

Testen mit AspectJ

AspectJ für den Test von J2EE Komponenten













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

Christian Dedek
<dedek@oio.de>

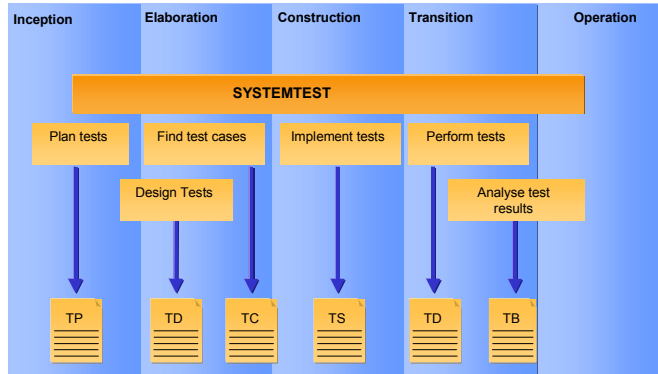
Matthias Niete
<niete@oio.de>

1

XP - the central principles

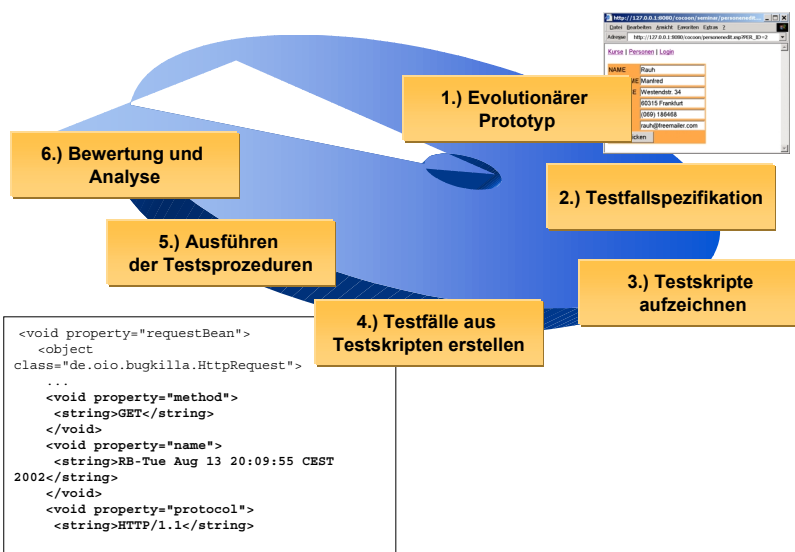
- Fast and detailed feedback
 - test driven design with acceptance tests 
 - Customer On-site 
 - pair programming 
- Common understanding
 - planning game 
 - simple design 
 - System metaphor 
 - collective code ownership 
 - coding conventions 
- Continuous process:
 - continuous integration 
 - refactoring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

Classical System Test

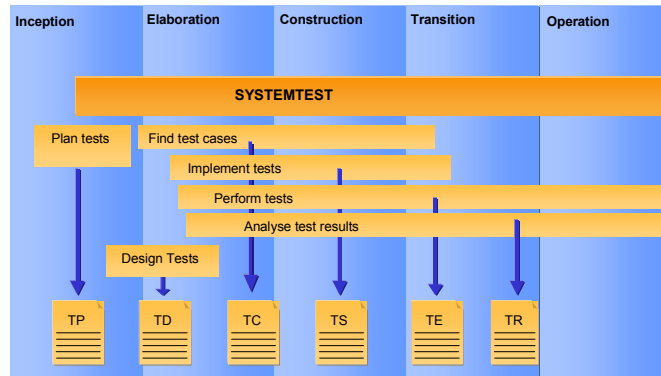


TP - Test plan
 TD - Test design specification
 TC - Test case specification
 TS - Test scripts
 TE - Test execution protocol
 TR - Test report

Fachliche Idee



System Test



TP - Test plan
TD - Test design specification
TC - Test case specification
TS - Test scripts
TE - Test execution protocol
TR - Test report

Vollständiger Funktionaler Systemtest

- 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



- 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 Zustandsübergänge
 - ???

