



– Grails –  
Der Gral der  
Webentwicklung

Stefan Rook

[stefan.roock@akquinet.de](mailto:stefan.roock@akquinet.de)

Bernd Schiffer

[bernd.schiffer@akquinet.de](mailto:bernd.schiffer@akquinet.de)

03.07.2008

JFS





①

**FAIL!**

②



**FAIL!**

**FAIL!**



3





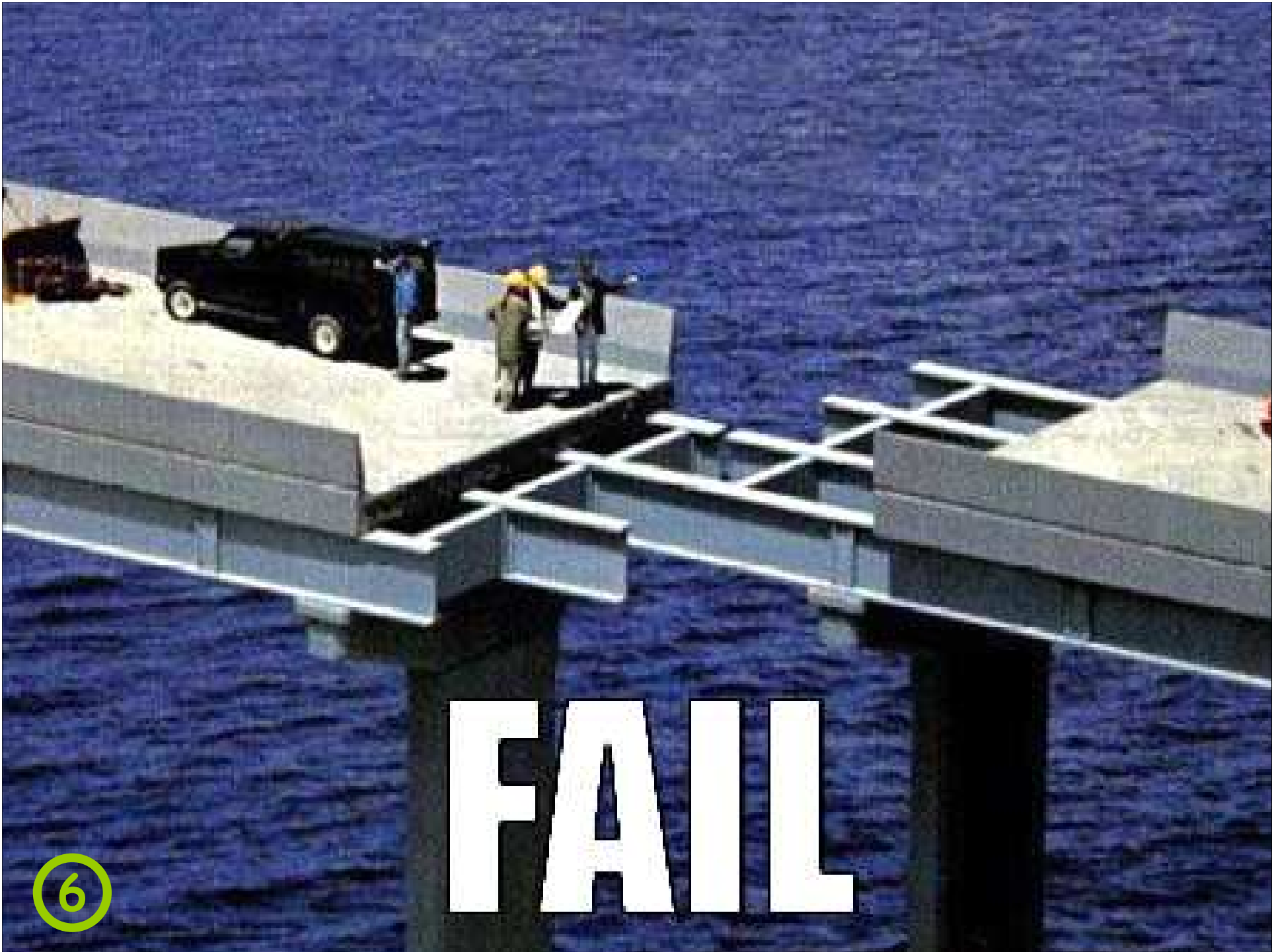
④

**FAIL!**



FAIL

5



6

FAIL

**FAIL**

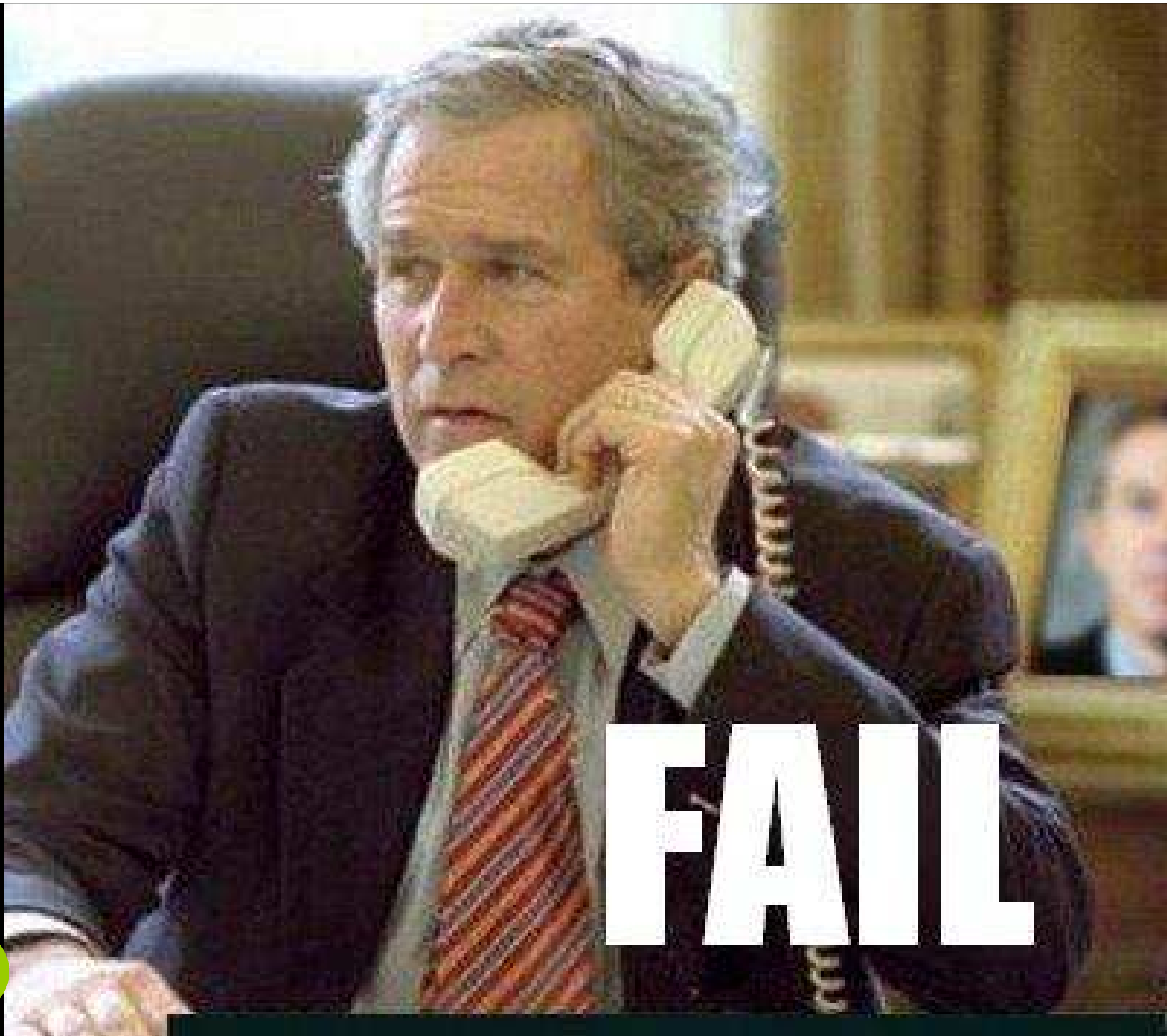






**FAIL**

⑧



9

**FAIL!**





11

**FAIL!**



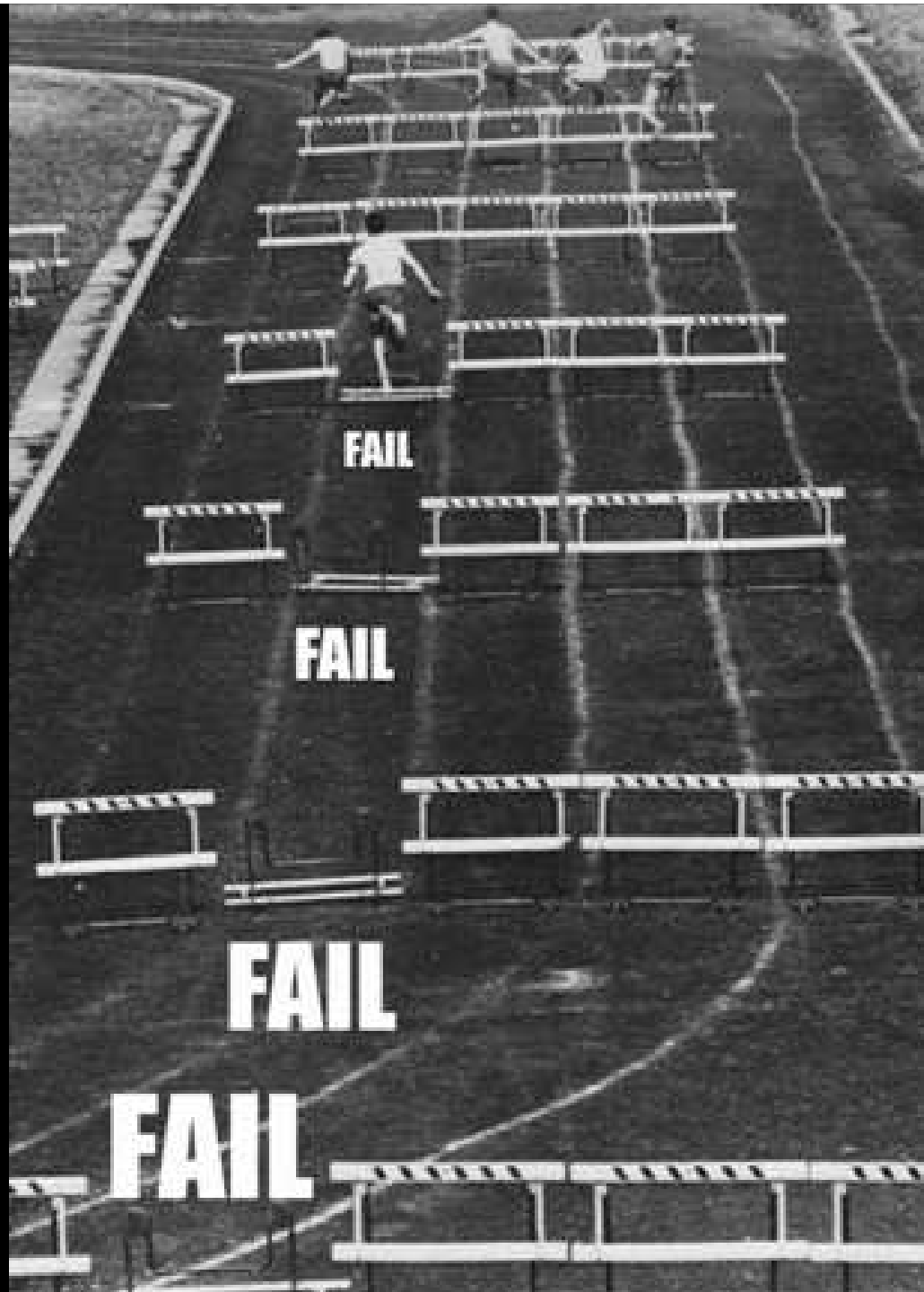


