

# Wenn's mal länger dauert...

## Continuous Integration beschleunigen mit Hudson

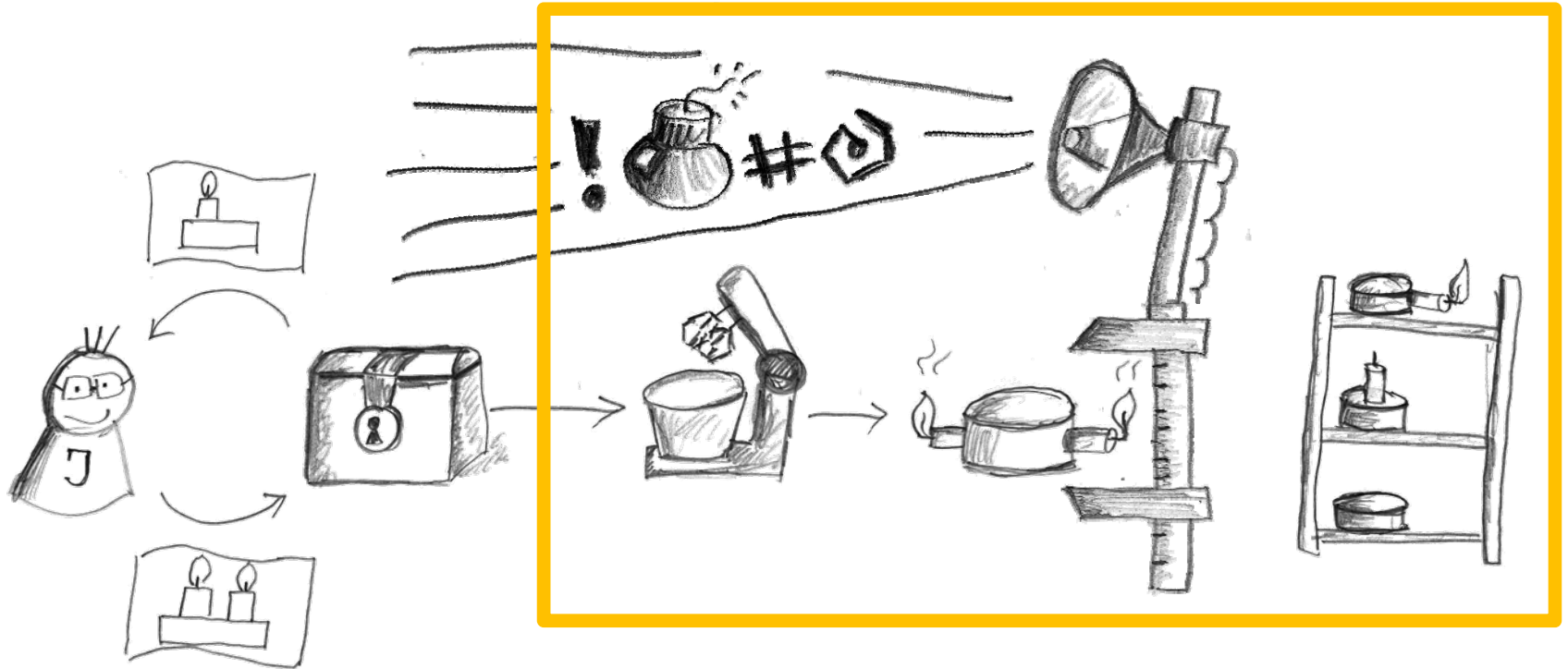
**Dr. Simon Wiest**  
[www.simonwiest.de](http://www.simonwiest.de)

