



Zwischenstand: Iteration 3 abgeschlossen

Rückblick: Iteration I

AspectJ?

PHP?

Java?

Haskell?

?

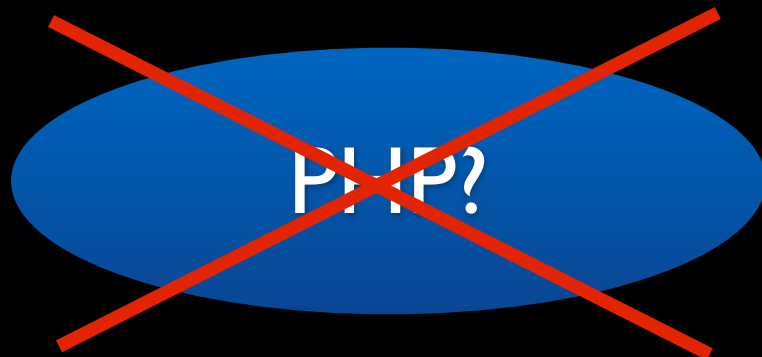
Scala?

Ruby on Rails?

Python?

PHP?

Java?



AspectJ

Scheduling
Prototyp 1

Scheduling
Prototyp 2

Scheduling
Prototyp 3

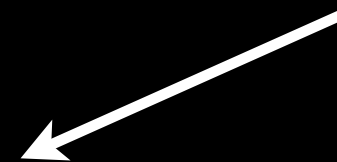
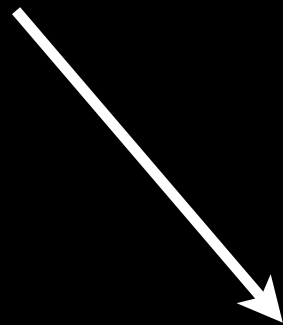
Iteration 2

Scheduling
Prototyp 1

Scheduling
Prototyp 2

Aktueller
Scheduler

Scheduling
Prototyp 3



Iteration 2

Aktueller
Scheduler

Relationales
Modell

Use Cases

Iteration 2

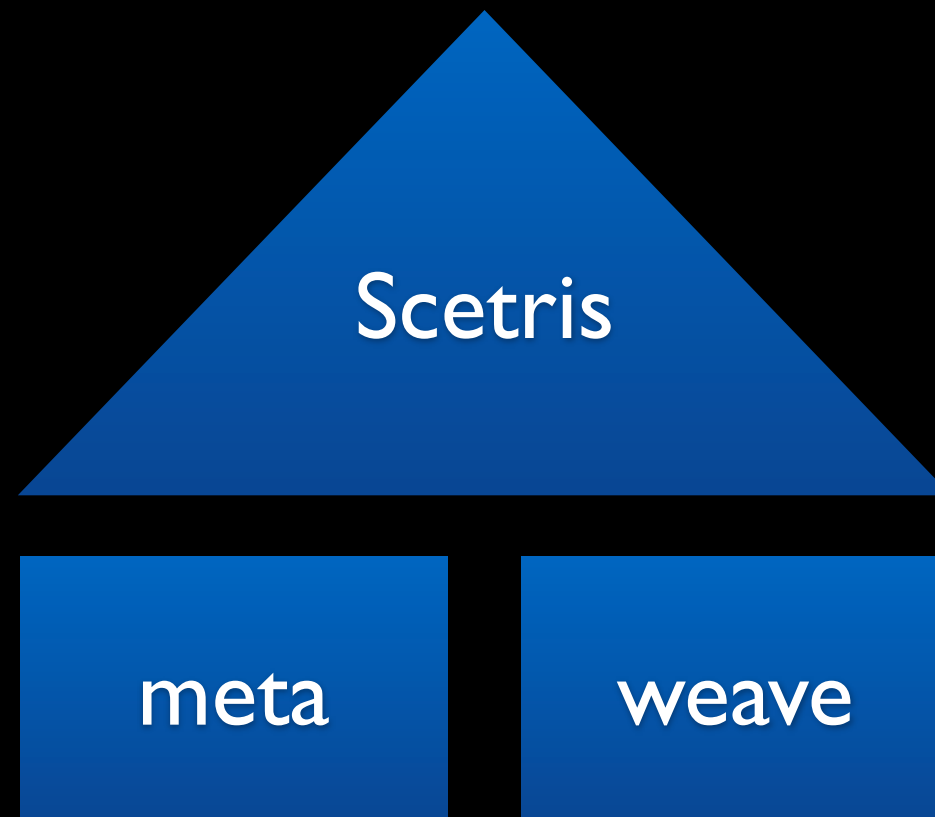
Relationales
Modell



Objekt-Relationales
Modell

- Konzepte
 - Beliebige Features
 - Beliebige Rollen / Erlaubnisse
 - Beliebige Attribute

