Bugkilla

funktionale Akzeptanztests für agile J2EE Entwicklung

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
http://www.oio.de
info@oio.de

Dirk M. Sohn
<sohn@oio.de>

Christian Dedek
<dedek@oio.de>

Agenda

- Motivation
- Projektentscheidung
- Fachliche Idee
- Demo 1
- Grobarchitektur
- Integrationsideen
- Vollständiger Funktionaler Test
- Demo 2
- Fazit

XP - the central principles

Orientation in Objects

- Fast and detailled feedback
  - test driven design with acceptance tests — 20%
  - Customer On-site — 30%
  - pair programming
- Common understanding
  - planning game
  - simple design
  - System metaphor — 80%
  - collective code ownership
  - coding conventions — 80%
- Continuous process:
  - continuous integration
  - refactoring
  - frequent and small releases
- Developer Welfare
  - a week has 40 hours — 95%

Waterfall model of system development

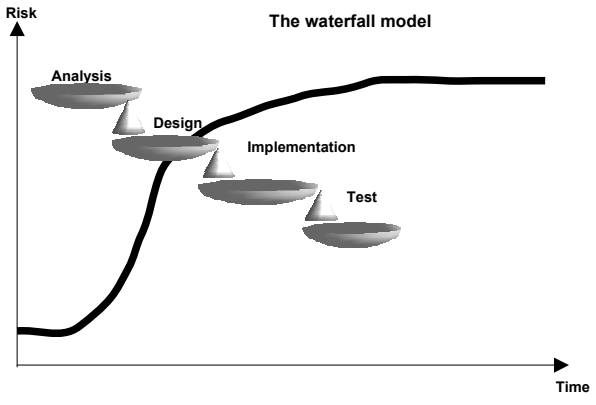
Orientation in Objects

four phases

each separated

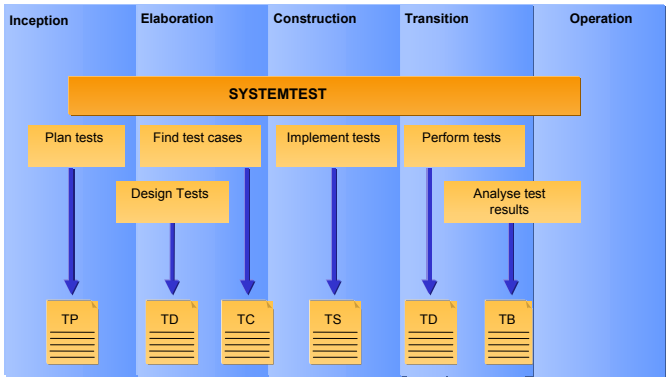
risk cumulated

**(Tom Gilb:
„if you do not
actively attack
the risks in your project,
they will actively
attack you.“)**



Classical System Test

Orientation in Objects



TP - Test plan

TD - Test design specification

TC - Test case specification

TS - Test scripts

TE - Test execution protocol

TR - Test report

Iterative Development

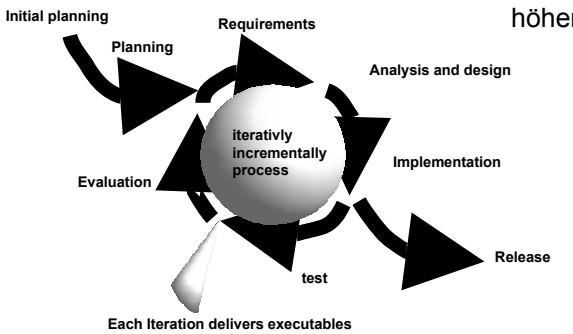
Orientation in Objects

Softwareentwicklung ist

weder Massenproduktion

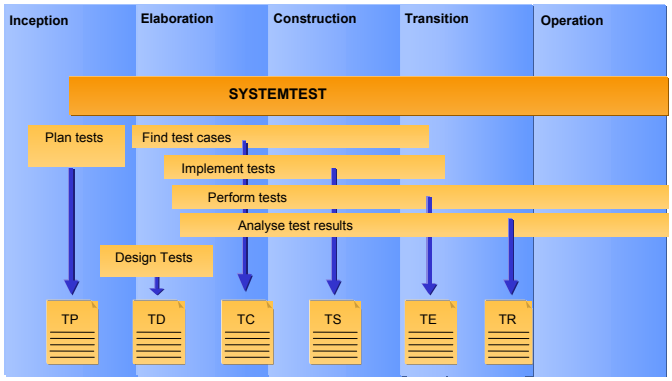
noch Ingenieurskunst

Risikominimierung durch viele Iterationen führt zu höheren Testaufwänden.



System Test - Bugkilla style

Orientation in Objects



TP - Test plan

TD - Test design specification

TC - Test case specification

