

FIT für Test getriebene Entwicklung

Akzeptanztests für Benutzer und
Programmierer

GI Regionalgruppe Dortmund 04.04.2011

Dipl.-Inform. (FH) Dirk Prüter

Dirk.Prueter@gmx.de

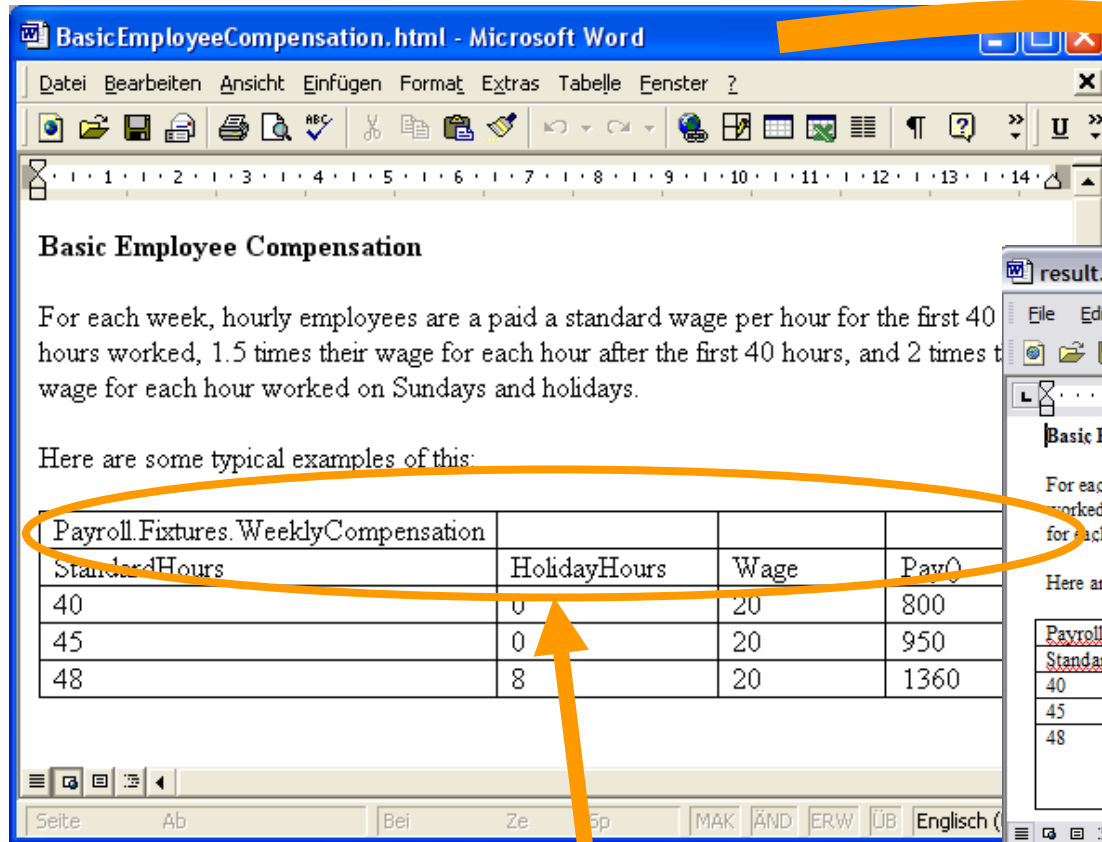
Überblick

- Was ist FIT
- Wie funktioniert FIT
- Wie arbeite ich Test getrieben
- Was leistet FIT nicht
- Was ist zu beachten
- Warum lohnt es sich, FIT anzuwenden