Gearconf 2009  
02.10.2009, Düsseldorf

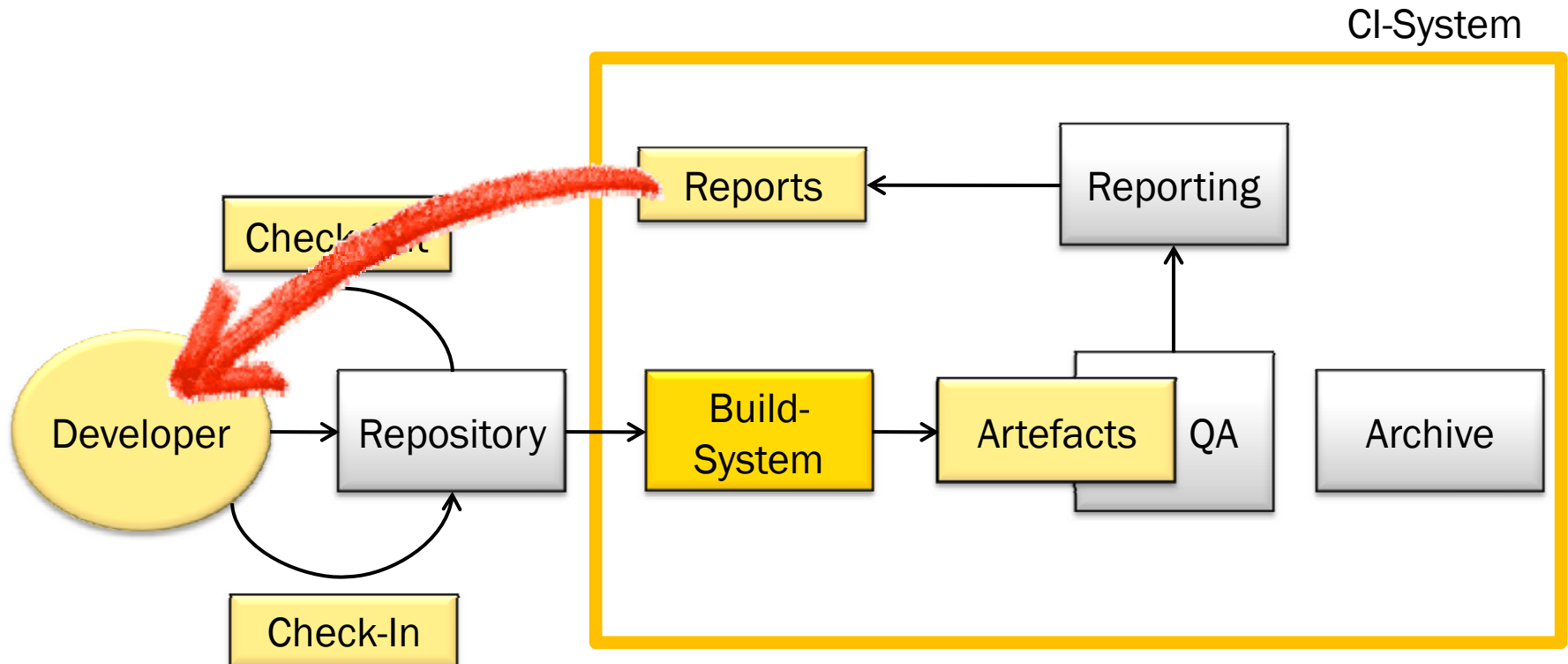


# Was ist kontinuierliche Integration (CI)?

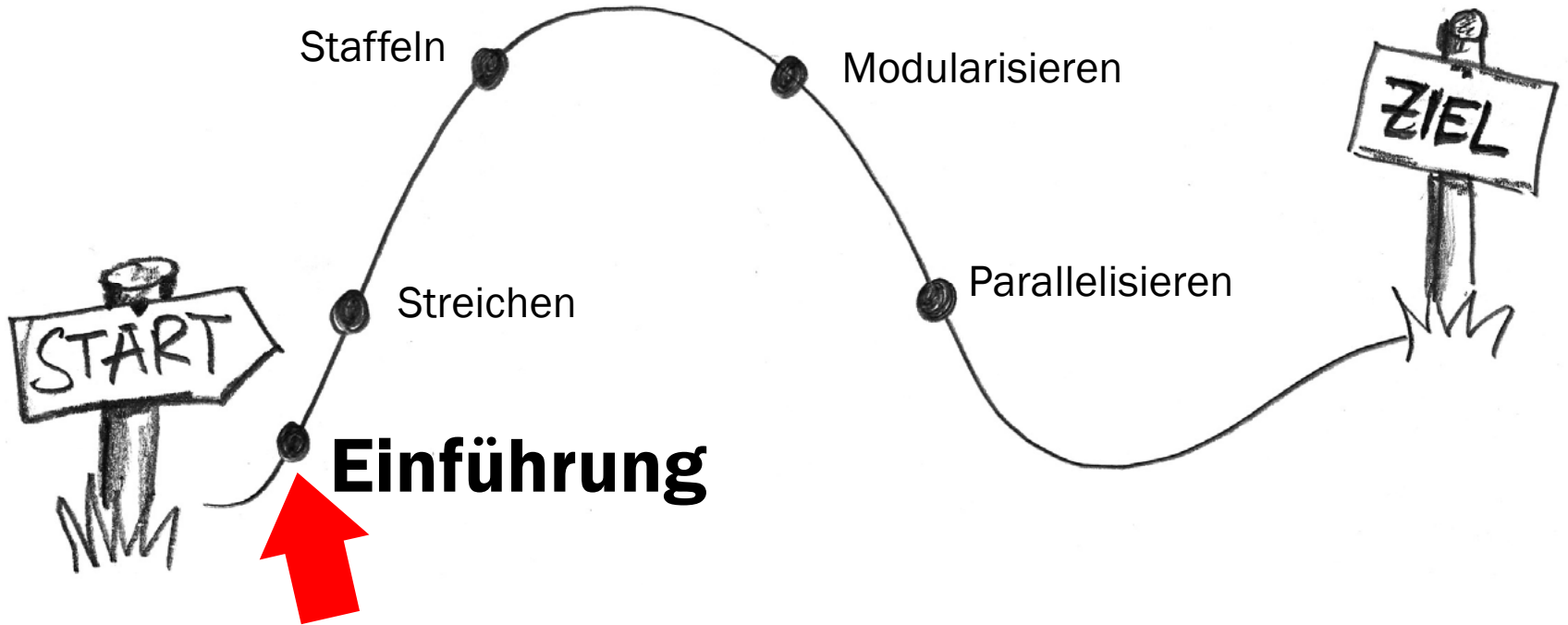
orchestriert durch CI-System



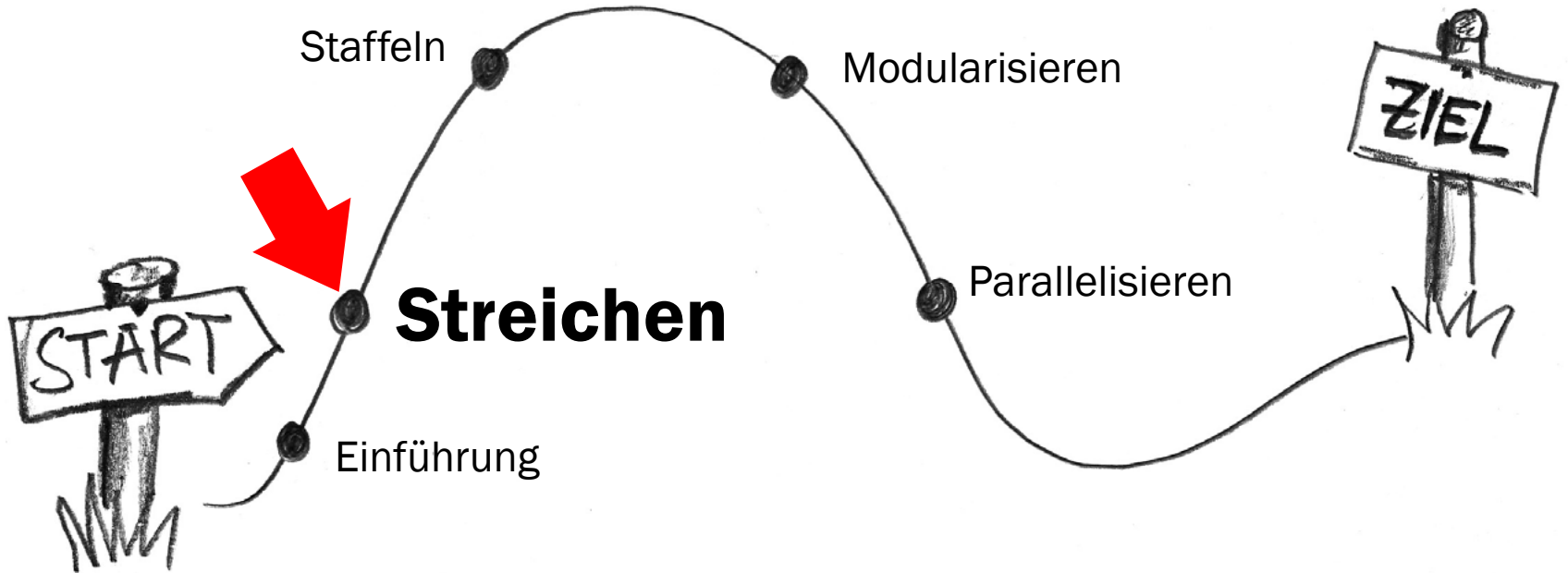
# Was ist kontinuierliche Integration (CI)?



