

Testgetriebene Entwicklung und interaktives Testen - Symbiose statt Widerspruch?

Markus Gebhard, Ilja Preuß
disy Informationssysteme GmbH

XP-Days Germany 2004
Testgetriebene Entwicklung und Interaktives Testen - Symbiose statt Widerspruch?



Übersicht

- Motivation
- Wer wir sind...
- Testgetriebene Entwicklung
- Einführung in JDemo
- Programmier-Session
- JDemo bei disy Informationssysteme
- Weitere Features von JDemo
- Zusammenfassung

XP-Days Germany 2004
Testgetriebene Entwicklung und Interaktives Testen - Symbiose statt Widerspruch?



Motivation

Beobachtung

- Entwickler wollen Ergebnisse sehen und anfassen
- Manche Aspekte nur schwer automatisiert testbar

Bisher

- Starten der Gesamtapplikation
- main-Methoden, sleep-Anweisungen

Motivation

Neu

- Schaffung/Einsatz eines Frameworks: JDemo

Ergebnis

- Feedback fördert Vertrauen und Kommunikation
- Positiver Einfluss durch Kombination Tests/Demos

Übersicht

- Motivation
- **Wer wir sind...**
- Testgetriebene Entwicklung
- Einführung in JDemo
- Programmier-Session
- JDemo bei disy Informationssysteme
- Weitere Features von JDemo
- Zusammenfassung

Wer wir sind...

disy Informationssysteme GmbH

- Gründung: 1997
- Mitarbeiter: 22, davon 15 Entwickler
- Sitz: Karlsruhe

- "Pure Java"

Wer wir sind...

disy Informationssysteme GmbH

Bereich Information

- Berichts- und Auswertesysteme
- Datenbankintegration
- Data Warehouse Lösungen
- GIS-Lösungen

Bereich Telekommunikation

- Telefon- und Webkonferenzen
- Rückrufsysteme
- Systeme für Virtuelle Call Center

XP-Days Germany 2004

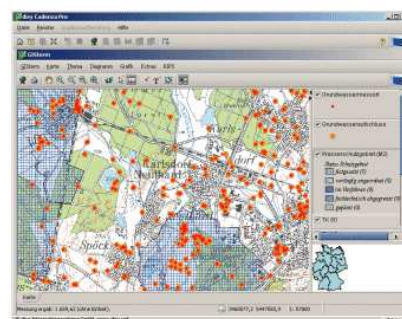
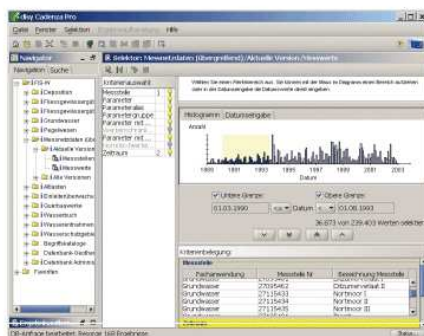
Testgetriebene Entwicklung und Interaktives Testen - Symbiose statt Widerspruch?



Wer wir sind...

disy Cadenza

- Integrationsplattform für Berichts- und Auswertesysteme mit Raumbezug
- Alle Vorgänge der Informationsrecherche und -analyse in einem Werkzeug
- Umweltmonitoring, E-Government, Geomarketing...



XP-Days Germany 2004

Testgetriebene Entwicklung und Interaktives Testen - Symbiose statt Widerspruch?



disy Cadenza - Projekt

- >8 Jahre
- ca. 0,5 MLOC
- >75 Module
- 4 Kern-Entwickler

- umfangreiche Swing-GUI
- vielseitige Benutzer-Interaktionen
- seit einiger Zeit auch Web-Interface

