

---

Fachausschuss

Jahrgang 32 Heft 1

**Management der  
Anwendungsentwicklung  
und -wartung (WI-MAW)**

ISSN 1610 5753

im FB Wirtschaftsinformatik

Juni 2026

---

## Inhalt

|                               |    |
|-------------------------------|----|
| Fachbeiträge.....             | 3  |
| Ankündigungen .....           | 42 |
| Berichte .....                | 46 |
| Buchhinweise, Rezension ..... | 60 |
| Organisation .....            | 64 |

## Inhaltsverzeichnis

### Fachbeiträge

|                                                                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <i>Erkenntnisse zum Einsatz von GenAI in den frühen Phasen der Softwareentwicklung</i><br>Alexander Rachmann, Hendrik Poschmann, Lucas Weißbeck ..... | 3  |
| <i>KI-Unterstützung im Software Requirements Engineering: Möglichkeiten und Grenzen</i><br>Ben Rymar, Andreas Schmietendorf .....                     | 17 |
| <i>KI in der frühen Softwarekonzeption: Ein praxisnaher Leitfaden für Lösungsskizzen</i><br>Kokulan Thanabalan, Axel Kalenborn .....                  | 30 |

### Ankündigungen

|                                                                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|----|
| PVM 2026 Call for Papers / Aufruf zur Einreichung von Beiträgen: Projekte im Wandel -<br>Disruption, Künstliche Intelligenz, hybride Methoden ..... | 42 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|----|

### Berichte

|                                                                                                                                                   |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tagungsbericht PVM 2025: Post-Agilität, Resilienz, Transformation .....                                                                           | 46 |
| KI-getriebene Veränderungen im Software Engineering – KI4SE 2025: ausführlicher Bericht<br>zum Workshop am Fraunhofer IESE in Kaiserlautern ..... | 53 |

### Buchhinweise, Rezension

|                                                                                                                                                                   |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Buchhinweis: Zusammenfassende Darstellung der Ergebnisse des Forschungsprojekts TAHAI<br>(TrustAdHocAI) .....                                                     | 60 |
| Buchhinweis: Software Engineering prototypische KI-Implementierungen im Zusammenhang<br>mit domänenspezifischen Problemen des Projekts TAHAI (TrustAdHocAI) ..... | 61 |
| Rezension zu: „IT-Management. Das umfassende Handbuch“ .....                                                                                                      | 62 |

### Organisation

|                                                                                                                                                                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Der Fachausschuss „Management der Anwendungsentwicklung und –wartung“ WI-MAW und<br>die Fachgruppen<br>Vorgehensmodelle für die betriebliche Anwendungsentwicklung WI-VM<br>Projektmanagement WI-PM<br>Software Produktmanagement WI-PrdM<br>stellen sich vor ..... | 64 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|



# Erkenntnisse zum Einsatz von GenAI in den frühen Phasen der Softwareentwicklung

Alexander Rachmann, Hendrik Poschmann, Lucas Weißbeck

Hochschule Niederrhein, Fachbereich Wirtschaftsingenieurwesen

<vorname>.<nachname> @hs-niederrhein.de

**Abstract:** **Ziel:** Der Beitrag fasst acht Metastudien zum Einsatz von GenAI, insbesondere großer Sprachmodelle, in frühen Phasen der Softwareentwicklung zusammen. Fokus sind Requirements Engineering, Design/Architektur und Test sowie deren Evaluation und Governance. **Methode:** 48 Forschungsfragen wurden fünf Komplexen zugeordnet: operative Nutzung in RE, Design/Architektur und Test, Evaluation/Evidenz sowie Risiken. Analysiert wurden Modelle, Prompting-Strategien, Artefakte und Evaluationsansätze. **Ergebnisse:** LLMs unterstützen vor allem Analyse und Spezifikation von Anforderungen sowie architektonische Entscheidungsprozesse. GPT-Modelle dominieren. Evaluationen sind meist laborbasiert, wenig standardisiert und selten industriell validiert. **Einordnung:** Das Potenzial ist breit, die empirische Reife jedoch begrenzt. Notwendig sind robustere Evaluationen und bessere Forschungsdesigns.

## 1 Einleitung

Anwendungen der künstlichen Intelligenz verändern rasant, wie Software entwickelt wird. In diesem Beitrag werden verschiedene Metastudien zusammengefasst, die den Stand der Technik erheben. Insbesondere dreht es sich hierbei um generative künstliche Intelligenz (generative artificial intelligence, „GenAI“), also solche künstliche Intelligenz (KI), die selbst Inhalte generiert. Praktisch sind dies derzeit vor allem große Sprachmodelle („Large Language Models“, LLMs), wie ChatGPT von OpenAI, Gemini von Google, Claude von Anthropic oder Grok von xAI. Von besonderem Interesse sind für die Veröffentlichung die frühen Phasen der Softwareentwicklung im Sinne von [AABC23].

## 2 Methodik

### 2.1 Übersicht der Literaturstudien

Es werden acht Studien untersucht. Diese Studien sind in Gesprächen mit Experten erhoben worden und bilden, nach Kenntnis der Autoren, den Stand der Erkenntnisse dar. Da derzeit viel Bewegung in diesem Thema ist, gehen die Autoren davon aus, dass bald neue Erkenntnisse erhoben werden. Es wurden nur Metastudien ausgewählt, die sich auf empirische Arbeiten beziehen und in hoch gerankten Journals und Konferenzen veröffentlicht wurden. In Tabelle 1 sind die acht Studien dargestellt, in welchem Suchraum die Studien gesucht haben und wie viele Quellen darin untersucht wurden. In Tabelle zwei sind die 48 Forschungsfragen („FF“) dargestellt.

| Veröffentlichung       | Suchraum                                                                                                           | Anzahl                  |
|------------------------|--------------------------------------------------------------------------------------------------------------------|-------------------------|
| [CHLR25]               | Literaturreview und Fokusgruppe                                                                                    | 255 Paper / 15 Personen |
| [ELMA25]               | Scopus, ACM Digital Library, IEEE Xplore, Web of Science, Google Scholar, and ArXiv                                | 238                     |
| [HWA25]                | ACM Digital Library, IEEE Xplore Digital Library, Scopus, Web of Science, Google, Google Scholar und Bing          | 36                      |
| [KoPB25]               | IEEE Explore, ACM Digital Library, Wiley, Scopus, Web of Science and Science Direct                                | unklar                  |
| [VaRe25]               | IEEE Xplore und Scopus                                                                                             | 33                      |
| [WATH25]               | IEEE Explore, Science Direct, ACM, Springer, Google Scholar                                                        | 44                      |
| [ZDAZ25]               | IEEE Xplore, ACM Digital Library, Wiley, Scopus, Web of Science und Science Direct; ergänzt durch snowballing      | 186                     |
| 29.05.2026<br>15:18:00 | CORE A*/A/B-bewerteten Konferenzen, Q1/Q2-bewerteten Journals und Workshops aus CORE A*/A/B-bewerteten Konferenzen | 74                      |

Tabelle 1: Übersicht der Literaturstudien

| #  | Forschungsfrage                                                                                                                                                                                                                  |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Wie kann GenAI die Erhebung von Anforderungen unterstützen?                                                                                                                                                                      |
| 2  | Wie kann GenAI effektiv Anforderungsspezifikationen aus hochrangigen Benutzerangaben erstellen?                                                                                                                                  |
| 3  | Wie kann GenAI die automatische Validierung von Anforderungen gegenüber domänenspezifischen Einschränkungen und Vorschriften erleichtern?                                                                                        |
| 4  | Wie kann GenAI genutzt werden, um Änderungsanforderungen vorherzusagen?                                                                                                                                                          |
| 5  | Welche Herausforderungen und Risiken bestehen bei der Einführung von GenAI für RE-Aufgaben in verschiedenen RE-Phasen (Vor-Erhebung, Erhebung, Spezifikation, Analyse)?                                                          |
| 6  | Wie kann GenAI dabei helfen, geeignete Design- und Architekturpatterns basierend auf Anforderungen und Einschränkungen zu identifizieren und auszuwählen?                                                                        |
| 7  | Welche Strategien können angewendet werden, um die Zusammenarbeit zwischen Fachleuten und GenAI im Software-Design-Prozess zu fördern?                                                                                           |
| 8  | Welche Einschränkungen und Risiken sind mit dem Einsatz von GenAI im Software-Design verbunden, und wie können diese gemindert werden, um Zuverlässigkeit und Validität sicherzustellen?                                         |
| 9  | Wie kann GenAI in einem kontinuierlichen Software-Design-Prozess eingesetzt werden, um Verbesserungen zu identifizieren und Refactorings zu automatisieren, um zentrale Qualitätsattribute zu optimieren?                        |
| 10 | Wie kann GenAI genutzt werden, um das Design und die Implementierung von Benutzeroberflächen zu automatisieren und die Benutzererfahrung in Softwareanwendungen zu verbessern?                                                   |
| 11 | Wie kann GenAI Kompromisse zwischen verschiedenen Qualitätsattributen wie Leistung, Skalierbarkeit, Sicherheit und Wartbarkeit optimieren?                                                                                       |
| 12 | Welche Strategien können angewendet werden, um GenAI zu befähigen, Systemarchitekturen über die Zeit anzupassen und weiterzuentwickeln, unter Berücksichtigung sich ändernder Anforderungen, Technologien und Geschäftskontexte? |
| 13 | Was sind die aktuellen Forschungstrends beim Einsatz von GenAI in der Anforderungsanalyse (RE)?                                                                                                                                  |
| 14 | Welche dominanten Ansätze und Techniken der GenAI werden aktuell in der Anforderungsanalyse eingesetzt und welche Qualitätsmerkmale haben sie?                                                                                   |
| 15 | Welche Evaluationsmethoden, Metriken, Werkzeuge und Datensätze unterstützen den Einsatz von LLMs in der Anforderungsanalyse?                                                                                                     |
| 16 | Was sind die Hauptprobleme beim Einsatz von GenAI in der Anforderungsanalyse, und wie spiegeln sie die aktuellen Forschungslücken wider? Welche zukünftigen Forschungsrichtungen können diese Lücken adressieren?                |

|    |                                                                                                                                                                           |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17 | Wie ist der aktuelle Stand und Trend der industriellen Nutzung von LLMs in der Anforderungsanalyse?                                                                       |
| 18 | Für welche Zwecke werden GenAI-Modelle in der Softwarearchitektur eingesetzt?                                                                                             |
| 19 | Welche GenAI-Modelle wurden verwendet?                                                                                                                                    |
| 20 | Wie wurde GenAI angewendet?                                                                                                                                               |
| 21 | In welcher Phase des Softwarearchitektur-Lebenszyklus wird GenAI eingesetzt?                                                                                              |
| 22 | Welche Architekturstile oder -patterns werden angestrebt?                                                                                                                 |
| 23 | Welche Aufgaben der Architekturpflege und qualitätsbezogenen Aktivitäten werden adressiert?                                                                               |
| 24 | Welche Methoden der Architektur-Analyse oder -Modellierung wurden verwendet, um die Ergebnisse der GenAI zu validieren?                                                   |
| 25 | Auf welche Anwendungsfälle wurde GenAI in der Softwarearchitektur angewendet?                                                                                             |
| 26 | Welche zukünftigen Herausforderungen werden für den Einsatz von GenAI in der Softwarearchitektur identifiziert?                                                           |
| 27 | Wie ist die Prompt-Technik aktuell gestaltet, um spezifische Aufgaben im Bereich des erneuerbaren Energiens zu unterstützen?                                              |
| 28 | Was sind die aktuellen Einschränkungen und die zukünftige Ausrichtung von PE4RE im Hinblick auf fortschrittliche PE-Technologien?                                         |
| 29 | Welche technischen Eigenschaften von GenAI schaffen Chancen und Herausforderungen für Architekten in agilen Umgebungen?                                                   |
| 30 | Welche aktuellen Anwendungsfälle von GenAI beeinflussen Architekturpraktiken und Entscheidungsfindung in agilen Organisationen?                                           |
| 31 | Wie wirkt sich GenAI auf traditionelle Rollen, Aufgaben und erforderliche Fähigkeiten von Architekten in agilen digitalen Transformationsprozessen aus?                   |
| 32 | Welche organisatorischen und technologischen Faktoren beeinflussen die Einführung von GenAI in Architekturpraktiken?                                                      |
| 33 | Welche Governance- und Kompetenzanpassungen sind erforderlich, um GenAI in Architekturpraktiken zu integrieren?                                                           |
| 34 | Gibt es GenAI-Modelle, die bei Aktivitäten der Anforderungsanalyse (RE) unterstützen?                                                                                     |
| 35 | Welche Arten von Anforderungsartefakten können diese Modelle verarbeiten?                                                                                                 |
| 36 | Welche RE-Aktivitäten werden durch diese Modelle unterstützt?                                                                                                             |
| 37 | Welche Ansätze verfolgen diese Modelle für RE?                                                                                                                            |
| 38 | Wie viel Evidenz liegt vor, um GenAI-Modelle in RE zu nutzen?                                                                                                             |
| 39 | Welche Einschränkungen bestehen bei aktuellen Anwendungen von GenAI-Modellen in RE?                                                                                       |
| 40 | Welche Eingabekonfigurationen, Formate und Notationen werden in den Anforderungen für anforderungsgetriebenes automatisiertes Softwaretesten verwendet?                   |
| 41 | Welche Frameworks, Werkzeuge, Verarbeitungsmethoden und Transformationstechniken werden in Studien zum anforderungsgetriebenen automatisierten Softwaretesten eingesetzt? |
| 42 | Welche Testformate und Abdeckungskriterien werden im anforderungsgetriebenen automatisierten Softwaretestprozess verwendet?                                               |
| 43 | Wie evaluieren bestehende Studien die generierten Testartefakte im anforderungsgetriebenen automatisierten Softwaretestprozess?                                           |
| 44 | Welche Einschränkungen und Herausforderungen bestehen aktuell bei den bestehenden Methoden des anforderungsgetriebenen automatisierten Softwaretestens?                   |
| 45 | Wie werden große Sprachmodelle im Requirements Engineering benutzt?                                                                                                       |
| 46 | Wie wird Prompt Engineering im LLM-basierten Requirements Engineering benutzt?                                                                                            |
| 47 | Welche Ressourcen werden für große Sprachmodelle benutzt?                                                                                                                 |
| 48 | Wie werden llm-basiertes Requirements Engineering evaluiert?                                                                                                              |

Tabelle 2: Untersuchte Forschungsfragen; Laufende Nummerierung nach alphabetischer Sortierung der Autoren, danach nach Rangfolge der Fragen in der jeweiligen Veröffentlichung. 1-12 [ABAC25]; 13-17 [CHLR25]; 18-26 [ELMA25]; 27 und 28 [HWA25]; 29 -33 [KoPB25]; 34 -39 [VaRe25]; 40-44 [WATH25]; 45-48 [ZDAZ25]

Um diese Menge durchsuchbar zu machen, wurden die Forschungsfragen zu fünf Fragenkomplexen gruppiert (siehe Tabelle 3):

1. Operative Nutzung von GenAI im Requirements Engineering: Diese Kategorie umfasst Forschungsfragen, die den direkten Einsatz von GenAI zur Erzeugung, Transformation oder Unterstützung von Anforderungen untersuchen.
2. Operative Nutzung von GenAI in Design und Architektur: Diese Kategorie umfasst Forschungsfragen, die den direkten Einsatz von GenAI zur Erzeugung, Transformation oder Unterstützung von Architekturartefakten untersuchen.
3. Operative Nutzung von GenAI im Test: Diese Kategorie umfasst Forschungsfragen, die den direkten Einsatz von GenAI zur Erzeugung, Transformation oder Unterstützung von Testartefakten untersuchen.
4. Evaluation / Evidenz: Diese Spalte bündelt Arbeiten, die sich mit Messbarkeit, Vergleichbarkeit und empirischer Absicherung beschäftigen. Die Trennung von operativer Nutzung ist notwendig, da ein Großteil der Studien zwar Anwendungsszenarien beschreibt, jedoch keine belastbare Evaluation liefert. Die explizite Sichtbarmachung dieser Perspektive erlaubt Aussagen über den empirischen Reifegrad der Forschung. Obgleich in diesem Fragenkomplex alle drei frühe Phase vorhanden sind, stellt Tabelle 3 noch eine Unterteilung in die Phasen dar. Dabei wird deutlich, dass die Phasen Design und Architektur und Tests deutlich weniger in der Literatur untersucht wurden als das Requirements Engineering.
5. Risiken, Governance und Grenzen: Diese Kategorie erfasst Forschungsfragen, die sich mit Validität, Zuverlässigkeit, Verantwortlichkeit und organisatorischer Einbettung von GenAI befassen. Sie ist bewusst nicht als Teil der Evaluation modelliert, da viele Risiken nicht quantitativ, sondern strukturell, ethisch oder organisatorisch begründet sind und sich quer über alle SDLC-Phasen erstrecken.

| Phase                           | Operative Nutzung von GenAI                                  | Evaluation und Evidenz | Risiken, Governance und Grenzen |
|---------------------------------|--------------------------------------------------------------|------------------------|---------------------------------|
| <b>Requirements Engineering</b> | 1, 2, 3, 4, 27, 34, 35, 36, 37, 45, 46, 47                   | 13, 14, 15, 38, 48     | 5, 16, 17, 28, 39               |
| <b>Design und Architektur</b>   | 6, 9, 10, 11, 12, 18, 19, 20, 21, 22, 23, 25, 29, 30, 31, 40 | 24                     | 7, 8, 26, 32, 33                |
| <b>Test</b>                     | 41, 42                                                       | 43                     | 44                              |

Tabelle 3: Zuordnung von Forschungsfragen zu Fragenkomplexen

Generative KI (GPT 5.2) wurde als Formulierungshilfe für die Zusammenfassungen der Studieninhalte genutzt.

### 3 Erkenntnisse

Im Folgenden werden die fünf Fragenkomplexe und ihre Antworten aus den untersuchten Studien vorgestellt. Die Ergebnisse werden vereinfacht und skizzenhaft dargestellt; aufgrund der Platzmöglichkeiten in diesem Format und der Sinnhaftigkeit einer Übersichtseitrag ist dies nicht anders möglich. Die Kapitel 3.1 bis 3.5 sind so strukturiert, dass die untersuchten Studien nacheinander, in der Reihenfolge von Tabelle 2, vorgestellt werden. Dies soll dem Leser ermöglichen einfach von einer Erkenntnis zu der erhebenden Studie folgen zu können. Dies hat allerdings auch den Nachteil, dass manche Themen mehrfach aufgegriffen werden: Z.B. wurde vielfach abgefragt, welcher großen Sprachmodelle benutzt werden; hierzu sind sehr unterschiedliche Antworten erhoben worden ([HWA25] und [ZDAZ25] erheben das die GPT-

Modelle am häufigsten genutzt werden, bei [VaRe25] werden BERT-Modelle am häufigsten erhoben).

### 3.1 Operative Nutzung von GenAI im Requirements Engineering

Die Forschungsfragen lassen sich als gemeinsamen Fragenkomplex wie folgt formulieren: „Wie unterstützen große Sprachmodelle das Requirements Engineering entlang der Phasen Elicitation, Spezifikation, Validierung und Änderungsmanagement, und welche Modelle, Artefakte, Methoden und Prompt-Engineering-Ansätze kommen dabei zum Einsatz?“

[ABAC25] erarbeiteten zu FF 1, 2 und 4, dass LLMs zwar Vorabmaterialien wie bestehende Systeme, Dokumente und App-Reviews zur Automatisierung von Domänen- und Stakeholderanalysen nutzen können, aber den Gesamtkontext nicht immer erfassen können. Fehlerhafte Identifikation zentraler Domänenkonzepte oder übersehene Beziehungen führen zu schwachen Domänenmodellen. Zwar unterstützen LLMs die Anforderungsgewinnung durch Analyse von Interviews, Umfragen und Beobachtungen, doch modellinhärente Verzerrungen können Kodierung, Auswertung und Zusammenfassungen verfälschen und damit fehlerhafte oder einseitige Anforderungen in frühen Projektphasen erzeugen. Zur FF 2 erhoben [ABAC25], dass LLMs die automatisierte Erstellung und Klassifikation von Anforderungen ermöglichen, doch die Qualität ohne menschliche Kontrolle unter Unklarheit und mangelnder Kohärenz leidet. Besonders kritisch ist ihr Einsatz bei regulatorischer Konformität oder Priorisierung von Anforderungen, da feine juristische, organisatorische oder kontextuelle Nuancen übersehen werden können. Dadurch besteht das Risiko falscher Gewichtungen, unvollständiger Compliance und Fehlentscheidungen im Anforderungsmanagement. Diese Risiken erfordern klare Prozesse, Validierung durch Experten und Kontrolle. Zur FF 3: Der Einsatz von LLMs in der Anforderungsanalyse, Rückverfolgbarkeit und im Änderungsmanagement kann RE-Prozesse effizienter machen, stößt jedoch bei komplexen Defekten, Abhängigkeiten und Änderungsszenarien an Grenzen. Kontextverlust kann zu inkonsistenten oder widersprüchlichen Anforderungen führen, was zusätzlichen Validierungsaufwand für SE-Teams erzeugt. Zudem bestehen ethische, sicherheitsrelevante sowie Bias-Risiken, insbesondere in sicherheitskritischen Domänen. [ABAC25] beantworteten FF 4 so, dass GenAI Änderungsanforderungen vorhersagen kann, indem es Anforderungen automatisiert analysiert, klassifiziert und ihre Qualität sowie Rückverfolgbarkeit überprüft. Durch die Auswertung von Backlogs, Anforderungen, Domänenkonzepten und Inkonsistenzen lassen sich potenzielle Defekte, fehlende Anforderungen oder Konflikte identifizieren, die spätere Änderungsanträge auslösen. Zudem unterstützen LLMs das Änderungsmanagement, indem sie Auswirkungen komplexer Änderungen abschätzen, auch wenn dabei Kontextverlust und Genauigkeitsgrenzen bestehen.

[HWA25] erarbeiten zu FF 27, dass GPT-Modelle den Markt dominieren, gefolgt von Llama. Die wichtigsten Strategien im Prompt Engineering sind die Kontextualisierung, Personalisierung und schrittweises Reasoning. Andere Strategien spielen eine Nebenrolle. Typisch sind Kombinationen aus Kontextualisierung und Chain-of-Thought, wissensbasierte Prompts als Zusatzkontext sowie meta-kognitive Verfahren, die fast immer auf explizitem Reasoning aufbauen.

[VaRe25] zeigen mit FF 34, dass LLMs die dominierenden Modelle in Requirements Engineering sind. Am häufigsten wird BERT eingesetzt (45 % der untersuchten Studien); GPT-Modelle machen rund 38 % aus und werden für Klassifikation, Analyse und Zusammenfassung genutzt. Weitere Modelle wie BARD, T5 oder Pegasus spielen mit etwa 5 % eine untergeordnete Rolle. In FF 35 und 36 wird erarbeitet, dass LLMs für RE-Aufgaben wie Klassifikation, Extraktion, Analyse und Validierung eingesetzt werden. Als Input dienen Anforderungen aus Dokumenten, User Stories, Use Cases, Verpflichtungsaussagen, Interviewleitfäden sowie aus Produkt- und App-Reviews. Die Modelle unterstützen

grundsätzlich alle zentralen RE-Aktivitäten unterstützen, darunter Anforderungserhebung, Analyse, Spezifikation, Validierung und Management. Am häufigsten werden sie in der Anforderungsanalyse eingesetzt, gefolgt von Spezifikation und Elicitation. Deutlich seltener finden sie Anwendung in der Validierung sowie im Management von Anforderungen. In FF 37 wird erarbeitet, dass Prompt Engineering und Prompt Patterns sowie Retraining und Fine-Tuning vortrainierter, transformerbasierter Modelle benutzt werden. Dabei kommen NLP-Techniken, Zero-Shot Learning und Transfer Learning zum Einsatz, um generative Modelle gezielt an Anforderungen des Requirements Engineering anzupassen und ihre Leistungsfähigkeit für unterschiedliche Aufgaben zu verbessern.

[ZDAZ25] finden in FF 45 eine ungleiche Verteilung der Einsatzfelder und Artefakte. Rund 20 % befassen sich mit Anforderungserhebung und -validierung, 12 % mit Modellierung und 15 % mit allgemeinen SE-Aufgaben. Anforderungsspezifikationen sind mit 38 % das häufigste Eingabematerial. Die restlichen 15% sind verschiedene Artefakte. Als Ergebnisse dominieren Spezifikations- und Analyseberichte, während Modelle, Benutzerprofile, Compliance-Dokumente und Code seltener generiert werden. FF 46 zeigt, dass Zero-Shot Prompting mit 38 % die am häufigsten genutzte Technik ist, gefolgt von K-Shot Prompting mit 26 %. Fortgeschrittene Methoden wie Reasoning, kontextreiche oder templatebasierte Prompts werden deutlich seltener eingesetzt. In 39 % der Studien bleibt die Prompt-Auswahl intransparent. Kriterienbasierte Verfeinerung und Trial-and-Error kommen mit 26 % bzw. 20 % zum Einsatz. In FF 47 wird deutlich, dass sehr wenige öffentlich verfügbare Datensets untersucht werden, diese werden in nur 16% der Studien verwendet. GPT-Modellen dominieren die Studien (77%), gefolgt von Llama (12%). 40% benutzten mehr als ein Modell.

### **3.2 Operative Nutzung von GenAI in Design und Architektur**

Die Forschungsfragen aus der Literatur lassen sich zum folgenden Fragenkomplex zusammenführen: „Welche Einsatzformen, Modelle und Anwendungsfälle von Generative AI existieren im Lebenszyklus, welche architektonischen Entscheidungen, Muster, Qualitätsattribute und Wartungsaktivitäten unterstützen sie, und wie verändern sie Entscheidungsfindung, Arbeitsweise und Rollenverständnis von Softwarearchitekten in agilen Umgebungen?“

[ABAC25] erarbeiten in FF 6, dass GenAI Anforderungen und Randbedingungen analysieren kann, sie mit bekannten Entwurfs- und Architekturmustern abgleichen und plausible Kandidaten vorschlagen. Durch dialogische Rückfragen kann sie fehlende Informationen explizieren, Alternativen vergleichen und Konsequenzen für Qualitätsattribute aufzeigen. Als Entscheidungshilfe unterstützt sie so Architekten, ersetzt aber kein Kontextwissen, da falsche oder unvollständige Eingaben zu irreführenden Empfehlungen führen können, insbesondere in frühen Phasen und bei der Strukturierung komplexer Problemräume im Projektalltag realistisch. In FF 9 erforschen [ABAC25], dass GenAI in einen kontinuierlichen Software-Design-Prozess eingebettet werden kann, indem es Architekturartefakte, Code und Qualitätsmetriken fortlaufend analysiert. Auf Basis von Kontext, Anforderungen und Nicht-funktionalen Zielen identifiziert es Architekturverschleiß, schlägt Refactorings, Muster und Alternativen vor und priorisiert Verbesserungen. Als Co-Designer unterstützt es Reviews, simuliert Auswirkungen auf Qualitätsattribute und lernt aus Feedback. Automatisierte Vorschläge werden versioniert, erklärt und gemeinsam validiert, um Fehlentscheidungen zu vermeiden und Konsistenz zu sichern. Zu FF 10: GenAI kann UI-Design automatisieren, indem sie Nutzeranforderungen, Nutzungsdaten und Designrichtlinien analysiert. Sie erzeugt Layouts, Komponenten und Designsysteme, passt Farben, Typografie und Interaktionen an Zielgruppen an und generiert Prototypen. Durch kontinuierliches Nutzerfeedback optimiert sie Usability, Barrierefreiheit und Konsistenz, testet Varianten automatisch und integriert Designs direkt in Frontend-Code, wodurch Entwicklungszeit sinkt und das Nutzererlebnis messbar verbessert wird, sowie kontextsensitive Personalisierung und adaptive Interfaces über Gerätegrenzen hinweg. Zu FF

11: GenAI kann Zielkonflikte zwischen Qualitätsattributen optimieren, indem sie Systemkontext, Anforderungen und Randbedingungen analysiert und deren Wechselwirkungen modelliert. Sie simuliert Entwurfsalternativen, prognostiziert Auswirkungen auf Performance, Skalierbarkeit, Sicherheit und Wartbarkeit und macht Konflikte früh sichtbar. Auf Basis von Erfahrungswissen, Mustern und Metriken schlägt sie ausgewogene Architekturentscheidungen vor, erläutert Konsequenzen transparent und unterstützt eine kontinuierliche Neubewertung bei sich ändernden Anforderungen oder Betriebsbedingungen. Zu FF 12: Strategien zur Anpassung von Systemarchitekturen durch GenAI umfassen kontinuierliches Monitoring von Anforderungen, Laufzeitmetriken und Geschäftskennzahlen. GenAI analysiert Architekturentscheidungen historisch, erkennt Trends und technischen Schuldenaufbau. Durch Szenariosimulation, Technologie-Scouting und Feedbackschleifen mit Architekten schlägt sie inkrementelle Evolutionen vor. Versionierte Entscheidungsmodelle, erklärbare Empfehlungen und Governance-Regeln sichern Nachvollziehbarkeit, während automatisierte Refactorings und Architekturtests die Anpassung an neue Kontexte unterstützen sowie resiliente Migrationen und langfristige strategische Ausrichtung des Unternehmens zusätzlich.

