Teaching at a University

Prof. Dr.-Ing. Carsten Bormann, Universität Bremen TZI
Railsconf Europe 2007, Berlin DE, 2007-09-19

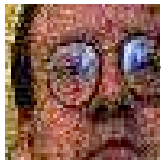
This public version sanitized for copyrights and privacy rights — sorry about that

If Ruby is so great,

why are Universities still teaching Java?

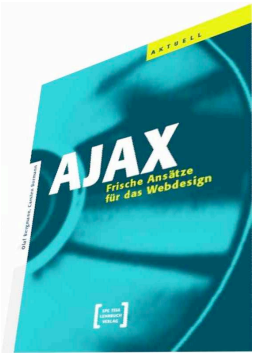
Who are you?

1. Who **teaches** (coaches, consults with newbies) Rails?
2. Who **works** at a University?
3. Intersection of 1 and 2 (teach && work)?
4. Who **is** at a University?
5. Intersection of 1 and 4 (teach && is)?



.class.ancestors

- I **teach** and do **research** at universities
  (Uni Bremen TZI, UdK Berlin IEB, TU Helsinki netlab)



- I occasionally
  - write **books**
  - do real **work** netCS
  - help create software

```
$ screen -r
$ toast
```

Germany

- You can't earn credit points for learning a programming language at an **Universität** (that wouldn't be **science**)

- Universitäten (research universities)
  - Science

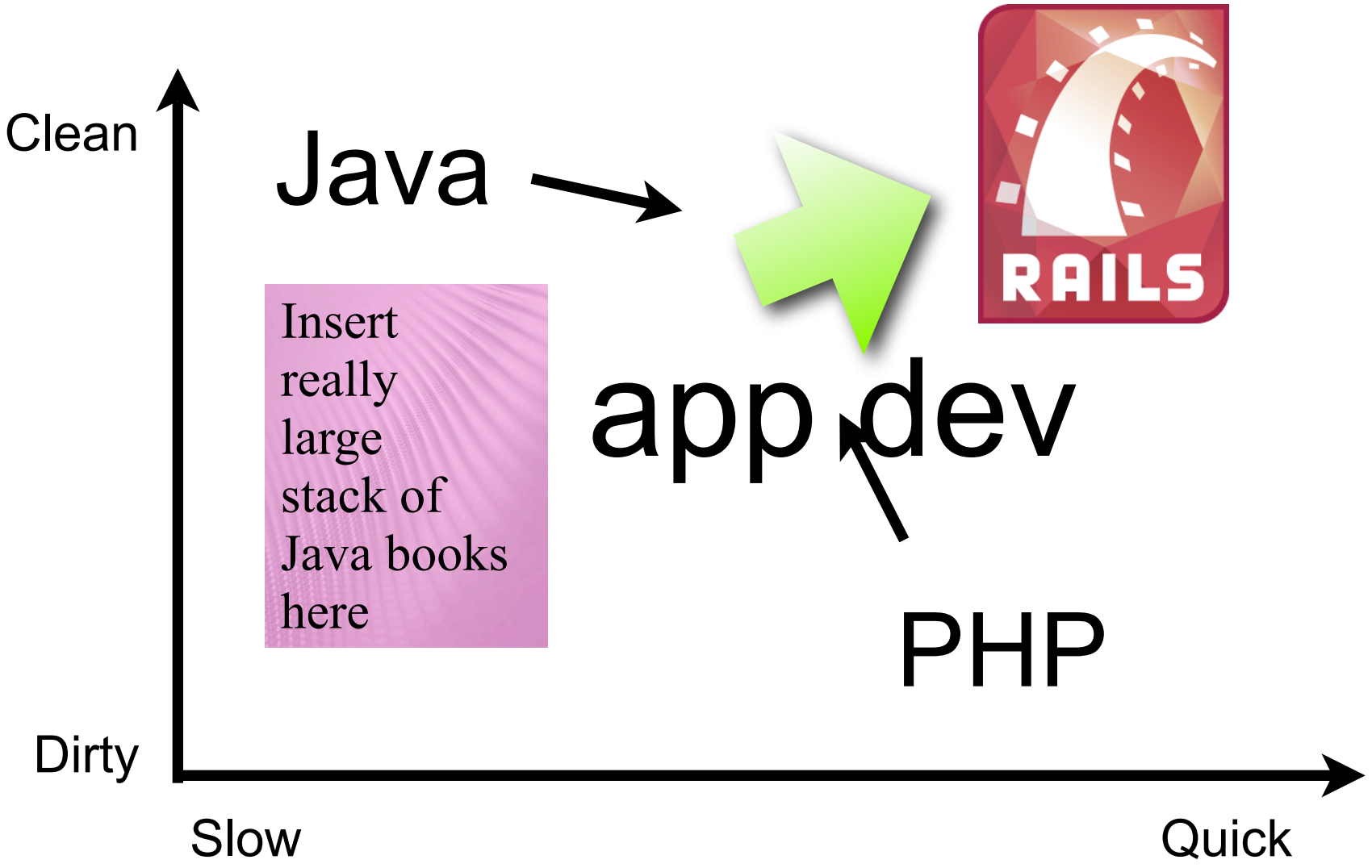
Insert image of Einstein here

- Fachhochschulen (no english term)
  - Skills

Insert image of mechanic here

How to motivate teaching Rails?

