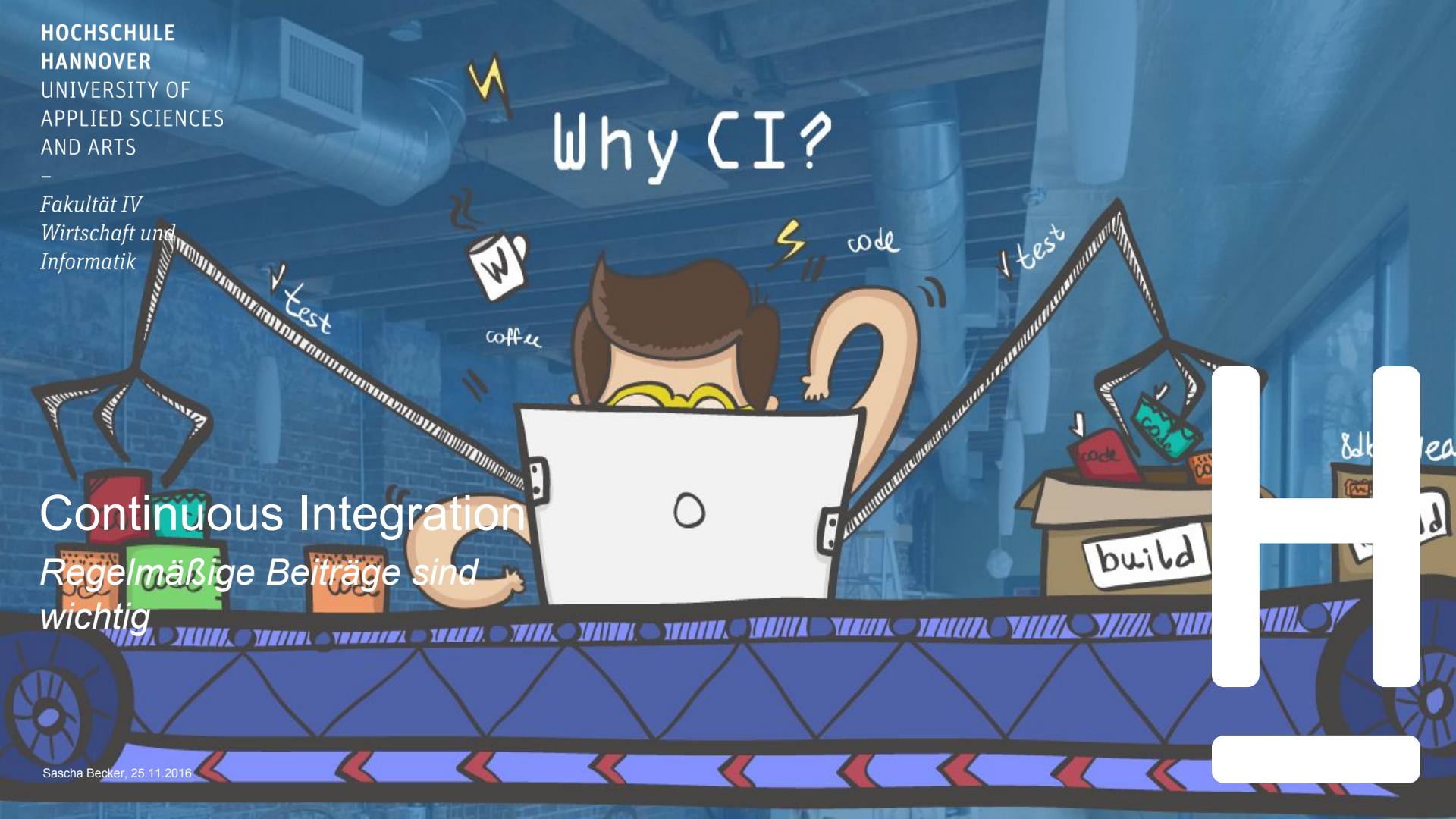
HOCHSCHULE
HANNOVER
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS
–
Fakultät IV
Wirtschaft und
Informatik



Continuous Integration

Regelmäßige Beiträge sind wichtig

Sascha Becker, 25.11.2016

Was ist CI?

Continuous Integration, kurz CI, ist ein Entwicklungsverfahren, welches Entwickler dazu drängt mehrmals am Tag bei einem gemeinsamen Repository beizutragen. Jeder Beitrag wird automatisch verifiziert. Dieser Vorgang erlaubt es dem Team Probleme früh zu erkennen.

Agenda

- Späte / Frühe Integration
- Kosten
- Projektstabilität
- Vorteile
- Anforderungen an System und Team
- Software

Späte / Frühe Integration

- Fehler später beheben ist teuer
  - Fehlersuche ist aufwendig
  - Es kann ein “Rattenschwanz” an Fehlern entstehen
- Frühe Integration = Fehler früh erkennen
  - Zeitaufwand gering da dahinter liegende Änderung gering
  - Leicht lokalisierbar (Fehler ist von heute)

“Continuous Integration doesn’t get rid of bugs, but it does make them dramatically easier to find and remove.” - Martin Fowler, Chief Scientist, ThoughtWorks

Kosten

Ständige Testabdeckung ist zeitaufwendig und teuer oder?

