4.3.5 Software Engineering und Verteilte Systeme

Software Engineering und Verteilte Systeme

Modulübersicht
EDV-Bezeichnung: QUCB350
Modulverantwortliche(r): Prof. Dr. NN
Modulumfang (ECTS): 6 CP
Einordnung (Semester): 3. Semester
Inhaltliche Voraussetzungen: Kenntnisse aus den Modulen Informatik 1 und 2
Voraussetzungen nach SPO: Nach SPO sind keine formellen Voraussetzungen erforderlich.
Lernergebnisse und Kompetenzen Die Studierenden vertiefen ihre Kenntnisse der Softwareentwicklung und erlernen Methoden zum systematischen Entwurf, zur Implementierung und zum Betrieb verteilter Systeme. Sie sind in der Lage, Softwareprojekte strukturiert zu planen, im Team umzusetzen und deren Qualität zu sichern. **Fachliche Kompetenzen** Die Studierenden • analysieren und strukturieren Softwareprojekte nach etablierten Vorgehensmodellen, • wenden agile Methoden (z. B. Scrum) und klassische Entwicklungsprozesse sicher an, • modellieren und dokumentieren Softwarestrukturen mit UML, • entwerfen und implementieren Architekturen verteilter Systeme unter Berücksichtigung von Modularität, Kommunikation, Fehlertoleranz, Replikation und Sicherheit, • nutzen aktuelle Plattformen, Frameworks und Tools zur Entwicklung und Integration von Softwarekomponenten. **Methodische Kompetenzen** Die Studierenden • wenden iterative Entwicklungsprozesse und Versionskontrolle in Teamprojekten an, • nutzen Werkzeuge für Modellierung, Build-Automatisierung, Test und Deployment, • reflektieren Architekturentscheidungen im Hinblick auf Wartbarkeit, Skalierbarkeit und Sicherheit, • dokumentieren und präsentieren Projektergebnisse adressatengerecht. **Sozial- und Selbstkompetenzen** Die Studierenden • arbeiten kooperativ in Entwicklungsteams,

• übernehmen Verantwortung für Teilaufgaben und Zeitmanagement im Projektkontext, • kommunizieren effektiv in technischen und organisatorischen Fragestellungen, • entwickeln Selbstorganisation und Problemlösungsfähigkeit in agilen Teams.
Prüfungsleistungen: Software Engineering und Verteilte Systeme: Klausur, 120 Minuten Labor Software Engineering und Verteilte Systeme (Studienleistung): Das Labor gilt als bestanden, wenn die Projekte erfolgreich bearbeitet und testiert wurden.

Lehrveranstaltung: Software Engineering und Verteilte Systeme
EDV-Bezeichnung: QUCB351
Dozierende(r): Prof. Dr. NN
Umfang (SWS): 4
Turnus: Wintersemester
Art, Modus: Vorlesung, Pflichtfach
Lehrsprache: Deutsch
Studieninhalte: • Grundlagen des Software Engineering: Lebenszyklus, Vorgehensmodelle, Rollen und Artefakte • Klassische und agile Entwicklungsprozesse (Wasserfall, V-Modell, Scrum, Kanban) • Anforderungsanalyse, Use Cases, Pflichtenheft, User Stories • Modellierung mit UML: Anwendungsfall-, Klassendiagramme, Sequenz- und Zustandsdiagramme • Softwarearchitekturen: Schichten-, Komponenten- und Serviceorientierung • Grundlagen verteilter Systeme: Kommunikationsmodelle, Synchronisation, Fehlertoleranz • Konzepte der Parallelität, Replikation und Konsistenz • Grundlagen der Netzwerksicherheit und Zugriffskontrolle • Qualitätssicherung: Tests, Reviews, Continuous Integration / Continuous Deployment • Einführung in aktuelle Frameworks (z. B. Spring Boot, REST, gRPC, Docker)
Empfohlene Literatur: • Ian Sommerville: Software Engineering, 10. Aufl., Pearson, 2018. • Craig Larman: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3. Aufl., Pearson, 2004. • Bernd Oestereich: Analyse und Design mit UML 2.1: Objektorientierte Softwareentwicklung, 8. Aufl., Oldenbourg, 2006. • Ken Schwaber, Jeff Sutherland: The Scrum Guide: The Definitive Guide to Scrum, Scrumguides.org, 2020. • Gen Kim, Jez Humble, Patrick Debois, John Willis: Das DevOps-Handbuch: Teams, Tools und Infrastrukturen erfolgreich umgestalten, O'Reilly, 2017. • Ivar Jacobson, Grady Booch, James Rumbaugh: The Unified Software Development Process, Addison-Wesley, 1999. • Andre S. Tanenbaum, Maarten van Steen: Verteilte Systeme: Prinzipien und Paradigmen, 2. Aufl., Pearson, 2007.

• Georg Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair: Distributed Systems: Concepts and Design, 5. Aufl., Addison-Wesley, 2011.

Lehrveranstaltung: Labor Software Engineering und Verteilte Systeme
EDV-Bezeichnung: QUCB352
Dozierende(r): Prof. Dr. NN
Umfang (SWS): 2
Turnus: Sommersemester
Art, Modus: Übungen, Pflichtfach
Lehrsprache: Deutsch
Studieninhalte: • Praktische Umsetzung der Vorlesungsinhalte in iterativen Entwicklungsprojekten • Teamarbeit mit Versionsverwaltung (GIT) und agiler Aufgabenplanung (Scrum / Kanban-Board) • Entwurf und Implementierung einer verteilten Beispielanwendung (z. B. Client-Server- oder Microservice-Architektur) • Nutzung moderner Frameworks und APIs (z. B. REST, WebSockets, MQTT) • Anwendung von Build-, Test- und Deployment-Werkzeugen (Gradle, JUnit, Docker, CI/CD-Pipeline) • Automatisierte Tests, Code-Reviews und Dokumentation • Präsentation und Reflexion der Projektergebnisse
Empfohlene Literatur: • Siehe Vorlesung Software Engineering und Verteilte Systeme