

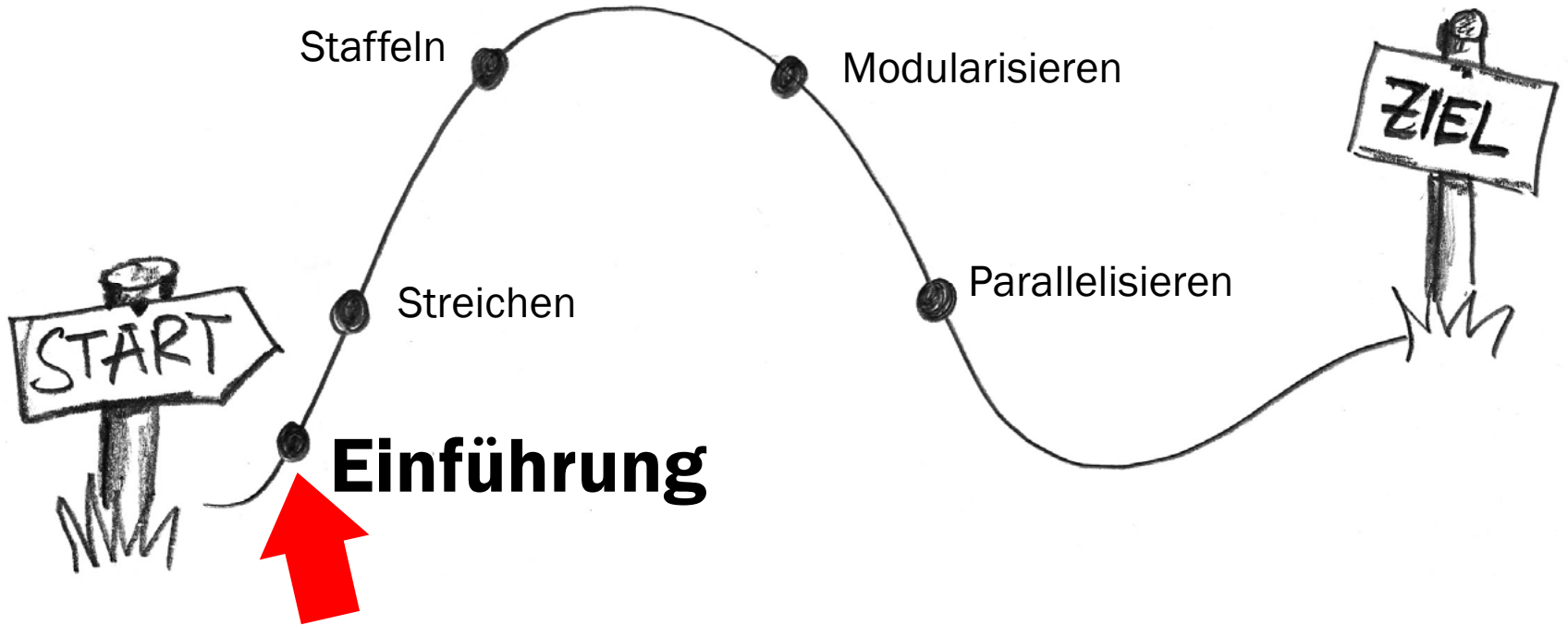
Wenn's mal länger dauert...
Continuous Integration
beschleunigen mit Hudson

Dr. Simon Wiest

Java User Group Cologne
05.10.2009, Köln



Agenda



Über den Referenten: Dr. Simon Wiest

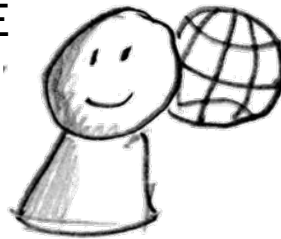
Privat: Hudson-Committer



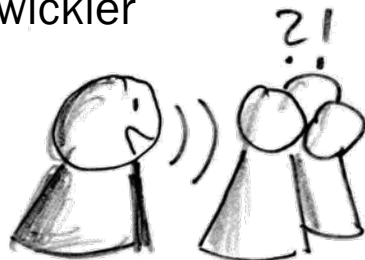
Lokalisierung DE



Plug-In Entwickler



Internationalisierung

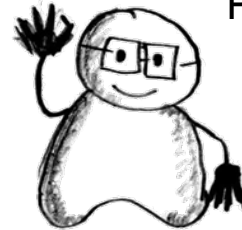


Support (Mailing Listen)

