



Groovy und Grails Quo vadis?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

Über mich



Falk Sippach

Trainer, Berater, Entwickler



Schwerpunkte
Architektur
Agile Softwareentwicklung
Codequalität

Java und XML

) 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, unternehmensweiten Systemen

Abstract

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



- Motivation
- Groovy
- Grails
- Politisches
- Ausblick

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 5 |

Bitte melden!

Wer arbeitet mit Groovy/Grails?

regelmäßig/häufig

selten/früher mal

noch nie

Enthaltungen?



Foto von Markgraf-Ave., available under a [CC0 Public Domain license](#).

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 6 |

Warum Groovy und Grails?



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



Foto von alankotok., available under a [CC0 Public Domain license](#).

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 7 |

Aber es ist ruhig geworden ...



Gefühlt tendiert die öffentliche
Aufmerksamkeit **gegen Null!**

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 8 |

Zu alt?



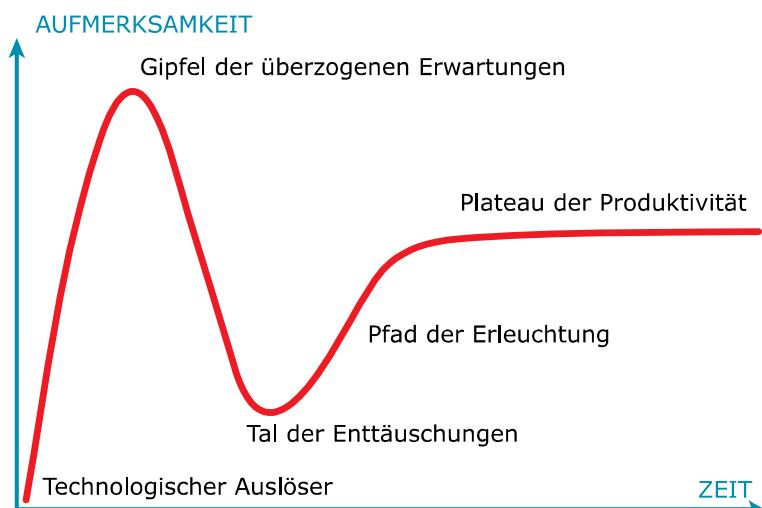
Sind Groovy und Grails nicht mehr hip genug?

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 9 |

Gardners Hype Kurve



Grafik von Idotter, available under a Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) license.

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 10 |

Gut Ding will Weile haben: Regina



Who the fuck is Regina?

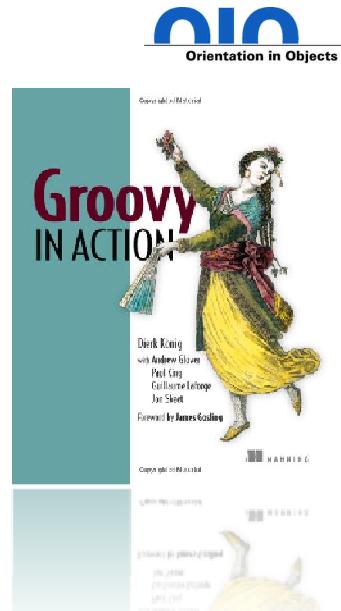
© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 11 |

Gehen wir zurück ins Jahr 2007

GinA



© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 12 |

Zwei Jahre später: viele Neuerungen mit 1.5 - 1.7



Planung einer zweiten Auflage

reGinA war geboren

und eine lange "Leidensgeschichte" begann

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 13 |

Anfänglicher Optimismus



*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 your understanding
Dierk*

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

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 14 |

Twitter vergisst nichts: #regina #groovy(lang)



nicht so
einfach ...

2009/2010

Dierk König (@mittie · 22 Aug 2009)
soooo many new methods in #groovy gdk since 1.0. Impossible to explain them all in #regina. thinking...

Dierk König (@mittie · 20 Aug 2009)
I guess I have to rewrite the #groovy type system section in #regina a third time - tomorrow. "sigh"

gt groovytweets (@groovytweets · 12 Jan 2010)
RT @the_very: Fresh MEAP update, yahoo! Groovy in Action, Second Edition - Chapters 1-6 is out! #regina #groovy

Dierk König (@mittie · 18 Apr 2010)
#groovy in action 2nd ed will now cover groovy 1.8 (due in oct/nov 2010). #regina (via @wmagyver) dec 2010 I would say...

weiter
geht's ...

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 15 |

Time flies by ...



2011

Guillaume Laforge (@glaforge · 31 Mar 2011)
@mittie and #regina's #groovy DSL chapter is now at 13 pages, but I feel like I'll need at least twice or three times as much :-)
View conversation

Dierk König (@mittie · 31 Mar 2011)
#regina MOP chapter draft is now at 22 pages. ~5 more to go. Simple explanations are sometimes difficult to find. #groovy

Guillaume Laforge (@glaforge · 29 Aug 2011)
Happy to have finished my first draft of the #groovy #DSL chapter for #regina. Should be on MEAP in days or weeks!

Erik Pragt (@repragt · 30 Nov 2011)
Please RT if you also would like #Groovy in Action 2 to be ready for Christmas. #regina I'll give out 5 books among RT-ers!

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 16 |

Passiert hier was?



2012

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 17 |

Da passiert was ...



2013

Dierk König @mittrie - 21 Mar 2013
#ReGina MEAP update is available, incl. mobile versions. Thx to @ManningBooks and my awesome crew of coauthors. #groovy

Wolfgang Schell @jetzigradnet - 21 Mar 2013
Welcome back, #reGINA, long time no see! Thanks for the update and finally with mobile formats, yeay! (@mitrie, @ManningBooks) #Groovy

Geschäftstüchtig

Erik Pragt @epragt - 26 May 2013
Deal of the Day May 26: Half off our book #Groovy in Action, Second Edition. Use code ddot0526au at manning.com/koenig2/ #regina

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 18 |

Weitere Schaffenspause ...



