

# Scala für Enterprise-Applikationen

## Eine Bestandsaufnahme

Dr. Halil-Cem Gürsoy

adesso AG

Twitter: @hgutwit

G+: <http://goo.gl/hljRS>

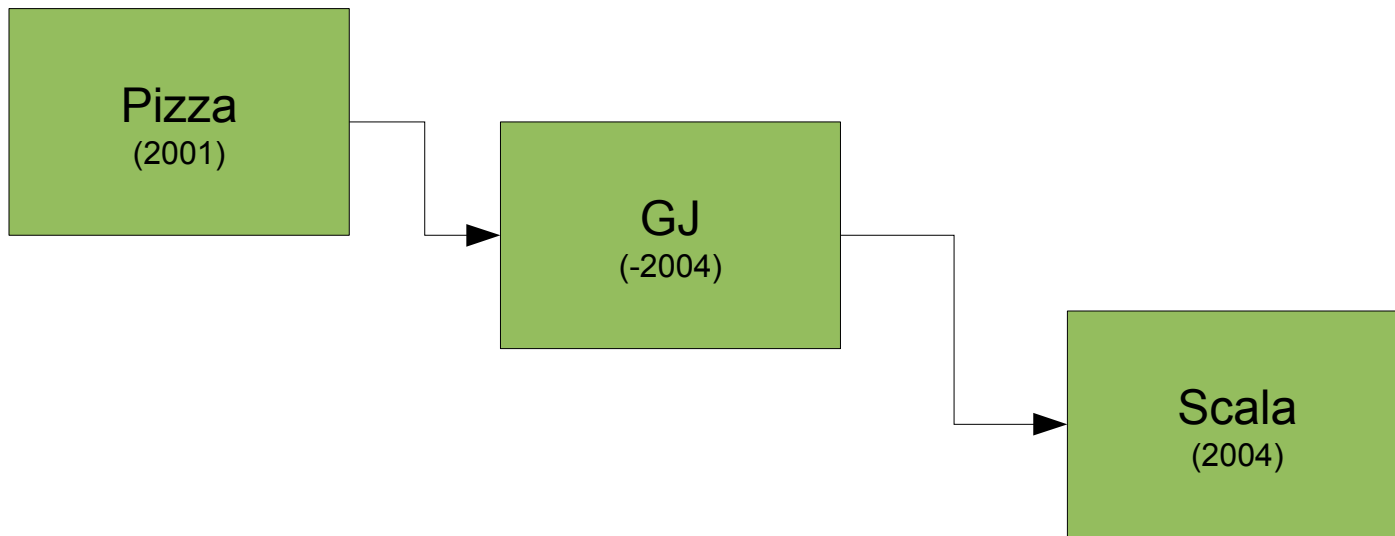
- ▶ Insgesamt über 10 Jahre Beratung und Entwicklung rund um Java
- ▶ davor Entwicklung im Forschungsumfeld
- ▶ Senior Software Engineer bei adesso AG, Dortmund
- ▶ Schwerpunkt EAI und SOA-Projekte im JEE Umfeld
  - > aber auch klassische JEE-Projekte
- ▶ Autor und Referent auf Konferenzen

- ▶ Motivation
- ▶ Enterprise ?
- ▶ Sprache und Syntax
- ▶ Scala-Versionen und Kompatibilität
- ▶ IDE's, Build-Systeme und Tool Support
- ▶ Support und Wartung
- ▶ Fazit

- ▶ Motivation
- ▶ Enterprise ?
- ▶ Sprache und Syntax
- ▶ Scala-Versionen und Kompatibilität
- ▶ IDE's, Build-Systeme und Tool Support
- ▶ Support und Wartung
- ▶ Fazit

- ▶ Blog(s) von A. Blewitt
- ▶ <http://alblue.bandlem.com/2009/08/is-scala-ready-for-enterprise.html>
- ▶ <http://alblue.bandlem.com/2009/10/scala-is-still-not-enterprise-ready.html>
  
- ▶ Scala Hype & Fanboyism (?)
- ▶ Scala zu Akademisch ! ?
- ▶ Kunden zu empfehlen ?

- ▶ Scalable *Language*
- ▶ Prof. Martin Odersky, EPFL



- ▶ Scala 2.9: 05.2011

- ▶ Motivation

- ▶ Enterprise ?

