

MASTERARBEIT

AKZEPTANZKRITERIEN MIT KI VERBESSERN

Produktdaten als Wissensbasis für höhere Anforderungsqualität

ausgeführt am



Studiengang

Informationstechnologien und Wirtschaftsinformatik

Von: Patrick Kainer B.Sc.

Personenkennzeichen: 52100751

Höf, am 10.12.2025

A handwritten signature in black ink that reads 'Kainer Patrick'. The signature is written in a cursive style with a horizontal dotted line underneath it.

Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich

- die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst,
- andere als die angegebenen Quellen nicht benutzt,
- die den Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht,
- und mich sonst keiner unerlaubten Hilfsmittel bedient habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Die vorliegende Fassung entspricht der eingereichten elektronischen Version.

Höf, 27.10.2025

Ort, Datum

Käner Patrick

Unterschrift des*der Studierenden

DANKSAGUNG

Zunächst möchte ich mich bei allen bedanken, die mich während meiner Masterarbeit begleitet, unterstützt und inspiriert haben.

Mein besonderer Dank gilt meinem Betreuer, Dipl.-Ing. Simon Jimenez, MA, für seine fachliche Unterstützung, die wertvollen Impulse und die offene Zusammenarbeit, die diese Arbeit bereichert haben.

Ein herzliches Dankeschön geht auch an mein Partnerunternehmen evon für die Möglichkeit, spannende Einblicke in die Praxis zu gewinnen und für die großartige Unterstützung während des Projekts.

Ganz besonders danke ich meiner Lebensgefährtin Katharina für ihre Geduld, ihre stetige Ermutigung und ihre Unterstützung, die mir auch in anspruchsvollen Phasen Rückhalt gegeben hat. Ihr Beistand war ein wertvoller Beitrag zum Gelingen dieser Arbeit.

Ein tief empfundener Dank geht an meine Familie, insbesondere an meine Eltern Anita und Gottfried, für ihre unermüdliche Unterstützung und Ermutigung. Ihre Liebe und ihr Verständnis waren die Säulen meines Erfolgs.

Abschließend gilt mein Dank dem Campus02 für die Bereitstellung der notwendigen Ressourcen und Möglichkeiten, die es mir ermöglicht haben, mein Studium erfolgreich zu absolvieren.

KURZFASSUNG

In dieser Arbeit wird ein Ansatz vorgestellt, der Methoden der Künstlichen Intelligenz (KI) zur automatisierten Erstellung von Akzeptanzkriterien für User Storys nutzt. Dazu wurde ein Prototyp entwickelt, der die Beschreibung einer User Story sowie zugehörige Dokumentationsinhalte in Vektoren überführt und mittels Clustering inhaltlich relevante Abschnitte identifiziert. Die gefundenen Dokumentationssegmente werden anschließend gemeinsam mit der User Story in einem Prompt an das Sprachmodell ChatGPT-5-mini übergeben, das darauf basierend geeignete Akzeptanzkriterien generiert.

Die Literaturrecherche umfasste verschiedene Aspekte des Requirements-Engineering-Prozesses und deren Verbindung zu KI-basierten Methoden. Diese Analyse bildete die Grundlage für die Auswahl der eingesetzten Technologien. Zudem wurden potenzielle Herausforderungen bei der Anwendung von KI im Requirements Engineering diskutiert. Besonders Themen wie Halluzinationen, Urheberrecht, ethische Fragestellungen und Compliance-Anforderungen erweisen sich als zentrale Aspekte, die zwingend eine menschliche Kontrolle und Bewertung erfordern.

Die Evaluation des entwickelten Prototyps zeigte, dass KI-basierte Ansätze die Erstellung von Akzeptanzkriterien in mehrfacher Hinsicht unterstützen und verbessern können. Nutzerbefragungen sowie maschinelle Analysen mittels BERTScore belegten eine Steigerung der Verständlichkeit, Einheitlichkeit und Prüfbarkeit der generierten Kriterien. Somit kann KI nicht nur als Werkzeug zur Qualitätssicherung, sondern auch als methodische Unterstützung für Requirements Engineers dienen. Gleichzeitig wurde deutlich, dass die fachliche Bewertung und Kontextinterpretation durch menschliche Experten unverzichtbar bleibt, um die technische Relevanz und organisatorische Passung der Ergebnisse sicherzustellen.

Zusammenfassend belegt die Arbeit, dass der Einsatz von Künstlicher Intelligenz, insbesondere in Kombination mit domänenspezifischen Produktdaten, einen Beitrag zur Steigerung der Anforderungsqualität leisten kann. KI-Systeme ermöglichen die Automatisierung von Routineaufgaben im Requirements Engineering, erhöhen die Nachvollziehbarkeit von Anforderungen und fördern die objektivere sowie reproduzierbarere Erstellung von Akzeptanzkriterien. Damit entsteht eine neue Schnittstelle zwischen datengetriebener Analyse und menschlicher Expertise, die langfristig zu einer höheren Reife und Effizienz in der Softwareentwicklung führen kann.

ABSTRACT

This thesis presents an approach that applies methods of Artificial Intelligence (AI) to automatically generate acceptance criteria for user stories. For this purpose, a prototype was developed that transforms the description of a user story and its related documentation into vector representations and uses clustering to identify semantically relevant sections. The resulting documentation fragments are then combined with the user story and passed as a prompt to the ChatGPT-5-mini language model, which generates suitable acceptance criteria based on this input.

The literature review covered various aspects of the requirements engineering process and their connection to AI-based techniques. This analysis informed the selection of the technologies employed. In addition, potential challenges arising from the use of AI in requirements engineering were discussed. Topics such as hallucinations, copyright, ethical considerations, and compliance requirements emerged as critical factors that necessitate human oversight and evaluation.

The evaluation of the developed prototype demonstrated that AI-based approaches can support and improve the creation of acceptance criteria in multiple ways. User surveys and machine analyses using BERTScore indicated increased clarity, consistency, and verifiability of the generated criteria. Consequently, AI can serve not only as a tool for quality assurance but also as methodological support for requirements engineers. At the same time, the results underscore that expert assessment and contextual interpretation remain indispensable to ensure the technical relevance and organizational fit of the outcomes.