2014

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 19 |

Hurra – endlich ist es da!



2015

Manning Publications favorite

 **Jeff Gortatowsky** @JeffGortatowsky · Jun 29
Should I believe in god? #ReGina aka #Groovy in Action 2nd edition is here . I long ago forgotten I'd ordered it. #recommended @ManningBooks

GR8Forums.org and 5 others follow

 **John Rellis** @Johnrellis · Jun 25
How many years in the making???? Woot, finally final! Congratulations to the authors! #groovy #regina



[View photo](#)

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 20 |

Selfies



Sergey Tselovalnikov @SerCeMan · 16h
#regina selfie. Got main #groovy book after year of waiting.:) May the groovy force with you!



View photo

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 21 |

MEAP Auslieferungen



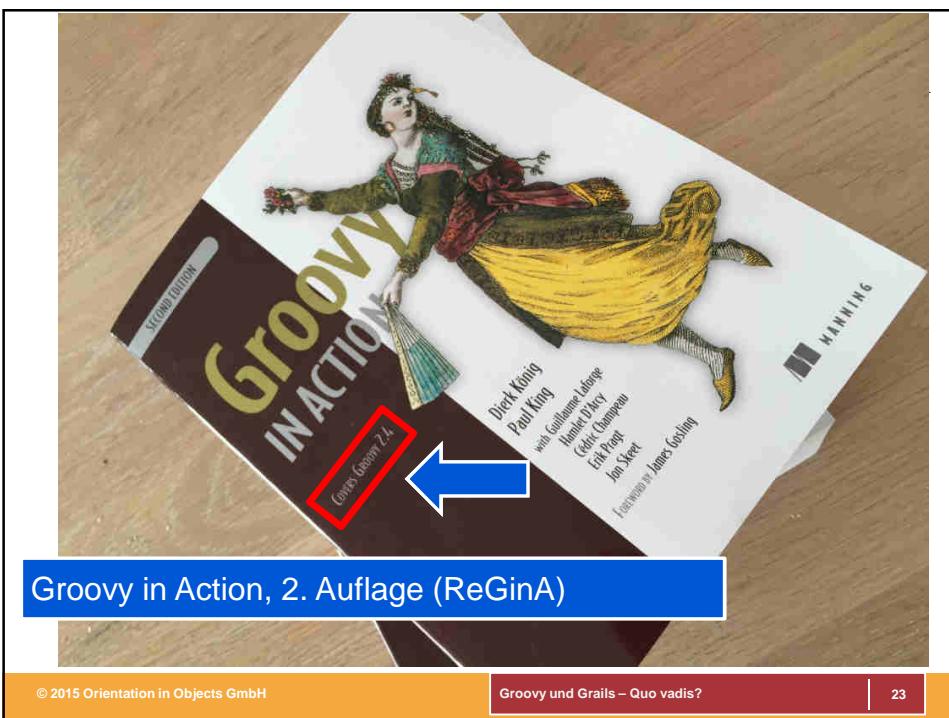
 GroovyinAction,SecondEdition	29.06.2011 13:16
 GroovyinAction,SecondEdition2	01.08.2011 08:06
 GroovyinAction,SecondEditionDezember	13.12.2011 09:04
 Groovy_in_Action_Se_v11_MEAP	13.02.2012 09:55
 Groovy_in_Action_Se_v12_MEAP	21.03.2013 15:17
 Groovy_in_Action_Se_v13_MEAP	10.05.2013 09:05
 Groovy_in_Action_Se_v14_MEAP	05.06.2013 13:41
 Groovy_in_Action_Se_v15_MEAP	12.07.2013 13:22
 Groovy_in_Action_Se_v16_MEAP	28.08.2013 08:20
 Groovy_in_Action_Se_v17_MEAP	21.11.2013 16:03
 Groovy_in_Action_Se_v18_MEAP	24.03.2015 10:28

 GroovyinAction,SecondEdition	29.06.2011 13:16
 GroovyinAction,SecondEdition2	01.08.2011 08:06
 GroovyinAction,SecondEditionDezember	13.12.2011 09:04
 Groovy_in_Action_Se_v11_MEAP	13.02.2012 09:55
 Groovy_in_Action_Se_v12_MEAP	21.03.2013 15:17
 Groovy_in_Action_Se_v13_MEAP	10.05.2013 09:05
 Groovy_in_Action_Se_v14_MEAP	05.06.2013 13:41
 Groovy_in_Action_Se_v15_MEAP	12.07.2013 13:22
 Groovy_in_Action_Se_v16_MEAP	28.08.2013 08:20
 Groovy_in_Action_Se_v17_MEAP	21.11.2013 16:03
 Groovy_in_Action_Se_v18_MEAP	24.03.2015 10:28

