

KI-getriebene Veränderungen im Software Engineering – KI4SE

Bericht zum am Fraunhofer IESE in Kaiserlautern durchgeführten Workshop

Andreas Schmietendorf (HWR Berlin, Otto-von-Guericke-Universität Magdeburg)

Motivation

Vielfältige Aufgaben im Software Engineering können vom Einsatz Künstlicher Intelligenz, insbesondere auch von Generativer KI und großen Sprachmodellen (LLMs) profitieren. Im Kern werden durch den KI-Einsatz Steigerungen bezüglich der Entwicklereffizienz, reduzierte Bereitstellungszeiten, aber auch eine Verbesserung der Softwarequalität erwartet. KI-gestützte Tools sollen bei der Code-Generierung, bei der Fehlererkennung und -behebung, aber auch beim automatisierten Softwaretests und der Dokumentation helfen. Zudem unterstützt KI die Wartung und Weiterentwicklung existierender Software durch automatische Quellcodetransformationen, aber auch mit Aussagen zu potenziellen Wartungsaufwänden und möglichen Fehlern in gewachsenen Softwarearchitekturen. In dem hochdynamischen Umfeld sukzessive reiferer KI-Modelle stellt sich zunehmend die Frage, welche Kompetenzen im Diskurs der Softwareentwicklung zukünftig benötigt werden. Aus Sicht der Initiatoren sind es insbesondere kreative und komplexe Aufgaben sowie die umfängliche Qualitätssicherung bereitgestellter KI-Ergebnisse, die des menschlichen Faktors bedürfen. Gerade die Bewertung und Nachvollziehbarkeit einer KI-gestützter Bereitstellung von Software-Artefakten stellt eine massive Herausforderung für die wissenschaftliche aber auch praktische Auseinandersetzung mit den Möglichkeiten und Grenzen der KI-Anwendung dar. Mit Hilfe eines Call for Papers wurde für die Bereitstellung von Beiträgen innerhalb der einschlägigen Community geworben. Zur inhaltlichen Orientierung erfolgte die Bereitstellung der folgenden Themen:

- KI im empirischen Software Engineering
- Fallstudien zum Einsatz von KI-Tools
- KI gestützte Entwicklungsaufgaben
- Mögliche Risiken (z.B. beim Vibe Coding)
- KI-Demokratisierung vs. Vendor „Lock In“
- KI und Low Code
- Test und Bewertung von „KI-Ergebnissen“
- Inhärente Review-Mechanismen mittels KI
- KI im DevOps-Diskurs
- Umgang mit Compliance-Anforderungen
- Veränderte Skill-Bedürfnisse

Überblick zu den Vorträgen

Im Folgenden findet sich eine kurze Charakterisierung der auf dem Workshop gehaltenen Vorträge, keinesfalls kann damit die umfängliche Auseinandersetzung innerhalb der Vorträge wiedergegeben werden. Für weiterführende Informationen sei auf den zum Anfang des Jahres 2026 erscheinenden Tagungsband verweisen.

Session 1:

Frank Elberzhager, Patrick Mennig (Fraunhofer IESE) - Shift-Left im Software-Engineering: Neues Rollenverständnis im SE-Prozess unter Nutzung einer Kontextmanagementplattform

- Die Qualität KI-gestützter Ausgaben wird maßgeblich durch die Qualität der bereitgestellten Eingaben (Prompting) bestimmt.
- Fachliche Spezifikationen und Architekturentscheidungen werden zum kritischen Erfolgsfaktor für KI-gestützte Implementierungen.
- Bedarf einer kontinuierlich lernenden Kontextmanagementplattform zum intelligenten Management des domänenspezifischen Unternehmenswissens.

