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

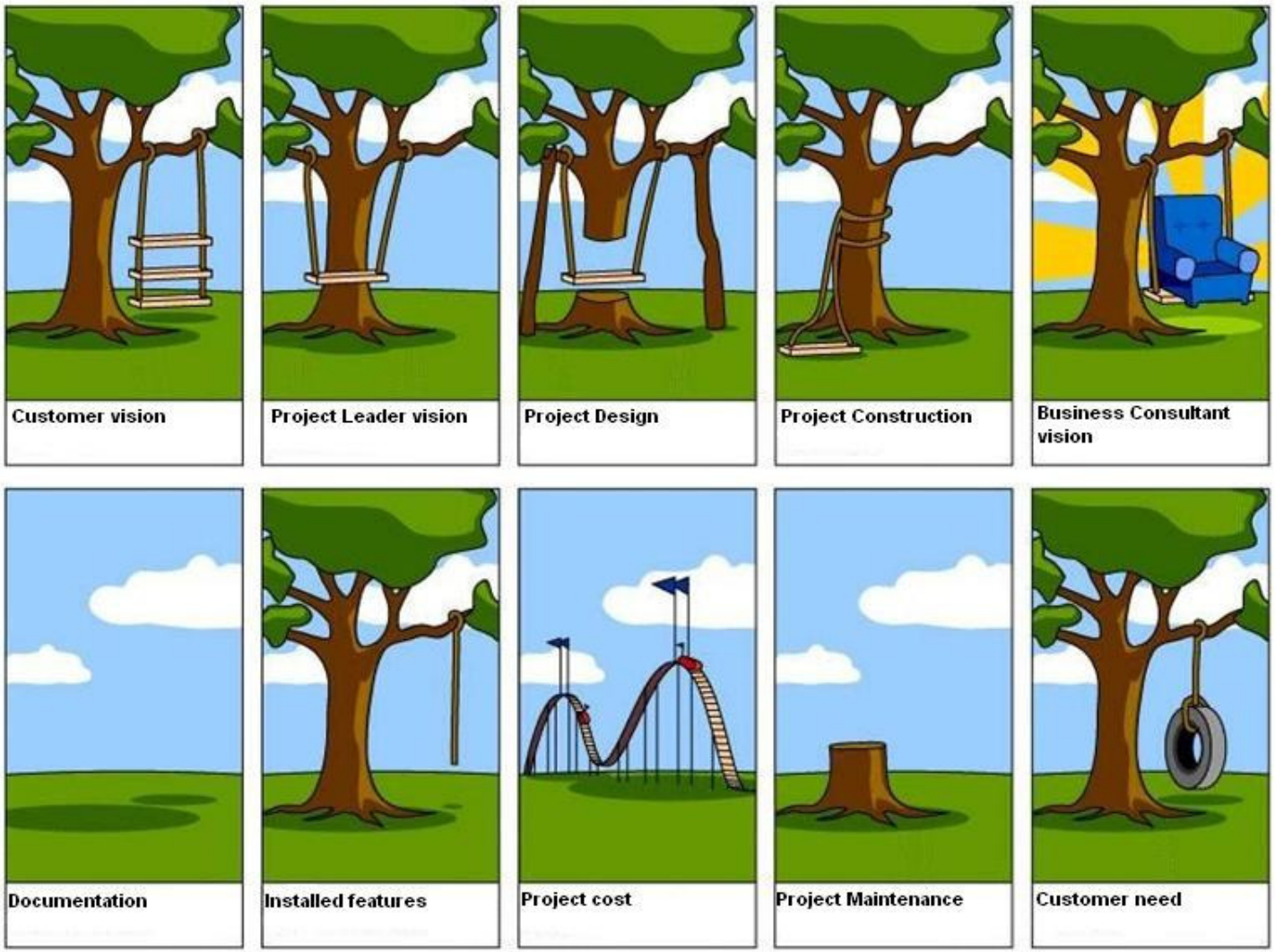
Test is a form of art

- 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

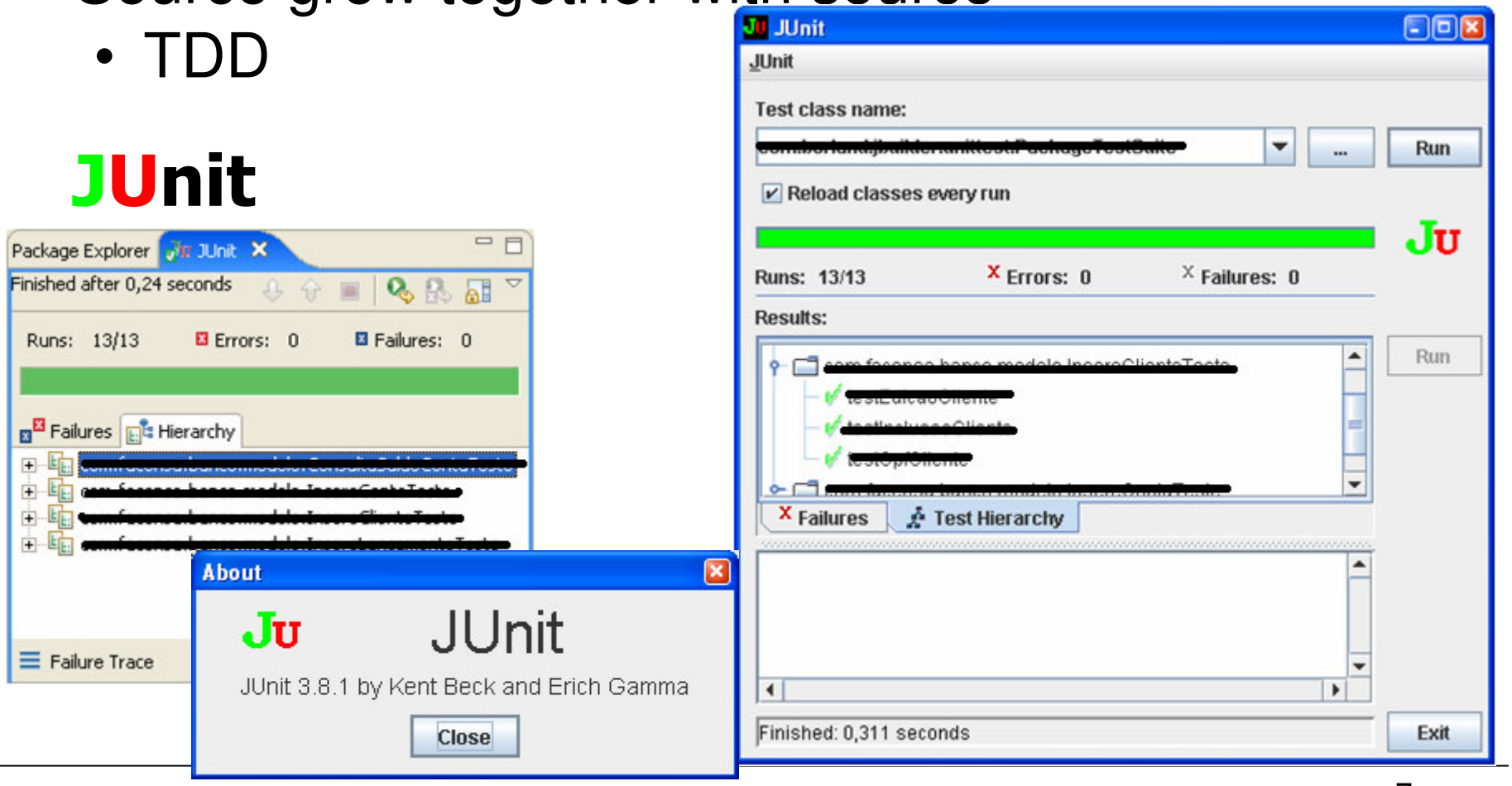


Agile Methodologies and Testing

White box testing techniques (Unit Testing)

- Source grow together with source
  - TDD

JUnit



White box testing techniques (Unit Testing)

- Easy integration for Business - FIT

result.htm - Microsoft Word

File Edit View Insert Format Tools Table Window Documents To Go Help

77% Recount

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 *expected* / $1040 *actual*

Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC TRK EXT OVR

```
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);
    }
}
```