In summary, the thesis shows that the use of Artificial Intelligence, particularly in combination with domain-specific product data, can contribute to improving requirements quality. AI systems enable the automation of routine tasks in requirements engineering, increase the traceability of requirements, and promote a more objective and reproducible creation of acceptance criteria. This establishes a new interface between data-driven analysis and human expertise that can, in the long term, lead to greater maturity and efficiency in software development.

INHALTSVERZEICHNIS

1	EINLEITUNG	1
1.1	Ausgangssituation	1
1.2	Zielsetzung und Forschungsfrage	1
1.2.1	Forschungsfrage	2
1.2.2	Ziele	2
1.2.3	Nicht Ziele	2
1.3	Struktur der Arbeit	2
1.4	Zielgruppe	3
2	LITERATURRECHERCHE PROZESS	4
2.1	Phase 1: Umfang der Literaturrecherche	4
2.2	Phase 2: Konzeptionierung der Suche	5
2.2.1	KI-gestützte Suche	5
2.3	Phase 3: Vorwärts- und Rückwärtssuche	7
2.4	Phase 4: Evaluierung der Literatur	8
2.5	Auswertung	12
3	REQUIREMENTS ENGINEERING	13
3.1	Bestandteile des Requirements-Engineering-Prozesses	13
3.2	Qualitätskriterien	14
3.3	Anforderung im agilen Umfeld	15
3.3.1	Epic	15
3.3.2	Features	15
3.3.3	User Story	15
4	KÜNSTLICHE INTELLIGENZ	17
4.1	Grundlagen Künstliche Intelligenz	17
4.1.1	Natural Language Processing	18
4.1.2	Large Language Models	19
4.1.3	Retrieval Augmented Generation	19

4.1.4	Prompt Optimierung	20
4.1.5	Fine Tuning	21
4.2	Künstliche Intelligenz und Anforderung	21
4.2.1	Anforderungserhebung	22
4.2.2	Analyse und Modellierung	23
4.2.3	Anforderungsspezifikation	23
4.2.4	Validierung	24
4.2.5	Anforderungsmanagement	25
4.3	Herausforderungen bei der Nutzung Künstlicher Intelligenz	26
4.3.1	Bias und Fairness	26
4.3.2	Ethik und Compliance	27
4.3.3	Sicherheit und Datensicherheit	28
4.3.4	Halluzinationen	29
4.3.5	Urheberrecht	31
5	UMSETZUNG	32
5.1	Design Science Research	32
5.2	Umsetzungskonzept	34
5.2.1	Anforderungen	34
5.2.2	Prozess der Datenverarbeitung	35
5.3	Technologie	36
5.3.1	Programmiersprache	36
5.3.2	Einlesen von Dokumenten	36
5.3.3	Verwendete KI	37
5.3.4	Vektorisierungsverfahren	37
5.3.5	Suchverfahren	38
5.3.6	Prompts	39
6	EVALUIERUNG DES ARTEFAKTS	40
6.1	Ergebnisse	40
6.1.1	User Story 1	40
6.1.2	User Story 2	42
6.1.3	User Story 3	44
6.2	Methoden der Evaluierung	46

6.2.1	Fragebogen	47
6.2.2	Maschinelle Validierung.....	48
6.3	Auswertung Fragebogen	49
6.3.1	Beschreibung der Stichprobe	50
6.3.2	Beschreibung der Ergebnisse	53
6.4	Maschinelle Auswertung.....	60
6.5	Interpretation Freitextkommentare	61
6.5.1	User Story 1	61
6.5.2	User Story 2.....	62
6.5.3	User Story 3.....	62
6.5.4	Zusammenfassung	63
6.6	Interpretation der Ergebnisse	64
6.7	Eingrenzung der Erkenntnisse	65
7	ZUSAMMENFASSUNG	67
7.1	Zusammenfassung	67
7.2	Kritische Reflexion	68
7.3	Ausblick	68
	ABKÜRZUNGSVERZEICHNIS.....	70
	ABBILDUNGSVERZEICHNIS	71
	TABELLENVERZEICHNIS	72
	LITERATURVERZEICHNIS	73

1 EINLEITUNG

Softwarespezifikationen zählen zu den zeitintensivsten und zugleich erfolgskritischsten Aspekten der Softwareentwicklung, führen aber häufig nicht zu den Ergebnissen, die den Erwartungen der Endnutzer entsprechen (Wei et al. 2023, S. 1).

Der wirtschaftliche Schaden, der durch mangelhafte Software verursacht wird, beläuft sich jährlich auf rund 2,08 Billionen US-Dollar (Berger 2023). Etwa 28 % aller Fehler, die im Verlauf eines Softwareprojekts auftreten, sind auf unvollständige oder mehrdeutige Anforderungen zurückzuführen (Mogyorodi).

1.1 Ausgangssituation

Der Prozess der Anforderungsgewinnung wird derzeit noch überwiegend manuell und somit maßgeblich durch menschliche Akteure und Akteurinnen durchgeführt (Zhao et al. 2022, S. 1).

Zwar existieren etablierte Verfahren und Methoden zur Erfassung und Bearbeitung von Anforderungen, wie beispielsweise SCRUM, Kanban und andere agile Frameworks. Allerdings berücksichtigen diese Frameworks keine Ansätze zur Prozessautomatisierung im Sinne einer automatisierten Generierung von Anforderungen (Bergsmann 2018, 25ff).

Auch im Bereich der automatisierten Generierung von Anforderungen existieren bereits erste Forschungsvorhaben, wie beispielsweise die Arbeit von Wei et al. (2023) mit dem Titel "An Automatic Generation and Verification Method of Software Requirements Specification". Darüber hinaus sind bereits erste Werkzeuge auf dem Markt verfügbar, die die automatische Erstellung von Anforderungen unterstützen, wie zum Beispiel Storywise. Das den gesamten Erstellungsprozessen von Anforderungen zu fertigen User Storys, Epics, und Akzeptanzkriterien unterstützt (Knuplesch 2024, 40ff). Allerdings fokussieren sich weder die genannten Forschungsvorhaben noch die existierenden Produkte in dieser Detailtiefe auf die automatisierte Generierung von Akzeptanzkriterien. Dieses Themenfeld bleibt bislang weitgehend unerschlossen.