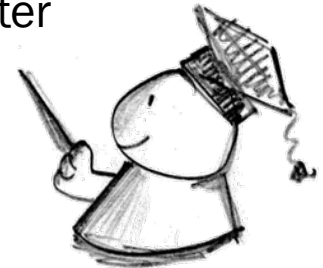
Beruflich: Hudson-Anwender



Projektleiter



Java Architekt

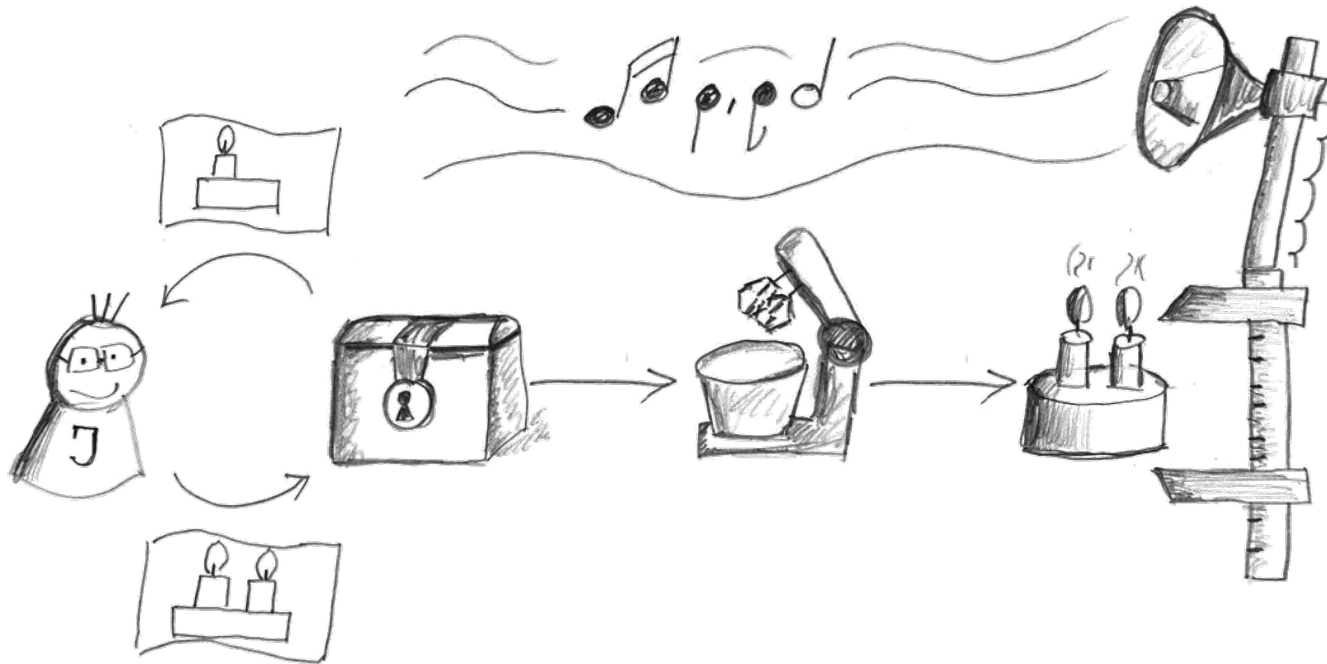


Coach



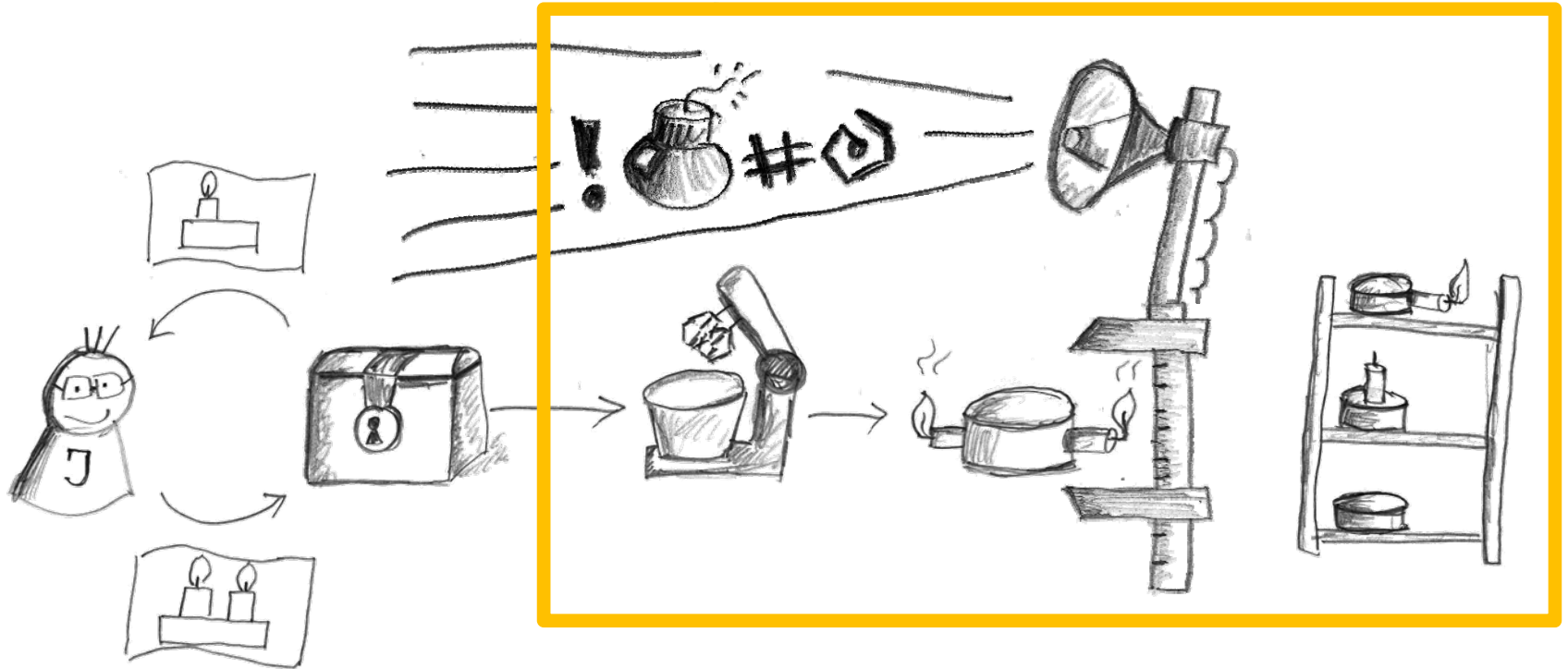
IT-Freiberufler

Was ist kontinuierliche Integration (CI)?

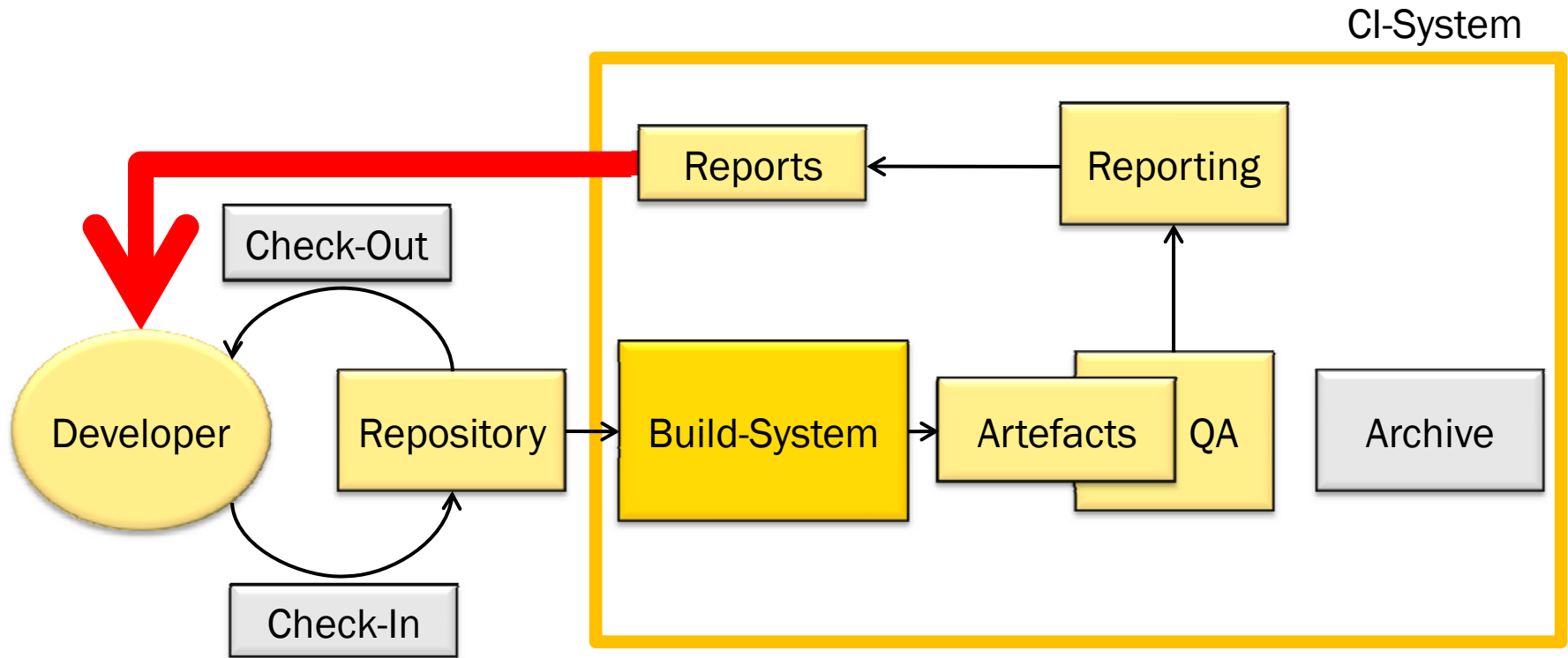


Was ist kontinuierliche Integration (CI)?

orchestriert durch CI-System



Was ist kontinuierliche Integration (CI)?



- Subversion
- CVS
- Perforce
- Git
- ...