Elise Wolf, Adam Trendowicz, Julien Siebert (Fraunhofer IESE) - Empirische Untersuchung von Multi-Agenten-Architekturen mit LLMs zur automatisierten Verbesserung von User Stories

- Multi-Agenten-Systeme (MAS) als Ansatz für kollaborative Aufgabenverteilungen zwischen spezialisierten KI-Agenten.
- Implementierung einer MAS-Pipeline zur vergleichenden Bewertung verschiedener LLM-Konfigurationen an realen User Stories.
- Modellhomogenität und inhaltliche Konsistenz sollten zentrale Auswahlkriterien für LLMs in Software-Engineering-Pipelines sein.

Julien Siebert (Fraunhofer IESE) - Causal Discovery in Software Engineering. Report on experiments on bug localization

- Auffinden kausaler Effekte zwischen Quellcodeeigenschaften, aktuell nur gering reflektiert im Diskurs des Software Engineerings.
- Ermittlung von Ursache-Wirkungs-Beziehungen zwischen Variablen mit Hilfe kausaler Graphen.
- Identifikation von (statischen) Metriken des Quellcodes mit einem potentiell kausalen Einfluss auf das Auftreten von Fehlern.

Session 2:

Martin Weiss (Bosch/Uni Magdeburg) - Robustheit von LLMs im Bereich der Quellcodekonvertierung

- Large Language Models (LLMs) als Alternative für eine idiomatische Transformation von Quellcode in eine entsprechende Zielsprache.
- Die praktische Anwendbarkeit erfordert eine umfassende Untersuchung der Robustheit von LLMs.
- Framework-basierte Transformation pertubierter Quellcodefragmente im Zusammenhang mit der Nutzung von 4 verschiedenen LLMs.

Jens Heidrich, Janek Groß, Elena Zentgraf (HS Mainz) - Auf dem Weg zu einem Evaluationsframework für den verlässlichen Einsatz von LLMs in Industrie 4.0

- Industrie 4.0 orientierte Anwendungsfelder von LLMs erfordern Vertrauen in die Zuverlässigkeit, Transparenz und Fairness der eingesetzten KI-Modelle.
- Konzeption und prototypische Umsetzung eines Frameworks, das die Verlässlichkeit generativer KI in industriellen Anwendungsfeldern evaluierbar macht.
- KI-basierte Generierung von Verwaltungsschalen (Asset Administration Shells) als digitale Zwillinge industrieller Assets.

Ben Rymar (HWR Berlin) - Hybride KI-Architekturen als Lösungsansatz für regulierte Domänen - Erkenntnisse aus der Medizintechnik-Branche

- „Gescheiterter“ Implementierungsversuch einer medizinischen Entscheidungslogik (Umgang mit Vorhofflimmer-Patienten) mit Hilfe feingetunter LLMs.
- Einsatz eines konventionellen (determinierten) Algorithmus zur Gewährleistung von Nachvollziehbarkeit und Präzision.
- Bereitstellung einer Lösung im Diskurs einer hybriden Architektur, d.h. ein LLM übernimmt die Aufbereitung der nicht-kritischer Kommunikation.

Session 3:

Violeta Bonet Vila, Adam Trendowicz, Andreas Jedlitschka (Fraunhofer IESE) - Energy-aware Software Engineering: Fine-grained energy consumption prediction of software using AI

- Berücksichtigung des energetischen Fußabdruckes im Diskurs der Aufgaben des Software-Engineerings bzw. insbesondere beim Einsatz von KI.
- Strukturierte Literaturanalyse zu existierenden Arbeiten im Diskurs korrespondierender Energieverbräuche (Softwareartefakte, Modelle, Benchmarks, ...)
- Statische Analyse von Quellcode-Statements (z. B. Nested Loops) zum Monitoring des referenzierten Energieverbrauchs.

Walter H. Letzel, Andreas Schmietendorf (TU Berlin, HWR Berlin) - Herausforderungen einer KI-basierten Transkription rollenbasierter Gesprächsverläufe

- Grundlegendes Verständnis der Transkription („Gesprächsauszeichnung“ – u.a. GAT) in Abgrenzung einer einfachen Übertragung von Sprache zu Text.
- KI-basierte Transkription mit Hilfe verschiedener Toolansätze und einhergehende Leistungsmerkmale.
- Vielfältige Anwendungsszenarien einer KI-basierten Transkription, u.a. im Diskurs der Erarbeitung von User Stories.