1.2 Zielsetzung und Forschungsfrage

Der zunehmende Einsatz von Künstlicher Intelligenz in Unternehmen wirft die Frage auf, in welchen Bereichen ihr Potenzial noch ungenutzt bleibt. Besonders in der Anforderungsanalyse, etwa bei der Formulierung von Akzeptanzkriterien, zeigt sich bislang wenig KI-Unterstützung. Abschnitt 1.2 stellt daher die zentrale Forschungsfrage sowie die Zielsetzung und Abgrenzung dieser Arbeit vor.

1.2.1 Forschungsfrage

Die Bedeutung von Künstlicher Intelligenz (KI) in Unternehmen nimmt kontinuierlich zu. Im Jahr 2024 gaben 20 % der Unternehmen an, KI einzusetzen; dies entspricht nahezu einer Verdopplung gegenüber dem Vorjahr, in dem lediglich 11 % der Unternehmen den Einsatz von KI vermeldeten (Statistik Austria 16.10.2024)

Trotz dieses starken Trends verwenden immer noch die wenigsten Unternehmen Künstliche Intelligenz zum Generieren oder Verbessern ihrer Anforderungen. Daraus ergibt sich meine Forschungsfrage:

Wie können Dokumentationen in Kombination mit Künstlicher Intelligenz zur Qualitätserhöhung von Akzeptanzkriterien von User Stories genutzt werden?

1.2.2 Ziele

Ziel dieser Arbeit ist es, zu evaluieren, inwieweit diese Form der Unterstützung in der Anforderungsanalyse sinnvoll einsetzbar ist und ob sie einen tatsächlichen Mehrwert bei der Erhebung von Anforderungen bietet.

1.2.3 Nicht Ziele

Es ist nicht Ziel dieser Arbeit, ein marktreifes Produkt zu entwickeln, das unmittelbar von Unternehmen eingesetzt wird. Vielmehr könnte eine solche Nutzungsperspektive Teil einer weiterführenden Anforderung oder eines anschließenden Entwicklungsschritts sein.

1.3 Struktur der Arbeit

Die vorliegende Arbeit beginnt mit einer thematischen Einführung, die den Rahmen und die Zielsetzung des Forschungsprojekts absteckt. Daran anschließend erfolgt eine Auseinandersetzung mit den Grundlagen des Requirements Engineering, wobei zentrale Begriffe, Prozesse und Methoden zur Erhebung und Verwaltung von Anforderungen dargestellt werden. Im darauffolgenden Abschnitt steht die Künstliche Intelligenz im Fokus. Zunächst werden grundlegende Konzepte und Ansätze der Künstlichen Intelligenz erläutert. Darauf aufbauend wird untersucht, wie KI-Technologien im Kontext von Anforderungen eingesetzt werden können, insbesondere mit Blick auf deren Potenzial zur Unterstützung und Automatisierung im Requirements Engineering. Im Anschluss werden spezifische Herausforderungen beleuchtet, die mit dem Einsatz von KI in diesem Bereich einhergehen, etwa in Bezug auf Transparenz, Nachvollziehbarkeit oder Datenqualität. Der darauffolgende Teil beschreibt die Umsetzung des gewählten Design Science Research (DSR)-Ansatzes, welcher als methodischer Rahmen zur Entwicklung und Evaluation des entworfenen Artefakts dient. Abschließend erfolgt eine kritische Reflexion der Ergebnisse sowie eine zusammenfassende Bewertung der Erkenntnisse dieser Arbeit.

1.4 Zielgruppe

Diese Masterarbeit richtet sich an Requirements Engineers und Product Owner, die in ihrem beruflichen Kontext den Einsatz von Künstlicher Intelligenz (KI) in Betracht ziehen oder bereits erste Schritte in diese Richtung unternommen haben. Ziel ist es, aufzuzeigen, wie KI-Technologien sinnvoll in den Anforderungs- und Produktentwicklungsprozess integriert werden können, um Effizienz, Qualität und Innovation zu fördern. Die Arbeit liefert praxisnahe Impulse und theoretische Grundlagen für Fachkräfte, die die Potenziale von KI systematisch erschließen und in ihre tägliche Arbeit einbinden möchten.

2 LITERATURRECHERCHE PROZESS

In diesem Kapitel erfolgt eine detaillierte Darstellung des Verfahrens der systematischen Literaturrecherche. Der erste Abschnitt widmet sich einer theoretischen Einführung in die Methodik der Recherche. Im darauffolgenden Abschnitt wird spezifisch auf die angewandte Vorgehensweise im Rahmen dieser wissenschaftlichen Arbeit eingegangen.

Für diese Arbeit wurde das Vorgehen von Brocke et al. verwendet. Dieses Vorgehen teilt sich grundsätzlich in 4 Phasen.

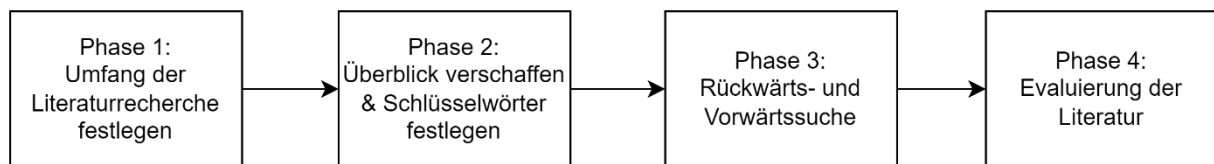


Abbildung 1: Literatur Recherche Prozess

2.1 Phase 1: Umfang der Literaturrecherche

Im initiativen Segment der Literaturrecherche wird primär das Untersuchungsfeld eingegrenzt. Hierzu dient die Taxonomie von Cooper als maßgebliches Instrument. Diese Taxonomie definiert die Bandbreite der Untersuchung und stellt gleichzeitig ein Framework zur strukturierten Organisation des Suchprozesses bereit, indem vier zentrale Parameter festgelegt werden.

Bei der Herangehensweise an den Literaturrechercheprozess differenziert man grundlegend zwischen sequenziellen und iterativen Ansätzen. Ein sequenzielles Vorgehen legt von Beginn an eine festgezurrte Suchstrategie fest, die auch bei nachträglichen Erkenntnissen unverändert bleibt. Konträr dazu passt sich der iterative Prozess dynamisch neuen Erkenntnissen an und wiederholt den Suchvorgang fortlaufend. In Anbetracht der Tatsache, dass es keinen branchenweit akzeptierten Konsens zu diesem gewählten Thema gibt, wurde in dieser Arbeit der iterative Ansatz präferiert, um Flexibilität im Rechercheverlauf zu gewährleisten.

