Technology
Arts Sciences
TH Köln

Lehrveranstaltung
SM - Software-Management

Version: 2 | Letzte Änderung: 30.07.2019 23:44 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

Allgemeine Informationen

Langname	Software-Management
Anerkennende LModule	SM_BaTIN
Verantwortlich	Prof. Dr. René Wörzberger Professor Fakultät IME
Organisation und Unterlagen	Ilu-Kurs
Niveau	Bachelor
Semester im Jahr	Wintersemester
Dauer	Semester
Stunden im Selbststudium	78
ECTS	5
Dozenten	Prof. Dr. René Wörzberger Professor Fakultät IME
Voraussetzungen	(1) fortschrittene Kenntnisse in der Programmierung in Java (2) Erfahrungen mit der Entwicklung im Team (3) Kenntnisse in Software-Engineering
Unterrichtssprache	deutsch, englisch bei Bedarf
separate Abschlussprüfung	Ja

Abschlussprüfung

Details

Diese abschließende, summarische Prüfung ist entweder eine mündliche Prüfung oder einer Klausurarbeit. Die Studierenden müssen in dieser abschließenden Prüfung und geleitet durch Teilaufgaben bzw. Fragen zeigen, wie sie ein einfaches System im Team entwickeln, den Build automatisieren, Tests gestalten und Cloud-Infrastrukturen aufbauen.

Mindeststandard

Ein Basiswissen in den im Prüfungkonzept angesprochenen Bereichen muss nachgewiesen werden. In Klausurarbeiten reichen dabei rechnerisch in der Regel 50% der erreichbaren erreichbaren Prüfungspunkte zum Bestehen.

Prüfungstyp

Diese abschließende, summarische Prüfung ist entweder eine mündliche Prüfung oder einer Klausurarbeit. Die Studierenden müssen in dieser abschließenden Prüfung und geleitet durch Teilaufgaben bzw. Fragen zeigen, wie sie ein einfaches System im Team entwickeln, den Build automatisieren, Tests gestalten und Cloud-Infrastrukturen aufbauen.

Vorlesung / Übungen

Lernziele

Kenntnisse

Interne Funktionsweise des Versionsverwaltungssytems Git

Team-Organiation mit Funktionalitäten von GitLab

Build-Automatisierung mit Apache Maven

Continuous-Integration and -Delivery (CICD) mit GitLab-Runner

Automatisierung von Tests mit JUnit

Erstellung von Mocks mit Mockito

Automatisierung von WebUI-Tests mit Selenium

Automatisierung von Lasttests mit Apache JMeter

Vermessung von Code-Qualität mit Sonarqube

Klassische und Cloud-Infrastrukturen

Container-Virtualisierung mit Docker

Container-Orchestrierung mit Kubernetes

Fertigkeiten

Erstellung eines System-Clusters in der Google Cloud

Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Vorlesung	2
Übungen (ganzer Kurs)	1
Übungen (geteilter Kurs)	0
Tutorium (freiwillig)	0

Separate Prüfung

keine

Praktikum

Lernziele

Fertigkeiten

Entwickeln im Team mit GitLab

Einpflegen und Weiterentwickeln der Code-Basis in/mit Git

Erstellung von Build-Scripts mit Maven

Implementieren von Tests mit JUnit, Mockito, Selenium und JMeter

Containerisierung und Deployment mit Docker und Kubernetes

Aufbau eines System-Clusters in der Google Cloud inklusive (kontinuierlichem) Deployment von Releases in diese.

Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Praktikum	1
Tutorium (freiwillig)	0

Separate Prüfung

Prüfungstyp

Projektaufgabe im Team bearbeiten (z.B. im Praktikum)

Details

Die Lösungen der Hausaufgabe und der Anwesenheitsaufgabe wird im Praktikum von den jeweiligen Teams demonstriert und mit den Betreuern besprochen. Bei ausreichender Lösungsqualität wird für den Praktikumstermin ein Testat erteilt. Jedes Team hat im Semester 3 bis 4 Praktikumstermine.

Mindeststandard

Lösungen müssen im Sinne der jeweiligen Aufgabenstellung funktionsfähig sein.