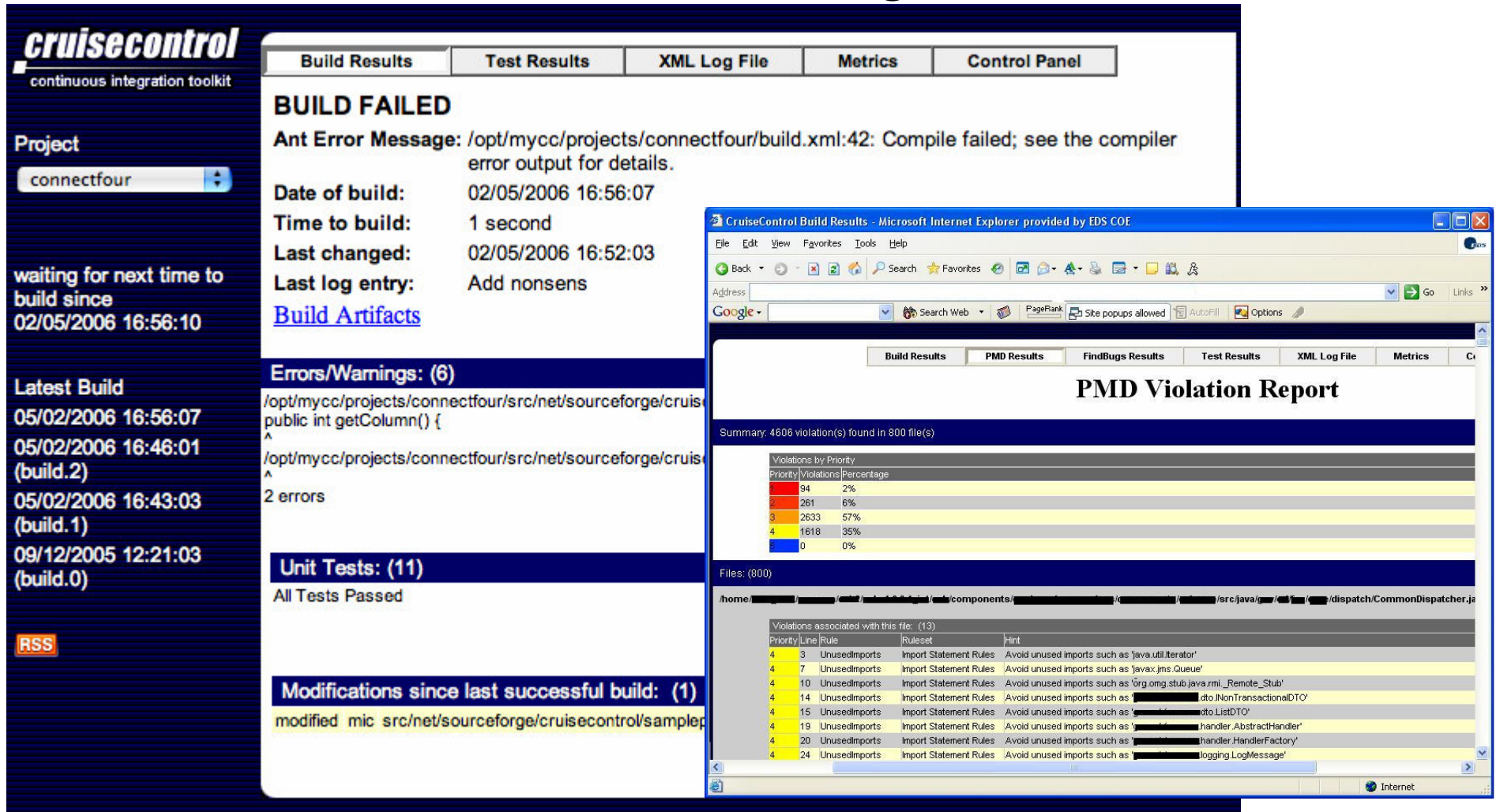
project	testcvs	build status	success
view	development	build finish date	2004-09-30 09:52
schedule	nightly	build cost time	1 minutes

Below the table, there's a section for 'Directory listing for artifacts/' with a table of files:

Filename	Size	Last modified
testcvs-devel-3.11.zip	11969570 bytes	2004-09-30 09:52

At the bottom of the interface, there are buttons for 'mkdir', '浏览...' (Browse...), and 'upload'. The footer of the interface says 'powered by luntbuild'.

