

# Continuous Integration mit GitHub Actions

Till Mahlburg

HTWK Leipzig

November 19, 2020

# Gliederung

## Motivation

Warum CI?

Warum GitHub?

## GitHub Actions - Begriffe

Einführung

Workflows

Events

Jobs

Steps

Actions

Runner

## GitHub Actions - Demo

## Abschluss

Zusammenfassung und Handout

Quellen und weiterführende Informationen

Schluss

# Motivation - Warum CI?

- ▶ deutsch: "fortlaufende Integration"
- ▶ regelmäßige Integration des aktuellen Entwicklungsstandes (z.B. täglich)
- ▶ Entwicklungsstand wird durch automatisierte Builds und Tests verifiziert
- ▶ häufigere, werkzeuggestützte Integration soll dabei helfen, Problemen zu minimieren
- ▶ Ziele: Minimierung von Bugs, bessere Vorhersagbarkeit der Projektdauer

# Motivation - Warum GitHub?

- ▶ sehr aktive Code-Hosting-Plattform - laut Eigenangabe mehr als 50 Millionen Nutzer
- ▶ gehört zu den performantesten Plattformen (laut [forgeperf.org](https://forgeperf.org))
- ▶ bietet u.A. Versionkontrolle, Issue Tracker, Teamverwaltung, Projektmanagement, Wikis und Automatisierung (GitHub Actions)

# Einführung

- ▶ GitHubs Automatisierungslösung
- ▶ ermöglicht Continuous Integration und Deployment
- ▶ führt zu bestimmten *Events* vom Nutzer festgelegte *Workflows* aus
- ▶ unterstützt verschiedene Plattformen und ist von der genutzten Programmiersprache unabhängig

# Workflows

- ▶ automatisierte Ausführungsanleitung, bestehend aus mehreren *Jobs* und durch *Events* ausgelöst.
- ▶ werden im YAML-Format im Verzeichnis `.github/workflows/` direkt im Quelltextrepository gespeichert

# Events

- ▶ Ereignis, das einen *Workflow* anstoßen kann
- ▶ interne Events: Ereignisse, die auf GitHub passieren
- ▶ Beispiele:
  - ▶ `schedule`
  - ▶ `push`
  - ▶ `pull_request`
  - ▶ `milestone`
- ▶ externe Events: Ereignisse außerhalb von GitHub
- ▶ mithilfe von Webhooks

# Jobs

- ▶ Ausführungsprozeduren, die von einem *Workflow* ausgeführt werden
- ▶ werden i.d.R. parallel ausgeführt
- ▶ alle Anweisungen eines Jobs werden auf dem selben *Runner* ausgeführt



# Steps

- ▶ Element eines *Jobs*
- ▶ hilft zusammengehörige Anweisungen logisch zu gruppieren
- ▶ werden auch auf dem selben *Runner* ausgeführt und können auf dieselben Daten zugreifen

# Actions

- ▶ einzelne Anweisungen, die dann in *Steps* gruppiert werden
- ▶ portabel → können wieder verwendet werden
- ▶ können auch anderen zur Verfügung gestellt werden oder es können die *Actions* anderer Nutzer importiert werden

# Runner

- ▶ Server, auf dem die GitHub runner Software installiert ist
- ▶ führt die *Jobs* eines angestoßenen *Workflows* aus
- ▶ Selfhosting von *Runnern* ist möglich
- ▶ GitHub stellt aber auch kostenfrei *Runner* zur Verfügung

# Demo

# Zusammenfassung und Handout

- ▶ GitHub Actions bietet umfangreiche Möglichkeiten zur Automatisierung von typischen Aufgaben der Continuous Integration
- ▶ Actions ermöglichen die Wiederverwendung bestimmter Teile eines Workflows und erleichtern die Einrichtung
- ▶ viele Möglichkeiten, um GitHub Actions an den eigenen Workflow anzupassen
- ▶ Handout:
  - ▶ wird als Cheatsheet verwendbar sein
  - ▶ bis spätestens 25.11. im OPAL im Studierendenaustausch-Ordner und im Demorepository

# Quellen und weiterführende Informationen

- ▶ <https://www.martinfowler.com/articles/continuousIntegration.html>
- ▶ <https://github.com/about>
- ▶ <https://github.com/features>
- ▶ <https://github.com/features/actions>
- ▶ <https://docs.github.com/en/free-pro-team@latest/actions>
- ▶ <https://docs.github.com/en/free-pro-team@latest/actions/learn-github-actions/introduction-to-github-actions>
- ▶ <https://docs.github.com/en/free-pro-team@latest/actions/reference/events-that-trigger-workflows>
- ▶ <https://docs.github.com/en/free-pro-team@latest/rest/reference/repos>
- ▶ <https://forgeperf.org>

Vielen Dank für die Aufmerksamkeit

Gibt es Fragen?