[ELMA25] erforschen in FF 18, dass LLMs hauptsächlich für die architektonische Entscheidungsunterstützung (38%) und das Reverse Engineering (21%) eingesetzt werden, weniger jedoch für Aufgaben wie Migration, Re-Engineering und Rückverfolgbarkeit. In FF 19 wird erhoben, dass GPT-Modelle von OpenAI dominieren (62 %) die Forschungslandschaft dominieren, während Alternativen wie Google LLMs und Llama-Modelle deutlich seltener eingesetzt werden. Die häufigste Technik ist das Few-Shot-Prompting (31%), RAG (22%) wird häufig zur Modellverbesserung eingesetzt, und 85% der Studien beinhalten menschliche Interaktion, was die unterstützende und nicht die autonome Rolle des LLM unterstreicht (FF 20). LLMs werden am häufigsten bei den Übergängen Anforderung-zu-Architektur (40%) und Architektur-zu-Code (32%) angewendet, während der Übergang Architektur-zu-Architektur (3%) am wenigsten erforscht ist (FF 21). In FF 22 wird erkannt, dass in 68% der untersuchten Studieng Architekturdetails nicht benannt werden. In den Fällen, in denen die Architektur dokumentiert wurde, werden hauptsächlich monolithische Architekturen benutzt werden. 38 % der Studien nutzen LLMs zur Erkennung von Anti-Pattern, zum Refactoring und zur Architekturrekonstruktion. Nur wenige integrieren externe Tools, was darauf hindeutet, dass LLMs traditionelle Wiederherstellungsmethoden ersetzen (FF 23). LLMs werden am häufigsten für Architektur Anforderungen und -snippets (15 %) eingesetzt, insbesondere in Unternehmenssoftware und IoT-Systemen (8 %). Groß angelegte, mobile und eingebettete Systeme werden hingegen weniger erforscht. Java (13 %) ist die am häufigsten verwendete Sprache in LLM-basierten Architekturstudien, jedoch geben 58 % der Studien keine Programmiersprache an, was eine Lücke in der Dokumentation von Implementierungsdetails verdeutlicht (FF 25).

[KoPB25] untersuchen FF 29, 30 und 31. GenAI beschleunigt die Softwareentwicklung durch automatisierte Code- und Dokumentationserstellung sowie optimierte NLP-basierte Anforderungsanalysen. Besonders in frühen agilen Phasen unterstützt sie die Ideenfindung und Abwägung von Qualitätsmerkmalen wie Skalierbarkeit. Die Rolle des Architekten wandelt sich dabei vom Ersteller zum Kurator und Validierer, wobei neue Kompetenzen wie Prompt Engineering und Modell-Governance entscheidend werden. Trotz dieser Vorteile bleiben Risiken wie mangelnde Transparenz („Black Box“), begrenztes Kontextfenster und die Gefahr qualitativ minderwertiger Ergebnisse bestehen. Eine zu starke Abhängigkeit kann zudem die menschliche Urteilskraft schwächen. Aufgrund des probabilistischen Charakters der Modelle ist eine kontinuierliche menschliche Kontrolle sowie die Validierung von Vorschlägen unerlässlich. Um regulatorische Hürden bei Datenschutz und Sicherheit zu meistern, müssen Organisationen zudem klare Richtlinien für einen verantwortungsvollen KI-Einsatz etablieren.

[WATH25] fanden in FF 40 zwei Anforderungstypen (funktionale, nicht-funktionale Anforderungen); vier Arten von Anforderungsformaten (textuell, modellbasiert, formal,

tabellarische/Matrix-Spezifikation); fünf Arten von Anforderungsnotation (natürliche Sprache, kontrollierte natürliche Sprache, einheitliche Modellierungssprache, Softwarekostenreduktion, domänenspezifische Sprache); fünf Anforderungsabstraktion (Produkt-, Merkmals-, Funktions-, Komponentenebene).

### **3.3 Operative Nutzung im Test**

Die Forschungsfragen der Studien lassen sich zum Fragenkomplex zusammenführen: Wie wird Generative AI in anforderungsgetriebenen Prozessen zur automatisierten UI-Generierung und zum automatisierten Softwaretesten eingesetzt, und welche Anforderungen, Formate, Werkzeuge, Methoden, Testartefakte und Abdeckungskriterien werden dabei verwendet?

Für FF 41 wird von [WATH25] ein Framework zur Transformation von Anforderungen in Testartefakte beschrieben. Es umfasst das Transformationsverfahren als mehrstufigen Prozess, verschiedene Transformationstechniken sowie Zwischenrepräsentationen. Ergänzend werden zusätzliche Eingaben wie Dokumente oder Quellcode berücksichtigt. Abschließend wird die Framework-Struktur erläutert, die den Gesamtprozess logisch organisiert, etwa sequenziell, parallel oder iterativ, um eine systematische und nachvollziehbare Testfallerzeugung zu ermöglichen. Für FF 42 wurden sechs Testformate, vier Testphasen von Planung bis Bericht, zehn unterschiedliche Testartefakte sowie neun Abdeckungsmethoden identifiziert. Diese Kategorisierung bietet einen strukturierten Überblick über Testdarstellungen, Prozessschritte, Ergebnisse und Strategien zur Sicherstellung ausreichender Testabdeckung und unterstützt eine methodische Planung, Durchführung und Bewertung von Tests in Softwareprojekten.

### **3.4 Evaluation und Evidenz**

Die Forschungsfragen zur Evaluation und zur Evidenz bilden sich im Fragenkomplex ab: „Wie wird GenAI in den frühen Phasen der Softwareentwicklung empirisch evaluiert, welche Methoden, Metriken, Werkzeuge und Datensätze verwendet, und welches Maß an belastbarer Evidenz liegt für deren Qualität und Einsatzreife vor?“

[CHLR25] untersuchen in FF 13 ihre Stichprobe darauf hin, dass die aktuelle Forschung vorwiegend analytische Aufgaben betont, möglicherweise weil GenAI-Techniken (z. B. LLMs für Text Mining und Clustering) gut für die Verarbeitung und Analyse großer Mengen anforderungsbezogener Daten geeignet sind. Ebenso profitieren die Anforderungserhebung und -spezifikation von den Fähigkeiten von GenAI zum Verstehen und Generieren natürlicher Sprache. Der begrenzte Fokus auf das Management könnte hingegen den Mangel an ausgereiften KI-gestützten Werkzeugen für die projektweite Anforderungsverfolgung und -entwicklung widerspiegeln, die häufig komplexe soziotechnische Faktoren anstelle rein linguistischer Aspekte beinhalten. In FF 14 wird erhoben, dass GPT-Modelle dominieren, während Open-Source-Alternativen kaum genutzt werden. Die Qualitätsbewertung fokussiert operative Leistung wie funktionale Eignung und Zuverlässigkeit, vernachlässigt jedoch Sicherheit, Wartbarkeit und Interpretierbarkeit. Meist werden vortrainierte Modelle ohne Feintuning genutzt. Prompt Engineering ist überwiegend instruktions- und beispielbasiert; interaktive und reasoning-fördernde Methoden bleiben selten, wodurch Tiefe, Anpassungsfähigkeit und langfristige Robustheit begrenzt sind sowie eingeschränkte methodische Reife und geringe Domänenspezialisierung. Zu FF 15: Trotz zunehmender Tool- und Datensatzentwicklung bleibt deren Offenlegung begrenzt, was Reproduzierbarkeit und Benchmarking erschwert. Bewertungsmetriken fokussieren meist oberflächliche Kennzahlen, während domänenspezifische Maße, triangulierte menschliche Bewertungen und Validierung fehlen. Zudem sind Evaluationsmethoden fragmentiert: Fallstudien und Experimente dominieren, während Mixed-Methods-, Längsschnitt- und nutzerzentrierte Designs selten sind. Dies zeigt eine methodische Unreife und den Bedarf an strengeren, transparenteren Bewertungsrahmen.

[ELMA25], FF 24, bemängeln zu geringe Standardisierung: Zwar ist UML mit 17 % die am häufigsten genutzte Modellierungssprache, doch 74 % der Arbeiten verzichten auf formale Architekturmodelle. Zudem nutzen 87 % keine modellgetriebene Entwicklung. Besonders kritisch ist die Validierung, da 93 % der Studien keine Strategie zur Überprüfung KI-generierter Ergebnisse vorweisen, während etablierte Methoden wie Architecture tradeoff analysis method oder Software architecture analysis method kaum Erwähnung finden.

[VaRe25] zeigen in FF 38, dass etwa 39% der untersuchten Literatur auf akademischen Experimenten und empirischen Vergleichen basieren. Weitere 30% nutzen Demonstrationen durch Prompt-Engineering, während 18% ihre Erkenntnisse aus industriellen Fallstudien beziehen. Lediglich 9% stützen sich auf Expertenmeinungen. Bis auf zwei systematische Literaturübersichten wiesen nahezu alle analysierten Arbeiten eine spezifische wissenschaftliche oder praktische Beweisgrundlage auf.

[WATH25], FF 43, erkennen zwei Evaluierungstypen (Konzeptionelle Falldemonstration, Datensatzbewertung) und vier Methoden zu Erhebung der Benutzerbarkeit (Methodologieerklärung, Fallbeispiel, Experiment, Diskussion).

In der Stichprobe von [ZDAZ25], FF 48, dominieren quantitative Evaluationsformen (42%). 16% wird qualitativ evaluiert und 42% gemischt. Laborexperimente dominieren (76%); Urteilsstudien (14%), Feldstudien (7%) und Experimente (4%) werden selten genutzt<sup>1</sup>. Die Anforderungserhebung und allgemeine Software Engineering-Aufgaben sind über mehrere Strategien geprüft.

### **3.5 Risiken, Governance und Grenzen**

Die Forschungsfragen der Studien lassen sich wie folgt kombinieren: „Welche Limitationen und Bedrohungen weist der derzeitige Einsatz von Generative AI in RE-, Design-, Architektur- und Testaktivitäten auf, wie beeinflussen Organisation, Governance und Mensch-KI-Zusammenarbeit deren Nutzung, und welche zukünftigen Forschungs- und Praxisstrategien sind notwendig, um diese Defizite zu adressieren?“

[ABAC25], FF 5, differenzieren Risiken nach Phasen im RE: In der Pre-Elicitation können LLMs den Gesamtkontext verfehlen und Domänenkonzepte falsch modellieren. In der Elicitation führen Modell-Bias zu verzerrten Anforderungen. In der Specification entstehen ohne menschliche Kontrolle unklare, inkohärente oder rechtlich ungenaue Anforderungen. In der Analysis haben GenAI-Modelle Schwierigkeiten mit komplexer Fehlererkennung, Rückverfolgbarkeit und Änderungsszenarien, was Analyse-Lücken begünstigt.

[CHLR25], FF 16 und 17, sehen Reproduzierbarkeit, Halluzinationskontrolle und Interpretierbarkeit als Risiken. Kontrollierbarkeit und Kosten bilden eine zweite Umsetzungsebene. Governance-Themen wie Bias und Ethik sind unteradressiert. Die Stichprobe zeigt eine deutliche Lücke zwischen akademischer Innovation und praktischer Nutzung von GenAI im Requirements Engineering. Über 90 % der Studien verbleiben auf Konzept- oder Prototypniveau. Zentrale Grenzen betreffen fragile Eingaben, methodische Defizite in der Evaluation sowie hohe Implementierungskomplexität. Diese systemischen Herausforderungen deuten auf einen noch geringen operativen Reifegrad hin, trotz hoher Innovationskraft und wachsender konzeptioneller Fortschritte im Forschungsfeld.

[HWA25], FF 28, zeigen mehrere Defizite: Es fehlen nicht-textuelle bzw. multimodale Prompting-Ansätze, die eigentliche Anforderungserhebung wird kaum behandelt, und systematische Evaluationen sind nicht vorhanden. Die vorgeschlagene Roadmap adressiert diese Lücken durch multimodale Datenerhebung mit Konsistenzprüfung, standardisierte

---

<sup>1</sup> Originalzahlen aus der Studie: Vermutlich durch einen Rundungsfehler ergibt die Summe Beträge 101.

Prompt-Patterns für Anforderungen, Ende-zu-Ende-Verfolgbarkeit mit Impact-Analyse sowie den Aufbau einer Benchmark-Suite.

[VaRe25], FF 39: Obwohl vielfältige GenAI-Werkzeuge existieren, konzentriert sich die Forschung bisher fast ausschließlich auf LLMs. Diese Einseitigkeit vernachlässigt spezialisierte Modelle, die RE-Herausforderungen potenziell besser bewältigen könnten. Zudem bergen LLMs Risiken durch ihre Abhängigkeit von hochwertigen Trainingsdaten, da minderwertige Daten zu verzerrten Ergebnissen führen. Die mangelnde Transparenz dieser „Black Box“-Systeme erschwert zudem die Nachvollziehbarkeit und Verantwortlichkeit im Entwicklungsprozess.

[ABAC25], FF 7 und 8 schlussfolgern, dass GenAI eine Entscheidungsunterstützung bieten kann und nicht als Entscheidungsträger genutzt werden sollte. Strukturierte Interaktionsformen, etwa Prompt-Vorlagen, gemeinsame Design-Reviews mit expliziter menschlicher Validierung und nachvollziehbarer Dokumentation der Begründungen, erleichtern die Integration. Eine Einbettung in bestehende Arbeitsprozesse unterstützt die kollektive Nutzung, während klar definierte Verantwortlichkeiten eine unkritische Übernahme fehlerhafter Vorschläge vermeiden. GenAI im Softwareentwurf ist durch begrenztes Kontextverständnis, schwache Berücksichtigung nichtfunktionaler Anforderungen und die Neigung zu plausibel klingenden, aber falschen Empfehlungen eingeschränkt. Risiken sind Fehlentscheidungen, Verfestigung schlechter Entwürfe und unangemessene Standardisierung. Zur Minderung sind eine klare Rollendefinition, sorgfältige Eingaben, systematische menschliche Reviews sowie eine Validierung anhand projektspezifischer Randbedingungen und empirischer Erfahrungen erforderlich.

Laut [ELMA25], FF 26, sind die größten Hürden beim Einsatz von LLMs deren mangelnde Genauigkeit (16 %), Halluzinationen (9 %) sowie ethische Bedenken (7 %). Ergänzt wird dies durch kritische Defizite bei Evaluierungsmetriken, der Kontextsensitivität und der Erklärbarkeit. Zudem fehlen systematische Validierungsmethoden, Verallgemeinerbarkeit und formale Verifizierungen. Auch ungeklärte Fragen zum geistigen Eigentum stellen eine erhebliche Barriere für den zuverlässigen Einsatz in professionellen Architekturprozessen dar.

[KoPB25], FF 32 und 33: Der Erfolg von GenAI hängt maßgeblich von gezielter Weiterbildung und einer offenen Organisationskultur ab. Ohne Investitionen in Kompetenzen wie Prompt-Engineering und KI-Ethik drohen Produktivitätsverluste sowie Verhaltensrisiken wie „Social Loafing“, bei dem sich Mitarbeiter zu stark auf die KI verlassen. Zudem sind eine hohe Governance-Reife mit klaren Richtlinien sowie die organisatorische Bereitschaft – inklusive technischer Infrastruktur und rechtlicher Konformität – essenziell. Ein Reifegradmodell kann dabei helfen, den Übergang von initialen Versuchen zu kontinuierlich optimierten, KI-gestützten Workflows systematisch zu steuern. Um das Potenzial von GenAI sicher zu nutzen, sollten Unternehmen dynamische Governance-Frameworks einführen, die sich kontinuierlich an technologische und regulatorische Änderungen anpassen. Ein iterativer Governance-Lebenszyklus integriert dabei Prinzipien wie Fairness und Transparenz. Durch gezielte Schulungen, klare Richtlinien für sensible Daten und Audit-Trails wird die Verantwortlichkeit gestärkt. Vertrauensbildende Maßnahmen und eine ausgeprägte Data-Governance-Kultur minimieren das Risiko von Fehlbedienungen und fördern eine ethisch verantwortungsvolle Einführung in der Softwarearchitektur.

[WATH25], FF 44: Bestehende Methoden kämpfen mit erheblichen Defiziten bei der Skalierbarkeit und Generalisierbarkeit, besonders bei komplexen Großsystemen. Die Automatisierung bleibt oft lückenhaft, da manuelle Eingriffe beim Parsing oder der Testimplementierung nötig sind. Zudem mindern eine unvollständige Anforderungsabdeckung sowie die starke Abhängigkeit von hochwertigen Eingaben die Robustheit. Hoher Zeitaufwand, methodologische Komplexität und ein Mangel an industrieller Validierung erschweren zusätzlich die praktische Anwendung dieser Frameworks.

## 4 Einordnung

Es wurden acht Veröffentlichungen untersucht. Dabei zeigt sich, dass die gewonnenen Erkenntnisse heterogen und in Details ggf. widersprüchlich sind. Eine konsistente und umfassende Zusammenführung ist daher nicht möglich. Wir fokussieren uns im Folgenden auf drei Themen der erhobenen Datenbasis und kommentieren sie aus unserem derzeitigen Kenntnisstand. Wir greifen diese Themen auf, weil sie aus unserer Sicht eine hohe Relevanz haben.

### 4.1 Potentielle breite Anwendbarkeit, aber wenig rigorose Erkenntnisse

In den Ergebnissen wird deutlich, dass in vielen Tätigkeiten in den frühen Phasen Potential für GenAI liegt; im Grunde wurde jede einzelne Aktivität benannt, die es gibt. Ob dieses Potential genutzt werden kann, scheint noch unklar zu sein, da die meisten Studien schlechte Forschungsdesigns aufweisen, nicht reproduzierbar sind oder nur auf Prototypniveau stattfinden. Einerseits sollten Forscher Wert auf Reproduzierbarkeit legen; andererseits scheint es für Softwareingenieure, die sich traditionell auf deterministische Systeme konzentrieren, schwierig zu sein, Systeme mit einem bewusst gesetzten pseudo-nicht-deterministischen Charakter reproduzierbar zu untersuchen.

### 4.2 Bestandteil einer Schichtenarchitektur oder Werkzeug?

In der Fachcommunity wird der Einsatz von GenAI mittlerweile als weitere Schicht, nach Hochsprachen und Frameworks, gehandelt (vgl. [Arvi25], Abb.1, auch [ArSa25] in etwas anderer Darstellung). Ob sich dies so entwickelt, ist aus Sicht der Autoren noch nicht klar; ebenso wahrscheinlich ist die Etablierung von GenAI als weiteres Werkzeug neben Code-Vervollständigung, Debugger, etc.

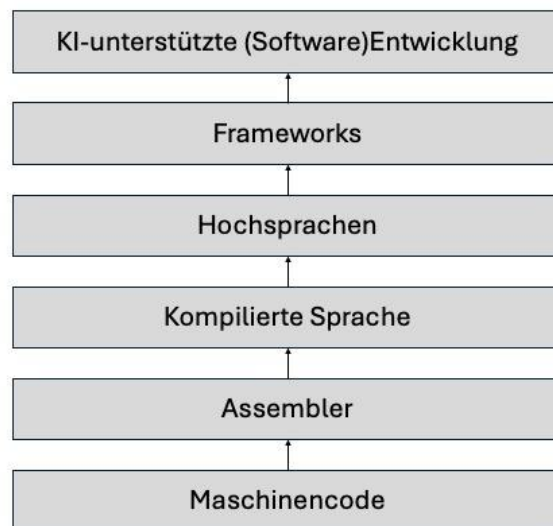


Abbildung 1: Schichtenarchitektur, nach [Arvi25]

Die Sichtweise auf GenAI als Werkzeug wird i.d.R. mit Human-in-the-Loop-Modellen betrachtet; [Moll24] machte frühzeitig nach Start des aktuellen GenAI-Booms darauf aufmerksam; [LSTL25] stellen eine Kollaboration zwischen Mensch und KI in einem Prozessmodell dar (MADE with AI, siehe Abbildung 2.). Z.B. [AMAF26] stellen schon eine detaillierte (prognostizierte) Tätigkeitsliste vor, die von KI in der Softwareentwicklung übernommen werden.

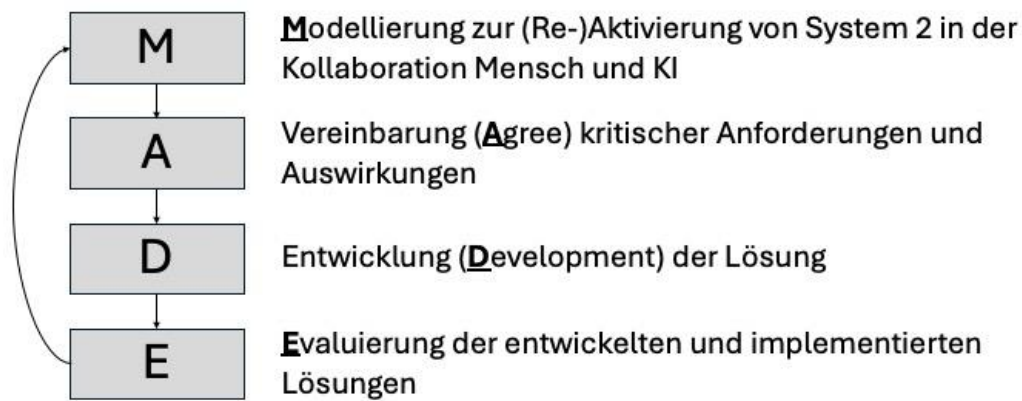


Abbildung 2: KI als Werkzeug in einem iterativen Kollaborationsprozess „MADE with AI“, [LSTL25]

### 4.3 Dominanz von OpenAI und ChatGPT

Die Mehrzahl der untersuchten Studien erhebt, dass OpenAI mit den GPT-Modellen die Nutzung dominiert. Das kann einerseits ein methodisches Problem für die Forschung sein, andererseits muss sich noch herausstellen, ob OpenAI diese Vormachtstellung halten kann. Empirische Befunde wie z.B. die Downloadzahlen auf Huggingface sprechen für eine starke Nutzung von Modellen außerhalb der GPT-Serie (<https://huggingface.co/models?sort=downloads>; 1. Platz: all-MiniLM-L6-v2; 2. Platz bert-base-uncased; 3. Platz nsfw\_image\_detection; erstes open-ai-Modell auf Platz 11); Forumteilnehmer auf Reddit für ChatGPT gibt es aber fünfmal so viel wie für Claude Code (1.800.000 wöchentliche Besucher auf r/ChatGPT vs. 348.000 auf r/ClaudeCode). Die tatsächlichen Nutzungszahlen (z.B. verarbeitete Token) der SaaS-betriebenen Modelle werden nicht veröffentlicht. Insgesamt bleibt es also schwierig zu erkennen, welches Modell oder Modellserie wie oft benutzt wird oder was sinnvolle Metriken hierfür sind.

Einerseits gibt es eine große Bandbreite an Benchmarks, die die Leistung von großen Sprachmodellen vergleichen, siehe [NCLC25]. Aber schon in klein angelegten Markenblindtests zeigt sich die Vermutung, dass Benutzer die Ergebnisqualität von verschiedenen großen Sprachmodellen gar nicht unterscheiden können ([Berl26, Serh26]). Sollten sich diese Vermutung bewahrheiten, wäre dies ebenfalls einen Grund zu überdenken, welche Modelle wie untersucht werden.

## 5 Fazit

GenAI und große Sprachmodelle sind jetzt schon fester Bestandteil der Softwareentwicklung; wie breit sie sich etablieren bleibt noch offen. Es scheint aber fraglos zu sein, dass die Forschung insbesondere in der Evaluation des Entwicklungsprozesses und in der Testphase vermehrt stattfinden muss und methodisch besser werden muss.

