

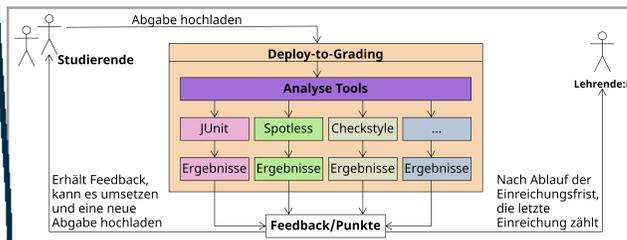
# Deploy to Grading

## Welche Ziele wurden verfolgt?

- Reduzierung des Workloads bei der Bewertung von Übungsaufgaben.
- Schnelleres Bereitstellen von Feedback für Studierende, damit das Feedback auch von Studierenden umgesetzt werden kann.
- Beibehalten von hoher Feedback-Qualität trotz automatisierten Feedback.

## Was wurde umgesetzt?

- CI/CD-Pipeline in GitHub, welche mithilfe von Spotbugs, Spotless, Checkstyle und JUnit Java-Übungsaufgaben kontrolliert und bewertet.



## Auto Grading, automatisiertes Feedback, CI/CD-Pipeline

## Stichworte zum Lehr-/Lernsetting

- Studierende bearbeiten wöchentliche Programmieraufgaben.
- Studierende stellen ihre Lösung Peers und Lehrenden vor.
- Lehrende bewerten diese und geben Feedback nach Abgabe.
- Erfahrung: Feedback wird in der Regel nicht umgesetzt, da die nächsten Übungsaufgaben schon bearbeitet werden.



## Materialien zur Nachnutzung

- Open Source auf GitHub
- OER Lehrmaterialien



## Welche Erfahrungen wurden gemacht?

- Vorhandene Tools können fachlich ausführliches Feedback geben (hier zunächst für Programmierer)

### Konflikt

- Die Anforderungen an die Erstellung der Übungsaufgaben steigen, da Vorgaben konkret genug sein müssen, um mit der Toolchain verarbeitet werden zu können.
- Aufgabenstellungen sollten andererseits nicht zu konkret sein - geben sonst den Lösungsweg vor



## Kontakt

Prof. Dr.-Ing Carsten Gips  
[carsten.gips@hsbi.de](mailto:carsten.gips@hsbi.de)

André Matutat  
[andre.matutat@hsbi.de](mailto:andre.matutat@hsbi.de)

