

## **Modul: Software Engineering Project**

**Semester:** 6

**Code der Veranstaltung:** SEP

**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 an.

**Modulverantwortliche/r:**

Prof. Dr. Rohde, Prof. Dr. Nüßer

**Lehrsprache:** Deutsch

**Dozenten:** Dr. habil. M. Borschbach, Prof. Dr. Rohde, Prof. Dr. Nüßer

### **Zugangsvoraussetzungen:**

- Software Testing & DevOps
- Software Engineering & Continuous Integration

### **Verwendbarkeit des Moduls:**

#### **Qualifikations- und Kompetenzziele des Moduls:**

Die Studierenden können mit modernen integrierten Entwicklungsumgebungen (IDE) umgehen und die technische Infrastruktur für ein integriertes Softwareprojekt aufbauen und einsetzen. Sie verstehen den Ablauf moderner Softwareentwicklung und können mit unterstützenden Techniken wie Kanban Boards, Bug-Tracking-Systemen, Source-Code-Verwaltung, Version-Control-Systemen usw. umgehen. Sie verstehen den Prozess der agilen Softwareentwicklung (z.B. Scrum) im Team und von Continuous Integration und können diese in kleinen Projektgruppen anwenden.

Die Modulinhalt werden im Rahmen des praktischen Projektes erarbeitet. Das genaue Thema der zu erstellenden Anwendung und die zugehörige Literatur werden in der Veranstaltung bekanntgegeben.

#### **Lehr- und Lernmethoden des Moduls:**

Unterschiedliche Lehr-/Lernumgebungen: Präsenzveranstaltungen, Vorträge, praktische Übungen, ein im Team selbst entwickeltes Softwareprojekt; Wechselnde Lehr-/Lernmethoden: Individuelles und kooperatives Lernen, problemorientiertes und integratives Lernen, synchrones und asynchrones Lernen; Fallstudienarbeit, Expertenvorträge.

## **Inhalt des Moduls:**

1. Einführung
  - 1.1 Software Application Lifecycle Management
  - 1.2 Agile Softwareentwicklung im Team
  - 1.3 Continuous Integration / Continuous Delivery / DevOps
  - 1.4 Industrielle Softwareentwicklung
  
2. Werkzeuge und Techniken
  - 2.1 Integrierte Entwicklungsumgebungen
  - 2.2 Source-Code-Verwaltung
  - 2.3 Automatische Build-Systeme
  - 2.4 Version-Control-Systeme
  - 2.5 Bug-Tracking-Systeme
  - 2.6 Release Management
  - 2.7 Programmierrichtlinien
  - 2.8 Continuous Inspection
  
3. Ausgewählte Anwendungsszenarien
  - 3.1 GUI-Anwendungen
  - 3.2 Webanwendungen
  - 3.3 Verteilte Systeme

### **Hinweis**

Software Application Lifecycle Management, agile Softwareentwicklung und Continuous Integration sind ebenfalls Inhalt im Modul „Software Engineering & Continuous Integration“ (alle Wirtschaftsinformatiker). Der Fokus sollte in diesem Modul auf der praktischen Anwendung und der Werkzeugunterstützung liegen.

### **Grundlegende Literaturhinweise:**

Grundlegende Literatur wie im Modul „Software Engineering & Continuous Integration“

### **Ergänzende Literaturempfehlungen:**

Weiterführende Literatur wird je nach Thema in der Veranstaltung bekanntgegeben.