## Literaturverzeichnis

- [AABC23] ALPAR, PAUL ; ALT, RAINER ; BENSBERG, FRANK ; CZARNECKI, CHRISTIAN: Phasenmodelle in der Softwareentwicklung. In: *Anwendungsorientierte Wirtschaftsinformatik: Strategische Planung, Entwicklung und Nutzung von Informationssystemen*. Wiesbaden : Springer Fachmedien Wiesbaden, 2023 — ISBN 978-3-658-40352-2, S. 389–448

- [ABAC25] ANH NGUYEN-DUC ; BEATRIZ CABRERO-DANIEL ; ADAM PRZYBYLEK ; CHETAN ARORA ; DRON KHANNA ; TOMAS HERDA ; USMAN RAFIQ ; JORGE MELEGATI ; U. A.: Generative Artificial Intelligence for Software Engineering—A Research Agenda. In: *Software: Practice and Experience* Bd. 55 (2025), Nr. 11
- [AMAF26] AMALFITANO, DOMENICO ; METZGER, ANDREAS ; AUTILI, MARCO ; FULCINI, TOMMASO ; HEY, TOBIAS ; KEIM, JAN ; PELLICCIONE, PATRIZIO ; SCOTTI, VINCENZO ; U. A.: A Research Roadmap for Augmenting Software Engineering Processes and Software Products with Generative AI. In: *ACM Trans. Softw. Eng. Methodol.* (2026). — Just Accepted
- [ArSa25] ARVIND NARAYANAN ; SAYASH KAPOOR: *AI as Normal Technology*. URL <http://knightcolumbia.org/content/ai-as-normal-technology>. - abgerufen am 2026-02-12. — Knight First Amendment Institute
- [Arvi25] ARVIND NARAYANAN: *Today's conversations ...* URL <https://t.co/QGrC2SiTbY>. - abgerufen am 2026-01-28. — Twitter
- [Berl26] BERLOZNIK, FLORIAN: Markenblindtest von Large Language Models (2026)
- [CHLR25] CHENG, HAOWEI ; HUSEN, JATI H. ; LU, YIJUN ; RACHARAK, TEERADAJ ; YOSHIOKA, NOBUKAZU ; UBAYASHI, NAOYASU ; WASHIZAKI, HIRONORI: Generative AI for Requirements Engineering: A Systematic Literature Review. In: *Software: Practice and Experience* (2025), S. spe.70029
- [ELMA25] ESPOSITO, MATTEO ; LI, XIAOZHOU ; MORESCHINI, SERGIO ; AHMAD, NOMAN ; CERNY, TOMAS ; VAIDHYANATHAN, KARTHIK ; LENARDUZZI, VALENTINA ; TAIBI, DAVIDE: Generative AI for Software Architecture. Applications, Challenges, and Future Directions, arXiv (2025)
- [HWA25] HUANG, KAICHENG ; WANG, FANYU ; HUANG, YUTAN ; ARORA, CHETAN: Prompt Engineering for Requirements Engineering: A Literature Review and Roadmap. In: *2025 IEEE 33rd International Requirements Engineering Conference Workshops (REW)*, 2025, S. 548–557
- [KoPB25] KOORY, STEFAN JULIAN ; PIEST, JEAN PAUL SEBASTIAN ; BEMTHUIS, ROB HENK: Impact and Implications of Generative AI for Enterprise Architects in Agile Environments: A Systematic Literature Review, arXiv (2025)
- [LSTL25] LUKYANENKO, ROMAN ; SAMUEL, BINNY M. ; TEGARDEN, DAVID P. ; LARSEN, KAI R. ; JABBARI, ARAZ ; AZIZ, RASHA ABDEL ; ALMEIDA, JOÃO PAULO ; AMROLLAHI, ALIREZA ; U. A.: Software Development and Modeling in the Age of Artificial Intelligence Manifesto. Rochester, NY, Social Science Research Network (2025)
- [Moll24] MOLLIK, ETHAN: *Co-Intelligence: The Definitive, Bestselling Guide to Living and Working with AI*. London : WH Allen, 2024 — ISBN 978-0-7535-6077-8
- [NCLC25] NI, SHIWEN ; CHEN, GUHONG ; LI, SHUAIMIN ; CHEN, XUANANG ; LI, SIYI ; WANG, BINGLI ; WANG, QIYAO ; WANG, XINGJIAN ; U. A.: A Survey on Large Language Model Benchmarks, arXiv (2025). — arXiv:2508.15361 [cs]
- [Serh26] SERHAN, GIHAN: Markenblindtests bei Large Language Models. Eine empirische Untersuchung zur Wahrnehmung, Präferenz und Erkennbarkeit von LLMs, Unveröffentlichte Hausarbeit an der Hochschule Niederrhein (2026)
- [VaRe25] VASUDEVAN, POONKUZHALI ; REDDIVARI, SANDEEP: The Role of Generative AI Models in Requirements Engineering: A Systematic Literature Review. In: *Proceedings of the 2025 ACM Southeast Conference, ACMSE 2025*. New York, NY, USA : Association for Computing Machinery, 2025 — ISBN 979-8-4007-1277-7, S. 188–194
- [WATH25] WANG, FANYU ; ARORA, CHETAN ; TANTITHAMTHAVORN, CHAKKRIT ; HUANG, KAICHENG ; ALETI, ALDEIDA: Requirements-Driven Automated Software Testing: A Systematic Review. In: *ACM Trans. Softw. Eng. Methodol.* (2025). — Just Accepted

- [ZDAZ25] ZADENOORI, MOHAMMAD AMIN ; DĄBROWSKI, JACEK ; ALHOSHAN, WAAD ; ZHAO, LIPING ; FERRARI, ALESSIO: Large Language Models (LLMs) for Requirements Engineering (RE): A Systematic Literature Review, arXiv (2025). — arXiv:2509.11446 [cs]

# KI-Unterstützung im Software Requirements Engineering: Möglichkeiten und Grenzen

Ben Rymar <sup>1</sup>, Andreas Schmietendorf <sup>1,2</sup>

<sup>1</sup> HWR Berlin / <sup>2</sup> Otto-von-Guericke-Universität Magdeburg

## 1 Einleitung

Ansätze der künstlichen Intelligenz (kurz KI) können vielfältige Aufgaben des Software Engineerings unterstützen, was auch für das Requirements Engineering gilt. Das Requirements Engineering (RE) umfasst die Ermittlung, Dokumentation und Prüfung von Anforderungen und bildet eine kritische Phase im Software-Lebenszyklus, um sicherzustellen, dass ein System die Bedürfnisse der Stakeholder erfüllt. Aktuelle LLMs wie z.B. OpenAIs GPT-4, Anthropic Claude, Googles Gemini, Metas LLaMA (Modelle Scout und Maverick) oder DeepSeeks-Familie bieten die Chance, RE-Prozesse effizienter zu gestalten, etwa durch automatische Anforderungsermittlung, Konsistenzprüfungen oder Generierung von Entwurfsskizzen, was die Entwicklungszeit verkürzen und menschliche Fehler reduzieren kann (Hemmat et al., 2025, S. 1–2, 8-9). Das Training aktueller KI-Modelle basiert im Schwerpunkt auf im Internet öffentlich zugängliche Quellen, woraus sich Kontext- und Domänenprobleme ergeben können: Typisch sind Schwierigkeiten im Umgang mit fachspezifischen Terminologien oder komplexen domänenspezifischen Abhängigkeiten woraus ggf. fehlerhafte oder auch halluzinierte Anforderungen resultieren. Ebenso ist die Formalisierung der von LLM generierten Anforderungen eine offene Frage - d.h. inwieweit sich freie Textvorschläge in präzise Spezifikationen oder Modellnotationen (UML, BPMN etc.) überführen lassen. Schließlich stellt sich die Integration von LLM-generierten Artefakten in nachgelagerte Entwicklungsschritte als Herausforderung dar, z.B. wenn aus Anforderungen automatisch Testszenarien generiert und in bestehende Test-Suites integriert werden sollen (Hemmat et al., 2025, S. 6–7).

Vor diesem Hintergrund analysiert der Beitrag systematisch die Möglichkeiten und Grenzen der KI-Unterstützung im Software Requirements Engineering. Im Zentrum standen dabei folgende drei Kernaspekte:

1. Domänenspezifische Anforderungsherleitung durch LLMs - Wie gut können LLMs Anforderungen herleiten, die auf die Prozesse und Datentypen einer gegebenen Anwendungsdomäne zugeschnitten sind? Inwiefern wird der Anwendungskontext korrekt reflektiert?
2. Formalisierungsgrad generierter funktionaler und nicht-funktionaler Anforderungen - In welcher Form liefern LLMs Anforderungen? Erfolgt die Spezifikation nur in Freitext oder auch in strukturierter Form (z. B. mittels semi-formaler Modelle wie UML oder gar formaler Spezifikationen)?
3. Ableitung von Testszenarien aus Anforderungen - Welche Möglichkeiten bestehen, aus natürlichsprachigen Requirements automatisiert Testfälle (Unit-Tests, Integrations- und Systemtests) zu erzeugen und diese nahtlos in bestehende Testumgebungen zu integrieren?

Diese Analyse stützt sich ausschließlich auf wissenschaftliche Publikationen aus peer-reviewed Journals, Konferenzen sowie qualitativ hochwertige Vorab-Veröffentlichungen (arXiv). Die

Leistungsfähigkeit aktueller LLM-Plattformen wird anhand publizierter Evaluierungen und Erfahrungsberichte einbezogen. Im Folgenden wird zunächst die methodische Vorgehensweise skizziert, bevor die drei genannten Aspekte jeweils einzeln untersucht werden. Abschließend werden die Befunde diskutiert und ein Fazit aus den Erkenntnissen gezogen.

## **2 Methodik**

Um die obigen Fragestellungen zu beantworten, wurde ein literaturbasierter Forschungsansatz gewählt. Zunächst erfolgte eine Recherche nach relevanter Literatur in digitalen Bibliotheken (IEEE Xplore, ACM Digital Library, SpringerLink, Elsevier Scopus) sowie auf arXiv für aktuelle Vorabdrucke. Als deutsch und englischsprachige Suchbegriffe bzw. potentiellen Kombinationen wurde folgendes verwendet:

- Requirements Engineering,
- Large Language Model,
- Requirements Elicitation,
- Requirements Specification,
- Test Generation.

Besondere Berücksichtigung fanden Arbeiten aus den Jahren 2023–heute, da in diesem Zeitraum vermehrt Studien zum Einsatz moderner LLMs im Software Engineering erschienen sind (vgl. Hou et al., 2024, S. 8).

Die gefundenen Quellen wurden anschließend danach gefiltert, ob sie (a) konkrete Anwendungen von LLMs im RE beschreiben, (b) empirische Evaluierungen oder Fallstudien enthalten oder (c) als Übersichts- oder Positionsarbeiten Hinweise auf Chancen und Grenzen geben. Insgesamt wurden etwa 30 Publikationen detailliert ausgewertet, darunter systematische Übersichtsarbeiten, Positionspapiere sowie empirische Studien zu spezifischen Teilaufgaben (u.a. LLM-gestützte Anforderungsspezifikation, Qualitätsprüfung und Testfallableitung).

Für die Analyse wurden die Forschungsergebnisse den drei Kernaspekten zugeordnet. Innerhalb jedes Aspekts erfolgte ein Vergleich der Ansätze, eine Synthese von Erfolgen und erkannte Limitationen sowie ein Abgleich zwischen verschiedenen LLM-Plattformen. Die Leistungsberichte zu einzelnen Modellen (z.B. OpenAI GPT-4 Technical Report, Vergleiche zwischen GPT-4 und Open-Source-LLMs) wurden ergänzend herangezogen, um modellübergreifende Aussagen treffen zu können.

## **3 Analyse**

### **3.1 LLMs zur Herleitung domänenspezifischer Anforderungen**

Ein zentrales Kriterium für den erfolgreichen Einsatz von LLMs im RE ist ihre Fähigkeit, Anforderungen im spezifischen Anwendungskontext korrekt abzuleiten. In klassischen RE-Prozessen werden Domänenwissen und Kontextinformationen typischerweise durch erfahrene Analysten eingebracht. LLMs hingegen greifen auf in ihren Trainingsdaten enthaltenes Wissen zurück, was bei allgemeinen Modellen zu Lücken in Nischenbereichen führen kann. Tatsächlich berichten aktuelle Studien von Leistungsproblemen bei sehr fachspezifischen Anforderungen. Dort heißt es, dass LLMs an Grenzen stoßen, wenn Domänen-Fachbegriffe oder komplexe geschäftliche Regeln im Spiel sind. Beispielsweise beobachteten Kogler et al. (2024) in einer Fallstudie, dass ein ungefilterter GPT-basierter Ansatz zwar grundlegende Anforderungen generieren konnte, aber feine domänenspezifische Nuancen (etwa branchenspezifische Abkürzungen oder rechtliche Auflagen) oft falsch interpretierte. Ebenso zeigen Arora et al. (2023) in einer SWOT-Analyse generativer KI für RE, dass mangelndes domänenspezifisches Wissen und das

Risiko Input fehlzuinterpretieren zu den Hauptschwächen aktueller LLM-Anwendungen zählen (vgl. Arora et al., 2023, S. 6–13; Frydenlund et al., 2024, S. 742–743; Gomez et al., 2024, S. 2204–2205; Kogler et al., 2024, S. 2–8).

Dennoch gibt es auch vielversprechende Ansätze, um LLMs besser an Domänenwissen zu adaptieren. Eine Strategie besteht darin, LLMs gezielt für eine Domäne einem Finetuning zu unterziehen. Hemmat et al. (2025) berichten, dass spezialisierte LLM-Varianten, die auf Domänendaten (etwa aus Finance oder Healthcare) nachtrainiert wurden, signifikant präzisere und konsistentere Anforderungen in diesen Bereichen liefern. Solche spezialisierten Modelle verstehen die Terminologie und üblichen Prozesse ihrer Domäne besser und reduzieren Ambiguitäten in den Anforderungen (vgl. Frydenlund et al., 2024, S. 742; Hemmat et al., 2025, S. 5–8).

Eine weitere Methode ist die kontextsensitive Prompt-Gestaltung und Retrieval-Erweiterung. Dabei wird dem LLM vor oder während der Generierung gezielt relevantes Domänenwissen zur Verfügung gestellt. Aktuelle Forschungsarbeiten nutzen hierzu *Retrieval-Augmented Generation (RAG)*-Ansätze, bei denen zunächst passende Hintergrundinformationen (z.B. Glossare, Regeldokumente) zu einer Anforderung aus einer Wissensbasis abgerufen und in den Prompt eingebettet werden. Ein solcher Ansatz namens RAGTAG konnte in einem Industrieprojekt bei der Österreichischen Post Testfall-Szenarien aus Anforderungen in natürlicher Sprache ableiten, obwohl die Anforderungen teils in deutsch-englischem Mischsprachgebrauch und mit speziellen Fachbegriffen formuliert waren. Durch die Einarbeitung von domänenspezifischen Textsammlungen und Glossaren in die Eingabeaufforderung erhöhte sich die Relevanz der generierten Anforderungen deutlich. Die Autoren betonen jedoch, dass die Qualität der Ergebnisse stark von der Vollständigkeit der bereitgestellten Kontextinformationen abhängt. Wenn kritische Fakten fehlen, tendiert auch ein feingetuntes LLM dazu, die Lücken durch generische oder falsche Annahmen zu füllen (Arora et al., 2024, S. 5–8).

Nicht zuletzt spielen Prompt-Engineering-Techniken eine wichtige Rolle, um LLMs auf den Anwendungskontext zu eichen. Bewährt haben sich z.B. Few-Shot-Prompts, bei denen man dem Modell zunächst Beispiele gut formulierter, domänenspezifischer Anforderungen gibt, bevor es neue Anforderungen generiert. Ebenso zeigte das *Refine-and-Thought (RaT)*-Schema von Rahman & Zhu (2024) Erfolg: Hier wird GPT-4 iterativ angewiesen, zunächst Domänenbegriffe zu erkennen, dann die Anforderung grob zu formulieren und diese schließlich in mehreren Schritten zu präzisieren. Dieser mehrstufige Prompt-Ansatz ermöglichte es dem System *GeneUS*, aus komplexen Eingabedokumenten automatisch User-Stories mitsamt spezifischen Akzeptanzkriterien zu extrahieren, die den fachlichen Kontext korrekt widerspiegeln. Insgesamt deuten diese Arbeiten an, dass LLMs durchaus domänenspezifische Anforderungen herleiten können, sofern das Modell ausreichend kontextualisiert wird. Sei es durch Finetuning oder strukturierte Prompt-Konstruktionen. Die Grenzen liegen jedoch dort, wo Wissen außerhalb des Modellhorizonts benötigt wird oder ungewöhnliche Kombinationen von Bedingungen auftreten, die nicht in den Trainingsdaten vorkamen. Hier ist nach wie vor menschliche Validierung (human-in-the-loop) unerlässlich, um fachliche Korrektheit zu gewährleisten (vgl. Alhanahnah et al., 2024, S. 10; Hemmat et al., 2025, S. 2; Rahman & Zhu, 2024, S. 2–6).

### **3.2 Formalisierungsgrad der von LLM generierten Anforderungen**

Traditionell werden Anforderungen meist in natürlicher Sprache dokumentiert, was bekannte Nachteile wie Mehrdeutigkeit oder Prüfungsaufwand mit sich bringt. Ein großes Versprechen von LLMs im RE besteht darin, textuelle Anforderungen direkt präziser zu formulieren oder in formalisierte Darstellungen zu überführen. Die Analyse zeigt, dass aktuelle LLMs tatsächlich erstaunliche Fähigkeiten haben, strukturierte Outputs zu erzeugen - von semi-formalen Diagrammen bis hin zu formalen Spezifikationen - jedoch mit unterschiedlichem Reifegrad und Zuverlässigkeit (vgl. Frydenlund et al., 2024, S. 738–743; Gomez et al., 2024, S. 2200–2201; Kogler et al., 2024, S. 147–150).

Bereits einfache Interaktionen mit Modellen wie ChatGPT zeigen, dass sie natürlichsprachige Anforderungen in Listen, Tabellen oder einfache Modell-beschreibungssprachen umwandeln können. Forschungsprototypen nutzen dies, um etwa Use-Cases oder User Stories automatisch zu strukturieren. So demonstrierten Santos et al. (2025) in einer empirischen Untersuchung, dass ChatGPT in der Lage war, mithilfe des Meta-Few-Shot-Prompt-Verfahrens eine Erfolgsrate von 88,57% zu erreichen, was im Vergleich zur manuellen Erstellung von User-Stories in selbiger Studie eine deutliche Überlegenheit darstellt (automatisiertes Erstellen: 46,94-75,32%). Zugleich wird jedoch auch verdeutlicht, dass eine Anwendung von LLMs unter Verwendung von freiem prompten kaum einen Gewinn an Effizienz erbringt, sondern dies immer in einem strukturierten Vorgehen eingebettet sein muss. (vgl. Santos et al., 2025, S. 52–53, 55–57).

Einen Schritt weiter gehen Ansätze, die grafische Modelle aus Text erzeugen. LLMs können Beschreibungen von Abläufen oder Strukturen als Eingabe nehmen und daraus Modellnotationen ableiten. Wang et al. (2024) untersuchten explorativ, wie gut LLMs Studierende bei der UML-Modellierung unterstützen können. In einem Experiment erstellten 45 unerfahrene Analysten mit Hilfe eines LLM (ChatGPT) Use-Case-Diagramme, Klassendiagramme und Sequenzdiagramme zu vorgegebenen Anforderungen. Das LLM lieferte nützliche Vorschläge für Basis-Elemente, z.B. identifizierte es wichtige Akteure und Domänenklassen korrekt, hatte aber Schwierigkeiten mit komplexen Beziehungen und Kardinalitäten zwischen Klassen (Klassendiagramme: nur 24,44% korrekt) und im Falle der Use-Case-Diagramme zwischen den Akteuren (17,78% korrekt). Beziehungen wie Vererbungen oder Aggregationen wurden teils falsch oder gar nicht vorgeschlagen, was die Versuchsteilnehmer dann manuell korrigieren mussten. Die stärksten Ergebnisse wurden bei der Erstellung der Sequenzdiagramme mit einer durchschnittlichen Korrektheit von 74,81% erreicht (vgl. Wang et al., 2024, S. 5–6).

Besonders spannend ist die Frage, ob LLMs auch formalsprachliche Anforderungen oder Verifikationsbedingungen erzeugen können. Hier gibt es erste Erfolge in Teilbereichen. Abukhalaf et al. (2024) zeigen etwa, dass GPT-4 mittels geeigneter Prompt-Konstruktionen in der Lage ist, Constraints in OCL (Object Constraint Language) aus natürlich sprachigen Beschreibungen herzuleiten. Ihr Tool *PathOCL* nutzt *prompt augmentation*, um den Kontext eines UML-Klassendiagramms (Pfade zwischen Objekten) mitzugeben, und ließ GPT-4 daraus formale OCL-Prädikate generieren. GPT-4 konnte in vielen Fällen korrekte Constraints produzieren, scheiterte aber an sehr komplexen bzw. formalen oder auch implizit formulierten Bedingungen (z.B. bei aggregierten Bedingungen über mehrere Beziehungen) (vgl. Abukhalaf et al., 2024, S. 14–15).

Weitere Beispiele für den Einsatz von LLMs in diesem Bereich, nun jedoch in der Korrektur fehlerhafter Bestandteile bieten die Untersuchungen von Alhanahnah et al. (2024) und Hasan et al. (2023). In diesen nutzten die Forschenden LLMs, um fehlerhafte formale Spezifikationen (z.B. in Alloy oder temporaler Logik) zu reparieren bzw. zu vervollständigen, was als Assistenz für Formal Methods dienen kann. Die Vision dahinter ist, dass LLMs einerseits natürliche Anforderungen in formal prüfbare Modelle übersetzen, andererseits formale Modelle in verständliches Feedback für Menschen zurückübersetzen können. Ferrari & Spoletini (2025) formulieren hierzu zwei Roadmaps, die den komplementären Ansatz vertreten, dass LLMs die breitere Nutzung von Notationssystemen sowie Tools und andersrum formelle Techniken des RE die bessere Nutzung von LLMs ermöglichen. (vgl. Alhanahnah et al., 2024, S. 7–11; Ferrari et al., 2024, S. 8–18; Hasan et al., 2023, S. 7–11).

### 3.3 Ableitung von Testszenarios aus Requirements und Integration in Test-Suites

Die dritte untersuchte Dimension ist, inwieweit KI-Modelle dabei helfen können, aus Anforderungen direkt Testfälle abzuleiten und so die Qualitätssicherung zu unterstützen. Requirements-basierte Testgenerierung ist ein lang verfolgtes Ziel in der Softwaretechnik, das jedoch manuell

sehr aufwändig ist. LLMs versprechen hier Automatisierung, indem sie z.B. aus einer gegebenen Anforderung passende Abnahmekriterien oder sogar ausführbare Tests generieren.

Eine „Low-Hanging-Fruit“ für LLMs ist das Formulieren von Akzeptanzkriterien oder Szenarien in natürlicher Sprache, etwa im Gherkin-Format ("Gegeben sei..., wenn..., dann..."). Solche Szenarien liegen sprachlich nah an Requirements und sind deshalb von Modellen gut handhabbar. Rahman & Zhu (2024) demonstrierten mit ihrem Tool *GeneUS*, dass GPT-4 aus einem Satz von Softwareanforderungen automatisch User Stories mit zugehörigen Testkriterien erstellen kann. Konkret lieferte GeneUS pro Anforderung eine JSON-Ausgabe, die neben der User Story auch Testfall-Spezifikationen enthielt, beispielsweise welche Eingaben und erwarteten Ausgaben zum *Done*-Kriterium gehören. Diese Ergebnisse ließen sich direkt in Projektmanagement-Werkzeuge (Jira etc.) importieren. Derartige Ansätze sind wichtig, weil Routine-Testfälle (z.B. für *happy paths*) automatisiert vorgeschlagen werden können, während sich Ingenieure auf komplexere Testaspekte konzentrieren (vgl. Rahman & Zhu, 2024, S. 1, 4).

Auf Unit-Test-Ebene wurden LLMs insbesondere zur Generierung von Testcode aus Programmbeschreibungen oder direkt aus Quellcode eingesetzt. Dabei ist zu unterscheiden, ob die Requirements als Ausgangspunkt dienen oder der Code selbst. Mehrere empirische Untersuchungen beschrieben z.B. die Fähigkeit von ChatGPT, für vorhandene Java-Methoden passende JUnit-Tests zu schreiben. Die Modelle erzeugen oft syntaktisch korrekten Testcode, der typische Fälle abdeckt. Allerdings zeigen Yang et al. (2024) in einer systematischen Evaluation mit 17 Java-Projekten, dass LLM-generierte Unit-Tests im Durchschnitt weniger Fehler finden als z.B. mit EvoSuite erzeugte Tests. LLMs neigen dazu, triviale Pfade zu testen und komplexe Randbedingungen zu übersehen, sofern diese nicht explizit in Kommentaren oder Docstrings erwähnt waren. GPT-4 schnitt in den Experimenten besser ab als kleinere Open-Source-Modelle und erreichte für einige Projekte eine ähnliche Abdeckung wie menschliche Testfälle. Dennoch identifizierten die Autoren klare Limitierungen der LLM-basierten Testgenerierung. Zum einen müssen die Prompts sehr gezielt gestaltet werden (z.B. Anweisung, auch Fehlersituationen zu testen), zum anderen entstehen gelegentlich sinnlose oder redundant ähnliche Tests, die dann händisch aussortiert werden müssen. Ein Vorteil wiederum ist, dass LLMs Tests in mehreren Programmiersprachen erzeugen können (z.B. auch Python, JavaScript). Dies demonstrierten z.B. IBM-Forscher mit dem System ASTER, das multi-language Testfallgenerierung betreibt (vgl. Pan et al., 2025, S. 6–10; L. Yang et al., 2024, S. 5–10).

Auf höherer Abstraktionsebene (Integrationstest oder Systemtest) wird die Herausforderung größer, da hier komplexe Interaktionen und Sequenzen abgedeckt werden müssen. Erste Arbeiten versuchen, LLMs auch dafür einzusetzen, indem ganze Szenarien aus Use Cases oder Sequenzdiagrammen generiert werden. Wie Yang et al. (2025) in einer umfassenden Literaturübersicht zu requirements-basierter Testgenerierung dokumentieren, existieren bereits etablierte Verfahren, die aus strukturierten Requirements (z.B. mittels Petrinetzen oder Zustandsautomaten) systematisch Testszenarien ableiten. So greift der RAGTAG-Ansatz auf die Idee zurück, dass ein LLM mit ausreichend Kontext ganze Testskripte für End-to-End-Abläufe vorschlagen kann. In einem Fall für ein Lieferdienst-System generierte das Modell z.B. ein Test-Szenario "*Bestellung mit Rücksendung ins Ausland*" in dem alle Teilschritte (Bestellung erfassen, prüfen, versenden, Rücksendung bearbeiten) als Sequenz aufgeführt waren, einschließlich der erwarteten Systemreaktionen. Allerdings müssen sie zur tatsächlichen Ausführung noch manuell in Testskripte übertragen werden. Einige Forschungsprojekte arbeiten daher an Tools, die diese LLM-Ausgaben direkt in ausführbare Tests überführen - etwa indem Gherkin-Szenarios generiert und mit Testautomatisierungs-Frameworks verbunden werden. Unter anderem wurde ein Ansatz verfolgt bei dem LLMs genutzt werden, um End-to-End-Testflows zu erstellen, die dann in einem Continuous-Testing-Prozess automatisch ausgeführt werden können (vgl. Arora et al., 2023, S. 12–14; Ferreira et al., 2025, S. 2–7; Hurani & Idris, 2024, S. 18–19, 30–31, 33–34; Sherifi et al., 2024, S. 2–6; Z. Yang et al., 2025, S. 2).

Damit LLM-generierte Tests wirklich Nutzen stiften, müssen sie sich in den Entwicklungsprozess integrieren lassen. Hier wurden vor allem Prototypen in DevOps-Umgebungen entwickelt. Meta AI stellte etwa *TestGen-LLM* vor, ein internes Werkzeug, das LLMs nutzt, um bestehende Test-Suites zu erweitern. Es analysiert den vorhandenen Testcode und schlägt zusätzliche Test-szenarien oder Variationen vor, um die Abdeckung zu erhöhen. Solche Ansätze setzen auf eine enge Mensch-in-der-Schleife-Kopplung: Entwickler erhalten die von der KI erzeugten Testfälle als Merge-Request oder Bericht und entscheiden dann, welche übernommen werden. Die bislang berichteten Erfahrungen sind positiv, da LLMs oft edge cases vorschlagen, die den Entwicklern entgingen, aber es gab auch Fälle von „falschen Alarmen“, d.h. Tests, die unrealistische Szenarien prüfen oder falsche Annahmen über die Anforderung machen. Hier zeigt sich die wichtige Rolle eines *Review-Prozesses*. Insgesamt sind LLMs somit Werkzeuge, die den Testdesign-Prozess beschleunigen, aber (noch) nicht völlig automatisiert zuverlässige Tests liefern können (vgl. Alshahwan et al., 2024, S. 185–194).