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

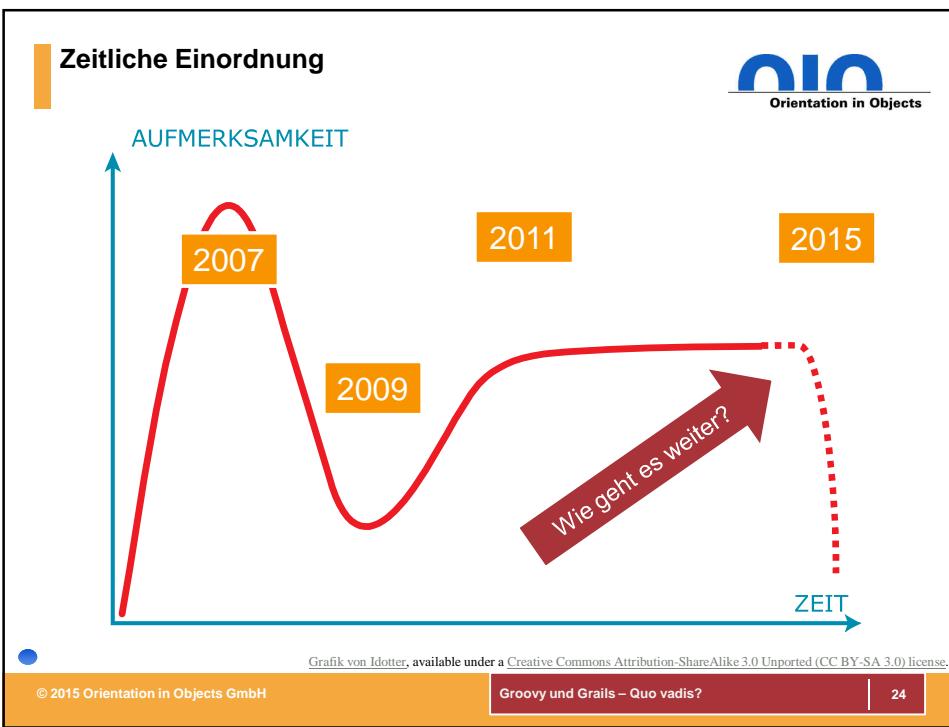
| 22 |



© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 23



Historie



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

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 25

About being a paid OSS Developer for Groovy (1)



Groovy is my Child in code you could say.

For over 10 years I helped the project. I was one of the first guys to be paid [...] and did so for about 9 years.

[...] nice experience. I mean you do your job, you help the community and work on your hobby at the same time!

Ideal, or not?

Well, not ideal.



<http://blackdragsview.blogspot.de/2015/04/about-being-paid-oss-developer-for.html>

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 26

About being a paid OSS Developer for Groovy (2)



G2One

[...] the workload was high. [...] adding feature after feature without testing or documentation.

SpringSource

A few liked us, a few hated us, most ignored us.
[...] also no real plan of continuation.

<http://blackdragsview.blogspot.de/2015/04/about-being-paid-oss-developer-for.html>

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 27

About being a paid OSS Developer for Groovy (3)



VMware

[...] somewhere deep down in the management chain before, now we have been even deeper down.

[...] pressure management as best [...] finally got 2013 Cedric on board.

Pivotal

Still I stayed a bit more happy. [...] being now Pivotal and them wanting to be so open-source-oriented [...]

<http://blackdragsview.blogspot.de/2015/04/about-being-paid-oss-developer-for.html>

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 28

About being a paid OSS Developer for Groovy (4)



Pivotal

Well, turns out Pivotal cares only about OSS if they can use it. [...] seeing people only as cost factors and not as human beings.

I expect several more OSS related problems with Pivotal in the future.

So for the time being it looks like there is no chance for me getting fulltime on Groovy.

I dare the community to fill the gaps and show they can do serious stuff in their free time.

<http://blackdragsview.blogspot.de/2015/04/about-being-paid-oss-developer-for.html>

Gliederung



- Motivation
- Groovy
- Grails
- Politisches
- Ausblick

A Groovy kind of love!



I am married to Java, but i have a lover ... Groovy
Java has been there for me many years.
Solid, stable, reliable, mature, old ...
I know java well, i think she knows me too.
Then I met Groovy. Young, sexy, dynamic, agile ...

I am getting to know Groovy,
I am getting butterflies in my stomach again.
The best things is that they get along just fine.
They even talk together in harmony.
What else can I say?

http://www.jroller.com/orrego/entry/i_am_married_to_java

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 31

Glück gehabt ...



2003 Projektstart, Buch 2008

"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>

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 32

Dynamische Skriptsprache



=

ausdrucksstarke Syntax

+

mächtige Bibliotheken

+

Meta-Programmierung

Hauptprinzipien von Groovy



1

Featurereich



3

Dynamisch



2

Java-freundlich



4

Robuste Plattform

Featurereich



- Closures (Lambda++), funktionale Programmierung **light**
- ausdrucksstarke Syntax (Collections, Reguläre Ausdrücke, GString, mehrzeilige Strings, ...)
- Out-of-the-box Import/Export von XML, JSON, ...
- AST-Transformationen
- Traits (Mehrfachvererbung)
- Skripting

1



Foto von 777546, available under a CC0 Public Domain license.

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

35

Java-freundlich



- Groovy kompiliert zu Java Bytecode
- flache Lernkurve
- leichte Migration (Groovy ist Superset der Java-Syntax)
- gegenseitig Klassen aufrufen, voneinander erben, ...
- GDK (Erweiterung des JDK)
- Android-Anwendungen in Groovy schreiben

2



Foto von 873770, available under a CC0 Public Domain license.

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

36

Dynamisch



- Property Support (getter/setter for free)
- Meta-Programmierung (Laufzeit-AOP)
- AST-Transformationen (Compilezeit-AOP)
- Builder (XML, JSON, Swing, ...)
- Überladen von Operatoren
- Erstellen von DSLs

3



Foto von WikiImages, available under a CCO Public Domain license.

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 37

Robuste Plattform



- kompiliert zu Bytecode, läuft auf der JVM
- JDK und Java-Bibliotheken nutzen
- gleiches Memory-, Threading-, Vererbungs- und Sicherheitsmodell
- Werkzeuge wiederverwendbar (Build-Tools, IDEs, Profiler, Debugger, ...)