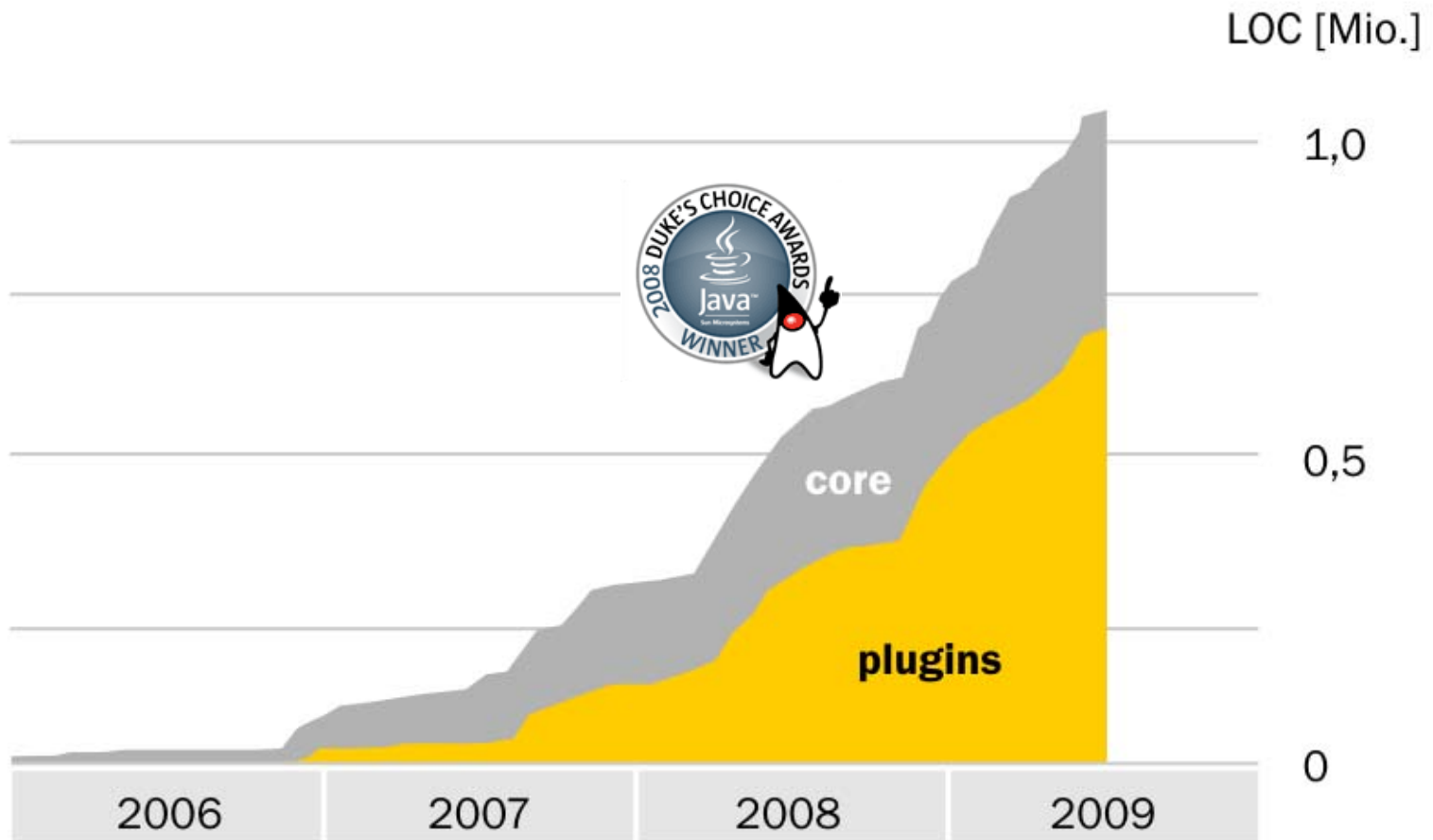
- Ant
- Maven
- Shell-Skript
- Batch-Datei
- ...

- JUnit
- TestNG
- CheckStyle
- PMD
- ...

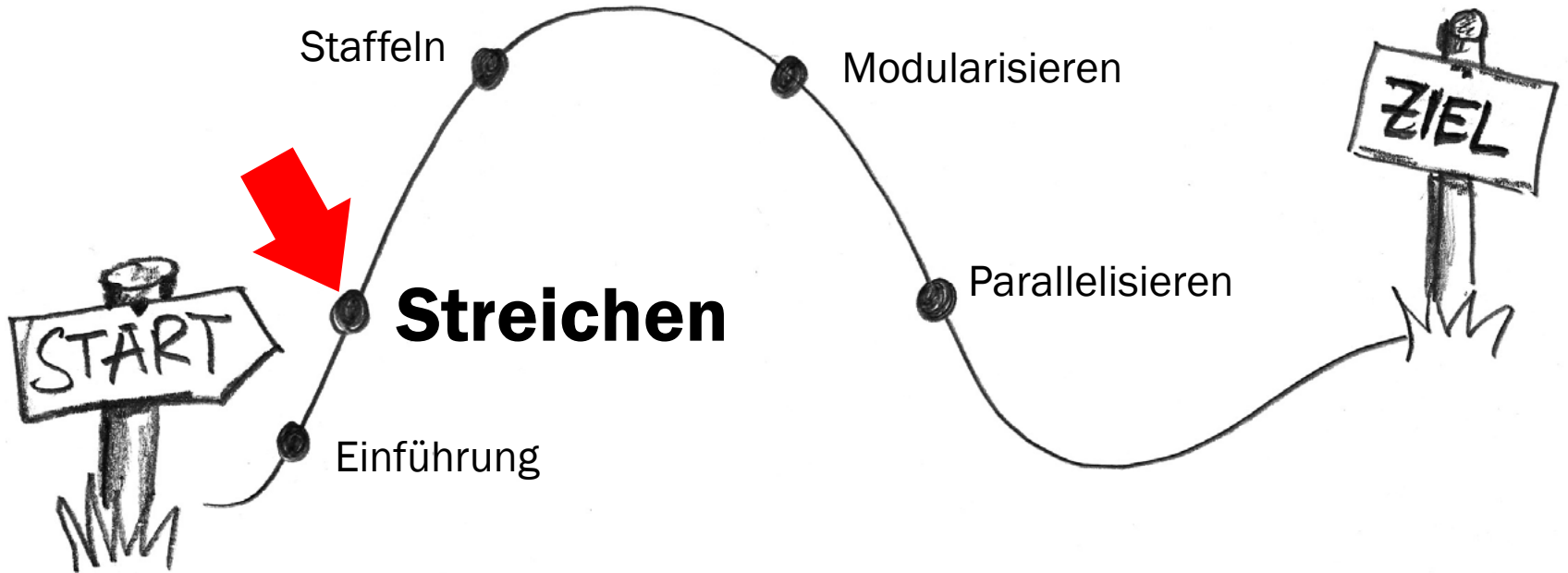
Wer verwendet Hudson (und darf es zugeben)?



Hudson: Ein Projekt in voller Fahrt.



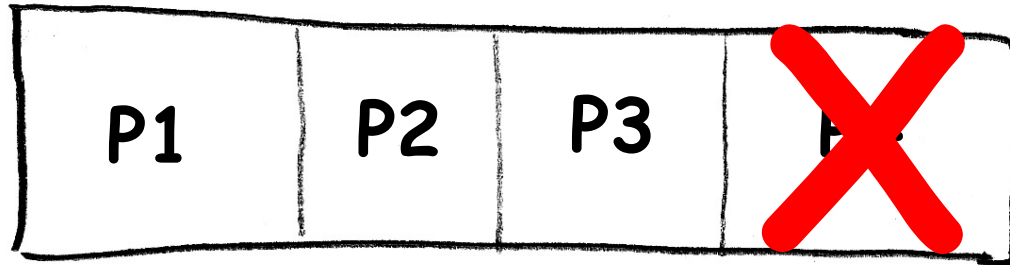
Agenda



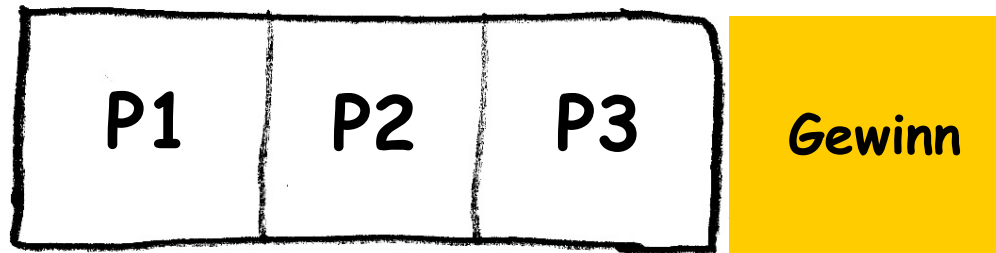
Streichen: Was kann entfallen?