# Agenda



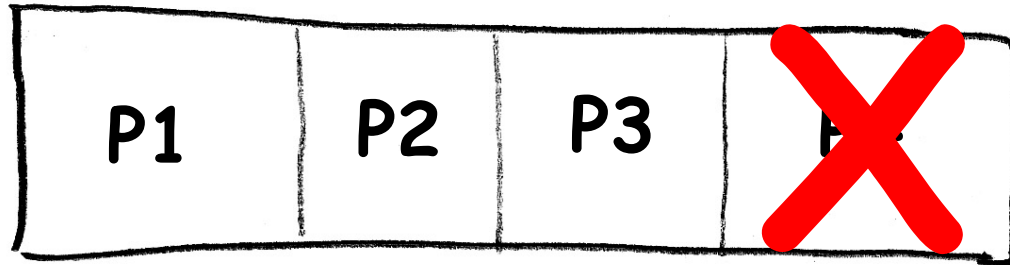
# Agenda



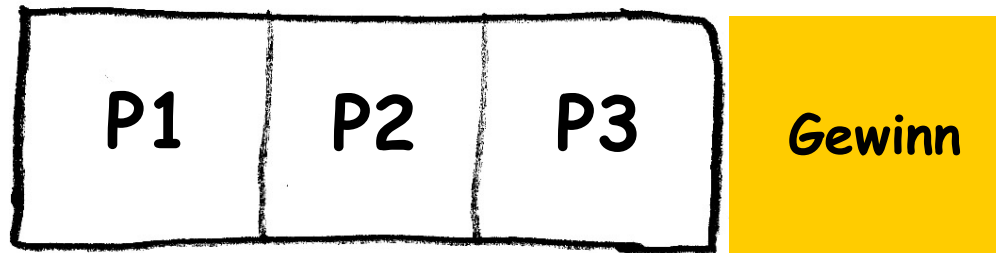
# Streichen: Was kann entfallen?



vorher



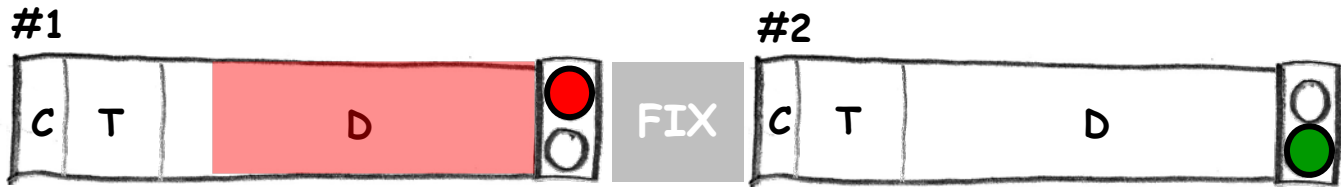
nachher



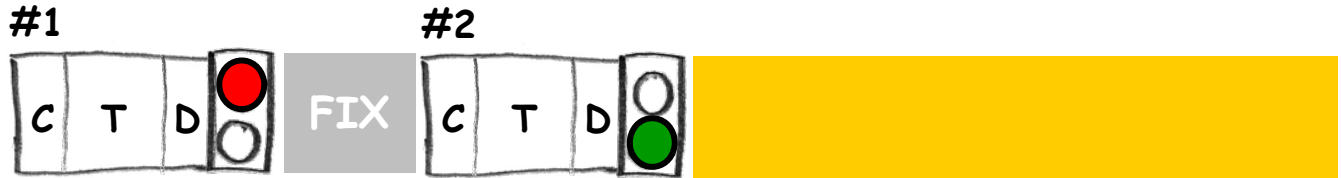
# Streichen: Beispiel



vorher



nachher



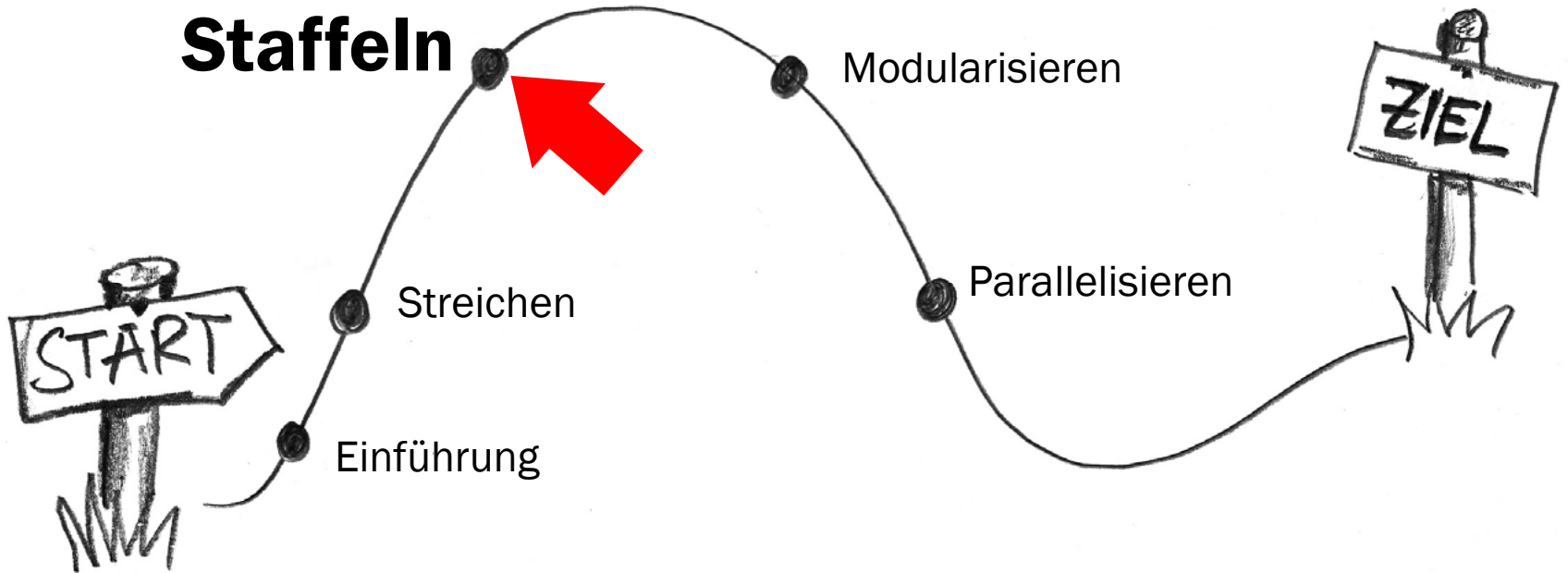
## **Streichen: Wie unterstützt hier Hudson?**

