

Modul: Improvement of Software (6 Credits)	
Name im Diploma Supplement	Improvement of Software
Verantwortlich	Prof. Dr. Dominik Sobania
Voraussetzungen	Siehe Prüfungsordnung.
Workload	180 Stunden studentischer Workload gesamt, davon: • Präsenzzeit: 50 Stunden • Vorbereitung, Nachbereitung: 70 Stunden • Prüfungsvorbereitung: 60 Stunden
Dauer	Das Modul erstreckt sich über 1 Semester.
Qualifikationsziele	Die Studierenden • wissen, was funktionale und nicht-funktionale Software-Eigenschaften sind • kennen die Grundlagen von Large Language Models und des Search-Based Software Engineerings • erwerben einen Überblick über klassische und agentische Verfahren zur automatischen Optimierung von Software und lernen wann sich welche Verfahren eignen • vertiefen den Vorlesungsstoff durch praktische Übungen und das Praxisprojekt
Praxisrelevanz	Kenntnisse zum Einsatz automatisierter Verfahren und Künstlicher Intelligenz gewinnen im Software Engineering zunehmend an Bedeutung.
Prüfungsmodalitäten	Zum Modul erfolgt eine modulbezogene Prüfung in Form einer semesterbegleitenden, projektbasierten Prüfungsleistung mit schriftlicher Ausarbeitung und Präsentation. Details zum Umfang, den Abgabefristen und Bewertungskriterien werden zu Beginn der Veranstaltung bekannt gegeben
Verwendung in Studiengängen	• M.Sc. Software and Network Engineering • M.Sc. Informatik
Bestandteile	• Vorlesung mit integrierter Übung: Improvement of Software (6 Credits)

Vorlesung mit integrierter Übung: Improvement of Software (6 Credits)

Name im Diploma Supplement	Improvement of Software		
Anbieter	AG Generative Künstliche Intelligenz im Software Engineering - https://aise.informatik.uni-due.de/		
Lehrperson	Prof. Dr. Dominik Sobania		
SWS	4	Sprache	deutsch
Turnus	Sommersemester	maximale Hörschaft	25

empfohlenes Vorwissen

Kenntnisse in Python und Java, Grundlagen des Software Engineerings

Lehrinhalte

- Funktionale- & nicht-funktionale Software-Eigenschaften
- Performance-Messung & Software-Benchmarks
- Search-Based Software Engineering (SBSE)
- Lokale und evolutionäre Suche
- Multi-Objective Optimization
- Genetic Improvement (GI)
- Automated Program Repair (APR)
- Grundlagen von Large Language Models (LLMs)
- Modelltypen und praktischer Einsatz
- Context Engineering
- Agentic AI
- Agenten-Architekturen für das Software Engineering
- Tool Use & Function Calling
- Zustands-Management (Memory)
- Hybride Ansätze

Literaturangaben

Literaturangaben und Links werden im Semester online zur Verfügung gestellt.