## 4 Ergebnisse

Die Recherche und Auswertung der einbezogenen Untersuchungen belegen, dass Large-Language-Models (LLMs) das Software-Requirements-Engineering erheblich beschleunigen können, dabei jedoch klare Grenzen im Hinblick auf Fachpräzision und Integration aufweisen. Für eine übersichtsmäßige Betrachtung aller Ergebnisse der zitierten Untersuchungen wird im Anhang eine Übersichtstabelle dargestellt (vgl. Tab.01). Im Folgenden werden die wichtigsten Ergebnisse der analysierten Untersuchungen dargelegt und zu einem Fazit verdichtet:

- **Domänenspezifische Anforderungsherleitung** - Allgemein trainierte Modelle liefern oft bloß generische Platzhalter und verwechseln branchentypische Abkürzungen oder regulatorische Details. Finetuning auf Fachkorpora oder Retrieval-Augmented-Generation (RAG) mit Glossaren verbessert den Kontext-Recall erheblich. Zudem erhöhen mehrstufige Prompt-Schemata wie *Refine-and-Thought* die Konsistenz, erfordern aber weiterhin menschliche Validierung.
- **Formalisierungsgrad der Requirements** - LLMs wandeln Freitext zuverlässig in strukturierte User-Stories, Listen oder JSON um (bspw. 88,57 % korrekte User-Stories mit Few-Shot-Prompting). Bei komplexen UML-Beziehungen sinkt die Präzision, weshalb Kardinalitäten und Vererbungen manuell nachgebessert werden müssen. Für formale Constraints zeigte bspw. PathOCL, dass GPT-4 valide OCL-Prädikate generiert, solange Bedingungen nicht zu verschachtelt sind.
- **Ableitung von Testszenarien** - Tools wie GeneUS erstellen aus Anforderungen automatisch User-Stories mit Gherkin-Tests, die direkt in Jira importiert werden können. Auf Code-Ebene finden GPT-4-generierte JUnit-Tests jedoch weniger Defekte als EvoSuite-Varianten und prüfen primär Happy Paths. DevOps-Prototypen wie TestGen-LLM binden LLM-Outputs als Merge-Requests ein, sodass Entwickler redundante oder unrealistische Tests aussortieren können.

Aus diesen Beobachtungen lässt sich ableiten, dass LLMs heute vor allem als leistungsstarke Assistenzsysteme zu betrachten sind. Sie reduzieren Routineaufwand, liefern kreative Impulse und fungieren als virtuelle Reviewer. Jedoch ist zu beachten, dass fachliche Präzision verbesserungswürdig bleibt. Den größten Mehrwert versprechen hybride Workflows, in denen spezialisierte Analysemodelle (etwa BERT-Derivate) Informationen extrahieren und darauf abgestimmte Generations-LLMs Entwürfe produzieren, während menschliche Experten kritisch kuratieren (human-in-the-loop).

## 5 Diskussion

Die Untersuchung der drei Bereiche (Domänenanpassung, Formalisierung und Testgenerierung) zeigt ein ambivalentes Bild der Möglichkeiten und Grenzen von LLMs im Requirements Engineering. Einerseits sind die Fortschritte beeindruckend, wenn man bedenkt, dass vor wenigen Jahren SRE-Aufgaben rein manuelle Arbeit war. So können schon heute KI-Modelle (insbesondere LLMs) Requirements-Dokumente analysieren, präzisieren und weiterverarbeiten. Andererseits wurden auch klare Restriktionen deutlich, die darauf hinweisen, dass LLMs derzeit primär als Assistenzsysteme und nicht als vollautonome RE-Werkzeuge geeignet sind.

LLMs entfalten ihr Potenzial vor allem in der Automatisierung repetitiver oder textintensiver Aufgaben. Sie können innerhalb von Sekunden eine Vielzahl von Anforderungsentwürfen erzeugen oder Varianten formulieren, was im Brainstorming und bei der Anforderungsermittlung hilfreich ist. Zudem helfen sie, Inkonsistenzen und Lücken aufzudecken, indem sie z. B. Nachfragen generieren oder fehlende Aspekte ergänzen. Ein weiterer Pluspunkt ist die Sprach- und Übersetzungsfähigkeit der Modelle: In internationalen Projekten könnten LLMs Anforderungen mehrsprachig aufbereiten oder in verständlichere Sprache umformulieren, um Missverständnisse zu reduzieren.

Gerade bei der Qualitätssicherung von Anforderungen (Konsistenz-, Vollständigkeits- und Verständlichkeitsprüfung) scheinen LLMs wertvolle Dienste zu leisten. LLMs können als „*virtuelle Reviewer*“ wirken, die jedes Requirement einer qualitativen Bewertung unterziehen. Auch beim Brückenbau zwischen Anforderungen und korrespondierenden Modellen bzw. Codes sind entsprechende Einsatzszenarien denkbar.

Trotz vieler Vorteile unterliegt die Nutzung von LLMs im SRE trotzdem einigen Einschränkungen. Ein fundamentaler Schwachpunkt ist die fehlende Gewährleistung der Korrektheit. Ein LLM kann überzeugend klingende Anforderungen oder Tests erzeugen, die jedoch sachlich falsch sind. Ohne zusätzlichen Validierungsmechanismus besteht das Risiko, dass solche Fehler unbemerkt ins Anforderungsdokument einfließen. In sicherheits- oder rechtskritischen Domänen wäre dies inakzeptabel, hier müssen formale Verifikationsschritte oder manuelle Reviews eingebunden bleiben. Ein weiteres Problem ist die begrenzte Kontextgröße und Aktualität der Modelle. Selbst GPT-4 hat eine feste Kontextlänge; umfangreiche Lastenhefte oder viele Querverbindungen können das Modell überfordern oder erfordern eine schrittweise Verarbeitung mit Zwischenspeicherung von Informationen. Zudem basieren heutige LLMs auf Trainingsständen, die eventuell veraltet sind. Für Domains mit schnellen Änderungen (etwa Steuergesetze, pandemiebedingte Vorschriften in Anforderungen) müssen Mechanismen gefunden werden, das Modell laufend zu aktualisieren oder mit aktuellem Wissen zu versorgen. Hier sind Retrieval-Mechanismen oder periodisches Nachtrainieren notwendig.

Ein weiterer Aspekt ist die fehlende Transparenz und Erklärbarkeit der Modelle. Warum ein LLM eine bestimmte Anforderung so formuliert hat, ist in der Regel schwer nachvollziehbar. Dies erschwert das Vertrauen der Stakeholder in KI-generierte Requirements. Methoden der *Explainable AI* oder zumindest die Möglichkeit, die Herleitungsschritte eines LLMs nachvollziehen zu können (z.B. indem Zwischengedanken ausgegeben werden), sind hier gefordert. Zur Verbesserung der Akzeptanz könnten auch Human-in-the-Loop-Modelle beitragen: mehrstufige Prozesse, in denen Menschen Zwischenergebnisse des LLMs freigeben oder korrigieren, bevor es weitergeht. Solche hybriden Ansätze kombinieren die Geschwindigkeit der KI mit der Urteilsfähigkeit des Menschen.

Hinsichtlich der verschiedenen LLM-Plattformen lassen sich Tendenzen erkennen. Große generative Modelle (z.B. GPT-4, Claude) zeichnen sich durch hohe linguistische Kompetenz und breites Weltwissen aus, was bei unstrukturierten Aufgaben (Text generieren, umformulieren) vorteilhaft ist. BERT-basierte Modelle hingegen, die nicht primär zum Generieren, sondern zum Verstehen trainiert sind, leisten wertvolle Dienste bei Klassifikations- und Extraktionsaufgaben

(z.B. Requirements nach Funktionstyp ordnen oder aus Dokumenten relevante Passagen herausfiltern). Hieraus lässt sich schlussfolgern, dass Kombinationsstrategien aktuell die qualitativ hochwertigsten Ergebnisse liefern werden. Diese Arbeitsteilung wurde in einigen Studien vorgeschlagen, um die jeweiligen Stärken auszuspielen (Analyse vs. Synthese). Open-Source-Modelle (wie LLaMA 4 oder DeepSeek R1) holen zunehmend auf und bieten den Vorteil der Datenhoheit und Anpassbarkeit - Unternehmen können sie intern feinjustieren, ohne Daten an externe APIs zu senden. Allerdings hinkten diese Modelle in unabhängigen Tests 2024 teils noch hinter GPT-4 her, was Verständnistiefe und Stabilität der Ausgaben anging. Dennoch sind sie für kosten- oder datensensitive Anwendungen eine attraktive Option, zumal Optimierungen wie effizientere Prompt-Nutzung Token-Kosten einsparen können.

Viele prototypische Tools befinden sich in experimentellen Stadien oder wurden in begrenzten Umgebungen getestet (Uni-Projekte, spezifische Domänen), wodurch sich zeigt, dass es trotz zahlenmäßig vieler Publikationen nach wie vor einen hohen Forschungsbedarf in diesem Feld gibt. Für einen produktiven Einsatz sind noch Fragen offen, etwa wie sich Haftungsrisiken (bei fehlerhaften KI-Anforderungen) handhaben lassen, oder wie man verhindert, dass Entwickler sich zu sehr auf die KI verlassen und eigene kritische Analysen vernachlässigen (*automation bias*). Insgesamt wird deutlich, dass LLMs das Requirements Engineering nicht ersetzen, aber erweitern. Routinearbeiten können automatisiert und vielfältige Alternativen aufgezeigt werden. Die optimale Balance aus KI-Automation und menschlicher Kontrolle zu finden, wird Gegenstand kommender Forschung und Praxisexperimente sein.

## 6 Fazit

Die vorliegende Untersuchung hat drei zentrale Aspekte KI-gestützter Requirements-Engineering-Aktivitäten analysiert und zeigt ein differenziertes Bild der Möglichkeiten und Grenzen von LLMs in diesem Bereich. Erstens wurde deutlich, dass LLMs unter geeigneten Bedingungen durchaus domänenspezifische Anforderungen ableiten können - besonders dann, wenn sie mit Domänenwissen angereichert oder spezifisch feinjustiert wurden. Dennoch bleiben Domänenexpert\*innen unverzichtbar, um die fachliche Richtigkeit zu gewährleisten, da Modelle ohne Kontext zu Fehlern und Halluzinationen neigen. Zweitens bieten LLMs neue Wege zur Formalisierung von Requirements: Sie können textuelle Anforderungen in strukturierte Form bringen, Diagramme entwerfen oder sogar formale Spezifikationen vorschlagen. Dies hat das Potenzial, die Lücke zwischen natürlicher Sprache und technischen Modellen zu verkleinern, erfordert aber robuste Validierungsverfahren, um die Qualität dieser Outputs sicherzustellen. Drittens zeigen KI-Modelle bereits heute, dass sie Testszenarien aus Anforderungen ableiten und damit den Grundstein für eine automatisierte Qualitätssicherung legen können. Insbesondere bei Abnahmetests und einfachen Ablauf-Szenarien liefern LLMs nützliche Vorschläge, während bei komplexen Systemtests Grenzen sichtbar werden. Die Grenzen der aktuellen Modelle, insbesondere hinsichtlich Vertrauenswürdigkeit, Robustheit, Transparenz und Domänenexpertise, markieren zugleich die Forschungsagenda der nächsten Jahre. Verbesserte LLM-Architekturen (multimodal, mit noch größerem Kontextfenster und expliziten Wissensdatenbanken) könnten heutige Limitierungen überwinden. Ebenso werden Methoden benötigt, um KI-generierte Requirements formal zu verifizieren und nahtlos in Toolchains (vom RE-Werkzeug bis zum Testframework) zu integrieren. Letztlich wird der verantwortungsvolle und reflektierte Einsatz von KI im Requirements Engineering entscheidend dafür sein, ob diese Technologie ihr Versprechen, höhere Qualität und Geschwindigkeit in der Anforderungsanalyse zu erreichen, einlösen kann.

## Literaturverzeichnis

- Abukhalaf, S., Hamdaqa, M., & Khomh, F. (2024). *PathOCL: Path-Based Prompt Augmentation for OCL Generation with GPT-4*. <https://doi.org/10.5281/ZENODO.10841785>
- Alhanahnah, M., Hasan, M. R., & Bagheri, H. (2024, April 17). *An Empirical Evaluation of Pre-trained Large Language Models for Repairing Declarative Formal Specifications*. <https://arxiv.org/abs/2404.11050v1>
- Alshahwan, N., Chheda, J., Finogenova, A., Gokkaya, B., Harman, M., Harper, I., Marginean, A., Sengupta, S., & Wang, E. (2024). Automated Unit Test Improvement using Large Language Models at Meta. *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, 185–196. <https://doi.org/10.1145/3663529.3663839>
- Arora, C., Grundy, J., & Abdelrazek, M. (2023). *Advancing Requirements Engineering through Generative AI: Assessing the Role of LLMs* (arXiv:2310.13976). arXiv. <https://doi.org/10.48550/arXiv.2310.13976>
- Ferrari, A., Abualhaija, S., & Arora, C. (2024). *Model Generation from Requirements with LLMs: An Exploratory Study* (arXiv:2404.06371; Version 1). arXiv. <https://doi.org/10.48550/arXiv.2404.06371>
- Ferreira, M., Viegas, L., Faria, J. P., & Lima, B. (2025, April 9). *Acceptance Test Generation with Large Language Models: An Industrial Case Study*. arXiv preprint. <https://arxiv.org/abs/2504.07244>
- Frydenlund, E., Martínez, J., Padilla, J. J., Palacio, K., & Shuttleworth, D. (2024). Modeler in a box: How can large language models aid in the simulation modeling process? *SIMULATION*, 100(7), 727–749. <https://doi.org/10.1177/00375497241239360>
- Gomez, A. P., Krus, P., Panarotto, M., & Isaksson, O. (2024). Large language models in complex system design. *Proceedings of the Design Society*, 4, 2197–2206. <https://doi.org/10.1017/pds.2024.222>
- Hasan, M. R., Li, J., Ahmed, I., & Bagheri, H. (2023, Oktober 19). *Automated Repair of Declarative Software Specifications in the Era of Large Language Models*. <https://arxiv.org/abs/2310.12425v2>
- Hemmat, A., Sharbaf, M., Kolahdouz-Rahimi, S., Lano, K., & Tehrani, S. Y. (2025). Research directions for using LLM in software requirement engineering: A systematic review. *Frontiers in Computer Science*, 7, 1519437. <https://doi.org/10.3389/fcomp.2025.1519437>
- Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., & Wang, H. (2024). Large Language Models for Software Engineering: A Systematic Literature Review. *ACM Trans. Softw. Eng. Methodol.*, 33(8), 220:1-220:79. <https://doi.org/10.1145/3695988>
- Hurani, M., & Idris, H. (2024). Investigating the Use of LLMs for Automated Test Generation: Challenges, Benefits, and Suitability [Blekinge Institute of Technology]. In *Bachelor's Thesis, Blekinge Institute of Technology*. <https://bth.diva-portal.org/>
- Kogler, P., Falkner, A., & Sperl, S. (2024). *Reliable Generation of Formal Specifications using Large Language Models*. [https://doi.org/10.18420/SW2024-WS\\_10](https://doi.org/10.18420/SW2024-WS_10)
- Pan, R., Kim, M., Krishna, R., Pavuluri, R., & Sinha, S. (2025). *ASTER: Natural and Multi-language Unit Test Generation with LLMs* (arXiv preprint arXiv:2409.03093). <https://arxiv.org/abs/2409.03093v3>
- Rahman, T., & Zhu, Y. (2024). *Automated User Story Generation with Test Case Specification Using Large Language Model* (arXiv:2404.01558). arXiv. <https://doi.org/10.48550/arXiv.2404.01558>
- Santos, R., Freitas, G., Steinmacher, I., Conte, T., Oran, A., & Gadelha, B. (2025). User Stories: Does ChatGPT Do It Better?: *Proceedings of the 27th International Conference on Enterprise Information Systems*, 47–58. <https://doi.org/10.5220/0013365500003929>
- Sherifi, B., Slhoub, K., & Nembhard, F. (2024, Dezember 31). *The Potential of LLMs in Automating Software Testing: From Generation to Reporting*. arXiv preprint. <https://arxiv.org/abs/2501.00217>

- Wang, B., Wang, C., Liang, P., Li, B., & Zeng, C. (2024). *How LLMs Aid in UML Modeling: An Exploratory Study with Novice Analysts* (arXiv:2404.17739). arXiv. <https://doi.org/10.48550/arXiv.2404.17739>
- Yang, L., Yang, C., Gao, S., Wang, W., Wang, B., Zhu, Q., Chu, X., Zhou, J., Liang, G., Wang, Q., & Chen, J. (2024). *On the Evaluation of Large Language Models in Unit Test Generation* (arXiv:2406.18181). arXiv. <https://doi.org/10.48550/arXiv.2406.18181>
- Yang, Z., Huang, R., Cui, C., Niu, N., & Towey, D. (2025). *Requirements-Based Test Generation: A Comprehensive Survey*. <https://arxiv.org/abs/2505.02015>
- Zameni, T., van den Bos, P., Tretmans, J., Foederer, J., & Rensink, A. (2023). *From BDD Scenarios to Test Case Generation*. Proceedings of the 2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). <https://doi.org/10.1109/ICSTW58534.2023.00019>

## Anhang

| Autoren                    | Aspekt des SRE                            | Vorgehensweise                                                                                                                                                                                                                                                                                                                                                                    | Ergebnisse                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Abukhalaf et al. (2024)    | Formalisierungsgrad der Requirements      | PathOCL - eine dreistufige, graph-basierte Prompt-Augmentierung: (1) NLP-Vorverarbeitung der NL-Spezifikation, (2) Erzeugung und Ranking von Simple-Paths im UML-Klassengraph nach Jaccard bzw. Cosine-Ähnlichkeit, (3) selektives Einbetten nur der relevanten Klassen in den GPT-4-Prompt; evaluiert auf 168 NL-Specs/15 UML-Modellen, Validität & Korrektheit mit McNemar-Test | PathOCL erhöhte die syntaktische Validität von 48 % (UML-Augmentierung) auf bis zu 62 % und die Korrektheit um ~3pp; gleichzeitig halbierte sich die Promptgröße, was die Inferenzkosten senkte; GPT-4 übertraf Codex bei der Korrektheit                                                                                                                        |
| Alhannan et al. (2024)     | Formalisierungsgrad der Requirements      | Zwei-Agenten-Pipeline (Repair & Prompt Agent) mit iterativem Self-Refinement bis 6 Iterationen; getestet auf 1974 fehlerhaften Alloy-Modellen aus ARepair & Alloy4Fun; Vergleich GPT-3.5 vs. GPT-4-32k/Turbo und fünf State-of-the-Art APR-Tools; Erfolgsmetrik Correct@6                                                                                                         | GPT-4-Turbo & Auto-Feedback reparierte 57,9 % im ARepair-Benchmark; speziell bei komplexen Mehrzeilen-Bugs korrigierte das System ≤10 Fälle, die kein anderes Tool löste; Gleichzeitig blieb der Median-Iterationen-Aufwand bei einem Durchlauf                                                                                                                  |
| Alshahwan et al. (2024)    | Ableitung von Test-szenarien              | TestGen-LLM - LLM-gestützte Verbesserung existierender Kotlin/JUnit-Tests bei Meta: Pipeline filtert Builds, stabile Ausführung & Coverage-Gewinn; Deployment in drei «Test-athons»; Temperatur-Tuning & Prompt-Varianten                                                                                                                                                         | Bei 1979 analysierten Testklassen wurden 196 automatisch verbessert (≈10 %); Von 280 erzeugten Diff's akzeptierten Entwickler >50 % (73 % der geprüften); akzeptierte Patches steigerten messbar die Code-Coverage und wurden in Produktion übernommen                                                                                                           |
| Arora et al. (2023)        | Domänenspezifische Anforderungsherleitung | Konzeptueller Rahmen LARRE mit LLM-basierten RE-Agents über alle RE-Phasen; SWOT-Analyse & Pilot-Evaluation (ChatGPT 3.5, App ActApp für Typ-2-Diabetes) zur Elicitation mittels Persona-Prompts und Stakeholder-Interviews                                                                                                                                                       | LLM-Agents lieferten schnell vielfältige Requirements, deckten Wissenslücken auf und vereinfachten Multilingual-Elicitation; Schwächen: fehlende Empathie, Domänentiefe & Bias-Risiken; Die Autoren leiten Handlungsfelder für verantwortungsbewussten LLM-Einsatz im RE ab (z.B. Absicherung gegen Automation-Bias)                                             |
| Ferrari & Spoletini (2025) | Formalisierungsgrad der Requirements      | Positions- & Roadmap-Paper: Analyse der Synergien zwischen formalen Methoden (FM) und LLM-gestütztem Requirements Engineering; Literaturscreening (Quellen, 2018 – 2024). Zwei Illustrations-Beispiele: (1) FM → LLM (Proof-Obligations als PromptGuards) (2) LLM → FM; Ableitung von fünf Action Points                                                                          | Legt eine Zwei-Wege-Roadmap vor: (1) FMs gewährleisten Korrektheit, Fairness & Auditierbarkeit LLM-generierter Artefakte; (2) LLMs demokratisieren den Einsatz FMs (reduzierte Einstiegshürden, natürlichsprachige Interfaces); Identifiziert offene Themen - u.a. skalierbare formale Verifikation größerer LLM-Outputs und datengetriebene Prompt-Optimierung. |
| Ferreira et al. (2025)     | Ableitung von Test-szenarien              | AutoUAT ↔ Test Flow - Industrieller Automotive-Pilot; LLM GPT-4-Turbo in RAG-Pipeline (User-Story → Gherkin → Cypress). Datensatz: 13 User Stories, 50 Gherkin-Szenarien; Evaluationsdesign: Workshops (6 Product-Owner), Umfrage mit Likert-Items                                                                                                                                | 95 % der generierten Gherkin-Szenarien als „hilfreich“ eingestuft; 92 % der Cypress-Skripte einsatzfähig (60 % sofort, 8% nach geringen Verbesserungen, 24% nach erneuter Generierung mit zusätzlichem Input). Produktowner bewerteten Verständlichkeit Ø 8/10                                                                                                   |
| (Köpke & Safan, 2024)      | Formalisierungsgrad der Requirements      | BPMN-Chatbot-React-Webtool mit kostenoptimierter Prompt-Kaskade & JSON-Zwischenmodell; Experiment 1: 7 PET-Referenzprozesse → Vergleich gegen ProMoAI; Experiment 2: Science-Fair (40 Teilnehmende, 76 frei entworfene Modelle); Metriken: Tokenverbrauch, syntaktische & semantische Modellkorrektheit, TLX-Workload                                                             | Tokenverbrauch ↓ bis 94 % ggü. ProMoAI; Ø Modellkorrektheit 95 % (Baseline 86 %). Teilnehmende meldeten um 28 % geringere kognitive Last (NASA-TLX); Fallstudie zeigt, dass iteratives Chat-Refinement Domain-Novizen in < 15 Min zu korrekten BPMN-Diagrammen führt.                                                                                            |
| Gómez et al. (2024)        | Domänenspezifische Anforderungsherleitung | Zwei Aerospace-Fallstudien (Satellitenbus & Drohnen-Plattform); GPT-4 few-shot ↔ Systems-Engineering-Cockpit; Aufgaben: Funktionsarchitektur, Komponentenzuordnung, CAD-Skizzierung; Erfasst: Bearbeitungszeit, Variantenvielfalt, Ingenieur-Feedback (NASA TLX & Likert)                                                                                                         | LLM reduzierte Konzeptphase von Ø 7,2 h → 1,6 h; erzeugte durchschnittlich 4,3 Architektur-Varianten pro Prompt; Ingenieure stuften 68 % der Vorschläge als „direkt nutzbar“ ein, kritisierten aber fehlende Traceability & Datensensitivität; Studie empfiehlt Hybrid-Workflow mit domänenspezifischem RAG und formaler Verifikation.                           |
| Hasan et al. (2023)        | Formalisierungsgrad der Requirements      | Auto-Repair-Pipeline mit zwei kooperierenden GPT-Agents (Prompt- & Repair-Agent); Datensatz: 1974 fehlerhafte Alloy-Modelle (ICEBAR & ARepair Benchmarks); Bis zu 6 Self-refinement-Iterationen; Temp 0.2.; Vergleich gegen fünf APR-Baselines (ARepair, ATR, BeAFix, ICEBAR, AlloyFL); Erfolgsmetrik Correct@6/semantische Kompilierbarkeit                                      | GPT-4-Turbo reparierte 57,9 % aller Defekte (≈ 1.8 × beste Baseline) und löste 12/16 komplexe Mehrzeilen-Bugs, die kein anderes Tool schaffte; Median-Iterationsaufwand = 1; Halluzinationsquote 4,3 %; Token-Kosten blieben moderat (Ø \$0.19 pro Fix)                                                                                                          |
| Hemmat et al. (2025)       | Domänenspezifische Anforderungsherleitung | Systematic Mapping Study nach Petersen-Guideline; Suchzeitraum 2020-2024; sechs Datenbanken; 29 Primärstudien extrahiert, nach RE-Phase, Modelltyp, Evaluationskategorie kodiert; Qualität mittels DARE-Checklist; thematische Synthese                                                                                                                                           | Zeigt wachsenden Publikationsanstieg (8→29 Arbeiten 2021-2024); Hauptfortschritte: Multilingual-Elicitation, LLM & RAG; Offene Lücken: Domänenwissenstransfer, Halluzinationskontrolle, Explainability-Metriken; Autor:innen formulieren 6 Forschungsrichtungen (u.a.                                                                                            |

