

A W V E 1 1

Carsten Bormann



Warum
gut für Dich und
für Deine Firma ist



Carsten Bormann <cabo@tzi.org>

Umfrage:

- ▶ Wer hat schon etwas von Ruby on Rails **gehört**?
- ▶ Wer hat es schon **ausprobiert**?
- ▶ Wer hat schon ein Projekt damit **durchgeführt**?

Dynamische Programmiersprachen

- ▶ Perl (1987, Perl 5: 1994): aus sh, awk, sed; \$-Zeichen
- ▶ PHP (1995, PHP 4: 2000): größlicher Perl-Verschnitt
- ▶ JavaScript (1995, aus Self 1986): die **unterschätzte** Sprache
- ▶ Python (1990, 2.x: 2000): Europas Antwort auf Basic und Perl
- ▶ **Ruby (1995): Japans Antwort auf Perl, Smalltalk und Lisp**





Was ist **RAILS** ?

- ▶ Ruby-basiertes Webentwicklungs-Framework
- ▶ **MVC: Model/View/Controller**
- ▶ **CoC: Convention** over Configuration
- ▶ **DRY: Don't repeat yourself**
- ▶ **TDD: Test-Driven Development**

Opinionated Software

- ▶ Ansatz: „Extracted Framework“
 - Löst tatsächliche Probleme der Entwickler
 - Wird zeitnah weiterentwickelt
 - Nah am Stand der Technik (z.B. **AJAX**)
- ▶ Rails versucht nicht, alle Probleme zu lösen
 - Rails adressiert den „Sweet-Spot“
- ▶ „Rails Core“: kleine Gruppe unter Führung von DHH
 - DHH hat eine Nachricht an alle, die nicht einer Meinung mit ihm sind...



Argumente für Rails

- ▶ Produktivität (verglichen mit Java-Entwicklung):
Im „Sweet Spot“: 5–20 fach
- ▶ Geringere Entwicklungszeiten
- ▶ (Durch Agilität)
größere Nähe zu tatsächlichen Anforderungen



Argumente für Rails

— *Of fun*



Glückliche RAILS-Entwickler



- ▶ Arbeiten produktiver
- ▶ Machen weniger Fehler
- ▶ Hinterlassen beim Kunden einen besseren Eindruck

▶ **Kündigen nicht**



Entwickler, die **RAILS** beherrschen

- ▶ Haben mehr Spaß an der Arbeit
- ▶ Sind weniger gestreßt
- ▶ Kriegen die interessanteren Aufgaben
- ▶ **Behalten ihren Job**

Ruby on Rails @ Uni Bremen

- ▶ Master-level, 6 CP (ECTS)
 - Master-/Diplomstudis... und alle, die sich fit genug fühlen
- ▶ Nicht nur Rails, sondern auch Methodik der Anwendungsentwicklung für das Web
 - Vorgehen: Projektorientiert
- ▶ **„Agile Web-Entwicklung“ (AWE)**
 - Direkte Umsetzung von Kundenanforderungen in Software

AWE11

- ▶ Vorkurs: 19.01., 02.02., 09.02. 09–12 im MZH 1400
 - Übungsaufgaben zur eigenen Bearbeitung
- ▶ Kompaktkurs: 21.02.–05.03.2011
 - 09–21 h in der Großen Halle des Volkes (P2)
 - 12 Tage (inkl. 2 Sonnabende)
 - Vorlesung, Projekt, Kommunikation, Ergebnisse

INTENSIV

Week 1

- ▶ Get people up and running **in two days**
- ▶ Give them all to be “feature-complete” **in the first week**

Mon	Intro (look at what I’m not doing), motivation, overview Get to know Rails Learn the environment
Tue	Migrations Basic Agile Development Understand the customer
Wed	More about views and controllers; AR, Validation; SQL; Security
Thu	AJAX RJS
Fri	(More about TDD) Lead into a weekend of work
Sat	<i>Sprint</i>

Week 2

- ▶ Three more days of refinements
- ▶ Focus is on **project completion**

Mon	Unicode Internationalization in Rails
Tue	REST
Wed	More Security Performance, Deployment
Thu	<i>Finishing</i>
Fri	<i>Project presentations</i>
Sat	<i>Finishing</i>

Bis zum 21.2.2011

- ▶ Drei Präsenztermine
 - 19.1., 2.2., 9.2. (9-12, MZH 1400)
- ▶ Übungsaufgaben
 - Unscharfe Aufgabenstellung, eigenverantwortliche Bearbeitung
 - Erfordert eigene Disziplin!
 - Bis 2.2. in Ruby laufen lernen
 - Bis 21.2. erste Rails-Erfahrung sammeln
 - Abgabe als (Pointer auf ein) Repository (!)
- ▶ Zweiergruppen bilden
 - Aufgaben in Paarprogrammierung oder einzeln bearbeiten!