Applying Test Driven Development and Continuous Integration



cruisecontrol
continuous integration toolkit

Project: connectfour

waiting for next time to build since 02/05/2006 16:56:10

Latest Build
05/02/2006 16:56:07
05/02/2006 16:46:01 (build.2)
05/02/2006 16:43:03 (build.1)
09/12/2005 12:21:03 (build.0)

[RSS](#)

BUILD FAILED
Ant Error Message: /opt/mycc/projects/connectfour/build.xml:42: Compile failed; see the compiler error output for details.
Date of build: 02/05/2006 16:56:07
Time to build: 1 second
Last changed: 02/05/2006 16:52:03
Last log entry: Add nonsens
[Build Artifacts](#)

Errors/Warnings: (6)
/opt/mycc/projects/connectfour/src/net/sourceforge/cruisecontrol/...
public int getColumn() {
^
/opt/mycc/projects/connectfour/src/net/sourceforge/cruisecontrol/...
^
2 errors

Unit Tests: (11)
All Tests Passed

Modifications since last successful build: (1)
modified mic src/net/sourceforge/cruisecontrol/sample...

PMD Violation Report
Summary: 4606 violation(s) found in 800 file(s)

Priority	Violations	Percentage
1	94	2%
2	261	6%
3	2633	57%
4	1618	35%
5	0	0%

Files: (800)

home/.../components/.../src/java/org/.../dispatch/CommonDispatcher.java

Priority	Line	Rule	Ruleset	Hint
4	3	UnusedImports	Import Statement Rules	Avoid unused imports such as 'java.util.Iterator'
4	7	UnusedImports	Import Statement Rules	Avoid unused imports such as 'javax.jms.Queue'
4	10	UnusedImports	Import Statement Rules	Avoid unused imports such as 'org.omg.stub.java.rmi._Remote_Stub'
4	14	UnusedImports	Import Statement Rules	Avoid unused imports such as '...dto.NonTransactionalDTO'
4	15	UnusedImports	Import Statement Rules	Avoid unused imports such as '...dto.ListDTO'
4	19	UnusedImports	Import Statement Rules	Avoid unused imports such as '...handler.AbstractHandler'
4	20	UnusedImports	Import Statement Rules	Avoid unused imports such as '...handler.HandlerFactory'
4	24	UnusedImports	Import Statement Rules	Avoid unused imports such as '...logging.LogMessage'

