

2.– 5. September 2013
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

XLS-based Testing!

Unit- und Integrationstests in Excel definieren

Matthias Reining

RGI Deutschland GmbH

Zu mir...

Matthias Reining
RGI Deutschland GmbH

Versicherungs-IT, Java EE, REST, JS-Frameworks

 @matthiasreining

 blog.matthias-reining.de

<http://about.me/matthiasreining>

Um was geht's hier?

Unit-Tests sind für jeden Entwickler im agilen Umfeld selbstverständlich. Nicht selten ist dies auch daher begründet, dass zum Teil auch Kunden eine Testabdeckung von über 80% in den Abnahmekriterien fordern. In diesem Vortrag wird gezeigt, wie man den Prozess der Testfallerstellung Richtung Excel verlagern kann und sich somit auch von Nicht-Entwicklern (bspw. dem Fachbereich) Unterstützung einholen kann. Dabei werden verschiedene Möglichkeiten und Java-Frameworks präsentiert, wie Excel in den Testprozess integriert werden kann. Um die Mächtigkeit xls-basierender Tests zu demonstrieren, sind viele kleine Beispiele in den Vortrag integriert.

Um was geht's hier?

Unit-Tests sind für jeden Entwickler im agilen Umfeld selbstverständlich. Nicht selten ist dies auch daher begründet, dass zum Teil auch Kunden eine Testabdeckung von über 80% in den Abnahmekriterien fordern. In diesem Vortrag wird gezeigt, wie man den Prozess der **Testfallerstellung** Richtung **Excel** verlagern kann und sich somit auch von Nicht-Entwicklern (bspw. dem Fachbereich) Unterstützung einholen kann. Dabei werden verschiedene Möglichkeiten und **Java-Frameworks** präsentiert, wie Excel in den Testprozess integriert werden kann. Um die Mächtigkeit xls-basierender Tests zu demonstrieren, sind viele kleine **Beispiele** in den Vortrag integriert.

Ziel

ANTWORTEN AUF

Wozu

kann Excel beim Testen eingesetzt werden?

Wann

macht der Einsatz von Excel beim Testen Sinn?

Wie

kann man Excel in Tests integrieren und wie komplex ist dies?

Agenda

- Was wird getestet?
- Wie kann der Fachbereich einbezogen werden?
- Testen mit Excel
 - Wie?
 - junit-Integration
 - Reporting
 - Frameworks
- Aber...

Was wird getestet?

- Alles?
- Was Sinn macht?
- Was der Kunde sich so ausdenkt?

Was wird getestet?

Was der Kunde sich so ausdenkt!

Wenn der Kunde sich schon Gedanken macht, dann sollten diese auch berücksichtigt werden!

Der Kunde wird Teil vom (Entwicklungs-) Team

Wie kann der Fachbereich einbezogen werden?

Verschiedenste Ansätze

- Workshops (Protokoll)
- Word
- Spezielle Test-Tools (HP Testmanagement Software, SQS Tools, ...)
- Fachbereich/BO bekommt Java Schulung und soll selbst Tests mitschreiben
- Lieferung von Testdaten (data in & expected), die automatisiert ausgeführt werden

Wie kann der Fachbereich einbezogen werden?

Lieferung von Testdaten inkl. Feedback der Testergebnisse

- Wiki / Website based
 - Robot Framework
 - FitNesse
 - ...

