

Liebling, ich hab' den Build geschrumpft!

Endlich wieder schnellere Builds mit Hudson

Dr. Simon Wiest

www.simonwiest.de

Java Forum Stuttgart 2010
01.07.2010, Stuttgart





compile unit tests 





compile

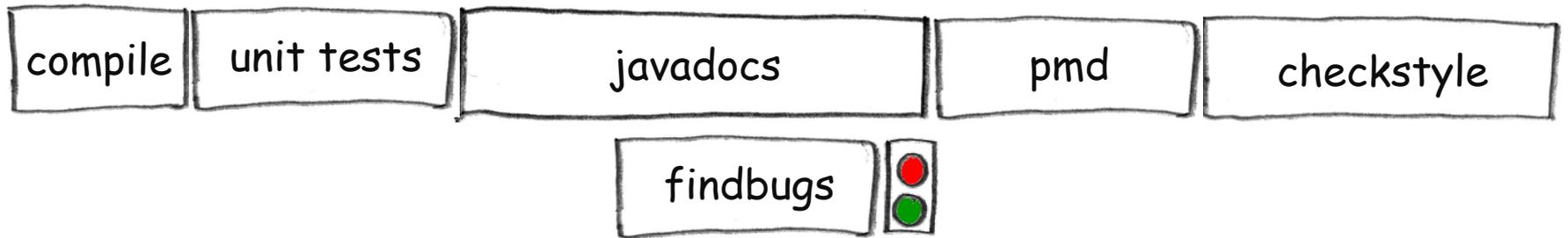
unit tests

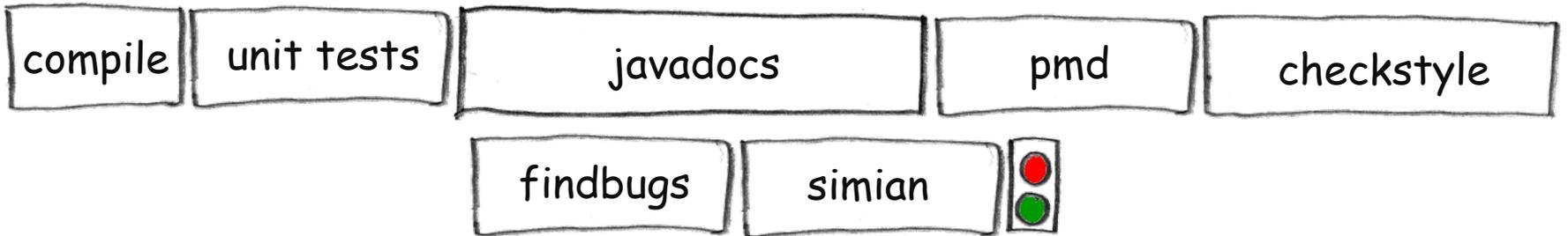
javadocs

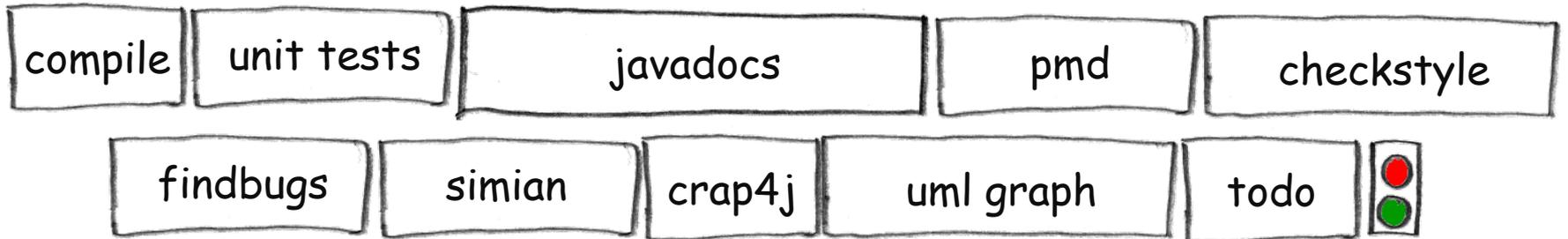
pmd

checkstyle









preprocess

codegen

magic

stamp

compile

unit tests

javadocs

pmd

checkstyle

findbugs

simian

crap4j

uml graph

todo

sign

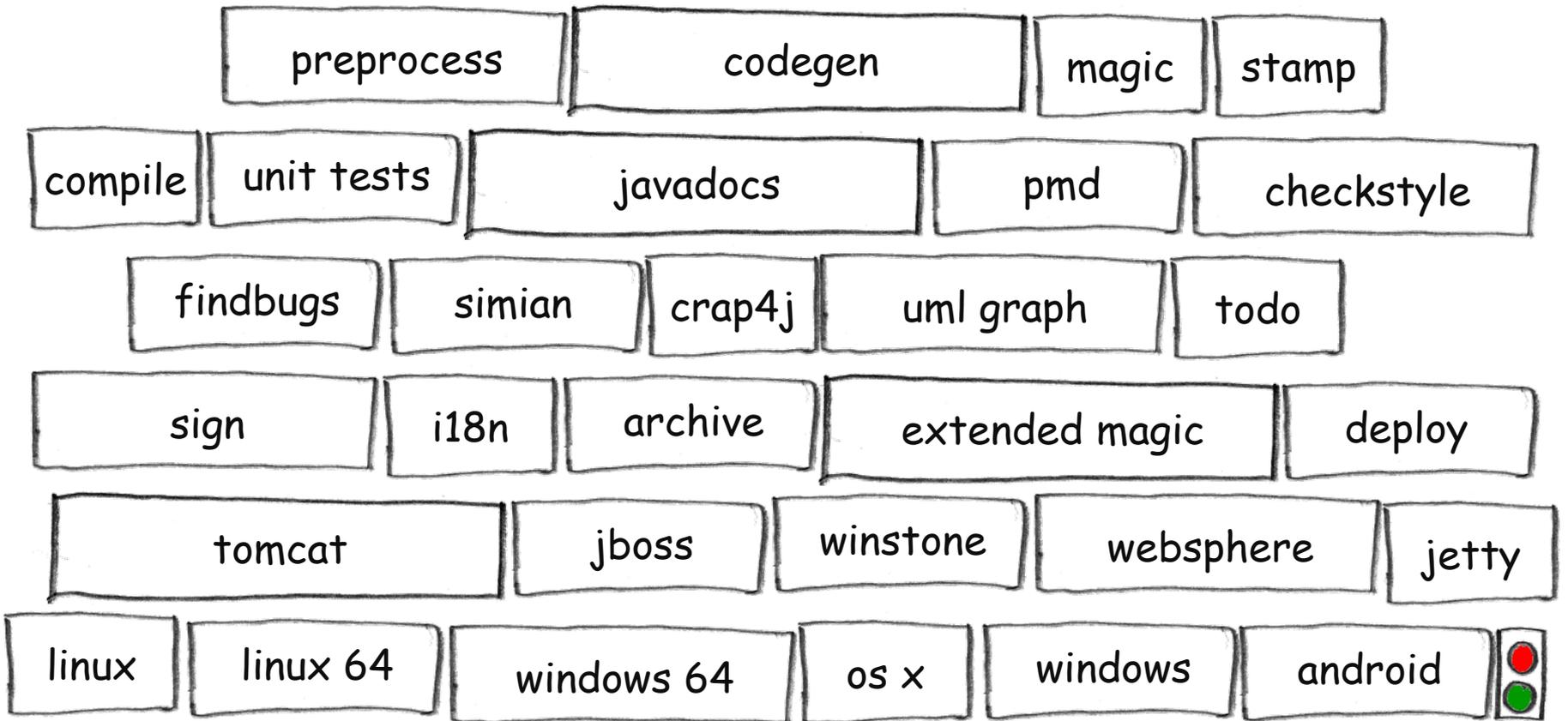
i18n

archive

extended magic

deploy





clean preprocess codegen magic stamp

compile unit tests javadocs pmd checkstyle

findbugs simian crap4j uml graph todo

sign i18n archive extended magic deploy

tomcat jboss winstone websphere jetty

linux linux 64 windows 64 os x windows android

oracle db2 ms sql mysql

postgresql upload twitter notifications feeds gui 

clean

preprocess

codegen

magic

stamp

compile

unit tests

javadocs

pmd

checkstyle

findbugs

simian

crap4j

uml graph

todo

sign

i18n

archive

extended magic

deploy

tomcat

jboss

winstone

websphere

jetty

linux

linux 64

windows 64

os x

windows

android

oracle

db2

ms sql

mysql

postgresql

upload

twitter

notifications

feeds

gui

backend

rule engine

cactus

legalese

tag branch

encrypt

trigger downstreams

kitchen

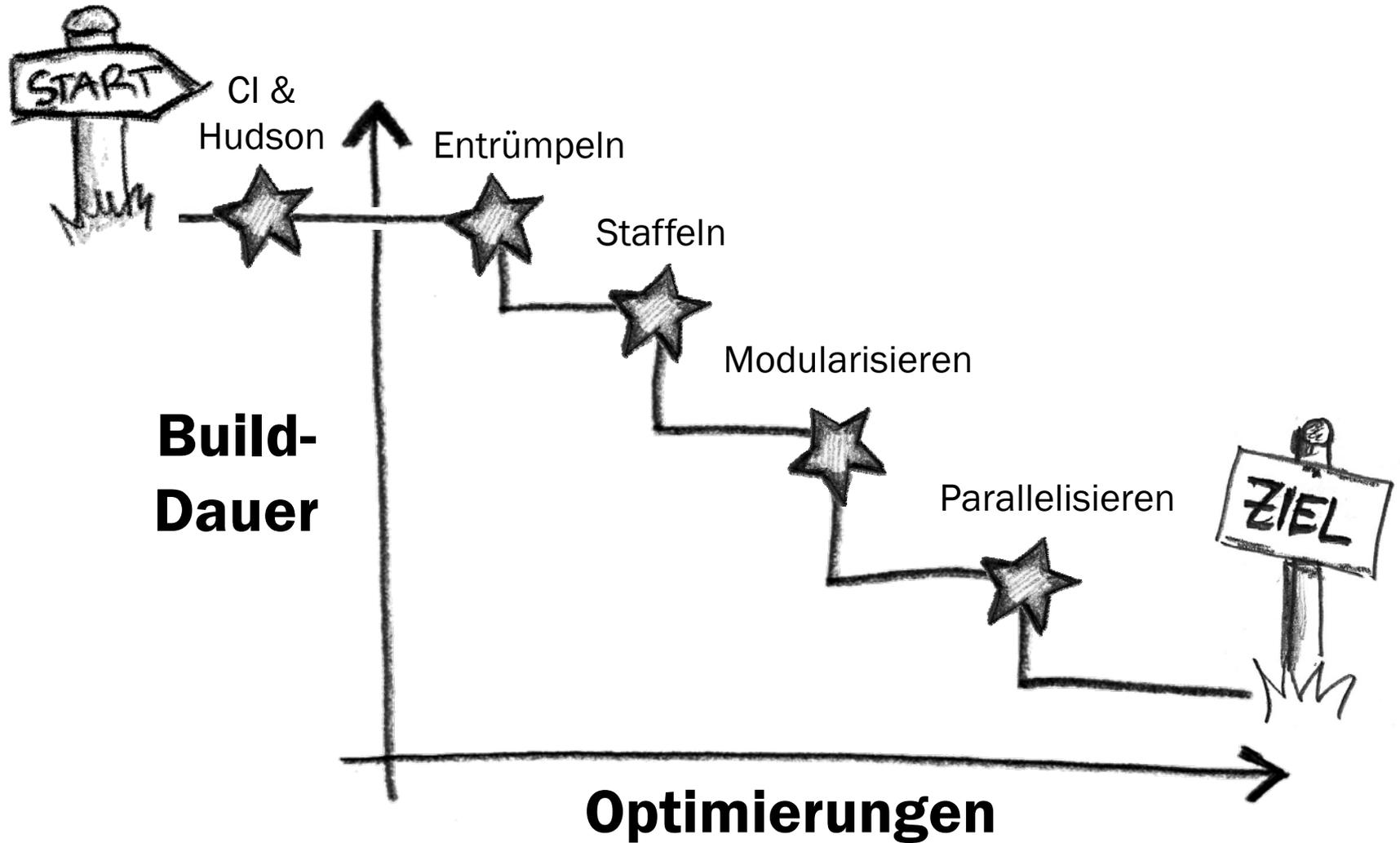
sink



~~Unser Build dauert
zu lange für CI!~~

CI macht unsere
Builds wieder schnell!

Agenda



Ihr Reiseleiter: Dr. Simon Wiest

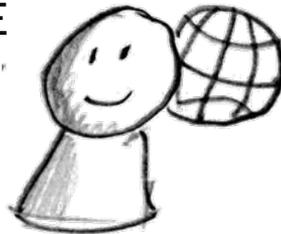
Privat: Hudson-Committer



Lokalisierung DE



Plugin Entwickler



Internationalisierung

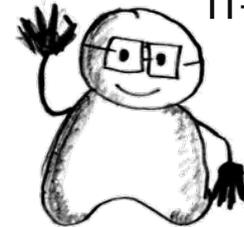


Supporter, Evangelist
(u.a. Buchautor)