Kosten

Szenario: Fehler wird spät gefunden

- Fehlersuche dauert lange
- Ressource (Mensch) ist nicht verfügbar für neue Features
- Andere Warten ggfs. auf die Korrektur

DAS ist teuer!

Kosten

Szenario: Fehler wird schnell gefunden

- Fehlersuche geht schnell
- Behebung ist dank geringer Änderung einfach
- Ressource (Mensch) kann wieder an Features arbeiten
- Keiner wartet

DAS ist günstig!

Projektstabilität

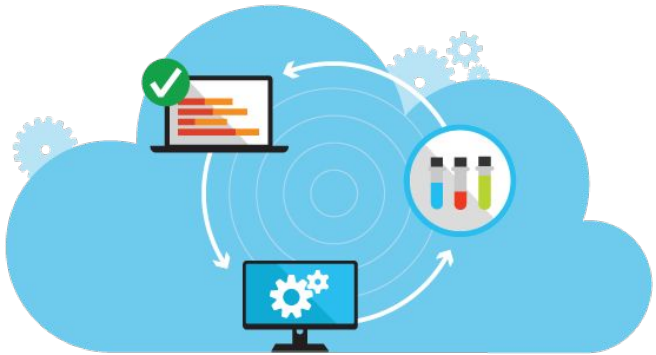
Was können spät erkannte Fehler noch anrichten?

Vorteile

- Die langen intensiven Integrationen entfallen
- Ein häufiger Beitrag erhöht die Sichtbarkeit der Codeänderungen für andere Mitarbeiter und fördert somit die Kommunikation
- Probleme werden schnell erfasst und behoben
- Es kann mehr Zeit in die eigentliche Entwicklung anstatt in die Fehlerbehebung investiert werden
- Aufbauende Änderungen basieren auf einem festen Fundament
- Direktes Feedback über Funktionsfähigkeit des neuen Codes
- Eine Reduzierung von Integrationsproblemen erlaubt eine häufigere Auslieferung der Software

Systemanforderungen

- Ein gemeinsames Repository
- Automatische Tests
- Automatische Builds
- Feedback über den Buildstatus
- Builds zur Verfügung stellen



5 goldene Regeln an das Team

- Check in frequently
- Do not check in broken code
- Do not check in untested code
- Do not check in when the build is broken
- Do not go home after checking in until the system builds

~ https://www.thoughtworks.com/de/continuous-integration

Software Scaffolding

YEOMAN

Software Git

GitHub  Bitbucket

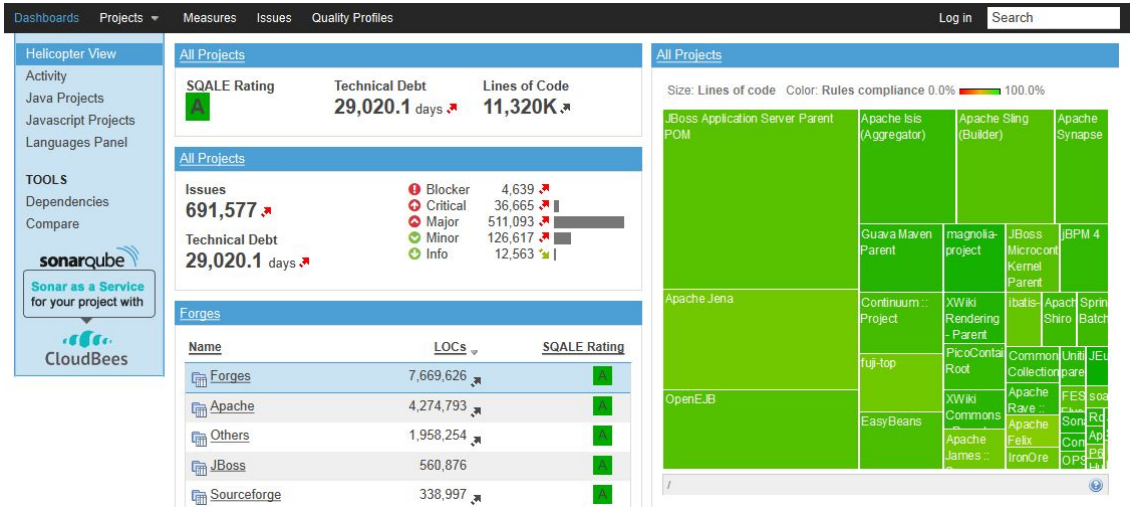
Software CI

Jenkins

Travis CI

Software - Code Coverage

SonarQube, Code Climate



Und weiter gehts...

Continuous Delivery