Exkurs: FitNesse

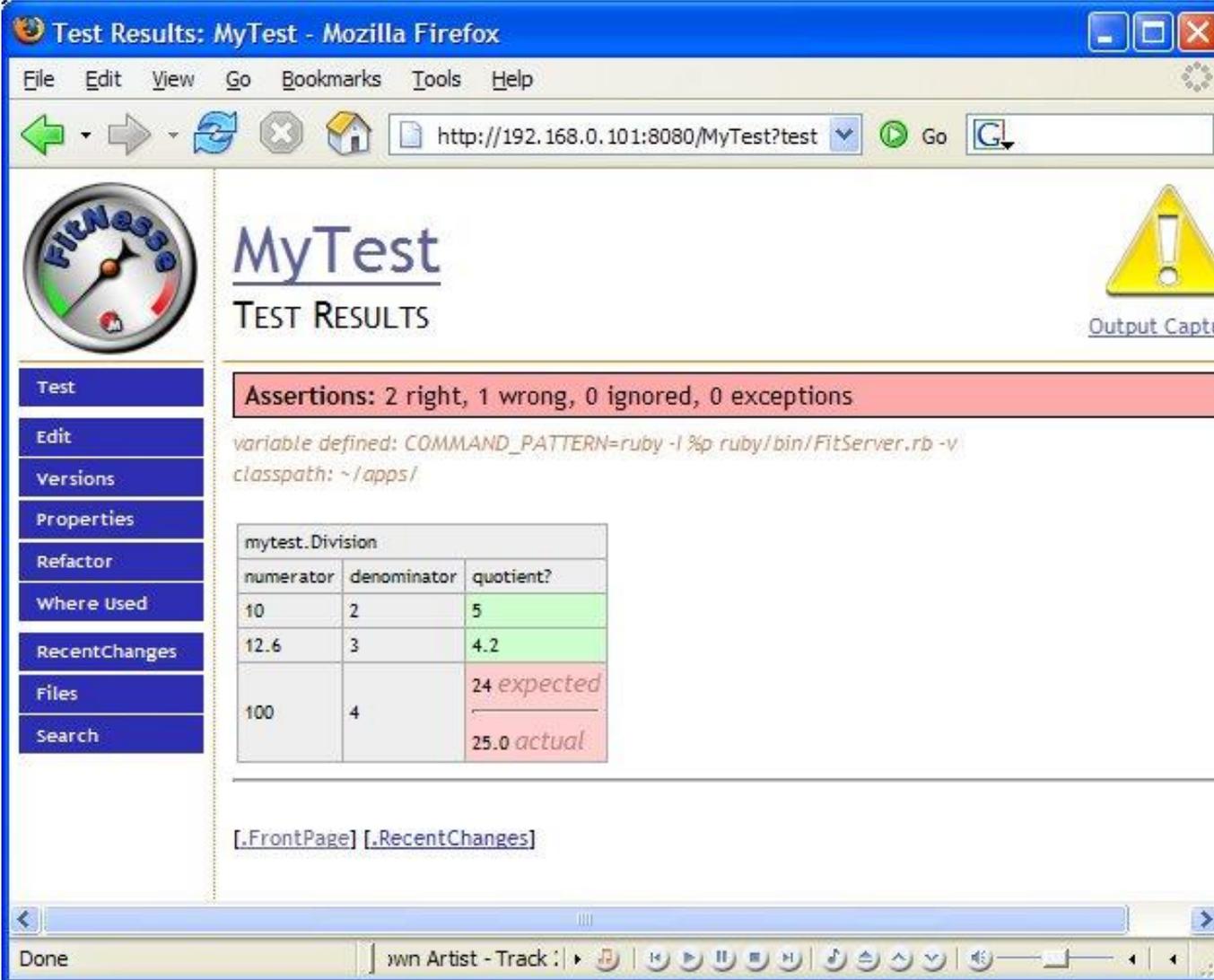
FitNesse is the acceptance test wiki.

In FitNesse, tests are expressed as tables of input data and expected output data. Here is one way to specify a few division tests in FitNesse:

eg.Division		
numerator	denominator	quotient?
10	2	5.0
12.6	3	4.2
22	7	~=3.14
9	3	<5
11	2	4<_ _<6
100	4	33

Exkurs: FitNesse

Ergebnis:



The screenshot shows a Mozilla Firefox browser window titled "Test Results: MyTest - Mozilla Firefox". The address bar shows the URL "http://192.168.0.101:8080/MyTest?test". The page content includes a FitNesse logo, a navigation menu on the left, and test results for "MyTest".