vorher



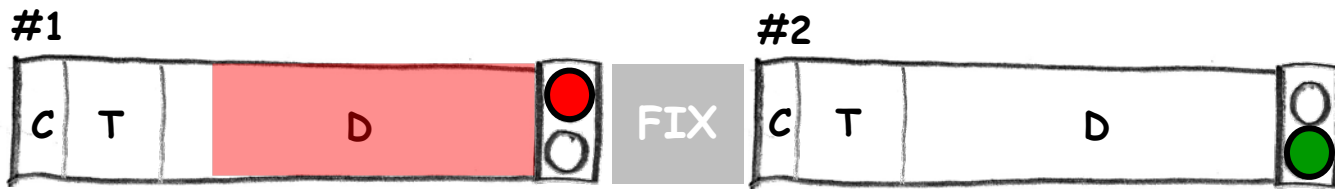
nachher



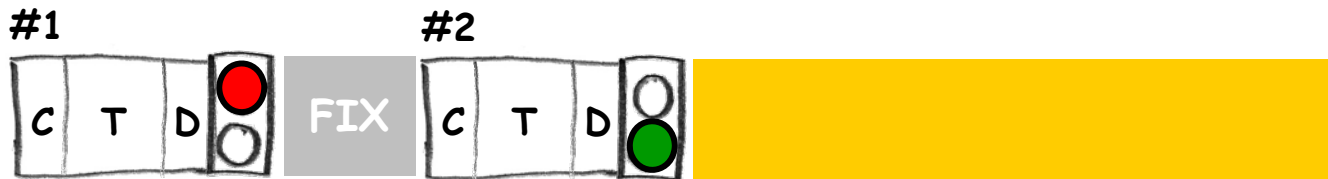
Streichen: Beispiel



vorher

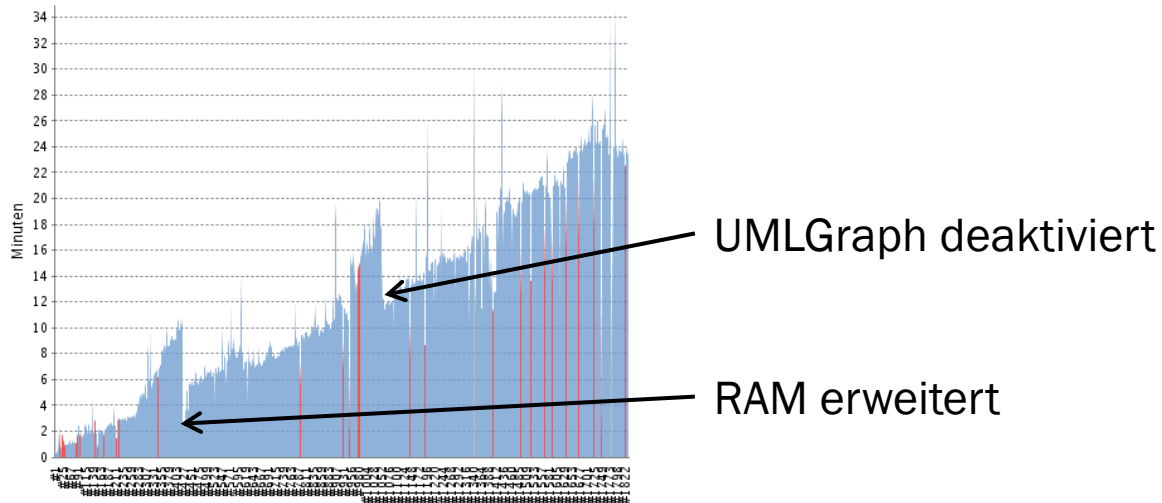


nachher



Streichen: Wie unterstützt hier Hudson?

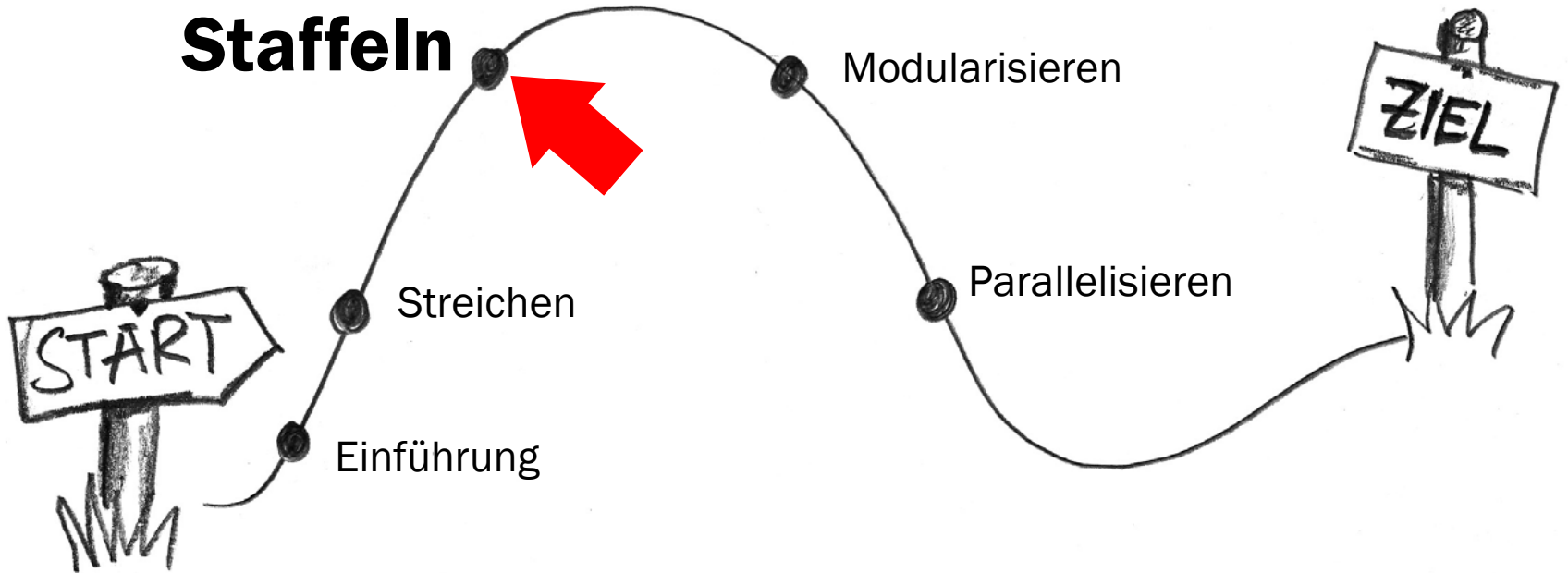
- Trends über Projektverlauf, z.B. Buildzeit



- Intuitive Ad-Hoc-Analyse in Web-Oberfläche



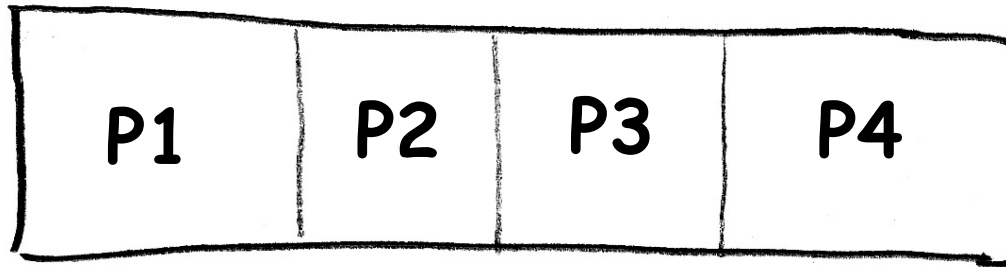
Agenda



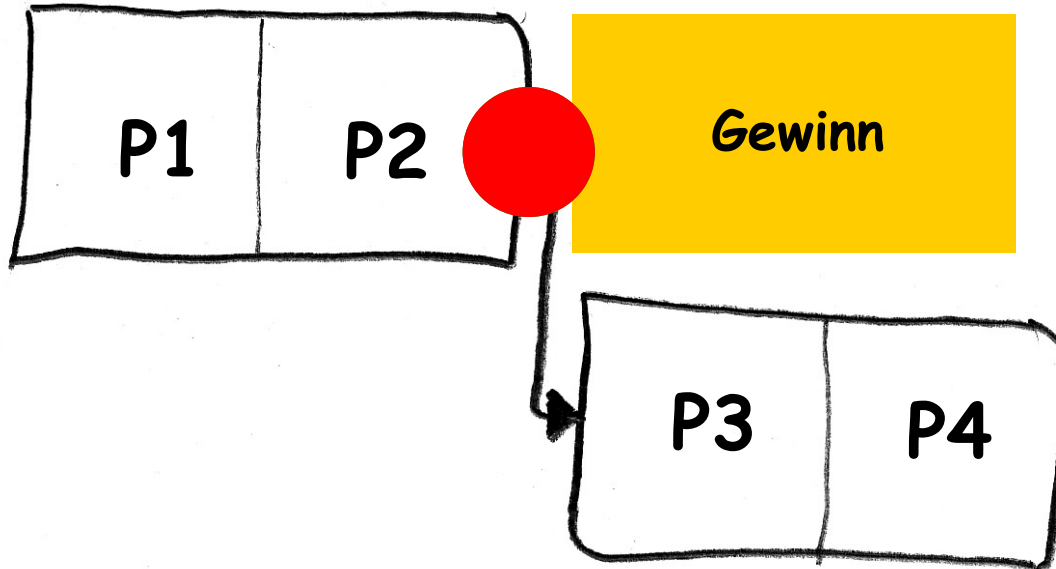
Staffeln: Das Wichtigste zuerst!



