Software Engineering & IT-Projektmanagement

Winter

(engl. Software Engineering & IT-Projectmanagement)

Modulnummer: FIN-INF-110476

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
SE&IT-PM	5	Winter	starting at 3.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	
Verantwortung:	Dr. Thomas Wilde
Dozent:in:	Dr. Thomas Wilde
Lehrveranstaltungen:	• Vorlesung *Software Engineering & IT-Projektmanagement* • Übung *Software Engineering & IT-Projektmanagement*
Verwendbarkeit:	- B.Sc. INF: Informatik I - Pflicht - B.Sc. INF: Kernmodule (SPO 2027) - B.Sc. CV: Informatik I - Pflicht - B.Sc. CV: Kernmodule (SPO 2027) - B.Sc. INGINF: Informatik I - Pflicht - B.Sc. INGINF: Kernmodule (SPO 2027) - B.Sc. WIF: Gestalten und Anwenden - Wahlpflicht - B.Sc. WIF: Kernmodule (SPO 2027) - B.Sc. INF (bilingual): Informatik I - Pflicht - B.Sc. INF (bilingual): Kernmodule (SPO 2027)

Angestrebte Lernergebnisse:
Students who have successfully completed the course ...

- understand the term “software engineering”, its meaning and the necessity of structured software development.
- know different process models, their advantages and disadvantages and the core activities of software engineering.
- apply test methods and refactoring to ensure software quality and avoid errors.
- understand and use agile methods such as Scrum and Kanban and compare them with traditional approaches.
- use version control, IDEs and CI/CD pipelines effectively and understand Continuous Integration & Deployment.
- write readable, maintainable and efficient code according to proven principles and conventions.
- estimate effort and resources, plan projects and monitor their progress.
- know architecture concepts and design patterns and apply them in software projects.
- work effectively in a team, present ideas clearly and document their work.

Inhalt:

- Software engineering - what is it and what is it used for?
- Process models: sequential & iterative,
- Process activities: specification, development, validation, evolution,
- Quality assurance: testing & refactoring,
- Agile software development,
- Tools & Tools,
- Clean coding / code conventions,
- Continuous Integration, Continuous Deployment, Continuous Delivery,
- IT project management: estimation & process planning,
- Software architecture,
- design patterns,
- Practical examples.

Arbeitsaufwand:

- Attendance time: 28h lecture + 28h exercise
- 94h independent work

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:

	Examination prerequisite required, will be announced at the beginning of the semester; • Exam: written exam, 120 minutes; • Ungraded (*Schein*) (SPO until 2022): Completion of the tasks & participation in the exercises.	• Lecture (2 SWS) • Exercise (2 SWS)

Voraussetzungen nach Prüfungsordnung:	**Empfohlene Voraussetzungen:**
none	Successful completion of modules: • Einführung in die Informatik, • Algorithmen und Datenstrukturen.

Medienformen:	**Literatur:**
	• Ian Sommerville. *Software Engineering*, 10. Auflage. Pearson, 2016 • Erich Gamma, Richard Helm, Ralph E. Johnson, John Vlissides: *Design Patterns. Elements of Reusable Object-Oriented Software*, Addison Wesley, 1997 • Robert Martin: *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall, 2008 Für alle Lehrbücher gilt, dass die Auflage und ggf. die Wahl zwischen englischem Original oder deutscher Übersetzung für uns irrelevant ist. Jede ältere Auflage ist in Ordnung.

Hinweise:

Revision #1
Created 2026-10-31 09:07:54 UTC by Fakultät für Informatik
Updated 2026-10-31 09:07:54 UTC by Fakultät für Informatik