XP @ disy Informationssysteme

- + Refactoring
- Collective Ownership
- Continuous Integration

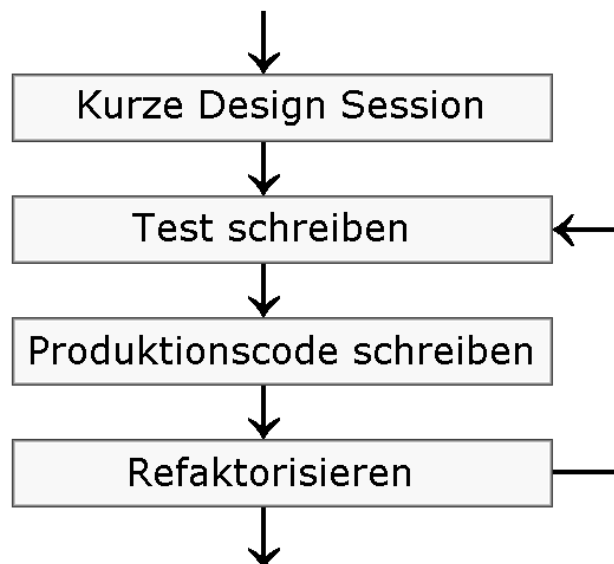
- Customer (Acceptance) Tests
- Metaphor
- Planning Game

Übersicht

- Motivation
- Wer wir sind...
- **Testgetriebene Entwicklung**
- Einführung in JDemo
- Programmier-Session
- JDemo bei disy Informationssysteme
- Weitere Features von JDemo
- Zusammenfassung

Testgetriebene Entwicklung

Definition Testgetriebene Entwicklung



Übersicht

- Motivation
- Wer wir sind...
- Testgetriebene Entwicklung
- **Einführung in JDemo**
- Programmier-Session
- JDemo bei disy Informationssysteme
- Weitere Features von JDemo
- Zusammenfassung

Einführung in JDemo

Beispiel: Suchmaschine für Bücher

Author	Title	Year

Entnommen aus:

The Test/Code Cycle in XP

William C. Wake

<http://www.xp123.com/xplor/xp0001/index.shtml>

"Interaktiver" Unit-Test

```
public void testRelativePosition() {
    SearchPanel panel = new SearchPanel();

    JFrame display = new JFrame("test");
    display.getContentPane().add(panel);
    display.setSize(500, 500);
    display.setVisible(true);

    //try {Thread.sleep(3000);} catch (Exception ex) {}

    assertTrue("label left-of query", panel.searchLabel
        .getLocationOnScreen().x < panel.queryField
        ...
    }
}
```

*"You can un-comment the sleep()
if you want to see it on-screen."*

"Interaktiver" Unit-Test

Bewertung

- Menschliche Anforderung: Sehen und Anfassen
- `sleep()` ist code smell
- Erfüllt Zweck nur unzureichend

Forderung

- "Geordneter Prozess" für interaktive Entwicklertests
- Reduktion auf das Wesentliche
- Unterstützung durch Werkzeuge und Bibliotheken

"Interaktiver" Unit-Test

```
public class SearchGUITest extends TestCase {

    public void testRelativePosition() {
        SearchPanel panel = new SearchPanel();

        JFrame display = new JFrame("test");
        display.getContentPane().add(panel);
        display.setSize(500, 500);
        display.setVisible(true);

        try {Thread.sleep(3000);} catch (Exception ex) {}

        assertTrue("label left-of query", panel.searchLabel
            .getLocationOnScreen().x < panel.queryField
            .getLocationOnScreen().x);
        ...
    }
}
```

"Interaktiver" Unit-Test - Refactored

```
public class SearchPanelDemo extends SwingDemoCase {

    public void demoSearchGui() {
        SearchPanel sp = new SearchPanel();
        show(sp);
    }
}
```

- Kompakt
- Einfach
- Analog zu JUnit

JUnit vs. JDemo

Validieren ↔ Demonstrieren

extends TestCase ↔ extends ...DemoCase

test... () ↔ demo... ()

assert... () ↔ show... ()

setUp ()

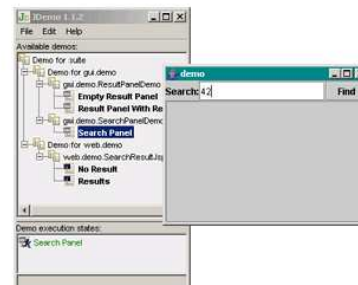
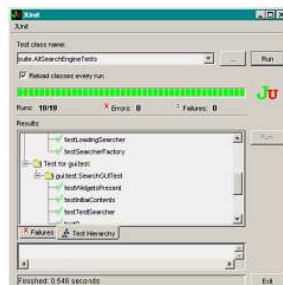
tearDown ()

