

Orientation in Objects

# Groovy und Grails
# Quo vadis?

Berlin Expert Days 2015
17.09.2015

**Orientation in Objects GmbH**

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

## Über mich

Orientation in Objects

Falk Sippach

*Trainer, Berater, Entwickler*

***Schwerpunkte***
*Architektur*
*Agile Softwareentwicklung*
*Codequalität*

## Java, XML und Open Source seit 1998

Orientation in Objects


### ) Software Factory )

- Schlüsselfertige Realisierung von Java Software
- Individualsoftware
- Pilot- und Migrationsprojekte
- Sanierung von Software
- Software Wartung

### ) Object Rangers )

- Unterstützung laufender Java Projekte
- Perfect Match
- Rent-a-team
- Coaching on the project
- Inhouse Outsourcing

### ) Competence Center)

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme
- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmens-weiten Systemen




## Abstract

Orientation in Objects

*Das Jahr 2015 begann turbulent für die beiden bekanntesten Projekte aus dem Groovy Universum. Von der bisherigen "Mutter" Pivotal den Laufpass erhalten, musste sich Groovy auch noch auf die Suche nach einem neuen Zuhause begeben und ist letztlich bei Apache fündig geworden. All diese Unsicherheiten haben die neuen Features der Releases 2.4 (Groovy) bzw. 3.0 (Grails) ziemlich in den Hintergrund gedrängt. Dabei sind die Projekte lebendiger denn je und vor allem schon längst reif für den produktiven Einsatz.*

*Wir werden uns die wichtigsten und interessantesten Neuerungen der vergangenen Releases anschauen und natürlich auch einen Ausblick auf die Zukunft und Roadmaps wagen.*

## Gliederung

Orientation in Objects

- **Motivation + Politisches**
- Groovy
- Grails
- Ausblick



## Warum Groovy und Grails?

Orientation in Objects

- Groovy-Fan seit 1.0 (2006)
- 5+ Jahre Projekterfahrung mit Grails

## Aber es ist ruhig geworden …

Orientation in Objects

Öffentliche Aufmerksamkeit
tendiert gefühlt **gegen Null**!

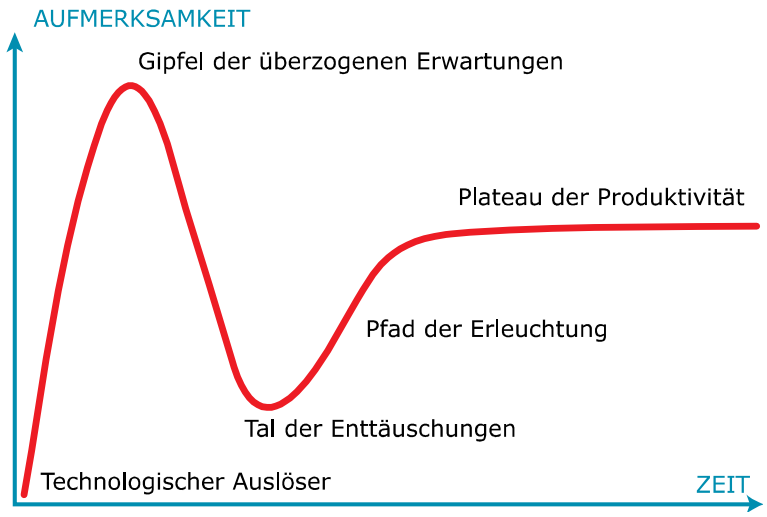


## Zu alt?

Orientation in Objects

Sind Groovy und Grails
**nicht mehr hip** genug?

## Erklärungsversuch 1








## Erklärungsversuch 2

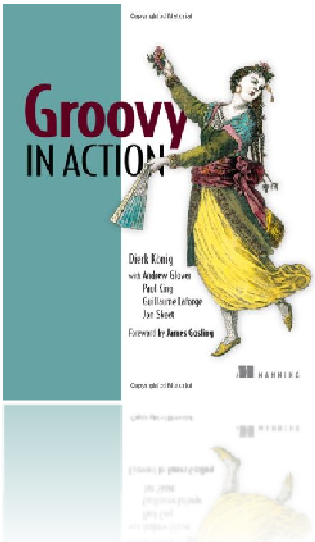
Regina – "Gut Ding will Weile haben"

# Who the fuck is Regina?

## Gehen wir zurück ins Jahr 2007

Orientation in Objects

GinA






## Zwei Jahre später: viele Neuerungen mit 1.5 - 1.7

Orientation in Objects

Planung einer zweiten Auflage von GinA:

