

	is able to select and to use appropriate cost estimation methods and project management software in practical projects. He or she is able to apply methods for risk management and to use maturity models for improving processes.
<i>Index</i>	<ul style="list-style-type: none"> ■ Brief recap of software project management basics (e.g. work break down structure, project organization) ■ Methods for planning and controlling projects ■ Estimation methods: efforts, costs ■ Network planning techniques ■ Project management software ■ Testing methods ■ Risk management ■ Maturity models
<i>Pre-Exam Requirements</i>	Presentation in class
<i>Method and Extent of Examination</i>	Oral Examination, 20 minutes
<i>Media</i>	Blackboard, Powerpoint, Computer Demonstration, moodle
<i>Recommended Literature (Excerpt)</i>	<ul style="list-style-type: none"> ■ Ahren, D.M., A. Clouse, R. Turner: CMMI Distilled 2nd Edition. Addison Wesley, 2003. ■ Futrell, R.T., D.F. Shafer, L.I. Shafer: Quality Software Project Management. Software Quality Institute Series. Prentice Hall, 2002. ■ IEEE Guide to the Software Engineering Body of Knowledge (SWEBOK); IEEE, 2004. ■ PMI Standards Committee: A Guide to the Project Management Body of Knowledge (PMBOK); Project Management Institute, 3rd edition, 2004. ■ subject specific additional literature, project management software

4 Software Architecture

<i>Course</i>	Master Software Technology, Digital Processes
<i>Name of Module</i>	Software Architecture
<i>Abbreviation</i>	SWA
<i>Semester</i>	1
<i>Responsible</i>	Prof. Dr. Deininger
<i>Lecturers</i>	Prof. Dr. Deininger, Prof. Dr. Wanner, Prof. Dr. Speiser
<i>Elective / compulsory</i>	Compulsory module
<i>Frequency</i>	Every year

<i>Weekly Contact Hours</i>	4h (3h Lectures + 1h Exercises)
<i>Method of Teaching</i>	Lecture with theoretical and practical exercises
<i>Student Work Load – Lectures</i>	68h
<i>Student Work Load – Self Studies</i>	112h
<i>European Credit Transfer Points</i>	6 ECTS Points
<i>Necessary Previous Knowledge</i>	Software Engineering, Object Oriented Software Implementation
<i>Final Knowledge and Skills</i>	<p><i>Knowledge and understanding</i> On completion the student knows the different interrelationships between requirements and design and architectural choices. He or she knows the principles of software design and the different design views and knows how a system design affects the testability of a system.</p> <p><i>Disciplinary / professional skills</i> On completion the student is able to develop different design views and select fitting patterns for certain problems and draw from architectural choices. He or she is able to select and use appropriate modeling techniques. He or she can rate the consequences of certain design decisions Students can earn the official iSAQB certificate at the end of the semester.</p>
<i>Index</i>	<p><i>Based on iSAQB Certified Professional for Software Architecture Foundation Level (https://www.isaqb.org/):</i></p> <ul style="list-style-type: none"> ■ Basic Concepts ■ Roles, Responsibilities, and Activities of Architects ■ Deriving Quality Goals and Design Constraints ■ Design Principles ■ Patterns ■ Additional design considerations ■ Design Approaches and Methods ■ Documentation and Communication ■ Software Architecture Evaluation ■ Domain Driven Design ■ QA-Architecture
<i>Pre-Exam Requirements</i>	Assignments
<i>Method and Extent of Examination</i>	Written examination, 120 minutes
<i>Media</i>	Blackboard, Powerpoint, Computer Demonstration, moodle
<i>Recommended Literature (Excerpt)</i>	<ul style="list-style-type: none"> ■ Bass, L., P. Clements, R. Kazman: Software Architecture in Practice, 3rd edition, Addison-Wesley Professional, 2012.

	<ul style="list-style-type: none"> ■ Buschmann, F., R. Meunier, H. Rohnert, P. Sommerlad, M. Stal: Pattern-Oriented Software Architecture: A System of Patterns, John Wiley & Sons, 1996. ■ Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, J. Stafford: Documenting Software Architectures, 2nd edition, Addison-Wesley, 2010. ■ Evans, E.: Domain-Driven-Design, Addison- Wesley, 2008. ■ Fowler, M.: Patterns of Enterprise Application Architecture; Addison-Wesley, 2014. ■ Gamma, E., R. Helm, R. Johnson, J. Vlissides: Design Patterns: Elements of Reusable OO Software. Addison-Wesley, 1997. ■ Martin, R.C.: Clean Architecture: A Craftsman's Guide to Software Structure and Design: A Craftsman's Guide to Software Structure and Design Addison-Wesley, 2018 ■ Meyer, B.: Object-Oriented Software Construction. Prentice Hall, 1997. ■ Shaw, M., P. Clements: The Golden Age of Software Architecture, IEEE SOFTWARE, March/April 2006, 31-39.
--	---

5 Intercultural Training

<i>Course</i>	Master Software Technology
<i>Name of Module</i>	Intercultural Training
<i>Abbreviation</i>	ICT
<i>Semester</i>	1 and 2

Anmerkung: Intercultural Training 1 und 2 wird zu Beginn des ersten und zweiten Semesters als Blockveranstaltung durchgeführt und dient im Wesentlichen zur Integration der neuen Studenten und insbesondere der Quereinsteiger in die bestehende Gruppe. Aus diesem Grund überschneiden sich auch die Inhalte der beiden Module.

<i>Course</i>	Master Software Technology
<i>Name of Module</i>	Intercultural Training (Part 1)
<i>Abbreviation</i>	ICT1
<i>Semester</i>	1
<i>Responsible</i>	Course Director Software Technology
<i>Lecturers</i>	External Lecturer
<i>Elective / compulsory</i>	Compulsory module
<i>Frequency</i>	Every year
<i>Weekly Contact Hours</i>	1h (held as a Block-course)