- Trends über Projektverlauf, z.B. Buildzeit
- Intuitive Ad-Hoc-Analyse in Web-Oberfläche





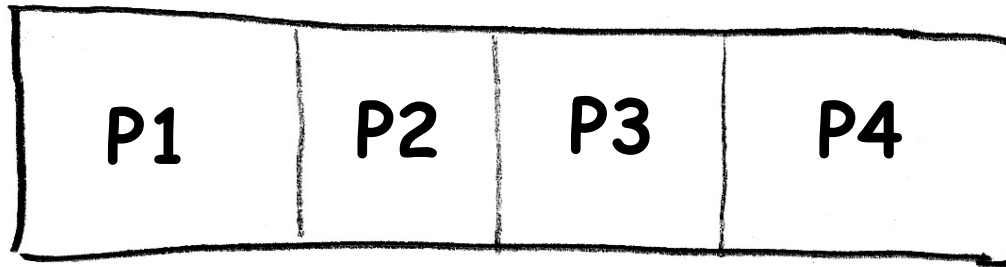
# Agenda



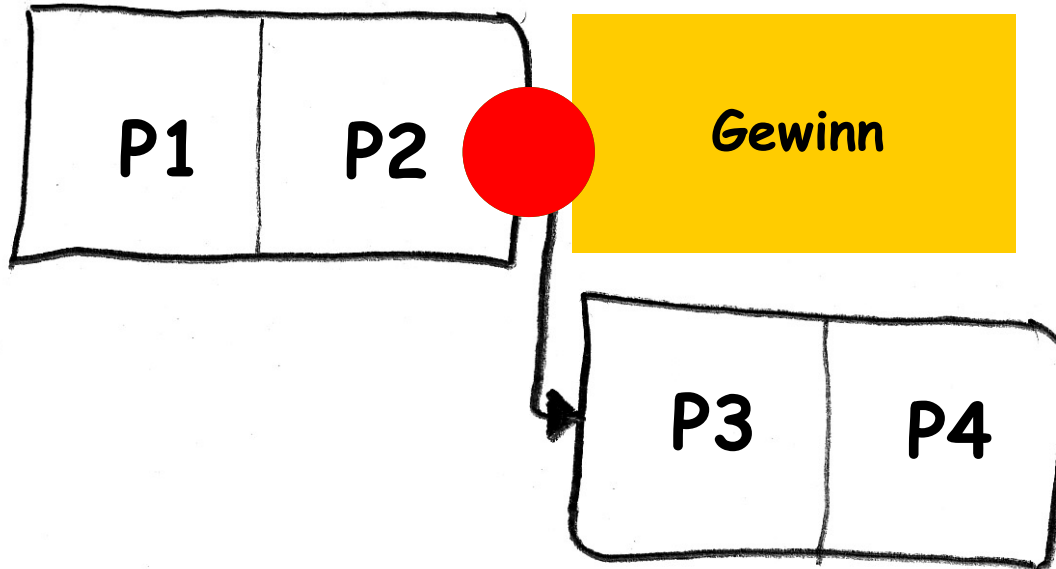
# Staffeln: Das Wichtigste zuerst!