Iteration 2



Iteration 2

meta

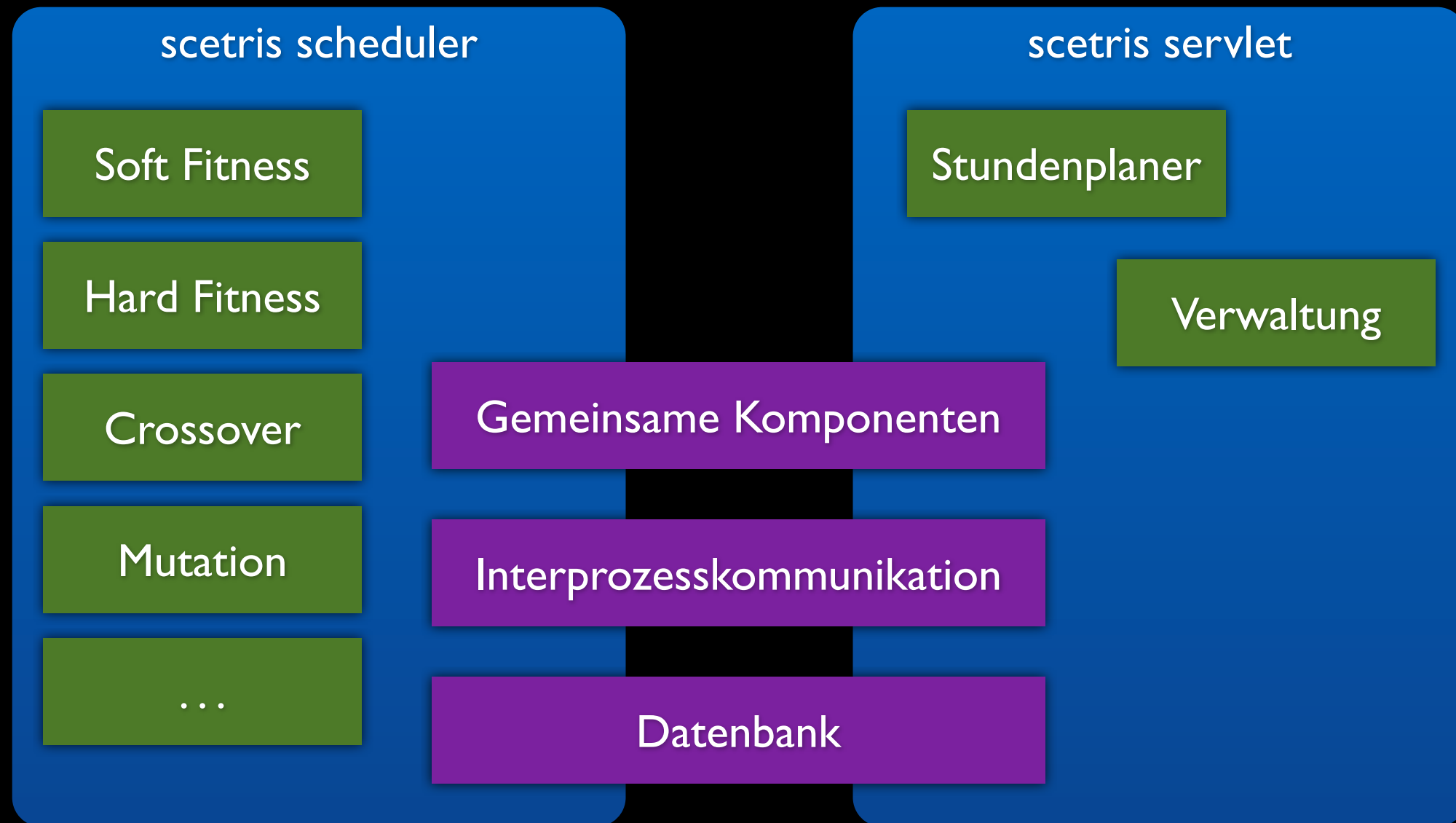
weave

O/R-Mapping via
Code Generation
from XML

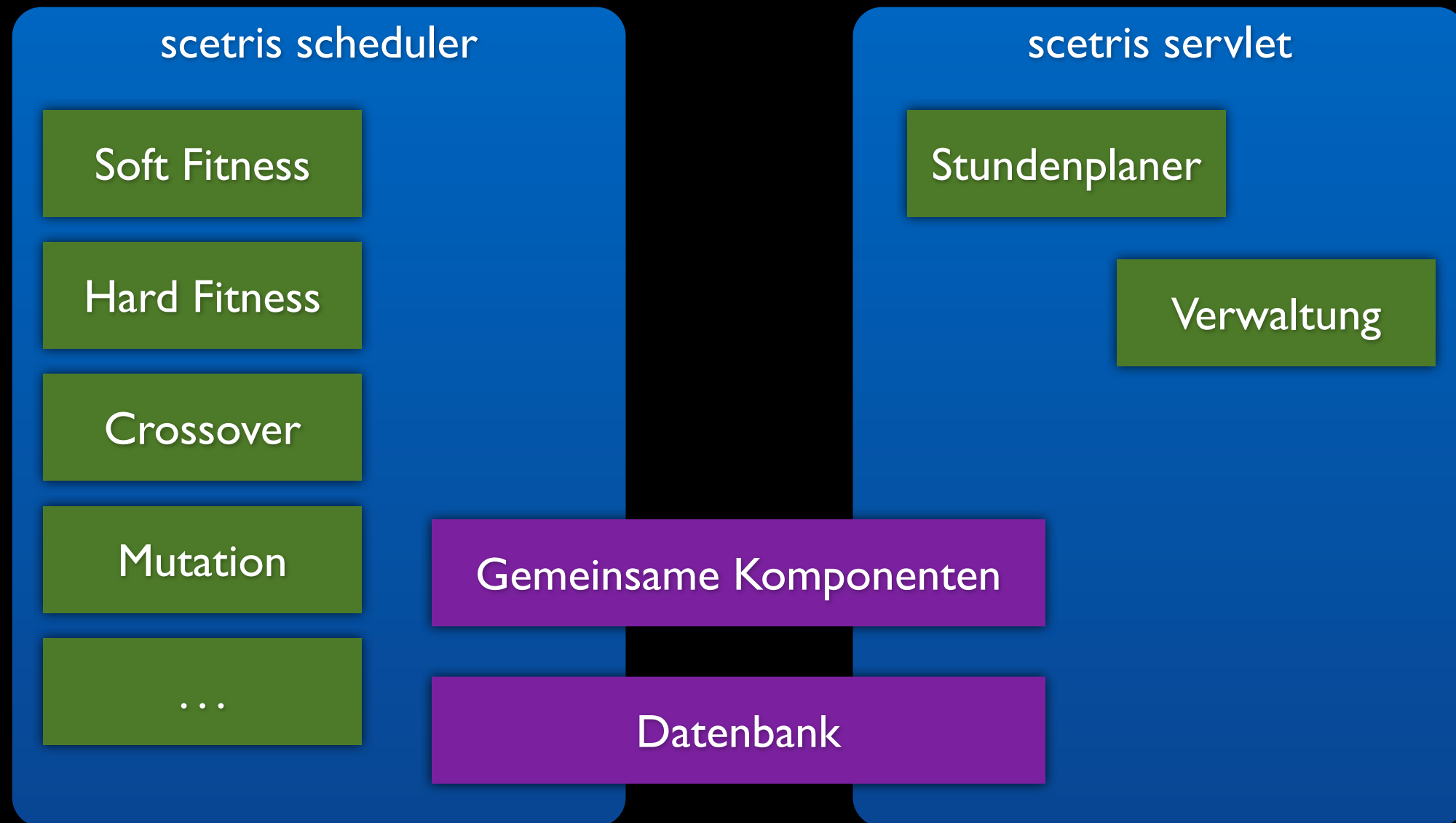
MVC-Webframework
using XSL-T as
Template Engine

Iteration 3

Integration Scheduler



Integration Scheduler



Was haben wir gemacht?