JUnit vs. JDemo

Automat. Ausführen ↔ Manuelles Ausführen

TestRunner ↔ DemoRunner

TestSuite ↔ DemoSuite



Common Public License

Übersicht

- Motivation
- Wer wir sind...
- Testgetriebene Entwicklung
- Einführung in JDemo
- **Programmier-Session**
- JDemo bei disy Informationssysteme
- Weitere Features von JDemo
- Zusammenfassung

Programmier-Session

Arbeiten mit Tests und Demos

*Interaktives Arbeiten mit JDemo und JUnit.
Erweiterung des Suchmaschinen-Beispiels.*

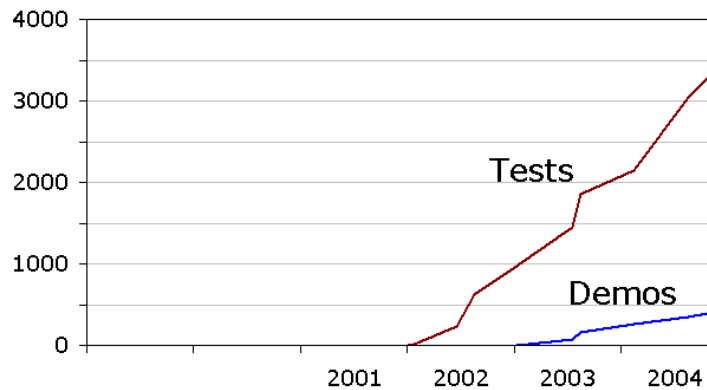
Erfahrungen

- Demos
 - decken schnell Designschwächen auf
 - ergänzen Tests
 - motivieren Tests
 - motivieren neue Features
- Dummy-Objekte lassen sich für Tests und Demos verwenden
- "Demogetriebene" Entwicklung

Übersicht

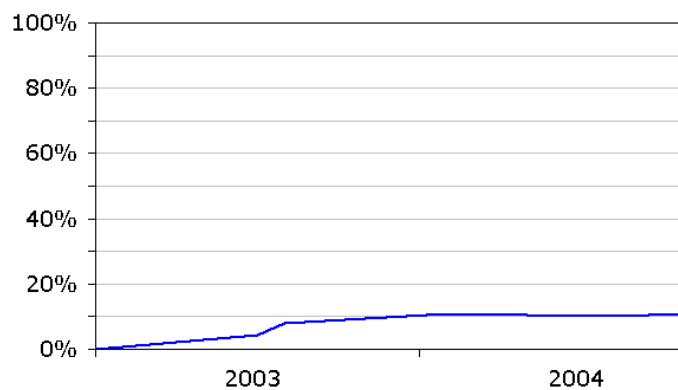
- Motivation
- Wer wir sind...
- Testgetriebene Entwicklung
- Einführung in JDemo
- Programmier-Session
- **JDemo bei disy Informationssysteme**
- Weitere Features von JDemo
- Zusammenfassung

Anzahl Tests/Demos



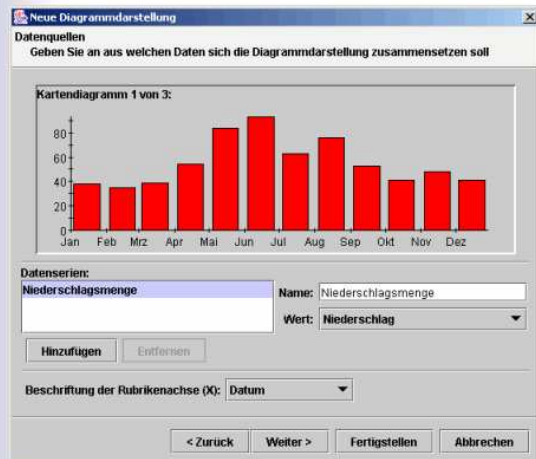
- 1996 - 2002 ungetesteter Code
- Ab 2002 Unit Tests
- Anfang 2003 erste Demos, seit Mitte 2003 regelmäßiger Einsatz

Anteil der Demos



- 1 Demo auf ca. 8 Tests
- Tests feingranularer und auch für "Innereien"