Beruflich: Hudson-Anwender



IT-Projektleiter



Java Architekt

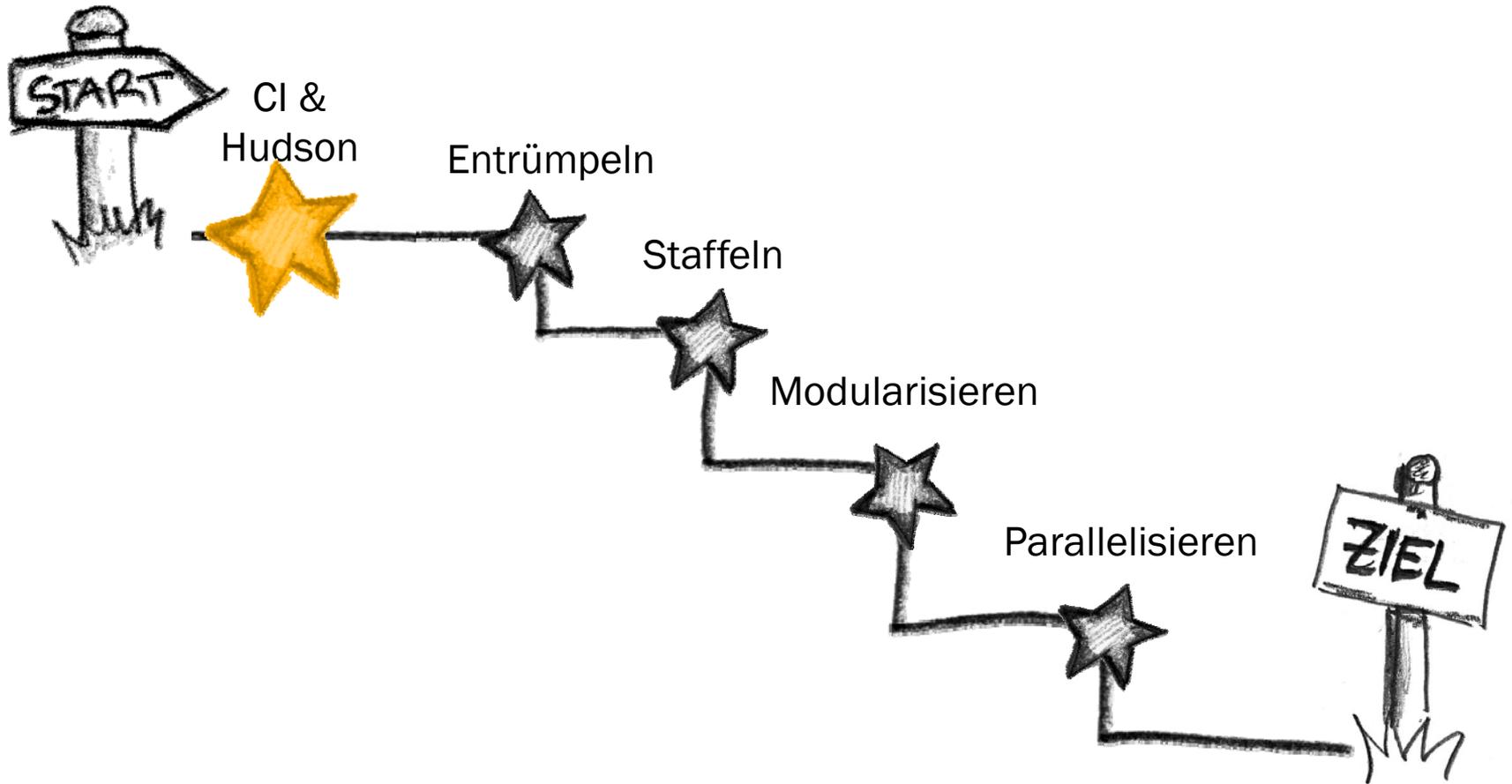


CI-Coach

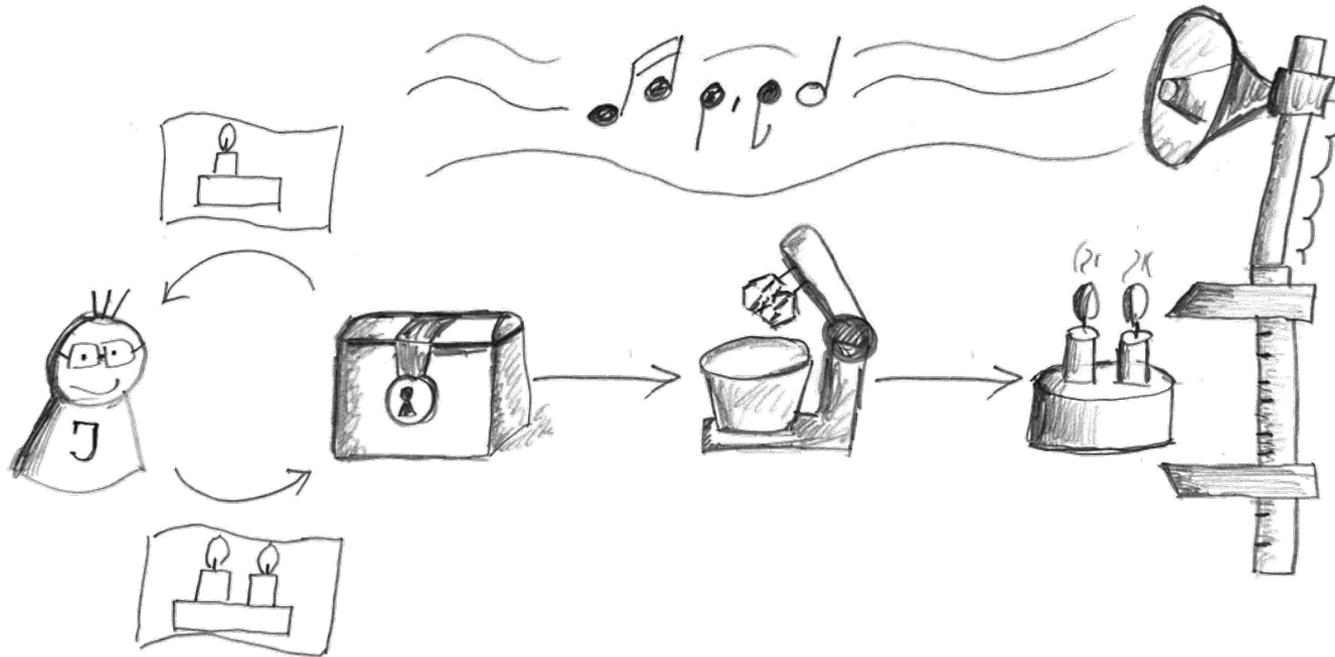


IT-Freiberufler

Agenda

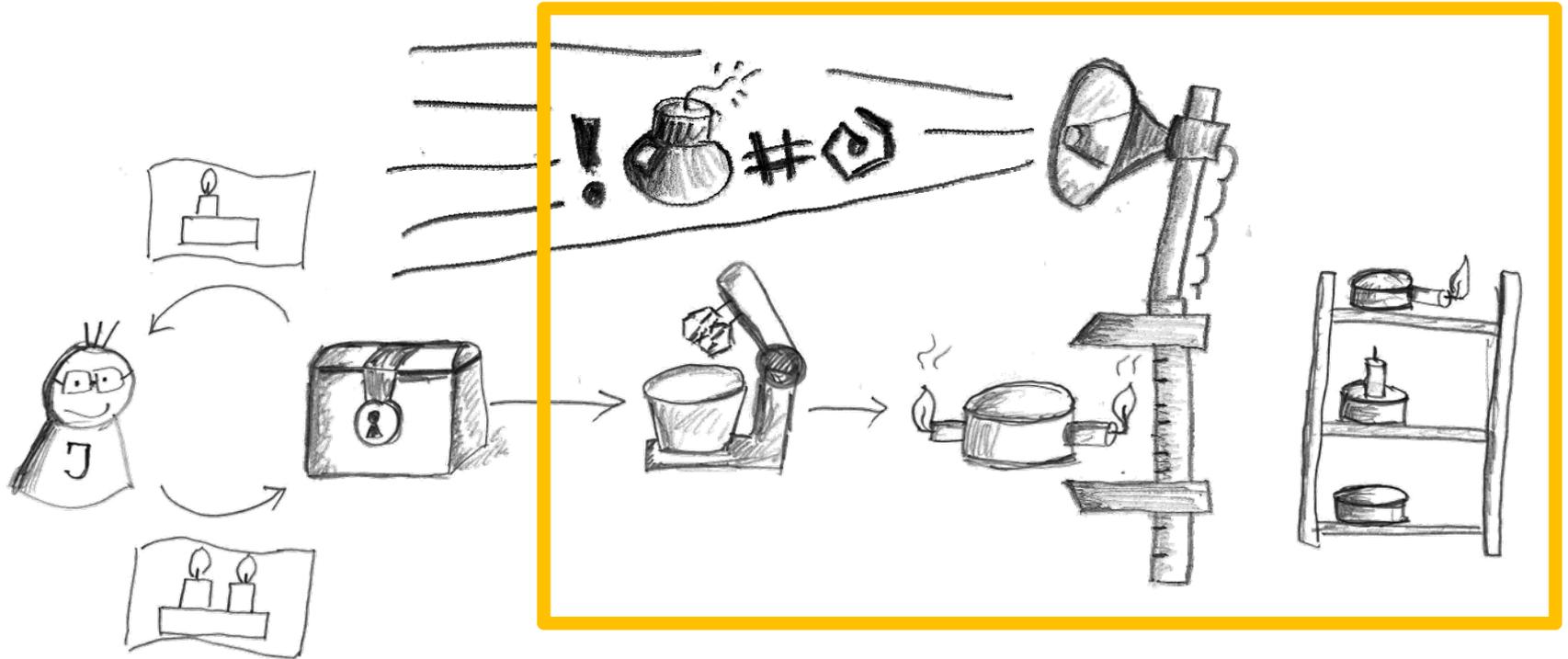


Was ist „Continuous Integration (CI)“?

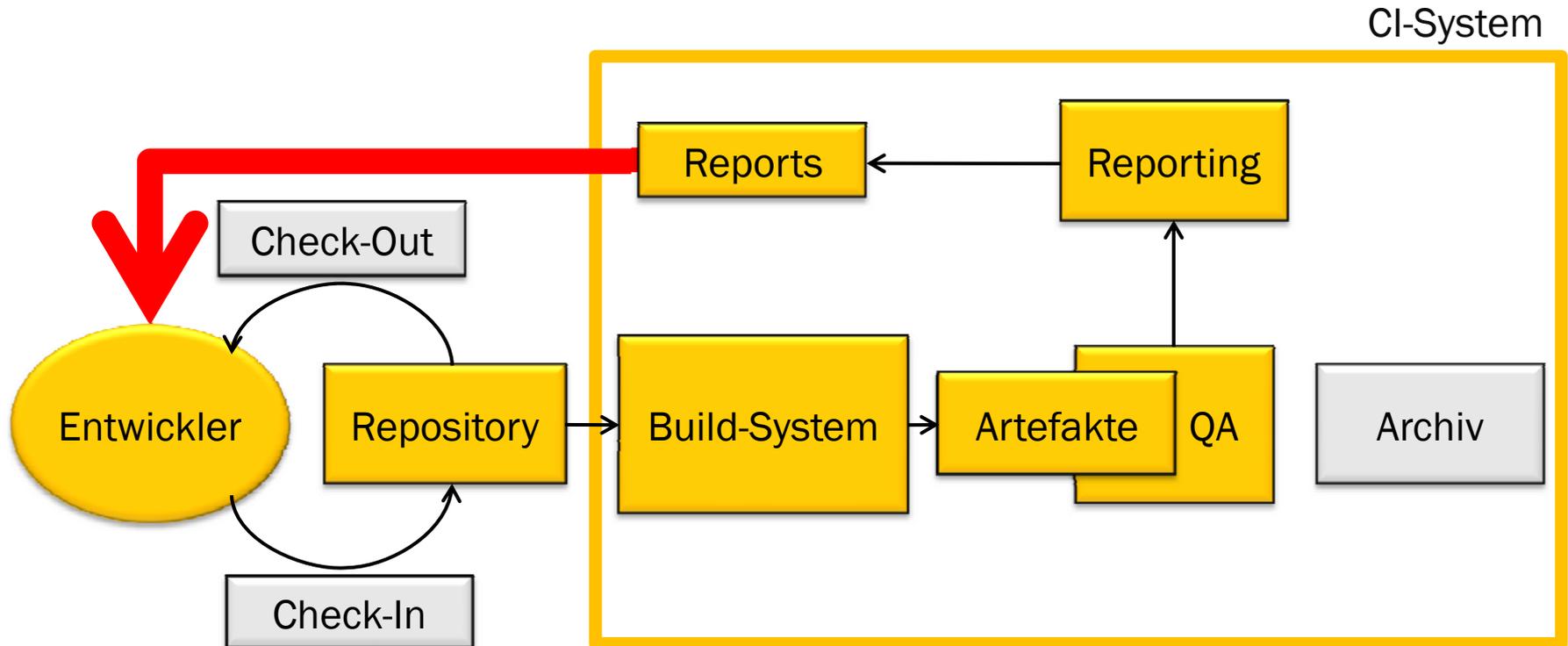


Was ist „Continuous Integration (CI)“?

orchestriert durch CI-System



Was ist „Continuous Integration (CI)“?



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