**fail**

13

Fail

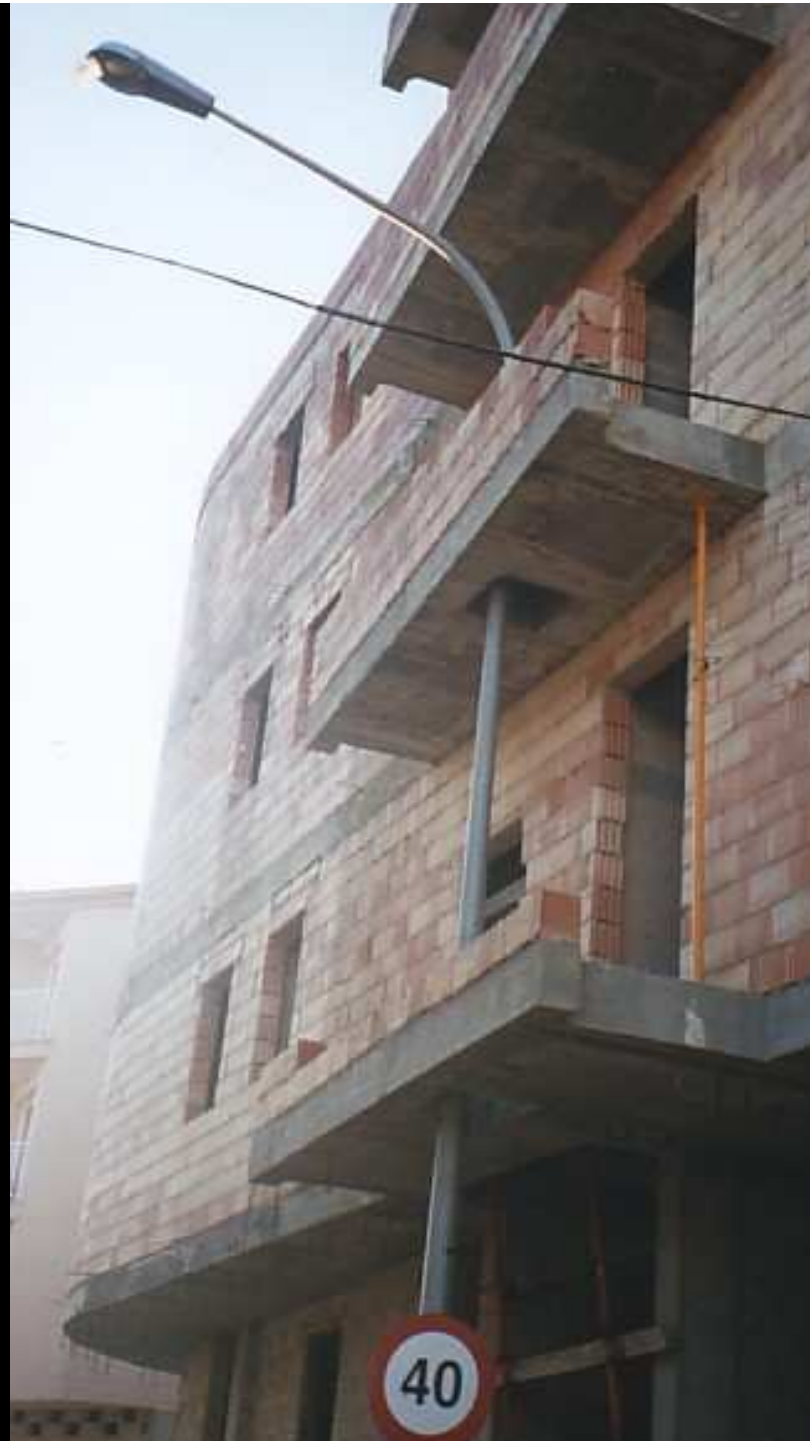




15



**FAIL!**



**16**

**FAIL!**



FAIL!



18

**FAIL WIN FAIL**



[WWW.SHIPMENTOFFAIL.CO](http://WWW.SHIPMENTOFFAIL.CO)

**FAIL**





## Herzlich Willkommen!



### Bernd Schiffer

- akquinet AG
- Coach für agile Methoden, Projektleiter, Softwareentwickler
- XP, Scrum, TDD, Groovy, Grails



### Stefan Roock

- akquinet AG
- Coach für agile Methoden, Projektleiter, Softwarearchitekt
- XP, Scrum, Akzeptanztests, TDD, Refactoring ...

