SCALA

BEWERTUNG FÜR DEN FINNOVA-KONTEXT

Lenzburg, 19. Februar 2015

finnova AG Bankware www.finnova.ch
Schweiz | Luxemburg | Singapur

finnova

SCALA IM KONTEXT VON FINNOVA

PRODUKTENTWICKLUNG ALS ANFORDERUNG

Grundsätzliche Vorgehensweise:

> Die Technologiewahl (inklusive Programmiersprachen) erfolgt anhand der Anforderungen und des Verwendungskontexts.

Motivation für den Einsatz von **Scala** bei finnova

- Scala unterstützt das **reaktive Paradigma** aufgrund des Objekt-Funktionalen-Paradigmas ideal
- Scala ermöglicht eine **hohe Effizienz** bei der Implementierung
- Scala fügt sich nahtlos in die **JVM**-Welt ein

Finnova setzt die **Bewertungskriterien für Scala** aus Sicht der **Produktentwicklung**

SCALA

BEWERTUNGSKRITERIEN

Die Gewichtung erfolgt anhand der Kriterien

- Produktivität
- Investitionsschutz
- Mitarbeiter-Recruiting

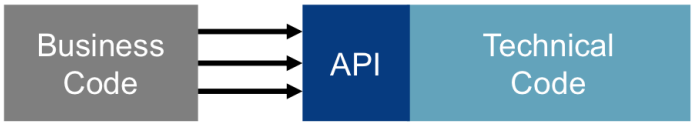
finnova

SCALA

BEWERTUNG: PRODUKTIVITÄT

Finnova Entwickler erstellen bankfachliche Anwendungen.

- Dies unterstützt Scala wesentlich besser als z.B. Java 8
- Komplexe Konstrukte werden durch das Framework verborgen



Feature	Java	Scala
Fluent API	+	+ +
Combinators	+	+ +
Control Structures	✗	+ +
Internal DSL	✗	+
Type-Safe Calls	+	+ +

Quelle: msg systems ag: Insights into the awesome Scala programming language (2010-07-08)

SCALA

BEWERTUNG: INVESTITIONSSCHUTZ

Finnova muss Nvestitionsschutz bieten.

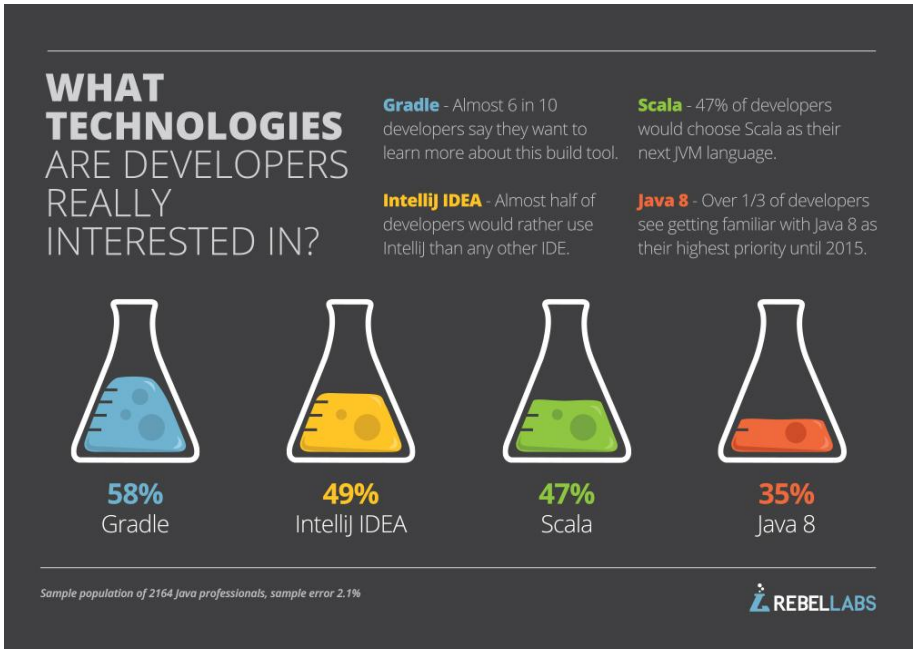
- Scala ist als Sprache im Mainstream angekommen:
  - Technology Radar von Thoughtworks seit 2012 auf dem Status „adopt“
  - http://www.thoughtworks.com/radar/languages-and-frameworks/scala
- Die Omega-Architektur unterstützt Technologiewechsel
- Die Umsetzungseinheiten erlauben einen wirtschaftlich sinnvollen Ersatz entlang der Produktweiterentwicklung

SCALA

BEWERTUNG: RECRUITMENT

Finnova möchte attraktiv für Mitarbeiter sein.

- Scala wurde von Entwicklern als interessante Sprache für 2015 benannt



Quelle: msg systems ag: Einschätzung und Ratschläge für eine Technologie-Entscheidung (Stand: Dezember 2014)

finnova

SCALA

ENTSCHEIDUNGSMATRIX FÜR FINNOVA

- Bewertungsmatrix auf Basis des msg-Berichts zu Scala verwendet.
- Die Gewichtung wurde zugunsten der Priorisierung der finnova-Bewertungskriterien angepasst.