Hinsichtlich der Quellendiversifizierung unterscheidet man primär zwischen Zitationsdatenbanken, bibliografischen Datenbanken und individuellen Publikationen. Für die vorliegende Arbeit wurden alle genannten Kategorien eingehend analysiert, wobei „Google Scholar“ als zentrale Datenbank herangezogen wurde.

Für die Abdeckung der Recherche wird ein selektiver Ansatz verwendet. Es wird versucht bedeutsame und repräsentative Literatur zu finden. Eine ganzheitliche Betrachtung der Literatur ist aufgrund der Menge an Literatur zu diesem Thema nicht möglich.

Abschließend, aber nicht minder relevant, ist die angewandte Technik innerhalb der Literaturrecherche. Man differenziert hier zwischen Vorwärtssuche, Rückwärtssuche und Schlüsselwortsuche. Während die Schlüsselwortsuche gezielt nach relevanten Wörtern in Datenbanken sucht, überprüft die Rückwärtssuche die Referenzen einer Publikation. Die

Vorwärtssuche hingegen fokussiert sich darauf, in welchen Veröffentlichungen eine gegebene Arbeit zitiert wurde. Für die vorliegende Untersuchung kamen sämtliche Techniken zum Einsatz, welche in den Phasen 2 und 3 vertiefend erläutert werden. (vom Brocke et al., 2015, 215f)

Prozess	Sequenziell		Iterativ
Quellen	Zitationsdatenbanken	Bibliografische Datenbanken	Publikationen
Abdeckung	Umfassend	Repräsentativ	Grundlegende Werke
Technik	Schlüsselwortsuche	Rückwärtssuche	Vorwärtssuche

Tabelle 1: Taxonomie nach Cooper

2.2 Phase 2: Konzeptionierung der Suche

Im Verlauf dieser wissenschaftlichen Auseinandersetzung wurde die bestehende Literatur analysiert. Hierfür wurde überwiegend die Methode der Schlüsselwortsuche angewandt, um aussagekräftige wissenschaftliche Beiträge und Referenzen zu identifizieren.

Nach dieser initialen Durchsuchung folgte eine Analysephase, in der nicht nur Synonyme, sondern auch thematisch verwandte Termini ausfindig gemacht und systematisch kategorisiert wurden. Zur visuellen Unterstützung und klaren Darstellung der in diesem Forschungsprozess angewandten Suchstrategie sind sowohl die zentralen Schlüsselwörter als auch die hierzu passenden Synonyme und assoziierten Begriffe in der anschließenden Grafik dargestellt.

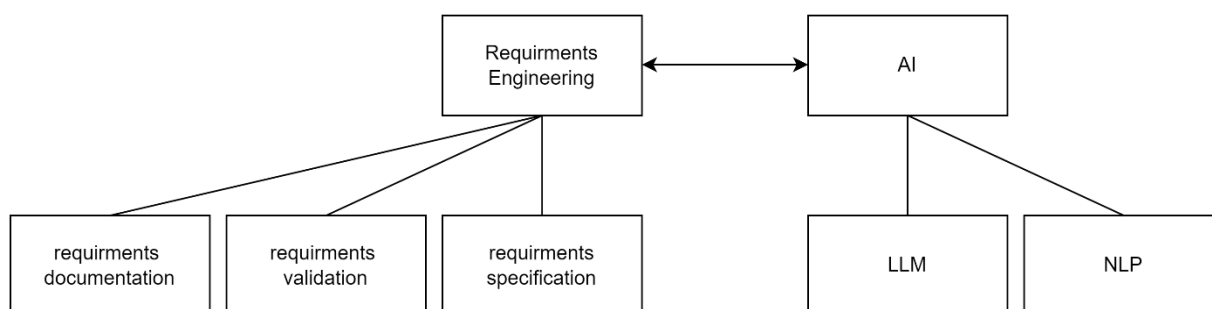


Abbildung 2: Clusterung Literaturrecherche

Im Zuge dieses Clustering-Prozesses kristallisierten sich bestimmte Suchbegriffe heraus. Es wurde bewusst die Entscheidung getroffen, englische Termini zu verwenden, da im Bereich der Requirements Engineering vornehmlich auf Englisch publiziert wird und somit ein breiteres Spektrum an relevanten Fachpublikationen erschlossen werden kann (vom Brocke et al., 2015, S. 215).

2.2.1 KI-gestützte Suche

Zudem wurde Künstliche Intelligenz zur Unterstützung der Literaturrecherche eingesetzt. Die spezifischen Abfragen (Prompts) wurden mithilfe von ChatGPT-4 in Kombination mit der Deep Search-Funktion erstellt.

Prompt 1

Ich: „Das ist die Forschungsfrage meiner Masterarbeit. Suche mir Statistiken und Literatur, die für diese Arbeit relevant sein könnte Welche Methoden eignen sich zur Extraktion und Integration von Feedback-Daten aus Produktbewertungen und Support-Tickets zur Steigerung der Vollständigkeit und Relevanz (automatisch generierter) Anforderungen?“

ChatGPT:

1. Planst du deinen Fokus eher auf technische Methoden (wie NLP, Sentiment Analysis, Topic Modeling, etc.) oder auch auf organisatorische Aspekte (wie Prozesse zur Anforderungsintegration)?
2. Gibt es bestimmte Branchen oder Produkttypen (z. B. Software, E-Commerce, Konsumgüter), auf die du dich beziehst?
3. Interessieren dich eher wissenschaftliche Quellen (z. B. IEEE, ACM, Springer) oder auch praxisorientierte Berichte, Whitepapers oder Marktstudien?
4. Möchtest du deutsch- oder englischsprachige Literatur (oder beides) bevorzugen?