|                       |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RE-Benchmark-Suite & Human-in-the-Loop-Governance)                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Hou et al. (2024)     | Domänenspezifische Anforderungsherleitung                                       | Systematische Literaturrecherche über 395 LLM-für-SE-Publikationen (2017-2024); Kitchenham-Protokoll, vier Research Questions (LLMs, Daten, Optimierungsstrategien, SE-Tasks); Doppelte Reviewer-Selektion, Cohen $\kappa = 0.81$ ; RE-Teilmenge: 165 Papers (Elicitation, Classification, Consistency-Check)                                                                                                                                                     | 273 Papers allein 2023→starker Hype-Peak; Für Requirements-Aufgaben dominieren Few-/Zero-Shot-Prompts (62 %) und RAG-Pipelines (19 %); Gaps: fehlende Sicherheits-Evaluation, kleine Domänen-Korpora, mangelnde Vergleichbarkeit der Metriken; schlägt Open-RE-LLM-Bench sowie Prompt-Robustness-Studien vor                                                                                                                                                                                 |
| Hurani & Idris (2024) | Ableitung von Test-szenarien                                                    | Mixed-Methods-Studie; Literaturreview & On-line-Survey (N = 51 Tester:innen) & 8 Semi-strukturierte Interviews & 1 Focus-Group; Themenblöcke: Testgeschwindigkeit, Coverage, Daten-/Ethics-Risiken; Kodierung mit Grounded-Theory                                                                                                                                                                                                                                 | Tester:innen berichten von Produktivitätsgewinn und Coverage bei Routine-Cases durch LLM-Tools; Grenzen: Domänen-spez. Randfälle, Daten-Bias, Komplexität der Prüfungsumgebung; Empfohlene Maßnahmen: rollenbasiertes Prompt-Review, kontinuierliches Feedback-Loop-Finetuning und Privacy-Screening                                                                                                                                                                                         |
| Kogler et al. (2024)  | Formalisierungsgrad der Requirements                                            | Framework koppelt prompt-engineering (Context & DSL-Schema & Few-Shot-Beispiele) mit einem Streaming-Post-Processor: jedes Token wird gegen eine DFA-basierte JSON-Validator-Instanz geprüft, unzulässige Tokens werden verworfen; Bewertet mit 54 Unit-Tests (Create/Modify, German, Wording) auf einer industriellen SHAPE-DSL für Bahn-Engineering; Modelle: StarCoder-Base 15B-CodeLlama 13B                                                                  | 100 % syntaktische Validität bei aktiviertem Post-Processing; CodeLlama 13B erreicht 91 % semantische Korrektheit, StarCoderBase $\leq 75$ %; Ergebnis steigt mit Modellgröße; Post-Processing verringert JSON-Fehler                                                                                                                                                                                                                                                                        |
| Pan et al. (2025)     | Ableitung von Test-szenarien                                                    | ASTER: statische Code-Analyse→LLM-Prompt→Test-Repair→Coverage-Augmentation. Sechs LLMs (GPT-4-turbo, Llama3-8/70 B, Code/Granite-34/8B) auf 4 Java-SE, 4 Java-EE & 1 Python-Suite; Vergleich EvoSuite/CodaMosa; 161-Entwickler-Survey zu Test-Naturalness                                                                                                                                                                                                         | ASTER gegenüber CodaMosa im Schnitt +9,8 pp Line- und +26,5 pp Branch-Coverage (sowie +22,5 pp Method-Coverage); > 70 % (Java) bzw. 88 % (Python) der Devs würden ASTER-Tests direkt übernehmen; Mock-Support ermöglicht 4–5× höhere Coverage auf XPath                                                                                                                                                                                                                                      |
| Rahman & Zhu (2024)   | Domänenspezifische Anforderungsherleitung/ Formalisierungsgrad der Requirements | Experimentelle Nutzerstudie mit Expertenbewertung: GPT-4 wandelt Anforderungsdokumente sowohl in User-Stories als auch Akzeptanztest-Spezifikationen um; führt <i>Refine-and-Thought (RaT)</i> -Prompting ein (zweistufige Bereinigung + Reasoning); 50 professionelle Entwickler bewerteten die Ergebnisse mit der RUST-Rubrik                                                                                                                                   | RaT reduziert Halluzinationen gegenüber einfachem CoT-Prompting deutlich; Durchschnittlicher Erfolgswert auf RUST = $\approx 88$ %; technische Auslassungen nur 5 %; JSON-Ausgabe lässt sich direkt in PM-Tools importieren.                                                                                                                                                                                                                                                                 |
| Ronanki et al. (2024) | Formalisierungsgrad der Requirements                                            | Vergleichsstudie mit quantitativer Analyse und menschlicher Validierung: ChatGPT (GPT-3.5) bewertet vorhandene Stories anhand von drei QUS-Kriterien; Vergleich mit AQUA-Tool & menschlichen Gutachtern; Drei Durchläufe $\times$ „One-Shot“-Prompt; zusätzlich „Best-of-3“-Voting getestet                                                                                                                                                                       | Rohkonsistenz nur 61 % – ChatGPT widerspricht stellenweise seinen eigenen Urteilen; Bei 2-von-3 innerer Übereinstimmung liegt F1 $\approx 90$ % und deckt sich zu 87 % mit menschlichen Ratings; Einfaches „Best-of-3“-Ensemble erhöht Präzision & Stabilität                                                                                                                                                                                                                                |
| Santos et al. (2025)  | Formalisierungsgrad der Requirements                                            | Experiment mit Studierenden & Prompt-Engineering: 30 Informatik-Studierende verfassten Stories für eine Reiseagentur-App; Zwei Phasen: (1) manuell, (2) GPT-3.5 mit freien Prompts; Qualität mit 7 QUS-Kriterien bewertet; Aufbau eines Meta-Few-Shot-Prompts (strukturiert & Beispiele) und erneute Generierung                                                                                                                                                  | Freies GPT $\approx$ manuell - kein signifikanter Unterschied ( $\emptyset$ -QUS $\approx 64$ %); Meta-Few-Shot-Prompt springt auf $\approx 89$ % Erfolgsrate und liefert konsistentere Stories; Größte Verbesserungen bei „minimal“ & „konzeptuell stimmig“.                                                                                                                                                                                                                                |
| Sarsa et al. (2022)   | Ableitung von Test-szenarien                                                    | Automatisierte Generierung & quantitative Inhaltsanalyse: Codex (GPT-3-basis) erzeugt komplette Programmieraufgaben - Problemstatement, Musterlösung, Unit-Tests - sowie Schritt-für-Schritt-Codeerklärungen; 240 generierte Aufgaben quantitativ & qualitativ ausgewertet (Sensibilität, Neuartigkeit, Lauffähigkeit, Testabdeckung); Zusätzlich 20 Codeerklärungen analysiert; Priming variierte Kontext- und Programmierschlüsselwörter, Temperatur 0 vs. 0.75 | 75 % der Aufgaben sinnvoll, 82 % neuartig; $\approx 90$ % der Musterlösungen kompilieren, aber nur 31 % bestehen die autogenerierten Tests; Wenn Tests bestehen, decken sie im Mittel 98 % der Statements ab (48 von 51 Fällen sogar 100 %); Kontext- und Konzept-Stichworte werden von Codex überraschend zuverlässig in neue, thematisch passende Aufgaben eingebaut; Bei 20 Codeerklärungen waren alle Codeteile zu 90 % abgedeckt, aber nur $\approx 67$ % der Zeilen inhaltlich korrekt |
| Sherifi et al. (2024) | Ableitung von Test-szenarien                                                    | Technische Fallstudie & Performance-Messung: Multi-Agent-Framework mit GPT-4/Gemini zum vollautomatischen Erzeugen, Ausführen und Berichten von Unit-Tests (inkl. DOT-Call-Graph & PDF-Report); Fallstudien: 2 Python- und 2 Java-Anwendungen (47 – 269 LoC); 20 Python-Runs & 24 Java-Runs (verschiedene Promptvarianten) gemessen: Zeiten, Erfolg, Coverage                                                                                                     | Alle Python-Durchläufe erfolgreich; in Java nur 3 Fehlschläge (12,5 %), meist wegen unklarer Prompts oder Kompilierfehlern in generierten Tests; Durchschnittliche End-to-End-Dauer: $\approx 80$ s (Python) / $\approx 87$ s (Java); davon > 60 s reine Test-Generierung; Sehr hohe Testabdeckung: 97,7 % (Java) bzw. 93,5 % (Python); Einzelfälle bis 100 %; Zusatzauswertung mit ChatGPT Web ergab ähnliche Coverage (98 %), aber doppelte Generierungszeit                               |
| Wang et al. (2024)    | Formalisierungsgrad der Requirements                                            | Explorative Klassenstudie mit 45 SE-Studierenden: Erstellung von Use Case-, Klassen- &                                                                                                                                                                                                                                                                                                                                                                            | LLMs halfen v.a. bei Akteuren & Klassenbeziehungen; modellweite Genauigkeit 59–75 %,                                                                                                                                                                                                                                                                                                                                                                                                         |

|                    |                              |                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                              |
|--------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    |                              | Sequenzdiagrammen für ein Order-Processing-System mittels ChatGPT; vollständige Prompt-Logs; Bewertung nach 11 struktur-/semantikbasierten Kriterien pro Modelltyp                                                                                                                                                                               | mit Defiziten v.a. bei Beziehungen; Output-Format (z.B. PlantUML vs. Hybrid) beeinflusst Bewertung signifikant; Limitationen: Novizenbias, subjektive Bewertung, stochastische Ausgaben                                                                                                                                                                      |
| Yang et al. (2024) | Ableitung von Test-szenarien | Empirische Vergleichsstudie auf 17 Defects4J-Java-Projekten; Fünf Open-Source-Code-LLMs (CodeLlama 7/13B, Phind-34B, DeepSeek-6.7/33B) wurden mit variierenden Prompt-Designs & In-Context-Learning verglichen; Referenzen: GPT-4 und EvoSuite; Kennzahlen: syntaktische Validität, Line- & Branch-Coverage, Defect-Detection; >3000 GPU-Stunden | Prompt-Gestaltung erwies sich als entscheidend: beste Designs steigerten Line-Coverage um bis zu +9,26pp; Größere Modelle schnitten besser ab, GPT-4 blieb führend, alle LLMs unterlagen jedoch EvoSuite (Line-Coverage 78.9 % vs. 40.4 %); 34 – 62 % der generierten Tests waren syntaktisch ungültig; CoT- und RAG-Varianten brachten keine Verbesserungen |
| Yang et al. (2025) | Ableitung von Test-szenarien | Systematische Literaturrecherche nach Kitchenham: 267 Publikationen (1994-2024) aus fünf Digital-Libraries, Snowballing; Klassifikation nach Requirement-Typ, Generierungsansatz, Tool-Support, Evaluation, Domäne                                                                                                                               | Trendwechsel von formalen/Modell-basierten zu NL-basierten Verfahren; identifiziert offene Felder: Transformation ATC→CTC, Non-Functional-RBTG, industriennahe Benchmarks und nutzbare Werkzeuge; Schlägt Forschung zu modulzentrierter Test-Priorisierung und leichtgewichtigen RBTG-Tools vor                                                              |

Tabelle 1: Übersicht einbezogener Quellen für Vorgehensweisen und Ergebnisse, Quelle: Eigene Darstellung:

Bem.: Die vorliegende Analyse wurde im Rahmen eines Vorlaufprojekts zu möglichen KI-Einsatzszenarien im Software Engineerings erstellt. Ziel war es den aktuellen Stand des KI-Einsatzes im Requirements Engineering zu erfassen.

# KI in der frühen Softwarekonzeption: Ein praxisnaher Leitfaden für Lösungsskizzen

Kokulan Thanabalan, Axel Kalenborn

Universität Trier, Wirtschaftsinformatik

thanabalan@uni-trier.de, axel.kalenborn@uni-trier.de

**Abstract:** Künstliche Intelligenz (KI), insbesondere generative Modelle wie Large Language Models (LLM), gewinnt in der Softwareentwicklung zunehmend an Bedeutung. Während KI in Entwicklungs- und Managementphasen bereits breit genutzt wird, bleibt ihr Potenzial in frühen Projektphasen wie Angebotserstellung und Lösungskonzeption häufig unsystematisch oder ungenutzt. Dieses Paper untersucht daher, wie Teams bereits in der Vorvertragsphase nachvollziehbar entscheiden können, ob und wie KI in eine Lösungsskizze integriert werden sollte. Dazu werden typische Einsatzfelder wie Mustererkennung, Klassifikation, Tracking und Textgenerierung mit konkreten Praxissituationen verknüpft. Betrachtet werden insbesondere die Strukturierung externer und unstrukturierter Daten, die Unterstützung bei der Datenerfassung sowie die Prüfung komplexer geschäftlicher Dokumente. Die Beispiele zeigen, dass KI vor allem dort Mehrwert bietet, wo sie menschliche Arbeit gezielt unterstützt, Fehler reduziert und Prozesse effizienter gestaltet. Auf dieser Grundlage entwickelt der Beitrag einen kompakten Entscheidungsrahmen, der fachlichen Nutzen, Umsetzbarkeit, Kontrollbedarf und Risiken verbindet. Ziel ist es, KI nicht nur als Werkzeug für Projektteams, sondern als bewusst geplanten Bestandteil moderner Business-Software und zukunftsfähiger Lösungsarchitekturen zu verstehen.

**Keywords:** Large Language Model, Künstliche Intelligenz, Software Engineering, Entscheidungsrahmen, KI-Integration

## 1 Einleitung

Künstliche Intelligenz (KI) hat sich in den letzten Jahren zu einer Schlüsseltechnologie in der Softwareentwicklung entwickelt. Insbesondere generative KI und maschinelles Lernen werden zunehmend für Aufgaben wie Codegenerierung, Testfallerstellung, Anforderungsanalyse oder Projektmanagement eingesetzt [Br20]. Eine zentrale technologische Grundlage vieler aktueller Anwendungen bilden Large Language Models (LLM), die natürliche Sprache auf hohem Niveau verarbeiten und generieren können [Sh23]. Werkzeuge wie ChatGPT, Gemini, GitHub Copilot oder KI-gestützte Assistenzsysteme unterstützen Entwicklerinnen und Entwickler sowie Projektverantwortliche bereits heute bei der effizienten Bewältigung komplexer Aufgaben.

Auch im Management von Softwareprojekten nimmt der Einsatz von KI-Technologien zu. KI-basierte Werkzeuge können beispielsweise bei der Strukturierung von Arbeitspaketen, der Risikobewertung oder der Aufwandsschätzung unterstützen [La20], [Lo24]. Damit wird KI zunehmend nicht nur als technisches Hilfsmittel in der Implementierung, sondern auch als unterstützende Technologie für Planung, Steuerung und Organisation von Softwareprojekten verstanden.

Während KI in Entwicklungs- und Managementphasen bereits vielfältig genutzt wird, bleibt ihr Potenzial in frühen Projektphasen häufig unzureichend ausgeschöpft. Gerade in der Vorvertragsphase, bei der Angebotserstellung und in der ersten Lösungskonzeption werden jedoch zentrale Weichen gestellt. Bereits hier entsteht ein Bild davon, wie ein fachliches

Problem technisch gelöst werden soll, welche Systembestandteile vorgesehen sind und welche Umsetzung als realistisch gilt. Diese frühen Entwürfe bilden eine Brücke zwischen Anforderungen und späterer Implementierung und enthalten häufig bereits grundlegende technologische Entscheidungen [Zh23b].

In der Praxis werden Angebote und Lösungsskizzen jedoch oft noch traditionell gedacht. Systeme werden vor allem regelbasiert oder algorithmisch entworfen, während KI-Potenziale entweder nicht berücksichtigt oder nur allgemein als mögliche Ergänzung erwähnt werden. Dadurch entstehen zwei gegensätzliche Probleme: Einerseits werden sinnvolle Einsatzmöglichkeiten zu spät erkannt und müssen später aufwendig ergänzt werden. Andererseits wird KI teilweise zu früh und zu unkonkret versprochen, ohne dass Aufgabe, Datenbasis und Qualitätssicherung klar beschrieben sind. Beides erschwert belastbare Entscheidungen in der Vorvertrags- und Planungsphase.

Für Projektverantwortliche, Vertriebsmitarbeitende sowie Softwarearchitektinnen und Softwarearchitekten entsteht damit eine neue Anforderung. KI muss nicht nur als technologische Option verstanden werden, sondern als konzeptioneller Baustein, dessen Einsatz früh begründet, eingegrenzt und kommuniziert werden kann. Benötigt wird daher ein Vorgehen, das Projektbeteiligte dabei unterstützt, typische Einsatzmuster zu erkennen, geeignete Aufgaben für KI abzugrenzen und Risiken bereits in der Lösungsskizze mitzudenken. Eine frühe KI-Konzeption sollte daher nicht nur beschreiben, was automatisiert oder unterstützt werden soll, sondern auch, wo menschliche Kontrolle erforderlich bleibt und welche technischen, organisatorischen oder rechtlichen Voraussetzungen zu berücksichtigen sind.

Im Mittelpunkt des Beitrags steht die Frage, wie sich der mögliche KI-Einsatz in frühen Projektphasen systematisch bewerten lässt. Dafür werden fachliche Problemstellungen herausgearbeitet, bei denen KI einen sinnvollen Beitrag leisten kann. Entscheidend ist dabei nicht die Technologie selbst, sondern die Frage, ob ein fachliches Problem durch KI besser, effizienter oder nutzerfreundlicher gelöst werden kann als durch rein klassische Verfahren. Ergänzend werden zentrale Risiken betrachtet, die bereits in der Lösungskonzeption berücksichtigt werden müssen. Aus der Verbindung von fachlicher Eignung und Risikobetrachtung wird ein praxisnaher Entscheidungsrahmen abgeleitet. Dieser soll Teams dabei unterstützen, in der Vorvertrags- und Konzeptionsphase nachvollziehbar zu klären, ob KI in eine Lösungsskizze aufgenommen werden sollte, welche Rolle sie im Prozess übernimmt und welche Voraussetzungen dafür erfüllt sein müssen. Ziel ist keine vollständige technische Architektur eines KI-Systems, sondern eine konzeptionelle Vorstrukturierung für Angebote und frühe Lösungsbeschreibungen.

Methodisch orientiert sich der Beitrag am Design-Science-Ansatz [He04], [VK07]. Im Sinne dieses Ansatzes wird zunächst ein praxisrelevantes Problem identifiziert: die bislang häufig unsystematische Berücksichtigung von KI in frühen Phasen der Softwarekonzeption. Aufbauend auf den in [Th25] diskutierten Grundüberlegungen werden wiederkehrende Anwendungssituationen aus der Praxis analysiert und zur Systematisierung früher KI-Entscheidungen genutzt. Daraus wird ein Artefakt in Form eines konzeptionellen Entscheidungsrahmens abgeleitet, der Projektteams dabei unterstützt, KI-Potenziale in Lösungsskizzen nachvollziehbar zu bewerten und risikobewusst zu beschreiben.

Der Beitrag ist wie folgt aufgebaut: Abschnitt 2 beschreibt den Problemkontext der frühen Softwarekonzeption und verdeutlicht die Bedeutung von Angeboten und Lösungsskizzen für die spätere Ausgestaltung von Softwareprojekten. Abschnitt 3 analysiert drei typische Anwendungssituationen aus kaufmännischen Softwareprojekten, um wiederkehrende Problemklassen für einen möglichen KI-Einsatz herauszuarbeiten. Abschnitt 4 überführt diese Erkenntnisse in das zentrale Artefakt des Beitrags: einen Entscheidungsrahmen für die frühe KI-Konzeption, der fachliche Einsatzmöglichkeiten mit einer strukturierten Betrachtung zentraler Risiken verbindet.

## **2 Von der Anforderung zur (KI-)Lösung**

Angebote und Lösungsskizzen haben in frühen Phasen von Softwareprojekten eine zentrale Kommunikations- und Planungsfunktion. Sie halten erste technische Ideen fest, beschreiben mögliche Lösungswege und schaffen ein gemeinsames Verständnis zwischen Fachbereich, Projektmanagement, Vertrieb und technischer Umsetzung. In der Praxis stehen dabei meist verständliche Darstellungen im Vordergrund, etwa Prozessbeschreibungen, Use Cases, Mockups oder erläuternde Texte [Ka14].

Die Auseinandersetzung mit einer Softwarelösung beginnt häufig bereits in der Vorvertragsphase. Auftraggeber erwarten in Angeboten nicht nur eine Kostenschätzung, sondern auch eine nachvollziehbare Darstellung, wie ihre Anforderungen verstanden wurden und welche Lösungsrichtung vorgeschlagen wird. Damit enthalten Angebote oft bereits erste Lösungsskizzen zu Abläufen, Systemideen, Schnittstellen oder technologischen Ansätzen.

Ob ein Prozess klassisch regelbasiert, manuell unterstützt oder durch KI ergänzt wird, beeinflusst die spätere Architektur und die Wahrnehmung der Lösung. Wird KI erst spät berücksichtigt, bleiben Potenziale zur Automatisierung, Assistenz oder Qualitätsverbesserung möglicherweise ungenutzt. Wird KI hingegen zu früh und unkonkret versprochen, entstehen Risiken, wenn Aufgabe, Datenbasis und Kontrolle der Ergebnisse nicht klar beschrieben sind.

Für die frühe Softwarekonzeption bedeutet dies, dass KI nicht als allgemeines Schlagwort in eine Lösungsskizze aufgenommen werden sollte. Stattdessen muss geprüft werden, ob eine intelligente Komponente für das konkrete fachliche Problem einen erkennbaren Mehrwert bietet. Eine KI-gestützte Lösungsskizze sollte daher beschreiben, welche Rolle KI im Prozess übernimmt.

Die frühe KI-Konzeption erweitert damit die klassische Lösungsskizze um zentrale Prüfungen: Welche Daten liegen vor? Welcher Automatisierungsgrad ist sinnvoll? Wo ist menschliche Kontrolle erforderlich? Welche rechtlichen, organisatorischen oder technischen Risiken bestehen? Ziel ist keine vollständige technische Architektur, sondern eine belastbare Vorstrukturierung der Lösung.

## **3 Drei typische Situationen aus der Praxis**

Die Frage, ob KI bereits in frühen Phasen eines Softwareprojekts berücksichtigt werden sollte, stellt sich in der Praxis meist nicht abstrakt. Projektteams stehen in der Regel nicht vor der allgemeinen Aufgabe, „KI einzubauen“, sondern vor konkreten fachlichen Herausforderungen, die effizient, wirtschaftlich und technisch tragfähig gelöst werden müssen. Gerade in der Vorvertragsphase ist es daher sinnvoll, nicht von einzelnen Technologien auszugehen, sondern von wiederkehrenden Problemklassen, bei denen KI einen erkennbaren Beitrag leisten kann.

Solche Problemklassen treten in vielen Unternehmensanwendungen in ähnlicher Form auf. Häufig müssen Informationen aus unterschiedlichen und teilweise unstrukturierten Quellen erschlossen, Daten aus freien Eingaben in strukturierte Systeme überführt oder komplexe Inhalte auf Auffälligkeiten geprüft werden. In diesen Situationen kann KI dazu beitragen, manuelle Aufwände zu reduzieren, Informationen schneller nutzbar zu machen und Nutzende bei wiederkehrenden oder fehleranfälligen Tätigkeiten zu unterstützen.

Die folgenden drei Anwendungssituationen stehen exemplarisch für solche typischen Herausforderungen [Th25]. Sie bilden keine vollständige Übersicht aller möglichen KI-Einsatzfelder, sondern eine praxisnahe Auswahl wiederkehrender Problemstellungen, die sich besonders gut für eine frühe Betrachtung in Angeboten und Lösungsskizzen eignen. Betrachtet werden die Strukturierung externer und unstrukturierter Daten, die Unterstützung bei der Datenerfassung sowie die Prüfung komplexer geschäftlicher Dokumente. Anhand dieser

Situationen lässt sich zeigen, wie KI nicht als isolierte Technologie, sondern als gezielt eingebetteter Bestandteil einer fachlichen Lösung konzipiert werden kann.

### **3.1 Strukturierung externer und unstrukturierter Daten**

Die Integration externer Datenquellen stellt Unternehmen häufig vor die Herausforderung, Informationen aus unstrukturierten oder semi-strukturierten Formaten in die eigene Systemlandschaft zu überführen. Relevante Daten liegen dabei nicht immer als standardisierte Datensätze vor, sondern beispielsweise in PDF-Dokumenten, E-Mails, Tabellen, Listen oder Freitexten. Besonders aufwendig wird dies, wenn diese Informationen regelmäßig manuell geprüft, übertragen, verglichen oder aktualisiert werden müssen.

KI kann in solchen Situationen eingesetzt werden, um unstrukturierte Inhalte zu analysieren, relevante Informationen zu extrahieren und in eine strukturierte Form zu überführen. Eine praktische Fallstudie zeigt beispielsweise, dass LLMs in der Lage sind, Inhalte aus unstrukturierten Datenquellen zu erfassen und tabellarisch aufzubereiten [Th23]. Damit eignet sich KI insbesondere für Aufgaben, bei denen Informationen zwar fachlich klar identifizierbar sind, ihre manuelle Aufbereitung jedoch zeitintensiv, fehleranfällig und schlecht skalierbar ist.

Ein typischer Anwendungsfall ist der Abgleich externer Preis- oder Produktinformationen mit Daten in einem kaufmännischen System. Müssen Mitarbeitende Preislisten aus PDF-Dokumenten manuell auslesen, mit vorhandenen Produktdaten vergleichen und anschließend aktualisieren, entsteht ein hoher wiederkehrender Aufwand. KI kann hier unterstützen, indem sie Produkte, Preise oder weitere relevante Angaben automatisiert erkennt und für die Weiterverarbeitung strukturiert bereitstellt. Dadurch lassen sich Bearbeitungszeiten reduzieren, Übertragungsfehler vermeiden und Daten schneller aktualisieren.

KI übernimmt hier keine abschließende fachliche Entscheidung, sondern unterstützt die vorgelagerte Informationsaufbereitung. Es sollte bereits in der Lösungsskizze festgehalten werden, aus welchen Quellen die Daten stammen, welche Informationen extrahiert werden sollen, wie stabil die Eingabeformate sind und an welchen Stellen eine fachliche Prüfung erforderlich bleibt. So wird KI nicht als isolierte Technologie beschrieben, sondern als gezielt eingebetteter Prozessbaustein zur effizienteren Datenintegration.

### **3.2 Unterstützung bei der Datenerfassung**

Die manuelle Erfassung von Daten über Formulare zählt in vielen Unternehmen zu den besonders zeitaufwändigen und fehleranfälligen Aufgaben. Prozesse wie das Anlegen neuer Kunden, Produkte oder Vorgänge erfordern häufig das Ausfüllen zahlreicher Formularfelder. Die dafür benötigten Informationen liegen jedoch in der Praxis oft nicht bereits strukturiert vor, sondern werden beispielsweise per E-Mail, als Notiz, Gesprächsprotokoll oder Freitext übermittelt. Mitarbeitende müssen diese Inhalte anschließend manuell lesen, interpretieren und in die passenden Felder eines Fachsystems übertragen.

An dieser Stelle können KI-Systeme wirkungsvoll unterstützen. Sie können unstrukturierte Texte analysieren, relevante Informationen erkennen und diese automatisch geeigneten Formularfeldern zuordnen [Kl24]. Die Nutzerinnen und Nutzer müssen dadurch nicht mehr jedes Feld einzeln ausfüllen, sondern können einen zusammenhängenden Textblock bereitstellen. Die KI übernimmt im Hintergrund die Strukturierung und erzeugt Vorschläge für die Befüllung der Eingabemaske.

Ein typisches Beispiel ist die Anlage eines neuen Kunden. Statt Name, Adresse, Ansprechpartner, Telefonnummer oder weitere Angaben einzeln aus einer E-Mail zu übertragen, kann der gesamte Nachrichtentext an das System übergeben werden. Die KI erkennt die enthaltenen Informationen und ordnet sie den vorgesehenen Feldern zu. Dadurch wird die

Datenerfassung beschleunigt, der manuelle Aufwand reduziert und die Wahrscheinlichkeit von Übertragungsfehlern verringert.

Ergänzend kann KI auch Spracheingaben unterstützen. Dabei wird gesprochene Sprache zunächst in Text umgewandelt und anschließend in strukturierte Formularinformationen überführt. Dies kann besonders hilfreich sein, wenn eine Tastatureingabe nicht praktikabel ist oder wenn mobile Arbeitssituationen vorliegen. Zugleich kann eine solche Funktion die Barrierefreiheit verbessern, da digitale Formulare für Menschen mit körperlichen Einschränkungen leichter nutzbar werden.

Für frühe Lösungsskizzen ist diese Anwendungssituation besonders anschaulich, weil der Nutzen direkt aus der Arbeitspraxis ableitbar ist. KI verändert hier vor allem die Interaktion mit dem System: Nicht die Nutzenden müssen ihre Eingaben vollständig an die Systemstruktur anpassen, sondern das System unterstützt sie bei der Überführung natürlicher Sprache in strukturierte Daten. Wichtig bleibt jedoch, dass automatisch befüllte Felder überprüfbar sind und bei Bedarf korrigiert werden können. KI sollte daher als Assistenzfunktion konzipiert werden, die Eingaben vorbereitet, aber die fachliche Kontrolle nicht vollständig ersetzt, solange die Korrektheit nicht garantiert werden kann.