4



Foto von PublicDomainPictures, available under a CCO Public Domain license.

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 38

Kritik an Groovy



- schwache dynamische Typisierung
- fehlende Tool-Unterstützung (Refactoring)
- Fehler erst zur Laufzeit
- Performance
- public Fields



Foto von ashishacoway, available under a [CC0 Public Domain license](#).

Dynamische Typisierung ist ein Feature



- ermöglicht Runtime-Metaprogrammierung

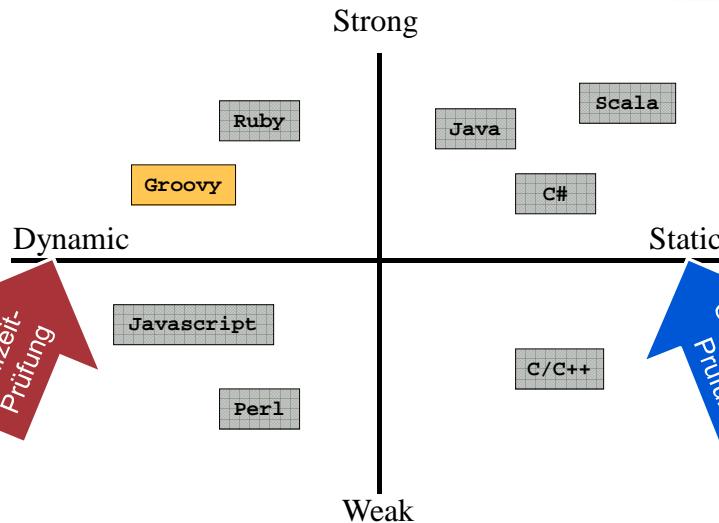
Alternativen:

- @TypeChecked und @CompileStatic
- AST-Transformation: Completetime-Metaprogrammierung
- Traits



Foto von Unsplash, available under a [CC0 Public Domain license](#).

Weakly != Dynamic Typing



Venkat Subramaniam: "Programming Groovy" (Pragmatic Programmers)

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 41

An die Kette nehmen - Statische Typüberprüfung



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

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

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 42

Fehler erst zur Laufzeit



- Erinnerung: dynamische Typisierung ist ein Feature für die Runtime-Metaprogrammierung
- Sicherheitsnetz durch Testen
 - agile Softwareentwicklung, TDD, BDD, ...
 - sowieso notwendig fürs Refactoring
- Syntax eignet sich wunderbar zum Schreiben von Tests
- Mocking-Framework eingebaut
 - Unit-Tests einfach isolieren dank Metaprogrammierung
- DSLs: Spock, Geb, ...

Performance: stetige Verbesserungen



```
@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

Public Fields



- Lange Diskussion, bisher keine Lösung:
 - It's not a bug, it's a feature.
 - Groovy nutzt das intern für seine Unit-Tests

It's not clear if it is a bug or by design.
Groovy treats visibility keywords as a hint for a programmer.

- <https://issues.apache.org/jira/browse/GROOVY-1875>
- <http://refaktor.blogspot.de/2012/07/private-fields-and-methods-are-not.html>

Coole Groovy Features - Top 5



Konzentration auf kleine, nützliche Funktionen

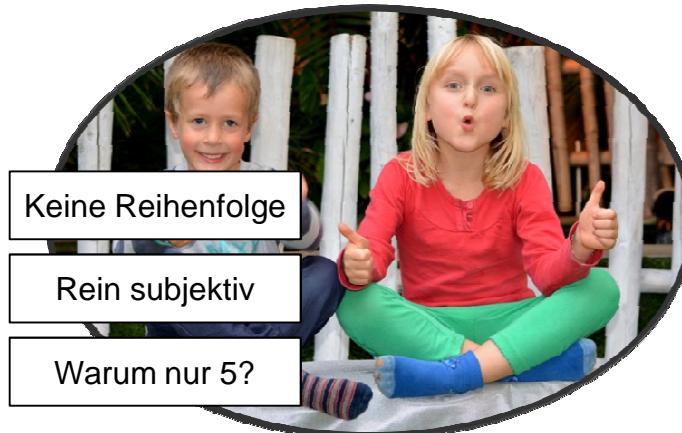


Foto von Ben Kerckx, available under a [CC0 Public Domain license](#).

Meine Top 5 Groovy Features



① Multiline Strings/GStrings

② Elvis Operator

③ Objektnavigation/Dereferenzieren

④ XyzSlurper/Parser

⑤ Power Asserts



Foto von Ben Kerckx, available under a [CC0 Public Domain license](#).

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

47

GStrings



```
String s1 = "Das ist GString!"  
  
String ersetzung = 'Ersetzungen'  
String s2 = "Ein GString kann $ersetzung  
enthalten!"  
  
println "Hallo ${user.name}!"
```

1

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

48

Slashy Syntax



```
String s1 = /\d{1,2}.\d{1,2}.\d{2,4}/  
  
java.util.regex.Pattern.compile(/\s+w+\s/)  
  
def dir = /c:\Schulung\workspace/
```

1

Mehrzeilige Strings



```
String s1 = '''Das ist  
ein besonders  
langer String'''  
  
String s2 = """Dieser GString  
reicht ebenfalls  
über mehrere Zeilen"""
```

1

Von if/else zum ternären zum Elvis Operator



```
def username = getUsernameFromAnywhere()