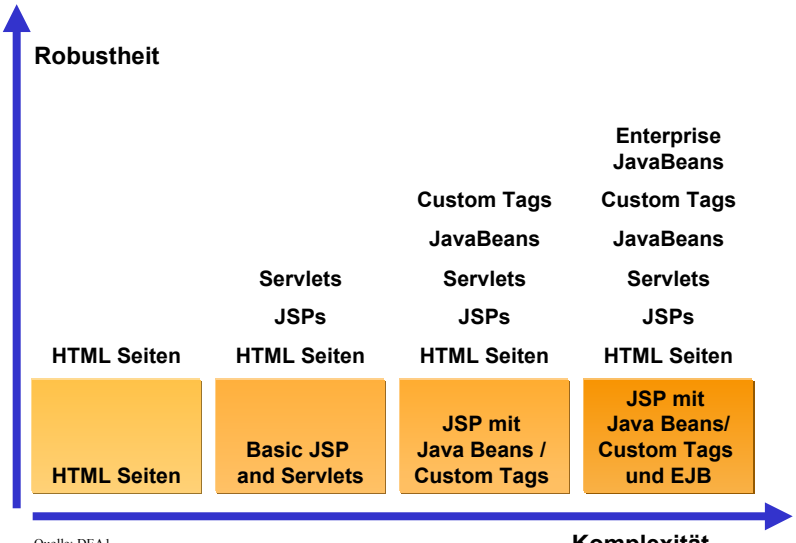
TS - Test scripts

TE - Test execution protocol

TR - Test report

Exkurs: Application Design Spektrum

Orientation in Objects



Quelle; DEA1

J2EE-Testwerkzeuge (Auswahl)

Orientation in Objects

- Im Server
  - Komponententests
    - **StrutsTestCase**
  - Integrationstests
    - **Cactus**
- Ausserhalb des Servers
  - Systemtest
    - **httpunit**
  - Akzeptanztests
    - **htmlunit**

Agenda

Orientation in Objects

- Motivation
- Projektentscheidung
- Fachliche Idee
- Demo 1
- Grobarchitektur
- Integrationsideen
- Vollständiger Funktionaler Test
- Demo 2
- Fazit

Vollständiger Funktionaler Systemtest

Orientation in Objects

- Schlüssel zur Kostensenkung ist Automatisierung.
- Testfallspezifikation enthält Ein- und Ausgaben des Systems innerhalb des Testfalls
- Hierzu gehört für einen vollständigen funktionalen Test:
  - Direkte Eingaben des Anwenders durch Interaktion mit Browser
  - Direkte Ausgabe als HTTP-Response-Streams (Dokumente)
  - Anfangszustände der Geschäftsobjekte sind indirekte Eingaben
  - Endzustände der Geschäftsobjekte sind indirekte Ausgaben

Automatische Erstellung

Orientation in Objects

- Aufgaben für Automatisierung:
  - Abbildung der direkten Ein- und Ausgaben des Users auf ein Aufzeichnungsformat (leicht bei Browser-Clients)
  - Übersetzung der Zustandsübergänge der fachlichen Geschäftsobjekte auf Zustandsübergänge von technischen Objekten.
- Lösung für die direkten Ein- und Ausgaben
  - Aufzeichnung der HTTP-Kommunikation. Eine Interaktion löst eventuell mehrere Requests aus, die jeweils von einer Response beantwortet werden.
  - Alternativen: Schaltflächenaufzeichnung, TCP-Pakete ohne Sitzung
- Lösung für die Zustandübergänge
  - ToDo