**ReGinA** war geboren

Und eine lange "Leidensgeschichte" begann …

## Anfänglicher Optimismus

Orientation in Objects

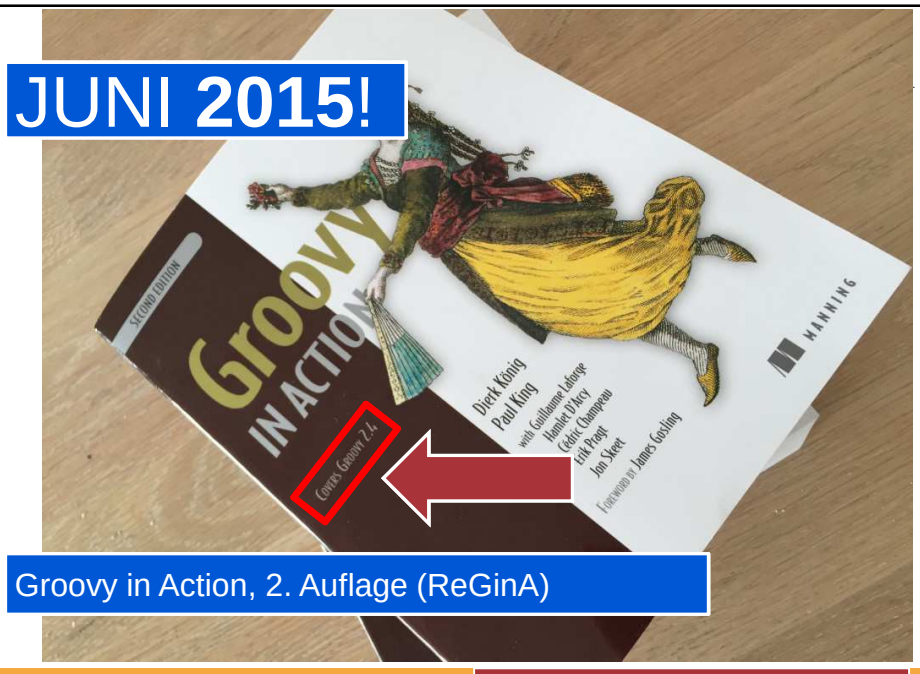
*Hi all,*
*announcing the start of MEAP for June (2009)…*

*Anyway, this summer is the time where we will do the majority of the work on the second edition and you can expect the MEAP progressing rather quickly. …*

*thanks for you understanding*
*Dierk*

https://forums.manning.com/posts/list/18629.page










## Was war passiert? (19.01.2015)

Orientation in Objects

**Groovy 2.4 And Grails 3.0 To Be Last Major Releases Under Pivotal Sponsorship**

> The decision … is part of Pivotal's larger strategy to concentrate resources on … its **growing traction in Platform-as-a-Service, Data, and Agile development**.

> Pivotal has determined that the **time is right** to let further development …be led by **other interested parties** … who can best serve the goals ...

http://blog.pivotal.io/pivotal/news-2/groovy-2-4-and-grails-3-0-to-be-last-major-releases-under-pivotal-sponsorship

## Historie

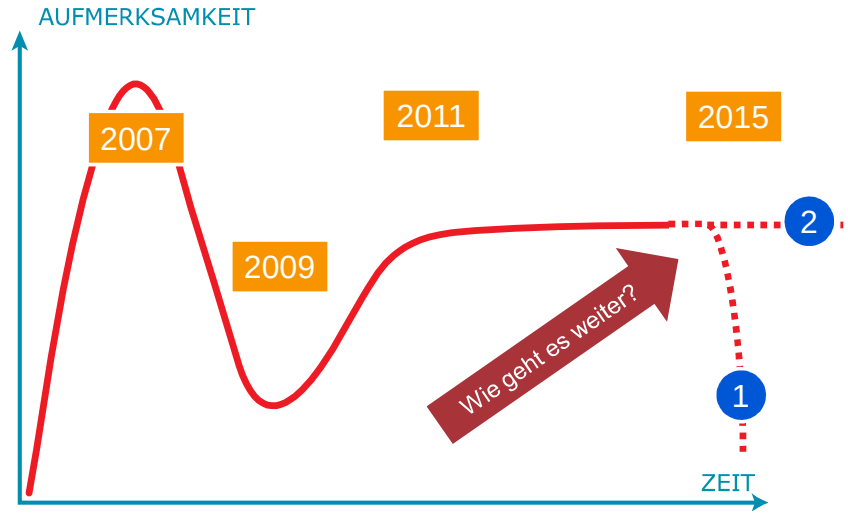
Orientation in Objects

| | Groovy | Grails |
|---|---|---|
| 2003 | Projektstart | |
| 2005 | | Projektstart |
| 2006 | 1.0 | |
| 2007 | G2One | |
| 2008 | | 1.0 |
| 2008 | SpringSource | |
| 2009 | VMware | |
| 2013 | Pivotal | |
| 2015 | ??? | ??? |