vorher



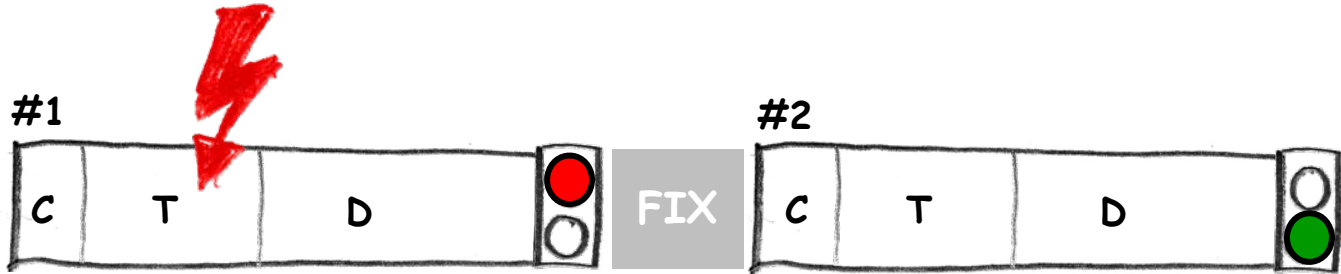
nachher



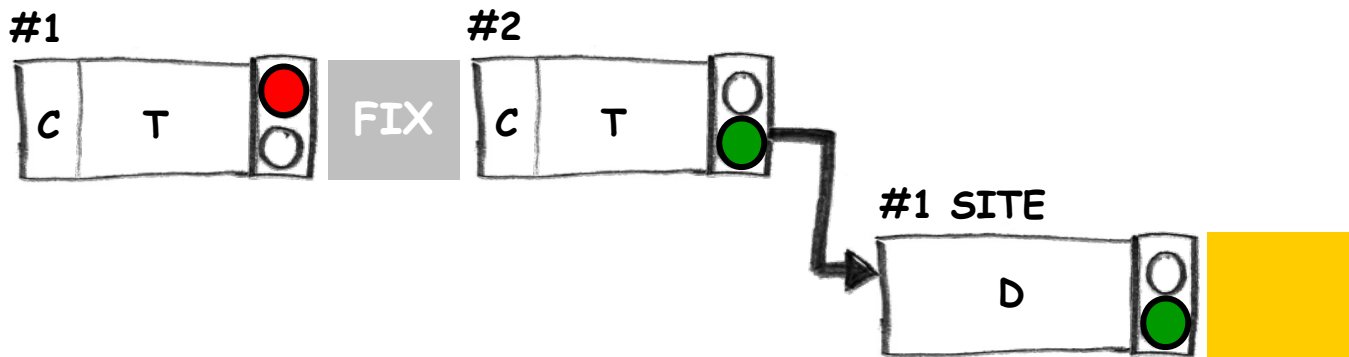
# Staffeln: Beispiel



vorher



nachher

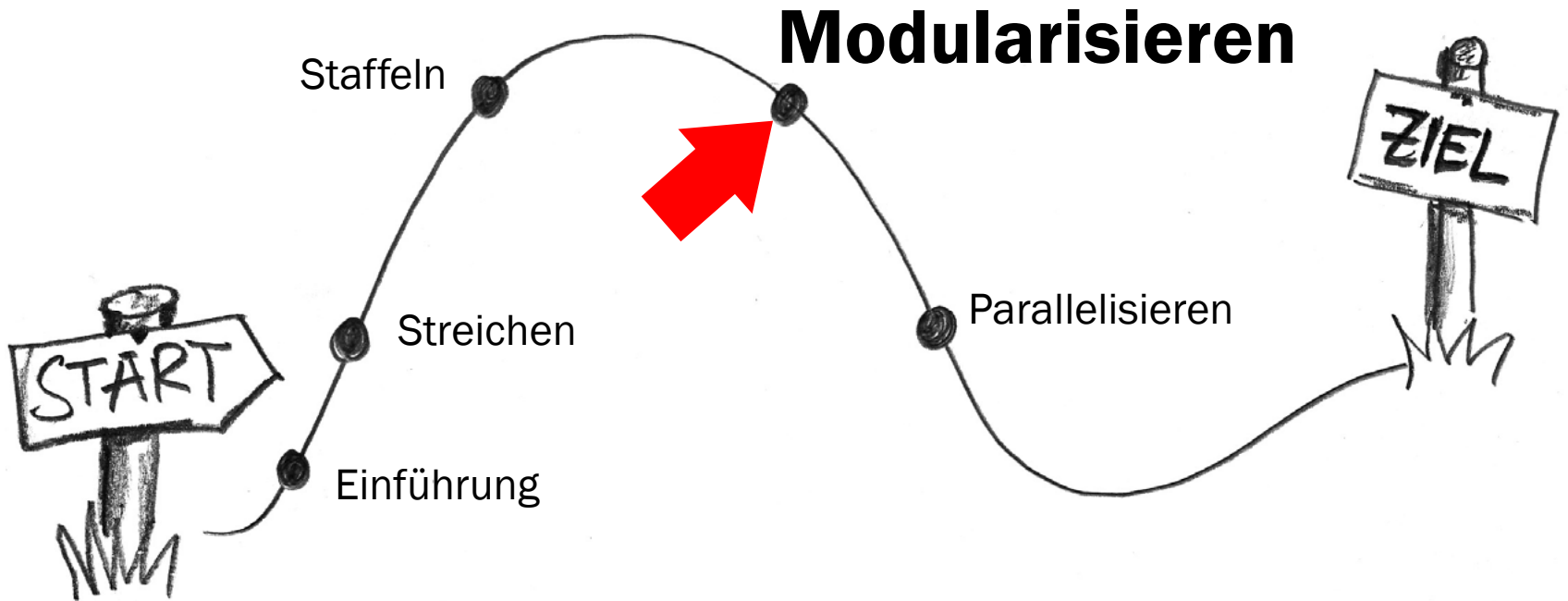


## Staffeln: Wie unterstützt hier Hudson?

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



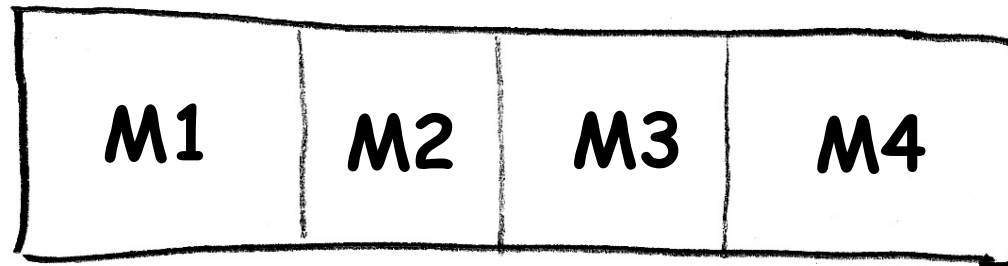
