

## Grundlagen Software Engineering (T3M70304)

### Foundations of Software Engineering

#### FORMALE ANGABEN ZUM MODUL

MODULNUMMER	VERORTUNG IM STUDIENVERLAUF	MODULDAUER (SEMESTER)	MODULVERANTWORTUNG	SPRACHE
T3M70304	-	1	Prof. Dr. Rainer Hoch	Deutsch/Englisch

#### EINGESETZTE LEHRFORMEN

LEHRFORMEN	LEHRMETHODEN
Vorlesung, Übung	Lehrvortrag, Diskussion

#### EINGESETZTE PRÜFUNGSFORMEN

PRÜFUNGSLEISTUNG	PRÜFUNGSUMFANG (IN MINUTEN)	BENOTUNG
Klausur	120	ja

#### WORKLOAD UND ECTS-LEISTUNGSPUNKTE

WORKLOAD INSGESAMT (IN H)	DAVON PRÄSENZZEIT (IN H)	DAVON SELBSTSTUDIUM (IN H)	ECTS-LEISTUNGSPUNKTE
150	50	100	5

#### QUALIFIKATIONSZIELE UND KOMPETENZEN

##### FACHKOMPETENZ

Die Studierenden können nach Abschluss des Moduls

- Ansätze zum Software-Engineering verstehen und umsetzen
- Vorgehensmodelle verstehen und anwenden
- Methoden des Software-Engineering nutzen sowie diese auf Problemstellungen anwenden

Vertiefung der Grundlagen durch praktische Übungen in Kleingruppen

##### METHODENKOMPETENZ

Die Studierenden haben die Kompetenz erworben:

- Einfache Problemstellungen analysieren und rechnergestützte Lösungen umsetzen und spezifizieren zu können.
- Ergebnisse der jeweiligen Phasen in ihren Inhalten und Zielrichtungen erfassen und dokumentieren zu können.
- Konkrete Ergebnisse innerhalb der einzelnen Projektphasen mit geeigneten Tools erarbeiten zu können.

##### PERSONALE UND SOZIALE KOMPETENZ

Die Kompetenz problemlösend und kooperativ zu arbeiten wird gefördert. Die Studierenden setzen sich im Team mit ihrer Vorgehensweise bei der Bearbeitung von Lösungsoptionen zur Softwareentwicklung reflexiv auseinander.

##### ÜBERGREIFENDE HANDLUNGSKOMPETENZ

Die Studierenden haben grundlegende Kenntnisse in den angegebenen Fachgebieten dieses Moduls und können sich an fachlichen Gesprächen und Diskussionen in der Softwareentwicklung beteiligen sowie eigene Beiträge in deutscher und englischer Sprache beisteuern.

#### LERNEINHEITEN UND INHALTE

LEHR- UND LERNEINHEITEN	PRÄSENZZEIT	SELBSTSTUDIUM
Grundlagen des Software Engineering	50	100

## LERNEINHEITEN UND INHALTE

### LEHR- UND LERNEINHEITEN

### PRÄSENZZEIT

### SELBSTSTUDIUM

Grundlagen des Software Engineering

- Definition, Motivation und Ziele des Software-Engineering
- Software-Entwicklungsprozess, Vorgehensmodelle und Phasen
- Anforderungsmanagement (incl. Pflichtenheft, Lastenheft)
- Methoden der Modellierung
- Software-Projektmanagement, Entwicklung im Team
- Produktstandards, Dokumentation, CASE-Methoden und Werkzeuge der Softwareentwicklung
- SW-Qualitätssicherung (Codequalität, statische & dynamische Code-Analyse, Testen)
- Automatisierung (Build-Prozess, Dokumentation, ...)
- Versionsverwaltung, Fehler-Tracking
- Vertiefende Methoden der Software-Analyse
- Software-Fehler: Fehlerquellen, Fehleranalyse, Fehlerbehebung
- Debugging: Verfahren, Tipps & Tricks
- Profiling: Werkzeuge, Verfahren

Praktische Umsetzung des Erlernten in Übungen + Gruppendiskussion, ggf. auch einer kleinen Projektarbeit in Kleingruppen mit vorgegebener Aufgabenstellung. Phasenspezifische werden die verschiedenen Arten der Dokumentation und Werkzeugunterstützung behandelt.

### BESONDERHEITEN

Praktische Aspekte der Softwareentwicklung (= Programmierung) werden im Modul "Einführung in die Programmierung" betrachtet; die Module passen inhaltlich zusammen.

### VORAUSSETZUNGEN

Keine besonderen Voraussetzungen (Grundlagen, die in einem technikorientierten Bachelor-Studium im erworben werden)  
Vorteilhaft: Kenntnisse in einer Programmiersprache, Objektorientierung, Projektmanagement

### LITERATUR

- Abts, D.: "Grundkurs JAVA- Von den Grundlagen bis zu Datenbank- und Netzanwendungen", Springer-Vieweg, 9. Auflage, 2016
- Balzert, H.: "Lehrbuch der Softwaretechnik", Bände 1, 2, 3 (Basiskonzepte und Requirements-Engineering, Entwurf, Implementierung, Installation und Betrieb, Softwaremanagement), Spektrum Akademischer Verlag, 3. Auflage, 2009
- Kleuker, S.: "Grundkurs Software-Engineering mit UML: Der pragmatische Weg zu erfolgreichen Softwareprojekten", Springer-Vieweg, 4. Auflage, 2018
- Liggesmeyer, P.: "Software-Qualität: Testen, Analysieren und Verifizieren von Software", Spektrum Akademischer Verlag, 2. Auflage, 2009
- Rupp, C.: "Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil", Carl Hanser, 6. Auflage, 2014
- Sommerville, I.: "Software Engineering", Addison-Wesley, 9. Auflage, 2012
- Wirdemann, R.: "Scrum mit User Stories", Carl Hanser, 3. Auflage, 2017
- Zörner, S.: "Softwarearchitekturen dokumentieren und kommunizieren: Entwürfe, Entscheidungen und Lösungen nachvollziehbar und wirkungsvoll festhalten", Carl Hanser, 2. Auflage, 2015