## Zeitliche Einordnung Groovy/Grails

Orientation in Objects



## Pivotal versucht gut aus der Sache rauszukommen

Orientation in Objects

Frist bis
**31.03.2015**

**Folge des Vert.x Desasters?**

http://www.heise.de/developer/meldung/VMware-beansprucht-Copyright-an-Vert-x-Projekt-1779511.html

**Schonfrist von knapp 3 Monaten**

**Unterstützung bei Sponsorsuche**

**Hosting von grails.org auch nach 31.03.**



## Rückzug von Pivotal: Auswirkungen auf Tools

Orientation in Objects

**Groovy/Grails Toolsuite (Eclipse) eingestellt**

aber weiterhin https://github.com/groovy/groovy-eclipse
Grails 3.0 benötigt kein spezielles Eclipse-Plugin mehr

**Gradle entwickelt jetzt eigenes Eclipse-Plugin**

## Wenn man sowieso am Boden liegt …

Orientation in Objects

- Codehaus schließt
- gestartet 2003
- 2015 der Übermacht von Github und Co. gebeugt
  - http://www.codehaus.org/history/
- Groovy braucht einen neuen Hosting Service
  - Source-Repo sowieso schon bei Github
  - aber Jira, Homepage, Wiki, ...

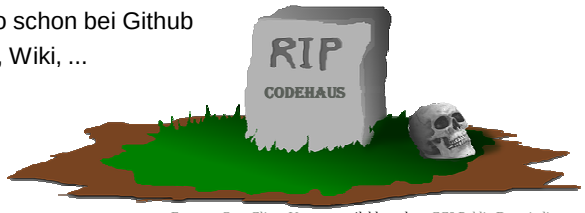



Foto von OpenClipartVectors, available under a CC0 Public Domain license.



## Groovy goes Apache

Orientation in Objects

**24.03.2015**

**Aufnahme im Inkubator**

- **5 Mentoren**
- **5 initiale Committer**
- **neue Mailinglisten**
- **Jira-Tickets verschoben**
- **neues Git-Repo**
- **weitere Committer**

By kOchstudiO [Public domain], via Wikimedia Commons

## Historie

| | Groovy | Grails |
|---|---|---|
| 2003 | Projektstart | |
| 2005 | | Projektstart |
| 2006 | 1.0 | |
| 2007 | G2One | |
| 2008 | | 1.0 |
| 2008 | SpringSource | |
| 2009 | VMware | |
| 2013 | Pivotal | |
| 2015 | 2.4 (Apache) | **???** |



## Grails has a New Home at OCI

OCI @ObjectComputing · 9. Apr.
We have announced some BIG news! @grailsframework and #Grails have a New Home at OCI!!
interact.stltoday.com/pr/business/PR …

OCI

"With the support and backing of OCI, an established IT solutions engineering firm, with deep roots in Open Source, Grails is moving forward with maximum momentum."

Object Computing, Inc.

St. Louis, MO

http://www.ociweb.com

## Historie

Orientation in Objects

| | Groovy | Grails |
|---|---|---|
| 2003 | Projektstart | |
| 2005 | | Projektstart |
| 2006 | 1.0 | |
| 2007 | G2One | |
| 2008 | | 1.0 |
| 2008 | SpringSource | |
| 2009 | VMware | |
| 2013 | Pivotal | |
| 2015 | 2.4 (Apache) | 3.0 (OCI) |



