Valid from 2025.HS

Module description: Software Engineering

Module Code	t.BA.WV.SE-EN.25HS
ECTS Credits	4
Language of Instruction/Examination	English
Organizational Unit	InIT
Module Coordinator	Michael Wahler
Legal Framework	The module description is part of the legal basis in addition to the general academic regulations. It is binding. During the first week of the semester a written and communicated supplement can specify the module description in more detail.
Module Characteristic	Type 2a 4 consecutive lecture lessons per semester week and class
Module Description	Building a treehouse is a simple task. With basic tools, you can construct a basic shelter over the course of a weekend. However, constructing a multi-story house with electricity, plumbing, waste disposal, heating, and ventilation, designed to last 50 years or more, requires a team of specialists, advanced tools, and a well-coordinated process. Writing a simple program is akin to building a treehouse. In contrast, creating a robust, efficient, and sustainable software system—one that is designed to endure over time—demands specialized skills and is rarely a task for just one person. In this module, students will explore the life cycle of complex software systems and the processes, skills, and tools necessary to create high-quality digital solutions.

Module description: Software Engineering

Module Content

Software Development Life Cycle and Processes

- SDLC (including release management, operations and evolution)
- Software engineering process models (agile, waterfall, …)
- Team setup and roles
- SCRUM planning and execution
- Risk management in development and operations
- Code management, code as team effort
- Sustainability aspects (e.g., maintainability, energy consumption)
- Managing technical debt

Requirements Elicitation and Management

- Use cases and user stories
- Requirements: functional vs. nonfunctional
- Prioritization of requirements
- What makes software valuable?
- Domain diagrams

Software Architecture

- Components and interfaces
- UML
- Clean Architecture, SOLID
- Design Patterns
- Top-down vs. bottom-up design, architect's elevator
- Frameworks and libraries
- Architectural decisions and knowledge management

Implementation, Testing, Integration, Deployment

- Programming languages and technology stack
- Testing concepts and test automation
- Continuous integration and deployment
- Software maintenance and evolution
- Managing change

Software Quality

- Software engineering standards (ISO/IEC 25010)
- Medical Informatics standards (IEC 62304, IEC 82304)
- Compliance with agile & audits

Prerequisite Knowledge

- Basic knowledge of object-oriented programming

Learning Objectives (Competences)

Students...	Competencies	Taxonomies
The students understand the phases of the software development life cycle and know process models to organize these phases.	M, F	K3
The students can elicit functional and non-functional requirements and document them.	F, SO, M	K3
The students can design and document a sustainable software architecture and design that meets the requirements.	M, F	K4
The students can implement their designs, test them, and deploy them.	F, M	K3
The students know what software quality means and how it can be applied to a software development process.	M, F	K3

Module description: Software Engineering

Performance Assessment

End-of-module exam	Assessment	Length (min.)	Weighting	Form
written exam	Grade	90	70	acc. to module agreement

Performance assessment during the semester	Assessment	Length (min.)	Weighting	Form
written + oral	Grade		30	acc. to module agreement

Classroom Attendance Requirement

None

Learning material

- Tulio Valente, M. (2024). Software Engineering: A Modern Approach. Self-published. ISBN 978-65-01-26467-7. https://softengbook.org/.
- Sommerville, I. (2015). Software Engineering. 10 Edition. Pearson. ISBN 9780137586691. https://www.oreilly.com/library/view/software-engineering-10th/9780137586691/.

Comments

30% of the module grade will consist of practical work and presentations during the semester.