- To colleagues?
  - hard (see above)
- To myself?
  - I need the students
- To students?
  - easy (see below)

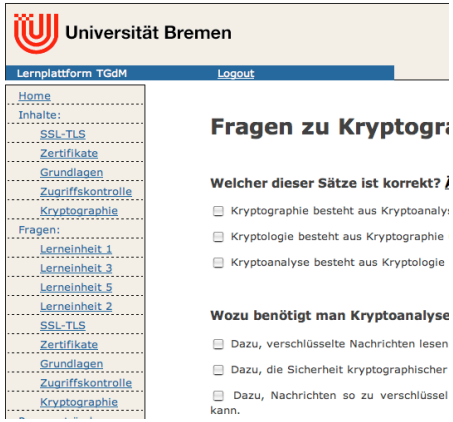

Clean
Java
RAILS
Insert really large stack of Java books here
app dev
PHP
Dirty
Slow
Quick

Can't really teach "Rails", anyway

- We actually teach **web application development**
- In: (Prerequisites)
  - Basics of Internet Technologies (Net and Web)
- Out: (Objective)
  - **Agile development**
  - Web-based apps as a specific form of **fast app dev**
  - Ruby as the leading **dynamic language**

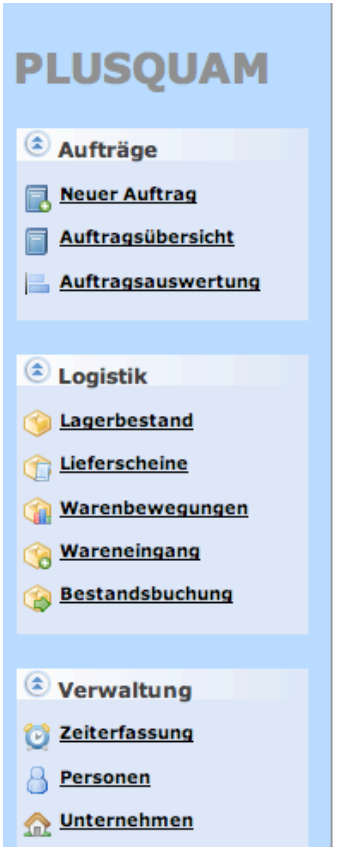
Eat your own dogfood.

- The TZI administrative/financial database is a Rails site.
- One afternoon, we knocked off a little course management system in Rails.
- We run internal and external projects using Rails.
- We have lots of project ideas that will be enabled by Rails.



Example for a Thesis Project

- Is it possible to get a German **SME** to contract and **pay** for an Agile Development Project?
- Are there maybe Agile Development issues specific to **Germany**?



Google Summer of Code 2007

- Atom Publishing Protocol Support for Ruby + Rails
  by **Gerrit Kaiser**, mentored by Benjamin Joseph Bleything
  (see **http://app.rubyforge.org**)
  - released, but not completely done yet
  - to lead to a Bachelor Report by November

Teaching Rails at a University

Prof. Dr.-Ing. Carsten Bormann, Universität Bremen TZI

One interesting approach

- Use Ruby/Rails to teach programming to **non-programmers**
- http://rubylearn.com (Charles Severance, UMich), http://wonderfullyflawed.com/course/ (Trek)

Welcome to Teaching Programming with Ruby and Rails

This is a very much under construction web site being initially built by Dr. Charles Severance to collect resources to help in teaching Ruby and Rails to beginning programming students.

The first course I am developing is **SI539 - Design of Complex Web Sites** being taught Fall 2007 at the University of Michigan. I am using this site to make my materials publically available under the Creative Commons Attribution 2.5.

Ruby on Rails @ Uni Bremen

- Master-level course, 6 CP (ECTS)
  - but open for younger people, if they feel up to it
- „Agile Web-Entwicklung“ (**AWE**)
  - (no, not “bewegliche Netzentwicklung mit Schienen”)
- Not just Rails, but the agile dev methods, too

How can you do agile development without **customers**?

- You can’t.
- So where do you get them?
- _____
- If you want to play **customer** in a future course, please contact me at

  cabo@tzi.org

Structure of the AWE course

- Learn **Ruby** in a no-credits pre-course
  - a couple of weeks before the Rails course
  - hard to keep up the motivation here
- Lock up the students in a computer room for two weeks
  - Start the day with some “lectures”
  - Let them work in pairs on projects for the rest of the day
  - Encourage communication

Insert image of freezing guy here

Organization

- Assign a **customer** to each **pair**
- have a hard deadline at the end of the 12 days

- Morning: **Lectures**
- Late afternoon: **communication break**
- Rest of the time: project work (incl. customer meetings)

- The usual stuff (SVN, wiki, ...)