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

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

Hudson auf einen Blick

- Java-basierter CI-Server
- Initiiert 2006/2007 von Kohsuke Kawaguchi (ehemals Sun, jetzt InfraDNA.com)
- Open Source (MIT-Lizenz)
- Ca. 17.000 Installationen
- 1,5 Mio. LOC
- Zur Zeit 200+ Plugins
- 150+ Beitragende

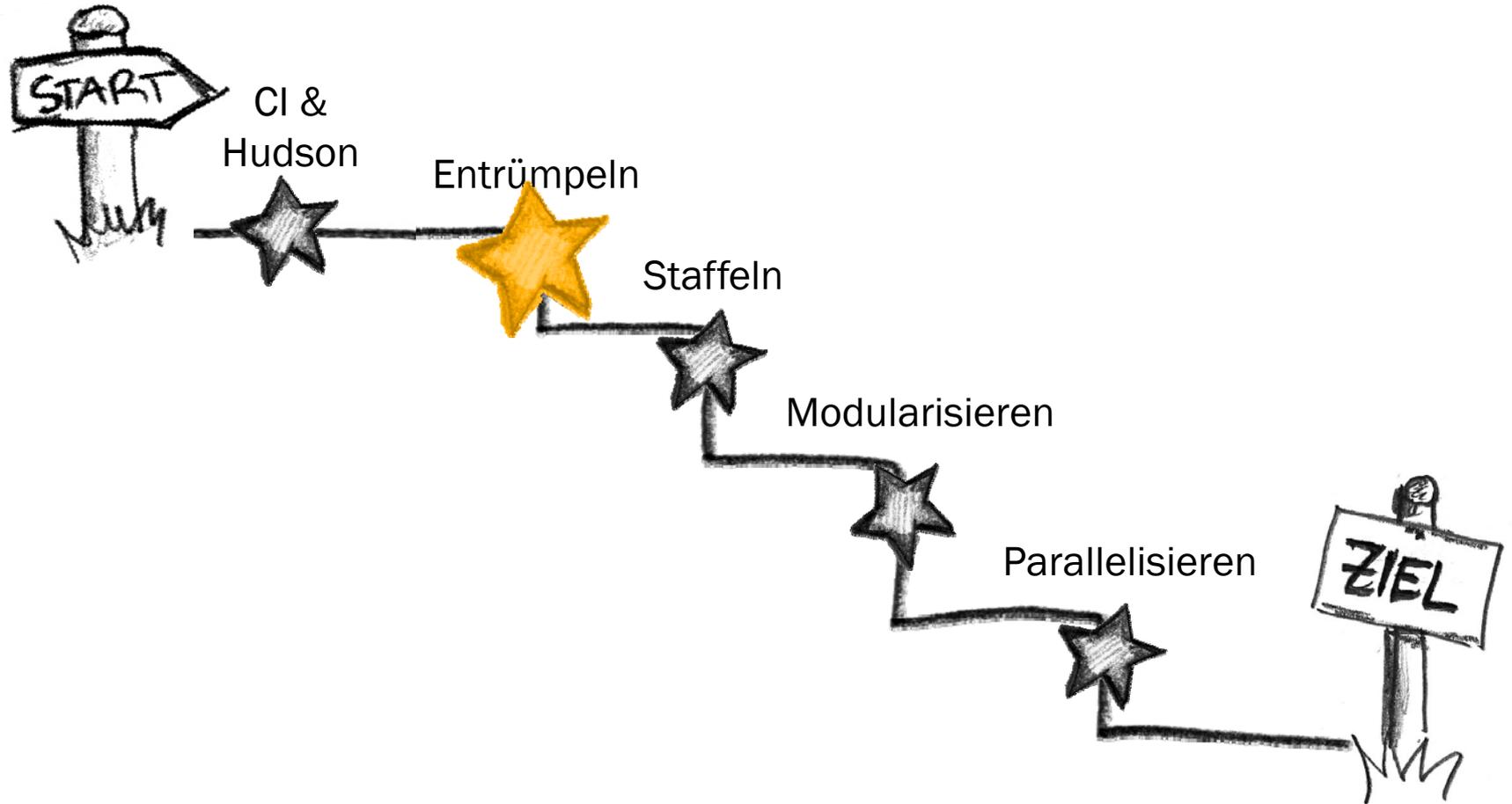


Wer verwendet Hudson (und darf es zugeben)?