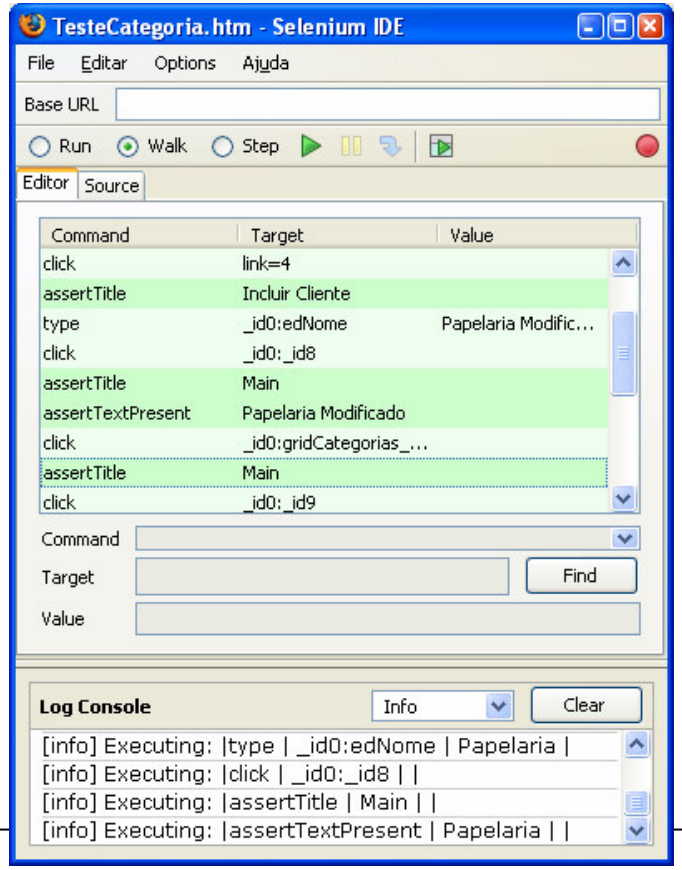
White box testing techniques (Unit Testing)