7

Automatische Auswertung



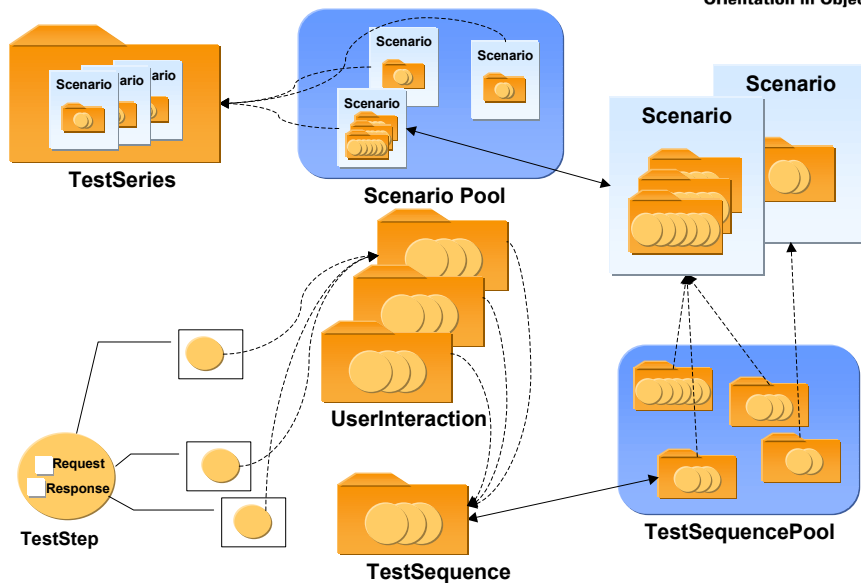
- Auswertung der direkten Ausgaben (Response-Dokumente)
 - Vergleich der Antworten der Testfallspezifikation mit denen der Testdurchführung - einfach nur HTML-Daten auswerten.
- Auswertung der Zustandsübergänge im Container
 - Übersetzung auf technische Objekte bei Einsatz von Frameworks deutlich präzisierbar.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

8

OIO Tooling philosophy

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.

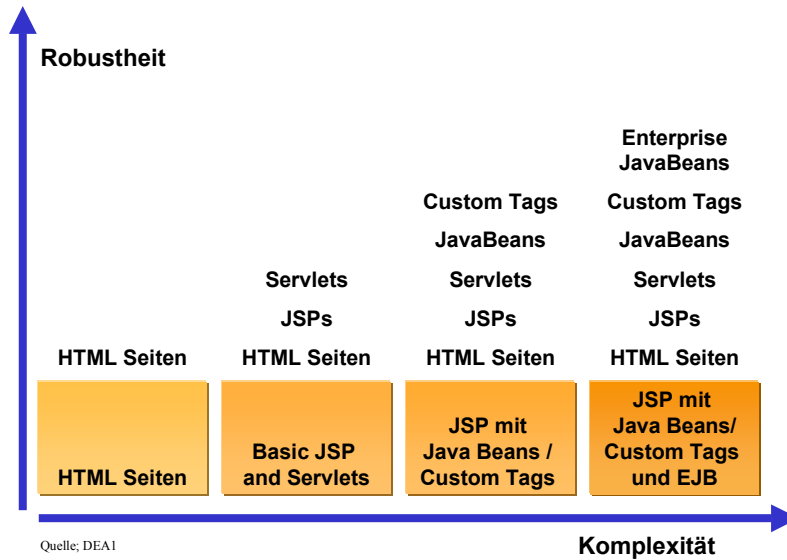
Objektmodell



- 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

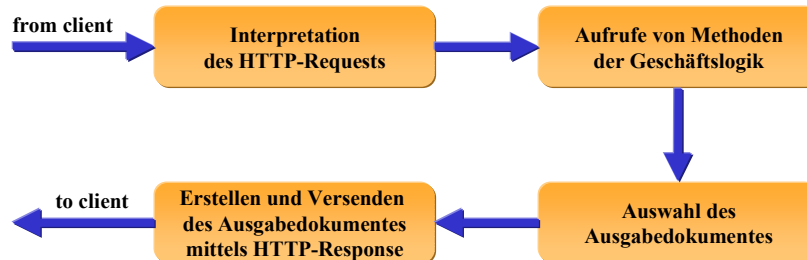
- SzenarioPool
 - beinhaltet sämtliche Szenarien in Form von Szenarien-Files (zusammenhanglos)
- TestSeries
 - Anordnung mehrerer Szenarien zum einer Testserie. Diese Szenarien haben untereinander keine Abhängigkeiten und werden der Reihe nach abgespielt. (eventuell bestehen Abhängigkeiten aber dann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