vorher



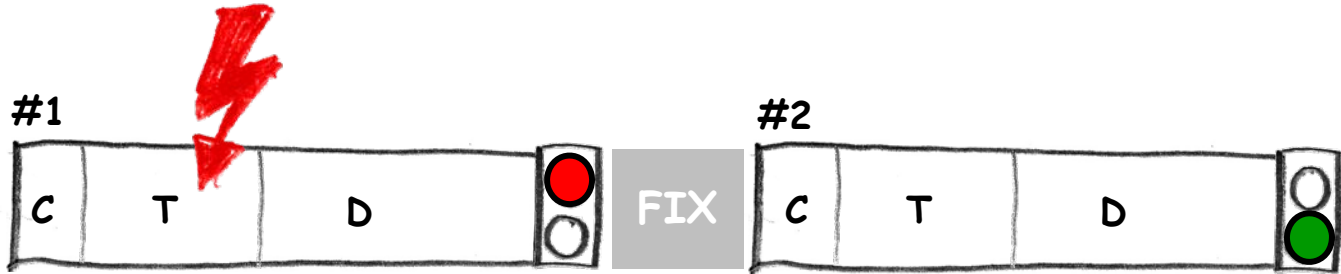
nachher



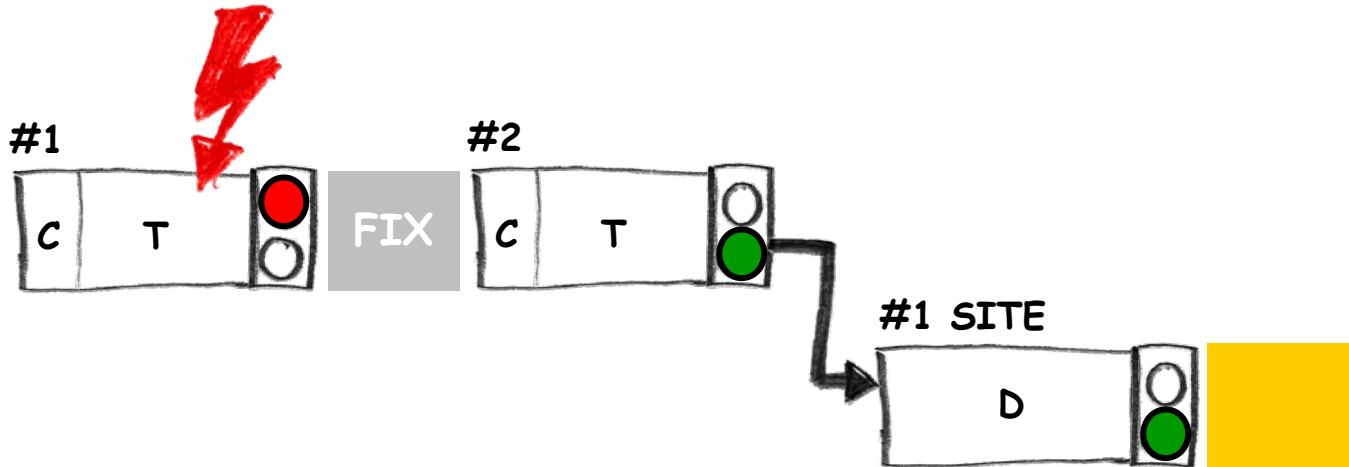
Staffeln: Beispiel



vorher



nachher

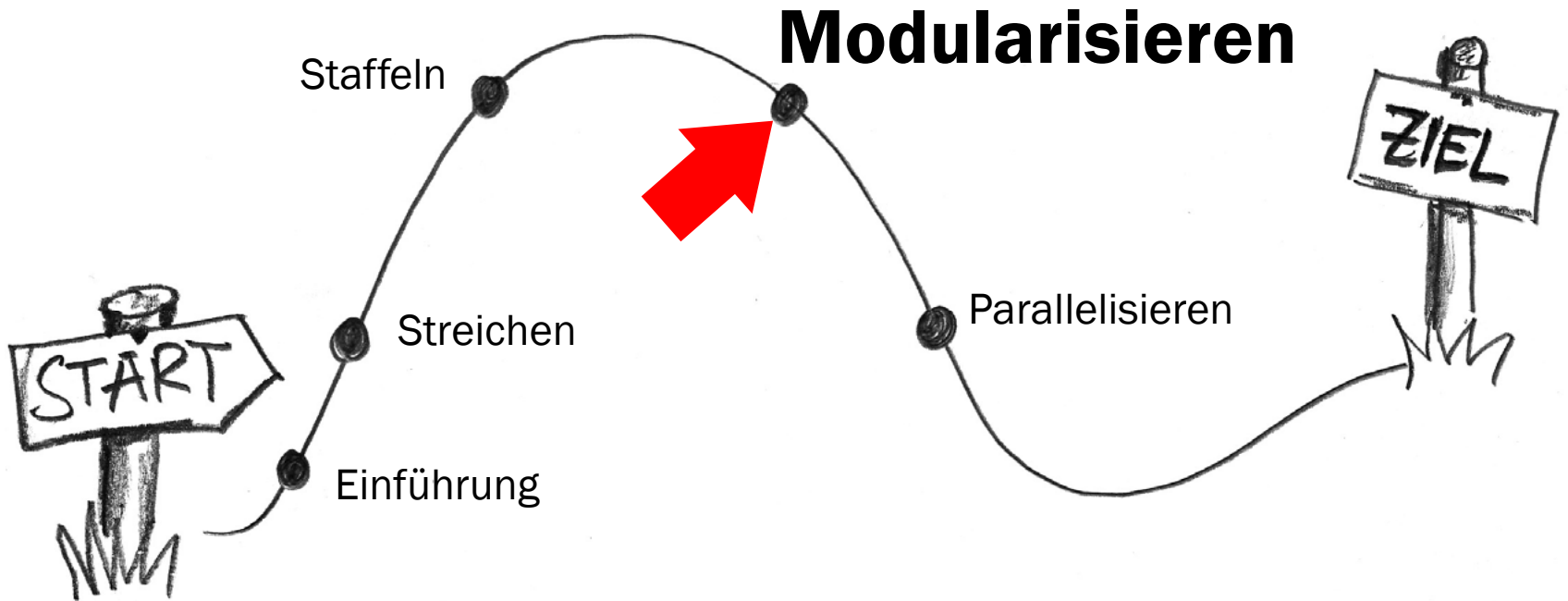


Staffeln: Wie unterstützt hier Hudson?

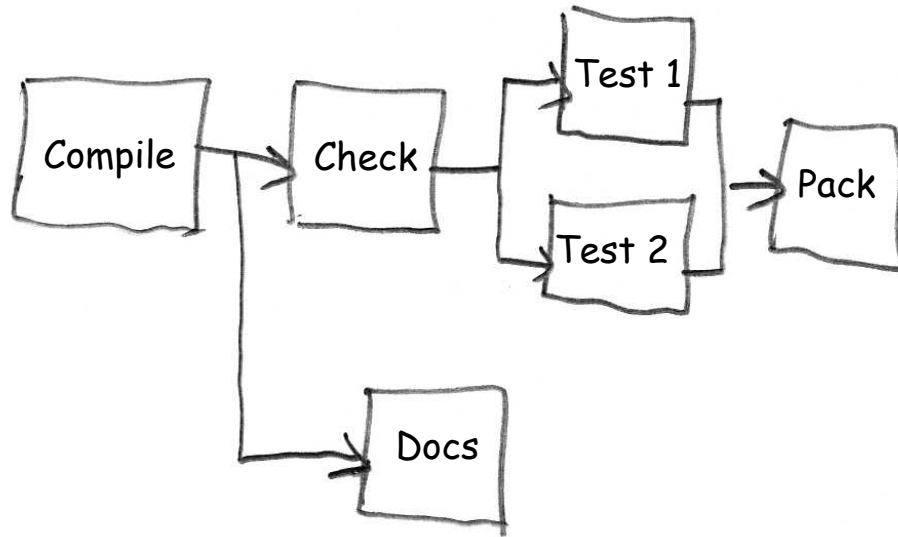
- Projektbeziehungen
(vor-/nachgelagerte Projekte)
- Speicherung von
Fingerabdrücken
- Nachvollziehbarkeit des
Buildprozesses über
verknüpfte Projekte hinweg