Christian Malek, Vasil Tenev, Martin Becker (Fraunhofer IESE) - AI-Storytelling: Vom Expertenwissen zur intelligenten Antwort

- Erfassung von Experten- und Erfahrungswissen zur effizienten und kontextsensitiven Bereitstellung innerhalb komplexer Arbeitssituationen.
- Überführung von Wissen in standardisierte Informationsmodelle, die auf der Industrie-4.0-Verwaltungsschale (digitaler Zwilling) basieren.
- Kombination semantischer Suchmechanismen unter Einsatz von Vektor- und Graph-Datenbanken sowie einer domänenspezifischen Sprache.

Session 4 - Diskussionsrunde:

Sandro Hartenstein (Uni Magdeburg) - Von Risiken zur Robustheit:
Framework-basierte Stärkung von LLM-Integrationen in Web-APIs

- Design Science Research: Risikobasiertes Framework für robuste Integration von LLMs
- Vom Prototyp zur Produktion – Fragen zu möglichen Einsatzszenarien und einhergehenden Herausforderungen
- Robustheit in der Praxis – Bedeutung einer robusten LLM-Integration sowie potentielle Lösungsansätze (u.a. Architektur-Pattern, Test und Monitoring)

Zusammenfassende Ergebnisse

Die folgenden Punkte geben eine subjektive Einschätzung der primär im Workshop diskutierten Themenbereiche wieder. Keinesfalls kann damit eine erschöpfende Charakterisierung der aktuellen Situation des Einsatzes von KI-Modellen für Aufgaben im Software Engineering erreicht werden.

- Sprachmodelle (LLMs) fokussieren sich entsprechend der häufigkeitsorientierten Trainingsdaten auf Aspekte/Themen die wir „gerne“ hören. Aktuell lässt sich ein grundsätzlicher Paradigmenwechsel in Richtung eines agentengetriebener KI-Einsatzes (u.a. Agentic RAG - Retrieval-Augmented Generation) beobachten.
- Weltweit sind viele Unternehmen mit großen Investitionen („to big to fail“) im Diskurs der KI-basierten Erzeugung von Artefakten zur Softwareentwicklung unterwegs. Häufig wird dabei mit einer zunehmenden Demokratisierung der Quellcodeerzeugung argumentiert wird.
- Der Einsatz von KI in der Softwareentwicklung führt zu einem „Shift Left“ der kreativ umzusetzenden Aufgaben im Softwarelebenszyklus. Im Detail fokussieren sich Aktivitäten zunehmend auf Fragen der (formalen) Spezifikation von Anforderungen bzw. Fragen zur fachlich und technisch benötigten Architekturentscheidungen.
- Die KI-basierte Automation innerhalb der Softwareentwicklung steigt mindestens ab der Phase der Implementierung stark an. Typische Werkzeuge finden sich mit GitHub Copilot, Claude Code CLI oder auch Cursor IDE. Daraus resultiert ein massiver Druck auf die Ergebnisqualität der vorhergehenden Entwicklungsphasen (Analyse, Design).

- Der unkritische Einsatz von KI ist dringend zu hinterfragen. Potentielle Probleme beziehen sich u.a. auf eine unzureichenden Nachvollziehbarkeit, die Gefahr der eingeschränkten Erklärbarkeit aber auch einer zunehmenden Abhängigkeit von wenigen (großen) Anbietern.
- KI-gestützte Softwareentwicklungen implizieren die Gefahr einer zunehmend verflachenden und damit auf Dauer unzureichenden Ausbildung. Gerade Berufseinsteiger benötigen den sukzessiven Erfahrungsaufbau im Diskurs komplexer Projektszenarien, was eine KI nur unzureichend reflektiert.
- Sicherheitsaspekte des KI-Einsatzes werden in der Praxis häufig zu gering reflektiert. Herausforderungen existieren z.B. im Umgang mit unerlaubten Funktionen gegenüber den LLMs, die ggf. data leaks erzeugen. Kritisch sind ebenso beim Training ggf. vergiftete KI-Modelle (Poisoning-Attacken).
- Die Evaluierung eingesetzter KI-Modelle mit Hilfe eigener bzw. domänenspezifischer Daten ist zwingend. Durch synthetische Benchmarks erzeugte Leistungsaussagen lassen sich zumeist nicht auf eigene Anwendungsfälle übertragen.