The image shows two side-by-side windows. The left window, titled 'Allgemeine Informationen', is a data entry form with several fields: 'Meldung als' (dropdown), 'Datum' (calendar icon), 'Fluggelände' (text input), 'Land' (dropdown), 'Flugausrüstung' (dropdown), 'Zeit' (time input), 'Ort' (text input), and 'Art des Fluges' (dropdown). The right window, titled 'Formular Assistent', is a chat interface. It contains a message: 'Hallo, ich bin Ihr KI-Assistent. Wie kann ich Ihnen weiterhelfen?'. At the bottom, there is a text input field labeled 'Nachricht' and a microphone icon.

Abbildung 1: Datenerfassung mit KI-Unterstützung [Th25]

### 3.3 Prüfung komplexer geschäftlicher Dokumente

Die Erstellung und Prüfung geschäftlicher Dokumente gehört in vielen Unternehmen zu den zentralen wiederkehrenden Aufgaben. Dazu zählen beispielsweise Rechnungen, Verträge, Anträge, Nachweise oder andere geschäftskritische Unterlagen. Besonders anspruchsvoll werden solche Prüfprozesse, wenn unterschiedliche gesetzliche Vorgaben, steuerliche Regelungen, länderspezifische Besonderheiten oder interne Richtlinien berücksichtigt werden müssen. Gerade bei internationalen Geschäftsvorfällen können sich Anforderungen an Steuersätze, Pflichtangaben, Währungsangaben oder formale Bestandteile je nach Kontext deutlich unterscheiden.

In solchen Situationen kann KI zur Qualitätssicherung eingesetzt werden. LLMs können fertig erstellte Dokumente analysieren, relevante Inhalte erkennen und potenzielle Abweichungen markieren. Bei Rechnungen kann dies beispielsweise fehlerhafte Steuersätze, fehlende Pflichtangaben, unvollständige Adressdaten oder widersprüchliche Währungsinformationen betreffen. Die KI übernimmt dabei nicht die abschließende rechtliche oder fachliche Bewertung, sondern unterstützt die Prüfung, indem sie Hinweise erzeugt und mögliche Korrekturen vorschlägt.

Der Nutzen liegt vor allem in der Entlastung der Mitarbeitenden bei wiederkehrenden Kontrollaufgaben. Neue oder weniger erfahrene Beschäftigte können durch KI-gestützte Hinweise unterstützt werden, während Expertinnen und Experten mehr Zeit für komplexe Sonderfälle erhalten. Gleichzeitig kann die Fehleranfälligkeit reduziert werden, da Dokumente systematisch auf typische Auffälligkeiten geprüft werden. Dies ist besonders relevant in

Prozessen, in denen formale Fehler zu Rückfragen, Verzögerungen oder finanziellen Nachteilen führen können.

Auch wirtschaftlich kann der Einsatz von KI in solchen Prüfprozessen attraktiv sein. Die automatisierte Vorprüfung vieler Dokumente verursacht im Vergleich zu einer vollständig manuellen Kontrolle häufig geringe variable Kosten. Bei der Nutzung leistungsstarker Modelle entstehen Kosten typischerweise abhängig von der Anzahl der verarbeiteten Eingabe- und Ausgab tokens. So können auch größere Mengen einfacher oder mittelkomplexer Dokumente kosteneffizient vorgeprüft werden. Beispielsweise lassen sich bei kurzen Dokumenten mit wenigen hundert Eingabetoken und einer begrenzten KI-Ausgabe mit vergleichsweise geringen Modellkosten zahlreiche Prüfungen automatisiert durchführen [Op25a], [Op25b]. Dadurch können Unternehmen nicht nur Bearbeitungszeiten verkürzen, sondern auch Fachkräfte gezielter für die finale Bewertung oder schwierige Ausnahmefälle einsetzen.

Für frühe Lösungsskizzen ist diese Anwendungssituation jedoch besonders sorgfältig zu beschreiben. Der mögliche Mehrwert ist hoch, zugleich steigen die Anforderungen an Nachvollziehbarkeit, Kontrolle und Verantwortungszuweisung. Eine belastbare Lösungsskizze sollte daher nicht den Eindruck erwecken, dass KI geschäftskritische Entscheidungen vollständig ersetzt. Sinnvoller ist die Beschreibung einer intelligenten Prüfhilfe, die Auffälligkeiten sichtbar macht, Risiken priorisiert und fachliche Entscheidungen vorbereitet. Die endgültige Verantwortung verbleibt dort, wo rechtliche oder wirtschaftliche Folgen entstehen können, weiterhin beim Menschen.

## 4 Ein Entscheidungsrahmen für die frühe KI-Konzeption

Die bisherigen Ausführungen zeigen, dass KI in frühen Phasen von Softwareprojekten nicht pauschal als technologische Ergänzung verstanden werden sollte. Entscheidend ist vielmehr, ob ihr Einsatz für ein konkretes fachliches Problem geeignet ist und ob die damit verbundenen Risiken kontrollierbar bleiben. Für Angebote und Lösungsskizzen wird daher ein kompakter Entscheidungsrahmen vorgeschlagen, der Projektteams bei der frühen Bewertung möglicher KI-Bausteine unterstützt.

Zur Konkretisierung des Rahmens werden die zentralen Prüffragen in Tabelle 1 zusammengeführt. Die Tabelle dient als Orientierung für frühe Projektphasen und zeigt, welche Aspekte bei der Entscheidung über einen möglichen KI-Einsatz in einer Lösungsskizze berücksichtigt werden sollten. Die einzelnen Prüfdimensionen werden in den folgenden Abschnitten näher erläutert. Abschnitt 4.1 betrachtet zunächst die fachliche Problemklasse und damit die Frage, ob überhaupt ein geeignetes KI-Problem vorliegt. Abschnitt 4.2 ergänzt diese Bewertung um Risiken, Kontrollbedarf und technologische Rahmenbedingungen.

| Prüfdimension           | Leitfrage                            | Mögliche Bewertung                                                  | Konsequenz für die Lösungsskizze              |
|-------------------------|--------------------------------------|---------------------------------------------------------------------|-----------------------------------------------|
| Fachliche Problemklasse | Liegt ein geeignetes KI-Problem vor? | Extraktion, Klassifikation, Tracking, Modellierung, Textgenerierung | KI als möglichen Lösungsbausteine beschreiben |

| <b>Prüfdimension</b> | <b>Leitfrage</b>                                                                     | <b>Mögliche Bewertung</b>                                               | <b>Konsequenz für die Lösungsskizze</b>                         |
|----------------------|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------|-----------------------------------------------------------------|
| Datenlage            | Sind geeignete Daten verfügbar?                                                      | strukturiert, unstrukturiert, sensibel, instabil                        | Datenquellen, Datenqualität und Einschränkungen benennen        |
| Prozessrolle der KI  | Welche Rolle übernimmt KI im Prozess?                                                | Assistenz, Vorprüfung, Automatisierung, Empfehlung                      | Aufgabe der KI klar von menschlicher Verantwortung abgrenzen    |
| Kontrollbedarf       | Wie hoch ist der Bedarf an menschlicher Kontrolle?                                   | niedrig, mittel, hoch                                                   | Prüf-, Freigabe- oder Korrekturmöglichkeiten festlegen          |
| Risiken              | Welche fachlichen, rechtlichen, technischen oder organisatorischen Risiken bestehen? | Datenschutz, Halluzinationen, Nachvollziehbarkeit, Anbieterabhängigkeit | Schutzmaßnahmen, Monitoring und Fallback beschreiben            |
| Entscheidung         | Soll KI in die Lösungsskizze aufgenommen werden?                                     | einsetzen, weiter prüfen, nicht einsetzen                               | Begründete Aufnahme, Einschränkung oder Ablehnung dokumentieren |

Tabelle 1: Entscheidungsrahmen für die frühe KI-Konzeption

#### 4.1 Welches fachliche Problem soll gelöst werden?

Die erste Leitfrage betrifft die fachliche Eignung des Problems. Ausgangspunkt einer frühen KI-Konzeption sollte nicht die Frage sein, welche konkrete Technologie verwendet wird, sondern welche Aufgabe im geplanten System unterstützt oder automatisiert werden soll. KI ist besonders dann relevant, wenn klassische regelbasierte Verfahren an Grenzen stoßen, weil Informationen unstrukturiert vorliegen, Muster schwer explizit beschreibbar sind oder große Datenmengen verarbeitet werden müssen.

Zur systematischen Einordnung können vier in der Literatur etablierte Einsatzfelder herangezogen werden: Extraktion und Mustererkennung, Klassifikation, Tracking und Modellierung sowie Textgenerierung [Zh20], [SIM23]. Diese Kategorien helfen dabei, fachliche Anforderungen in typische KI-Aufgaben zu übersetzen.

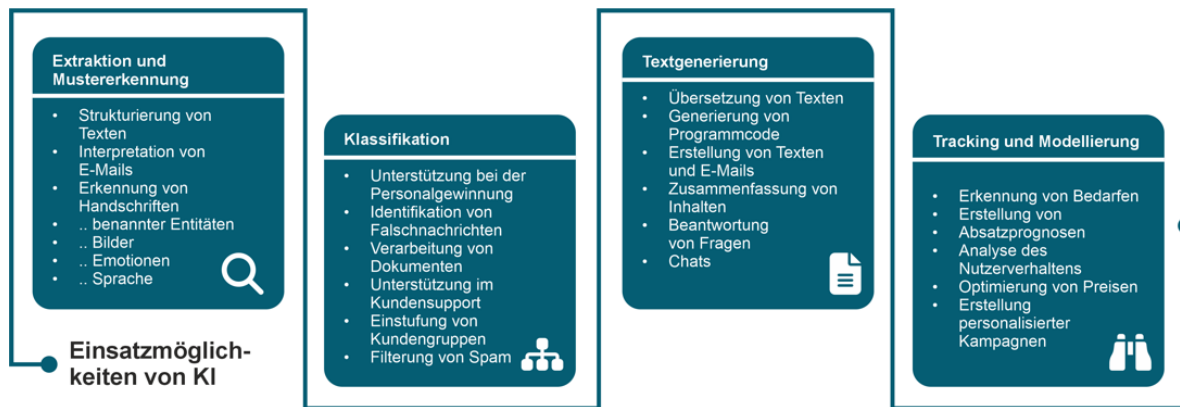


Abbildung 2: Einsatzmöglichkeiten von KI [Th25]

**Extraktion und Mustererkennung:** Ein erster Anwendungsbereich liegt in der Extraktion und Mustererkennung. Dabei geht es darum, relevante Informationen aus unstrukturierten oder semi-strukturierten Quellen zu erkennen und in eine weiterverarbeitbare Form zu überführen. Typische Datenquellen sind, wie das Praxisbeispiel verdeutlicht, PDF-Dokumente, E-Mails, Freitexte, Tabellen oder eingescannte Dokumente. Für frühe Lösungsskizzen ist dieses Einsatzfeld besonders geeignet, wenn wiederkehrend Informationen aus externen Quellen in interne Systeme übertragen werden müssen. Die KI übernimmt hier vor allem eine vorbereitende Funktion. Sie ersetzt nicht die fachliche Verantwortung, sondern reduziert manuelle Routinearbeit und macht Daten schneller nutzbar.

In einer Lösungsskizze sollte daher konkret beschrieben werden, welche Informationen extrahiert werden sollen, aus welchen Quellen sie stammen und in welcher Form sie dem System oder den Nutzenden anschließend bereitgestellt werden. So wird deutlich, dass KI nicht allgemein zur „Dokumentenverarbeitung“ eingesetzt wird, sondern eine klar abgegrenzte Aufgabe übernimmt.

**Klassifikation:** Ein zweites Einsatzfeld ist die Klassifikation. Darunter wird die Fähigkeit verstanden, Texte, Dokumente oder Vorgänge bestimmten Kategorien zuzuordnen. Diese Funktion ist besonders relevant, wenn eingehende Informationen sortiert, priorisiert oder unterschiedlichen Bearbeitungswegen zugewiesen werden müssen.

In Unternehmensanwendungen kann Klassifikation beispielsweise genutzt werden, um Kundenanfragen nach Themenbereichen zu ordnen, Risiken einzuschätzen, Dokumenttypen zu erkennen oder Vorgänge bestimmten Prozesspfaden zuzuweisen [PVB24]. Auch bei der Personalisierung digitaler Systeme kann Klassifikation eine Rolle spielen, wenn Nutzungsverhalten bestimmten Gruppen zugeordnet wird und Inhalte entsprechend angepasst werden.

Für die frühe KI-Konzeption ist entscheidend, ob die vorgesehenen Kategorien fachlich klar genug beschrieben werden können. Eine Lösungsskizze sollte daher festhalten, welche Klassen oder Zuordnungen benötigt werden und was mit dem Ergebnis der Klassifikation geschieht. Je stärker eine Klassifikation operative Konsequenzen hat, etwa eine automatische Eskalation oder Weiterleitung, desto wichtiger werden Kontroll- und Korrekturmöglichkeiten.

**Tracking und Modellierung:** Ein weiteres Einsatzfeld betrifft Tracking und Modellierung. Hier geht es um das kontinuierliche Erfassen, Analysieren und Auswerten von Informationen aus internen oder externen Datenströmen. Ziel ist es, Veränderungen, Muster oder Abweichungen frühzeitig sichtbar zu machen. Dies kann etwa in der Prozessüberwachung, bei der Analyse von Nutzerinteraktionen oder bei der Beobachtung dynamischer Geschäftsdaten relevant sein [WWS23].

Ergänzend dazu kann KI historische Daten modellieren, um Entwicklungen abzuleiten oder Prognosen zu unterstützen. Beispiele sind Vorhersagen zu Nachfrageverläufen, Durchlaufzeiten, finanziellen Kennzahlen oder Risiken [Fa23].

Die KI übernimmt in diesem Fall eine analytische Rolle. Sie erkennt Muster, weist auf Abweichungen hin oder unterstützt bei der Ableitung von Handlungsempfehlungen. Eine Lösungsskizze sollte jedoch vermeiden, solche Modelle als sichere Vorhersageinstrumente darzustellen. Stattdessen sollte beschrieben werden, dass sie Entscheidungsgrundlagen liefern und menschliche Bewertungen unterstützen.

**Textgenerierung:** Das vierte Einsatzfeld ist die Textgenerierung. Dabei erzeugen KI-Systeme neue Texte oder überführen bestehende Inhalte in eine andere sprachliche Form. Dazu gehören Zusammenfassungen, Übersetzungen, Formulierungsvorschläge, Berichte oder auch Codegenerierung [Zh23a], [Br20].

Für frühe Softwarekonzepte ist Textgenerierung besonders dann interessant, wenn Nutzende regelmäßig Informationen verdichten, formulieren oder dokumentieren müssen. Beispiele sind die automatische Zusammenfassung von E-Mail-Verläufen, die Verdichtung langer Vertragsdokumente auf zentrale Inhalte, die Erstellung von Antwortvorschlägen oder die Generierung technischer Textbausteine.

Gleichzeitig ist dieses Einsatzfeld besonders sorgfältig zu beschreiben, da generierte Texte sprachlich überzeugend wirken können, ohne fachlich korrekt zu sein. Eine Lösungsskizze sollte daher festlegen, ob generierte Inhalte automatisch übernommen werden dürfen oder ob sie als Vorschlag angezeigt und durch Nutzende freigegeben werden müssen. Gerade bei rechtlich, finanziell oder fachlich relevanten Texten sollte KI nicht als alleinige Autorität auftreten, sondern als Assistenzsystem.

## 4.2 Welche Risiken sprechen für oder gegen den Einsatz?

Neben der fachlichen Eignung muss in frühen Projektphasen geprüft werden, welche Risiken mit dem KI-Einsatz verbunden sind. Diese Risikoprüfung ist ein wesentlicher Bestandteil des Entscheidungsrahmens, weil sie verhindert, dass KI-Funktionen zu unkonkret oder zu optimistisch in Lösungsskizzen aufgenommen werden. Die Literatur unterscheidet dabei drei zentrale Risikobereiche: operationale Herausforderungen, soziale und rechtliche Risiken sowie technologische Abhängigkeiten [PVB24].

**Operationale Herausforderungen:** Operationale Herausforderungen betreffen die praktische Einbettung von KI in betriebliche Abläufe. Eine zentrale Schwierigkeit besteht in der eingeschränkten Nachvollziehbarkeit und Erklärbarkeit von LLM-Ergebnissen. Wenn ein Modell Informationen aus einem Dokument extrahiert, eine Zuordnung vornimmt oder einen Geschäftsvorfall bewertet, ist nicht immer transparent, warum genau dieses Ergebnis erzeugt wurde. Dies erschwert die Validierung und kann besonders in sensiblen Bereichen problematisch sein.

Ein Beispiel ist die automatisierte Prüfung von Rechnungen. Wenn eine KI einen Steuersatz als fehlerhaft markiert oder eine Pflichtangabe übersieht, muss nachvollziehbar sein, wie mit diesem Hinweis umzugehen ist. Ohne geeignete Validierung besteht die Gefahr, dass falsche Ergebnisse in den Prozess übernommen werden oder dass berechtigte Hinweise nicht ernst genommen werden.

Hinzu kommen wirtschaftliche und organisatorische Faktoren. Die Verarbeitung großer Datenmengen und häufige Anfragen an cloudbasierte KI-Dienste können Kosten verursachen. Außerdem muss geklärt werden, wie KI-Ergebnisse protokolliert, überwacht und durch Nutzerfeedback verbessert werden können. Für die frühe Lösungsskizze bedeutet dies, dass KI-Bausteine nicht isoliert beschrieben werden sollten. Es muss auch erkennbar sein, wie sie kontrolliert, überwacht und in bestehende Prozesse eingebunden werden.

**Soziale und rechtliche Risiken:** Ein zweiter Risikobereich betrifft soziale und rechtliche Aspekte. Besonders relevant ist dabei das Risiko sogenannter Halluzinationen. LLMs können Inhalte erzeugen, die sprachlich plausibel wirken, aber sachlich falsch sind [Mi24]. In betrieblichen Anwendungen kann dies zu fehlerhaften Informationen, falschen Schlussfolgerungen oder Desinformation führen.

Dieses Risiko ist vor allem dann kritisch, wenn KI-Ergebnisse ohne nachgelagerte Prüfung in operative Prozesse einfließen. Deshalb sollte bereits in der Lösungsskizze festgelegt werden, ob eine menschliche Kontrolle erforderlich ist und an welchen Stellen Ergebnisse bestätigt, korrigiert oder verworfen werden können.

Darüber hinaus spielen Datenschutz und rechtliche Vorgaben eine zentrale Rolle. Wenn personenbezogene Daten, Kundendaten oder vertrauliche Geschäftsinformationen verarbeitet werden, muss geklärt werden, ob diese Daten an externe KI-Dienste übermittelt werden dürfen. Bei der Verarbeitung von Freitexten, etwa im Rahmen von Personendaten, können sensible Informationen enthalten sein. Es muss daher sichergestellt werden, dass diese nicht dauerhaft gespeichert oder unrechtmäßig weitergegeben werden. In diesem Zusammenhang sind auch regulatorische Anforderungen wie der AI Act der Europäischen Union zu berücksichtigen [LTD24].

Eine mögliche Maßnahme zur Risikoreduktion besteht darin, LLMs lokal oder innerhalb einer kontrollierten Unternehmensinfrastruktur zu betreiben. Dadurch kann verhindert werden, dass sensible Daten das Unternehmen verlassen. Für die frühe KI-Konzeption bedeutet dies, dass nicht nur die Funktion der KI beschrieben werden sollte, sondern auch das vorgesehene Betriebsmodell.

**Technologische Abhängigkeiten:** Der dritte Risikobereich betrifft technologische Abhängigkeiten. Der Einsatz externer KI-Dienste führt dazu, dass Anwendungen von proprietären Modellen, Schnittstellen, Lizenzbedingungen und Verfügbarkeiten abhängig werden können [PVB24]. Dies ist besonders relevant, wenn LLMs über Programmierschnittstellen von Anbietern wie OpenAI oder Google in eine Systemlandschaft eingebunden werden.

Änderungen an Modellen, Preisen, Nutzungsbedingungen oder Zugriffsbeschränkungen können direkte Auswirkungen auf produktive Anwendungen haben. Ebenso können Systemausfälle oder temporäre Inkompatibilitäten dazu führen, dass automatisierte Abläufe unterbrochen werden. Dieses Risiko ist vor allem dann kritisch, wenn geschäftskritische Prozesse dauerhaft auf externe KI-Dienste angewiesen sind.

Für frühe Lösungsskizzen folgt daraus, dass Abhängigkeiten nicht verschwiegen werden sollten. Stattdessen sollte beschrieben werden, wie Ausfallsicherheit, Monitoring und Fehlerbehandlung berücksichtigt werden können. Dazu gehören beispielsweise funktionale Redundanzen, alternative Prozesswege oder ein manueller Fallback. Eine KI-gestützte Lösung ist nur dann belastbar, wenn sie auch bei eingeschränkter Verfügbarkeit der KI-Komponente kontrolliert weiterbetrieben werden kann.

## 5 Fazit

Der Beitrag zeigt, dass KI in frühen Projektphasen nicht als bloßes Schlagwort, sondern als bewusst zu prüfender Lösungsbaustein verstanden werden sollte. Da die Integration von KI in Business-Software zunehmend an Bedeutung gewinnt, sollten ihre Potenziale bereits zu Beginn der Softwarekonzeption berücksichtigt werden.

Dazu müssen geplante Funktionen mit typischen KI-Einsatzfeldern abgeglichen werden. Besonders dort, wo klassische algorithmische Lösungen aufwendig, teuer oder schwer wartbar

wären, kann KI neue Möglichkeiten eröffnen. Die betrachteten Praxissituationen zeigen, dass der Nutzen vor allem in der Unterstützung menschlicher Arbeit liegt: KI kann unstrukturierte Informationen aufbereiten, Datenerfassung erleichtern, Dokumente prüfen, Fehler reduzieren und damit Prozesse effizienter sowie nutzerfreundlicher machen.

Gleichzeitig erfordert der KI-Einsatz eine klare Begrenzung. Eine Lösungsskizze sollte beschreiben, welche Aufgabe die KI übernimmt, welche Daten benötigt werden, wie Ergebnisse kontrolliert werden und welches Betriebsmodell vorgesehen ist. Datenschutz, Nachvollziehbarkeit und technologische Abhängigkeiten bleiben dabei zentrale Aspekte, können aber durch geeignete technische und organisatorische Maßnahmen früh berücksichtigt werden.

Der Beitrag ist als konzeptioneller und praxisorientierter Rahmen zu verstehen. Eine empirische Evaluation in realen Angebots- oder Projektkontexten steht noch aus. Zukünftige Arbeiten sollten den Rahmen anhand konkreter Fallstudien anwenden, Bewertungskriterien weiter operationalisieren und untersuchen, wie Projektteams den Rahmen in Angebotsprozesse integrieren können.

Projektbeteiligte entscheiden somit nachvollziehbar über KI, indem sie fachlichen Nutzen, Umsetzbarkeit und Risiken bereits in der Konzeptionsphase gemeinsam bewerten. KI wird dadurch nicht nachträglich ergänzt, sondern als verantwortungsvoll geplanter Bestandteil moderner Business-Software verankert.

## Literaturverzeichnis

- [Br20] Brown, T. & Mann, B. & Ryder, N. & Subbiah, M. & Kaplan, J. & Dhariwal, P. & Neelakantan, A. & Shyam, P. & Sastry, G. & Askell, A. & Agarwal, S. & Herbert-Voss, A. & Krueger, G. & Henighan, T. & Child, R. & Ramesh, A. & Ziegler, D. & Wu, J. & Winter, C. & Amodei, D.: Language Models are Few-Shot Learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20). Curran Associates Inc., Red Hook, NY, USA, Article 159, 1877–1901, 2020.
- [Fa23] Fatouros, G. & Soldatos, J. & Kouroumali, K. & Makridis, G. & Kyriazis, D.: Transforming sentiment analysis in the financial domain with ChatGPT. Machine Learning with Applications. 14, 2023.
- [He04] Hevner, A. R. & March, S. T., & Park, J. & Ram, S.: Design science in information systems research. MIS Quarterly, 28(1), 75–105, 2004.
- [Ka14] Kalenborn, A.: Angebotserstellung und Planung von Internet-Projekten: die werkzeuggestützte "Modeling by Example"-Methode, Springer Vieweg, 2014.
- [Kl24] Klose, N.: Unterstützung bei der Eingabe und Validierung von Daten in Webanwendungen mit Hilfe von KI-Tools, Masterarbeit, Universität Trier, 2024.
- [La20] Lamsweerde, A.: Requirements Engineering in the Year 00: A Research Perspective. In Proceedings of the 22nd international conference on Software engineering (ICSE '00). Association for Computing Machinery, New York, NY, USA, 5–19, 2000.
- [LTD24] Lee, D. & Todorova, C. & Dehghani, A.: Ethical Risks and Future Direction in Building Trust for Large Language Models Application under the EU AI Act. In Proceedings of the 2024 Conference on Human Centred Artificial Intelligence - Education and Practice (HCAIep '24). Association for Computing Machinery, New York, NY, USA, S. 41–46, 2024.
- [Lo24] Lo, D.: Requirements Engineering for Trustworthy Human-AI Synergy in Software Engineering 2.0. IEEE 32nd International Requirements Engineering Conference (RE), Reykjavik, Iceland, 2024, pp. 3-4, 2024.
- [Mi24] Minaee, S. & Mikolov, T. & Nikzad, N. & Chenaghlu, M.A. & Socher, R. & Amatriain, X. & Gao, J.: Large Language Models: A Survey. ArXiv, abs/2402.06196, 2024.

- [Op25a] OPENAI: Tokenizer: Learn about language model tokenization, Verfügbar unter: <https://platform.openai.com/tokenizer>, abgerufen am 29. Juni 2025, 2025.
- [Op25b] OPENAI: API-Preisgestaltung, Verfügbar unter: <https://openai.com/de-DE/api/pricing>, abgerufen am 29. Juni 2025, 2025.
- [Op25c] OPENAI: Best Practices for Prompt Engineering with the OpenAI API. Verfügbar unter: <https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api>, abgerufen am 28. Juni 2025, 2025.
- [PVB24] Pessanha, G. & Vieira, A. & Brandão, W.: Large Language Models (LLMs): A systematic study in Administration and Business. In *RAM, Rev. Adm. Mackenzie*. 2024. Vol. 25, 2024.
- [Sh23] Shen, Y. & Heacock, L. & Elias, J. & Hentel, K. & Reig, B. & Shih, G. & Moy, L.: ChatGPT and Other Large Language Models Are Double-edged Swords. In *Radiology*. 307, 2023.
- [SIM23] Sinjanka, Y. & Ibrahim, U. & Malate, F.: Text Analytics and Natural Language Processing for Business Insights: A Comprehensive Review. *International Journal for Research in Applied Science and Engineering Technology*, S. 1626 – 1653, 2023.
- [Th23] Thanabalan, K.: Information Extraction from PDF Documents: A Practical Approach with ChatGPT, Masterarbeit, Universität Trier, 2023.
- [Th25] Thanabalan, K & Kalenborn, A.: Von Anfang an mit KI: Lösungsskizzen mit intelligenten Komponenten gestalten. *Projektmanagement und Vorgehensmodelle 2025 - Post-Agilität, Resilienz, Transformation*. DOI: 10.18420/pvm2025\_06. Gesellschaft für Informatik e.V.. ISSN: 2944-7682, 2025.
- [VK07] Vaishnavi, V. K. & Kuechler, W.: *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*, 2007.
- [WWS23] Weingart, P. & Wambsgans, T. & Soellner, M.: A taxonomy for deriving business insights from user-generated content. In *ECIS 2023 Research Papers*. 401, 2023.
- [Zh20] Zhao, L. & Alhoshan, W. & Ferrari, A. & Letsholo, K. & Ajagbe, M. & Chioasca, E. & Batista-Navarro, R.: Natural Language Processing (NLP) for Requirements Engineering: A Systematic Mapping Study. In: *ACM Computing Surveys*, Vol. 54, No. 3, 2020.
- [Zh23a] Zhao, W. & Zhou, K. & Junyi, L. & Tianyi, T. & Wang, X. & Hou, Y. & Min, Y. & Zhang, B. & Zhang, J. & Dong, Z. & Du, Y. & Yang, C. & Chen, Y. & Chen, Z. & Jiang, J. & Ren, R. & Li, Y. & Tang, X. & Liu, Z. & Wen, J.: A Survey of Large Language Models. *arXiv.Org*, abs/2303.18223, 2023.
- [Zh23b] Zhang, Q. & Fang, C. & Xie, Y. & Zhang, Y. & Yang, Y. & Sun, W. & Yu, S. & Chen, Z.: A Survey on Large Language Models for Software Engineering. *CoRR* abs/2312.15223, 2023.