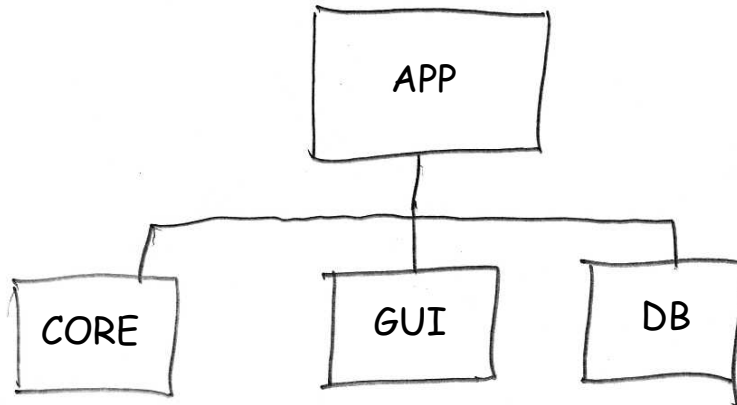
Agenda



Die nächste Ebene: Wir betrachten Module statt Phasen.



Abhängigkeiten
zwischen
(Build-)Phasen

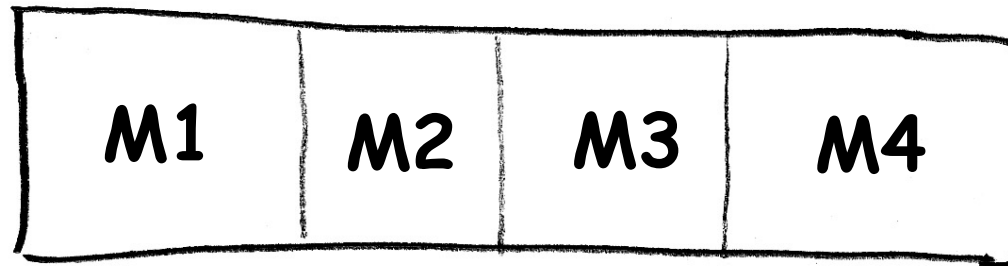


Abhängigkeiten
zwischen **Modulen**

Modularisieren: Was lässt sich wiederverwenden?



#1



vorher



wiederverwenden

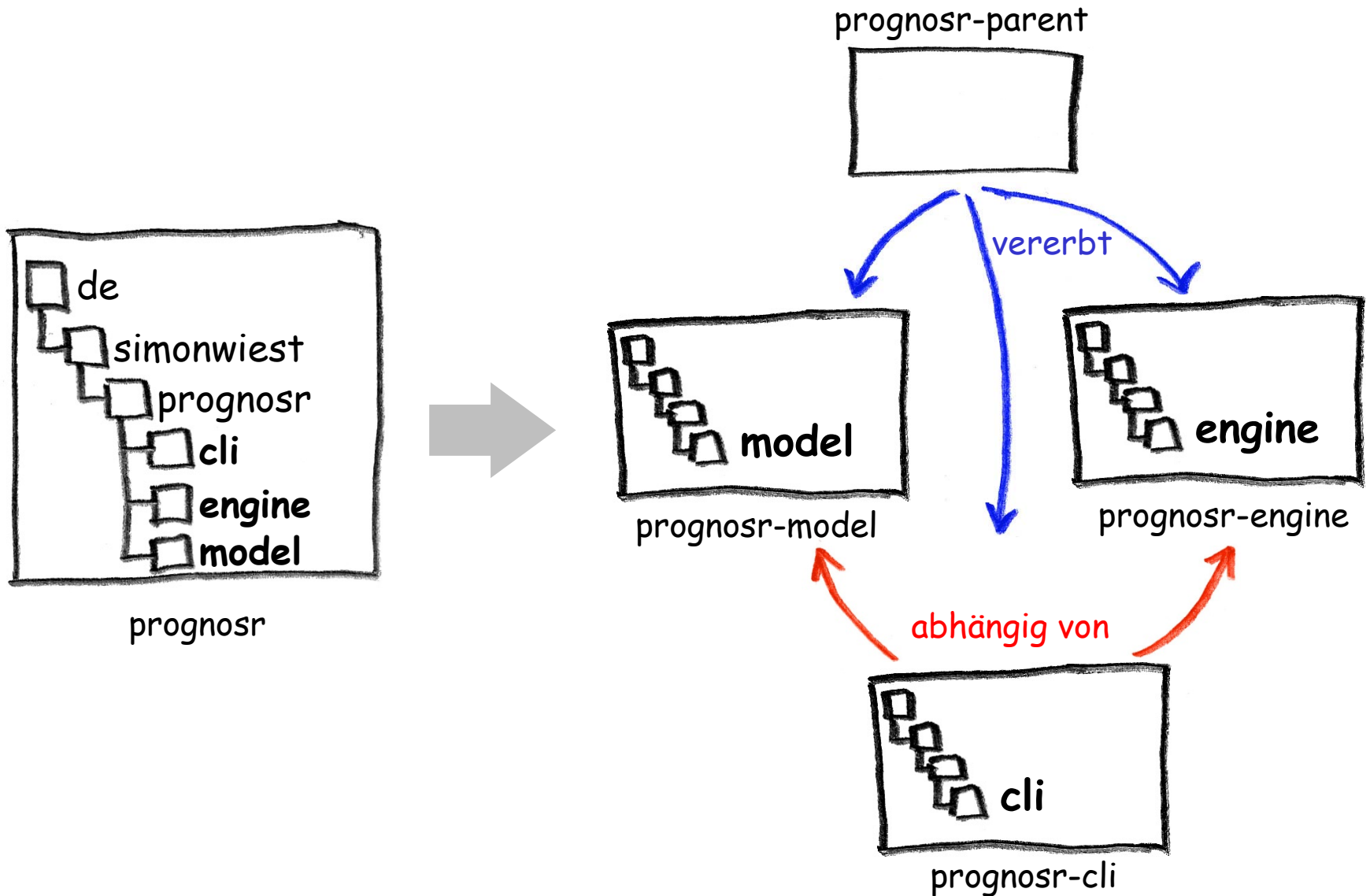
#2



nachher

neu bauen

Modularisieren: Das Projektlayout wird verändert.