Ausblick und Dank

Auch für das Jahr 2025 planen die Organisatoren einen entsprechenden Workshop zum Thema KI im Software-Engineering. Der korrespondierende Call for Paper wird zeitnah über die Kommunikationskanäle der Gesellschaft für Informatik (speziell GI-Fachgruppe Measurement & Data Science) und der Central Europe Computer Measurement Group kommuniziert.

Ein besonderer Dank gilt allen Referenten für die Bereitstellung und Präsentation eines entsprechenden Beitrags. Dank gilt ebenso dem Fraunhofer IESE (speziell: Dr. Andreas Jedlitschka) für die Rolle als Gastgeber bzw. Jens Heidrich für die informationstechnische Abbildung des Workshops im Rahmen der Gesellschaft für Informatik. Abschließend sei noch der Hinweis auf einen zu Beginn des Jahres 2026 vorgesehenen Tagungsband erlaubt. Dieser wird einen Artikel zu jeweils den präsentierten Beiträgen bereitstellen.

Schmietendorf, A.; Heidrich, J.; Jedlitschka, A. (Hrsg.): Tagungsband zum Workshop KI-getriebene Veränderungen im Software Engineering – KI4SE, in Berliner Schriften zu modernen Integrationsarchitekturen, Shaker-Verlag, Düren, Band 31, ISBN 978-3-8191-0106-9

Anlage - Initiale Umfrage (Mentimeter)

Mit Hilfe einer zu Beginn des Workshops durchgeführten Umfrage (Dauer ca. 10 min.) sollte ein initiales Meinungsbild zum Einsatz von KI-Ansätzen im Software-Engineering erzeugt werden. Die SWOT-orientierte und interaktiv gestalteten Umfrage (Einsatz des Tools Mentimeter) sollte darüber hinaus die Diskussion zwischen Teilnehmern anregen. Zum Abschluss der Umfrage wurde eine im Vorfeld realisierte KI-Analyse im Diskurs der aufgeworfenen Fragen präsentiert.

Fraunhofer IES
Institut für Software Engineering und Knowledge Engineering
Karlsruhe School of Information Technology

Einsatz von KI im Software Engineering - intern fokussierte Stärken (Strengths)?



10 15

Einsatz von KI im Software Engineering - intern fokussierte Schwächen (Weaknesses)?



10 15

Einsatz von KI im Software Engineering - extern fokussierte Chancen (Opportunities)?



Einsatz von KI im Software Engineering - extern fokussierte Risiken (Threats)



SWOT-Analyse: Einsatz von KI im Software Engineering	
STÄRKEN	SCHWÄCHEN
• Automatisierung von Routineaufgaben • Produktivitätssteigerung • Verbesserte Codequalität • Schnellere Entwicklung	• Abhängigkeit von Trainingsdaten • Mangelnde Transparenz / Erklärbarkeit • Begrenzte Kontextverständnis • Kosten und Aufwand der Implementierung
CHANCEN	RISIKEN / BEDROHUNGEN
• Innovation durch neue Werkzeuge • Steigerung der Wettbewerbsfähigkeit • Personalisierte Entwicklung • Verbesserung der Software-Wartung	• Arbeitsplatzunsicherheit • Ethische und rechtliche Fragen • Sicherheitsrisiken • Technologische Abhängigkeit

Quelle des Ergebnisses: ChatGPT - genutzt am 04. November 2025

Prompt: Wie könnte eine SWOT-Analyse zum Einsatz von KI im Software Engineering aussehen?