- ▶ Sprache und Syntax

- ▶ Scala-Versionen und Kompatibilität

- ▶ IDE's, Build-Systeme und Tool Support

- ▶ Support und Wartung

- ▶ Fazit

- ▶ Definitionen ?
  - > „*A business organization*“
  - > Unternehmen = *Enterprise*
- ▶ A. Blewitt's Definition
  - > Twitter – Nach seiner Definition kein *Enterprise* (!)
  - > Startups – auch kein *Enterprise*, da zu klein und unbedeutend
  - >  $10^n$  Mitarbeiter, X Millionen € Umsatz = *Enterprise*



***„Enterprise“ sind alle  
Unternehmen und  
Unternehmungen, von deren  
Erfolg Menschen existentiell  
abhängen***

- ▶ Motivation
- ▶ Enterprise ?
- ▶ Sprache und Syntax
- ▶ Scala-Versionen und Kompatibilität
- ▶ IDE's, Build-Systeme und Tool Support
- ▶ Support und Wartung
- ▶ Fazit

- ▶ Große Freiheitsgrade der Sprache und der Syntax
- ▶ Source-Lesbarkeit leidet unter Komplexität
- ▶ „*the source is obfuscated by the syntax*“

```
def quickSort[A <% Ordered[A]](xs: List[A]): List[A] = xs match {  
  case Nil      => xs  
  case y :: ys => ys partition (_ <= y) match { case (l1, l2) => quickSort(l1)  
    ++ (y :: quickSort(l2)) }  
}
```

(aus Wikipedia.de)

- ▶ Erfahrung notwendig!

- ▶ Große Freiheitsgrade der Sprache und der Syntax
- ▶ Source-Lesbarkeit leidet unter Komplexität
- ▶ „*the source is obfuscated by the syntax*“

```
def quickSort[A <% Ordered[A]](xs: List[A]): List[A] = xs match {  
  case Nil      => xs  
  case y :: ys => ys partition (_ <= y) match { case (l1, l2) => quickSort(l1)  
    ++ (y :: quickSort(l2)) }  
}
```

(aus Wikipedia.de)

- ▶ Erfahrung notwendig!

- ▶ „*pixie magic called implicits*“
- ▶ Casten von Typen durch Implicits
- ▶ (Komplexes) Regelwerk, wann welche Implicits herangezogen werden (Scopes)
- ▶ Sehr mächtiges Feature!
- ▶ Übersicht in großen Projekten?
- ▶ Sind Implicits ein Nachteil?

- ▶ Traits
  - > Zwischen Java-Interfaces und Klassen
  - > Ermöglicht „Mehrfachvererbung“
  - > Mächtiges Feature
- ▶ Einige ausgewählte Vorteile
  - > Alles ist ein Objekt
  - > Auch Funktionen sind Objekte (FP!)
  - > Native XML-Unterstützung
  - > Actors

- ▶ Viel Dokumentation (zu viel?)
- ▶ Viele Beispiele
- ▶ Inzwischen viele gute Bücher
- ▶ „akademische Bücher vs. Praxisbücher“

- ▶ Flache Lernkurve im Vergleich zu Java oder C#
- ▶ Weniger gut ausgebildete Entwickler
- ▶ weniger gute Entwickler als bei Java / C# ?
- ▶ „einfache“ Syntax ist auch ein Nachteil von Java!
- ▶ verleitet zur (Selbst)Überschätzung!

*„Ich bin ein Java/Groovy/Python/Ruby-Experte!“*



- ▶ Grundsätzliche Frage
- ▶ Ist die Syntax von Scala und die Sprachkonstrukte zu komplex?
- ▶ Was trauen Sie Ihren Entwicklern zu?
- ▶ Sind Ihre Entwickler Enterprise-Ready?

- ▶ Motivation
- ▶ Enterprise ?
- ▶ Sprache und Syntax
- ▶ Scala-Versionen und Kompatibilität
- ▶ IDE's, Build-Systeme und Tool Support
- ▶ Support und Wartung
- ▶ Fazit

- ▶ Scala 2.8 nicht binärkompatibel mit 2.7
- ▶ Scala 2.9 nicht binärkompatibel mit 2.8
- ▶ Verwendete Frameworks müssen im richtigen „Binärformat“ vorliegen
- ▶ Upgrade auf neue Scala-Version?
  - > update Dependencys
  - > u.U. update des Sourcecodes

- ▶ Tatsächlich ein Problem?
- ▶ Wie oft wechseln Projekte die Basis (z.B. JDK)?
- ▶ Aber: Tool's in einer neuen Version erfordern u.U. neue Scala-Version
- ▶ Ist Binärkompatibilität tatsächlich ein Problem?
- ▶ Wie ist es in Java?