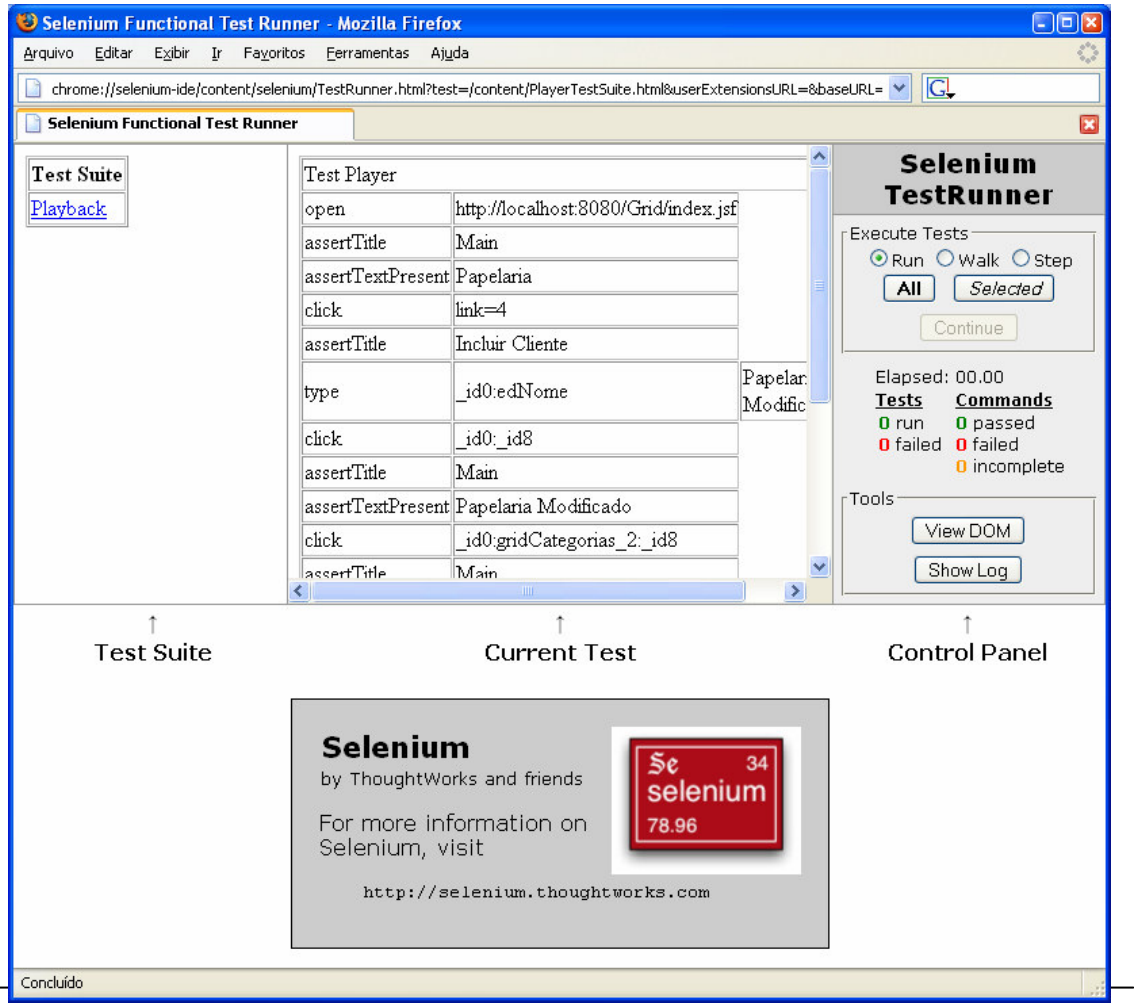
- You don't have it?
  Mock!


```
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)

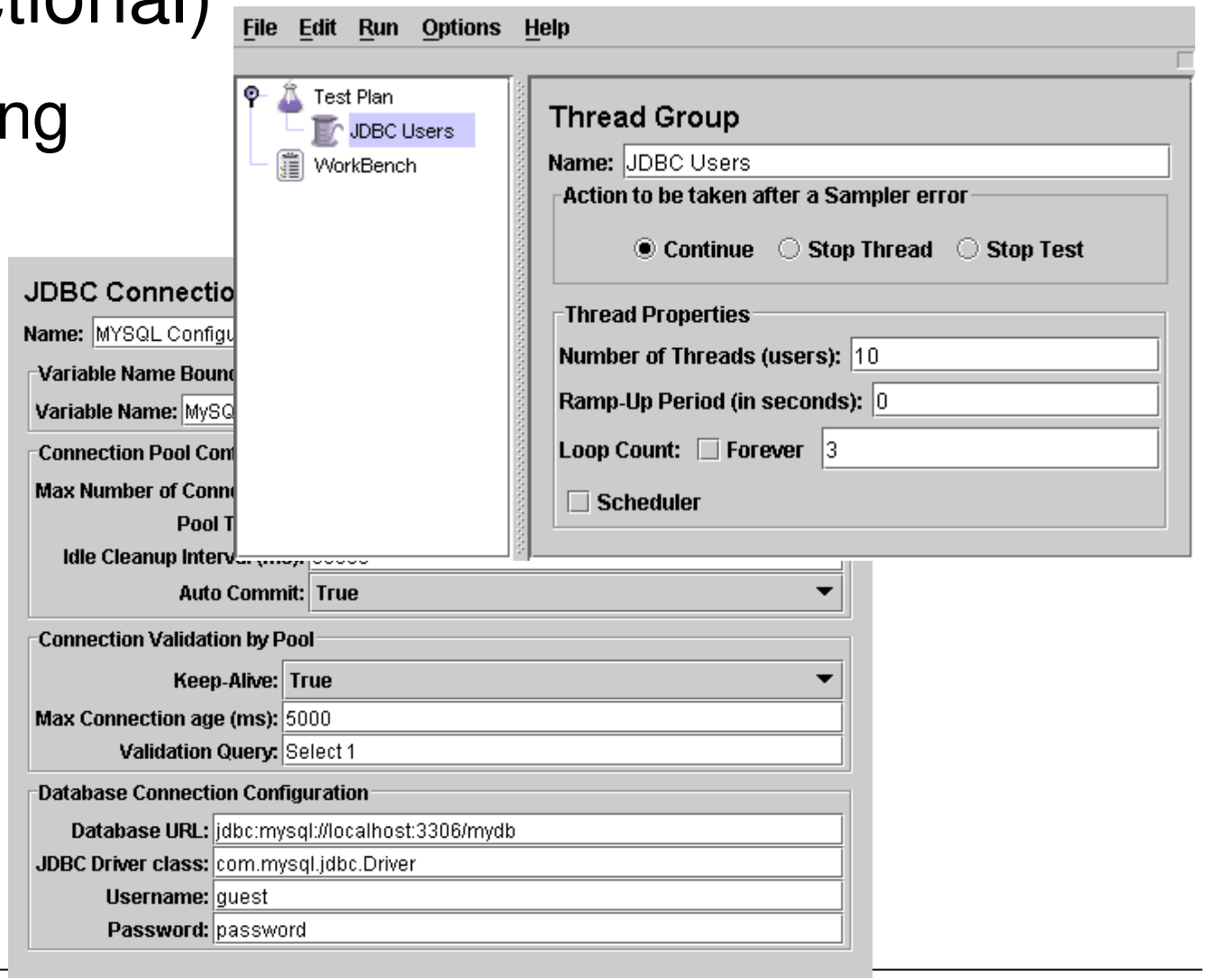
