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	ab 3.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	false
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:
Studierende, die den Kurs erfolgreich absolviert haben ...

- ... verstehen den Begriff "Software Engineering", seine Bedeutung und die Notwendigkeit strukturierter Softwareentwicklung.
- ... kennen verschiedene Prozessmodelle, deren Vor- und Nachteile sowie die Kernaktivitäten des Software Engineerings.
- ... wenden Testmethoden und Refactoring an, um Softwarequalität zu sichern und Fehler zu vermeiden.
- ... verstehen und nutzen agile Methoden wie Scrum und Kanban und vergleichen sie mit traditionellen Ansätzen.
- ... setzen Versionskontrolle, IDEs und CI/CD-Pipelines effektiv ein und verstehen Continuous Integration & Deployment.
- ... schreiben lesbaren, wartbaren und effizienten Code nach bewährten Prinzipien und Konventionen.
- ... schätzen Aufwand und Ressourcen ein, planen Projekte und überwachen deren Fortschritt.
- ... kennen Architekturkonzepte und Design-Patterns und wenden diese in Softwareprojekten an.
- ... arbeiten effektiv im Team, präsentieren Ideen klar und dokumentieren ihre Arbeit.

Inhalt:

- Software Engineering - Was ist das und wozu wird es gebraucht?
- Prozessmodelle: sequentielle & iterativ,
- Prozessaktivitäten: Spezifikation, Entwicklung, Validierung, Evolution,
- Qualitätssicherung: Testint & Refactoring,
- Agile Softwareentwicklung,
- Tools & Werkzeuge,
- Clean-Coding / Code-Conventions,
- Continuous Integration, Continuous Deployment, Continuous Delivery,
- IT-Projektmanagement: Schätzung & Prozessplanung,
- Softwarearchitektur,
- Design-Patterns,
- Praktische Beispiele.

Arbeitsaufwand:

- Präsenzzeit: 28h Vorlesung + 28h Übung
- 94h selbstständige Arbeit

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:

• Um an der Prüfung teilnehmen zu können, sind Prüfungsvorleistungen erforderlich. Details werden in der ersten Vorlesung bekannt gegeben.	• Prüfung: schriftliche Klausur, 120 Minuten; • Schein (SPO bis 2022): Bearbeitung der Aufgaben & Teilnahme an den Übungen.	• Vorlesung (2 SWS) Übung (2 SWS)

Voraussetzungen nach Prüfungsordnung:	Empfohlene Voraussetzungen:
keine	Erfolgreicher Abschluss von: • 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 2026-10-01 07:06:13 UTC by Fakultät für Informatik
Updated 2026-10-01 07:06:13 UTC by Fakultät für Informatik