Exkurs: Application Design Spektrum



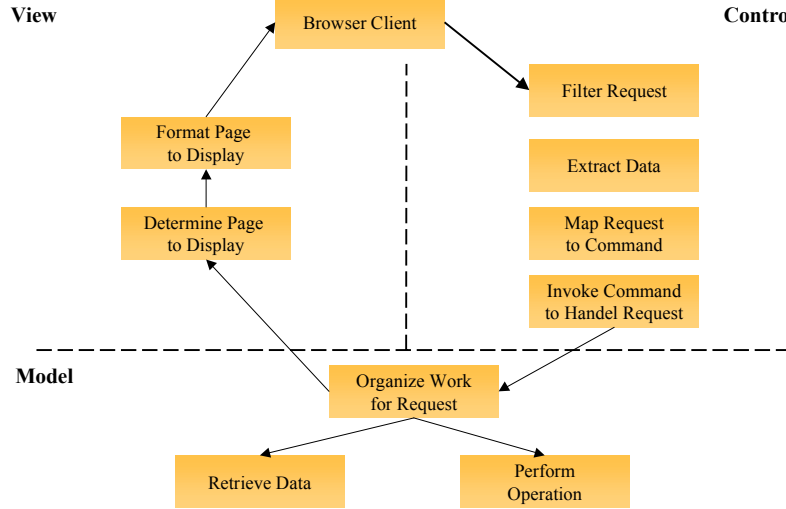
13

Web tier service cycle

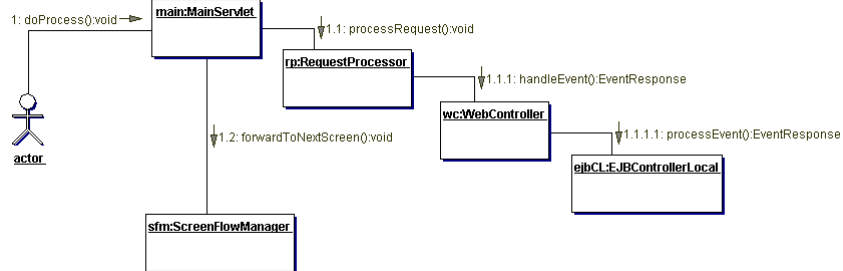


14

WAF - MVC



WAF



Aufzeichnung einer Testsequence I



Orientation in Objects

The screenshot shows the Eclipse IDE with the Bugkilla framework. The main window is a web browser displaying the 'Java Pet Store demo' website, which features a large green parrot and several smaller animal icons. On the left, the Package Explorer shows the project structure. The TestSequencesView on the right lists several test sequences, including 'add a lot to shopping cart (534)', 'sign in user2 (scenario265)', and 'sign in user1 (944)'. The Control Unit panel on the far right contains buttons for 'Recorder Control', 'Start Recorder', 'Start Sequence', 'Interaction (Count 2)', 'Stop Sequence', 'Reset', 'Stop Recorder', 'DocServer Control', 'Start DocServer', and 'Stop DocServer'.

17

Aufzeichnung einer Testsequence II



Orientation in Objects

The screenshot shows a Windows command prompt window titled 'cmd.exe - run.bat - c:\petstore'. It displays a log of test execution steps, including messages like 'Request - http://localhost:8080...', 'FilterManager.processIestepState()', and 'Command: start - resolved <Bugkilla>'. To the right, the Control Unit dialog box is open, showing buttons for 'Recorder Control', 'Start Recorder', 'Start Sequence', 'Interaction (Count 1)', 'Stop Sequence', 'Reset', 'Stop Recorder', 'DocServer Control', 'Start DocServer', and 'Stop DocServer'. Two blue callout boxes with the number '1.' point to the 'Start Recorder' and 'Start Sequence' buttons, and another callout box with the number '2.' points to the 'Interaction (Count 1)' button.

18

Aufzeichnung einer Testsequence III



Orientation in Objects

19

Testen mit AspectJ

© 2004 Orientation in Objects GmbH

Aufzeichnung einer Testsequence IV



Orientation in Objects

20

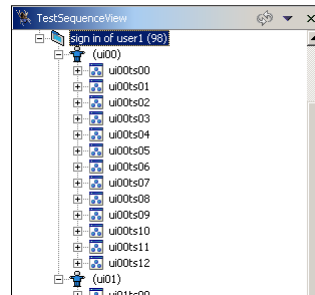
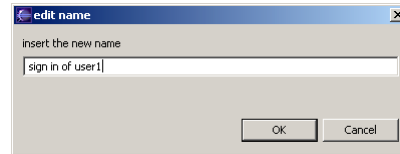
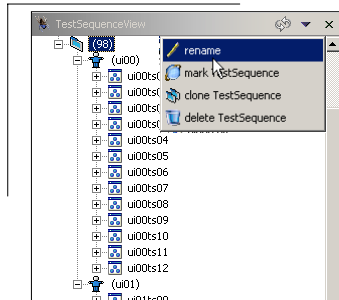
Testen mit AspectJ

© 2004 Orientation in Objects GmbH

Kennzeichnung der Testsequence



Orientation in Objects

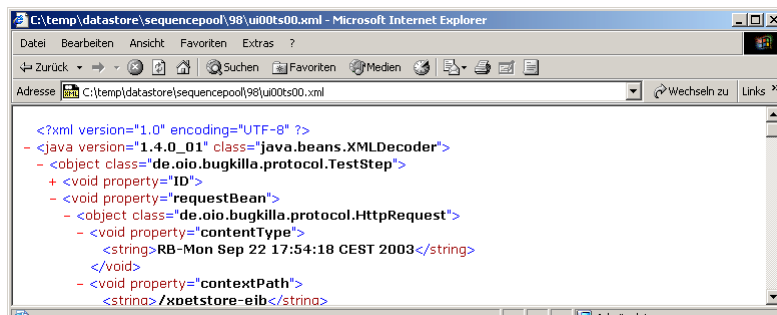
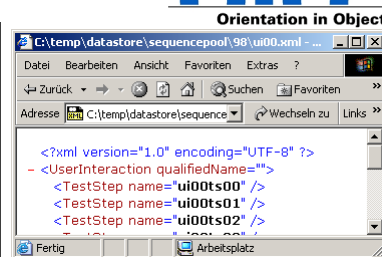
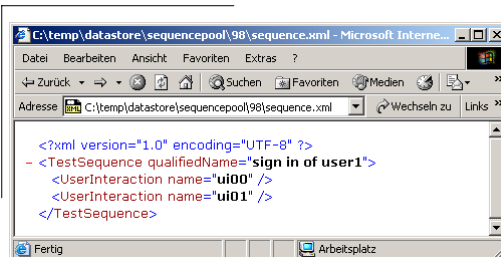