Ich: „1. Eher technisch 2. Software 3. Quellen sollten bevorzugt frei zugänglich sein“

Prompt 2:

Ich: „Am häufigsten genutzte LLMs, um Requirements Engineering zu unterstützen. Bitte direkte Links zu den Quellen. Die Quellen sollen passend für eine Masterarbeit sein“

ChatGPT: „Könntest du bitte noch etwas genauer beschreiben, worauf du beim Einsatz von LLMs im Requirements Engineering fokussierst? Zum Beispiel:

- Geht es dir um Tools, die LLMs nutzen, um Anforderungen zu analysieren, zu generieren oder zu validieren?
- Soll der Fokus auf akademischer Forschung oder industriellen Anwendungen liegen?
- Bevorzugst du deutschsprachige Quellen oder sind englische wissenschaftliche Publikationen auch in Ordnung?
- Gibt es ein bestimmtes Zeitfenster (z. B. nur Veröffentlichungen ab 2023)?

Je mehr Details du mir gibst, desto gezielter kann ich die passenden Quellen für deine Masterarbeit finden.“

Ich: 1. Nein es geht mir wirklich um Modelle, die verwendet werden wie z. B. LLM oder NLP bzw. andere 2. akademischer Forschung 3. Englisch und Deutsch 4. Nicht vor 2015

Prompt 3:

Ich: „Am häufigsten genutzte LLMs, um Requirements Engineering zu unterstützen. Bitte direkte Links zu den Quellen. Die Quellen sollen passend für eine Masterarbeit sein“

ChatGPT: „Könntest du bitte noch etwas genauer beschreiben, worauf du beim Einsatz von LLMs im Requirements Engineering fokussierst? Zum Beispiel:

- Geht es dir um Tools, die LLMs nutzen, um Anforderungen zu analysieren, zu generieren oder zu validieren?
- Soll der Fokus auf akademischer Forschung oder industriellen Anwendungen liegen?
- Bevorzugst du deutschsprachige Quellen oder sind englische wissenschaftliche Publikationen auch in Ordnung?
- Gibt es ein bestimmtes Zeitfenster (z. B. nur Veröffentlichungen ab 2023)?

Je mehr Details du mir gibst, desto gezielter kann ich die passenden Quellen für deine Masterarbeit finden.“

Ich: „1. Nein es geht mir wirklich um Modelle, die verwendet werden wie z. B. LLM oder NLP bzw. andere 2. akademischer Forschung 3. Englisch und Deutsch 4. Nicht vor 2015“

2.3 Phase 3: Vorwärts- und Rückwärtssuche

Im Rahmen meines ersten Suchlaufs für diese Masterarbeit stieß ich auf zwei Fachveröffentlichungen: „Automated Generating Natural Language Requirements based on Domain Ontology“ und „Generative AI for Requirements Engineering: A Systematic Literature Review“. Diese wurden aufgrund der breiten Abdeckung des Themas für die Vor- und Rückwärtssuche ausgewählt.

Um den Prozess und die Zusammenhänge der durchgeführten Literatursuche zu visualisieren, habe ich eine Abbildung erstellt. In dieser Abbildung verdeutlichen die Pfeilrichtungen stets den Ursprung der jeweiligen Suche.

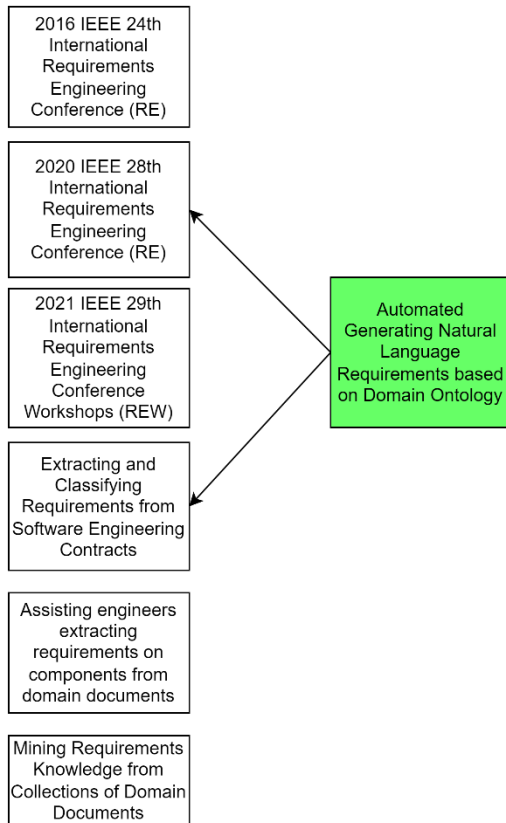


Abbildung 3: Vorwärts- und Rückwärtssuche 1

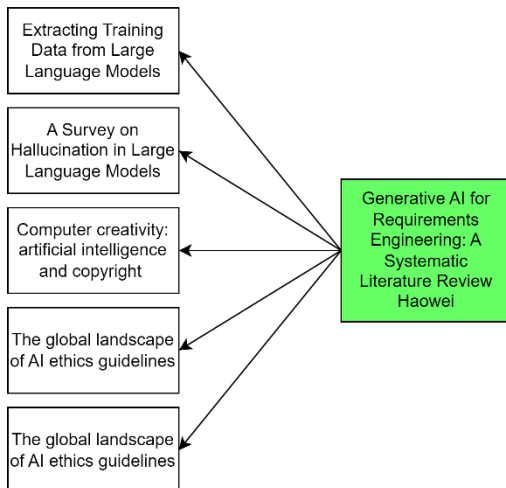


Abbildung 4: Vorwärts und Rückwärtssuche 2

2.4 Phase 4: Evaluierung der Literatur

In Phase 4 gefundene Literatur wurde einer Überprüfung hinsichtlich ihrer Relevanz unterzogen. Diese Betrachtung erfolgte mithilfe klar definierter Einbeziehungs- und Ausschlusskriterien. Für diesen Prozess wurde ein Whitelist-Verfahren angewandt: Literatur, die nicht den festgelegten Einbeziehungskriterien entsprach, wurde konsequent aus der Untersuchung ausgeschlossen.

Die Einbeziehungskriterien sind:

1. Der Artikel konzentriert sich insbesondere auf die Schnittstelle zwischen Requirements Engineering und Künstlicher Intelligenz. Hierbei könnten Aspekte wie Best Practices, potenzielle Herausforderungen, Vorteile, einschlägige Werkzeuge oder spezialisierte Techniken im Fokus stehen, die in diesem speziellen Kontext relevant sind.
2. Zeitraum der Veröffentlichung: Angesichts der schnellen Entwicklung in der Softwareentwicklung und Softwarearchitektur werden nur Publikationen aus den letzten 15 Jahren betrachtet.