Automatische Auswertung

Orientation in Objects

- Auswertung der direkten Ausgaben (Response-Dokumente)
  - Vergleich der Antworten der Testfallspezifikation mit denen der Testdurchführung - einfach nur HTML-Daten auswerten.
- Auswertung der Zustandsübergänge im Container
  - Übersetzung auf technische Objekte bei Einsatz von Frameworks deutlich präzisierbar.
  - Mechanismus zur Abfrage der Zustände fachlich relevanter Objekte
  - Kontrollflußsteuerung der Zustandsabfrage im Container nötig

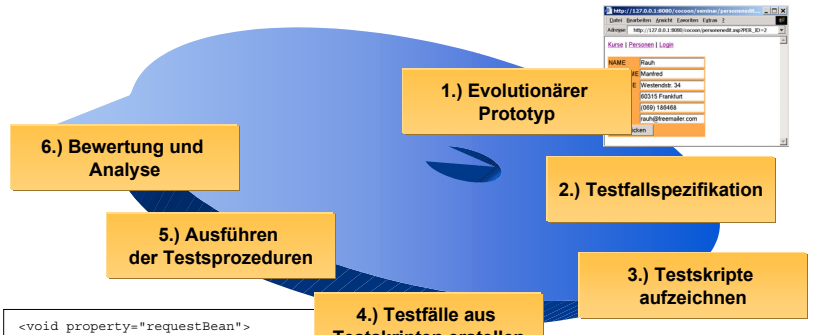
Agenda

Orientation in Objects

- Motivation
- Projektentscheidung
- Fachliche Idee
- Demo 1
- Grobarchitektur
- Integrationsideen
- Vollständiger Funktionaler Test
- Demo 2
- Fazit

Fachliche Idee von Bugkilla

Orientation in Objects




```
<void property="requestBean">
   <object
class="de.oio.bugkilla.HttpRequest">
    ...
    <void property="method">
     <string>GET</string>
    </void>
    <void property="name">
     <string>RB-Tue Aug 13 20:09:55 CEST
2002</string>
    </void>
    <void property="protocol">
     <string>HTTP/1.1</string>
```


OIO Tooling philosophy

Orientation in Objects

1. Define clearly what is a tools job.
2. There is always a tool doing the job right. A project team has to find out, how to use it right.
3. If no tool doing the job right could be found by the team, the extension of a tool found during the evaluation is the solution.
4. Integration of tools doing their jobs right is the key to lasting success.

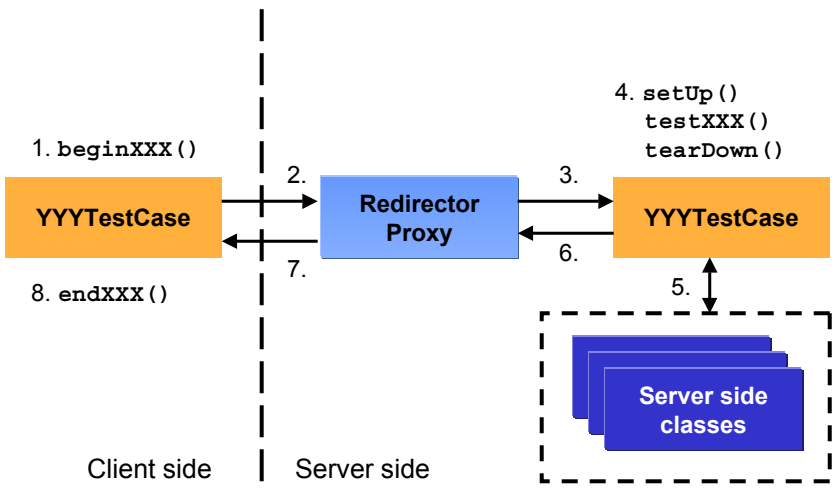
Erweiterung eines der gefundenen Tools?