Modularisieren: Beispiel



vorher



nachher



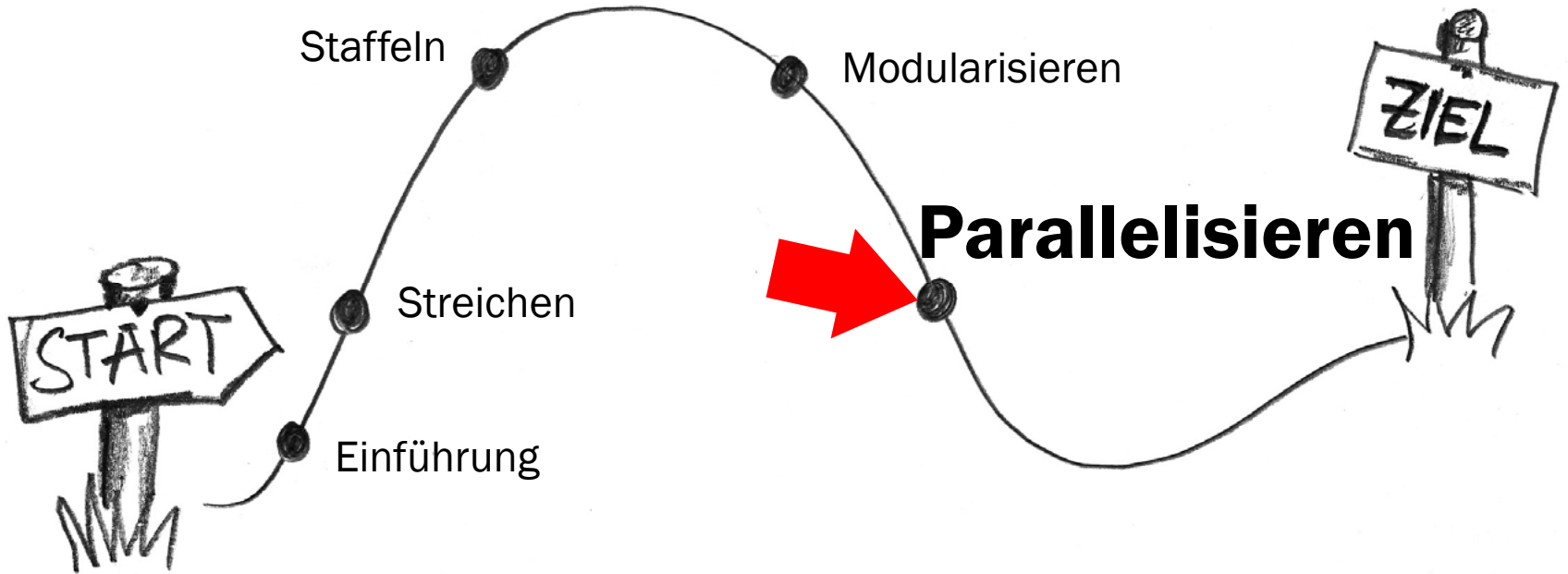
Voraussetzung: Artefakte müssen archiviert werden, z.B. in Maven Repository.

Modularisieren: Wie unterstützt hier Hudson?

- Direkte Unterstützung des Maven-Modulkonzeptes
- Verwendungsnachweis der Build-Ergebnisse
- Filterung der angezeigten Projekte über reguläre Ausdrücke in Ansichten



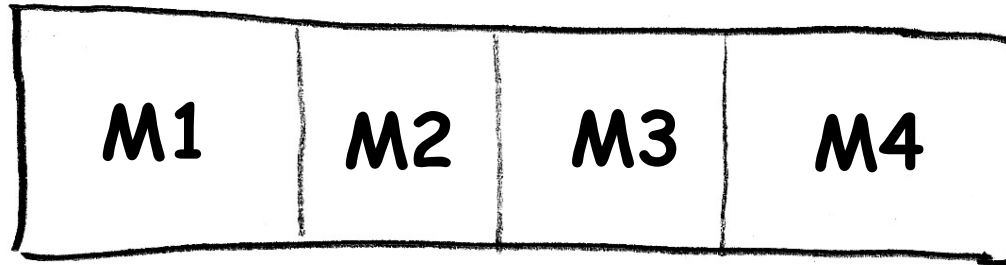
Agenda



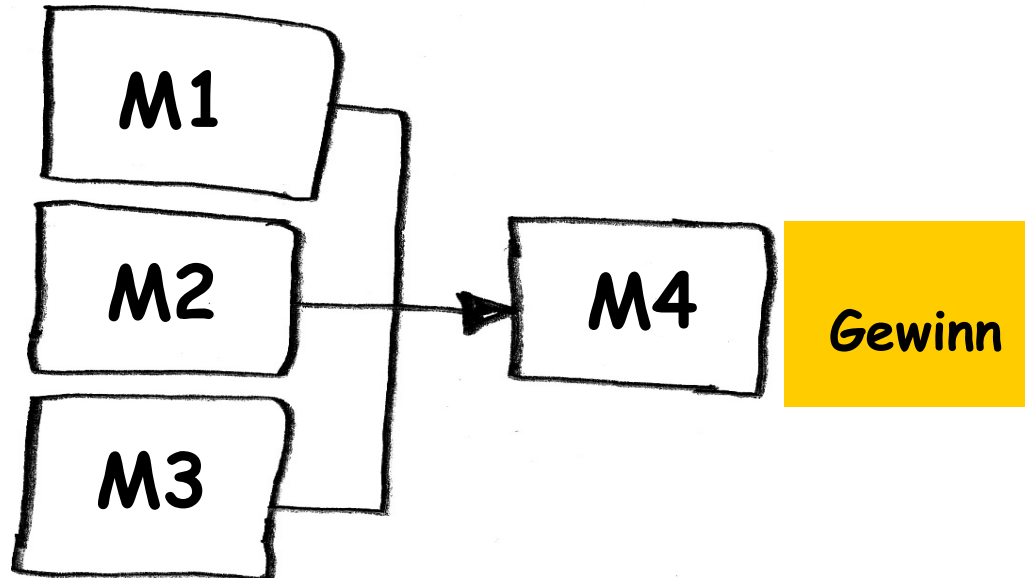
Parallelisieren: Was lässt sich gleichzeitig bauen?



vorher



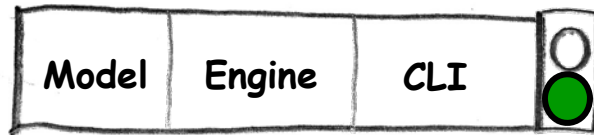
nachher



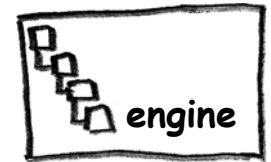
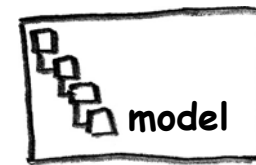
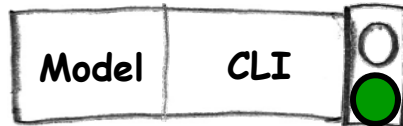
Parallelisieren: Beispiel



vorher



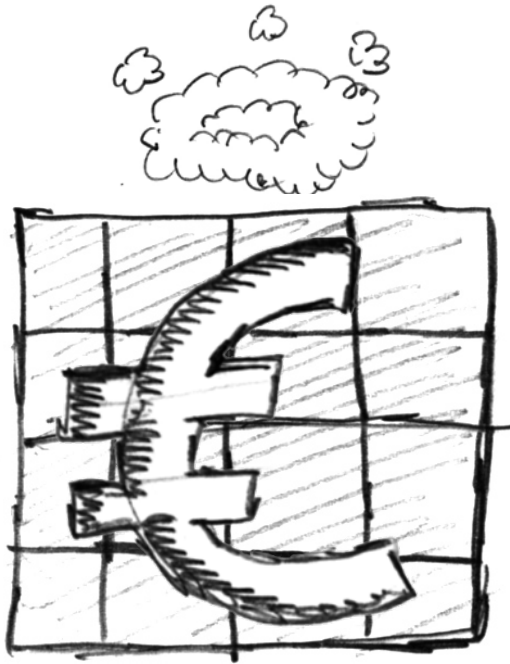
nachher



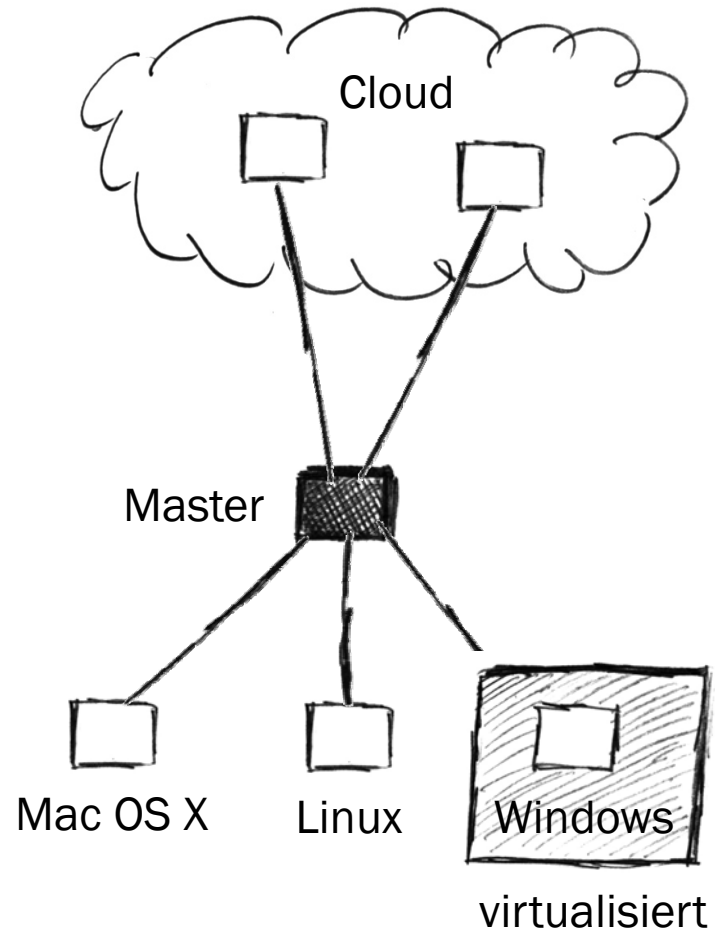
abhängig von



Parallelisieren: Verteilte Builds sind meist wirtschaftlicher.