Anwendungsfall: Diagrammassistent

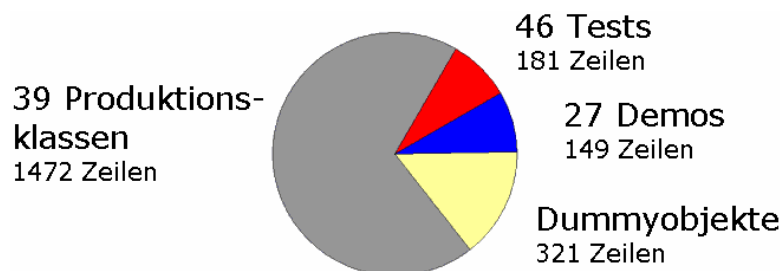


Besonderheiten:

- Verschiedene Ausgangsdaten
- Sehr GUI-lastig, 4 Seiten
- Implementierung gegen vorhandenen Code

Anwendungsfall: Diagrammassistent

Ergebnis:



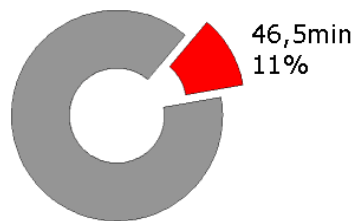
- Großer Anteil Dummy-Objekte für Demos **und** Tests
- Verhältnismäßig viele Demos
- Kein Mehraufwand - stattdessen: Synergien

Anwendungsfall: Diagrammassistent

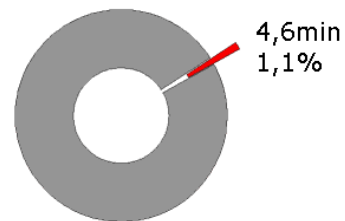
Startup-Zeiten

Applikation ca. 45s, Demos ca. 4s

Anteil am Arbeitstag:



62 mal Starten der Applikation



61 mal Starten der Demos,
1 mal Starten der Applikation

Allgemeine Erfahrungen

- Unterstützt den "Flow" durch reduzierte Leerlaufzeit
- Hilfe bei Analyse und Entkopplung von Legacy Code
- Entwicklung wiederverwendbarer Komponenten
- Dokumentation der API
- Überblick über vorhandene Komponenten
- Guter Wiedereinstieg für Bugfixes/Erweiterungen
- Spielerischer Umgang mit GUI fördert Usability

Übersicht

- Motivation
- Wer wir sind...
- Testgetriebene Entwicklung
- Einführung in JDemo
- Programmier-Session
- JDemo bei disy Informationssysteme
- **Weitere Features von JDemo**
- Zusammenfassung

Weitere Features von JDemo

Demo-Tests

Beobachtung

- Unbenutzter Code zerbricht
- Jeder Test wird täglich mehrmals ausgeführt
- Demos werden nur bei Bedarf einzeln ausgeführt

Lösung

- DemoSuites zusammen mit Unit-Tests ausführen:

```
testSuite.addTest(  
    Demo2TestConverter.createTest(demoSuite));
```


Weitere Features

- Demos für
 - AWT & SWT
 - Dateien (HTML, Office etc.)
 - Servlets/JSPs (beta)

- Unterstützung für Swing Look'n'Feels
- Eclipse-Integration
- Automatisierte Screenshots (Apache Ant)
- Java 5 Annotations

Übersicht

- Motivation
- Wer wir sind...
- Testgetriebene Entwicklung
- Einführung in JDemo
- Programmier-Session
- JDemo bei disy Informationssysteme
- Weitere Features von JDemo
- **Zusammenfassung**

Zusammenfassung

- Ausprobieren ist ein Grundbedürfnis des Entwicklers
- JDemo als Framework analog zu JUnit
- Demos
 - erlauben koordiniertes interaktives Testen
 - ergänzen Testgetriebene Entwicklung
 - erzeugen Synergien und steigern die Produktivität

Vorsicht!

- Demos
 - verleiten dazu, Tests zu vernachlässigen
 - müssen gepflegt werden

Vielen Dank!

Links

- JDemo www.jdemo.de
- disy Informationssysteme GmbH www.disy.net
- Artikel: Interactive Testing Refactored
<http://www.onjava.com/pub/a/onjava/2004/09/08/jdemo.html>
- Simulators
<http://today.java.net/pub/a/today/2004/06/23/simulators.html>

Kontakt

- Ilja Preuß preuss@disy.net
- Markus Gebhard gebhard@disy.net