Was ist FIT

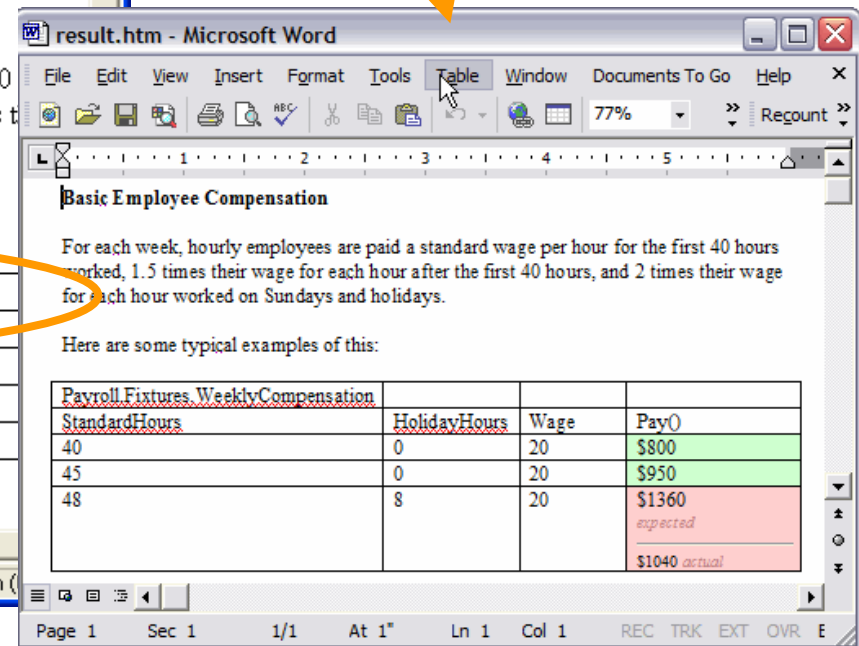
- Framework for Integrated Testing
- Ansatz für Test getriebene Entwicklung
- Dynamische Tests (Akzeptanztests)
- Testautomation
- Multi Platform (Java, C++, C#, Python)
- Open Source

Wie funktioniert FIT



Entwicklung „Fixture“

Ausführung „Runner“



Fixtures

ColumnFixture	checking rules and calculations
ActionFixture	step-by-step processing
RowFixture	checking sets of data
ReferenceFixture	referencing input files
Summary	test result summary

ColumnFixture

eg.Division		
numerator	denominator	quotient()
1000	10	100.0000
-1000	10	-100.0000
1000	7	142.85715
1000	.00001	100000000
4195835	3145729	1.3338196

Division.java

```
public class Division
extends ColumnFixture {
    public float numerator;
    public float denominator;
    public float quotient() {
        return
            numerator / denominator;
    }
}
```

ActionFixture

fit.ActionFixture		
start	eg.music.Browser	
enter	library	Source/eg/music/Music.txt
check	total songs	37

RowFixture

eg.music.Display				
title	album	genre	size	date
Another Grey Morning	JT	Pop	3284199	9/7/02 11:32 PM
Ananas	Hourglass	Pop	6897450	9/7/02 11:47 PM
Copperline	New Moon Shine	Pop	5248087	9/7/02 9:52 PM
Handy Man	JT	Pop	3976956	9/7/02 11:36 PM
Sailing To Philadelphia	October Rose	Pop	6581911	9/7/02 10:45 PM

ReferenceFixture

eg.AllFiles

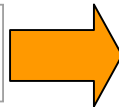
Documents/*Example.html

Documents/Music*.html

fat.ReferenceFixture			
Description	Location	Result()	Note
Parse HTML documents	parse.html	pass	
Modify HTML documents	annotation.html	pass	
Interpret tables using fixtures	fixtures.html		<i>incomplete</i>
Provide a command-line interface	cli.html		<i>incomplete</i>

Summary

fit.Summary



fit.Summary	
counts	70 right, 0 wrong, 0 ignored, 0 exceptions
input file	C:\projects\fit\imp\java\..\spec\parse.html
input update	Thu Mar 31 17:08:20 PST 2005
output file	C:\projects\fit\imp\java\output\spec\parse.html
run date	Thu Apr 07 23:25:16 PDT 2005
run elapsed time	0:00.14

Wie arbeite ich Test getrieben

- Funktionalität definieren
- Testfälle definieren
- Ggf. Fixtures programmieren
- Funktionalität implementieren
- Tests ausführen

Was leistet FIT nicht

- Last-/Stress-/Massen-/Performancetests
- Tests auf Robustheit, Stabilität
- Capture Funktionalitäten (GUI Tests)

Was ist zu beachten

- Pflege der Testfälle
- Testfälle mit Datums/Uhrzeit Bezug
- Laufzeit bei Parallel-/Echtzeitverarbeitung

Warum FIT anwenden

- Keine Werkzeugkosten (HTML Editor)
- Anpassbar an individuelle Bedürfnisse
- Für Anwender les-/bedienbar
- Agile Dokumentation
- Automatisierbare (Regressions-) Tests

FIT Implementierungen

- FitLibrary – weitere Fixtures und „Runner“
- FitNesse – Testfälle im Wiki
- WinFitRunnerLite – Testfälle in Excel
- GuiRunner, FITpro – GUIs
- ...

Wo gibt es weitere Informationen

- <http://fit.c2.com/>
- Frank Westphal: Testgetriebene Entwicklung mit JUnit und FIT, dpunkt.verlag 2005
- <http://video.google.com> bzw. youtube
Using Fit: An Open-Source Testing Framework

vielen Dank!

Fragen?