Hierbei wurden allerdings Artikel ausgeschlossen, welche in der Einleitung oder im Literaturrechercheprozess verwendet wurden.

In dieser Literaturarbeit wurden ursprünglich 79 Papers betrachtet. Nach einer sorgfältigen Prüfung wurden 33 Papers aufgrund ihres Themas und 4 weitere wegen thematischer Unstimmigkeiten ausgeschlossen. Somit basiert die Analyse auf 42 qualitativ ausgewählten Papers, die sowohl thematisch relevant als auch aktuell sind.

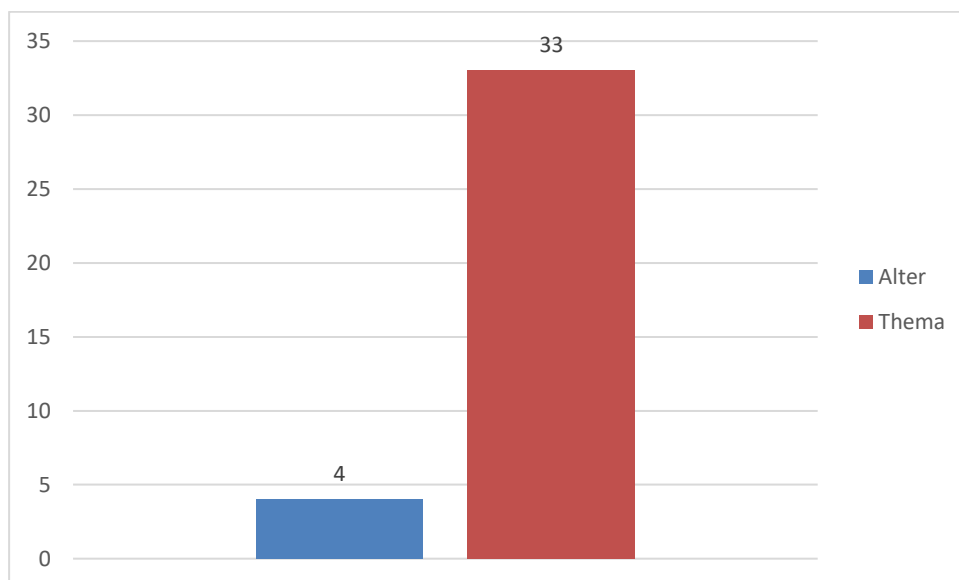


Abbildung 5: Evaluierung der Literatur

Die Darstellung der verwendeten Literaturarten in der vorliegenden Arbeit erfolgt in der nachstehenden Abbildung. Diese visualisiert die Aufteilung aufgrund der Publikationsart, zum Beispiel Fachzeitschrift, Konferenzbeitrag und viele mehr. Diese strukturierte Darstellung ermöglicht eine klare Übersicht über die Zusammensetzung und Diversität der für die Analyse herangezogenen Literaturquellen.

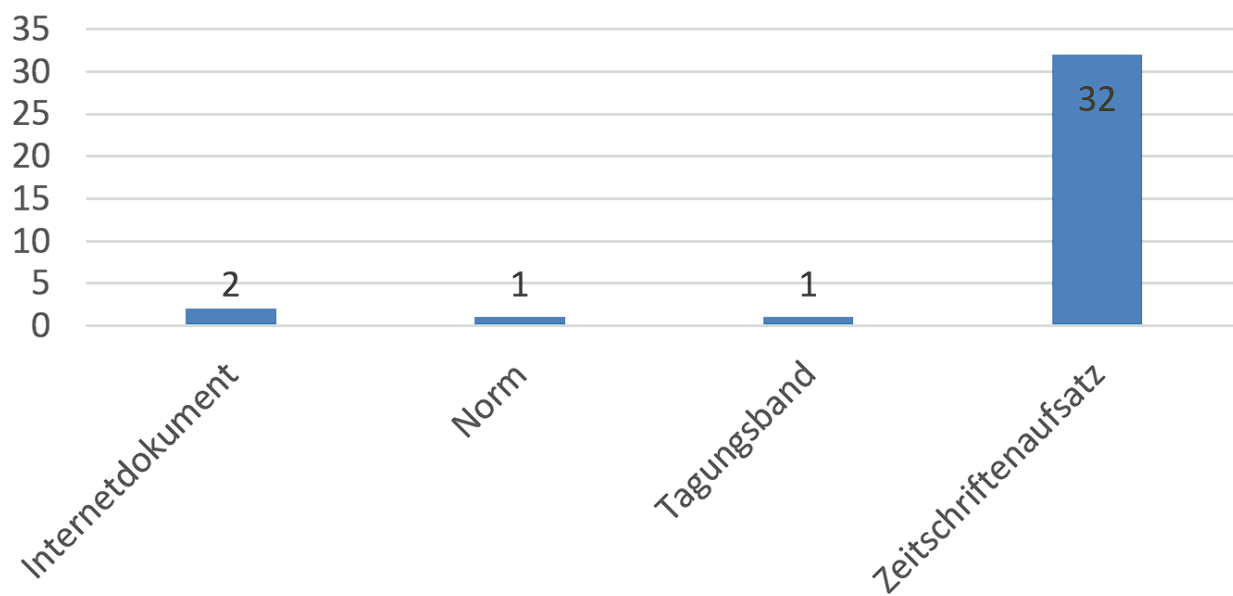


Abbildung 6: Art der Literatur

Die dynamische Natur des Themenfelds „Requirements Engineering in der KI“ erfordert eine besondere Aufmerksamkeit hinsichtlich der Aktualität der verwendeten Literaturquellen. In Anbetracht der rapiden Entwicklungen in diesem Bereich wurde ein besonderes Augenmerk auf die Auswahl möglichst aktuelle Forschungsarbeiten gelegt. Das Ergebnis dieser Selektion ist, dass 50% der in dieser Arbeit verwendeten Literatur nicht älter als 3 Jahre sind. Dies unterstreicht das Bestreben, die neuesten Trends, Technologien und Methodiken in diesem Bereich zu berücksichtigen und die Relevanz sowie die praktische Anwendbarkeit der Forschungsergebnisse zu gewährleisten.