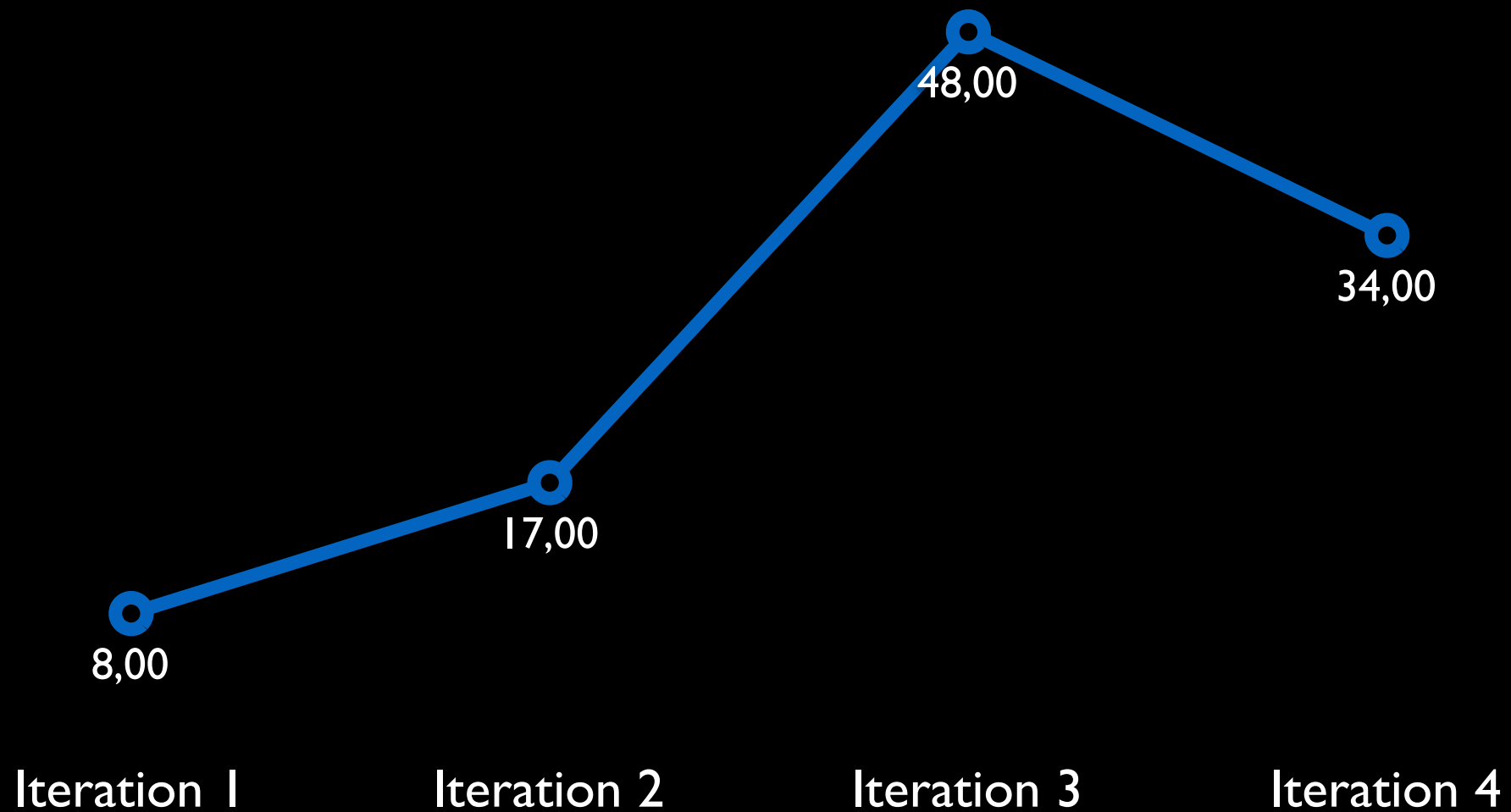
- Unit-Tests mit JUnit 4
- Import / Export
- konsolenbasierter Installer
- Scheduler integriert
- Funktionalität Webinterface erweitert

Vorführung

questionable facts

- 57486 lines of code
 - 13% Templates (XSL-T)
 - 11% XML-Dateien (build.xml, Relations.xml, ...)
- kurz vor Revision 1000
- 13392 lines of code actually
 - ~50% covered by tests!