## Call for Papers / Aufruf zur Einreichung von Beiträgen

### **Projekte im Wandel: Disruption, Künstliche Intelligenz, hybride Methoden**

Herman Hollerith Zentrum | Böblingen | 5. & 6. November 2026

12. gemeinsame Tagung der Fachgruppen Vorgehensmodelle und Projektmanagement im Fachbereich Wirtschaftsinformatik der Gesellschaft für Informatik e. V. (GI), in Kooperation mit der Fachgruppe IT-Projektmanagement der GPM e. V., sowie dem PMI Germany Chapter e. V.

#### **Ausgangslage**

Technologische Durchbrüche, geopolitische Unsicherheiten, beschleunigte Innovationszyklen und neue Formen der Zusammenarbeit führen zu einer grundlegenden Neujustierung von Projektarbeit. Disruption wird zu einem permanenten Begleiter in der Projektarbeit.

Künstliche Intelligenz hält in immer mehr Phasen des Projektmanagements Einzug, von Planung und Risikoanalyse über Entscheidungsunterstützung bis hin zur teilautonomen Steuerung von Projekten. KI verändert nicht nur Werkzeuge und Prozesse, sondern stellt auch etablierte Rollen, Verantwortlichkeiten und Führungsmodelle infrage.

Gleichzeitig zeigt die Praxis, dass weder rein klassische noch agile Vorgehensmodelle den vielfältigen Anforderungen moderner Projekte vollständig gerecht werden. Hybride Methoden und postagile Vorgehensmodelle, die unterschiedliche Paradigmen kombinieren, haben sich in vielen Kontexten etabliert. Ihre konkrete Ausgestaltung ist jedoch stark situativ und reicht von pragmatischen Ansätzen bis hin zu systematisch entwickelten Vorgehensmodellen, die das Team in den Vordergrund stellen.

#### **Zentrale Fragestellungen**

- Wie verändern disruptive Rahmenbedingungen die Planung, Steuerung und Organisation von Projekten?
- Welche Potenziale und Grenzen hat der Einsatz von Künstlicher Intelligenz im Projektmanagement – technisch, organisatorisch und ethisch?
- Wie lassen sich hybride und postagile Vorgehensmodelle sinnvoll gestalten, evaluieren und weiterentwickeln?
- Welche zentralen Kompetenzen benötigen Projektleitende, Teams und Organisationen in einer zunehmend KI-gestützten Projektwelt?

- Welche Spannungsfelder entstehen durch datengetriebene Entscheidungsmodelle und KI-gestützte Assistenzsysteme?
- Welche neuen Gestaltungsmöglichkeiten entstehen auf diese Weise?
- Welche Rolle spielt die Produkt- vs. die Projektsicht im Unternehmen heute?

## Mögliche Themenschwerpunkte

1. **Disruption und Projektmanagement**  
Auswirkungen disruptiver Technologien, Märkte und Organisationsformen auf Projektstrukturen, Entscheidungslogiken und Erfolgsdefinitionen.
2. **Künstliche Intelligenz im Projektmanagement**  
Einsatz von KI in Planung, Controlling, Risiko- und Ressourcenmanagement, Entscheidungsunterstützung sowie ethische, rechtliche und organisationale Fragestellungen.
3. **Hybride und postagile Vorgehensmodelle und Methodenlandschaften**  
Gestaltung, Evaluation und Weiterentwicklung hybrider Modelle zwischen klassisch, agil, postagil und datengetrieben.
4. **Rollen, Führung und Kompetenzen im Wandel**  
Veränderungen von Teamrollen, Leadership-Modellen und Kompetenzanforderungen in KI-gestützten und hybriden Projektumgebungen.
5. **Organisation, Governance und Skalierung**  
Einbettung von Projekten in organisationale Strukturen, Portfolio- und Programmmanagement, Governance-Modelle sowie Skalierung hybrider und KI-basierter Ansätze.

## Zielsetzung der Fachtagung

Die GI-Fachtagung **Projektmanagement und Vorgehensmodelle 2026 (PVM 2026)** bietet ein Forum für den Austausch zwischen Wissenschaft und Praxis zu diesen Fragestellungen. Ziel ist es,

- aktuelle Forschungsergebnisse und theoretische Konzepte vorzustellen,
- praktische Erfahrungen, Fallstudien und Lessons Learned zu diskutieren,
- neue Impulse für die Weiterentwicklung von Vorgehensmodellen zu geben und
- den Dialog zwischen Disziplinen, Branchen und Perspektiven zu fördern.

Die Tagung richtet sich an Wissenschaftlerinnen und Wissenschaftler, Praktikerinnen und Praktiker sowie alle, die sich mit der Gestaltung und Steuerung von Projekten im Spannungsfeld von Disruption, KI und hybriden Methoden beschäftigen.

## Beitragskategorien

### Wissenschaftliche Beiträge / Fachbeiträge (Langbeiträge)

Hierunter fallen Ergebnisse abgeschlossener Forschungsvorhaben und fundierte wissenschaftliche Ergebnisse, um diese mit den Teilnehmern aus Wissenschaft, Studium und Praxis zu diskutieren.

## **Denkanstöße und Impulse (Kurzbeiträge)**

Eine wichtige Aufgabe der GI-Fachgruppen ist es, sich mit der Zukunft des Fachgebiets zu beschäftigen. Reifende Ideen oder kontroverse bzw. provokative Ansichten können in Impulsvorträgen vorgestellt werden, welche anschließend im Auditorium diskutiert werden sollen. Durch den Austausch sollen Denkanstöße und Impulse für die Teilnehmenden und für die künftige Fachgruppenarbeit entstehen. Die Einreichung kann als Kurzbeitrag erfolgen.

## **Berichte aus der Praxis (Foliensatz)**

Auch Praxisberichte sind willkommen, um sie mit anderen erfahrenen Praktikern und Wissenschaftlern zu diskutieren und daraus neue Impulse für Organisationen und die wissenschaftliche Forschung zu gewinnen. In den Berichten können sowohl positive als auch negative Projekterfahrungen dargestellt werden. Praxisberichte können alle relevanten Themen des Projektmanagements abdecken. Die Einreichung kann in Form eines Foliensatzes. Gerne können Sie vorab einen Abstract einreichen, um eine erste Einschätzung zum Vortragsthema durch das Programmkomitee zu erhalten.

## **Diskussion bereits publizierter Arbeiten**

Ähnlich dem „Journal-first“-Modell anderer internationaler Konferenzen, können Beiträge eingereicht werden, welche bereits in renommierten Journalen und Konferenzen publiziert oder zur Publikation/Präsentation angenommen wurden.

## **Hinweise für Studierende und Nachwuchswissenschaftlerinnen**

Insbesondere fordert das Programmkomitee auch Studierende und Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftler auf, mit eigenen Beiträgen in allen oben aufgeführten Kategorien aktiv an der Tagung teilzunehmen, um auf diesem Weg die Community der Fachgruppen kennenzulernen. Studierende können einen Zuschuss zur Teilnahme an der Tagung beantragen.

## **Formate, Einreichung und Fristen**

- Langbeiträge mit maximal 10 Seiten
- Kurzbeiträge mit maximal 5 Seiten
- Foliensätze (für Praxisberichte)

Weitere Details zur Einreichung (Vorlagen, Seitenzahlbegrenzungen, Einreichungssystem) findet man auf der Website <https://pvm-tagung.de/beitrag-einreichen>.

Ein herausragender Beitrag wird mit dem Best Paper Award prämiert.

## **Bitte beachten Sie folgende Fristen:**

- **14.06.2026:** Einreichung des Beitrags durch die Autoren
- **14.07.2026:** Benachrichtigung der Autoren
- **16.08.2026:** Einreichung finaler Version
- **31.10.2026:** Einreichung der finalen Vortragsfolien

## Tagungsband und Indizierung

Die Textbeiträge werden im gedruckten Tagungsband in den GI Lecture Notes in Informatics publiziert (<https://www.gi.de/service/publikationen/lni.html>) und von dblp (<http://dblp.uni-trier.de/>) indiziert.

Foliensätze werden nicht im Tagungsband publiziert.

## Tagungsort und Datum

Die Tagung findet am **5. und 6. November 2026** im Hermann Hollerith Zentrum, Danziger Str. 6, 71034 Böblingen statt ([www.hhz.de/](http://www.hhz.de/)).

## Kontakt

Auf der Webseite (<https://www.pvm-tagung.de>) und auf LinkedIn (<https://www.linkedin.com/company/pvm-tagung>) werden laufend aktualisierte Informationen zur Tagung bereitgestellt.

Für Rückfragen wenden Sie sich bitte an: [info@pvm-tagung.de](mailto:info@pvm-tagung.de)

Für das Programmkomitee der Tagung:

Dr. Martin Bertram (Vorstandsmitglied PMI Germany Chapter)

Prof. Dr. Martin Engstler (Hochschule der Medien, Stuttgart)

Prof. Dr. Timm Eichenberg (Hochschule Weserbergland, Hameln)

Prof. Dr. Eckhart Hanser (Sprecher der Fachgruppe Vorgehensmodelle)

Prof. Dr. Axel Kalenborn (Stv. Sprecher der Fachgruppe Projektmanagement)

Prof. Dr. Marco Kuhrmann (Hochschule Reutlingen)

Prof. Dr. Ninja Leikert-Böhm (Zürcher Hochschule für Angewandte Wissenschaften, Winterthur)

Prof. Dr. Oliver Linssen (Stv. Sprecher FG Vorgehensmodelle & FG IT-Projektmanagement der GPM)

Luca Randecker (Sprecher der Fachgruppe Projektmanagement)

Alexander Volland (Union Investment Services & IT GmbH)

Dr. Enes Yigitbas (Stv. Sprecher der Fachgruppe Vorgehensmodelle)

# Tagungsbericht PVM 2025

## Post-Agilität, Resilienz, Transformation

Die gemeinsam durchgeführte Fachtagung **Projektmanagement und Vorgehensmodelle 2025** der GI-Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) fand am **20. und 21. November 2025** an der Hochschule Weserbergland in Hameln statt. Die Veranstaltung stand unter dem Leitthema „**Post-Agilität, Resilienz, Transformation**“ und beleuchtete zentrale Fragen zur Zukunft der Projektarbeit in einem Zeitalter zunehmender Unsicherheit und technologischer Transformation.



Abbildung 1 (Publikum der PVM 2025)

**Alexander Volland**, Sprecher der Fachgruppe Projektmanagement, eröffnete die Tagung mit der provokanten Frage, ob Agilität angesichts zunehmender Komplexität wirklich die alleinige Lösung sein kann oder ob sie lediglich ein Teil davon ist. Er regte an, ob die Post-Agilität nicht vielmehr eine Erweiterung und Neuausrichtung des bisherigen Verständnisses sein muss, um den Anforderungen moderner Projektumgebungen besser gerecht zu werden. Diese Fragestellung warf den Kernimpuls für die PVM 2025 auf: Wie kann Projektmanagement in einer Welt gestaltet werden, die neben Agilität auch Resilienz und technologische Innovationen wie KI integriert? Die PVM 2025 stellte dabei bewusst die Frage in den Mittelpunkt, ob der Begriff der Agilität in der heutigen Zeit noch zeitgemäß ist oder ob wir uns bereits in einer neuen Entwicklung befinden, der sogenannten Post-Agilität.

**Eckhart Hanser**, Sprecher der Fachgruppe Vorgehensmodelle, versuchte insbesondere, den Begriff „Post-Agilität“ abzugrenzen. Damit ist eben nicht gemeint, dass Agilität „tot“ ist, sondern, dass sie sich weiterentwickelt. Die Forschungen in den Fachgruppen zeigen, dass Prozesse auf den Firmenkontext angepasst werden müssen und Mensch und Kultur – also auch das Team – zunehmend berücksichtigt werden muss.

Die Tagung thematisierte, wie Organisationen in einer zunehmend komplexen und unsicheren Welt gleichzeitig agil und widerstandsfähig, also resilient, agieren können. Darüber hinaus wurde die Rolle der künstlichen Intelligenz und der digitalen Transformation für die Gestaltung zukunftsfähiger Projektorganisationen intensiv diskutiert. In diesem Spannungsfeld von Agilität, Resilienz und technologischer Innovation schuf die PVM 2025 eine Plattform für den

Austausch und die Vernetzung von Wissenschaft, Praxis und Beratung, um gemeinsam Lösungen und Perspektiven für das Projektmanagement der Zukunft zu entwickeln.

## **Keynote: Erfolg ist Kein Zufall, Erfolg ist Eine Haltung**

Der erste Tag der Tagung wurde abgeschlossen mit einer inspirierenden Keynote von **Kathrin Boron**, vierfache Olympiasiegerin und achtfache Weltmeisterin. Unter dem Titel „Erfolg ist kein Zufall, Erfolg ist eine Haltung“ vermittelte Boron lebensnahe Lektionen aus dem Spitzensport, die unmittelbar auf Projektmanagement und Organisationsentwicklung übertragbar sind.

Ihr Vortrag betonte die Bedeutung von mentaler Haltung, kontinuierlicher Verbesserung und dem bewussten Umgang mit Herausforderungen – Aspekte, die sich durch das gesamte Tagungsprogramm zogen und die zentrale Erkenntnis der PVM 2025 verdeutlichten: Erfolgreiche Projektarbeit erfordert nicht nur methodische Kompetenz, sondern eine tiefgreifende kulturelle und persönliche Transformation.

## **Fachvorträge: Von Post-Agilität zu Künstlicher Intelligenz**

Die Fachvorträge der PVM 2025 präsentierten ein breites Spektrum aktueller Forschungsergebnisse und innovativer Ansätze im Projektmanagement. Dabei durchliefen die Beiträge einen strengen wissenschaftlichen Begutachtungsprozess mit einer Annahmequote von 47%.

Larissa Koch de Souza und Martin Engstler präsentierten den Beitrag „Zwischen Agilität und Resilienz – Spannungsfelder und das Potenzial post-agiler Ansätze in digitalen Plattformen“. Ihr Vortrag beleuchtete das komplexe Wechselverhältnis zwischen Agilität und Resilienz, insbesondere im Kontext digitaler Plattformen, und zeigte die Notwendigkeit einer bewussten Balance zur erfolgreichen Gestaltung organisationaler Infrastruktur auf.

Jan Wehinger widmete sich mit „Leadership für AI-Agents – Wer führt die Künstliche Intelligenz?“ der Herausforderung, wie KI nicht nur als Werkzeug, sondern als potenzieller virtueller Partner in Projekten geführt werden kann. Hierbei thematisierte er die neue Dynamik in Mensch-KI-Teams und die sich daraus ergebenden Führungs- und Verantwortungsfragen.

In „Einsatz agiler Projektmanagementmethoden in KMU – Eine qualitative Untersuchung in der Maschinen- und Anlagenbau-Branche“ diskutierte Timm Eichenberg, wie kleine und mittlere Unternehmen agile Methoden adaptieren und welche spezifischen Herausforderungen sich daraus ergeben.

Ninja Leikert-Boehm stellte mit „Fit for Agility? Exploring Person-Job Fit in SAFe Teams“ eine explorative Studie vor, die untersuchte, inwieweit die Passung von Personen zu ihren Rollen in agilen Skalierungsrahmen wie SAFe ein Erfolgsfaktor für agile Transformationen darstellt.

Der Beitrag „Von Anfang an mit KI – Lösungsskizzen mit intelligenten Komponenten gestalten“ wurde von Kokulan Thanabalan und Axel Kalenborn präsentiert. Er liefert einen Beitrag zur Integration intelligenter Komponenten bereits in den frühen Phasen von Projekten, um innovative und zukunftsorientierte Lösungen zu entwickeln.

Der Vortrag „Wege zur Post-Agilität – Eine literaturbasierte Analyse der letzten zehn Jahre PVM-Fachtagung“ von Larissa Koch de Souza, Luca Randecker und Martin Engstler thematisierte wie sich die Tagung hinsichtlich „Post-Agilität“ im Laufe der Zeit entwickelte. Die Autoren analysierten die Entwicklung des Agilitätsverständnisses anhand von 58 PVM-Beiträgen von 2014 bis 2024 und zeigten einen Paradigmenwechsel hin zu kulturellen und

kontextsensitiven Interpretationen auf. Post-Agilität wurde dabei als strategische und operative Weiterentwicklung verstanden, nicht als Ablehnung agiler Prinzipien. Für ihren Beitrag wurden sie mit dem Best Paper Award 2025 ausgezeichnet.

Schließlich beschrieb Marco Kuhrmann mit „Towards a Hybrid Process for Integrating Systems, Software, and Games Engineering“ die Entwicklung eines spezialisierten hybriden Prozessmodells in technischen Projekten unter dem Einfluss verschiedener Engineering-Disziplinen.

Dieser vielfältige Fachvortragsblock zeigte eindrücklich die aktuelle Bandbreite und Tiefe der Forschung im Projektmanagement und setzte wichtige Impulse für die Gestaltung moderner Projektumgebungen.

## **Diskussionsbeiträge im Future Track: Zukunftsorientierte Perspektiven**

Neben den Hauptvorträgen präsentierten ausgewählte Autoren im sogenannten Future Track ihre innovativen Beiträge, die neue Perspektiven auf zentrale Herausforderungen des Projektmanagements eröffnen. Dieser Future Track ist ein essenzieller Bestandteil der Tagung, da er nicht nur aktuelle Forschungsergebnisse, sondern auch zukunftsweisende Diskussionsbeiträge umfasst, die in einem intensiven Austausch mit dem Publikum diskutiert werden. Die zusätzliche Sparte ermöglicht es den Teilnehmenden, frühzeitig Trends zu erkennen, innovative Ansätze kritisch zu reflektieren und sich aktiv an der Gestaltung der zukünftigen Entwicklung des Projektmanagements zu beteiligen. Die Integration eines solchen Future Tracks sichert somit die kontinuierliche Weiterentwicklung des Fachgebiets und fördert die Brücke zwischen Wissenschaft und Praxis.

Sarah Brandt, Sven Theobald diskutierten die „Post-Agilität - Einordnung und Blick in die Zukunft“. Der Beitrag systematisiert das Konzept der Post-Agilität und legt dar, wie Organisationen diese strategische Weiterentwicklung praktisch umsetzen können.

Rebecca Fischer präsentierte „Resiliente Verwaltung – Sicherstellung öffentlicher Dienstleistungen durch die Implementierung eines Business Continuity Management Systems in deutschen Kommunalverwaltungen“. Der Beitrag transferiert Resilienzkonzepte auf den öffentlichen Sektor und zeigt auf, wie systematische Kontinuitätsmanagementsysteme zum Schutz kritischer Infrastrukturen beitragen.

Sven Theobald präsentierte „Challenges of Integrating Agile Development and Analytical Quality Assurance“ – ein kritischer Beitrag, der die Spannungen zwischen agiler Entwicklung und systematischer Qualitätssicherung analysiert.

Claus Hüsselmann präsentierte „Zwischen Stabilität und Anpassungsfähigkeit – Entwicklung und Anwendung eines Reifegradmodells für Lean-Adaptive Project Portfoliomanagement“. Der Beitrag entwickelt ein Modell zur Bewertung und Entwicklung von Portfoliomanagementsystemen, die sowohl Stabilität als auch Anpassungsfähigkeit bieten.

Jil Klünder wurde virtuell zur Konferenz geschaltet und stellte in ihrem Beitrag „How to Mitigate Technical Debt in Startups – Extending the Entrepreneurial Software Engineering Model“ vor. Dabei stellte sie dar, wie technische Schulden in der schnelllebigen und ressourcenbegrenzten Umgebung von Startups entstehen und wie eine erweiterte Version des Entrepreneurial Software Engineering Models helfen kann, diese technischen Schulden systematisch zu vermeiden, zu managen und zu priorisieren.

Jessica Nagel präsentierte „Digitale Kompetenz im Projektmanagement – Ein Kompetenzstrukturmodell für das digital strukturierte Projektmanagement“ und definiert damit zentrale Kompetenzdimensionen für das Projektmanagement im digitalen Zeitalter.



Abbildung 2 (Dr. Jessica Nagel)

„Table of PM Elements“, ein praxisnahes und strukturiertes Methodenbaukastensystem, das bewährte Methoden und Werkzeuge für das Projektmanagement vereint wurde von Bernhard Schloß präsentiert. Mit diesem Beitrag leistet Schloß einen wichtigen Beitrag für den pragmatischen Umgang mit der methodischen Vielfalt in modernen Projekten.

Der Beitrag von Kendra Pöhlmann und Julia Hodapp zu „Transformational Skills für die digitale und nachhaltige Transformation – Innere Entwicklung als Schlüssel zur Veränderung von Projektmanagement-Kultur“ musste aus organisatorischen Gründen entfallen, kann jedoch im Tagungsband nachgelesen werden.

## Praxisberichte: Erfahrungen aus der Unternehmensrealität

Die PVM 2025 würdigte die Bedeutung der praktischen Erfahrung durch die zusätzliche Kategorie der Praxisberichte, die direkte Einblicke in reale Transformationsprozesse boten.

Christian Zender und Jens Zimmermann präsentierten den Praxisbericht „Agilität im Alltag der Atruvia – Zwischen Theorie und Praxis“, in dem eindrucksvoll dargestellt wird, wie ein großes Finanzdienstleistungsunternehmen agile Prinzipien tief in seiner Organisationskultur verankert hat. Ein zentrales Element der Transformation war die Neugestaltung der Strukturen zwischen Linien- und Projektorganisation: Die Linieneinheiten wurden zu agilen Lieferorganisationen umgebaut, während Projekte vor allem dann initiiert werden, wenn übergreifende Aufgaben vorliegen, die von einzelnen Liefertteams nicht bearbeitet werden können. Sämtliche Prozesse und Rollen wurden klar dokumentiert, um Transparenz und Verbindlichkeit zu schaffen. Die wichtigste Erkenntnis dabei war, dass der Schwerpunkt der Transformation auf der kulturellen Entwicklung lag – nicht nur auf der Einführung neuer Methoden. Dieses Vorgehen spiegelt wider, dass nachhaltiger agiler Wandel vor allem durch Veränderungen in der Haltung, Zusammenarbeit und Führung gelingt. Dies wird auch in zahlreichen Best-Practice-Beispielen großer Unternehmen und in der einschlägigen Literatur immer wieder betont.

Thorsten Nottebaum präsentierte einen kritisch-reflektierenden Beitrag: „Agil gescheitert? Warum agile Transformationen ins Stocken geraten – und wie Projekte trotzdem gelingen“. Der Bericht analysiert Erfolgsfaktoren und Fallstricke agiler Transformationen und bietet konkrete Handlungsempfehlungen.

Beide Beiträge berichten aus der Praxis, dass Methoden allein nicht genügen, sondern vor allem die Haltung, Kultur und Führung in Unternehmen entscheidend sind. Somit ergänzen sich die Beiträge und bieten ein umfassendes Bild von der Bedeutung der kulturellen Dimension für agile Transformationsprozesse.

## **Kompaktbriefings: Intensive Weiterbildung**

Neben diesen zukunftsweisenden Perspektiven bietet die PVM ein integriertes Schulungsprogramm an: Die Kompaktbriefings wurden erneut sehr gut angenommen und von Firmenvertretern besonders gelobt. Neben praxisnahen Workshops wie „DevOps in 45 Minuten“, in dem Eckhart Hanser DevOps als wichtige Brücke zwischen Entwicklung und Betrieb mit Fokus auf agile und kontinuierliche Deployment-Praktiken vorstellte, wurden auch innovative Themen wie „Künstliche Intelligenz im Projektmanagement“ durch Axel Kalenborn behandelt. Zudem beleuchteten André von Zobeltitz und Lukas Zärtner in ihrem Briefing „Zwischen Buzzword und Businessnutzen: Was Unternehmen an einer KI-Nutzung hindert“ kritisch die Herausforderungen und Hindernisse bei der KI-Einführung in Unternehmen. Klaus Schopka erläuterte, wie „Programmmanagement zur Steuerung strategischer Transformationen“ die Projektmanagementmethoden im Unternehmen ergänzen muss, um komplexe Transformationsprogramme in Organisationen umzusetzen.

## **Rahmenprogramm und Networking**

Das Rahmenprogramm der PVM 2025 bot reichlich Gelegenheit für informellen Austausch und Networking. Abends fanden thematische Erlebnisführungen durch Hameln statt, darunter „Der Rattenfänger von Hameln“ und „Der verführte Verführer“, die kulturelle Erfahrung und Networking kombinieren. Das festliche Konferenzdinner im historischen Hotel zur Krone in Hameln schuf einen eleganten Rahmen für den fachlichen Austausch und war zudem der feierliche Moment der Verleihung des Best Paper Awards. Organisiert wurde das Rahmenprogramm maßgeblich von der Hochschule Weserbergland unter der Leitung der Professoren Timm Eichenberg und Michael Städler.



Abbildung 3 (Erlebnisführung „Der Rattenfänger von Hameln“)

## Zusammenfassung und Ausblick

Die PVM 2025 hat mehrere zentrale Erkenntnisse hervorgebracht, die entscheidend für die zukünftige Entwicklung des Projektmanagements sind. Post-Agilität wird dabei nicht als Ablehnung der Agilität gesehen, sondern als notwendige Weiterentwicklung, die kontextsensitive Anpassungen, kulturelle und menschliche Aspekte, Governance und Skalierung auf Organisationsebene sowie Nachhaltigkeits- und regulatorische Anforderungen integriert. Dabei funktioniert Agilität in den einzelnen Teams in aller Regel gut, nicht aber unbedingt in der Organisation. Resilienz ist als komplementäre Kompetenz neben Agilität anerkannt, wobei Organisationen sowohl schnelle Anpassungsfähigkeit als auch stabile Widerstandskraft benötigen und eine bewusste Balance zwischen diesen Ansätzen gewinnen müssen. Künstliche Intelligenz eröffnet als Technologie neue Potenziale im Projektmanagement, bringt jedoch auch Herausforderungen hinsichtlich Führung, Verantwortung und der Neuinterpretation von Mensch-Maschine-Beziehungen mit sich. Einen besonderen Fokus legt die Tagung auf transformative Leadership und die innere Entwicklung von Führungskräften und Teams, bei der mentale Haltung, psychologische Sicherheit und kontinuierliches Lernen als Erfolgsfaktoren gelten. Hinzu kommen nachhaltige und regulatorische Dimensionen, die agile Praktiken neu definieren müssen.