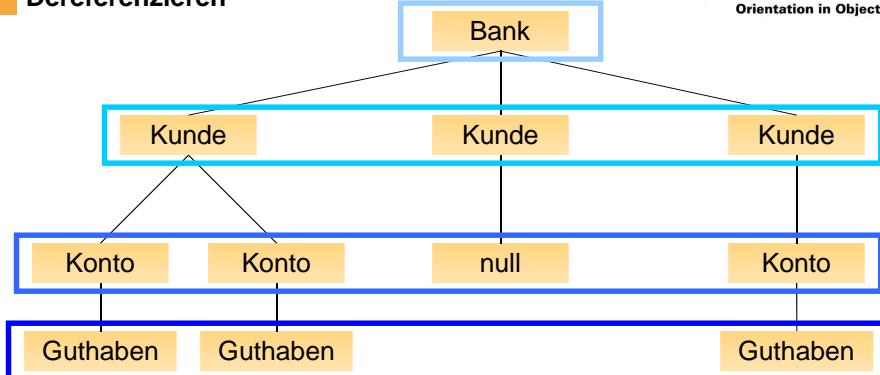
if (username) { username } else { 'Unknown' }

username ? username : 'Unknown'

username ?: 'Unknown'
```

2

Effiziente Objektnavigation und sicheres Dereferenzieren



- Summe der Guthaben aller Konten der Bank:
bank.kunden.konten ?.guthaben .sum()

3

JSON einlesen



```
def jsonSlurper = new JsonSlurper()

def object = jsonSlurper.parseText(
    '{ "name": "John Doe" } /* comment */')

assert object instanceof Map

assert object.name == 'John Doe'
```

4

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 53

JSON erzeugen



```
class Author {
    String name
}

def authors = [new Author (name: "Guillaume"),
    new Author (name: "Jochen"),
    new Author (name: "Paul")]

def json = new JsonBuilder()
json authors, { Author author ->
    name author.name
}
assert json.toString() ==
    '[{"name": "Guillaume"}, {"name": "Jochen"}, {"name": "Paul"}]'
```

4

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 54

Java Asserts



```
a = 10  
b = 9  
  
assert 91 == a * b  
  
Exception thrown: Expression: (92 == (a * b)).  
Values: a = 10, b = 9  
  
java.lang.AssertionError: Expression: (92 == (a *  
b)). Values: a = 10, b = 9  
at ConsoleScript2.run(ConsoleScript2:4)
```

5

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 55

Power Asserts



```
a = 10  
b = 9  
  
assert 91 == a * b  
  
Exception thrown  
Assertion failed:  
assert 91 == a * b  
| | | |  
| 10 | 9  
| 90  
false  
  
at ConsoleScript2.run(ConsoleScript2:4)
```

5

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 56

Coole Features von Groovy - viele, viele mehr ...



- Multiple Assignments
- NullObjectPattern
- Spaceship-Operator
- diverse GDK-Methoden und -Operatoren
- diverse AST-Transformationen
- dynamische Method-Interception
- Default-Parameter in Methoden
-

- siehe: <http://stackoverflow.com/questions/303512/hidden-features-of-groovy>

Releases Groovy



- ...
- Januar 2013: 2.1
- November 2013: 2.2
- Mai 2014: 2.3
- Februar 2015: 2.4

Groovy Neuerungen



- diverse neue AST-Transformationen (2.3, 2.4)
- Traits (Groovy 2.3)
- Android Support (Groovy 2.4)

AST Transformationen



- "Build-in" Lombok
- erweiterbar durch eigene AST-Transformationen
- viele fertige
 - @Singleton, @Immutable, @Lazy
 - @Delegate, @Newify, @Category, @Mixin
 - @PackageScope
 - @Bindable, @Vetoable

AST Transformation @Builder



```
@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"
```

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 61

Traits



- Interfaces mit Default Implementierungen und Zustand

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

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

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

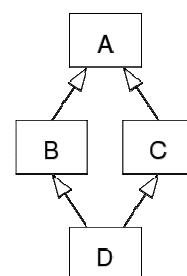
| 62

Abstrakte Methoden

```
trait Greetable {  
    abstract String name()  
    String greeting() {  
        "Hello, ${name()}!"  
    }  
}  
  
class Person implements Greetable {  
    String name() { 'Duke' }  
}  
  
def p = new Person()  
assert p.greeting() == 'Hello, Duke!'
```

Konflikte bei Mehrfachvererbung

```
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



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

Implementierung von Traits zur Laufzeit



```
trait Extra {  
    String extra() { "I'm an extra method" }  
}  
class Something {  
    String doSomething() { 'Something' }  
}  
  
def s = new Something() as Extra  
s.extra()  
s.doSomething()
```

SAM Type Coercion



```
trait Greeter {  
    String greet() { "Hello $name" }  
    abstract String getName()  
}  
  
Greeter greeter = { 'Alice' }  
println greeter.greet()  
  
// oder  
  
void greet(Greeter g) { println g.greet() }  
println greet { 'Alice' }
```

Single Abstract Method

Closure erzeugt SAM Type.

