Software Engineering II
(W3WI_SE304)

FORMAL INFORMATION ON THE MODULE

MODULE #	LOCATION IN THE COURSE OF STUDY	MODULE DURATION (SEMESTER)	SEMESTER	LANGUAGE
W3WI_SE304	2nd academic year	1	Spring Term	English

FORMS OF TEACHING USED

Lecture, exercise, laboratory exercise, case study

FORMS OF EXAMINATION USED

EXAM PERFORMANCE	EXAM DURATION (IN MINUTES)	GRADING
Written exam or portfolio	120	yes

WORKLOAD AND ECTS CREDITS

TOTAL WORKLOAD (IN H)	OF WHICH ATTENDANCE TIME (IN H)	OF WHICH SELF-STUDY (IN H)	ECTS CREDIT POINTS
150	55	95	5

QUALIFICATION OBJECTIVES AND COMPETENCIES

PROFESSIONAL COMPETENCE

Students know the phase-specific approaches and methods of the software life cycle, the tasks of software architects and the characteristics of modern software architectures.

METHODOLOGICAL COMPETENCE

Students will be able to assess the relevance of methods and techniques in the software life cycle and consider when they should be used. They can use common design patterns and current frameworks.

PERSONAL AND SOCIAL COMPETENCE

Students can independently develop appropriate architectures and solutions in the software life cycle in case studies and justify them in a comprehensible manner.

OVERARCHING COMPETENCE

Students understand the overarching relationships between process steps and architecture models in software development and can successfully apply this understanding in practical projects and advanced lectures.

LEARNING UNITS AND CONTENT

TEACHING AND LEARNING UNITS	PRESENCE TIME	SELF-STUDY
Core concepts and methods of software engineering	28	48

- Software life cycle: Agile process models and process models.
- Requirements engineering: requirements, software quality, security, reliability.
- Design and implementation: design concepts, design patterns, tools,
development environments, version management and configuration control.
- Testing: black box, white box, control flow, data flow, usability and test levels and test types.
- Maintenance and further development: regression tests, reverse engineering, reengineering.
- Management: introduction and integration of software, quality management,
Software metrics, configuration management, risk management, change management.

LEARNING UNITS AND CONTENT

TEACHING AND LEARNING UNITS	PRESENCE TIME	SELF-STUDY
Modern software architectures	27	47

Software architecture concept, tasks and methods of the software architect,
service-oriented architectures,
Enterprise architectures, architecture patterns, current frameworks and modern
software architectures using examples.

SPECIAL FEATURES

The examination duration only applies to the written examination.

PREREQUISITES

Basic concepts of IT; Programming II;
Software Engineering I

LITERATURE

- Cohn, M.: Agile Software Development: With Scrum to Success, Addison-Wesley, Munich.
- Sommerville, I.: Software Engineering, Pearson Studium, Munich.
- Spillner, A. and Linz, T.: Basiswissen Softwaretest, dpunkt.verlag, Heidelberg.
- Starke, G.: Effektive Software-Architekturen: Ein praktischer Leitfaden, Hanser, Munich.
- Wolf, H. and Bleek, W.-G.: Agile Softwareentwicklung, dpunkt.Verlag, Heidelberg.