- Functional Test Selenium IDE





Black box testing techniques (Functional)

- Stress Testing
- JMeter



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

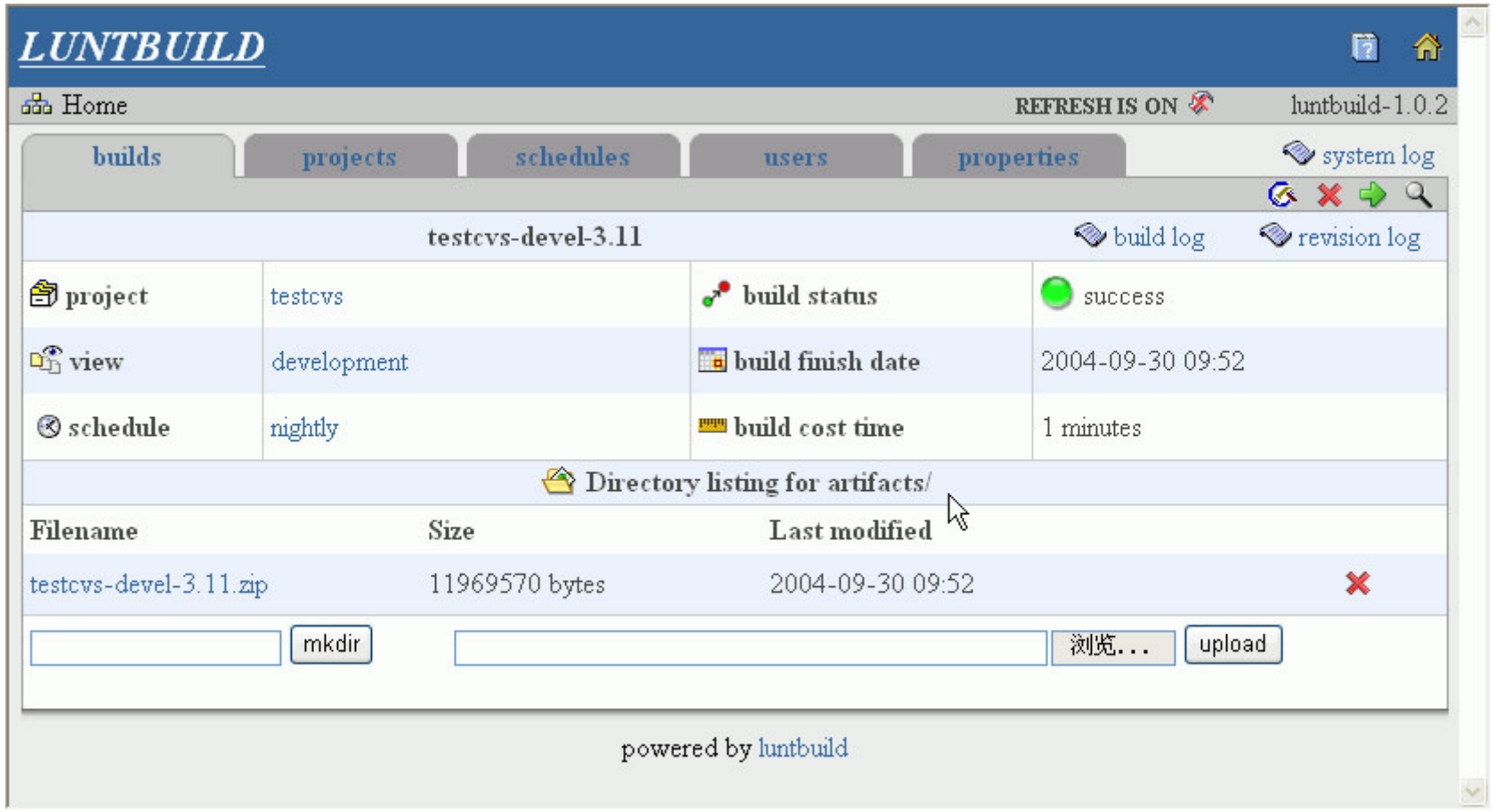