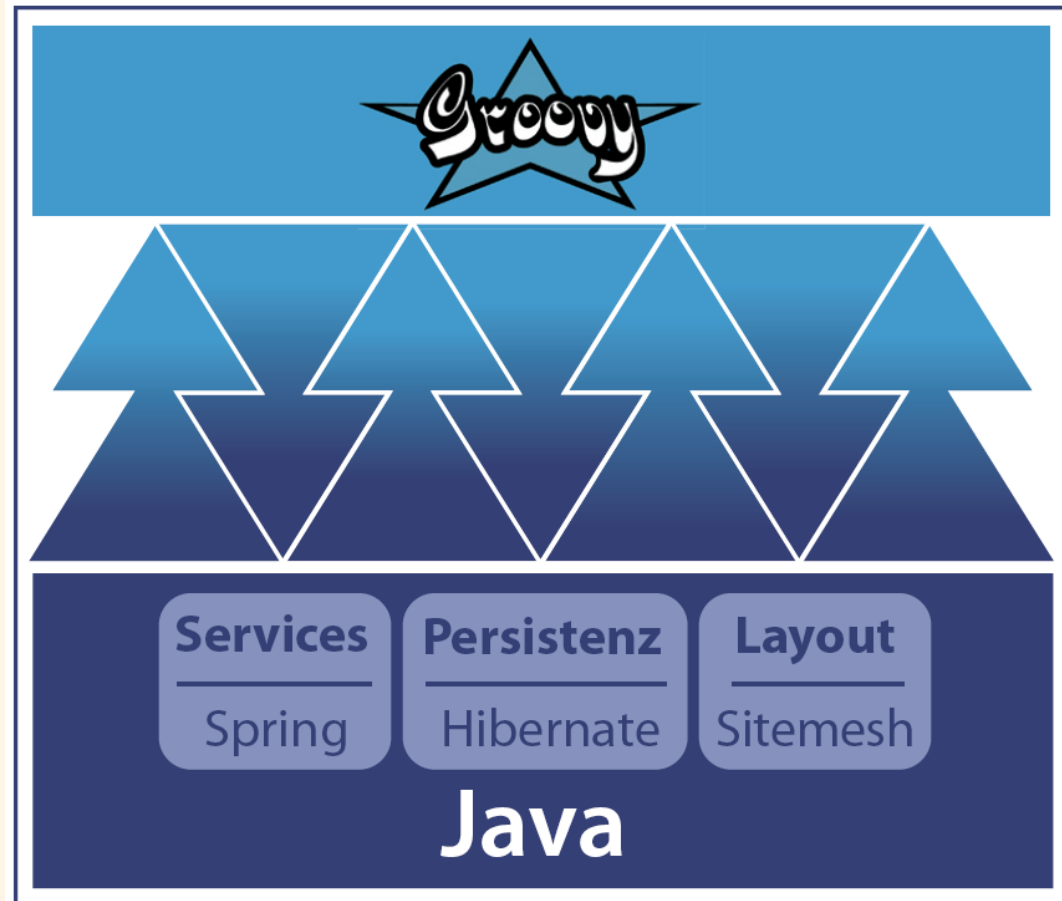
## Was wir heute zeigen

- Vorstellung von Grails
- Live-Demo: Wir basteln uns eine Session





## Grails – Der Zuckerguss 1/2



## Grails – Der Zuckerguss 2/2



**Grails-Prinzipien:**

- Konvention statt Konfiguration
- DRY: Don't Repeat Yourself

### Ajax

Dojo

Prototype

Yahoo UI

...

### Batch

Quarz

### Sicherheit

Acegi

...

### Testen

JUnit

Canoo  
Web Test

...



## Groovy – Grails' Dynamik 1/2

- Literale

```
def list = [1, 2, 3, 4]  
def map = [a:1, b:2]
```

- GStrings

```
def vorname = 'Micky'  
def nachname = 'Maus'  
assert 'Mein Name ist Maus, Micky Maus.' ==  
"Mein Name ist $nachname, $vorname $nachname."
```

## Groovy – Grails' Dynamik 2/2

- Closures und Listenoperationen

```
def list = [5, 2, 1, 3, 4, 4].findAll{ it > 2 }.sort().unique()
assert [3, 4, 5] == list
```

- RegExp

```
'23.12.2007, 23.09.2004'.eachMatch(/(\d{2}\.\d{2}\.\d{4})/) { alles, datum ->
    println datum
}
```

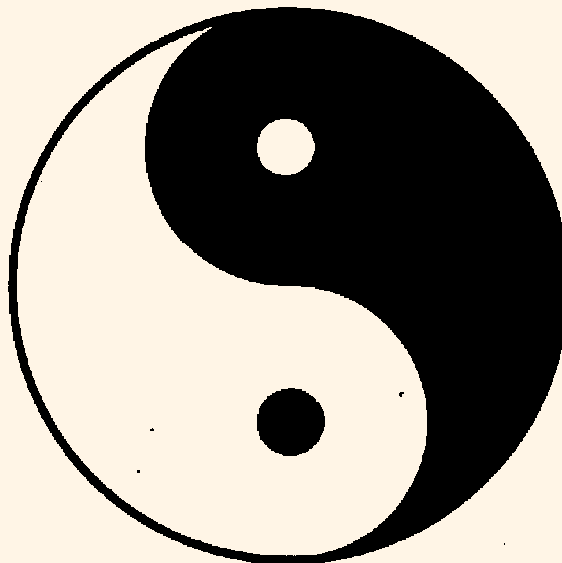
- IO

```
new File('file.txt').eachLine{
    line -> println line
}
```