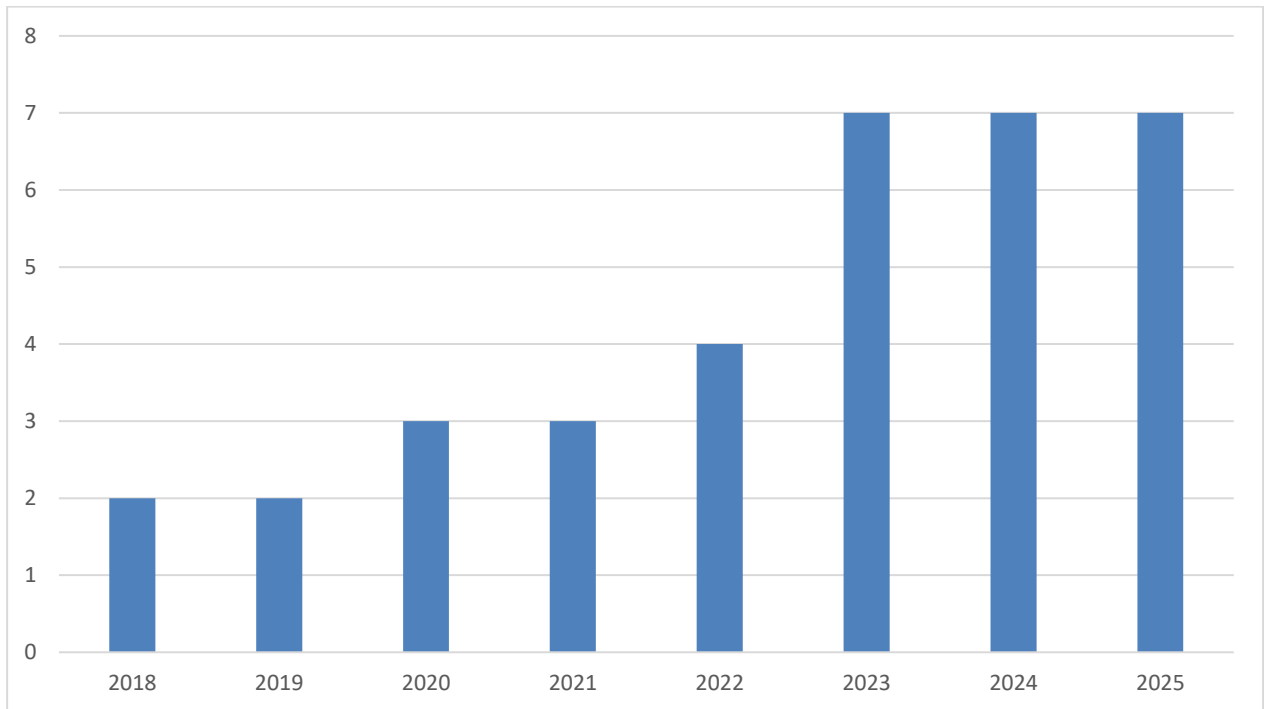


Abbildung 7: Altersverteilung Literatur

2.5 Auswertung

Zur Identifikation relevanter Literatur wurde ein mehrstufiges Vorgehen gewählt. Zunächst erfolgte die Recherche über verschiedene Zugänge: eine manuelle Suche ($n = 10$), die Nutzung Künstlicher Intelligenz ($n = 8$), eine Keyword-Suche in Datenbanken ($n = 50$) sowie eine Vorwärts- und Rückwärtssuche in Referenzlisten ($n = 11$). Insgesamt konnten auf diese Weise 79 Publikationen identifiziert werden.

Im nächsten Schritt wurde eine Altersüberprüfung vorgenommen, wodurch vier Publikationen ausgeschlossen wurden, sodass 75 Arbeiten verblieben. Anschließend erfolgte eine thematische Überprüfung. Dabei wurden 33 Arbeiten aufgrund mangelnder thematischer Relevanz ausgeschlossen. Schließlich ergab sich eine Auswahl von 42 Publikationen, die für die weitere Bearbeitung in dieser Arbeit verwendet wurden.

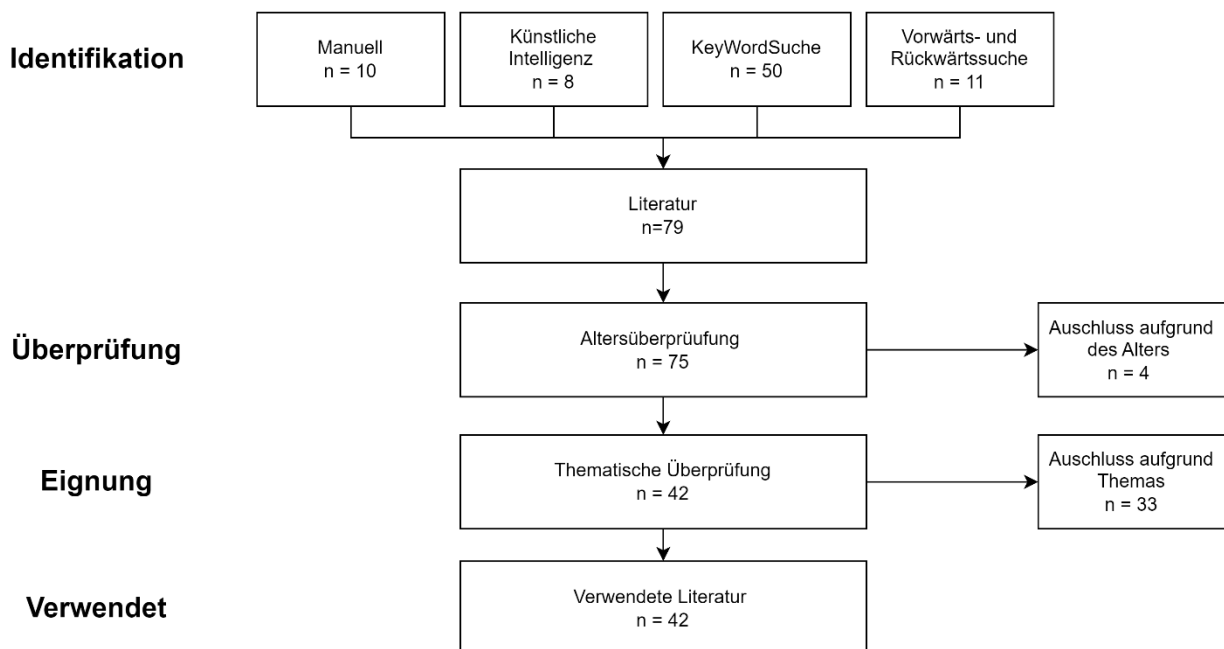


Abbildung 8: PRISMA Flowchart

3 REQUIREMENTS ENGINEERING

In der Literatur wird Requirements Engineering (RE) als ein Prozess beschrieben, der das Erheben, Analysieren, Dokumentieren, Validieren und Verwalten von Softwareanforderungen umfasst. Ziel dieses Prozesses ist es, sicherzustellen, dass die entwickelte Software den Bedürfnissen und Erwartungen der jeweiligen Stakeholder entspricht (Sharma und Pandey 2013, S. 2).