The communication break

- Groups are encouraged to do peer-teaching (mutual coaching)
- 17:00–18:00 each day:
  - I **pick** a subject
  - groups (2) are **randomly** paired (4)
  - group pairs (4) are paired again (8)
- Result: 2 to 3 distilled presentations on the subject prepared in a common Wiki

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 More about AR, Validation; SQL Security
Thu	AJAX RJS
Fri	(More about TDD) Lead into a weekend of work

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	*Project presentations*
Fri	*Finishing*

6 CP = 180 h

- About 36 h of mandatory preparation (learn Ruby)
- About 32 h of frontal instruction (with some discussion)
- About 8 h of student presentations (with lots of discussion)
- About 104 h of:
  - conversation with customers
  - implementation, testing
  - on-line learning
  - peer-teaching
- 144h/12d = 12h/d = INTENSE

Who is going?

- Funny:
  - Half is sophomore (1st term)
  - Half is graduate level (7th term or more)
- Great mix for getting students to learn and teach at the same time.

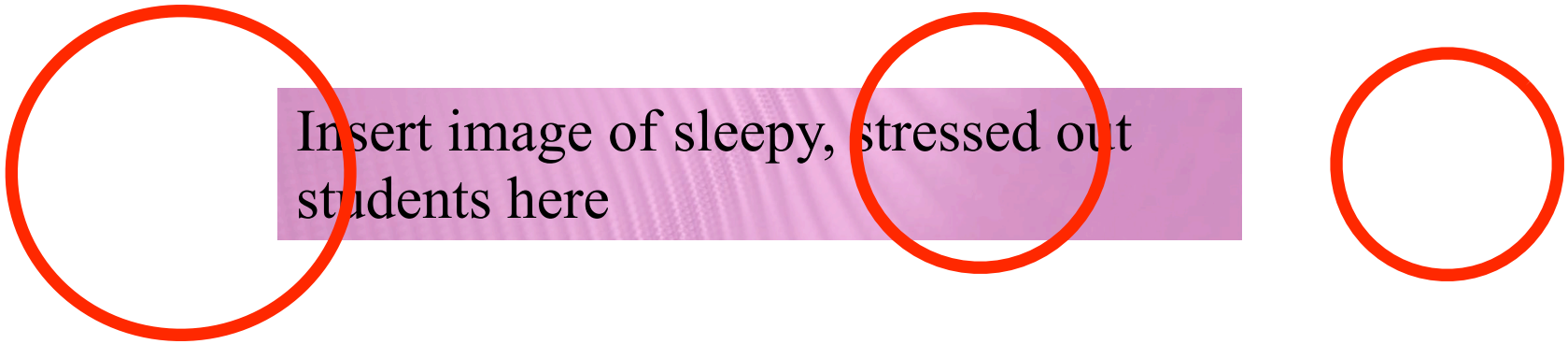
What they liked

- Ruby
- Rails
- The format (2 intense weeks in one room with many iMacs)
- The combination/mix of presentations and practice
  - some discussion about the distribution in time
- The snowballing/communication breaks

What they didn't like (1)

- No time for groceries and laundry
  - Next time, we'll tell them beforehand

INTENSE



What they didn’t like (2)

- I brought up TDD only on the fifth day
  - to be fixed
- I used a (long) screencast in class
  - don’t do that
- Some of my slides still at the previous level of Rails
  - shame on me

What I liked

- Incredible level of motivation
  - coupled with high level of (voluntary) mutual cooperation
- Students' own experience mixes with my guidance and their ad-hoc research
  - In the end, I probably learned even more than the students
- Students like AWE
  - "This has been the best course I have attended at a university"

Rails is moving — fast

- Much of the material out there is in 1.0-land
  - or somewhere not far beyond
- Start-with-SQL vs. migration approach
- Classic vs. REST
- (see DHH's keynote)

- You want to demonstrate best practices from the start
  - but they change

What could be fixed in Rails

- (to make setup of teaching environments easier:)
  fix the SQLite support
  - most groups migrate (!) to MySQL after a couple of days
- the “docs” (actual comments unprintable)
- symbols vs. strings

- i18n, I10n (at least make the plugins more reliable)

- Today’s students want IDEs, code browsers, refactoring, ...

So, can you teach Rails at a Uni?

- Yes, you can!
  - But you probably should think fresh
  - Model **teaching** on actual **usage**
- AWE course: Demanding
  - 2-week **blackout**
  - Have to **re-work** much of it every year
- Why isn't it done in more places?
  - Unis are conservative?
  - Ignorance? Haughtiness?