COVERAGE BREAKDOWN BY PACKAG

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

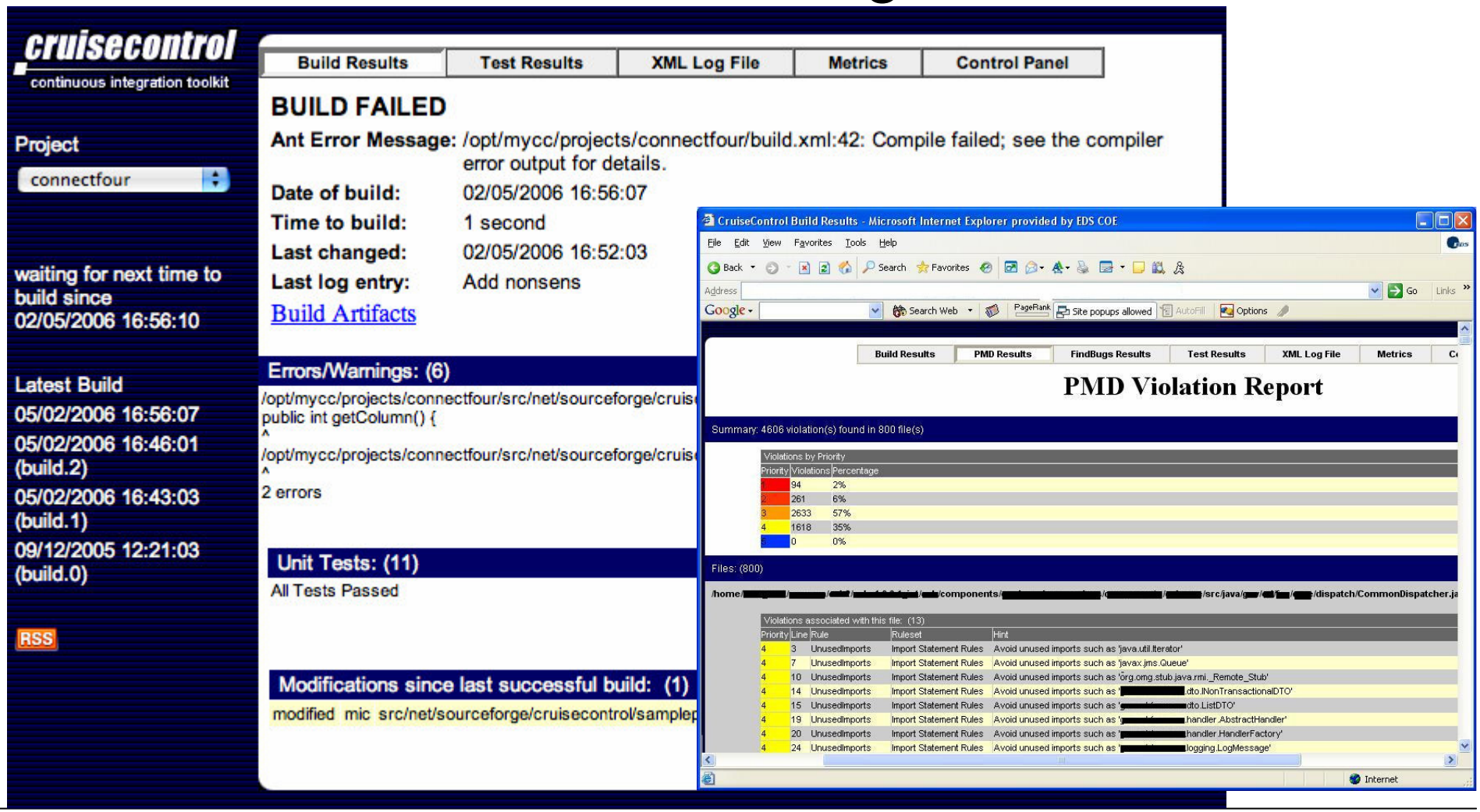
[all classes]
EMMA 2.0.4015 (stable) (C) Vlad

