

Modul: Software Engineering and Continuous Integration

Semester: 2

Code der Veranstaltung: SEN

ECTS Punkte: 5

Kontaktstunden: 40

Selbststudium: 85

Dauer des Moduls: 1 Semester

Art des Moduls: Pflicht

Häufigkeit des Angebots: jedes Studienjahr

Gewichtung der Note in der Gesamtnote: 5/180

Art und Umfang der Prüfungsleistung: KRS 90: Die Prüfung besteht entweder aus einer Klausur oder einem Referat oder einer Studienarbeit; im Fall einer Klausur gibt die Zahl den Umfang der Klausur in Minuten.

Modulverantwortliche/r:

Prof. Dr. Kahnert, Prof. Dr. Ewering

Lehrsprache: Deutsch

Dozenten: Prof. Dr. Ewering, Prof. Dr. Kahnert, Prof. Dr. Rohde, Prof. Dr. Weigand

Zugangsvoraussetzungen:

Einführung in Wirtschaftsinformatik und Digitalisierung

Verwendbarkeit des Moduls:

- Spezialisierung Software Engineering
- Spezialisierung IT-Consulting
- Webtechnologies and Applications
- Aktuelle Themen in der IT

Qualifikations- und Kompetenzziele des Moduls:

Die Studierenden kennen die Relevanz und Bandbreite des Software Engineerings. Sie verstehen ausgewählte Themenfelder und zugehörige Instrumentarien, beherrschen deren Anwendung und können sie praxisorientiert umsetzen.

Lehr- und Lernmethoden des Moduls:

Unterschiedliche Lehr-/Lernumgebungen: Präsenzveranstaltungen, Eigenstudium; Wechselnde Lehr-/Lernmethoden: Individuelles und kooperatives Lernen, problemorientiertes und integratives Lernen, synchrones und asynchrones Lernen; Fallstudienarbeit, Expertenvorträge.

Besonderheiten

Übungsaufgaben, Literaturstudium, Literaturquellensuche und -nutzung, Internetrecherche, Transfer und Querverbindungen zu Praxisunternehmen und anderen Modulen

Inhalt des Moduls:

- 1 Einführung
 - 1.1 Motivation
 - 1.2 Software Life Cycle
 - 1.3 Ethik des Software Engineerings

- 2 Vorgehensmodelle im Software Engineering
 - 2.1 Wasserfall- und allg. V-Modell
 - 2.2 „Plangetriebene“ Modelle
 - 2.3 „Agile“ Modelle
 - 2.4 Continuous Integration

- 3 Requirements Engineering
 - 3.1 Relevanz und Herausforderungen
 - 3.2 Arten von Anforderungen
 - 3.3 Vorgehensweisen in der Praxis

- 4 Spezifikationstechniken
 - 4.1 UML
 - 4.2 Objektorientierte Analyse

- 5 Software-Tests
 - 5.1 Teststufen
 - 5.2 Testprozesse
 - 5.3 Ausgewählte Testarten

Grundlegende Literaturhinweise

BALZERT, H., 2009. Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, Heidelberg: Spektrum

SOMMERVILLE, I., 2017. Software Engineering, München: Pearson.

Ergänzende Literaturempfehlungen

LUDEWEG, J. und H. LICHTER., 2013. Software Engineering, Heidelberg: dpunkt.verlag

BALZERT, H., 2008. Lehrbuch der Softwaretechnik: Softwaremanagement, Heidelberg: Spektrum.