Diese Erkenntnisse bilden die Basis für weitere Diskussionen und praktische Entwicklungen, wobei wichtige Fragen offenbleiben, etwa wie resilient-agile Organisationsformen realisiert werden können, welche Kompetenzen durch KI nötig werden, wie Nachhaltigkeit mit agilen Methoden verbunden werden kann und welche Standards post-agile Transformationen unterstützen.

Eckhart Hanser stellte im Schlusswort fest, dass zwar noch keine abschließende Definition von Post-Agilität existiert, aber der begonnene Dialog von großem Wert ist. Die PVM 2025 fungierte somit als wichtiger Impulsgeber für die Weiterentwicklung des Projektmanagements in einer von Post-Agilität, Resilienz und digitaler Transformation geprägten Welt.



Abbildung 4 (v.l.n.r. Prof. Dr. Timm Eichenberg, Dr. Martin Bertram, Oliver Linssen, Prof. Dr. Eckhart Hanser, Alexander Volland, Prof. Dr. Michael Städler)

## **Publikation und Weitere Informationen**

Der Tagungsband der PVM 2025 wurde in den **GI Lecture Notes in Informatics (Band P-371)** veröffentlicht und ist Bestandteil der renommierten Publikationsreihe der Gesellschaft für Informatik e. V.

### **Herausgeber:**

- Prof. Dr. Oliver Linssen (FOM Hochschule, Universität Wuppertal)
- Alexander Volland (Union Investment Services IT GmbH)
- Dr. Enes Yigitbas (Universität Paderborn)
- Prof. Dr. Martin Engstler (Hochschule der Medien Stuttgart)
- Dr. Martin Bertram (PMI Germany Chapter)
- Prof. Dr. Eckhart Hanser (Duale Hochschule Baden-Württemberg Lörrach)
- Prof. Dr. Axel Kalenborn (Universität Trier)

**Gastgeber:** Hochschule Weserbergland, Hameln

### **Veranstalter:**

- Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) der Gesellschaft für Informatik e.V. (GI) e.V.

**Weitere Informationen:** <https://www.pvm-tagung.de>

**Verfassungsdatum:** November 2025

**Ort:** Hameln, Hochschule Weserbergland

# **KI-getriebene Veränderungen im Software Engineering – KI4SE 2025: ausführlicher Bericht zum Workshop am Fraunhofer IESE in Kaiserlautern**

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 zu meist nicht auf eigene Anwendungsfälle übertragen.

## **Ausblick und Dank**

Auch für das Jahr 2026 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 den zu Beginn des Jahres 2026 publizierten Tagungsband erlaubt.



Hochschule für  
Wirtschaft und Recht Berlin  
Berlin School of Economics and Law

**Berliner Schriften zu modernen Integrationsarchitekturen**

Hrsg.: Prof. Dr.-Ing. habil. Andreas Schmietendorf

**Hochschule für Wirtschaft und Recht Berlin**

Fachbereich II

Wirtschaftsinformatik – Systementwicklung

**KI-getriebene Veränderungen  
im Software Engineering (KI4SE)**

**Ergebnisse eines öffentlichen Expertenworkshops  
am Fraunhofer IESE**

**SHAKER  
VERLAG**

*Schmietendorf, 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

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.

A word cloud containing various software engineering terms. The most prominent words are "effizienz" and "produktivitätssteigerung". Other visible terms include "optimierung", "anforderungsgenerierung", "refactoring vorschläge", "funktioniert rudimentär", "wissensmanagement", "prototypen", "geschwindigkeit", "neue lösungen", "chance", "stabilere software", "automation", "automatisierung", "durchsatz", "traceability", "work slop", "effizienzsteigerung", "produktivität", "tooloptimierung", "bug fixing support", "effizientes prototyping", "assistenz bei codeanalyse", "neue anwendungsgebiete", and "unterstützung".



A word cloud of terms related to AI hallucinations. The most prominent word is 'halluzination' in the center. Other large words include 'unsicherheit', 'bias', and 'redundancy'. Smaller words include 'zuverlässigkeit unklar', 'schwachstellen', 'reproduzierbarkeit', 'supply chain attacks', 'fokus senior developer', 'höhere kosten', 'codequalität', 'soziale bedrohung', 'sicherheitsrisiken', 'sicherheit', 'nacharbeit', 'kontrollierbarkeit', 'qualitätsverlust', 'vorbehalte', 'überprüfungsbedarf', 'bullshitting von genai', 'wartbarkeitsprobleme', 'verlässlichkeit', 'prompt injection', 'degradation bei entwickle', 'intransparenz', 'consistency', 'explainability', 'erklärbarkeit', 'vendor lock in', 'blindes vertrauen', 'fehlinterpretation', and 'sicherheitsrisiken'.



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



### SWOT-Analyse: Einsatz von KI im Software Engineering

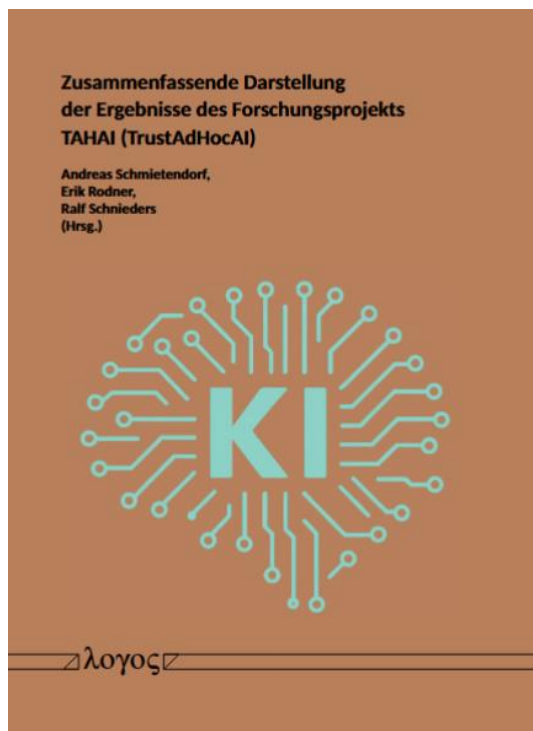
| STÄRKEN                                                                                                                                                                                                                | SCHWÄCHEN                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Automatisierung von Routineaufgaben</li> <li>• Produktivitätssteigerung</li> <li>• Verbesserte Codequalität</li> <li>• Schnellere Entwicklung</li> </ul>                      | <ul style="list-style-type: none"> <li>• Abhängigkeit von Trainingsdaten</li> <li>• Mangelnde Transparenz / Erklärbarkeit</li> <li>• Begrenzte Kontextverständnis</li> <li>• Kosten und Aufwand der Implementierung</li> </ul> |
| CHANCEN                                                                                                                                                                                                                | RISIKEN / BEDROHUNGEN                                                                                                                                                                                                          |
| <ul style="list-style-type: none"> <li>• Innovation durch neue Werkzeuge</li> <li>• Steigerung der Wettbewerbsfähigkeit</li> <li>• Personalisierte Entwicklung</li> <li>• Verbesserung der Software-Vartung</li> </ul> | <ul style="list-style-type: none"> <li>• Arbeitsplatzunsicherheit</li> <li>• Ethische und rechtliche Fragen</li> <li>• Sicherheitsrisiken</li> <li>• Technologische Abhängigkeit</li> </ul>                                    |

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

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



# **Zusammenfassende Darstellung der Ergebnisse des Forschungsprojekts TAHAI (TrustAdHocAI)**



## **Inhalt:**

- Identifikation von KI-Szenarien,
- Auswahl von KI-APIs,
- Bewertung von Datenquellen,
- KI-Risikoanalysen,
- KI-Robustheitsanalysen,
- Ethische und rechtliche Fragen.

ISBN 978-3-8325-5949-6,  
120 Seiten,  
Erscheinungsjahr: 2026  
Logos Verlag Berlin GmbH

Das Buch stellt die Ergebnisse aller im Forschungsprojekt TAHAI (TrustAdHocAI) bearbeiteten Arbeitspakete vor. Das Interesse dieses Projekts galt einer vereinfachten und fachgetriebenen Implementierung von Anwendungsszenarien der künstlichen Intelligenz. Im Mittelpunkt standen vorgefertigten KI-APIs und Low-Code orientierte Ansätze zur Softwareentwicklung. Die Dokumentation der Arbeitspakete greift die verfolgten Ziele, die zur Bearbeitung gewählten Vorgehensweisen bzw. Methoden und schließlich die Ergebnisse auf.

Zunächst wird die für das Projektmanagement eingesetzte Informationsdrehscheibe beschrieben, die dem Termin- und Aufgabenmanagement sowie der Sicherung von Dokumenten diene. Weitere Arbeitspakete beschäftigten sich mit der methodischen Identifikation fachlicher KI-Szenarien, der Auswahl API-basiert angebotener KI-Modelle und der qualitativen Bewertung der für KI-Lösungen einsetzbaren Datenquellen. Im Kern des Projekts realisierte KI-Lösungen (fachlicher Fokus: Professionsforschung, Forstwirtschaft, Bahnwirtschaft) werden hinsichtlich ihrer wesentlichen Eigenschaften charakterisiert. Risiko- und Robustheitsanalysen von KI-Lösungen stellten einen weiteren Schwerpunkt des Projekts dar, wofür entsprechende Ansätze für eine Evaluierung und Zertifizierung entwickelt wurden. Zwei weitere Arbeitspakete beschäftigten sich schließlich mit ethischen und rechtlichen Fragen des KI-Einsatzes im Diskurs der fachlich orientierten KI-Einsatzszenarien.

# Software Engineering prototypische KI-Implementierungen im Zusammenhang mit domänenspezifischen Problemen des Projekts TAHAI (TrustAdHocAI)



## Inhalt:

- Auswahl KI-Modelle
- Domänenspezifische Lösungen
- Anonymisierung
- Transkriptionsanalyse
- Bild- und Videoverarbeitung

ISBN 978-3-8325-5906-9,  
110 Seiten,  
Erscheinungsjahr: 2025  
Logos Verlag Berlin GmbH

Die vorliegende Monografie entstand im Zusammenhang mit der Durchführung des Forschungsprojekts TAHAI (TrustAdHocAI). Allgemein setzte sich das Projekt mit einer vereinfachten und vor allem fachgetriebenen Implementierung von Anwendungsszenarien der künstlichen Intelligenz auseinander. Dafür sollte insbesondere auf vorgefertigte KI-APIs (ggf. bezogen aus der Cloud) und Low-Code-Ansätze zur Softwareentwicklung zurückgegriffen werden. Sehr häufig wird in diesem Zusammenhang auch von einer Demokratisierung des KI-Einsatzes gesprochen.

Mit Hilfe von experimentell realisierten KI-Prototypen sollten entsprechende Ansätze praxisnah erprobt werden, weshalb es sich bei den Ausführungen der Monografie um eine unmittelbare Transferleistung zwischen Forschung und Praxis handelt. Entsprechende Szenarien des KI-Einsatzes bezogen sich auf die Softwareentwicklung selbst, die Professionsforschung (speziell Mediation) und Bahninfrastrukturen. Mit Hilfe der Experimente sollten Potentiale, Grenzen und Risiken eines vereinfachten KI-Einsatzes gegenüber den Stakeholdern des Projekts bzw. potentiell Interessierten verdeutlicht werden. Darüber hinaus galt das Interesse eingehenden Veränderungen im Software Engineering, aber auch der Bewertung der Vertrauenswürdigkeit derart bereitgestellter Lösungen.

# Rezension zu „IT-Management. Das umfassende Handbuch“

Prof. Dr. Rüdiger Weißbach

Wirtschaftsinformatik, Digitale Transformation und Geschäftsmodelle

Fakultät Management, Governance und Medien

Berliner Tor 5, 20099 Hamburg

ruediger.weissbach@haw-hamburg.de

## Rezensiertes Buch

*Maik Wienströer (Hg.), Monika Dürr, Andrea Herrmann, Christian Hochwarth, Pia von Lilien-Waldau, Thomas Stensitzki, Sujeewan Vijayakumaran, Tobias Xu:*

### **IT-Management. Das umfassende Handbuch.**

Bonn: Rheinwerk Computing 2025, ISBN 978-3-8362-8710-4

832 Seiten, EUR 59,90

## Disclaimer

Vorab der Disclaimer: An dem vorgestellten Buch ist mit Andrea Herrmann eine Autorin beteiligt, mit der ich seit langen Jahren in Arbeitsgruppen der GI aktiv bin und mit der ich auch zusammen publiziert habe.

## Rezension

Wenngleich heute kaum noch jemand die 2003 verfasste These von Nicholas Carr „IT doesn't matter“ unterschreiben würde, hinkt jedenfalls in Deutschland das Angebot von Büchern zum Thema IT-Management dem Angebot zu Programmiersprachen, einzelnen Technologien oder „strategischen Nutzenbringen“ deutlich hinterher. Das ist leider schon sehr lange so; auf dem deutschen Markt hatte das erstmals vor über 2 Jahrzehnten erschienene Buch von Tiemeyer lange Zeit eine Alleinstellung inne. Im letzten Jahr ist nun im Rheinwerk-Verlag ein weiteres Werk zu diesem Themenfeld erschienen, herausgegeben von Maik Wienströer, aufgeteilt in zwölf Kapitel von acht Autor\*innen. Nach einer Klärung des Scopes des IT-Managements folgen Abschnitte über operative, taktische und strategische Ebenen, Personalaspekte, Finanzaspekte, IT-Security, ITSM, Sourcing, Lizenzmanagement, IT-Projektmanagement, RE, „Cloud und DevOps“ sowie „Umgebungen betreiben und pflegen“. Verglichen mit dem Buch von Tiemeyer sind die abgedeckten Inhalte grundsätzlich sehr ähnlich; die interne Struktur erscheint etwas grobgranularer.

Dies dürfte der Intention entsprechen, ein „Buch für angehende Führungskräfte“, „von Praktikern für Praktiker“, zu sein. Verstehen durch Lesen, nicht Lernen durch Nachschlagen, so könnten die Unterschiede der Konzepte zusammengefasst werden. Als Zielgruppe werden besonders angehende IT-Manager genannt, die aus dem IT-Bereich kommend erste Führungsaufgaben übernehmen. Auch für Menschen, die aus einem anderen Schwerpunkt kommend nun auch Führungsaufgaben in „der IT“ übernehmen sollen, erscheint mir das Konzept sinnvoll, da es im Regelfall nicht in tiefe technische Details hineinsteigt, sondern auf Zusammenhänge abzielt. Das bringt es auch mit sich, dass manche Kapitel etwas redundant zueinander erscheinen, was beim punktuellen Lesen nicht stört, aber schon irritieren kann, wenn man das Buch „am Stück“ liest. Es greift auch viele aktuelle Bezüge auf. Inwieweit dies das Buch schneller veralten lässt, als ein systematischeres, wird die Zukunft zeigen.

Für die benannten Zielgruppen finden sich hier viele Trigger „zum Weiterdenken“, sowohl Stichworte zu aktuellen Themen, Aufzeigen von wichtigen, aber bisher evtl. nicht bewusst gewordenen Aspekten wie auch Bekanntes als Anknüpfungspunkte. In diesem „Nudging“ dürfte der zentrale Nutzen dieses Buches liegen.

# **Der Fachausschuss und die Fachgruppen WI-VM, WI-PM, WI-PrdM stellen sich vor**

## **Fachausschuss WI-MAW:**

### ***Management der Anwendungsentwicklung und -wartung***

Anwendungssysteme sind aus Sicht der Wirtschaftsinformatik Aufgabenträger im Rahmen der Erfüllung der betrieblichen Gesamtaufgabe. Ihre Aufgabenstellungen werden aus den Unternehmenszielen und den strategischen Zielen der Informationsverarbeitung abgeleitet. Die Entwicklung von Anwendungssystemen erfolgt nicht "kontextfrei", sondern i.A. in einem bestimmten betrieblichen Umfeld. Dies bedeutet zum einen, dass sich das einzelne Anwendungssystem in bereichsübergreifende bzw. unternehmensweite Daten- und Funktionsmodelle oder Objektmodelle einordnen muss. Zum anderen existieren häufig bereits Anwendungen für andere betriebliche (Teil-)Aufgaben, mit denen das System zusammenarbeiten muss.

Der Fachausschuss beschäftigt sich aus dieser Sicht mit der Planung, der Entwicklung, der Einführung, dem Einsatz und der Wartung betrieblicher Anwendungssysteme. Im Vordergrund stehen Vorgehensweisen, Prinzipien und Methoden für die Anwendungsentwicklung im betrieblichen Umfeld sowie ihre Unterstützung durch Softwarewerkzeuge. Im Einzelnen setzt sich der Fachausschuss mit Themen wie den folgenden auseinander:

- Integration von Anwendungssystemen in eine existierende betriebliche DV-Landschaft;
- Sicherung der Investitionen in das Wirtschaftsgut Software; Bewertung von Vorgehensmodellen, Methoden und Werkzeugen zur Anwendungsentwicklung sowie Einsatzerfahrungen;
- Management von Softwareentwicklungsprojekten (Projektplanung, -durchführung und -kontrolle, Projektorganisation, Projektmanagementsysteme, Kosten/ Wirtschaftlichkeit),
- Software Produktmanagement, Configuration Management, Change Management, Migration Management, Reengineering.

Mitgliederzahl: ca. 500

#### **FA-Sprecher**

Prof. Dr. G. Herzwurm  
Universität Stuttgart  
Lehrstuhl für Allgemeine  
Betriebswirtschaftslehre und  
Wirtschaftsinformatik II  
(Unternehmenssoftware)

#### **stellv. FA-Sprecherin**

Dr.-Ing. Birgit Demuth  
Technische Universität Dresden  
Institut für Software- und  
Multimediatechnik

## **Fachgruppe WI-VM:**

### ***Vorgehensmodelle für die betriebliche Anwendungsentwicklung***

Betrachtungsgegenstand der Fachgruppe sind die als "Vorgehensmodelle" bezeichneten Beschreibungen der Aufbau- und Ablauforganisation von Projekten zur Entwicklung und Wartung von Anwendungssystemen. Solche Beschreibungen helfen, die Durchführung von Projekten innerhalb eines Unternehmens oder darüber hinaus zu standardisieren und zu verbessern. Der Begriff Anwendungssystem sei hier sehr weit gefasst: von technischen über betriebswirtschaftliche bis zu organisatorischen Systemen.

Um eine effektive und effiziente Gestaltung der Vorgehensmodelle und damit der Projekte zu erreichen, ist die Berücksichtigung der Schnittstellen zur Betriebswirtschaftslehre einerseits, insbesondere der Organisations- und der Managementlehre, und dem Software Engineering andererseits wesentlich.

Das Thema "Vorgehensmodelle" wird daher von der Fachgruppe aus verschiedenen Blickrichtungen betrachtet:

- Grundlagen: Begriffsdefinitionen, Bestandteile, (formale) Beschreibung von Vorgehensmodellen, Vorgehensmodell-Typen.
- Inhaltliche Bausteine: Konzepte, Methoden, Phasen, Projektmanagement, Qualitätssicherung.
- Werkzeugunterstützung: Vorgehensmodell-Driver, Meta-Modelle, Data-Dictionaries.
- Ökonomische, soziale und psychologische Aspekte: Einführung und Betrieb von Vorgehensmodellen, organisatorisches Umfeld.
- Beispiele aus der Praxis: Standard-Vorgehensmodelle in Organisationen, Branchen und für Anwendungstypen, spezielle Vorgehensmodelle von Unternehmen.
- Standardisierung von Vorgehensmodellen: V-Modell XT, Hermes

Die Fachgruppe fördert einen intensiven Gedankenaustausch durch die Pflege persönlicher Kontakte und unterstützt einen offenen und kritischen Dialog zwischen Wissenschaft und Praxis. Ein weiteres Ziel der Fachgruppe ist die Erarbeitung von Empfehlungen und Stellungnahmen zu den technischen, wirtschaftlichen, organisatorischen und sozialen Aspekten bei Auswahl und Einsatz von Vorgehensmodellen - dies insbesondere vor dem Hintergrund nationaler, europäischer und internationaler Normungs- und Standardisierungsbestrebungen.

Weitere Informationen über Vorgehensmodelle und die Arbeit der Fachgruppe sind im Internet zu finden unter [www.vorgehensmodelle.de](http://www.vorgehensmodelle.de).

#### **FG-Sprecher**

Prof. Dr. Eckhart Hanser  
DHBW Lörrach

#### **stellv. FG-Sprecher**

Dr. rer. nat. Enes Yigitbas  
Universität Paderborn  
Fachgruppe Datenbanken-  
und Informationssysteme

#### **Stellv. FG-Sprecher**

Oliver Linssen  
Bergische Universität  
Wuppertal

## **Fachgruppe WI-PM:**

### ***Projektmanagement***

Die Fachgruppe befasst sich mit dem Einsatz, der Verbreitung sowie der Weiterentwicklung des Projektmanagements. Neben Vertretern aus den Hochschulen sollen vor allem Praktiker die Arbeitsschwerpunkte der Fachgruppe definieren, Ergebnisse erarbeiten und Erfahrungen weitergeben. Für die Aufgabengebiete des Projektmanagements sollen Methoden, Werkzeuge und Techniken untersucht werden. Neben den klassischen Aufgabengebieten wie beispielsweise Projektorganisation, Aufwandschätzung, Projektverfolgung und Projektsteuerung stehen folgende Themen im Vordergrund:

#### **Bedeutung und Dimensionierung des Projektmanagements.**

Die Bedeutung des DV-Projektmanagements als entscheidender Faktor für den Erfolg oder das Mißlingen von DV-Projekten wird von vielen Entscheidungsträgern unterschätzt. Daher sollte die grundsätzliche Bedeutung sowie der Nutzen einer angemessenen Ausstattung des Projektmanagements mit eigenen Ressourcen transparent gemacht werden.

#### **Human Factors.**

In zahlreichen Projekten liegen die größten Projektrisiken bei den sogenannten Human Factors (oder "weichen" Faktoren). Der Umgang mit solchen Risiken erfordert Kompetenz bei Themen wie Motivation, Führung, Teamfähigkeit, Überwindung "politischer" Widerstände u.a.m.

#### **Programm Management.**

Immer öfter gefordert wird das Management eines Portfolios von Projekten, wobei nicht alle Projekte des Portfolios eigentliche DV-Projekte zu sein brauchen. Solche Projektportfolios können beispielsweise als Folge einer veränderten Unternehmensstrategie entstehen und sollen dann einen größeren Veränderungsprozess bewirken. Hauptaufgabe eines Programme Managements ist dabei die zielorientierte Steuerung der Abarbeitung des Projektportfolios, wobei insbesondere unternehmerische Gesichtspunkte zu beachten sind.

#### **FG-Sprecher**

Luca Randecker  
Hochschule der Medien,  
Stuttgart

#### **stellv. FG-Sprecher**

Prof. Dr. Axel Kalenborn  
Universität Trier

## **Fachgruppe WI-PrdM:** ***Software Produktmanagement***

Effizientes und effektives Management softwareintensiver Produkte ist zu einer kritischen Kernkompetenz von Unternehmen geworden. Unternehmen sind mit einer stetig wachsenden Anzahl von Herausforderungen konfrontiert, die durch unterschiedliche Lebenszyklen von Systemen und unterschiedliche Kritikalität im Systemeinsatz in immer mehr – und neuen – Anwendungsfeldern entstehen. Hybride Systeme, z.B. im Internet-of-Things, in Automobilen, Flugzeugen, Drohnen, medizinischen Geräten oder in der Unterhaltungselektronik geben Software eine nie dagewesene Bedeutung. Zusätzlich entstehen durch die vielfältigen Initiativen im Rahmen Digitalisierung neue Arbeits- und Geschäftsmodelle und eröffnen vollkommen neue, durch Software getriebene Möglichkeiten zur Innovation.

In diesem dynamischen Umfeld findet softwaregetriebene Innovation an der Schnittstelle zwischen Informatik/Software Engineering und Wirtschaft statt, zwischen Forschung und industrieller Praxis. Das Produktmanagement umfasst hierbei die Entwicklung, Wartung und Evolution klassischer Softwarelösungen im gesamten Produktlebenszyklus, aber insbesondere auch innovative softwarebasierte Innovation. Die Fachgruppe befasst sich einerseits mit Konzepten, Methoden und Werkzeugen der Informatik/Wirtschaftsinformatik zur Gestaltung des Produktmanagements und der Produktinnovation. Andererseits wird insbesondere auch ein starker Fokus auf die praktische Anwendbarkeit theoretischer Konzepte gelegt.

Die Fachgruppe fördert auf dem genannten Gebiet den intensiven Gedankenaustausch, die Pflege persönlicher Kontakte und die Zusammenarbeit interessierter Personen und Gruppen. Dazu zählt u.a. die gegenseitige Information über Veranstaltungen, Projekte und Veröffentlichungen.

### **FG-Sprecher**

Prof. Dr. rer. pol. Andreas Helferich  
ISM International School of Management  
Campus Stuttgart

### **stellv. FG-Sprecher**

Dr. Dimitri Petrik  
Universität Stuttgart  
Abt. VIII: Wirtschaftsinformatik II  
(Software-intensive Business)

## **Mitglieder des Fachausschusses Management der Anwendungsentwicklung- und wartung (GI-MAW)**

Die Mitglieder des Leitungsgremiums des Fachausschusses finden Sie unter:

<https://fa-wi-maw.gi.de/fachausschuss/leitungsgremium>

# Impressum

Der Rundbrief des Fachausschusses *Management der Anwendungsentwicklung und -wartung (WI-MAW)* ist das Publikationsorgan des Fachausschusses sowie der Fachgruppen

WI-VM *Vorgehensmodelle für die betriebliche Anwendungsentwicklung*

WI-PM *Projektmanagement*

WI-PrdM *Software Produktmanagement*

Der Rundbrief erscheint einmal jährlich elektronisch. Durch den Rundbrief sollen wichtige Erfahrungen, neue Erkenntnisse und aktuelle Informationen unter den Mitgliedern ausgetauscht werden. Rundbriefbeiträge von Mitgliedern und Interessenten sind daher besonders willkommen. Es können Beiträge zu folgenden Rubriken eingereicht werden:

- Fachbeiträge: *Erfahrungsberichte; Theoretische Beiträge; Projektberichte*
- Ankündigungen: *Veranstaltungen; Call for Papers; Einladungen; Programme*
- Berichte: *Tagungsberichte; Buchbesprechungen; Vorstellung von Arbeitsgruppen ;*

Es wird gebeten, Beiträge in elektronischer Form (Word) an die Rundbriefredaktion zu senden. Ein Ausdruck sollte keine Seitennummerierung enthalten.

Die Beiträge können in deutscher oder englischer Sprache abgefasst sein. Mit der Zusendung eines Beitrags ist das Einverständnis zur Veröffentlichung im Rundbrief verbunden. Jeder Beitrag wird ohne Begutachtung veröffentlicht.

|                    |                                                                        |                              |
|--------------------|------------------------------------------------------------------------|------------------------------|
| <b>Herausgeber</b> | Fachausschuss <i>Management der Anwendungsentwicklung und -wartung</i> |                              |
| <b>Auflage</b>     | 500                                                                    |                              |
| <b>Redaktion</b>   | Christian Kop                                                          | E-mail: christian.kop@aau.at |
|                    | Institut für Artificial Intelligence and                               | Tel.: +43 463 2700 3735      |
|                    | Cyber Security                                                         | Fax: +43 463 2700 993735     |
|                    | Universität Klagenfurt                                                 |                              |
|                    | A-9020 Klagenfurt                                                      |                              |

Redaktionsschluß für das nächste Heft: 31.01.2027