Quelle: Kohsuke Kawaguchi, wiki.hudson-ci.org//x/EYQ5Ag

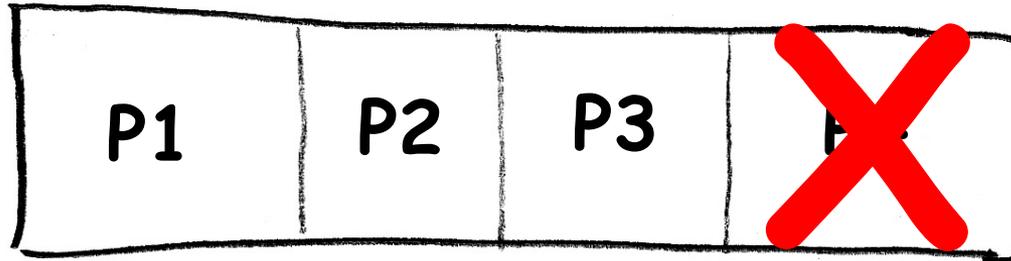
Agenda



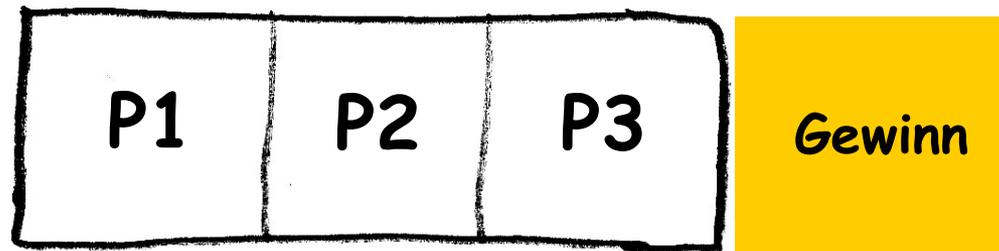
Entrümpeln: Was kann entfallen?



vorher



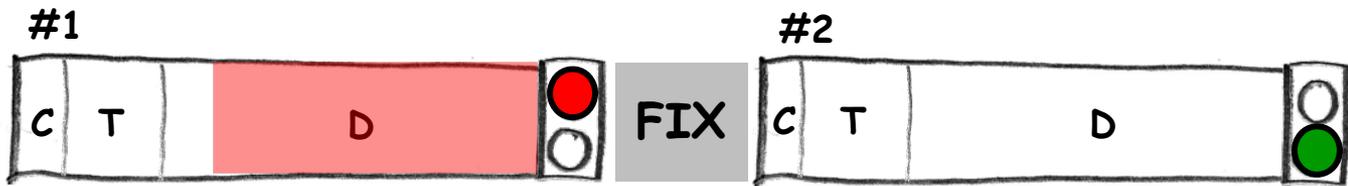
nachher



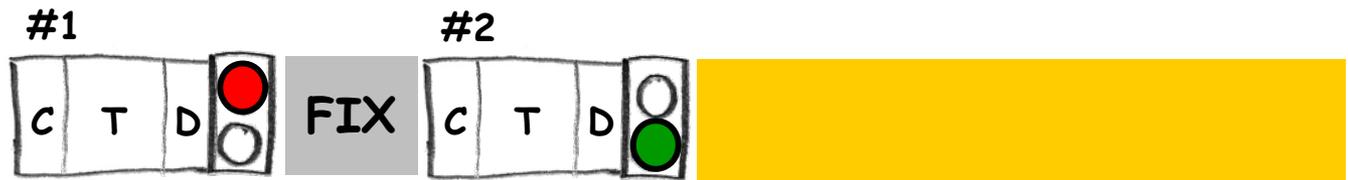
Entrümpeln: Beispiel



vorher

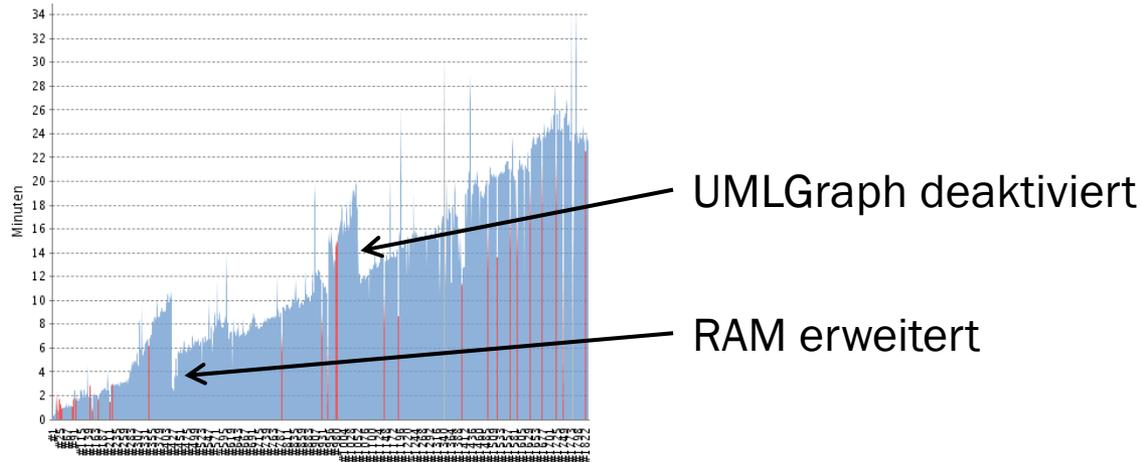


nachher



Entrümpeln mit Hudson

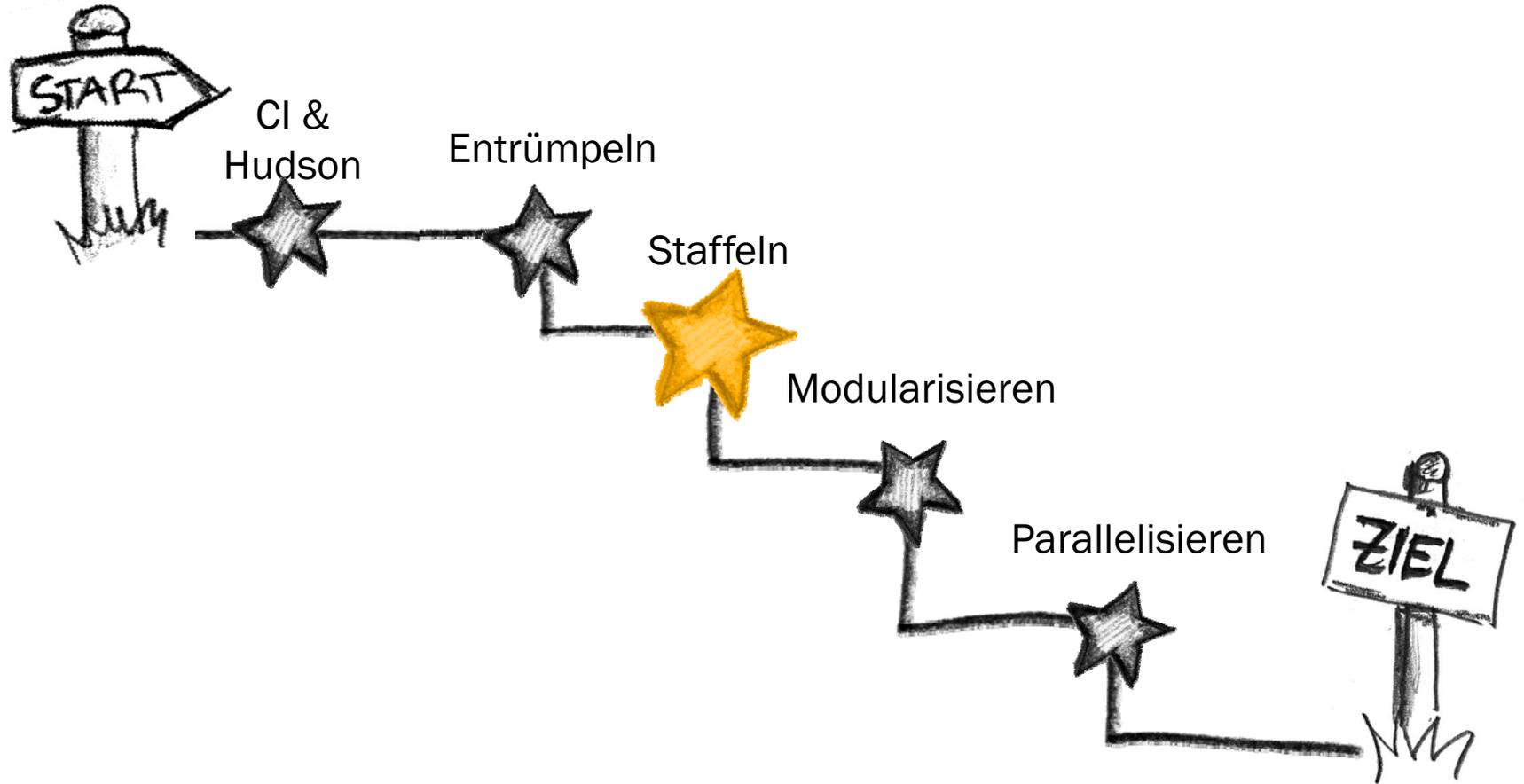
- Trends über Projektverlauf, z.B. Zunahme der Build-Dauer



