

Modulbeschreibung: AISE502 - AI in SE II

Autor/in: Herzog Florian
Ausgabestelle: Institute for Data Analysis, Artificial Intelligence, Visualization, and Simulation (DAViS)
Geltungsbereich: Departement
Klassifizierung: Nicht klassifiziert
Version: 1.0.0
Ausgabedatum: 05.12.2024

Modul

Name	AI in Software Engineering II		
Kürzel	AISE502	ECTS-Punkte	4
Typ	Pflichtmodul		
Verantwortliche/r			
Leitidee	Das Modul "AI in Software Engineering II" baut auf den Grundlagen des ersten Teils auf und vertieft das Verständnis für die Anwendung fortgeschrittener Künstlicher Intelligenz (KI) Techniken im Softwareentwicklungsprozess. Der Fokus liegt auf der Verbesserung der Codequalität durch KI-gestützte Methoden, der Entwicklung intelligenter AI Agents sowie der Anwendung komplexer Prompt Engineering Techniken wie Chain of Thoughts und Graphs of Thoughts. Zusätzlich werden wesentliche Aspekte der Softwarearchitektur behandelt, um die Integration von KI in robuste und skalierbare Softwarelösungen zu gewährleisten. Studierende lernen, wie sie fortschrittliche KI-Modelle nutzen können, um effizienteren, saubereren und robusteren Code zu schreiben sowie komplexe Entwicklungsaufgaben zu automatisieren und zu optimieren. Durch praxisorientierte Projekte und intensive Übungen erwerben die Studierenden die Fähigkeiten, komplexe AI-gestützte Softwarelösungen zu entwerfen, zu implementieren und zu evaluieren, die den gesamten Softwareentwicklungszyklus unterstützen und verbessern.		
Voraussetzungen	1. AI in Software Engineering 2. Software Technik I 3. Semantik von Programmiersprachen		
Lernergebnisse	1. Fortgeschrittene KI-Techniken verstehen: – Die Konzepte und Anwendungen von AI Agents im Software Engineering kennen.		

	– Fortgeschrittene Prompt Engineering Techniken wie Chain of Thoughts und Graphs of Thoughts verstehen und anwenden können. Komplexe AI-gestützte Software-Tools entwickeln: – Methoden zur automatisierten Code-Optimierung und -Refactoring mit KI einsetzen. – Tools zur statischen Codeanalyse und Fehlerbehebung mithilfe von KI entwickeln und nutzen. Softwarearchitektur integrieren: – Grundlegende und fortgeschrittene Prinzipien der Softwarearchitektur verstehen und anwenden. – Architekturmuster für die Integration von KI in Softwarelösungen kennenlernen und implementieren, insbesondere Microservices. – Best Practices für die Gestaltung skalierbarer und wartbarer KI-gestützter Systeme anwenden. Intelligente AI Agents entwickeln: – AI Agents konzipieren und implementieren, die spezifische Aufgaben im Softwareentwicklungsprozess autonom ausführen. – Interaktive und adaptive AI Agents erstellen, die mit Entwicklern kommunizieren und kooperieren können. Komplexe Prompt Engineering Techniken anwenden: – Chain of Thoughts zur Verbesserung der Problemlösungsfähigkeiten von LLMs einsetzen. – Graphs of Thoughts zur Strukturierung und Visualisierung von Denkprozessen in KI-Modellen nutzen. Fortgeschrittene Projekte realisieren: – Komplexe AI-gestützte Softwarelösungen planen, entwickeln und dokumentieren. – Best Practices der Softwareentwicklung und KI-Integration anwenden, um hochqualitative und wartbare Softwareprojekte zu erstellen. Performance und Skalierbarkeit optimieren: – Methoden zur Evaluierung und Verbesserung der Performance von fortgeschrittenen KI-Modulen in Softwareanwendungen anwenden. – Skalierbare und effiziente KI-Lösungen für größere Softwareprojekte entwickeln. Ethik und Verantwortung weiter vertiefen: – Vertiefte ethische Überlegungen und gesellschaftliche Implikationen des Einsatzes von fortgeschrittener KI im Software Engineering reflektieren. – Verantwortungsbewusste und nachhaltige KI-Lösungen entwickeln, die ethischen Standards entsprechen.
Leistungsnachweis	Der Leistungsnachweis besteht aus : – 40% Praxisprojekt

	– 20% Präsentation – 10 % Assignments – 30 % schriftliche Prüfung
Nachprüfung	

Unterrichtsprache	Deutsch
Eingangskompetenzen	
Inhalte	1. Fortgeschrittene Einführung in AI und Software Engineering II 1. Inhalte: i. Rückblick auf AI in Software Engineering I. ii. Einführung in fortgeschrittene Konzepte und Techniken. iii. Überblick über die Kursstruktur und die Projektanforderungen. 2. Praktische Anwendung: i. Analyse von erweiterten AI-gestützten Code-Editoren und Entwicklungswerkzeugen. 2. Fortgeschrittene Large Language Models (LLMs) und ihre Anwendungen 1. Inhalte: i. Vertiefung der Architektur und Funktionsweise von LLMs. ii. Anpassung und Feinabstimmung von LLMs für spezifische Softwareentwicklungsaufgaben 2. Praktische Anwendung: i. Feinabstimmung eines LLM für spezifische Code-Optimierungsaufgaben. ii. Aufsetzen und Serverkonfiguration von RNN oder linearisierten LLM für hohe Geschwindigkeit 3. Chain of Thoughts und Graphs of Thoughts im Prompt Engineering 1. Inhalte: i. Einführung in Chain of Thoughts: Konzept und Anwendung. ii. Einführung in Graphs of Thoughts: Strukturierung und Visualisierung von Denkprozessen. iii. Vergleich und Einsatzmöglichkeiten beider Techniken. 2. Praktische Anwendung:

	i. Implementierung von Chain of Thoughts und Graphs of Thoughts zur Lösung komplexer Programmierprobleme. ii. Vergleich zum Few-Shot Prompting 4. Entwicklung und Integration von AI Agents 1. Inhalte: i. Konzepte und Architektur von AI Agents im Software Engineering. ii. Methoden zur Entwicklung autonomer und interaktiver AI Agents. iii. Einsatzmöglichkeiten von AI Agents zur Automatisierung und Optimierung von Entwicklungsaufgaben 2. Praktische Anwendung: i. Entwicklung eines AI Agents zur Unterstützung bei der Code-Analyse und -Optimierung. ii. Zusammenspiel von Tools (Search, Compiler, etc.) mit den LLM in der Softwareentwicklung 5. Softwarearchitektur für KI-gestützte Systeme 1. Inhalte: i. Grundlegende Prinzipien und Muster der Softwarearchitektur. ii. Architekturmuster für die Integration von KI in Softwarelösungen (z.B. Microservices, Event-Driven Architecture). iii. Designprinzipien zur Gewährleistung von Skalierbarkeit, Wartbarkeit und Flexibilität in KI-gestützten Systemen. iv. Best Practices für die Gestaltung robuster und skalierbarer KI-Komponenten. 2. Praktische Anwendung: i. Entwurf und Implementierung einer Softwarearchitektur, die KI-Module effektiv integriert. ii. Aufbau von 1-2 einfachen Microservices, welche den LLM Agenten zur Verfügung stehen 6. Automatisierte Code-Optimierung und Refactoring mit KI 1. Inhalte: i. Methoden zur automatisierten Code-Optimierung. ii. Einsatz von KI zur Verbesserung der Codequalität und Wartbarkeit. iii. Best Practices für automatisiertes Refactoring. 2. Praktische Anwendung:
--	---

	i. Entwicklung eines KI-gestützten Tools zur automatisierten Code-Optimierung und Refactoring. ii. Weiterentwicklung der AI Agents und Toolservices 7. Fortgeschrittene Fehlererkennung und Debugging mit KI 1. Inhalte: i. Techniken zur automatisierten Fehlererkennung und -behebung. ii. Einsatz von AI zur Identifikation und Korrektur komplexer Syntax- und Laufzeitfehler. 2. Praktische Anwendung: i. Implementierung eines fortgeschrittenen Fehlererkennungs- und Debugging-Tools mit KI-Unterstützung. 8. Skalierung und Performance-Optimierung von KI-Modulen 1. Inhalte: i. Methoden zur Evaluierung der Performance von KI-Modulen. ii. Techniken zur Optimierung der Reaktionszeiten und Ressourcennutzung. iii. Skalierung von AI-Lösungen für größere Softwareprojekte. iv. Einfluss von sauberem Code auf die Performance und Skalierbarkeit. 2. Praktische Anwendung: i. Durchführung von Performance-Tests und Implementierung von Optimierungen an eigenen KI-Modulen. ii. Refactoring zur Verbesserung der Performance und Wartbarkeit. 9. Fortgeschrittene Projektplanung und Management für AI-gestützte Softwareprojekt 1. Inhalte: i. Methoden zur Planung und Durchführung komplexer AI-gestützter Softwareprojekte. ii. Einsatz von KI zur Projektplanung, Ressourcenmanagement und Risikoanalyse. 2. Praktische Anwendung: i. Anwendung von AI-Methoden zur Optimierung der Projektplanung und -Durchführung im Abschlussprojekt. 10. Abschlussprojekt: Entwicklung eines fortgeschrittenen AI-gestützten Code-Editor
--	---

	1. Inhalte: i. Planung und Design eines fortgeschrittenen AI-gestützten Code-Editors unter Berücksichtigung der erlernten Techniken und Prinzipien. ii. Implementierung und Integration von fortgeschrittenen AI-Modulen wie Chain of Thoughts, Graphs of Thoughts, AI Agents und automatisierter Code-Optimierung. iii. Testen, Evaluieren und Präsentieren des fertigen Projekts. 2. Praktische Anwendung: i. Entwicklung und Präsentation eines hochfunktionalen AI-gestützten Code-Editors als Gruppenprojekt 11. Präsentation und Evaluation der Abschlussprojekte 1. Inhalte: i. Präsentation der entwickelten AI-gestützten Code-Editoren. ii. Kritische Bewertung der angewandten Methoden und der Einhaltung von Softwarearchitektur-Standards. iii. Feedback und Verbesserungsvorschläge für zukünftige Entwicklungen. 2. Praktische Anwendung: i. Durchführung von Präsentationen und Reviews der Abschlussprojekte.
Lehr- und Lernmethoden	Lehrtechniken – Vorlesungen – Übungen – Gruppenarbeit – Online-Lehre – Workshops Lernaktivitäten – Studium von Literatur – Übungsaufgaben – Bearbeiten von Problemen und deren Lösungsfindung – Zusammenarbeit mit anderen Studierenden Lehrmethode – Präsentation – Einzelarbeit, Partnerarbeit, Gruppenarbeit, – E-Learning

Struktur	42h Online-, Hybrid- oder Präsenzunterricht 78h Selbststudium 120h Total
Literatur	– Modernes Software-Engineering: Entwurf und Entwicklung von Softwareprodukten - Ian Sommerville – Aktuelle wissenschaftliche Artikel und Branchenforschung im Bereich der fortgeschrittenen AI und Software Engineering