

Lehrveranstaltungshandbuch 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

Gültig ab Wintersemester
2022/23

Organisation und Unterlagen [Ilias-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

Literatur

wird in Vorlesung bekannt gegeben

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.

separate Ja
Abschlussprüfung

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

Prüfungstyp Klausur

– Vorlesung / Übungen

Lernziele

Zieltyp	Beschreibung
Kenntnisse	Interne Funktionsweise des Versionsverwaltungssystems Git
Kenntnisse	Team-Organisation mit Funktionalitäten von GitLab
Kenntnisse	Build-Automatisierung mit Apache Maven
Kenntnisse	Continuous-Integration and -Delivery (CI/CD) mit GitLab-Runner
Kenntnisse	Automatisierung von Tests mit JUnit
Kenntnisse	Erstellung von Mocks mit Mockito
Kenntnisse	Automatisierung von WebUI-Tests mit Selenium
Kenntnisse	Automatisierung von Lasttests mit Apache JMeter
Kenntnisse	Vermessung von Code-Qualität mit Sonarqube
Kenntnisse	Klassische und Cloud-Infrastrukturen
Fertigkeiten	Erstellung eines System-Clusters in der Google Cloud
Kenntnisse	Container-Virtualisierung mit Docker
Kenntnisse	Container-Orchestrierung mit Kubernetes

Besondere Voraussetzungen

keine

Begleitmaterial (1) Vorlesungsfolien
(2) ggf. Vorlesungsskript
(3) Übungsblätter

Separate Prüfung Nein

Aufwand Präsenzlehre

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

Tutorium (freiwillig)

0

– Praktikum

Lernziele

Zieltyp	Beschreibung
Fertigkeiten	Entwickeln im Team mit GitLab
Fertigkeiten	Einpflegen und Weiterentwickeln der Code-Basis in/mit Git
Fertigkeiten	Erstellung von Build-Scripts mit Maven
Fertigkeiten	Implementieren von Tests mit JUnit, Mockito, Selenium und JMeter
Fertigkeiten	Containerisierung und Deployment mit Docker und Kubernetes
Fertigkeiten	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

Besondere Voraussetzungen

keine

Begleitmaterial Aufgabenstellung zur Vorbereitungsaufgabe (Hausaufgabe), Aufgabenstellung zur Präsenzaufgabe

Separate Prüfung Ja

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.