# Agenda



# Modularisieren: Was lässt sich wiederverwenden?



#1



vorher



wiederverwenden

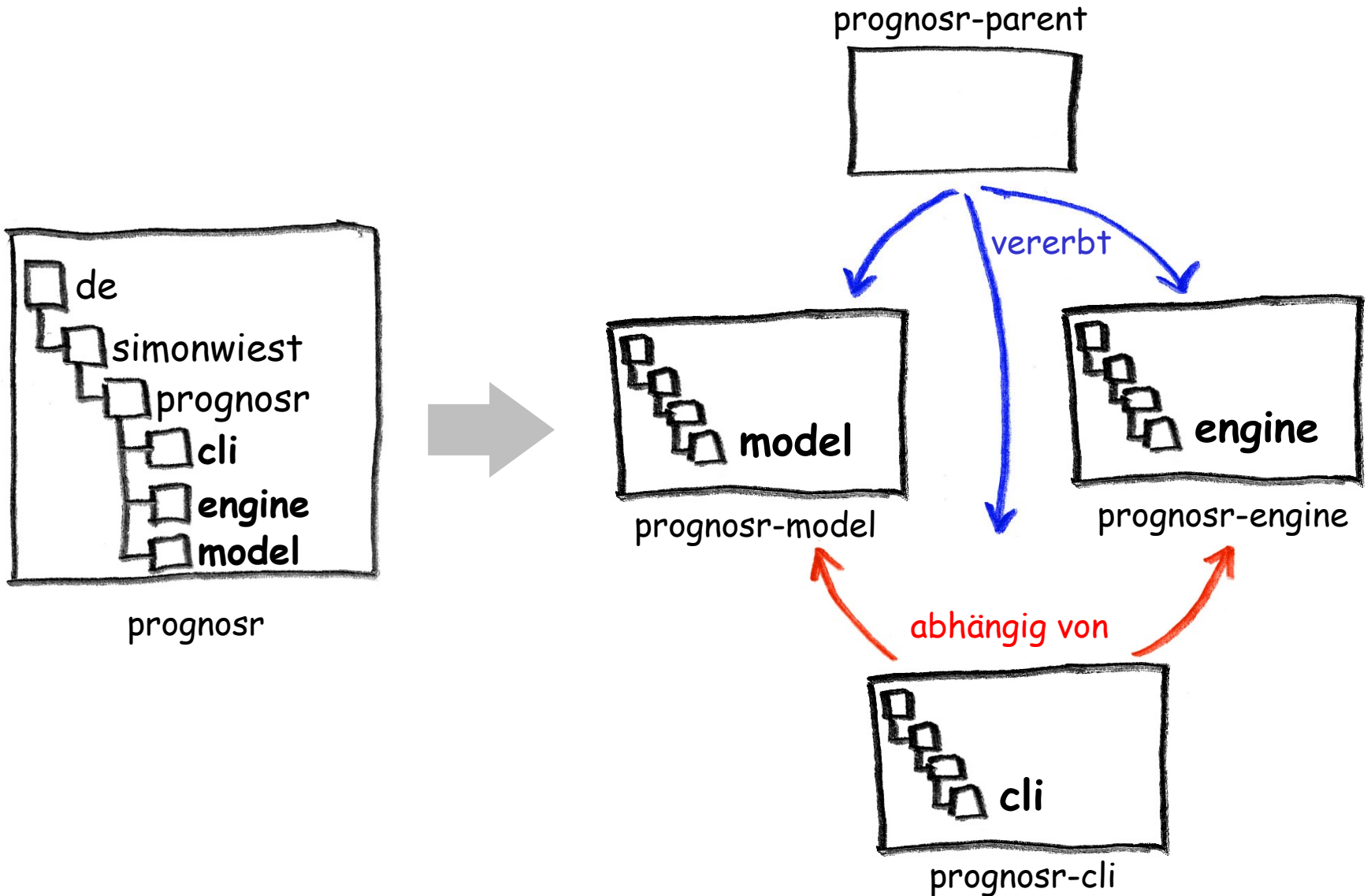
#2



nachher

neu bauen

# Modularisieren: Zerteilen des Monolithen.





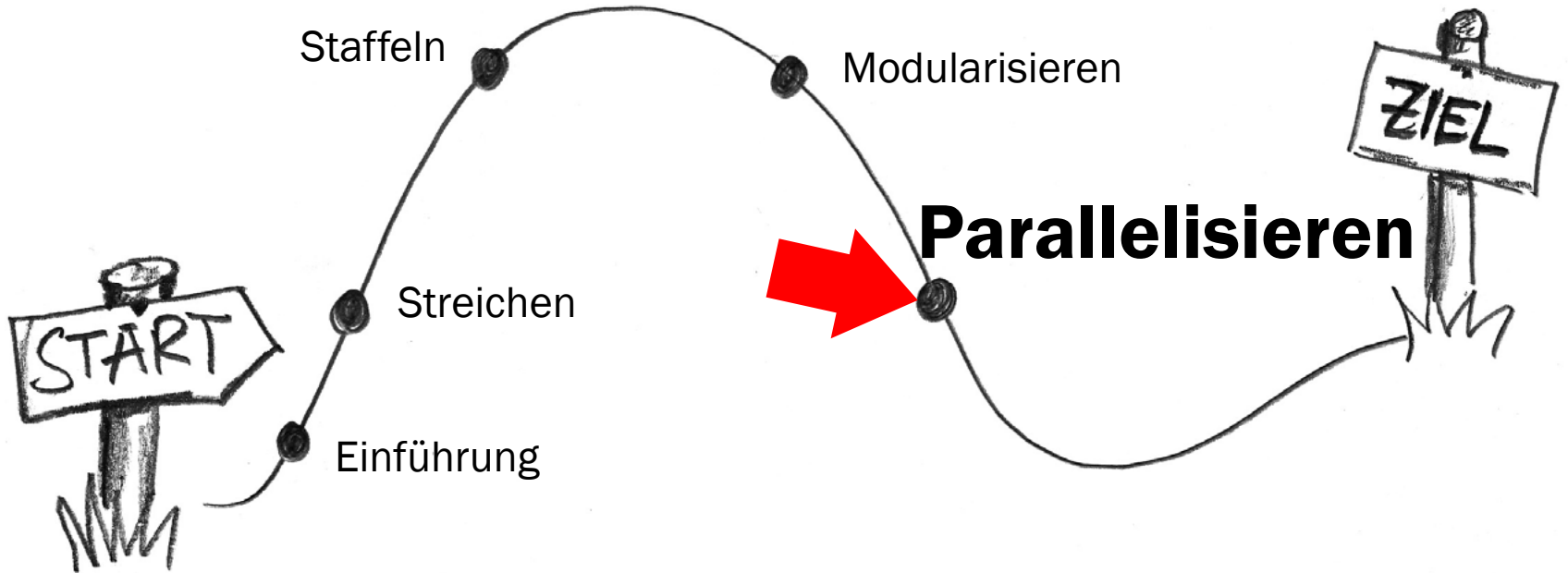


## Modularisieren: Wie unterstützt hier Hudson?

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



