

Akzeptanztests mit FIT

Entwicklertag Karlsruhe, 17. Juni 2005

Johannes Link

Tel. (0721) 61 05 - 122

Fax. (0721) 61 05 - 140

Email: Johannes.Link@andrena.de

Internet: www.andrena.de

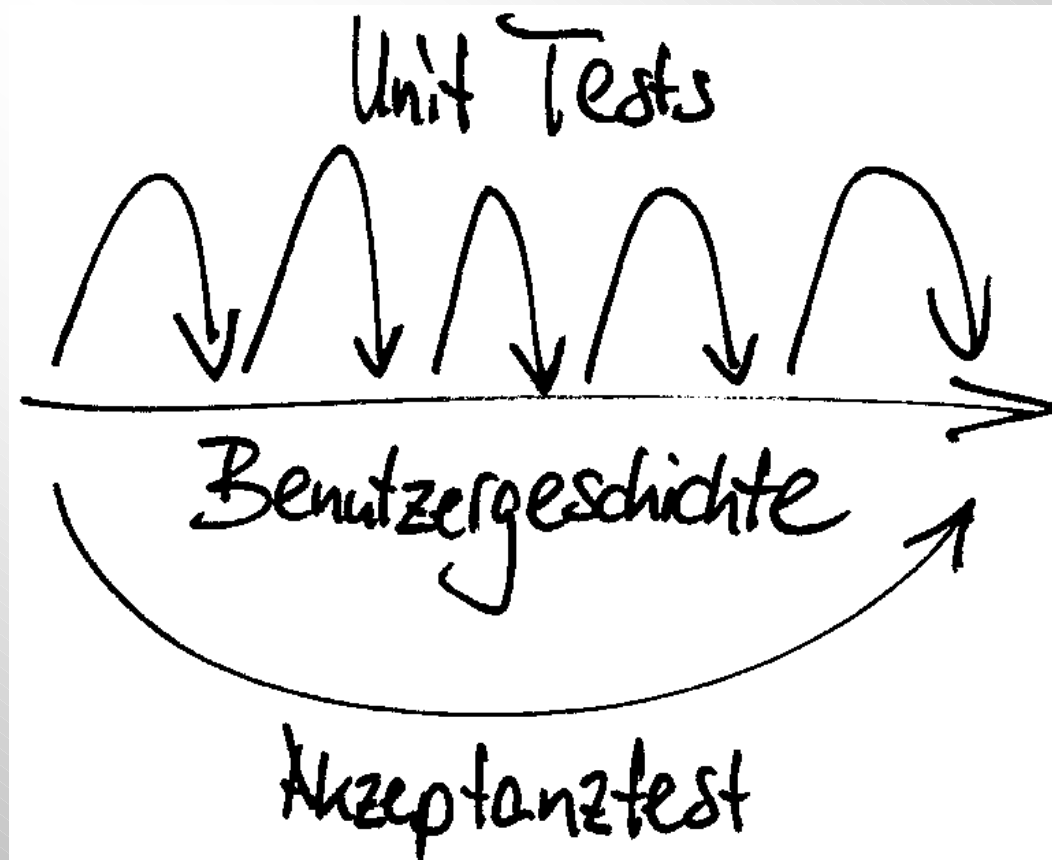
Softwarequalität

„Quality is value to some person“

„Real quality improvement always starts with knowing what your customers want.“

Gerald M. Weinberg

Zwei Arten von Tests



Systemtests / Akzeptanztests

- testen das System als Ganzes
 - klären die Anforderungen frühzeitig an konkreten Beispielen
 - geben unseren Kunden Vertrauen in die gelieferte Software
 - müssen die Sprache des Kunden sprechen
 - müssen vom Kunden verstanden, geändert und erweitert werden
 - machen den Projektfortschritt sichtbar
- => Aber: Entwickler sind für Automatisierung verantwortlich

FIT - Framework for Integrated Test

- Java-Framework zum Schreiben und Ausführen automatischer Akzeptanztests
- Testdaten werden tabellarisch erstellt (in HTML, mit Excel oder im Wiki)
- Anbindung ans System in Java
- Portierung für aktuelle Sprachen auf dem Weg
- <http://fit.c2.com>

FIT...

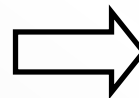
- liest HTML-Dokumente ein,
- führt die enthaltenen Testfälle aus
- reichert HTML um Testergebnis an
- gibt HTML-Dokumente wieder aus

Drei Basis-Fixture-Arten

- ColumnFixture testet Ein-/Ausgabewertemengen.
- ActionFixture spielt ein Benutzerszenario durch und ist deshalb gut für GUI-orientierte Tests geeignet.
- RowFixture prüft eine Ergebnismenge von Objekten mit ihren Attributen.