Orientation in Objects

- Gefundene Kandidaten
  - Canoo (open source)
  - httpunit (open source)
  - Exacum (kommerziell)

Warum nicht Cactus ?

Orientation in Objects



Eigenentwicklung nach Punkt 3 Philosophy

Orientation in Objects

- Kommerzielle oder freie Lizenz, Open oder closed source?
  - Budget für Entwicklung und Marketing
  - Dienstleister und Produktfirma
  - Strategische Ausrichtung zum Support Center für OS-Server
  - also Open Source mit freier Lizenz

Die Vorgeschichte eines OS-Werkzeugs

Orientation in Objects

- Idee
  - *,,Every good work of software starts by scratching a developer's personal itch.``* - Eric Raymond
- 1. Version
  - *,,It's fairly clear that one cannot code from the ground up in bazaar style. One can test, debug and improve in bazaar style, but it would be very hard to originate a project in bazaar mode.``* - Eric Raymond

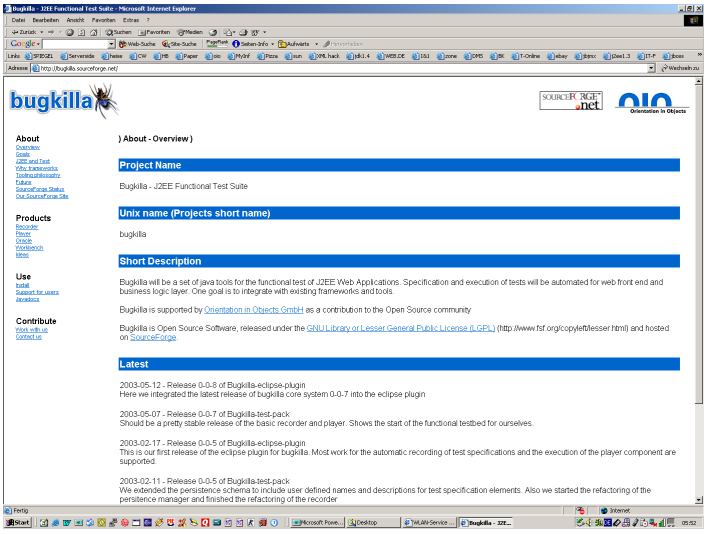
Der Start der Basars

Orientation in Objects

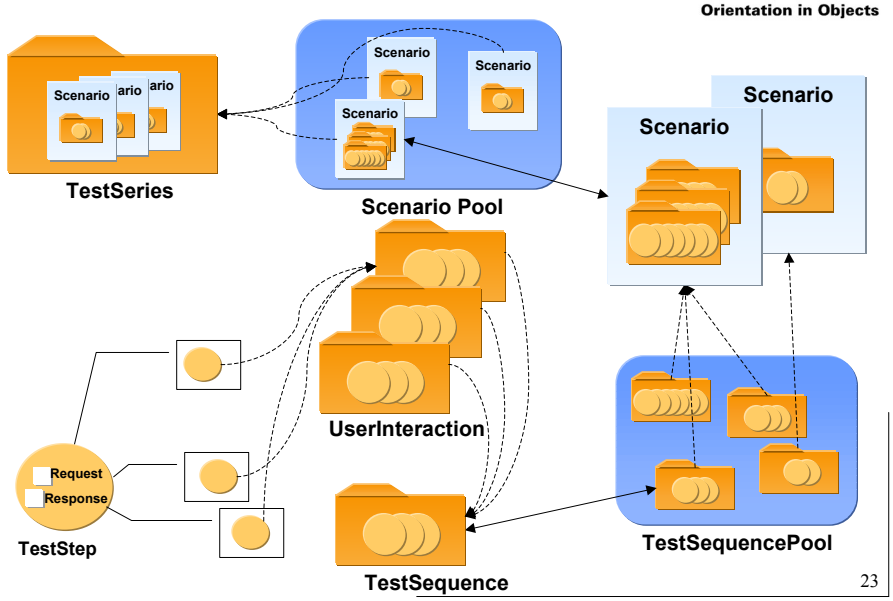
- Allgemeiner Funktionsumfang - muß für viele interessant sein
- Am besten stark modularisiert - bessere Arbeitsteilung
- Motivation
  - *,,There's one lesson that's really obvious: You cannot motivate the best people with money. Money is just the way to keep score. The best people in any field are motivated by passion. This becomes more true the higher the skill level gets.``* - Eric Raymond
- Infrastruktur
  - *,,The real lesson to be learned from open source communities are the techniques of networked collaboration that they've pioneered.``* - Tim O'Reilly

