

Modul

SM - Software Management

Bachelor Technische Informatik 2020

Version: 5 | Letzte Änderung: 30.07.2019 23:24 | Entwurf: 0 | Status: vom Modulverantwortlichen freigegeben | Verantwortlich: Wörzberger

^ Allgemeine Informationen

Anerkannte Lehrveranstaltungen	SM_Wörzberger
Modul ist Bestandteil der Studienschwerpunkte	NVS - Netze und Verteilte Systeme SOS - Software-Systeme
Dauer	1 Semester
ECTS	5
Zeugnistext (de)	Software-Management
Zeugnistext (en)	Software Management
Unterrichtssprache	deutsch
abschließende Modulprüfung	Ja

Modulprüfung

Benotet	Ja
Frequenz	Jedes Semester

Prüfungskonzept

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.

^ Allgemeine Informationen

Inhaltliche Voraussetzungen

PI1 - Praktische Informatik 1	Ein fortgeschrittenes Verständnis des "gemanagten" Subjekts im Java-Source-Code ist erforderlich.
PI2 - Praktische Informatik 2	Ein fortgeschrittenes Verständnis des "gemanagten" Subjekts im Java-Source-Code ist erforderlich.
SP - undefined	Ein fortgeschrittenes Verständnis des "gemanagten" Subjekts im Java-Source-Code ist erforderlich. Zudem sind für die Bearbeitung von Praktikumsaufgaben fortgeschrittene Kenntnisse mit der Arbeit im Team erforderlich.

Kompetenzen

Kompetenz	Ausprägung
In Systemen denken	diese Kompetenz wird vermittelt
fachliche Probleme abstrahieren und formalisieren	diese Kompetenz wird vermittelt
Konzepte und Methoden der Informatik, Mathematik und Technik kennen und anwenden	diese Kompetenz wird vermittelt
Systeme analysieren	diese Kompetenz wird vermittelt
Systeme entwerfen	diese Kompetenz wird vermittelt
Systeme realisieren	diese Kompetenz wird vermittelt
Systeme prüfen	diese Kompetenz wird vermittelt

Informationen beschaffen und auswerten; Technische Zusammenhänge darstellen und erläutern	diese Kompetenz wird vermittelt
Typische Werkzeuge, Standards und Best Practices der industriellen Praxis kennen und einsetzen	diese Kompetenz wird vermittelt
In vorhandene Systeme einarbeiten und vorhandene Komponenten sinnvoll nutzen	diese Kompetenz wird vermittelt
Komplexe technische Aufgaben im Team bearbeiten	diese Kompetenz wird vermittelt
Befähigung zum lebenslangen Lernen	diese Kompetenz wird vermittelt

^ Vorlesung / Übungen

Exemplarische inhaltliche Operationalisierung

Vorlesungen und Übungen zur (1) Entwicklung im Team mit Git und GitLab; (2) Build-Automatisierung mit Maven; (3) Testautomatisierung mit JUnit, Mockito, Selenium und JMeter, (4) Aufbau von Infrastrukturen in der Google Cloud, (5) Virtualisierungstechniken insb. Container-Virtualisierung mit Docker und -Orchestrierung mit Kubernetes

Separate Prüfung

keine

^ Praktikum

Exemplarische inhaltliche Operationalisierung

Der Source-Code eines Systems wird vorgegeben. Studierende müssen diesen Source-Code in kleinen Teams in aufeinander aufbauenden Praktikumsaufgaben (1) unter Git-Versionsverwaltung stellen, (2) den Build automatisieren, (3) automatisch ausgeführte Regressions- und Lasttests implementieren, (4) das Deployment über Container realisieren und (5) das System in einer aufzusetzenden Cloud-Infrastruktur installieren.

Separate Prüfung

Benotet	Nein
Frequenz	Einmal im Jahr

Prüfungskonzept

Der Erfolg der Teilnahme am Modulelement "Praktikum" wird durch begleitende Testate überprüft und belegt (3 bis 4 Testate pro Teilnehmer). Das Bestehen dieser unbenoteten Leistungsprüfung (ULP) ist Voraussetzung für die Teilnahme an der abschließenden Modul-Prüfung.