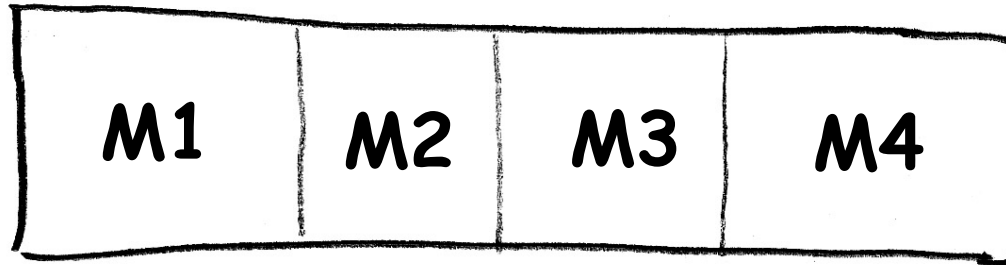
# Agenda



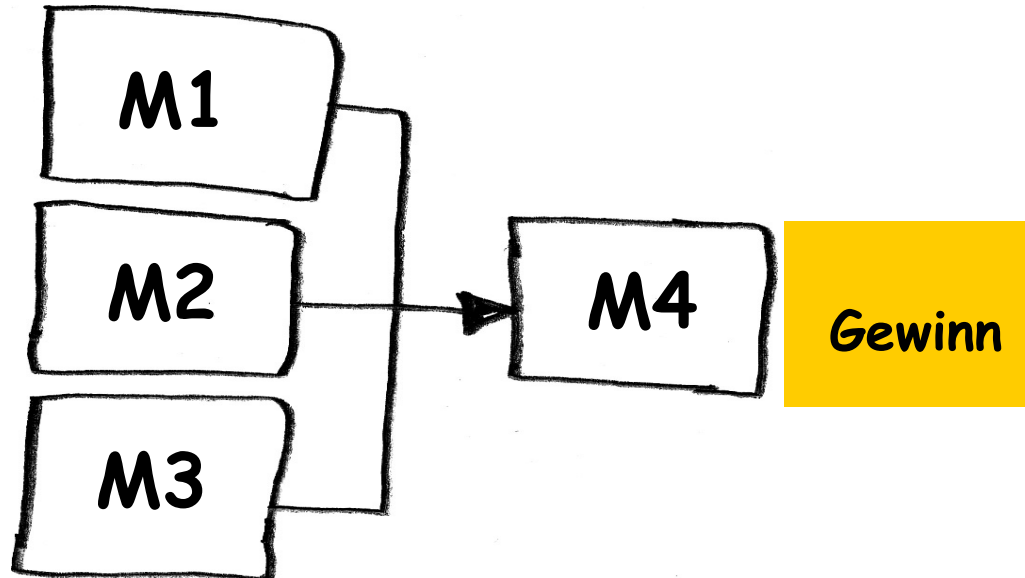
# Parallelisieren: Was lässt sich gleichzeitig bauen?



vorher



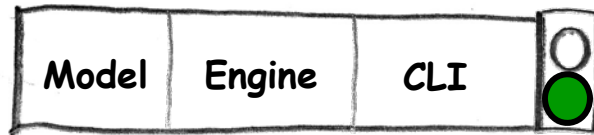
nachher



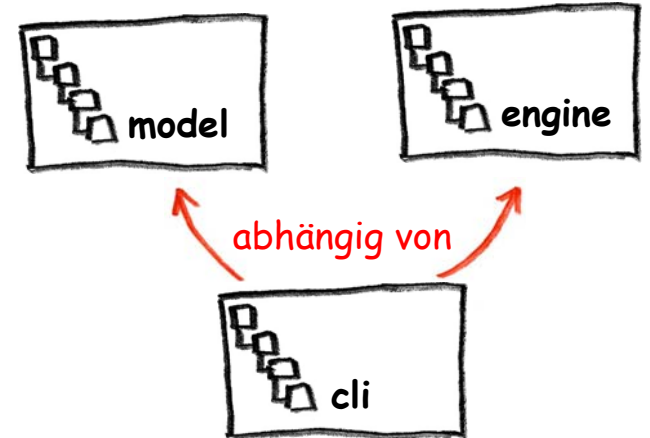
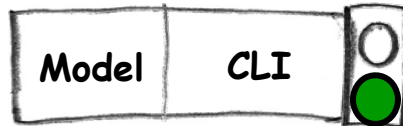
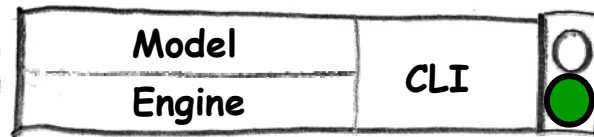
# Parallelisieren: Beispiel