Der Basar der Basare

Orientation in Objects



Objektmodell



Glossar

- TestStep
  - einzelner Testschritt, bestehend aus einem Request- und einem Response-Objekt
- TestSequence
  - ordnet TestSteps zu einem zusammengehörigen Ablauf
  - beinhaltet Informationen zum Environment (Aufzeichnungsdatum, Servername)
- TestSequencePool
  - beinhaltet sämtliche aufgezeichnete Sequenzen in Form von TestSequence-Files (zusammenhanglos)
- Scenario
  - ordnet mehrerer Sequenzen zu einem fertigen Testablauf, läuft in einer HTTP-Session ab, ist fachlich eindeutig zu beschreiben

Glossar II

Orientation in Objects

- SzenarioPool
  - beinhaltet sämtliche Szenarien in Form von Szenarien-Files (zusammenhanglos)
- TestSeries
  - Anordnung mehrerer Szenarien zum einer Testserie. Diese Szenarien haben untereinander keine Abhängigkeiten und werden der Reihe nach abgespielt.(eventuell bestehen Abhängigkeiten aber dann muss ein reset-des Servers erfolgen)
- Testspezifikation
  - Abstraktion von Teststeps, Testsequenzen, Szenarien, Testserien

Agenda

Orientation in Objects

- Motivation
- Projektentscheidung
- Fachliche Idee
- Demo 1
- Grobarchitektur
- Integrationsideen
- Vollständiger Funktionaler Test
- Demo 2
- Fazit

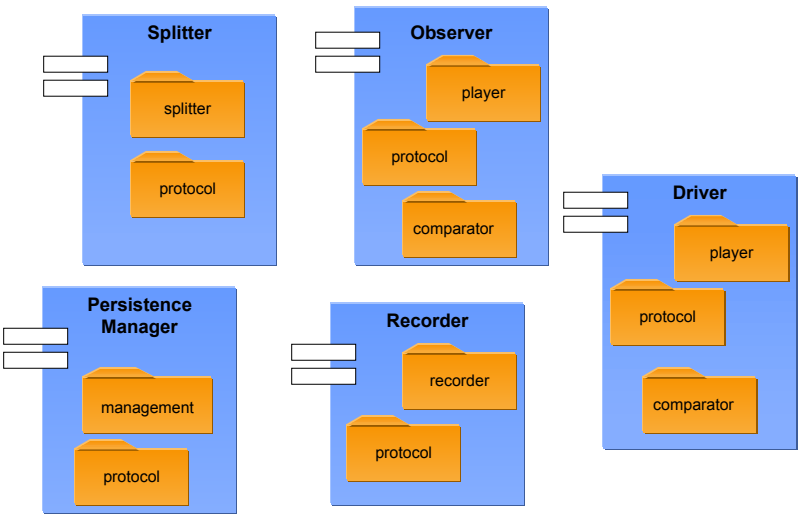
Agenda

Orientation in Objects

- Motivation
- Projektentscheidung
- Fachliche Idee
- Demo 1
- Grobarchitektur
- Integrationsideen
- Vollständiger Funktionaler Test
- Demo 2
- Fazit