Android Support



- Bauen von Android Apps mit Groovy
- SwissKnife-Bibliothek
- minimiert den Android-Boilerplate-Code (AST-Transformationen)
 - Behandeln von UI-Events
 - Background-Threads

Gliederung



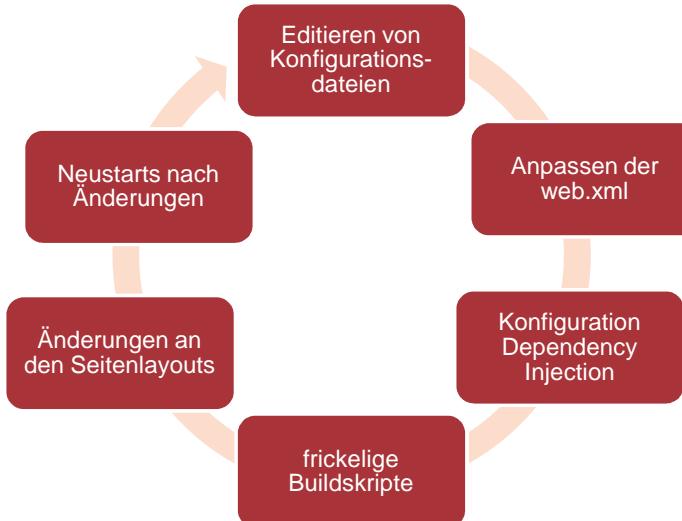
- Motivation
- Groovy
- **Grails**
- Politisches
- Ausblick

Groovy auf Schienen



Inspiriert durch
Ruby on Rails

Schmerzen mit klassischen Java Web Frameworks



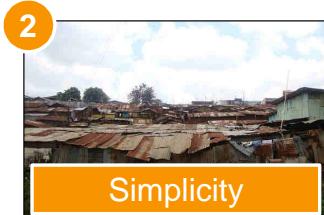
Hauptprinzipien von Grails



Produktivität



Erweiterbarkeit



Simplicity



Stabile Basis

Produktivität



Convention
over
Configuration

Don't repeat
yourself

Prototyping

1



Einfachheit



- Groovy
- GORM – groovyfied Hibernate
- Spring (MVC) unter der Haube
- wohldefinierte Projektstruktur
- keine unnötige Konfiguration
- KISS (Design Komplexität)
- Vermeiden von Boilerplate Code

2



Erweiterbarkeit



- Grails modular aufgebaut, alles austauschbar
 - Hibernate/RDBMS ⇔ NoSQL
 - Tomcat ⇔ Jetty
- Funktionalität über Plugins hinzufügbar
 - Core-Plugins
 - Community-Plugins
- modularer Aufbau von Grails-Applikationen über Plugins
 - bessere Wiederverwendung

3



Stabile Basis



basiert auf Java EE Standards

Integration von Spring,
Hibernate, Sitemesh

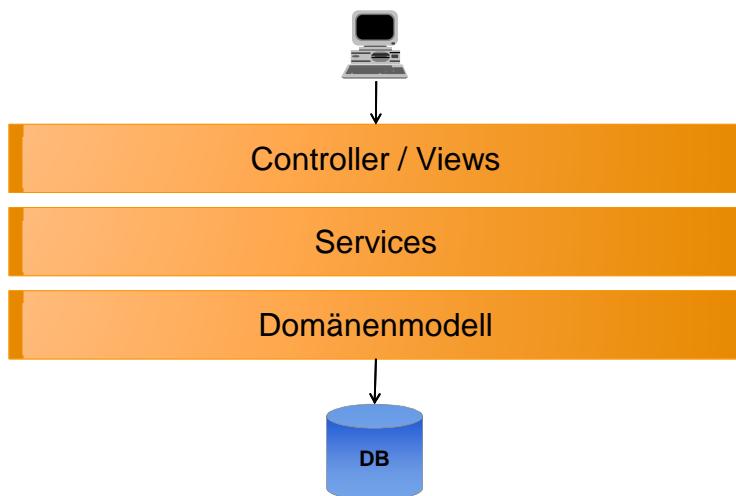
Ausführungsumgebung dabei

4

Fullstack



Grails Schichtenmodell

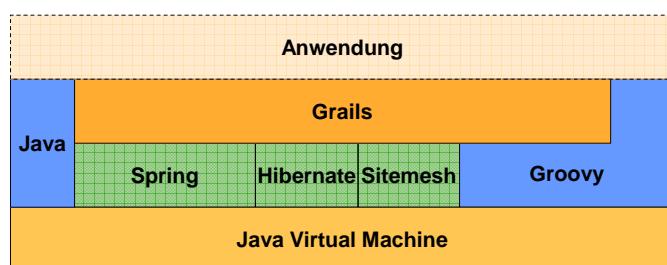


© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 77 |

Technologien in Grails



© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 78 |

Kritik an Grails



- problematisch in großen Projekten
- Abwärtskompatibilität, hoher Aufwand bei Upgrades
- Aktualität der Plugins
- Stacktraces

Große Projekte



- schwierig bei Misch-Masch von Java und Groovy-Entwicklern
- zwingend projektübergreifende Code-Conventions festlegen
 - werden Typen ausgeschrieben?
 - benutzen wir return
 - ...
- gute Testabdeckung!
 - sonst gibt es wenig Chancen auf Refactoring
- 80/20 Regel
 - 80 % funktionieren in Grails super (CRUD, ...)
 - 20 % sind umständlicher, aber nicht unlösbar

Aufwand bei Upgrades



- leider ja
- aber es gibt einen Migrationsguide in der Grails-Dokumentation
- möglichst alle neuen Release-Sprünge sofort upgraden, nicht zu lange warten

Aktualität der Plugins



- Plugins der Core-Entwickler sind super
 - werden gewartet
 - bei neuen Grails-Versionen erfolgt schnell ein Upgrade
- Vorsicht bei kleinen Plugins ohne nennenswerte Aktivitäten mit vielen Bugs, die vor 3 Jahren eingestellt wurden
- mit Grails 3 sind einige Plugins obsolet geworden



Meine Top 5 Grails Features

A photograph of a person's hand giving a thumbs-up gesture. To the right of the hand, the word "Like" is written in a large, light-colored, chalky font on a dark surface. Above the hand is a large green five-pointed star.

① Automatische DI

② Validation

③ Tag Libs

④ CRUD-Methoden

⑤ Where Queries

OIO
Orientation in Objects

Foto von [gerallt](#), available under a [CC0 Public Domain license](#).