Entwicklungsprozess



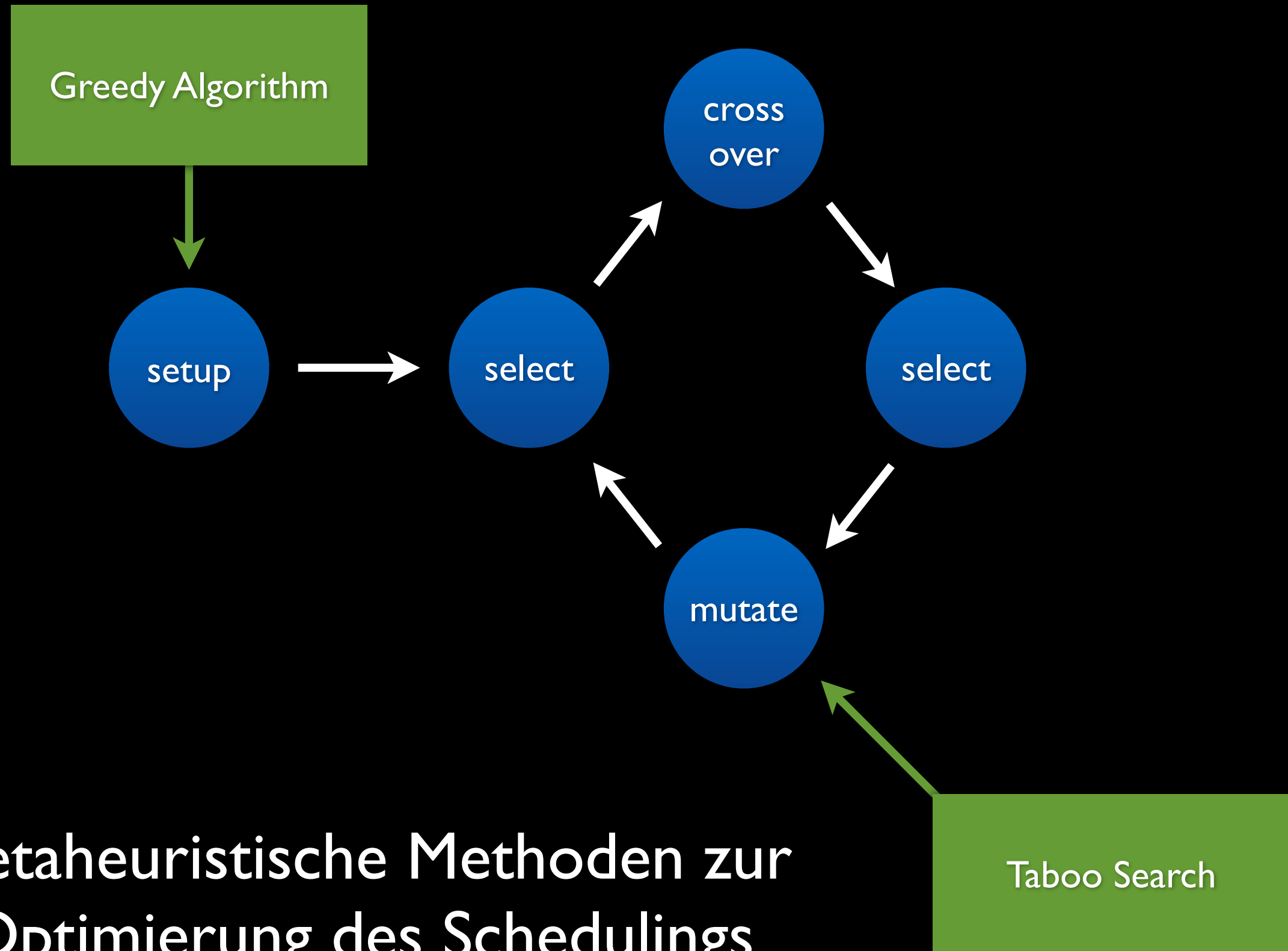
Team Aufstellung

- Experten haben sich herausgebildet
- Kompetenzteam
- Aufgaben lassen sich schnell zuteilen
- Konstruktive Atmosphäre

Ausblick: Iteration 4

Ausblick: Iteration 4

- Funktionalität aller UseCases bereitstellen
- Scheduler: Optimierung, Benchmark
- Kollaborative Scheduling-Findung
- PDF-Ausgabe mittels XSL-FO
- Ziel: Beta Release



Metaheuristische Methoden zur
Optimierung des Scheduling

Roadmap

- 28.11.2010 **Beta RC** (Iteration 4)
- 30.12.2010 **Release Candidate** (Iteration 5)
- 16.12.2010 **SCORE Summary Draft**
- 08.01.2011 **SCORE Summary**
+ **v1.0 Final**

Danke!