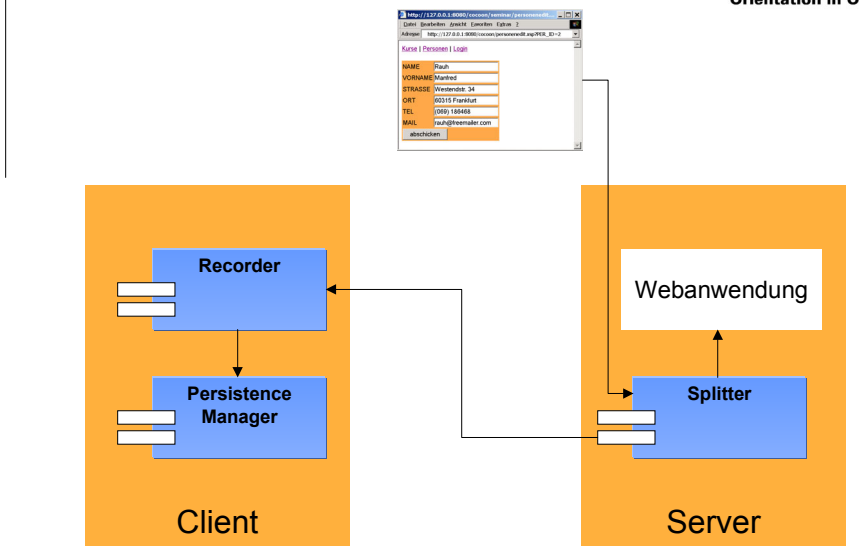
Komponenten

Orientation in Objects



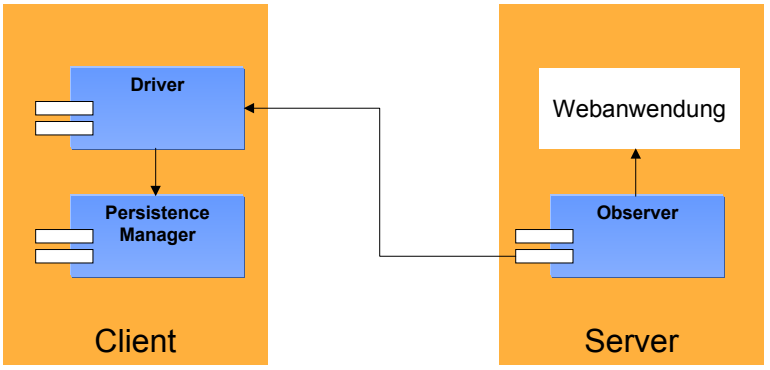
Konfiguration: Recorder

Orientation in Objects



Konfiguration: Player

Orientation in Objects



Agenda

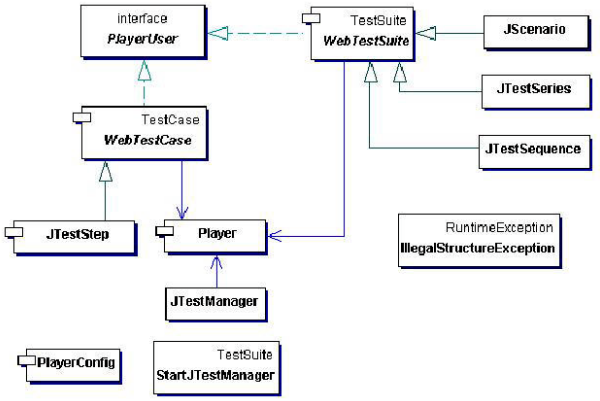
Orientation in Objects

- Motivation
- Projektentscheidung
- Fachliche Idee
- Demo 1
- Grobarchitektur
- Integrationsideen
- Vollständiger Funktionaler Test
- Demo 2
- Fazit