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

84

Automatische Dependency Injection von ...



- ... Service in Services, Controller, TagLib, Domain-Klassen
- ... Logger in Controller, Service, TagLib
- über Spring-Konfiguration manuell steuerbar

```
class BuchController {  
    BuchService buchService  
}  
  
Class User {  
    def securityService  
    static transients = ['securityService']  
}
```

1

Implizite Attribute im ...



- ... Controller
- ... View
- ... TagLib
- ... Service

```
class BuchController {  
    String action() {  
        log.info(params)  
        servletContext.getResourceAsStream( "...")  
    }  
}
```

1

Validierung



```
class User {  
    ...  
  
    static constraints = {  
        login size: 5..15, blank: false, unique: true  
        password size: 5..15, blank: false  
        email email: true, blank: false  
        age min: 18  
    }  
}
```

2

Übersicht Validatoren



Name	Beschreibung
blank	String darf nicht leer sein
email	String muss eine syntaktisch korrekte Email-Adresse sein
inList	Wert muss in angegebener Collection enthalten sein
matches	String muss auf regulären Ausdruck matchen
max	Wert darf Maximalwert nicht überschreiten
maxSize	size des Werts darf Maximalgröße nicht überschreiten
nullable	Wert darf null sein. Default: false
range	Wert muss ist in angegebener Range enthalten sein
size	size des Werts muss ist in angegebener Range enthalten sein
unique	Wert muss in Datenbank einzigartig sein
url	String muss eine syntaktisch korrekte URL sein
validator	Eigene Validierungslogik anwenden

2

Beispiel für eigenen Validator



```
class KalenderEintrag {  
    Date von  
    Date bis  
  
    static constraints = {  
        von(nullable: true)  
        bis(nullable: true, validator: bisDatumValidator)  
    }  
  
    private static Closure bisDatumValidator = { val, obj ->  
        return (!obj.von || val?.after(obj.von))  
    }  
}
```

2

Dynamic Tag Libraries



```
class MyTagLib {  
    static namespace = "oio"  
  
    def greeting = { attrs, body ->  
        out << "Hallo ${attrs.name}"  
    }  
  
    def isAdmin = { attrs, body ->  
        def user = attrs['user']  
        if(user != null && checkUserPrivils(user)) {  
            out << body()  
        }  
    }  
}
```

3

Dynamic Tag Libraries – Beispiel in GSP



```
<oio:greeting name="Fritz" />  
  
<oio:isAdmin user="${myUser}">  
    // Nur fuer Admin sichtbar  
</oio:isAdmin>
```

3

CRUD-Methoden von Domänenklassen



- Dynamisch hinzugefügte Methoden für CRUD-Operationen
 - get(id) lädt ein Objekt aus der DB
 - list() lädt alle/viele Objekte aus der DB
 - save() speichert das Objekt in die DB (initial oder update)
 - delete() löscht ein Objekt aus der DB
 - exists(id) prüft, ob Objekt existiert
 - count() zählt die Objekte in der DB

4

CRUD-Methoden von Domänenklassen - Beispiel



```
Buch buch = new Buch(titel:'Groovy')
buch.save()
assert 1 == Buch.count()

buch.titel = 'Groovy in Action'
buch.save()

buch.delete()
assert !Buch.exists(buch.id)
```

4

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

93

Where Queries



- Setzt auf Detached Criteria auf
- Compile-time checked query DSL
- „Entwickler-freundliche“ Syntax
- IDE support / code completion
- Groovy Operatoren werden auf Restrictions gemappt

```
def query = Buch.where {
    titel == "Grails in Action"
}
Buch grailsInAction = query.find()
```

5

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

94

Gegenüberstellung Criteria Queries vs. Where Queries



```
// select * from Buch where titel like '%grails%'  
  
// Criteria Builder  
Buch.createCriteria().list {  
    ilike('titel', '%grails%')  
    authors {  
        ilike('nachname', 'r%')  
    }  
}  
  
// Where Query  
Book.findAll {  
    title =~ '%grails%' && authors.lastName =~ 'r%'  
}
```

5

Gegenüberstellung Criteria Queries vs. Where Queries II



```
// Criteria Builder  
Book.createCriteria().list {  
    ilike('title', '%grails%')  
    maxResults(1)  
    firstResult(1)  
    order('title', 'asc')  
}  
  
// Where Query  
Book.findAll(max:1, offset:1, sort:'title', order:'asc') {  
    title =~ '%grails%'  
}
```

5

Releases Grails



- ...
- Dezember 2012: 2.2
- September 2013: 2.3
- Mai 2014: 2.4
- März 2015: 2.5, 3.0

Grails Neuerungen



- 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



- Spring 4.1
- Spring Boot 1.2
- Grails enthält nun eine Klasse mit main-Methode
 - lauffähige JAR-Datei (eingebetteter Jetty, Tomcat, Undertow)
 - ohne Container start-/debugbar
 - keine spezielle IDE-Unterstützung für Webcontainer bzw. Grails

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 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
 - hilft beim Erstellen einer Webanwendung

Applikationsprofile



- 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



- Build-Management-Integration eines Grails-Projekts in eine Projektlandschaft war ziemliche Qual
- properitäres und buggy 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



- Grails Metaprogrammierung hat auch manche Probleme gemacht
- Traits werden vom Compiler verarbeitet (keine Runtime-Metaprogrammierung)
- vieles der Grails-Magie mittlerweile über Traits => Stabilität
 - Domain-Klassen
 - Controller
 - Services
 - Taglibs
- Schnittstellen aufgeräumt
 - öffentliche unter grails.*
 - interne jetzt unter org.grails.*

Gliederung



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

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 105

Pivotal fokussiert sich – 19.01.2015



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

31.03.2015

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>

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 106

Pivotal versucht gut aus der Sache
rauszukommen



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.