## Jeweils 3 Core Committer waren bei Pivotal

Orientation in Objects

## Who is Groovy?

Orientation in Objects

- interessante Statistik der Committer
- insgesamt 100+ Committer seit 2003
- Sieger: **Paul King** (nicht bezahlt)

http://melix.github.io/blog/2015/02/who-is-groovy.html



## Gliederung

Orientation in Objects

- Motivation + Politisches
- **Groovy**
- Grails
- Ausblick

## Glück gehabt …

2003 Projektstart Scala

2008 Programming in Scala

Orientation in Objects

> "I can honestly say if someone had shown me the **Programming in Scala** book by Martin Odersky, Lex Spoon & Bill Venners back in **2003** I'd probably have **never created Groovy**."
>
> James Strachan

http://macstrac.blogspot.de/2009/04/scala-as-long-term-replacement-for.html



## Dynamische Skriptsprache

Orientation in Objects

GROOVY

=

ausdrucksstarke Syntax

+

mächtige Bibliotheken

+

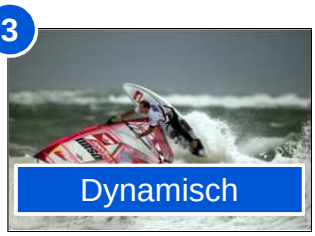
Meta-Programmierung

## Hauptprinzipien von Groovy

Orientation in Objects

1


3




2


4




## Kritik an Groovy

Orientation in Objects

~~schwache~~ dynamische Typisierung

fehlende Tool-Unterstützung (Refactoring)

Fehler erst zur Laufzeit

Performance

Ignorieren der Kapselung

Dynamische Typisierung ist ein Feature

ermöglicht Runtime-Metaprogrammierung

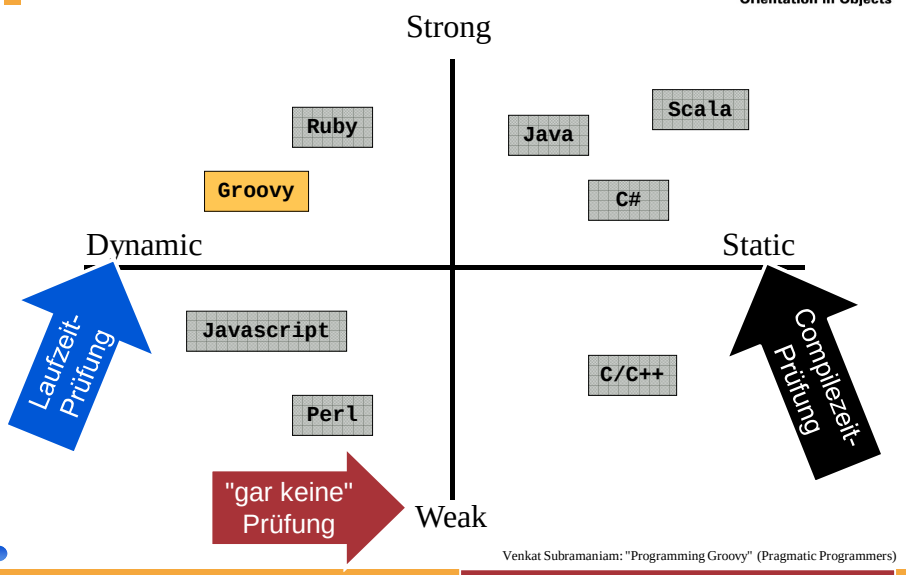
# Alternativen

- @TypeChecked und @CompileStatic
- AST-Transformation
- Traits





## Weakly != Dynamic Typing




Venkat Subramaniam: "Programming Groovy" (Pragmatic Programmers)

## An die Kette nehmen - Statische Typüberprüfung

Orientation in Objects

```
@TypeChecked
class MeineKlasse {
  def meineMethode1() {
     // nur statisch getypter Code erlaubt
  }

  @TypeChecked(TypeCheckingMode.SKIP)
  def meineMethode2() {
     // dynamisch getypter Code möglich
  }
}
```



## Fehler erst zur Laufzeit

Orientation in Objects

# Sicherheitsnetz durch Tests

## Chancen

- Syntax wunderbar geeignet
- Mocking-Framework eingebaut
- DSLs: Spock, Geb, …

## Performance: stetige Verbesserungen

Orientation in Objects

```
@TypeChecked
@CompileStatic
class MeineKlasse {
   [..]
}
```

| | Fibonacci | Pi quadrature | Binary trees |
|---|---|---|---|
| Java | 191 ms | 97 ms | 3.6 s |
| Groovy 2.x: Static Compilation | 197 ms | 101 ms | 4.3 s |
| Groovy 1.8: Primitive optimizations | 360 ms | 111 ms | 23,7 s |
| Groovy 1.7: No primitive optimizations | 2590 ms | 3220 ms | 50 s |



## Coole Groovy Features - Top 5

Orientation in Objects

Konzentration auf kleine, nützliche Funktionen

Keine Reihenfolge

Rein subjektiv

Warum nur 5?