Testdriver

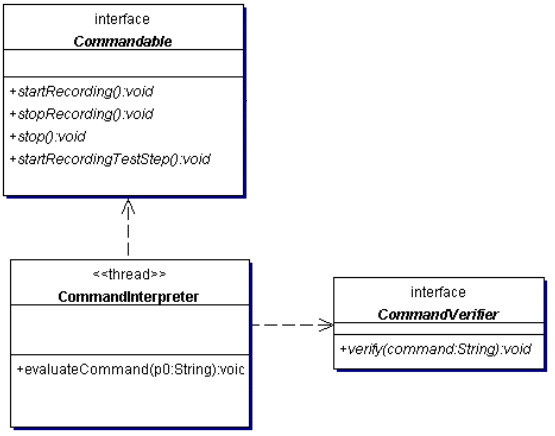
Orientation in Objects

- JUnit Framework (HttpUnit - Schritt I)



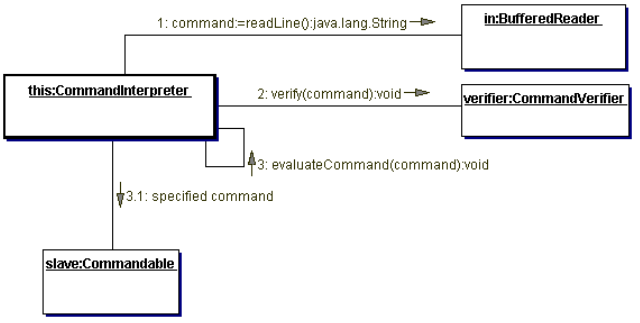
Commandable

Orientation in Objects



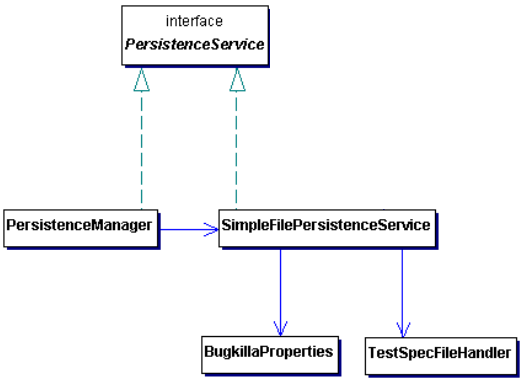
Commadable - collaboration

Orientation in Objects



TestSpec Persistence

Orientation in Objects



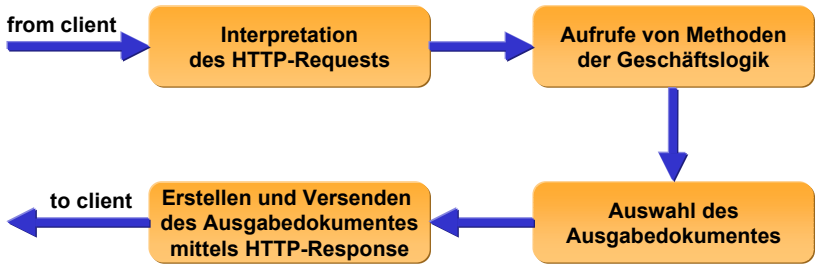
Agenda

Orientation in Objects

- Motivation
- Projektentscheidung
- Fachliche Idee
- Demo 1
- Grobarchitektur
- Integrationsideen
- Vollständiger Funktionaler Test
- Demo 2
- Fazit