Gewichtete Entscheidungsmatrix für finnova

	Gesamtsumme	18	50	87

Auf Prioritäten reduzierte gewichtete Entscheidungmatrix für finnova

1	Produktivität	-3	9	42
2	Investitionsschutz	2	16	14
3	Mitarbeiter-Recruitment	-9	-3	12
	Summe	-10	22	68

- Die vollständige Bewertungsmatrix finden Sie in der Anlage.

SCALA/TYPESAFE

AUSGEWÄHLTE REFERENZEN FINANCE

Financial

Fidelity: Fidelity Investments

- **mission-critical trading platform**
  - the code base was reduced from 479,000 to 5,000 lines of code

Goldman Sachs: Investment Banking

- **price/risk platform**: Rollout 2015
  - stratigic initiative: better support for development team

Morgan Stanley

- **Direct delivery of financial data to their customers**
  - large fanout of data: from ~30,000 destinations to roughly 12 million
- **Universal Order Status**
  - provides near-real-time status on orders to their team of ~20K Financial Planners

Nomura: Investment Banking

- Nomura re-architected their platform using Akka
  - Akka made delivering blazing-fast performance in a distributed systems environment accessible to all developers

Details finden Sie in der Anlage

Hinweis - Vereinbarung mit Typesafe: Diese Informationen unterliegen NDA

ANLAGEN

Kriterien	Gewichtete Entscheidungsmatrix für finnova			Java 5/6/7	Java 8	Scala		Bemerkung
		Punkte:		18.0	50.0	87.0		
	Organisatorisch	Punkte:		**11.0**	**10.0**	**20.0**		
	Projektrisiken sind maximal minimiert	2.0		2	1	-2		
2	Kein Umdenken bei den Entwicklern notwendig (Programmierparadigmen)	3.0	*(1,0)*	2	1	1	*(-2,0)*	Ausgangslage: PL/SQL
2	Existierendes Know-How bei den Entwicklern weiterverwenden	3.0	*(1,0)*	2	1	1	*(-2,0)*	Ausgangslage: PL/SQL
	Initiale Weiterbildungsaufwände sind minimiert	1.0		2	1	0	*(-2,0)*	Fokus: Fachentwicklung
	Initiale Einarbeitungsaufwände sind minimiert	1.0		2	1	0	*(-2,0)*	Fokus: Fachentwicklung
1	Hohe Produktivität der Entwickler	3.0	*(2,0)*	0	1	2		
3	Hohe Motivation für Entwickler	3.0	*(2,0)*	-1	0	2		
3	Den Kunden und Mitarbeitern gegenüber Innovationskraft beweisen können	3.0	*(2,0)*	-2	-1	2		
	Technologisch	Punkte:		**9.0**	**36.0**	**34.0**		
2	Zukunftssicherheit der Technologie	4.0		-2	2	1		
2	Hohe Nutzung der Technologie-Möglichkeiten	2.0		-1	1	2		
1	Gute Werkzeug-Unterstützung (Integrated Developer Environment)	3.0	*(2,0)*	2	2	1	*(0,0)*	Einsatz intelliJ IDE
1	Gute Werkzeug-Unterstützung (Build Automation Tools)	3.0	*(2,0)*	2	2	1		
1	Unterstützung der existierenden Technologien in der Scala-Welt	3.0	*(1,0)*	-1	0	2		
1	Unterstützung der existierenden Technologien in der Java-Welt	3.0	*(4,0)*	2	2	2		
	Compile-Time Performance	0.0	*(1,0)*	2	2	-1		Nicht relevant
	Laufzeit-Performance	4.0		1	2	2		
	Sprachlich	Punkte:		**-2.0**	**4.0**	**33.0**		
1	Ausdrucksstärke der Programmiersprache (für Produktivität und Wartbarkeit)	3.0	*(2,0)*	-2	-1	2		
1	Flexibilität der Programmiersprache (z.B. für "Internal DSLs", etc)	3.0	*(1,0)*	-2	-2	2		
	Skalierbarkeit der Programmiersprache (von kleinen bis großen Anwendungen)	2.0		1	1	2		
	Unterstützung des Programmierparadigmas Objekt-Orientierung (OO)	2.0		2	2	2		
1	Unterstützung des Programmierparadigmas Funktionale Programmierung (FP)	3.0	*(1,0)*	-2	-1	2		
	Leichtigkeit der Entwicklung von Anwendungs-Code	4.0		2	2	2		
	Leichtigkeit der Entwicklung von Bibliotheks-Code	1.0		2	2	-1		
	Auf Prioritäten reduzierte gewichtete Entscheidungmatrix für finnova							
1	Produktivität			-3	9	42		
2	Investitionsschutz			2	16	14		
3	Mitarbeiter-Recruitment			-9	-3	12		
		Punkte:		**-10.0**	**22.0**	**68.0**		

ANLAGEN

REFERENZ: FIDELITY

Fidelity Investments

FMR LLC or Fidelity Investments is an American multinational financial services corporation. Founded in 1946, it is one of the largest mutual fund and financial services groups in the world.