- Schnelle Ad-Hoc-Analysen innerhalb der Web-Oberfläche



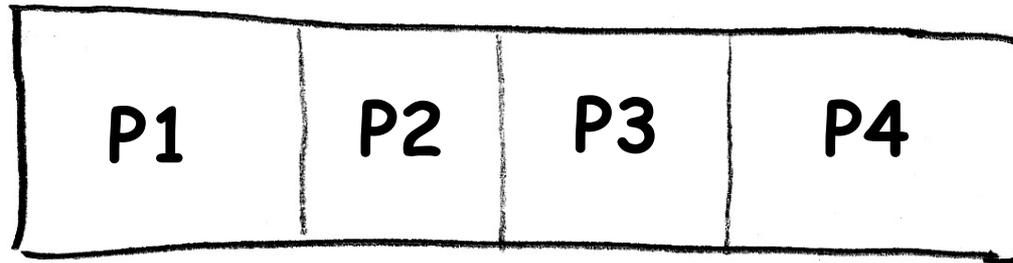
Agenda



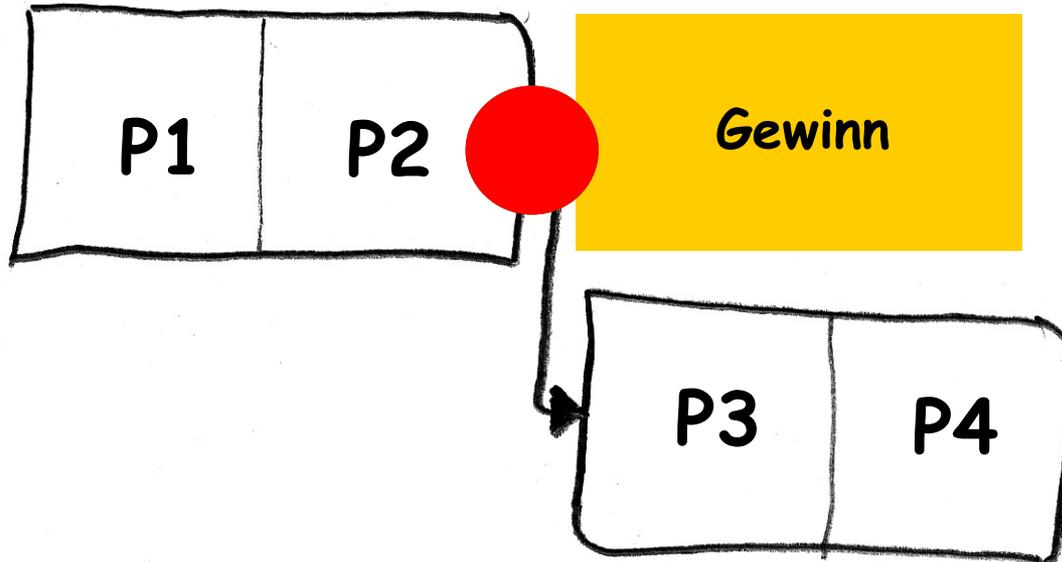
Staffeln: Das Wichtigste zuerst!