## Meine Top 5 Groovy Features

Orientation in Objects

1. Multiline Strings/GStrings
2. Elvis Operator
3. Objektnavigation/Dereferenzieren
4. XyzSlurper/Parser
5. Power Asserts





## Releases Groovy

Orientation in Objects

| | |
|---|---|
| 2.1 | Januar 2013 |
| 2.2 | November 2013 |
| 2.3 | Mai 2014 |
| 2.4 | Februar 2015 |
| 3.0 | ??? |

## Groovy Neuerungen

Orientation in Objects

neue AST-Transformationen **2.x**

Traits **2.3**

Android Support **2.4**



## AST Transformationen

Orientation in Objects

"Built-in" Lombok (schon seit 1.6)

viele fertige

| | | |
|---|---|---|
| @Singleton | @Immutable | @Lazy |
| @TypeChecked | @Compilestatic | @Immutable |
| @Grab | @Synchronized | … |

selbst erweiterbar **2.x**

## Beispiel: @Builder

Orientation in Objects

```
@Builder
class Person {
    String firstName, lastName
    int age
}
def person = Person.builder()
                   .firstName("Dieter")
                   .lastName("Develop")
                   .age(21)
                   .build()

assert person.firstName == "Dieter"
assert person.lastName == "Develop"
```



## Traits

= Interface, mit …

Orientation in Objects

```
trait Fahrbar {
    int geschwindigkeit
    void fahren() {
        println "Fahren mit " +
            "${geschwindigkeit} km/h!"
    }
}

class Bobbycar implements Fahrbar {}
// 100 km/h
new Bobbycar(geschwindigkeit:100).fahren()
```

Zustand ~~Java 8~~

Verhalten Java 8

2.3

## Konflikte bei Mehrfachvererbung

Orientation in Objects

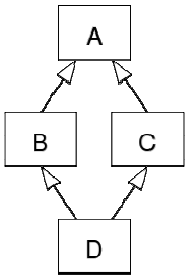
```
trait A {
    String exec() { 'A' }
}

trait B extends A {
    String exec() { 'B' }
}

trait C extends A {
    String exec() { 'C' }
}

class D implements B, C {}

def d = new D()
assert d.exec() == 'C'
```




Last wins!



## Manuelles Auflösen Mehrfachvererbung

Orientation in Objects

```
class D implements B, C {
    String exec() { B.super.exec() }
}
def d = new D()
assert d.exec() == 'B'
```

## Bauen von Android Apps mit Groovy

Orientation in Objects

schlanker weniger Boilerplate-Code als mit Java

prägnantere Syntax als Java

SwissKnife-Bibliothek (AST-Transformationen)

2.4



## Gliederung

Orientation in Objects

- Motivation + Politisches
- Groovy
- **Grails**
- Ausblick

## Groovy auf Schienen

Orientation in Objects

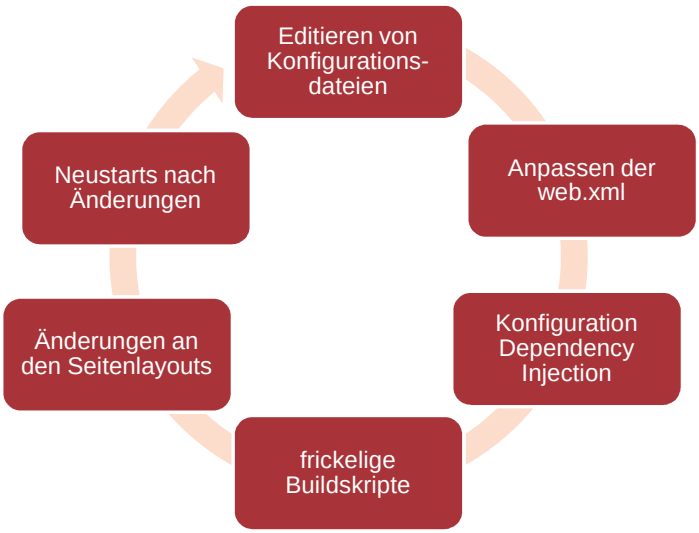
Aber mittlerweile eigenständiges, gestandenes Framework!

