

AGILE ENTWICKLUNG VON WEB-ANWENDUNGEN MIT HTTPUNIT UND TAPESTRY

Christian Noack

Juli 2003





ZIELE

- Zusammenhang Prozess und Technologie beleuchten
- Beispielhafte Technologiezusammenstellung
- Erfahrungsaustausch
- Überblick über die Konzepte von Tapestry
- Verständnis von Web-Testing



AGENDA

- ✘ Web-Anwendungsentwicklung und Agiler Prozess
- ✘ Technologieauswahl für Agile Entwicklung
 - HttpUnit und Umfeld
 - Tapestry und Umfeld
- ✘ Diskussion



ENTWICKLUNG VON WEB-ANWENDUNGEN

Probleme, die bei klassischer Anwendungsentwicklung nicht bekannt sind:

- Server-seitige Ausführung, Kontrollfluss über den Server
- Kaum Kontrolle über Client bzgl. Layout
- Kein durchgängiger Anwendungsfluss
- Transaktionskonzept
- Session-Handling



WEB-ANWENDUNG VS. XP

- Vier Werte
 - Simplicity
 - Feedback
 - Courage
 - Communication
- Zwölf Kernpraktiken

Unterstützung je nach Technologie (Servlet, JSP, Struts, Tapestry, WebObjects) in unterschiedlichem Maße gegeben.



SIMPLICITY

- Java-Web-Anwendungen
- Hoher Aufwand zum Aufbau der Infrastruktur zu Beginn eines Projekts → Behinderung der iterativen Entwicklung



FEEDBACK

- JSP-basierte Anwendungen → schlechte Testbarkeit durch Unit Tests
- Hohe Turn Around-Zeiten durch Auslieferung an Servlet Container → Erschwerung regelmäßigen Testens und kontinuierlicher Integration
- Verteilung des Kontrollflusses auf Client und Server → JUnit reicht nicht aus



COURAGE

- Fehlende Unit Tests verhindern mutiges Vorgehen
- Refactoring bei Web-Anwendungen schwierig



COMMUNICATION

- ggf. neue Rolle im Team: Web-Designer
- Keine weiteren Web-Anwendungsspezifischen Kommunikationsprobleme



XP KERNPRAKTIKEN

Keine Web-Anwendungsspezifischen Probleme bei:

- Planning Game
- Metapher
- Pair Programming
- Kunde vor Ort
- 40-Stunden-Woche
- Code-Standards



EINFACHES DESIGN

XP	Web-Anwendung
Nur das Benötigte ist implementiert	Viel technischer Overhead, statt Konzentration auf eigentliche Inhalte
Gute Testbarkeit	JUnit nicht ausreichend, Aufwendige Testumgebung
Gute Verständlichkeit	Bei JSP: Code in HTML, Kein Separation of Concerns, Kein MVC; Keine Abstraktion
Gute Anpassbarkeit und Erweiterbarkeit	Bedingt, da in JSP Code in HTML enthalten ist



TEST FIRST

- Testumgebung kann aufwendig sein
- Servlet Container
- ggf. EJB-Container
- ggf. Datenbank
- HttpRequests müssen simuliert werden
- Generiert HTML-Code muss ausgewertet werden



REFACTORING

XP	Web-Anwendung
Anpassung bzw. Umstrukturierung des Codes	Schwierig, wenn Code auf JSP und Klassen verteilt ist
Funktionalität bleibt unverändert	Erfordert Tests
Interne Struktur/Design wird verbessert	Technisch manchmal nicht möglich



COLLECTIVE CODEOWNERSHIP

Kompliziertes Design und unübersichtlicher Code erschweren gemeinsame Code-Eigentümerschaft



KONTINUIERLICHE INTEGRATION

- Aufwendige Testumgebung
- Aufwendiges Deployment

⇒ Schwierige Integration



KLEINE RELEASES

Setzt kontinuierliche Integration und umfangreiches Testing voraus



MASSNAHMEN - SIMPLICITY

Auswahl der Technologie:

- Trennung gemäß MVC
- Einfache Benutzung, durchgängiges Konzept
- Wenig technischer Overhead
- Wenig Infrastruktur-Aufwand
- Gute Verständlichkeit
- Schnelle Erfolge
- Welche Technologien sind notwendig (z.B. EJB?)