lokale Builds



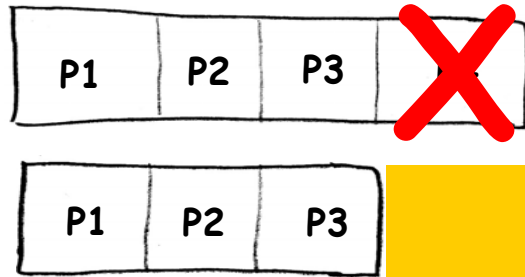
verteilte Builds

Parallelisieren: Wie unterstützt hier Hudson?

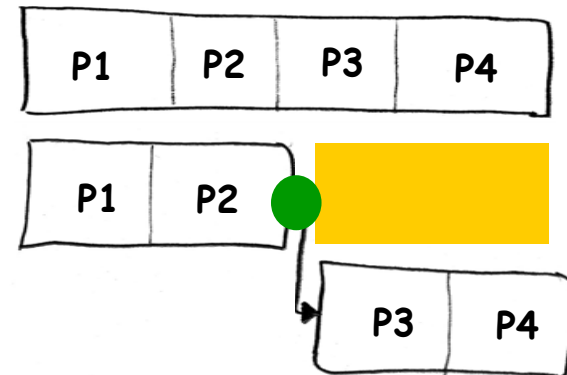
- Master-/Slave-Architektur
- Einfache Konfiguration von Knoten
- Automatischer Start/Stopp von Slave-Knoten
- Zusammenfasste Darstellung der Ergebnisse verteilter Builds
- Breite Unterstützung von Betriebssystemen und Startmechanismen



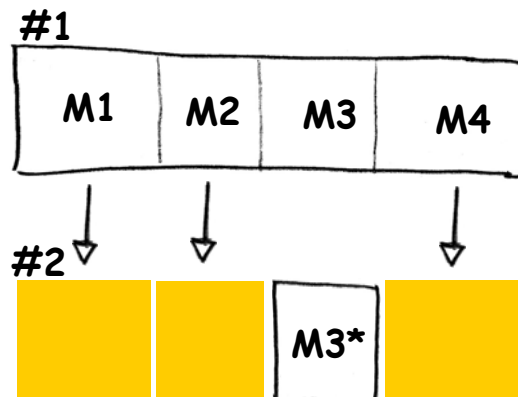
Die vier Strategien zusammengefasst:



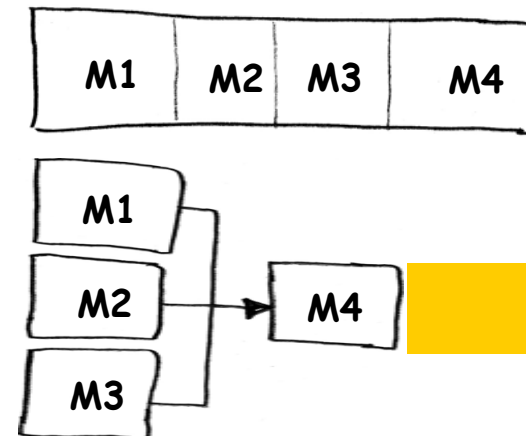
Streichen



Staffeln

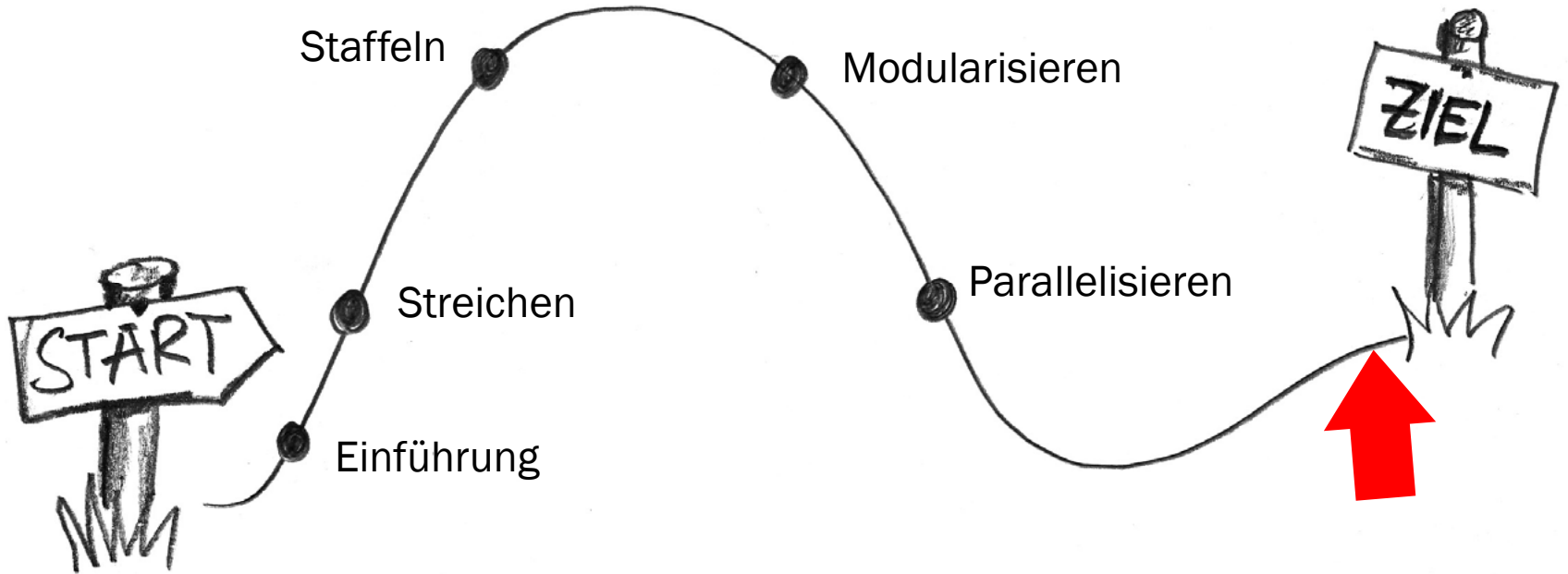


Modularisieren



Parallelisieren

Agenda



HERUNTERLADEN

EINSETZEN

WEITERSAGEN

MITMACHEN

hudson.dev.java.net

Vielen Dank fürs Zuhören.

Dr. Simon Wiest

Ingenieurbüro für Softwaretechnik
Wiesfleckenstrasse 13
72810 Gomaringen
www.simonwiest.de



BACKUP

Empfohlene Nachlese

■ Hudson

- Projektseite. hudson.dev.java.net
- JBoss Hudson CI Server. hudson.jboss.org/hudson
- Interview mit Hudson-Initiator Kohsuke Kawaguchi
blogs.sun.com/glassfishpodcast (Episode #007)

■ Continuous Integration

- White Paper von Martin Fowler. www.martinfowler.com
- CI Feature Matrix. confluence.public.thoughtworks.org
- P.M. Duvall: Continuous Integration. 2007.

■ Projektautomatisierung (allgemein)

- M. Hüttermann: Agile Java-Entwicklung in der Praxis. 2007.
- G. Popp: Konfigurationsmanagement mit SVN, Maven, Redmine. 2009.
- J. F. Smart: Java Power Tools. 2008.

Wie funktionieren die Bären?

Mehr dazu im Hudson Wiki
wiki.hudson-ci.org//x/AQA1AQ