vorher



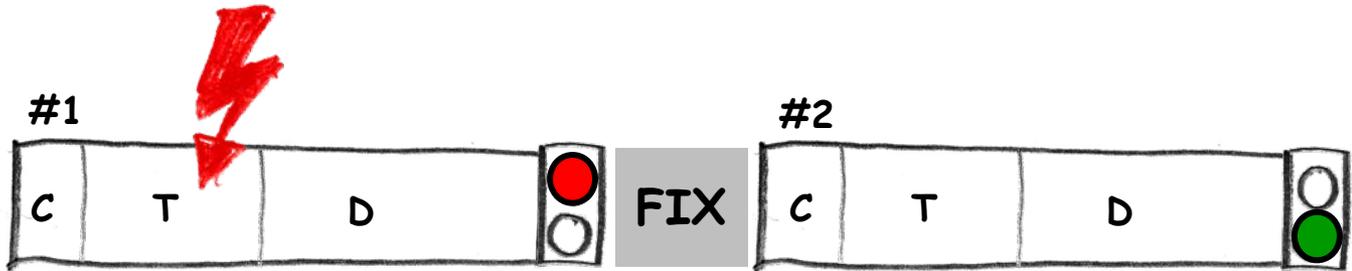
nachher



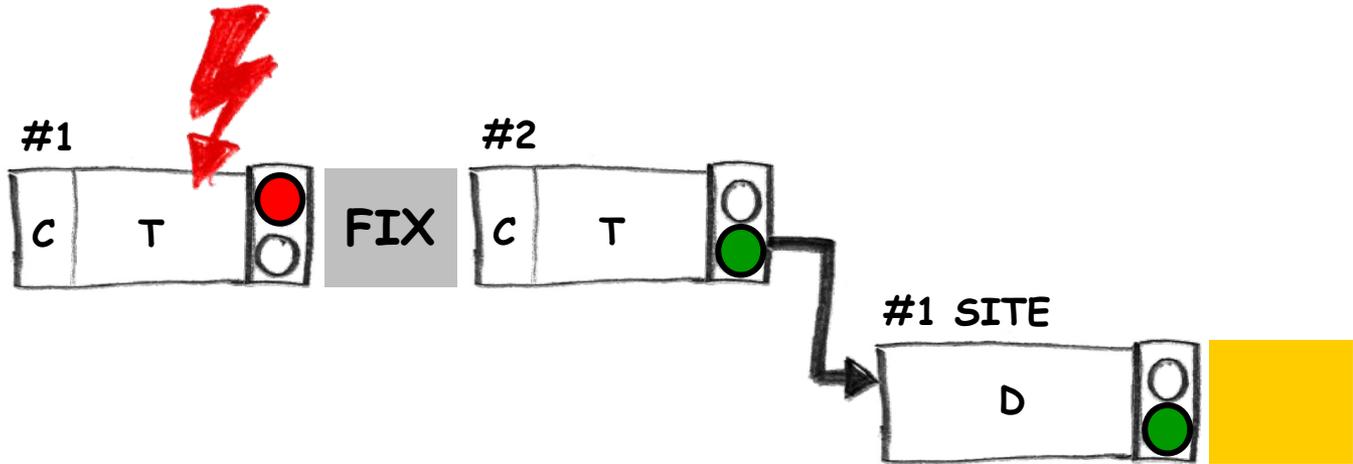
Staffeln: Beispiel



vorher

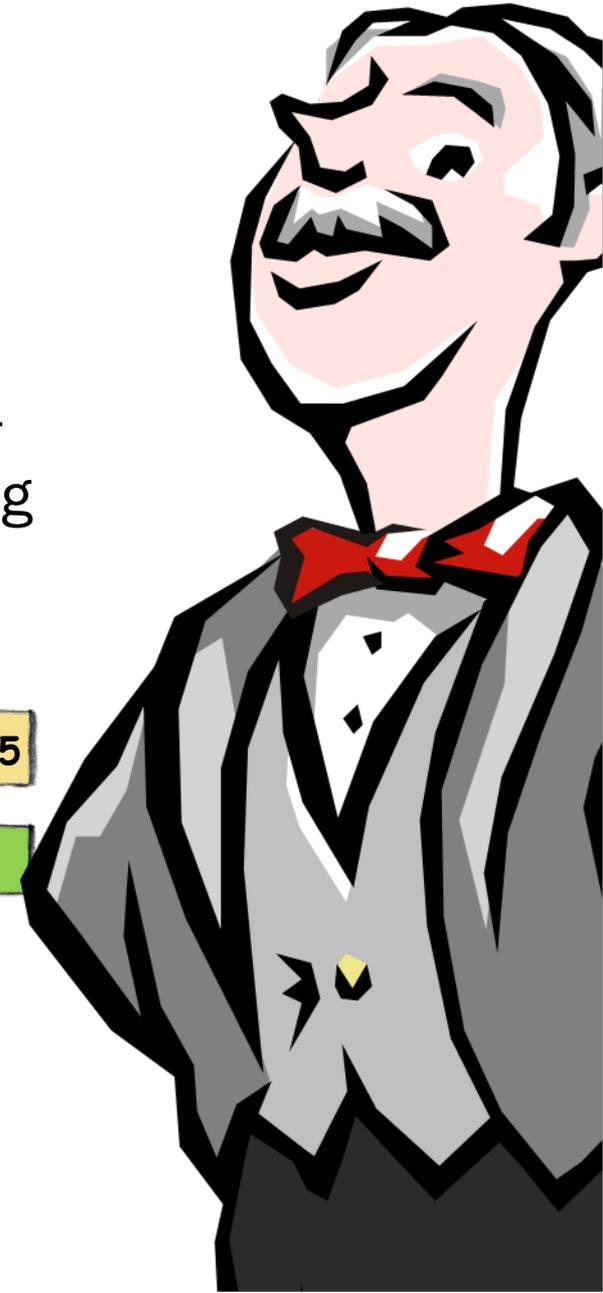


nachher

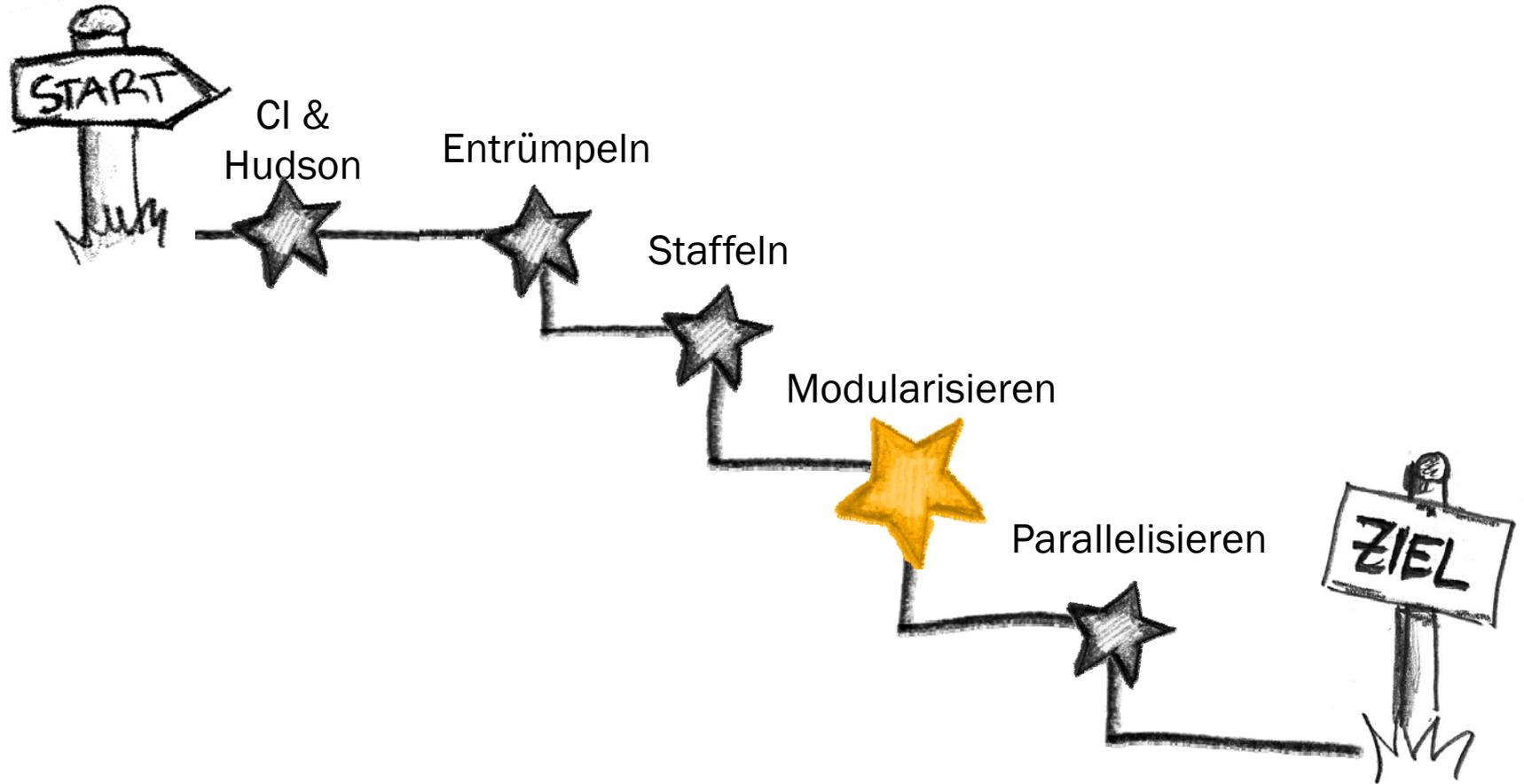


Staffeln mit Hudson

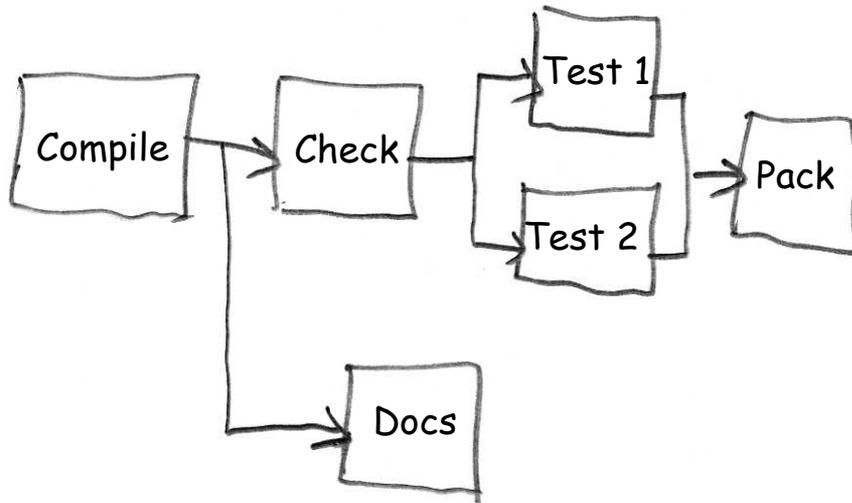
- Überwachung von Projektbeziehungen (vor-/nachgelagerte Projekte)
- Nachvollziehbarkeit des kompletten Build-Prozesses über verknüpfte Projekte hinweg
- Warteschlangen-Management



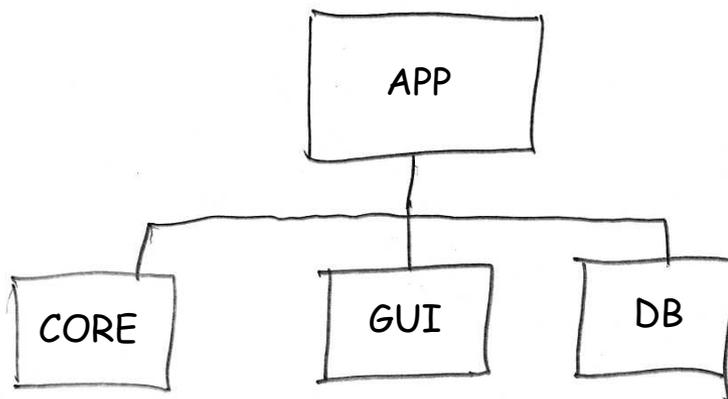
Agenda



Ab jetzt: Module statt Phasen



Abhängigkeiten
zwischen
(Build-)Phasen

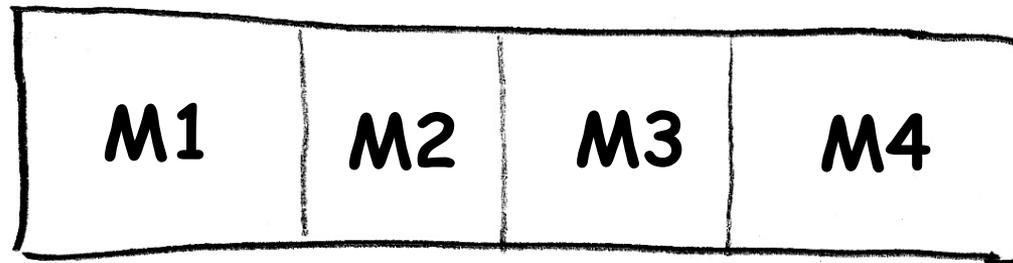


Abhängigkeiten
zwischen Modulen

Modularisieren: Was kann wiederverwendet werden?



#1



vorher



wiederverwenden

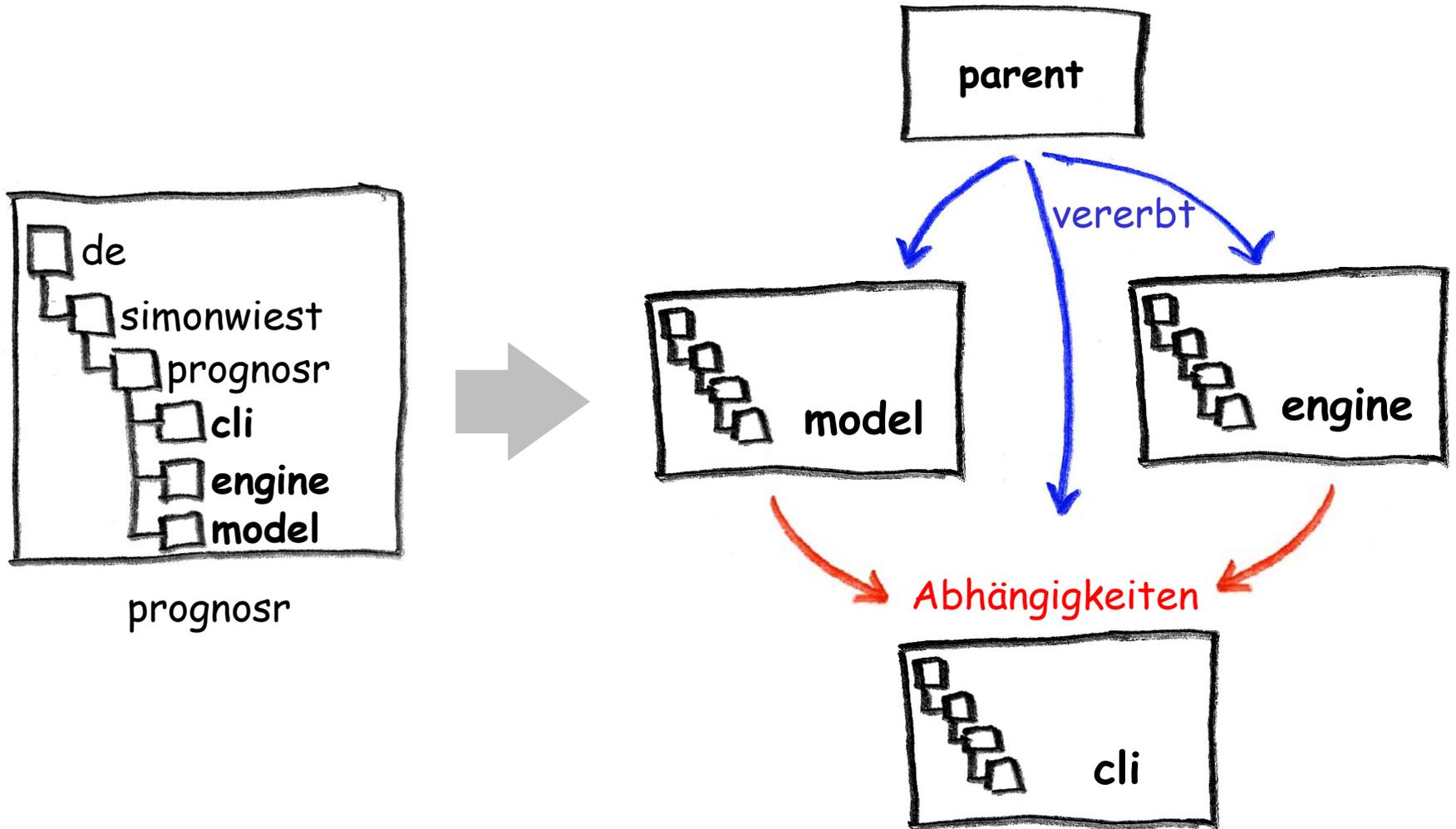
#2



nachher

neu bauen

Das Projekt-Layout wird verändert...