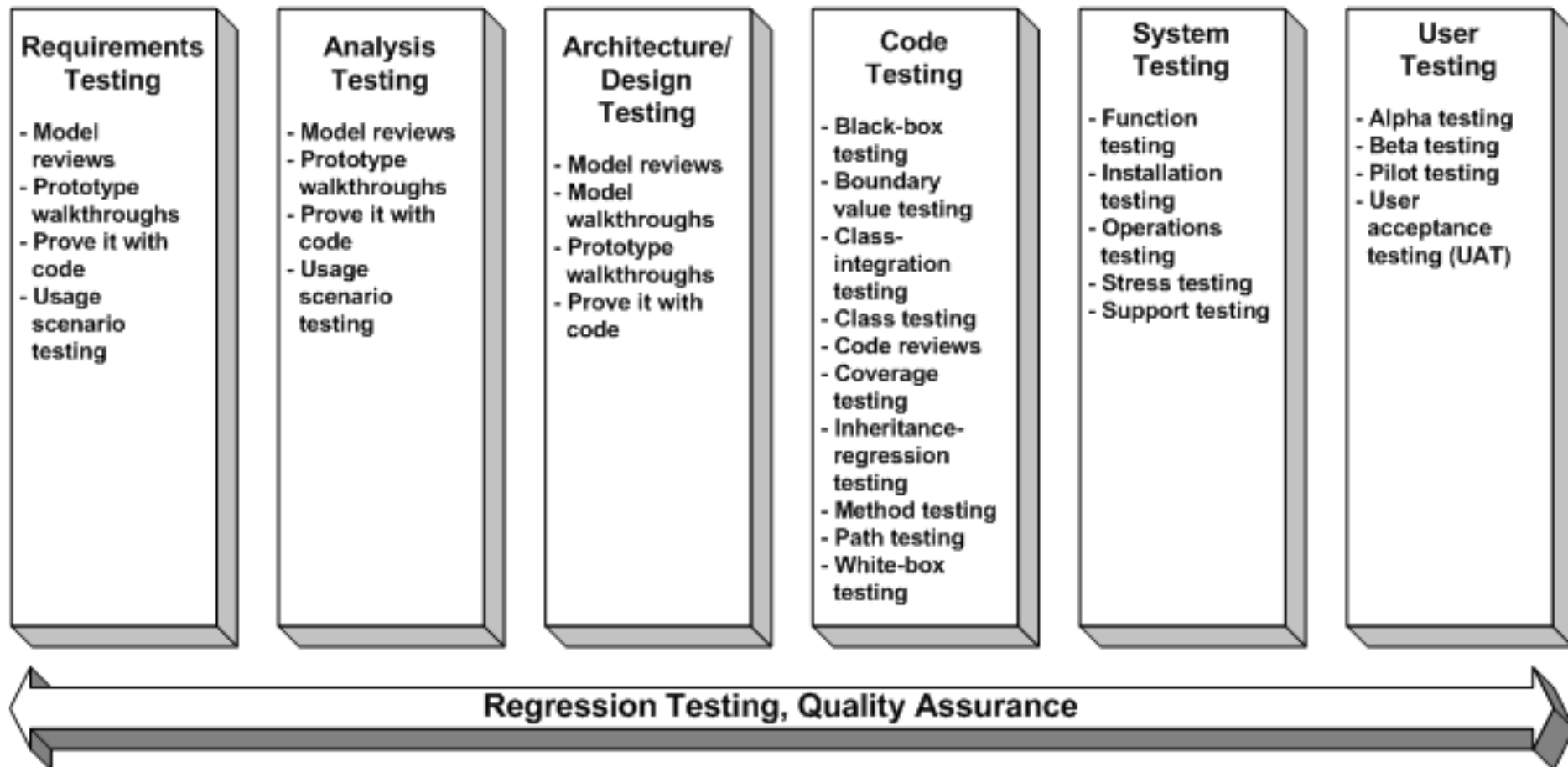
Final words

- Always think about this:

Verification
X
Validation
?

- **More: check CMMI – Level 3**

Final words



Copyright 2004 Scott W. Ambler

<http://www.ambysoft.com/essays/floot.html>

References

- Beck, Kent; Andres, Cynthia. Extreme Programming explained: embrace change. 2^a edition.
- Pressman, Roger S. Software Engineering.
- Agile Manifesto
<http://www.agilemanifesto.org>

References

- Tinkha, Andy; Kaner, Cem. Exploring Exploratory Testing.
<http://www.testingeducation.org/a/explore.pdf>
- Scott Ambler essay about FLOOT.
<http://www.ambysoft.com/essays/floot.html>
- Cyclomatic Complexity.
http://www.sei.cmu.edu/str/descriptions/cyclomatic_body.html

- JUnit – Unit Testing
 - <http://www.junit.org>
- JMeter – Stress Testing
 - <http://jakarta.apache.org/jmeter/>
- Emma – Code Coverage
 - <http://emma.sf.net>
- Selenium – Functional testing
 - <http://www.openqa.org/selenium/>
- PMD – Code Audit
 - <http://pmd.sf.net>

- CheckStyle –Code Audit
 - <http://checkstyle.sf.net>
- Easy Mock
 - <http://www.easymock.org/>
- FIT
 - <http://fit.c2.com/>
- Lunt Build
 - <http://luntbuild.javaforge.com/>
- Cruise Control
 - <http://confluence.public.thoughtworks.org/display/CC>



Questions?



Thanks!