- ▶ Motivation
- ▶ Enterprise ?
- ▶ Sprache und Syntax
- ▶ Scala-Versionen und Kompatibilität
- ▶ IDE's, Build-Systeme und Tool Support
- ▶ Support und Wartung
- ▶ Fazit

- ▶ Eclipse
  - > Scala IDE
  - > Seit Scala IDE 2.x (relativ) OK
  - > Entscheiden während der Installation: Scala 2.x oder 2.9 ?
  - > Teil des Typesafe-Stacks

# Scala IDE in Action

The screenshot shows the IntelliJ IDEA IDE with the Scala IDE in Action. The main editor window displays the following Scala code:

```
2+ * Created by IntelliJ IDEA.
8
9 package de.adesso.hgu.scalatrain
10
11 object Time {
12   def fromMinutes(minutes: Int): Time = {
13     re def fromMinutes(minutes: Int): de.adesso.hgu.scalatrain.Time
14     Time(minutes / 60, minutes % 60)
15   }
16 }
17 }
18 case class Time(hours: Int = 0, minutes: Int = 0) {
19   require(hours >= 0, "hours must not be negative")
20   require(hours < 24, "hours must be less than 24")
21   require(minutes >= 0, "minutes must not be negative")
22   require(minutes < 60, "minutes must be less than 60")
23 }
24 lazy val asMinutes =
25   minutes + 60 * hours
26
27 def - (that: Time): Int = {
28   minus(that)
29 }
30
31 def minus (that: Time): Int = {
32   require(that != null, "that must not be null")
33   this.asMinutes - that.asMinutes
34 }
```

The Project Explorer on the left shows the project structure, including the source code files. The Task List and Outline at the bottom left show the current file and its members.

- ▶ IntelliJ
  - > Gute Unterstützung *out of the Box*
- ▶ NetBeans
  - > Gute Unterstützung (?)
- ▶ Vim / Emacs
  - > Sowieso ;-)



- ▶ Simple Build Tool (sbt)
  - > In Scala für Scala
  - > Scala-Versionshandling
  - > Hudson / Jenkins Plugin
- ▶ Maven
  - > Ist möglich...
  - > gleiche Bugs und Probleme wie bei Java-Projekten
- ▶ Gradle
  - > Scala Plugin

# Beispiel für sbt-Konfiguration

```
name := "MyScalaProj"
version := "0.1"
organization := "de. adesso.hgu"
javacOptions ++= Seq("-source", "1.6", "-target", "1.6")
scalacOptions += "-deprecation"
scalaVersion := "2.8.1"
libraryDependencies += "org.mockito" % "mockito-all" % "1.8.5" %
"test"
libraryDependencies += "org.scala-tools.testing" %% "specs" %
"1.6.7"
libraryDependencies += "org.scala-tools.testing" %% "scalacheck" %
"1.8"
```

- ▶ Herausforderungen mit sbt
  - > sbt-Projekt in Eclipse importieren mit sbt-plugin *sbteclipse*
  - > sbt-Projekt in IDEA importieren sbt-Plugin vorhanden
  - > Bisher keine Integration von sbt in Eclipse oder IDEA analog zu Ant oder Maven (m2e)
  - > Dadurch größere Roundtrips
- ▶ sbteclipse
  - > Wird aktiv entwickelt

- ▶ Fast alle Java Frameworks
  - > Hibernate / JPA, Log4J, Spring (!) usw. usw.
  - > Conversions beachten !
  - > AOP-Frameworks...
- ▶ Pure Scala Frameworks
  - > Wachsende Anzahl, z.B. für Unittests (specs)
  - > Webframeworks
  - > Aber immer Abhängigkeit zur Scala-Version

- ▶ Motivation
- ▶ Enterprise ?
- ▶ Sprache und Syntax
- ▶ Scala-Versionen und Kompatibilität
- ▶ IDE's, Build-Systeme und Tool Support
- ▶ Support und Wartung
- ▶ Fazit

- ▶ „Wen kann ich im Notfall verklagen?“
- ▶ Problem in konservativem Projektumfeld
- ▶ Typisch Deutsch ;-)
- ▶ Seit 2011: Typesafe
  - > Von den „Machern“ von Scala und Akka
  - > Kommerzieller Support für den „Typesafe Stack“
  - > Scala, Akka, Scala IDE, sbt

- ▶ Motivation
- ▶ Enterprise ?
- ▶ Sprache und Syntax
- ▶ Scala-Versionen und Kompatibilität
- ▶ IDE's, Build-Systeme und Tool Support
- ▶ Support und Wartung
- ▶ Fazit

- ▶ Sprache und Syntax ●
- ▶ Lernkurve, Verfügbarkeit ●
- ▶ Scala-Versionen und Binärkompatibilität ●
- ▶ IDE-Integration ●
- ▶ Build-Systeme ●
- ▶ Tools und Frameworks ●
- ▶ Support und Wartung ●



Scala ist Reif für Enterprise!



**Danke für Ihre Aufmerksamkeit!**