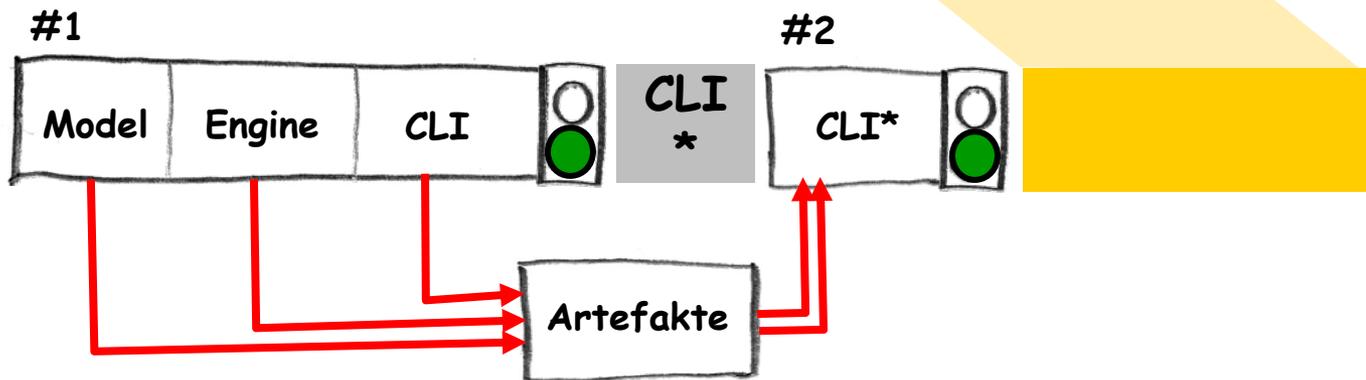
Modularisieren: Beispiel



vorher



nachher



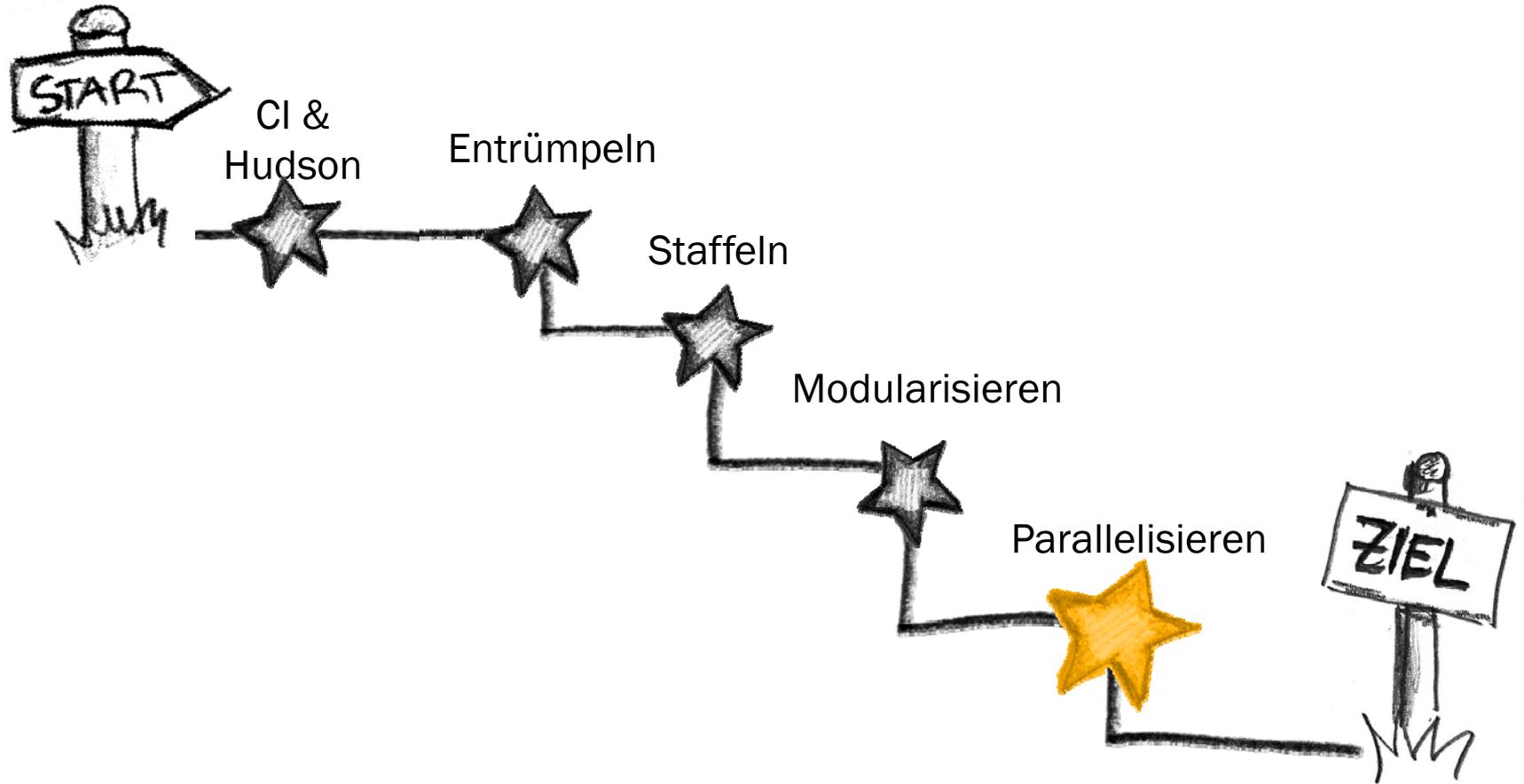
Voraussetzung: Artefakte müssen archiviert werden, z.B. in Maven Repository.

Modularisieren mit Hudson

- Direkte Unterstützung des Maven-Modulkonzeptes
- Aufbewahrung von Artefakten pro Build
- Funktionen zur Verwaltung zahlreicher Projekten (z.B. Views, Dashboard)
- Verwendungsnachweis der Build-Ergebnisse (Fingerprints)