- Challenge
  - The challenge at Fidelity is to continually stay at the forefront of technology. This ethos permeates every facet of the company. While they are the largest asset manager in the world, Fidelity sees themselves first and foremost as a technology company.
  - Fidelity views the Typesafe Reactive Platform as a promising key component of their architecture over the next 10 years.
  - Fixed income, quant trading, e-trading/counterparty resolution, portfolio management, and infrastructure have all taken an interest in using Scala and Akka. These groups are using the technology under an architectural exception.
- Solution
  - Like many large companies with complex environments, Fidelity's requirements include the need to co-existseamlessly with existing "legacy" systems.
  - Fidelity chose Akka because the actor based system enabled them to take advantage of massively multicore processing and scale up and out dynamically without threatening the investment in Java at the firm.
  - Once developers started using Akka (which is written in Scala), they began using Scala as well.
- Result
  - One of the first applications Fidelity chose to rewrite to Scala/Akka from Java was its mission-critical trading platform.
  - **Using Scala/Akka, the code base was reduced from 479,000 to 5,000 lines of code.**
  - Less code means fewer errors, better performance, less cost, more productivity.

ANLAGEN

REFERENZ: GOLDMAN SACHS

Goldman Sachs

Founded in 1869, Goldman Sachs offers a gamut of investment banking and asset management services to corporate and government clients worldwide, as well as institutional and individual investors. It is one of the largest investment banking firms in the world.

- Challenge
  - The challenge was to build a responsive and scalable price risk management platform that would be available to their traders located around the globe.
  - In the past the bank created its very own programming language.
  - The fact that they have made a large multi year commitment to Typesafe is enormous news.
- Solution
  - **Scala has provided a more expressive, concise, and readable option to the London based development team.**
- Result
  - **The price/risk platform** is now in the test phase and will be rolled out in 2015.
  - The investment bank has established a support agreement with Typesafe to ensure that there are no hiccups with the roll out of this strategic initiative.

ANLAGEN

REFERENZ: MORGAN STANLEY

Wealth Management - Morgan Stanley

- Challenge
  - **Direct delivery of financial data to their customers**
    - Morgan Stanley has an internal system designed to deliver customer-related trading data to their team of financial advisers. Their goal was to open up this data stream directly to customers. This expansion massively increased the outflow of data from their systems due to a large fanout of data: **from ~30,000 destinations to roughly 12 million**. They enlisted Typesafe to consult on how to apply Akka to this task.
    - Morgan Stanley implemented a hybrid solution consisting of DB + JMS queues and a hub-and-spoke fan-out implemented with Akka. They engaged Typesafe to devise approaches for integrating back-pressure into the system.
  - **Universal Order Status**
    - Engaged Typesafe to review system that provides **near-real-time status on orders to their team of ~20K Financial Planners**. System utilizes Play and Akka.
- Solution
  - Direct delivery of financial data to their customers
    - After reviewing a number of options, a solution was devised that would meet their needs and abide by their internal constraints on how the data stream needs to be treated (order preservation). This solution used Akka in two ways: 1. to broadcast financial data to multiple "delivery shards". 2. Manage the "delivery shards" with on-line redistribution of work to maintain load and minimize resource utilization.
    - Several approaches were worked out for retro-fitting back-pressure into their system (which was not designed with back-pressure in mind). A key take-away was that Akka provided all the necessary signaling and data flow to detect when back-pressure was needed.
  - Universal Order Status
    - Answered many questions the development team had, and provided guidance and advice on best practices. Also collected some great feedback that was passed on to teams within Typesafe.
- Result
  - Direct delivery of financial data to their customers

ANLAGEN

REFERENZ: NOMURA

Nomura Securities International
Established in 1925, Nomura is one of the largest investment banking and securities firms in the world.

- Challenge
  - **Building an architecture relying on traditional multi-threading** is a challenging undertaking.
  - Developers struggle to grasp the complexities introduced, often by accident, in multi-threaded applications.
  - Constrained by complexity, deadlines, and limited developer resources capable of programming multi-threaded applications with legacy tools, it was time for a change.
- Solution
  - **Nomura re-architected their platform using Akka**, which provided a modern and easily graspable concurrency framework. Scala provided the functional backbone to tame the accidental complexities of imperative, multi-threaded code.
  - The combination of Scala and Akka helped Nomura scale not only their systems, but also their team; a team capable of delivering a distributed system on time and on budget.
  - Akka provided the necessary abstraction from complexity in order to achieve this.
  -
- Result
  - Implemented a grid of 600 workers that starts up in seconds, which is the **fastest performance Nomura has seen** without relying on UDP or other tricks reserved for only the most expert-level developers.
  - **Akka made delivering blazing-fast performance in a distributed systems environment accessible to all developers**.

ANLAGEN

REFERENZEN: WEITERE QUELLEN

- https://typesafe.com/resources/case-studies-and-stories
- https://prezi.com/3wmewobf8i4z/ibm-and-scala/