MASSNAHMEN - FEEDBACK

Festlegung der Testmethodik

- Separation of Concerns
- Model → JUnit
- Datenbank → JUnit/DBUnit
- View/Control → JUnit/HttpUnit

Auswahl der Technologie

- EJB notwendig?
- Schneller Servlet Container
- ggf. integrierte Testumgebung



MASSNAHMEN - COURAGE

Erleichterung und Transparenz von Veränderungen

- Durchsetzung des Testverfahrens
- Wiederholbare automatisierte Tests
- Einfaches Design

Auswahl der Technologie

- Unterstützung von Refactoring
- Einfaches Design



AGENDA

- ✓ Web-Anwendungsentwicklung und Agiler Prozess
- ✗ Technologieauswahl für Agile Entwicklung
 - HttpUnit und Umfeld
 - Tapestry und Umfeld
- ✗ Diskussion



TECHNOLOGIE FÜR AGILE ENTWICKLUNG

- JDK \geq 1.3
- JUnit
- HttpUnit
- Jetty
- Log4j
- Tapestry
- Eclipse
- Spindle



HTTPUNIT

- HttpUnit ist Testframework und bietet:
- Erzeugen von Requests

```
request = new GetMethodWebRequest("http://localhost:8090");
```

- Abschicken von Requests

```
wc = new WebConversation();  
response = wc.getResponse(request);
```

- Zugriff auf Response u. Auswertung

```
Assert.assertEquals("Wrong Page", "Login", response.getTitle());
```

- Zugriff auf Elemente einer Seite

```
WebForm form = response.getForms()[0];  
form.setParameter("inputUserId", userId);  
WebRequest request = form.getRequest();
```

HTTPUNIT UND AGILE ENTWICKLUNG

- Anwendung läuft im Servlet-Container
- Benötigt Testumgebung
- Start der Testumgebung ist aufwendig
⇒ Test Resource aufsetzen und automatisiert starten
- Open Source

HttpUnit ermöglicht automatisiertes Testen von View und Control ⇒ Unterstützt Agilen Prozess



JETTY UND AGILE ENTWICKLUNG

- Kleiner Servlet Container
- Open Source
- Einfache Installation und Benutzung
- Geringe Turn Around-Zeiten

Jetty ist klein und einfach zu benutzen, ermöglicht schnelle Tests

⇒ Unterstützt Agilen Prozess



TECHNOLOGIE - SERVLETS

- Funktionsweise vergleichbar zu CGI
- Komponente des Servers / Servlet Container
- Servlet in Java geschrieben
- Laufen im Prozessraum des Servers ab
- Server startet und wartet auf Anfrage
- Zielt Request auf Servlet, wird es geladen (einmal)
- Während einmaliger Initialisierung werden notwendige Ressourcen aufgebaut
- Servlet verarbeitet get oder post requests
- HTML-Ausgabe wird im Servlet generiert

Reine Servlet-Programmierung ist umständlich, keine Trennung gemäß MVC \Rightarrow Erschwert Agile Entwicklung



TECHNOLOGIE - JSP

- Java Server Pages
- Einbettung von Java-Code in HTML-Code
- JSP-Servlet-Engine
 - liest Server-seitig Request
 - Parsing von JSP-Files
 - Generierung, Compilierung und Instanziierung des Servlets
- Servlet generiert HTML-Ausgabe anhand von JSP-Files

Aufweichung von MVC, Schwieriges Debugging, Schwierige Entwicklung \Rightarrow Erschwert Agile Entwicklung



TECHNOLOGIE- KOMPONENTEN

- Webobjects / Tapestry
- Basiert auf Servlets
- Unabhängige Definition von Struktur und Layout und Inhalten
- Server-seitige Ausführung mittels Komponenten
- Höheres Abstraktionsniveau
- Keine Generierung von Servlets erforderlich



TAPESTRY

- Java-Framework zur Entwicklung von Web-Anwendungen
- Howard Ship
- Open Source
- Alternative zu JSP
- Ausgereiftes Komponenten-Modell