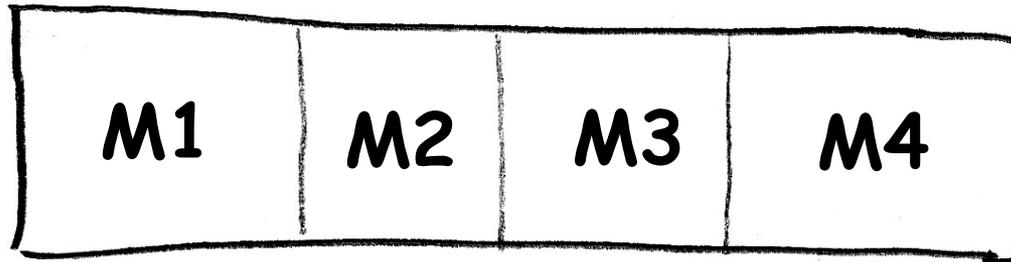
Agenda



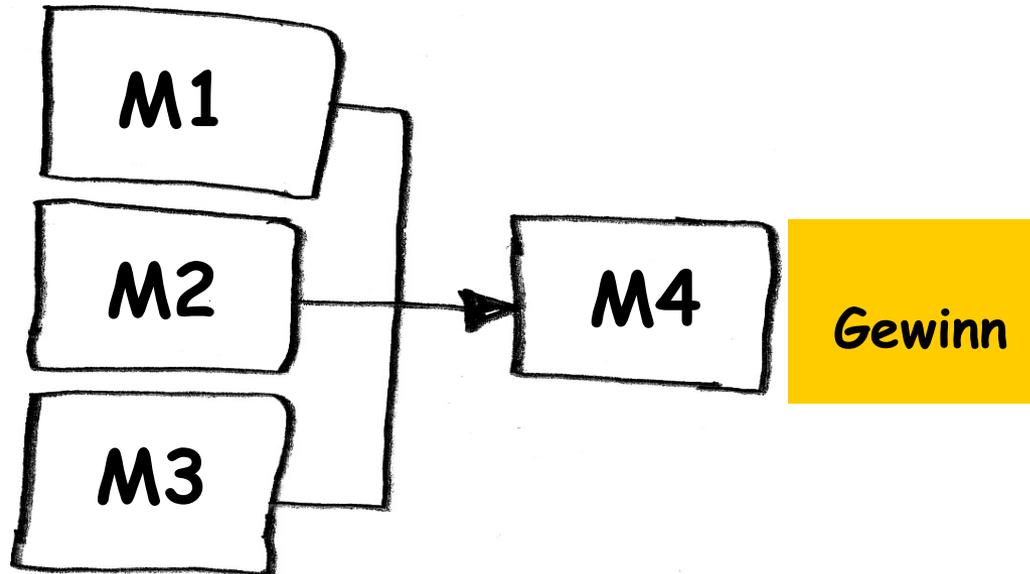
Parallelisieren: Gleichzeitig bauen



vorher



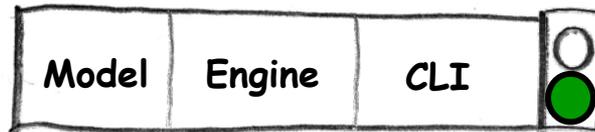
nachher



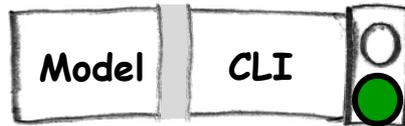
Parallelisieren: Beispiel



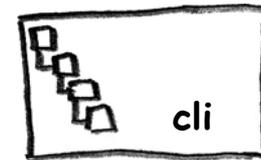
vorher



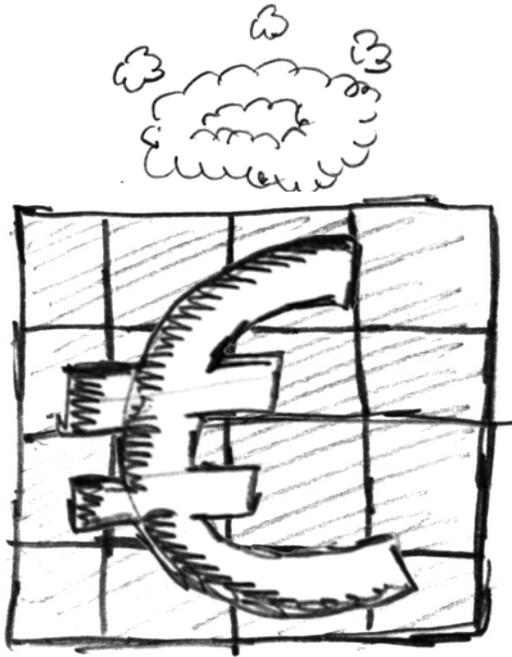
nachher



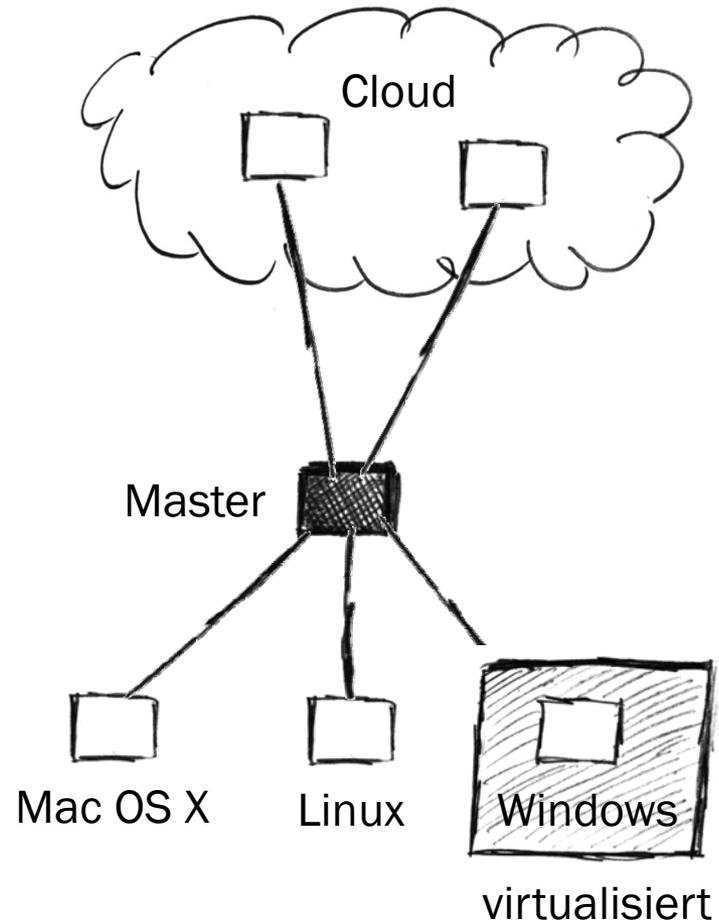
Abhängigkeiten



Verteilte Builds meist wirtschaftlicher



lokale Builds



verteilte Builds

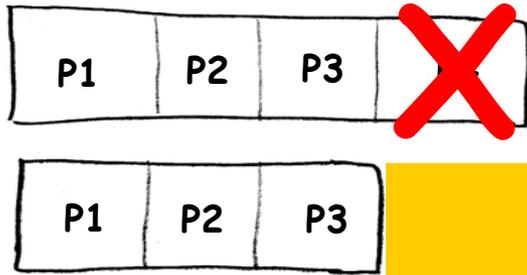
Parallelisieren mit Hudson

- Master-/Slave-Architektur
- Einfache Konfiguration von Knoten
- Automatischer Start/Stop von Slave-Knoten
- Zusammengefasste Darstellung, auch von verteilten Matrix-Builds

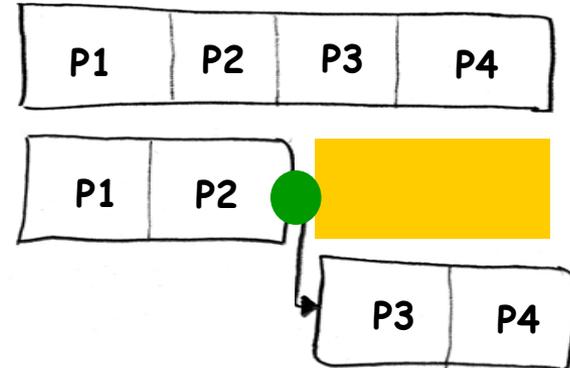
	jdk-1.5	jdk-1.6	jdk-1.7
oracle			
db2			
ms-sql			
mysql			



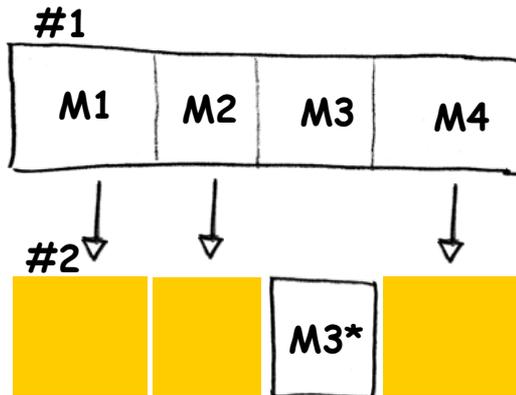
Zusammenfassung



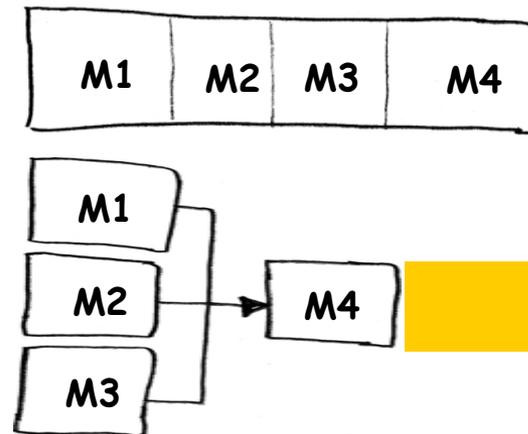
Entrümpeln



Staffeln



Modularisieren

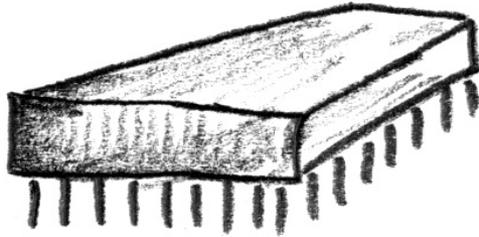


Parallelisieren

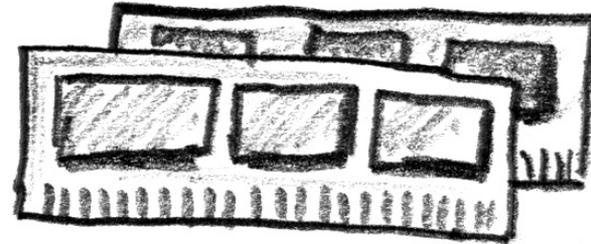
~~Unser Build dauert
zu lange für CI!~~

CI macht unsere
Builds wieder schnell!

Bonus-Folie: Hardware aufrüsten lohnt oft.



CPU



Hauptspeicher

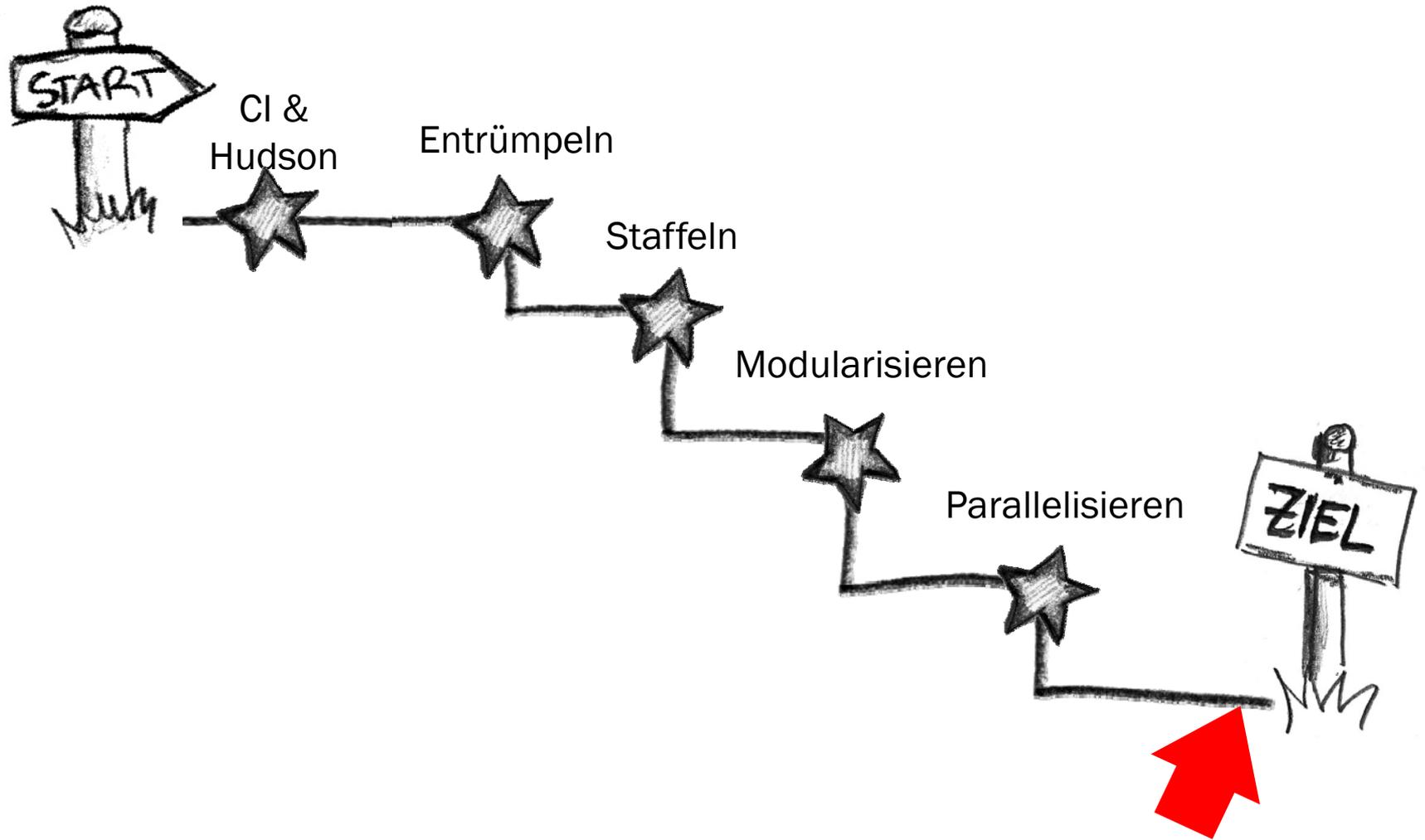


Festplatte



Netzwerk

Agenda



HERUNTERLADEN

EINSETZEN

WEITERSAGEN

MITMACHEN

hudson-ci.org

Und wenn Sie Lust auf mehr haben...



Vortrag

Kostenlose Vor-Ort
Einführungsvorträge
Sprechen Sie mich an!



Buch

„CI mit Hudson“
dpunkt.verlag
Herbst 2010



Beratung

Beratung und Coaching
rund um Continuous
Integration



Dr. Wiest – Ingenieurbüro für Softwaretechnik

www.simonwiest.de