21

Aufzeichnungen

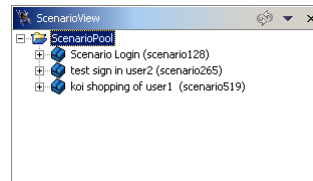
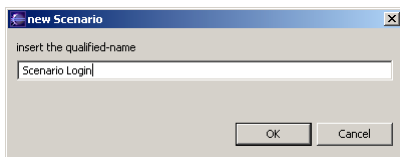
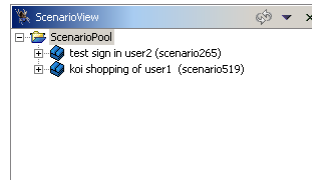
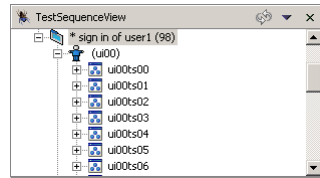


Orientation in Objects



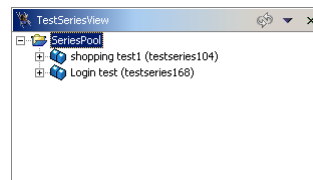
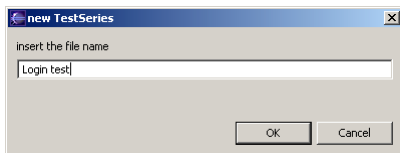
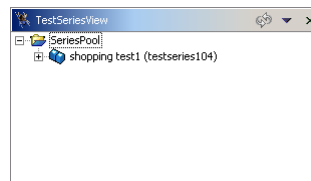
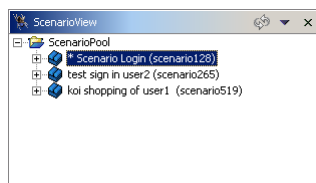
22

Erstellen eines Szenarios



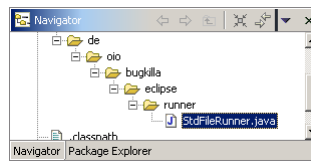
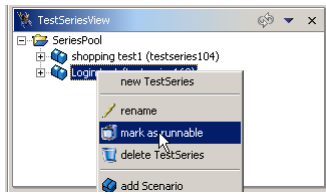
23

