# Scala

Gedanken zu einer Programmiersprache

Sebastian Eidecker

16. März 2016

# Wer als Werkzeug nur einen Hammer hat, sieht in jedem Problem einen Nagel.

Paul Watzlawick

# Inhalt IT im Wandel

# IT im Wandel

Inhalt

Herausforderungen

# IT im Wandel

Inhalt

Herausforderungen

Manifeste

IT im Wandel

Manifeste

Herausforderungen

Scala





IT im Wandel

Herausforderungen

Manifeste

Scala

Management Summary

IT im Wandel

Herausforderungen

Manifeste

Scala

Management Summary

Ein wenig Code

IT im Wandel

Herausforderungen

Manifeste

Scala

Management Summary

Ein wenig Code

Spannendes

**Software Engineering** 

# **Software Engineering**



IT im Wandel

Herausforderungen

· Stabilität und Resilienz





- · Stabilität und Resilienz
- Wertbeitrag

- · Stabilität und Resilienz
- Wertbeitrag
- Businesstreiber

- · Stabilität und Resilienz
- Wertbeitrag
- Businesstreiber

Matthias Magnor – CEO Surface und Contract Logistics

# IT im Wandel









Manifeste

 Antwortbereit, Widerstandsfähig, Elastisch, Nachrichtenorientiert (2013)

• Gut gefertigt, Stets Mehrwert, Gemeinschaft aus Experten, Produktive Partnerschaften (2009)

 Individuen und Interaktionen, Funktionierende Software, Zusammenarbeit mit dem Kunden, Reagieren auf Veränderung (2001)

- Antwortbereit, Widerstandsfähig, Elastisch, Nachrichtenorientiert (2013)
- Gut gefertigt, Stets Mehrwert, Gemeinschaft aus Experten, Produktive Partnerschaften (2009)
- Individuen und Interaktionen, Funktionierende Software, Zusammenarbeit mit dem Kunden, Reagieren auf Veränderung (2001)

# Wo stehen wir?

# Wo stehen wir im Wettbewerb?

Businesslogik?

// ToDo

# Scalable Language

# Scala

**Management Summary** 



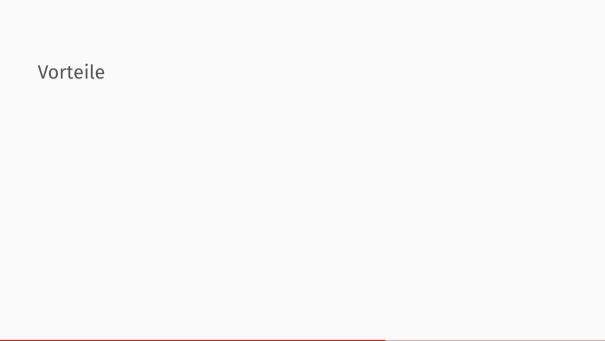
• Java 2.0

- Java 2.0
- Objektorientiert

- Java 2.0
- Objektorientiert
- Funktional

- Java 2.0
- Objektorientiert
- Funktional
- · Java Virtual Machine

- Java 2.0
- Objektorientiert
- Funktional
- · Java Virtual Machine
- Entwicklungsprozess wie gehabt



Vorteile

Modern

# Vorteile

- Modern
- Verständlich funktional

- Modern
- · Verständlich funktional
- · Java-Ökosystem

- Modern
- · Verständlich funktional
- · Java-Ökosystem
- Macht Spaß

- Modern
- · Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

# Nachteile

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

# Nachteile

Komplex

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

# Nachteile

- Komplex
- Statisch typisiert (nicht mehr cool)

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

# Nachteile

- Komplex
- Statisch typisiert (nicht mehr cool)
- Java-Ökosystem (nicht mehr cool)

# Scala

Ein wenig Code



# Eine Klasse

case class Person(firstName:String, lastName:String)

# Scala

**Spannendes** 



# Scalable real-time transaction processing

Akka

1 9

#### Akka

- Scalable real-time transaction processing
- · Will die aktuellen Probleme lösen

### Akka

- Scalable real-time transaction processing
- · Will die aktuellen Probleme lösen





# ScalaTest . .

ScalaTest				
•				

# Scala

Mehr für Nerds



Mehr für Nerds

Sprecht mich an











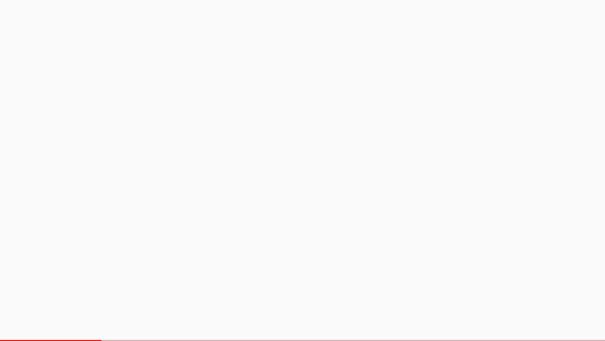
# Mehr für Nerds

- Sprecht mich an
- · Eigener Termin bei Interesse

### Mehr für Nerds

- · Sprecht mich an
- · Eigener Termin bei Interesse
- Heiko Seeberger: "Durchstarten mit Scala. Tutorial für Einsteiger (2. Aufl.)"

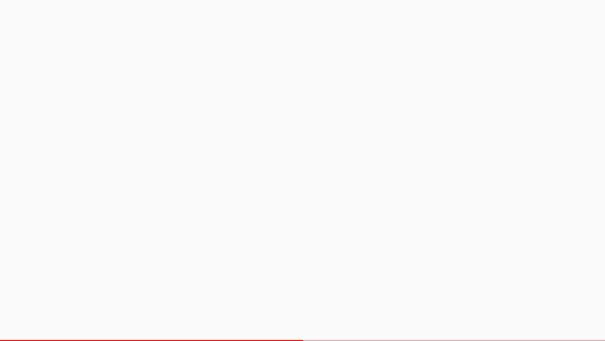


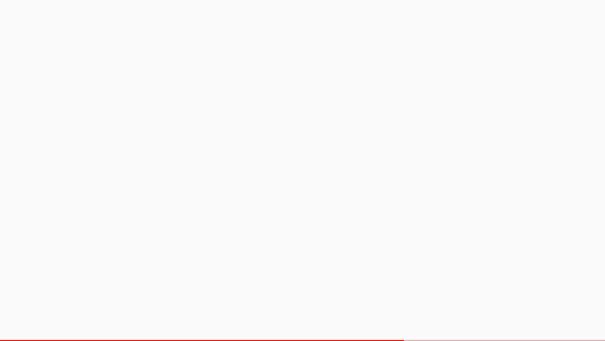


# Primzahlen

# Primzahlen

```
def swap(i: Int, j: Int) {
   val t = xs(i); xs(i) = xs(j); xs(j) = t
   ()
}
```









Titel			
•			

•

Titel			
•			

•



Titel ·

# Titel .

Titel			
•			
•			

Titel	Titel
•	

Titel	Titel
•	•
•	

.

Titel	Titel
•	•

•

Titel	Titel
•	•