# Inspiriert durch Ruby on Rails



## Schmerzen klassische Java (Web) Frameworks

Orientation in Objects



## Hauptprinzipien von Grails

Orientation in Objects

1


3

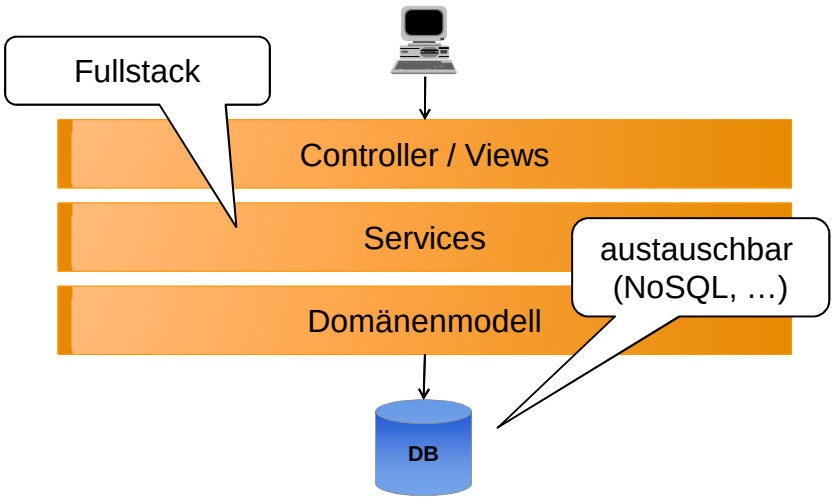

2


4

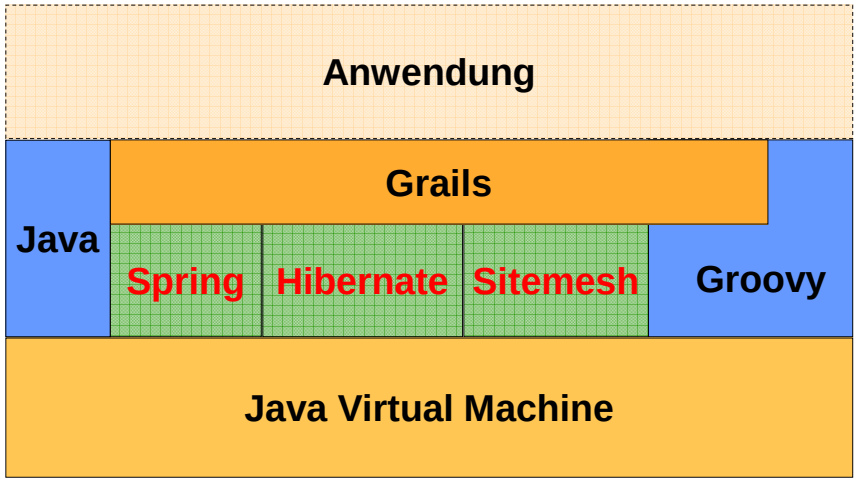



## Grails Schichtenmodell

Orientation in Objects



## Technologien in Grails






## Kritik an Grails

- Problematisch in großen Projekten
- Abwärtskompatibilität/Upgrades
- Aktualität der Plugins
- Stacktraces

## Große Projekte

Orientation in Objects

schwierig bei Misch-Masch
von Java und Groovy- Entwicklern

Festlegen projektübergreifender Code-Conventions

hohe Testabdeckung

80 % funktionieren in Grails super (CRUD, ...)
20 % sind umständlicher, aber nicht unlösbar

80/20 Regel



## Aufwand bei Upgrades

Orientation in Objects

leider ja

Aber es gibt einen Migrationsguide!

Möglichst frühzeitig upgraden!

## Aktualität der Plugins

Orientation in Objects

### Plugins der Core-Entwickler sind gut
- werden gewartet
- schnelle Upgrades bei neuen Grails-Versionen

### Hände weg von 3rd-Party-Plugins …
- mit vielen offenen Bugs
- ohne nennenswerte Aktivitäten

### Grails 3 macht einige Plugins obsolet



## Meine Top 5 Grails Features


1. Automatische DI
2. Validation
3. Tag Libs
4. CRUD-Methoden
5. Where Queries





## Releases Grails

- 2.2 Dezember 2012
- 2.3 September 2013
- 2.4 Mai 2014
- 2.5 / 3.0 März 2015
- 3.1 Herbst 2015

## Grails Neuerungen

Orientation in Objects

- Basis Spring Boot 3.0
- Interceptor API 3.0
- Anwendungsprofile 3.0
- Gradle als Buildsystem 3.0
- API Redesign mit Traits 3.0