Erstellen einer TestSeries



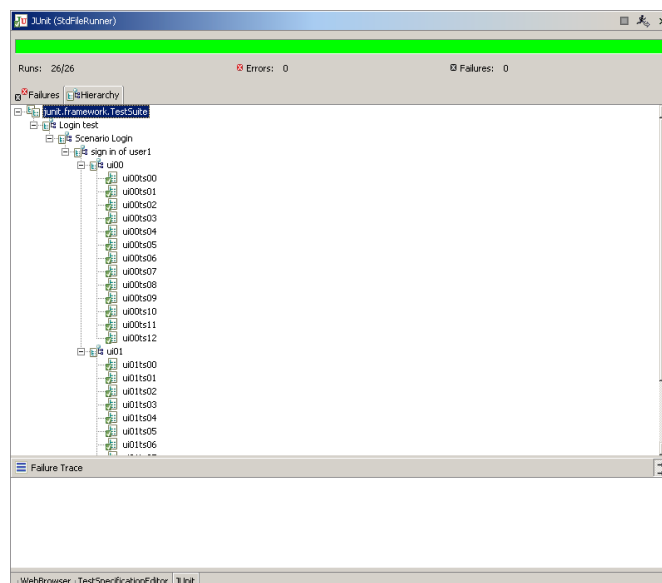
24

Ausführung einer TestSeries



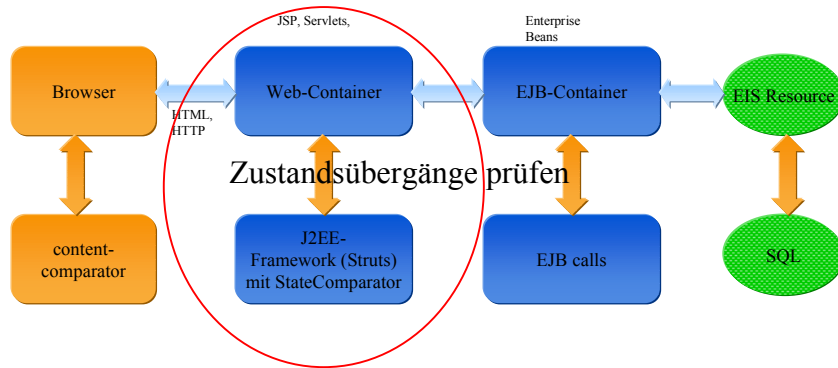
25

JUnit Test View

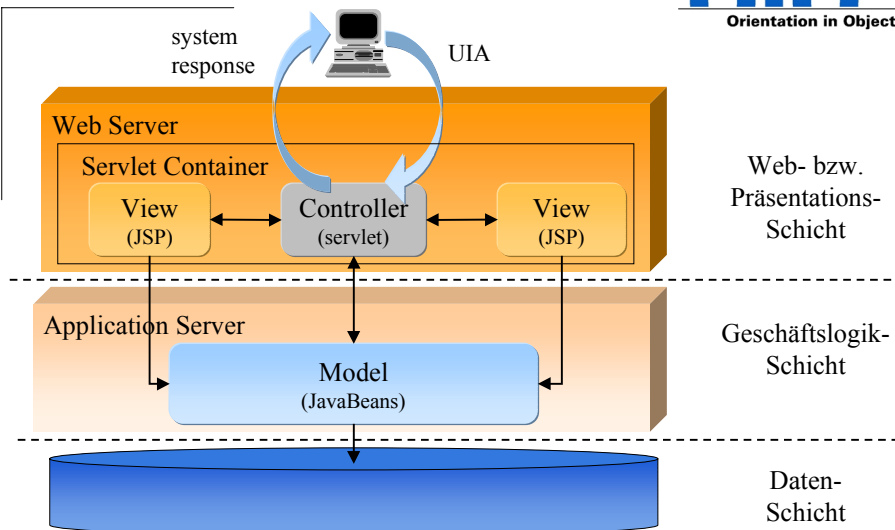


26

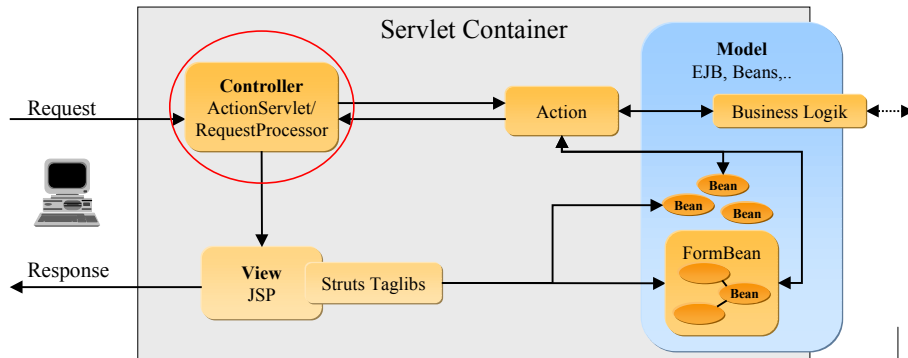
Der Testort



JSP - Model 2



Jakarta Struts Framework



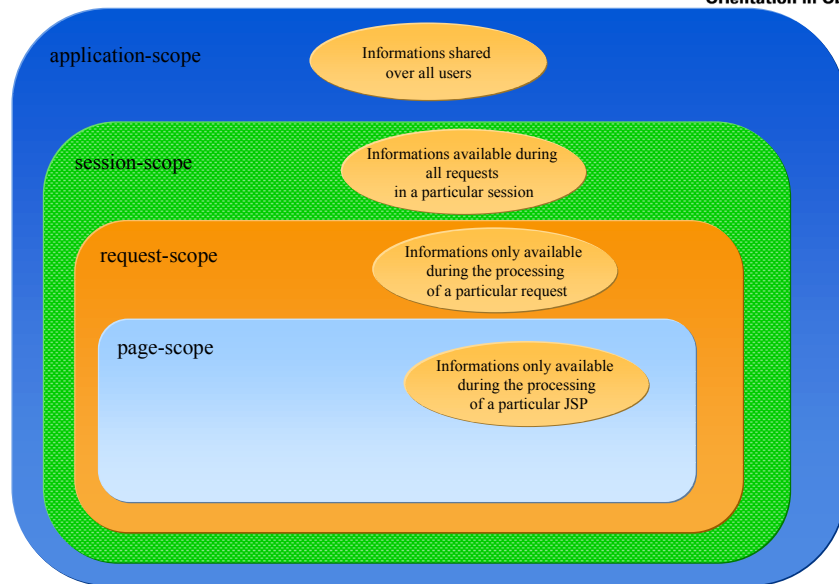
29

Informationen des Controllers

- aktuelle Anfrageinformationen des Anwenders und sein derzeitiger Anfrage- und Sitzungszustand.
 - Referenzen auf DTO
- Welche Geschäftslogik auszuführen ist
 - Ausführungszeitpunkte der Geschäftslogik
- Ergebnis der Ausführung der Geschäftslogik

30

Zustände in der Web-Schicht



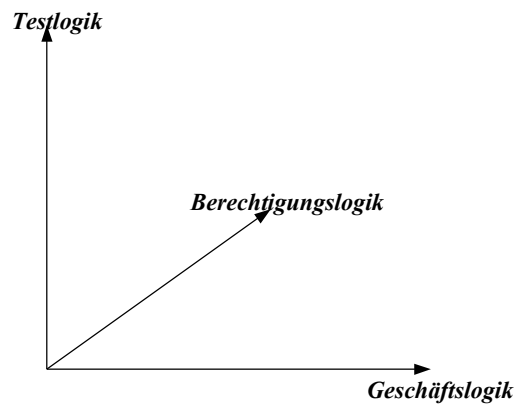
31

Glossar

- POT (Point of Test)
 - Definiert den Ort in Verbindung mit einem Verarbeitungszeitpunkt an dem ein Test innerhalb eines Testlings durchgeführt wird
- Komparatoren
 - Extraktion von Testwerten (Objektzuständen) aus dem Testling
 - Messung eines Testlings durch Vergleichsoperationen der aktuellen Zustände des Testlings mit den erwarteten Zustand
 - Zustandsvergleiche erfolgen über direkte Werte-Vergleich oder über Verhaltens-Vergleiche der Testobjekte
- Komparatorensteuerung
 - Aktivierung und Auswahl durch den Testtreiber bei Erreichen von POTs