23.12.2007  
23.09.2004

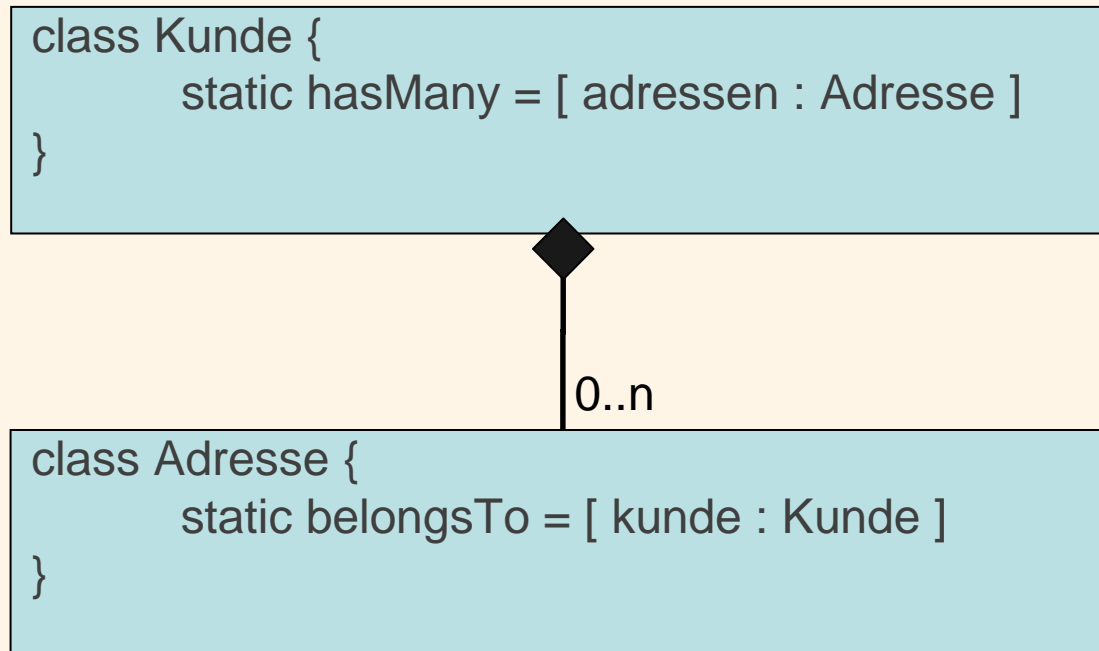
## Java – Die Basis

- Nahtlose Verschmelzung von Java und Groovy
- Nutzung aller vorhandenen Java-Ressourcen
- Objekte, Klassen, Threads, Security – keine Überraschungen



## Domain-Objects & GORM 1/2

- Domain-Objects werden automatisch gemappt



## Domain-Objects & GORM 2/2

- Statische Typisierung zwecks Mapping

```
class Adresse {  
    String strasse  
    int hausnummer  
    String hausnummernzusatz  
    int postleitzahl  
    String ort  
    Date gueltigVon  
    Date gueltigBis  
}
```

## Constraints – Validierung der Domäne

```
class Adresse {  
    String strasse  
    int hausnummer  
    ...  
    Date gueltigVon  
    //...  
    static constraints = {  
        strasse(blank:false)  
        hausnummer(blank:false, min:1)  
        gueltigBis(validator: {  
            gueltigBis == null || gueltigVon <= gueltigBis  
        })  
    }  
}
```



## Dynamic Finder – jetzt wird's interessant

```
Adresse.findAllByPostleitzahl(21502)
```

```
Adresse.findAllByPostleitzahlBetween(21000, 21999)
```

```
Adresse.findAllByPostleitzahlAndGueltigBis(21502, null)
```

```
Adresse.findAllByPostleitzahlBetween(21000, 21999,  
  [sort:'postleitzahl', order:'desc'])
```

```
Adresse.list(max:10, offset:100, sort:'postleitzahl', order:'asc')
```