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 107

Es fehlt was ...



Álvaro Sánchez @alvaro_sanchez · Apr 29
ow.ly/lazPj the box is now more empty... #groovylang #grailstw #springio15

Guillaume Laforge @glaforgo · Apr 29
@alvaro_sanchez that was my own graphics... But there are like holes now... Go figure... 😕

Josh Long (龙之春)
@starbuxman

Following

@glaforgo @alvaro_sanchez that hurt when I had to edit it. I still wear the Pivotal multi logo t-shirt. Sorry mon ami

12:51 PM - 29 Apr 2015

<https://twitter.com/starbuxman/status/593366929550643200?s=03>



© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 108

One third and one G-Teams



© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 109

Jeweils 3 Core Committer bei Pivotal



Grafik von ClkerFreeVectorImages , available under a CCO Public Domain license.

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 110

Who is Groovy?



- 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>

Rückzug von Pivotal: Auswirkungen auf Tools



- Groovy/Grails Toolsuite (Eclipse) eingestellt
 - weiterhin <https://github.com/groovy/groovy-eclipse>
 - Grails 3.0 benötigt kein spezielles Eclipse-Plugin mehr
- Gradle entwickelt jetzt eigenes Eclipse-Plugin

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, ...

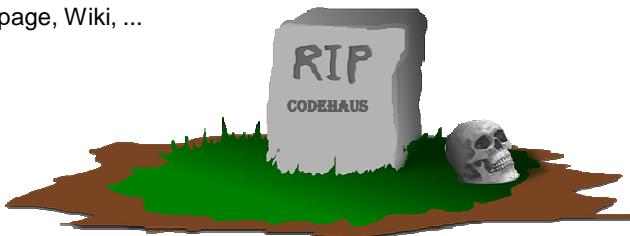


Foto von OpenClipartVectors, available under a CCO Public Domain license.

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

113

Groovy goes Apache



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

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

114

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>

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 115

OCI gibt Gas



oci @ObjectComputing · 16. Apr

Our very own @jeffscottbrown is speaking
at #GIDS2015. Check out updates
@greatindiandev



Sponsor und Auftritte
auf Konferenzen

oci @ObjectComputing · 1. Juni

We are pleased to announce that
@daveklein & @colinharrington have
joined The Grails Team at OCI. Welcome!
#grailsfw

Weitere
Einstellungen

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 116

Gliederung

- Motivation
- Groovy
- Grails
- Politisches
- Ausblick



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



MESSAGE TO THE COMMUNITY

HERE TO STAY
INDEPENDENCE

COMMUNITY ABOVE ALL

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

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

119



Stärken und Grenzen von Groovy und Grails kennen

Wahl haben – bewusst entscheiden

Foto von Efraimstochter, available under a CCO Public Domain license.

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

120

Groovy ist z. B. geeignet für ...



- Scripting
- Schreiben von Tests
- Buildmanagement (Gradle)
- Admin-Konsole in Java EE Anwendungen
- und natürlich in Kombination mit Grails

Grails ist z. B. geeignet für ...

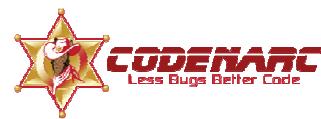


- Prototypen
- kleine Intranet-Anwendungen (keine Bange, falls es wächst)
- Microservices

Groovy Ecosystem



Gradle
a better way to build



GVM

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 123

Community



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

groovy/grails

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 124

Roadmap Grails



angekündigt für Juni 2015, leider noch nebulös

Foto von Unsplash, available under a [CC0 Public Domain license](#).

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 125

Roadmap Grails - Last update



Steve Bowman
@sbowman96

Follow

Cool hearing Graeme speak about Grails
3.0 & 3.1 roadmap in STL w/key OCI client
[@ObjectComputing](#) [@grailsframework](#)



RETWEETS 5

FAVORITES 5



11:42 PM - 22 Jul 2015

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 126

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

Rewrite MOP in 3.0



Plan vor Abschied von Pivotal (2014)



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

The Groovy roadmap



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

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 129



© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 130

Können sich diese Firmen irren?



They all use Groovy!



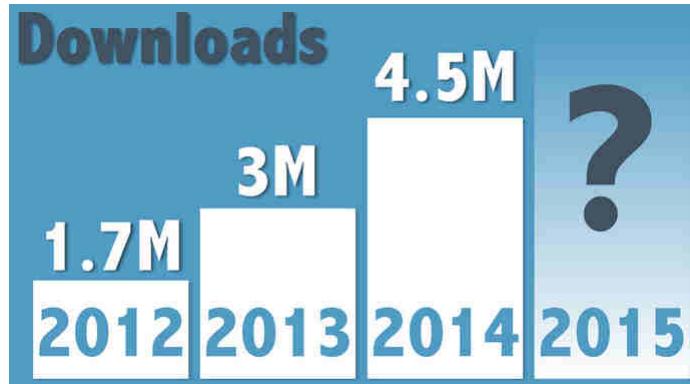
<http://groovy-lang.org/>

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 131

Steigende ...



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

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

| 132

... explodierende Downloadzahlen

**2.1 MILLIONS
FOR FIRST 4 MONTHS
OF 2015**

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

MAVEN CENTRAL ONLY

© 2015 Orientation in Objects GmbH Groovy und Grails – Quo vadis? 133

Orientation in Objects

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



**Vielen Dank für ihre
Aufmerksamkeit !**

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de