TAPESTRY UND AGILE ENTWICKLUNG

- Hohe Wiederverwendbarkeit innerhalb von Projekten und Projekt-übergreifend
- Entwickler werden vom Schreiben langweiligen Codes entlastet
- Anwendungskomplexität kann gut skalieren
- Entwicklung robuster Anwendungen möglich
- Vermeidung von Code zur Entwicklung von Web-Anwendungen
- Fast sämtlicher Code behandelt Anwendungsfunktionalität
- Nur wenig Verschmutzung
- Keine Beschäftigung mit Servlets, URL-Parsing und -Zusammenbau, Session-Handling

• • • TAPESTRY UND AGILE ENTWICKLUNG

Konzentration auf das Wesentliche, Saubere Trennung,
Gutes Debugging \Rightarrow erleichtert Agile Entwicklung

ECLIPSE UND AGILE ENTWICKLUNG

- Frei verfügbar
 - Hohe Integration (JUnit, CVS ...)
 - Gutes Debugging
 - Plugins
 - Unterstützt Collective Codeownership
 - Unterstützt Refactoring
 - Unterstützt Testing
 - Konzentration auf Entwicklung
- ⇒ unterstützt Agile Entwicklung

SPINDLE UND AGILE ENTWICKLUNG

- Elcipse-Plugin
 - Wizzards und Views für Tapestry
 - Beschleunigt Entwicklung
 - Verringert Fehler
 - Refactoring-Unterstützung noch schwach
 - Erhöht Kommunikation
- ⇒ unterstützt Agile Entwicklung



TAPESTRY- BASISKLASSEN

- Engine
 - Application Engine verfolgt Aktionen eines Clients innerhalb der Anwendung
 - Wenn Client sich mit Anwendung verbindet, wird Application Engine instanziiert.
- ApplicationServlet
 - Unterklasse von ApplicationServlet
 - Brücke zwischen Servlet Container und Engine
 - Intanziiert oder Lokalisiert Application Engine
- Visit-Objekt
 - Verstecktes Session-Handling
 - Verbindungsglied zwischen den Seiten
 - Von Engine instanziiert und gelöscht

ÜBERBLICK ÜBER PROJEKT-DATEIEN

- java
- html, css
- page (Komponentenbeschreibung einer Seite)
- jwc (java web component, Beschreibung einer Komponente)
- application
- jetty.xml
- web.xml



KOMPONENTENBESCHREIBUNG

- Zu jeder Html-Seite gehört page-Datei mit Definition für jede verwendete Komponente
- Komponente hat Typ (z.B. Insert) und häufig Bindings, die definierte Parameter deklarieren
- Binding hat Namen und eine Expression
- Anbindung an Spezifikationsklasse Accessors/Methoden
- Binding kann dynamisch Inhalte holen und setzten
- Delegation möglich

```
(visit.userId ⇒ this.getVisit().getUserId())
```
- Binding kann Aktionen bei Spezifikationsklasse auslösen
- Zu jedem Komponententypen gehört jwc-Datei mit Komponentenbeschreibungen



BENUTZUNG VON KOMPONENTEN

Komponenten-Typen

- Insert
- TextField
- Page
- Service
- ForEach
- Action
- ValidField
- \$remove\$
- ...

Arten des Binding

- dynamic
- static
- field
- inherited

Bindings

- value
- page
- service
- format
- element
- ...