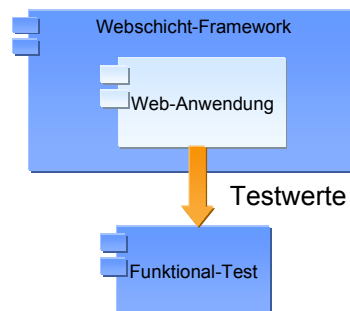
32

Orthogonale Aspekte der SE



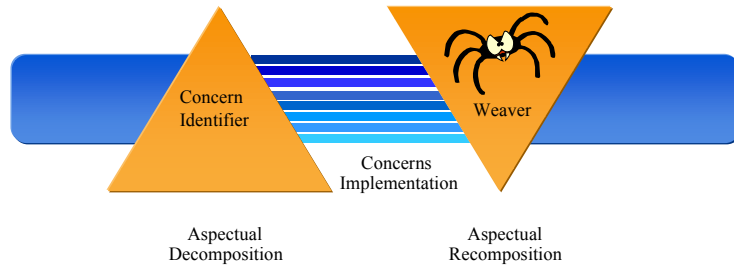
33

Schlechte Abhängigkeit



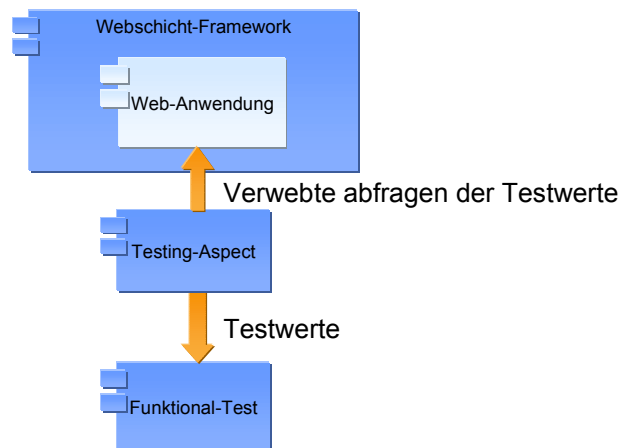
34

Aspektorientierung



35

Aufhebung der Abhängigkeit



36

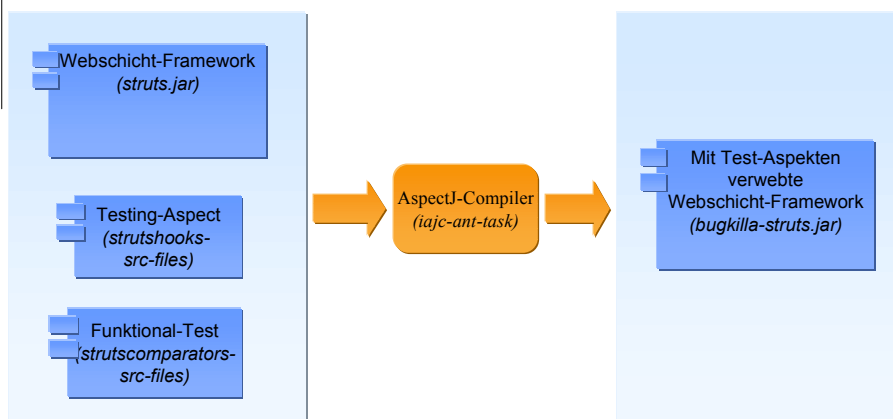
AspectJ

- 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

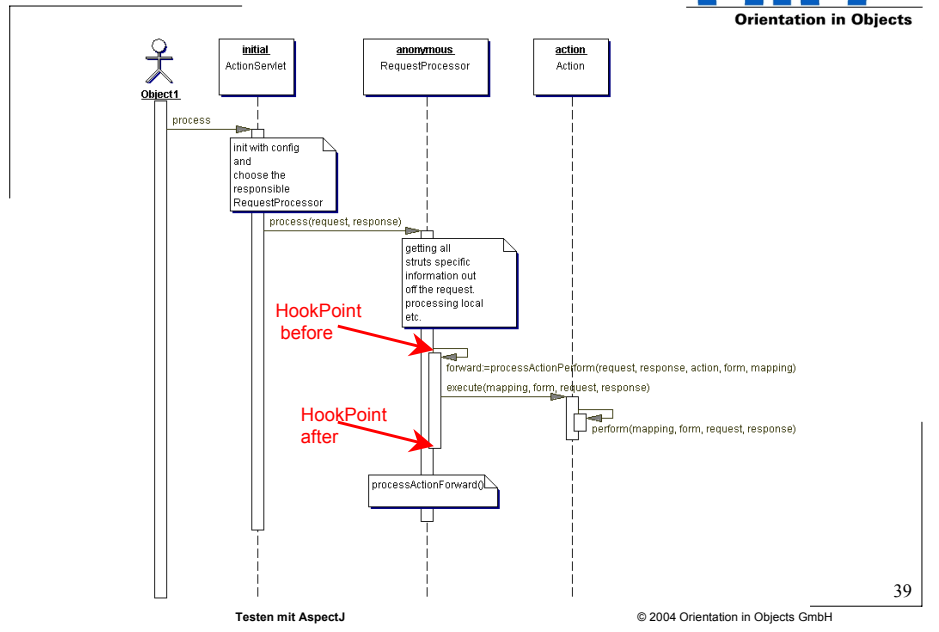
37

Verweben der Komponenten



38

POTs im Struts Framework

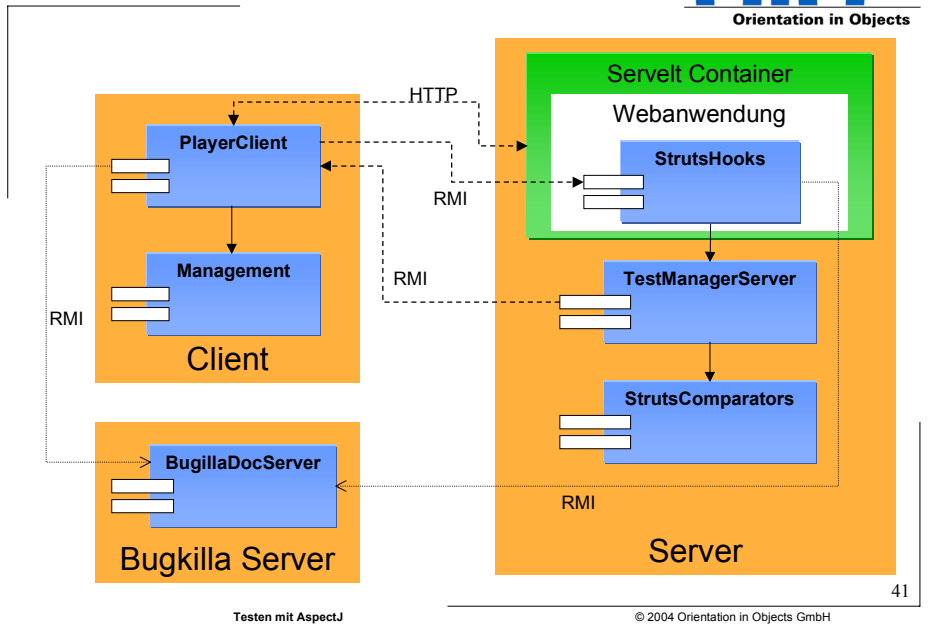


AspectJ im Struts Framework

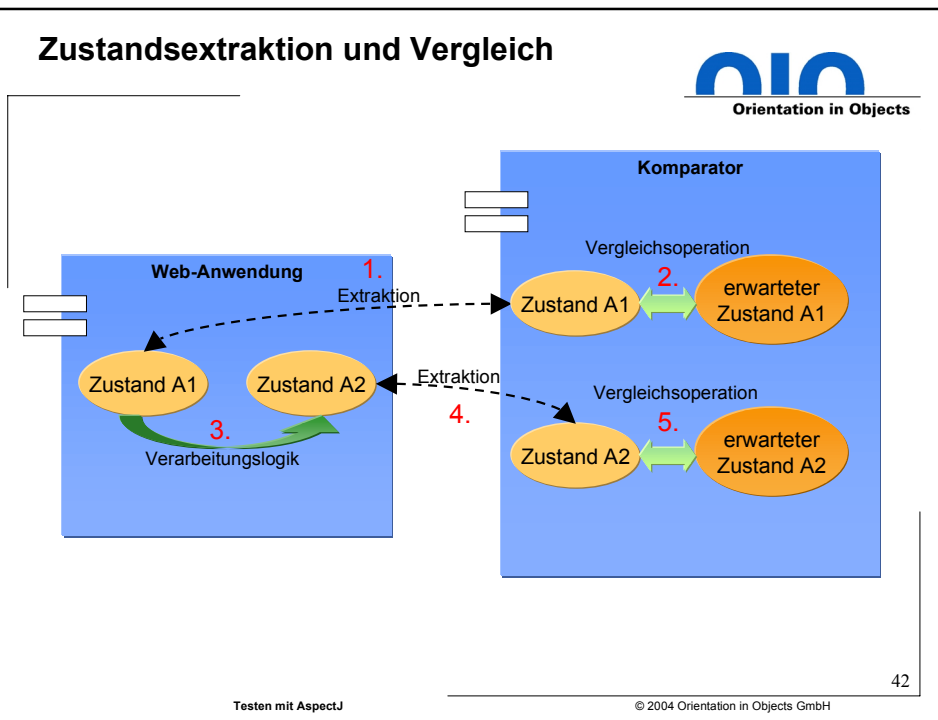
```
protected ActionForward
    processActionPerform(HttpServletRequest request,
                        HttpServletResponse response,
                        Action action,
                        ActionForm form,
                        ActionMapping mapping)
        throws IOException, ServletException {
    RequestProcessorHook.pointCutBefore(request, ...);
    try {
        return (action.execute(mapping, form, request, response));
    } catch (Exception e) {
        return (processException(request, response,
                                e, form, mapping));
    }
    RequestProcessorHook.pointCutAfter(request, ...);
}
```