```
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,
190                                                   getString
("SplitPaneDemo.invalid_divider_size"),
191                                                   getString("SplitPaneDemo.error"),
192                                                   JOptionPane.ERROR_MESSAGE);
193                 }
194             }
195         });
```

Applying Test Driven Development and Continuous Integration



Applying Test Driven Development and Continuous Integration



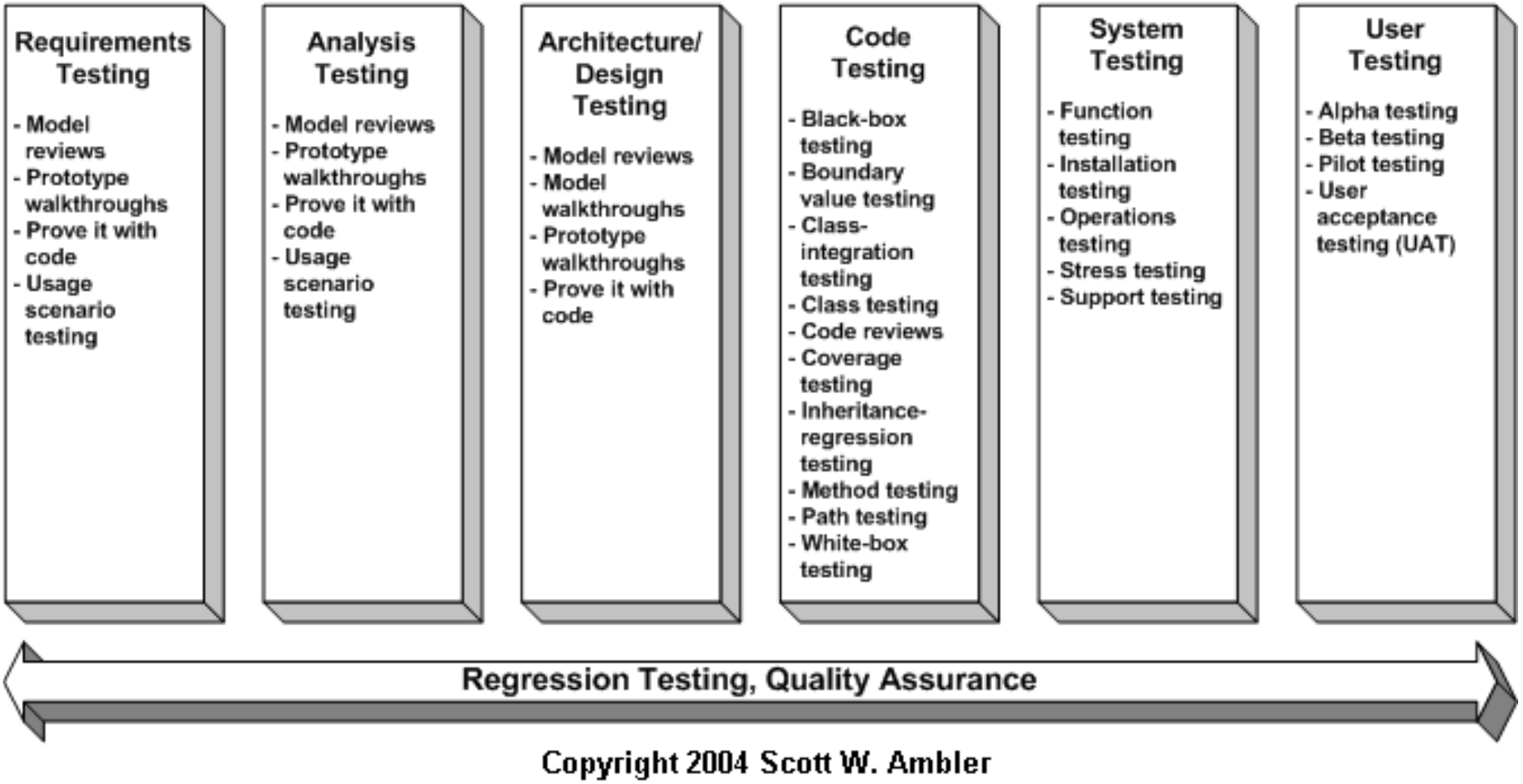
Final words

- Always think about this:

Verification
X
Validation
?

- **More: check CMMI – Level 3**

Final words



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

References

- Beck, Kent; Andres, Cynthia. Extreme Programming explained: embrace change. 2ª 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
- Ciclomatic Complexity. http://www.sei.cmu.edu/str/descriptions/cyclomatic_body.html

Links

- 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

Links

- 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!