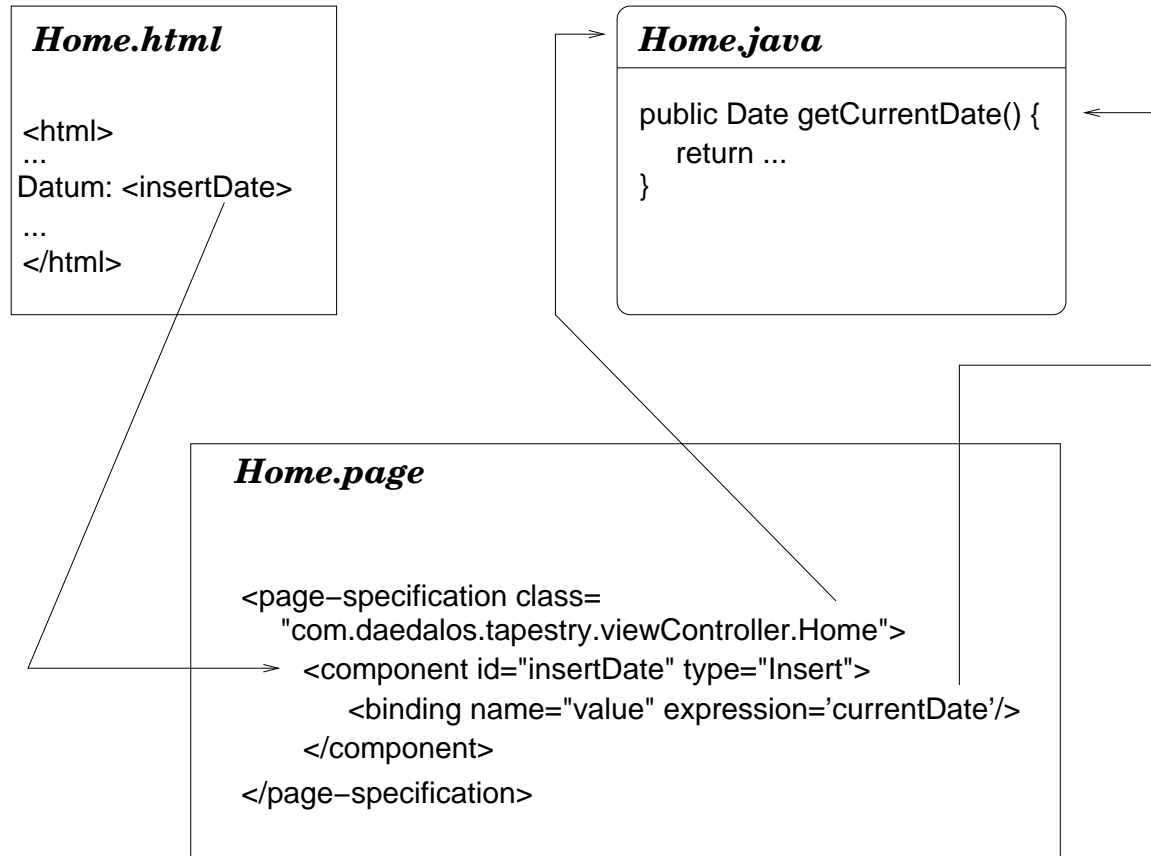


ZUSAMMENHÄNGE

- ID einer Komponenten ist frei wählbar;
z.B. `jwtid="insertName"`
- Wird in page definiert und in html referenziert
- Seitenname ist frei wählbar; z.B. *Home*
- Festlegung in application, Referenzierung in page (Page-Binding) und in Java-Klasse (`cycle.setPage("Home")`)
- Zusammengehörende html- und page-Dateien sollten den gleichen Namen tragen (z.B. *Home.page*, *Home.html*)
- Zuordnung zum Seitennamen in der application
- Name einer Spezifikationsklasse (z.B. *Home.java*) frei wählbar
- Wird in der application einer Seite zugeordnet



ZUSAMMENHÄNGE





AGENDA

- ✓ Web-Anwendungsentwicklung und Agiler Prozess
- ✓ Technologieauswahl für Agile Entwicklung
 - HttpUnit und Umfeld
 - Tapestry und Umfeld
- ✗ Diskussion



DISKUSSION

- Unterstützt die vorgestellte Technologie Agile Entwicklung?
- Gibt es Alternativen?



AGENDA

- ✓ Web-Anwendungsentwicklung und Agiler Prozess
- ✓ Technologieauswahl für Agile Entwicklung
 - HttpUnit und Umfeld
 - Tapestry und Umfeld
- ✓ Diskussion



VIELEN DANK

- <http://www.daedalos.com>
- <http://tapestry.sourceforge.net>
- <http://spindle.sourceforge.net>
- <http://www.eclipse.org>
- <http://jetty.mortbay.org>
- <http://httpunit.sourceforge.net>
- <mailto:Christian.Noack@daedalos.com>