## Views – Wir wollen was sehen!

```
<html>
  <head>
    <title>${kunde.name}</title>
  </head>
  <body>
    <ol>
      <g:each in="${kunde.adressen}">
        <li>${it.postleitzahl} ${it.ort}</li>
      </g:each>
    </ol>
  </body>
</html>
```

## Controller – Model und View verbinden

```
class KundeController {  
  
  def index = { redirect action:show }  
  
  def show = { [kunde:Kunde.get(params.id)] }  
  
  def list = { [kunden:Kunde.list()] }  
  
  def save = {  
    if(!new Kunde(params).save()) {  
      render view: 'error'  
      return  
    }  
    redirect action:list  
  }  
}
```

## AJAX – Grails' Web-2.0-Tauglichkeit

- eingebaut (teilweise via Plugins)
  - Prototype/Script.aculo.us
  - Yahoo! UI (Plugin)
  - Dojo (Plugin)

- eingebunden

```
<g:javascript library="prototype" />
```

- benutzt

```
<div id="message"></div>  
<g:remoteLink action="delete" id="1" update="message">  
  Lösche Kunde</g:remoteLink>
```

- JSON, XML-(Un-)Marshalling

```
import grails.converters.*  
...  
render kunde as JSON
```

## TagLibs – Vergessen ist JSP!

```
<html>
  <head>
    <title>${kunde.name}</title>
  </head>
  <body>
    <g:liste von="${kunde.adressen}">
      ${it.postleitzahl} ${it.ort}
    </g:liste>
  </body>
</html>
```

```
class MeineTagLib {
  def liste = { attrs, body ->
    out << '<ol>'
    attrs.von.each{
      out << '<li>'
      out << body(it)
      out << '</li>'
    }
    out << '</ol>'
  }
}
```

## Plugins

- Grails ist selbst aufgebaut aus Plugins
- per Plugin einbindbare Techniken
  - Canoo Webtest, GWT, DWR, X-Fire, XML-RPC, Quartz, Spring Security (Acegi), Jcaptcha, Compass + Lucene, LiquiBase, dbmigrate, u.v.v.m.
  - <http://grails.codehaus.org/plugins>
- Modularisierung mit eigenen Plugins
  - pro Modul ein Plugin
  - technisch oder fachlich geschnitten
- Beeinflussung durch Plugins
  - Artefakte, Skripte, MetaObject-Programmierung, EventHandlerler, Bibliotheken, uvm.

## Testen

- Grails ist agil ...
  - Unittests, Integrationstests (inhärent)
  - Akzeptanztests (Plugins: Canoo Webtest und Selenium)
- ... weil es Testen einbindet ...
  - Test-Skelette pro Domain-Class und Controller generiert
  - Ausführung der Tests per Skript (einschl. Reports)
- ... aber es ist noch viel zu un-extreme ...
  - Zu langsam!
- ... woran gearbeitet wird!
  - Interactive mode

## REST (1/2)

- URL-Mapping

```
class UrlMappings {
  static mappings = {
    "/kunde/$id?"(controller: 'kunde') {
      action = [
        GET:      'show',
        PUT:      'update',
        DELETE:   'delete',
        POST:     'save'
      ]
    }
  }
}
```