## Basis Spring Boot

Orientation in Objects

- Spring 4.1 + Spring Boot 1.2

lauffähige JAR (Container eingebettet)

ohne Container start- und debugbar

keine IDE-Unterstützung notwendig

## Interceptor API löst Grails Filter ab


- Eigener Interceptor implementiert  Interceptor Trait
- 3 Methoden: before, after, afterView
  - vor der Controller-Action
  - nach Aufruf der Action
  - nach dem die View gerendert wurde
- Convention over Configuration: Namenskonvention
  - BookInterceptor für BookController



## Applikationsprofile


- ähnlich Java EE Profilen (Web, Full, ...)
- Profil kapselt die Anwendungsstruktur
  - Kommandos
  - Plugins
  - Skeletons, Templates
  - Ressourcen

```
grails create-app myapp --profile=web-plugin
```

- Default: Web-Profil
  - Projektstruktur für Webanwendung

## Applikationsprofile

Orientation in Objects

- Verwaltung in einem Repository (USER_HOME/.grails/repository)
- Profil = Verzeichnis mit folgender Struktur

```
web
    * commands
        * create-controller.yml
        * run-app.groovy
        …
    * skeleton
        * grails-app
            * controllers
            …
        * build.gradle
    * templates
        * artifacts
            * Controller.groovy
    * profile.yml
```



## Gradle als Build-System

Orientation in Objects

- Build-Management-Integration eines Grails-Projekts in eine Projektlandschaft war ziemliche Qual
- properitäres und fehleranfälliges Gant ist Geschichte
- Ivy ist Geschichte (eigener Dependency Resolver)
- keine IDE mit speziellen Grails-Plugins mehr nötig
  - nur Gradle-Support notwendig
  - theoretisch reichen die Commandline + Sublime/Atom/Vi/Emacs/...

## API-Redesign mit Traits

Orientation in Objects

Compiletime-Metaprogrammierung

Stabilität

weiterhin Flexibiltät

Schnittstellen aufgeräumt (grails.* vs. org.grails.*)



## Gliederung

Orientation in Objects

- Motivation + Politisches
- Groovy
- Grails
- **Ausblick**

Schon reif für den Java Framework Friedhof?






Aussage vom Groovy-Projektleiter


https://speakerdeck.com/glaforge/groovy-state-of-the-union-gr8conf-europe-2015






## Szenarien Groovy

Orientation in Objects

- Scripting
- Testen (Spock, Geb, …)
- DSLs (Gradle, …)
- Admin-Konsole (Java EE Apps)
- Grails

## Szenarien Grails

Orientation in Objects

- Prototyping
- (kleine) Intranetanwendungen
- Microservices



## Großes Ökosystem

Orientation in Objects

GRIFFON

GRAILS

GAELYK

GPARS

Spock

Groovy

Gradle
a better way to build

CODENARC
Less Bugs Better Code

Gant

Ratpack
A toolkit for JVM web applications

Geb

GVM

## Große Community

Orientation in Objects

Konferenzen

- GR8Conf Europe (Kopenhagen ) + GR8Conf US
- Greach (Spanien)

Podcasts

- http://groovypodcast.podbean.com/

Weekly Newsletter

- http://www.groovy-lang.org/groovy-weekly.html

Stackoverflow und aktive Mailinglisten

22,350
questions tagged
grails

Synonyms
groovygrails



## Roadmap Grails

Orientation in Objects

## Roadmap Grails - Last update

Orientation in Objects




## Roadmap nun offiziell

Orientation in Objects

**3.1**

- Profile-Support ausbauen – Repos, AngularJS-Profil
- REST-Support ausbauen – REST-Profil, JSON-Erweiterungen
- Hibernate 5 und MongoDB 3.0 in GORM