Action-Fixture

fit.ActionFixture		
start	fhso.AccountAdministrationFixture	
enter	Kundenname	Link
press	Neues Konto	
check	Kontonummer	1
enter	Einzahlung	55.00
check	Kontostand	55.00
enter	Einzahlung	45.00
check	Kontostand	100.00
enter	Kundenname	Westphal
press	Neues Konto	
check	Kontonummer	2
enter	Einzahlung	75.00
check	Kontostand	80.00



fit.ActionFixture		
start	fhso.AccountAdministrationFixture	
enter	Kundenname	Link
press	Neues Konto	
check	Kontonummer	1
enter	Einzahlung	55.00
check	Kontostand	55.00
enter	Einzahlung	45.00
check	Kontostand	100.00
enter	Kundenname	Westphal
press	Neues Konto	
check	Kontonummer	2
enter	Einzahlung	75.00
check	Kontostand	80.00 <i>expected</i>
		75.0 <i>actual</i>

Action Fixture: Code

```
public class AccountAdministrationFixture  
extends fit.Fixture {  
  
public void Kundename(String name) { ...  
}  
  
public void NeuesKonto() { ... }  
  
public String Kontonummer() {  
    return ...;  
}  
  
public void Einzahlung(double money)  
    throws AccountException { ... }  
  
public double kontostand() {  
    return ...;  
}  
}
```

Column Fixture: Testspezifikation

fhso.AccountTransferFixture			
Quellkonto	Zielkonto	Betrag	TransaktionErfolgreich()
1	2	50	true
2	1	150	false
2	1	125	true
1	1	50	false
1	5	50	error

Column Fixture: Testergebnis

fhso.AccountTransferFixture			
Quellkonto	Zielkonto	Betrag	TransaktionErfolgreich()
1	2	50	true
2	1	150	false
2	1	125	true
1	1	50	false
1	5	50	error

Column Fixture: Code

```
import fit.ColumnFixture;  
public class AccountTransferFixture  
    extends ColumnFixture {  
    public String Quellkonto;  
    public String Zielkonto;  
    public double Betrag;  
    public boolean TransaktionErfolgreich( )  
        throws AccountException {  
        ...  
        return ...;  
    }  
}
```

Row Fixture: Testspezifikation

bank.AccountListingFixture		
Kontonummer	Kontostand()	Kundenname()
1	175	Link
2	0	Westphal

Row Fixture: Testergebnis

bank.AccountListingFixture		
Kontonummer	Kontostand()	Kundenname()
1	175	Link
2	0	Westphal

Row Fixture: Code (1)

```
import fit.RowFixture;  
public class AccountListingFixture  
    extends RowFixture {  
    public Object[] query() throws  
        Exception {  
        List result = ...;  
        return result.toArray();  
    }  
    public Class getTargetClass() {  
        return AccountListingItem.class;  
    }  
}
```

Row Fixture: Code (2)

```
public class AccountListingItem {  
    public String Kontonummer;  
    public AccountListingItem(String  
        accountNumber,  
        Account account) {  
        Kontonummer = accountNumber;  
    }  
    public double Kontostand() {  
        return ...;  
    }  
    public String Kundename() {  
        return ...;  
    }  
}
```

Domänenspezifische Fixture

- Gründe:
 - Keine Fixture-Grundart passt richtig
 - Verletzung der Java-Namenskonventionen
 - Erweiterung des internen Protokolls (z.B. für Debugging, Logging)
- Abgeleitet von `fit.Fixture`
- Mögliche Hooks:
 - `doCell(..) / doCells(..)`
 - `doRow(..) / doRows(..)`
 - `doTable(..) / do Tables(..)`

Demo...

FitNesse

- Wiki-Server mit FIT-Bridge
- Tests werden im Wiki spezifiziert und von dort direkt ausgeführt
- Sub-Wikis zur weiteren Strukturierung
- <http://fitnesse.org>

FitNesse-Erweiterungen

- Einfache Definition von Testsuiten
 - SetUp/TearDown
 - SuiteSetUp/SuiteTearDown
- Paste aus Excel-Tabellen
- Unterstützung von Variablen
- Unterstützung von Makros
- Weitere Fixtures:
 - TableFixture ermöglicht den direkten Zugriff auf Tabellenzellen per Koordinaten.
 - HtmlFixture testet Elemente einer HTML-Seite.
 - RowEntryFixture dient der zeilenweise Eingabe.

Nachteile von FitNesse

- Nicht alle Kunden mögen Wikis
- Versionierung der Tests schwierig
- Abhängigkeit von laufendem Wiki-Server

Warum FIT?

- Tests sind von Entwicklung entkoppelt.
- Testsprache ist frei definierbar.
- Macht explizit, dass die Automatisierung eine Entwicklungstätigkeit ist.
- FIT und FitNesse sind kostenlos

Referenzen

- <http://frankwestphal.de/AkzeptanztestsmitFIT.html>
- Rick Mugridge, Ward Cunningham: Fit for Developing Software : Framework for Integrated Tests