MyTest TEST RESULTS

Assertions: 2 right, 1 wrong, 0 ignored, 0 exceptions

variable defined: `COMMAND_PATTERN=ruby -I %p ruby/bin/FitServer.rb -v`
classpath: `~/apps/`

mytest.Division		
numerator	denominator	quotient?
10	2	5
12.6	3	4.2
100	4	24 <i>expected</i>
		25.0 <i>actual</i>

[.FrontPage] [RecentChanges]

Exkurs: FitNesse

The wiki markup for our table above (with some omissions) looks like this:

```
|eg.Division|  
|numerator|denominator|quotient?|  
|10 |2 |5 |  
|12.6 |3 |4.2 |  
|100 |4 |33 |
```

Exkurs: FitNesse

Note that if you don't like mucking about with a markup language at all, you can create your test tables in a spreadsheet such as Excel, copy them to the clipboard, and use the Spreadsheet to Fitnessse button on the Edit page to have FitNesse format your table properly for you.

Excel ist das Tool



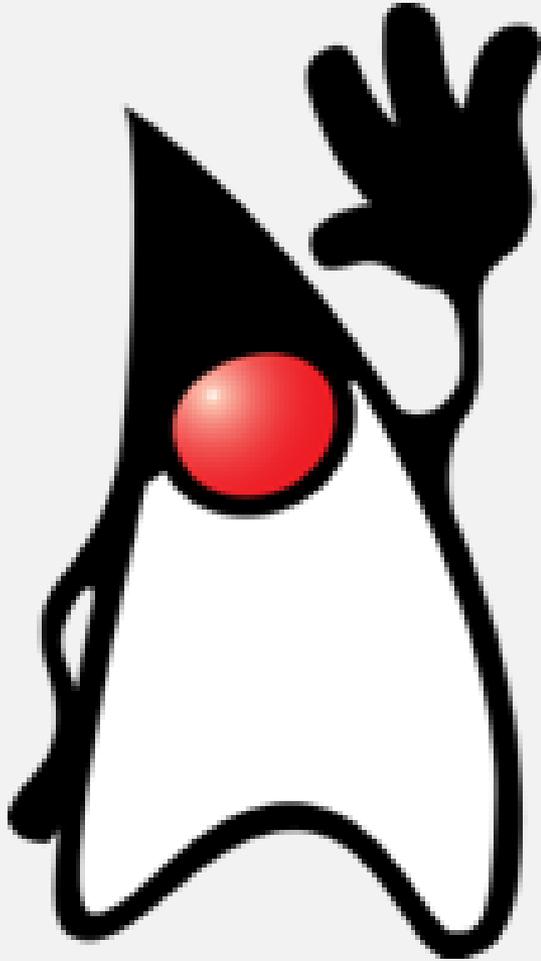
Testen mit Excel!

Agenda:

Testen mit Excel!

- Wie?
- junit-Integration
- Fachbereich & CI Server Integration
- Frameworks

Excel und Java...



Excel und Java...

- Java Excel API (jexcelapi)
- Aspose Cells
- Apache POI
- JACOB
- *und viele mehr*



<https://github.com/mr678/HC13XIsBasedTesting>

Xls Testing Frameworks

easytest

(<https://github.com/EaseTech/easytest>)

jexunit

(<https://github.com/fhm84/jexunit>)



<https://github.com/mr678/HC13XIsBasedTesting>

Integrationstest

- Fertige Frameworks?
- Selenium & Excel
 - Glue Code muss geschrieben werden.



<https://github.com/mr678/HC13XIsBasedTesting>

Aber...

- Im Enterprise Umfeld ist es oft nicht die Komplexität einer Methode, sondern das Zusammenspiel verschiedener Services in der passenden Reihenfolge.
- Ob der Roundtrip UI Action, Service, WFE / ESB, Asynchrone Jobs, UI Statusmeldung funktioniert, lässt sich oft nicht mehr testen von „Nicht-Entwicklern“

Aber trotzdem:

XLS basierte Tests eignen sich sehr gut für Anwendungen / Klassen / Methoden die,

- **Berechnungen** durchführen
- Viele **Fallunterscheidungen** beinhalten und dementsprechend viele mögliche Kombinationen von Eingangsparameter benötigen um eine gute **code-coverage** zu erhalten
- Wenn der **Fachbereich** eh schon mit Testfällen in Excel um die Ecke kommt...



...Danke