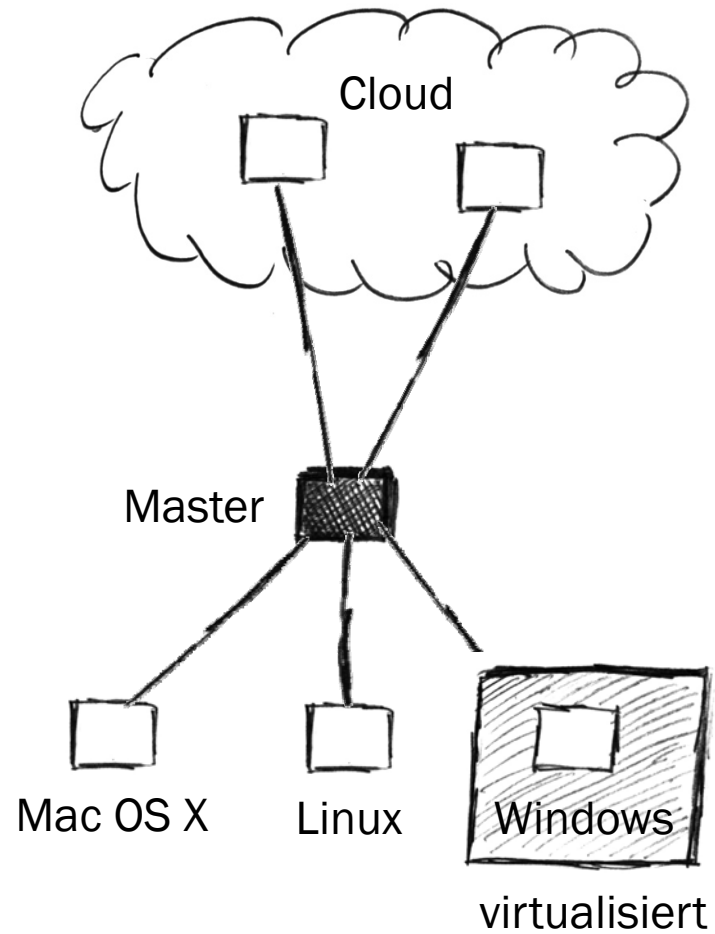
vorher



nachher



# Parallelisieren: Flexibel und wirtschaftlich durch Verteilen.

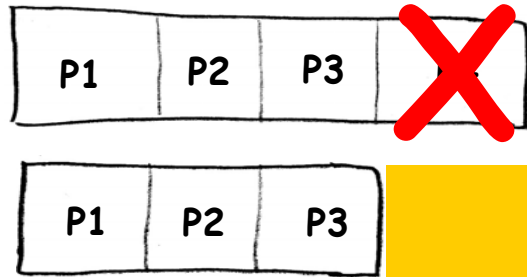


## Parallelisieren: Wie unterstützt hier Hudson?

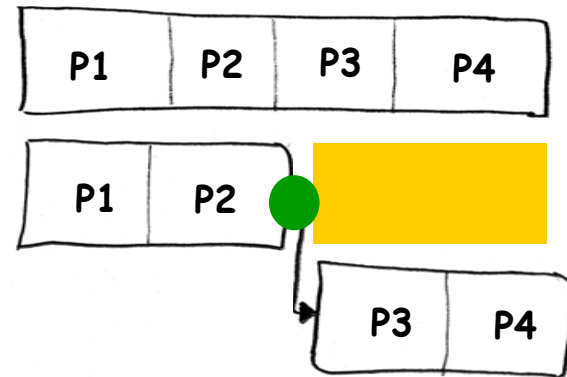
- Master-/Slave-Architektur
- Einfache Konfiguration von Knoten
- Automatischer Start/Stopp von Slave-Knoten
- Zusammenfasste Darstellung der Ergebnisse verteilter Builds
- Breite Unterstützung von Betriebssystemen und Startmechanismen



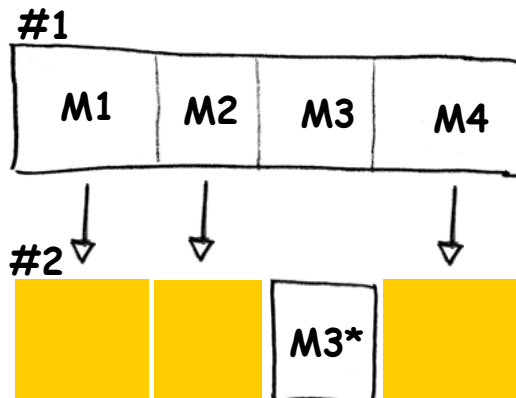
# Die vier Strategien zusammengefasst:



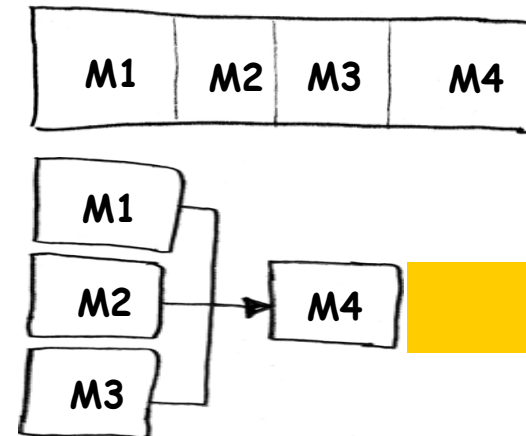
Streichen



Staffeln

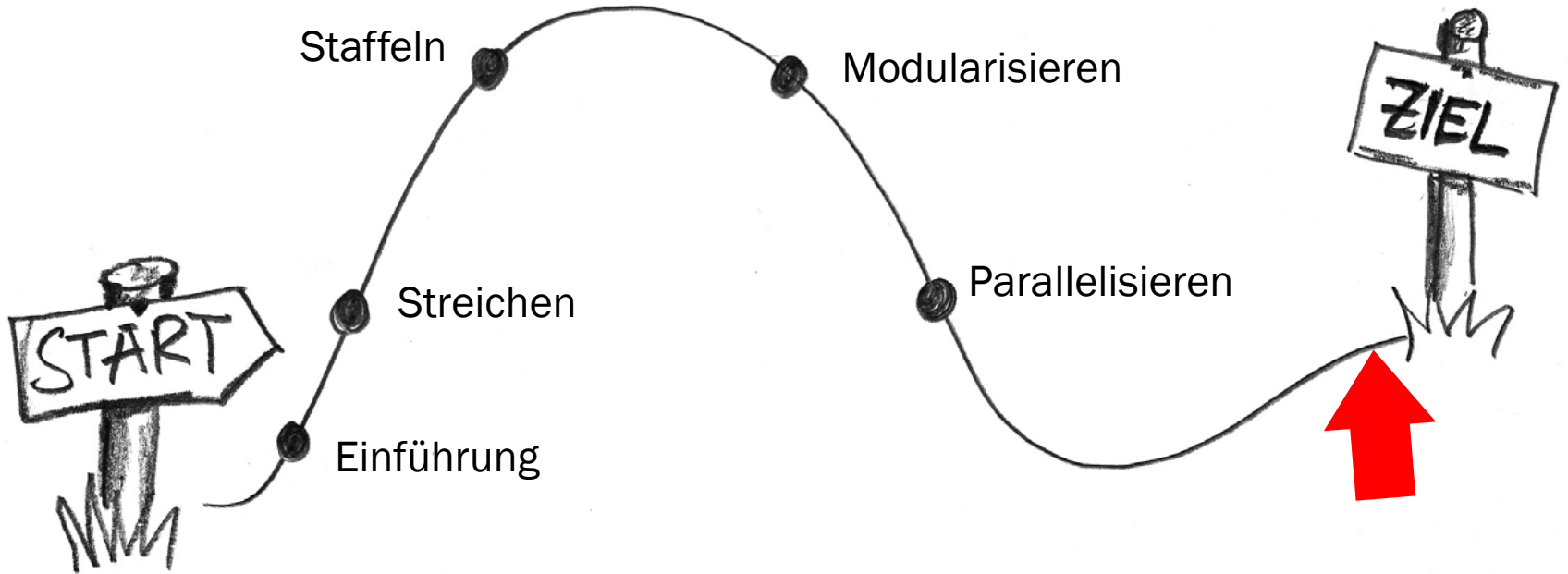


Modularisieren



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](http://www.simonwiest.de)



# Empfohlene Nachlese

## ■ Hudson

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

## ■ Continuous Integration

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

## ■ Projektautomatisierung (allgemein)

- J. F. Smart: Java Power Tools. 2008.
- M. Hüttermann: Agile Java-Entwicklung in der Praxis. 2007.

**BACKUP**

# Wie funktionieren die Bären?

Mehr dazu im Hudson Wiki  
[wiki.hudson-ci.org//x/AQA1AQ](http://wiki.hudson-ci.org//x/AQA1AQ)