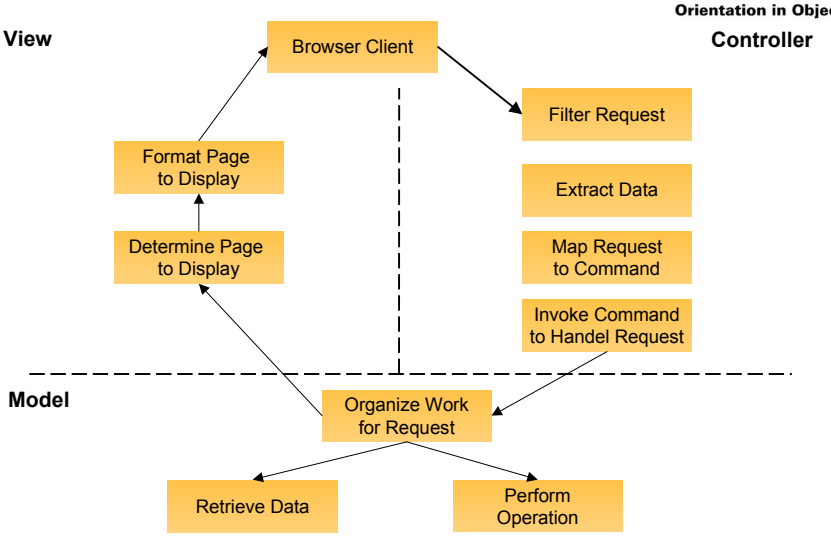
Web tier service cycle

Orientation in Objects



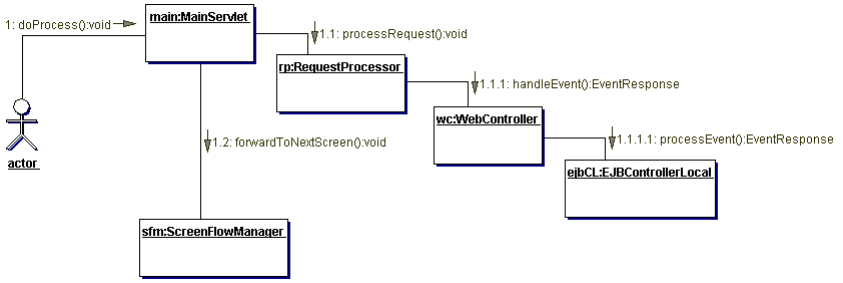
WAF - MVC

Orientation in Objects



WAF

Orientation in Objects



AspectJ

Orientation in Objects

- Instrumentierung - ein orthogonaler Aspekt
- Vollständige Trennung der Anwendung von der Instrumentierung (generativer Ansatz)

- Formulierung einer Strategie
- Definition von Eingriffspunkten
- Integration in die Anwendung („Einweben“)
- Veränderung des Entwicklungsprozesses
  - Konfigurationsmanagement
  - weiteres Paradigma notwendig

Agenda

Orientation in Objects

- Motivation
- Projektentscheidung
- Fachliche Idee
- Demo 1
- Grobarchitektur
- Integrationsideen
- Vollständiger Funktionaler Test
- Demo 2
- Fazit

Agenda

Orientation in Objects

- Motivation
- Projektentscheidung
- Fachliche Idee
- Demo 1
- Grobarchitektur
- Integrationsideen
- Vollständiger Funktionaler Test
- Demo 2
- Fazit

Stand der Dinge

Orientation in Objects

- Recorderkonfiguration
  - integriert und „lauffähig“ seit mehreren Releases
- Playerkonfiguration - Schritt I
  - HttpUnit vollständig integriert
- Playerkonfiguration - Schritt II
  - als Architekturprototyp im CVS Head
- IDE Integration
  - Eclipse Plugin
- Qualitätssicherung
  - automatisierter Komponententest
  - Selbsttest
  - automatisierte Code Coverage Messung
  - Cruise Control

Stand der Dinge II

Orientation in Objects

- Unhandliche Recorderkonfiguration
- manuelle Erstellung von Testorakeln
- keine Unterstützung von EJB-Instrumentierung
- Objektmodell noch nicht vollständig IEEE konform

- Wenig Tools für RAD von J2EE Webanwendungen
- keine build-in-Tests in J2EE

Planung

Orientation in Objects

- Q 3
  - Release mit Strutsinstrumentierung
- Q 4
  - Weber für weitere Webframeworkinstrumentierung
  - Instrumentierung der EJB-Schicht

Ausblick

Orientation in Objects

- Proxy-Recorder-Konfiguration
- Benutzung von Aspekten in Recorderkonfiguration
- Automatisierung der Spezifikation von Zustandsübergängen
- UML Testprofile

Orientation in Objects

Vielen Dank für Ihre Aufmerksamkeit

Dirk M. Sohn
<sohn@oio.de>

Christian Dedek
<dedek@oio.de>

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
http://www.oio.de
info@oio.de

http://bugkilla.sourceforge.net/