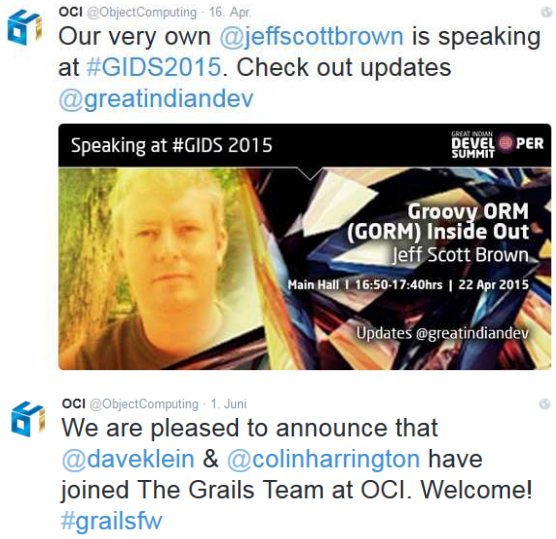
**3.2**

- Non Blocking IO
- Non Blocking GORM

**3.3**

- Hadoop YARN Profil
- HBase, Cassandra

## OCI gibt Gas




Sponsor und Auftritte auf Konferenzen

Anstellung von weiteren Leuten



## Ab auf die Überholspur …

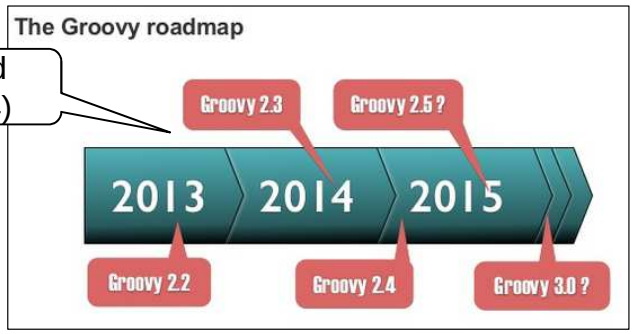

Sie suchen weiter Leute

## Roadmap Groovy

Orientation in Objects

- neues Meta-Object Protokoll
- Laufzeit auf Basis von Invoke Dynamic
- Sprachgrammatik neu in Antlr v4

Plan vor Abschied von Pivotal (2014)




http://de.slideshare.net/SpringCentral/groovy-in-2014andbeyond



## Prioritäten haben sich geändert

Orientation in Objects

**erstes Release nach Apache Richtlinien**

**• 2.4.4 vom 16.07.2015** 

**(enthält wichtiges Sicherheitsupdate)**

**kleinere Änderungen für 2.5 geplant**
**• AST-Transformationen**
**• GDK-Verbesserungen**

## Was passiert mit größeren Vorhaben?

Orientation in Objects

# Rewrite MOP in 3.0



## Können sich diese Firmen irren?

Orientation in Objects


http://groovy-lang.org/

## And the winner is Grails …

Orientation in Objects

Studie von ZeroTurnaround/RebelLabs von 2013/2014 ?

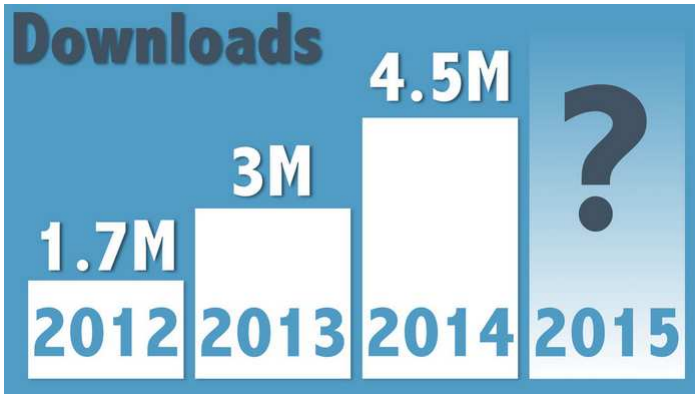
IN THE END
GRAILS & VAADIN
ARE THE BIG WINNERS,
FOLLOWED REASONABLY CLOSELY BY GWT, JSF, AND PLAY.
THERE YOU HAVE IT. NOW YOU CAN GO HOME ;-)

http://de.slideshare.net/hamedhatami2012/curious-coders-java-web-frameworks-comparison



## Zudem steigende ...

Orientation in Objects




https://speakerdeck.com/glaforge/groovys-history-and-current-status

## ... Downloadzahlen

Orientation in Objects

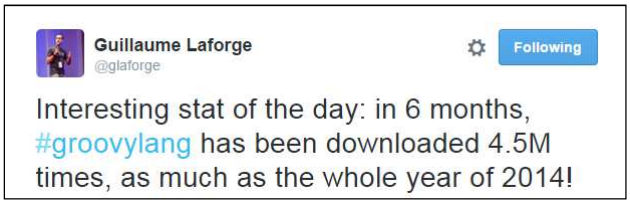
• nur Maven Central
• es fehlen Codehaus + Bintray


https://speakerdeck.com/glaforge/groovys-history-and-current-status



## Explodierende Downloadzahlen

Orientation in Objects





**JULI 2015**

**4,5 MILLIONEN DOWNLOADS**






oio
Orientation in Objects

# Fragen ?

**Orientation in Objects GmbH**

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Orientation in Objects

# Vielen Dank für ihre Aufmerksamkeit !

**Orientation in Objects GmbH**

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de