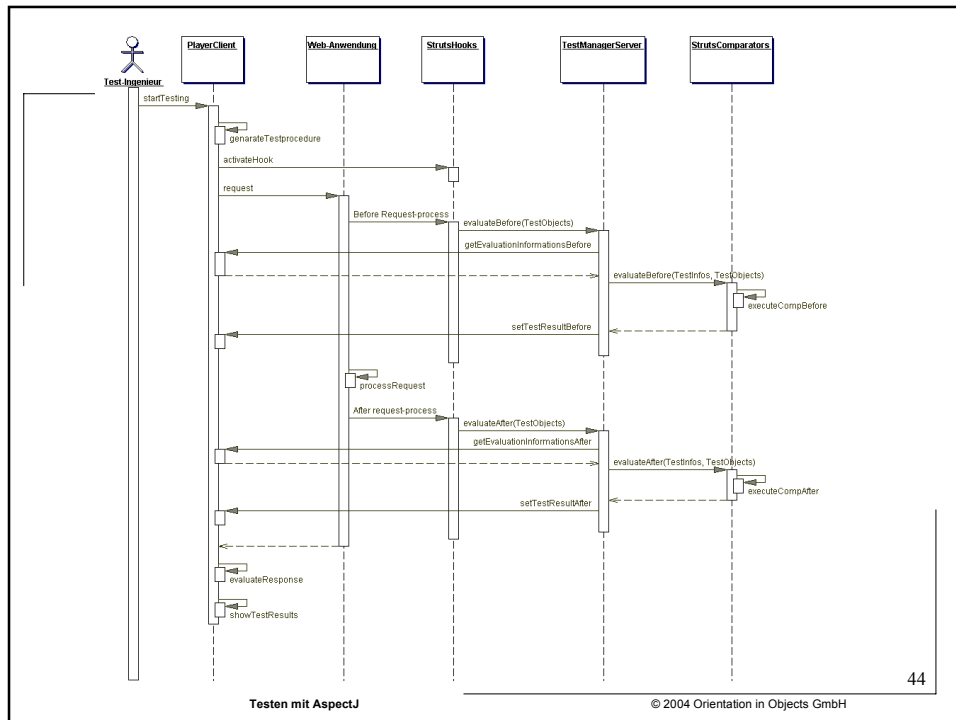
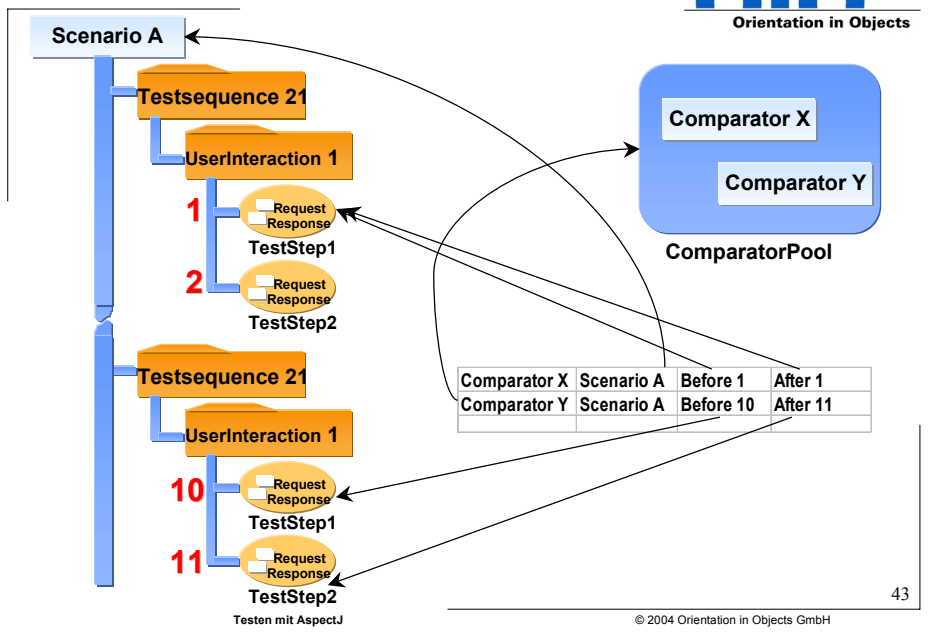
Konfiguration: Player



Zustandsextraktion und Vergleich



Komparatorenverwaltung



Vielen Dank für Ihre Aufmerksamkeit

Christian Dedek
<dedek@oio.de>

Matthias Niete
<niete@oio.de>

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