## REST (2/2)

- Content-Negotiation

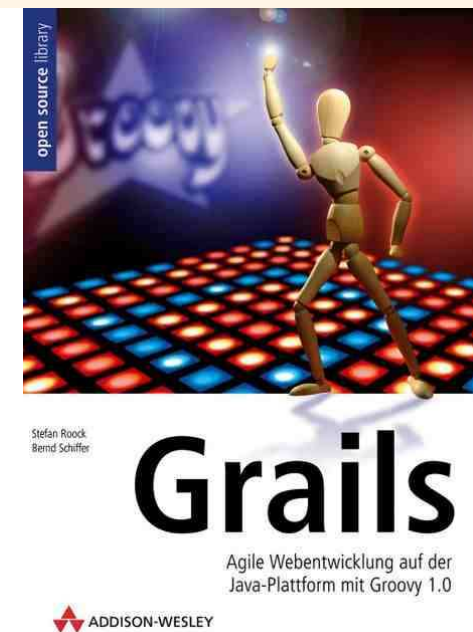
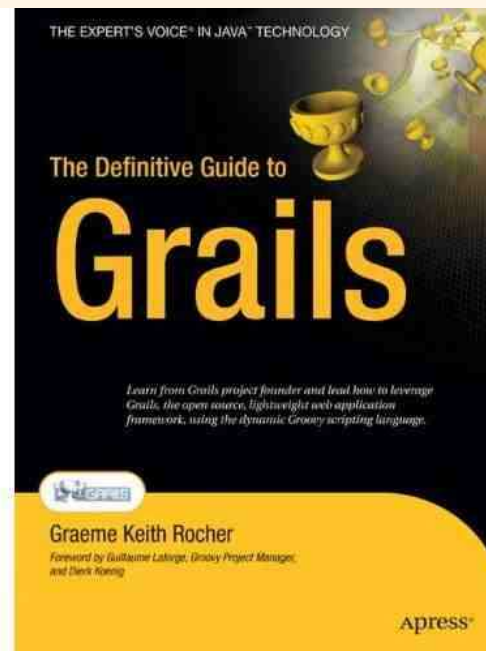
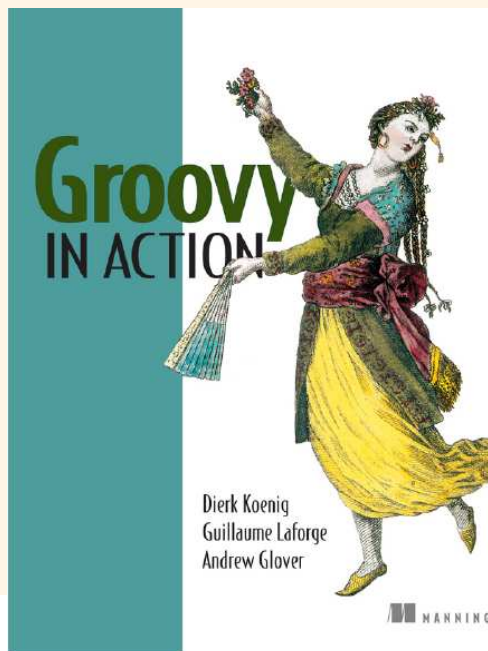
```
import grails.converters.*
...
def show = {
    def kunden = params.id ? kunde.get(params.id) : kunde.list()
    withFormat {
        html { render template: 'show', model: [kunden: kunden] }
        xml { render kunden as XML }
        json { render kunden as JSON }
    }
}
```

## Grails auf Erfolgskurs

- Grails 1.0 seit Anfang 2008
- Grails 1.0.3 momentan stabil
- großer internationaler Zuspruch:
  - SAP: Composition on Grails (Netweaver basiert)
  - Webhoster Contegix' Management auf Grails-Basis
  - showbiz.sky.com: > 180 Mio. Hits/Monat
- erste Webseiten in Deutschland
  - <http://www.jobinsiders.de>
  - <http://www.schnaepchenfuehrer.com>
  - <http://www.planet-ria.org>
  - <http://team-radar.it-agile.de>
  - <http://werkannwann.de> (alpha)

## Lust bekommen auf mehr?

- Groovy: <http://groovy.codehaus.org>
- Grails-Mailinglisten: <http://grails.codehaus.org/Mailing+lists>
- Grails: <http://grails.org/>



## Foto-Quellen

- Wrong: <http://www.doingitwrong.com/>
  - 2, 10, 16, 18
- Shipment of Fail: <http://www.shipmentoffail.com/>
  - 19, 21
- The FAIL Blog: <http://failblog.wordpress.com/>
  - 1, 3-9, 11-15, 17, 20



## Und jetzt die Live-Demo!



- Schulungen von akquinet it-agile GmbH zu
  - Groovy: <http://www.akquinet.de/content/java-sap/landing-pages/groovy.html>
  - Grails: <http://www.akquinet.de/content/java-sap/landing-pages/grails.html>
- Beratung, Coaching, Inhouse-Schulung
- Entwicklungsunterstützung



Schulung verlängerte Werkbank  
**agile Softwareentwicklung**  
Festpreisprojekte Coaching  
RCP **Systemintegration** Eclipse  
h3270 Hostintegration  
Scrum Refactoring testgetriebene Entwicklung  
Hibernate SAP-Netweaver **OpenSource**  
Ajax JBoss/JEMS Groovy  
**it-agile** eXtreme Programming