3.1 Bestandteile des Requirements-Engineering-Prozesses

Im Rahmen des Requirements Engineerings (RE) werden typischerweise mehrere systematische Arbeitsaufgaben durchgeführt, um Anforderungen an ein Softwaresystem klar, präzise und nachvollziehbar zu spezifizieren. Zu den zentralen Aufgaben zählen dabei:

1. Ermittlung

Diese erste Phase umfasst die systematische Sammlung und Erhebung der Anforderungen durch Interaktion mit Stakeholdern. Typische Methoden hierfür sind Stakeholder-Interviews, Workshops, Szenarien Entwicklung, Beobachtungstechniken und Fragebögen. Ziel ist es, sowohl explizite als auch implizite Anforderungen zu identifizieren (Pohl und Rupp 2021, 125ff).

2. Dokumentation

Die gesammelten Anforderungen werden anschließend klar, vollständig und konsistent dokumentiert. Hierfür werden etablierte Techniken und Formate genutzt, darunter textuelle Beschreibungen, Anwendungsfälle (Use Cases), User Stories sowie formale oder semi-formale Modellierungstechniken wie UML-Diagramme (Chris Rupp 2014, 98ff).

3. Prüfung und Validierung

Die dokumentierten Anforderungen werden systematisch auf ihre Vollständigkeit, Konsistenz, Korrektheit und Verständlichkeit geprüft. Techniken wie Reviews, Inspektionen, Prototypen oder Simulation werden angewandt, um sicherzustellen, dass die Anforderungen die tatsächlichen Wünsche und Ziele der Stakeholder richtig widerspiegeln (ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes 2015, 74ff).

4. Verhandlung und Abstimmung

Konflikte zwischen Stakeholdern und Anforderungen werden in dieser Phase identifiziert und durch gezielte Abstimmungsgespräche gelöst. Der Fokus liegt darauf, eine für alle Beteiligten akzeptable Balance zwischen oft widersprüchlichen Anforderungen zu finden und Kompromisse transparent festzuhalten (Pohl und Rupp 2021, 136ff).

5. Verwaltung und Pflege

Das Management der Anforderungen umfasst die strukturierte Verwaltung, Versionierung und Nachverfolgung aller Anforderungen über den gesamten Projektlebenszyklus. Hierbei werden Anforderungen auf Veränderungen überprüft und diese Änderungen systematisch dokumentiert

und kommuniziert. Die Nachvollziehbarkeit (Traceability) zwischen Anforderungen und deren Umsetzung wird durch geeignete Tools, wie Jira oder Azure DevOps, unterstützt und gewährleistet (Pohl und Rupp 2021, 183ff).

6. Priorisierung und Klassifikation

Anforderungen werden hinsichtlich ihrer Relevanz, Dringlichkeit und Umsetzungskosten priorisiert und kategorisiert. Ziel ist es, die wichtigsten Anforderungen zuerst umzusetzen und ein transparentes Vorgehen bei Ressourcenengpässen sicherzustellen (Pohl und Rupp 2021, 205ff).

7. Qualitätssicherung der Anforderungen:

Die Sicherstellung der Qualität umfasst den Einsatz von Qualitätskriterien und Checklisten, beispielsweise auf Basis des ISO/IEC 25010 Qualitätsmodells. Dadurch wird gewährleistet, dass Anforderungen hinsichtlich Klarheit, Testbarkeit, Vollständigkeit und Konsistenz definierte Standards erfüllen (Pohl und Rupp 2021, 114ff).

Zusammenfassend bildet die systematische Durchführung dieser Aufgaben die Grundlage für eine erfolgreiche Softwareentwicklung, indem sichergestellt wird, dass Anforderungen frühzeitig und vollständig erfasst, konsistent dokumentiert, angemessen validiert und effizient verwaltet werden.

3.2 Qualitätskriterien

Damit Anforderungen im Rahmen eines Softwareentwicklungsprojekts als qualitativ hochwertig gelten können, müssen sie bestimmte Kriterien erfüllen. Diese Qualitätsmerkmale dienen dazu, sicherzustellen, dass Anforderungen klar, nachvollziehbar und überprüfbar formuliert sind und somit als verlässliche Grundlage für die weitere Entwicklung dienen können. Im Folgenden werden zentrale Qualitätskriterien beschrieben, die in der Fachliteratur häufig genannt werden und als wesentliche Maßstäbe für gute Anforderungen gelten: Die Qualitätskriterien sind:

Angemessen: Die Anforderung beschreibt tatsächliche und mit den Stakeholdern abgestimmte Bedürfnisse (Glinz et al. 2020, 111ff).

Notwendig: Die Anforderung gehört zum relevanten Systemumfang und trägt somit zur Erreichung mindestens eines Ziels oder Bedürfnisses eines Stakeholders bei (Glinz et al. 2020, 111ff).

Eindeutig: Es besteht ein gemeinsames Verständnis der Anforderung, was bedeutet, dass alle Beteiligten sie in gleicher Weise interpretieren (Glinz et al. 2020, 111ff).

Vollständig: Die Anforderung ist in sich geschlossen, das heißt, es fehlen keine Bestandteile, die für ihr Verständnis erforderlich sind (Glinz et al. 2020, 111ff).

Verständlich: Die Anforderung ist für die Zielgruppe nachvollziehbar, das heißt, die Zielgruppe kann die Anforderung vollständig erfassen (Glinz et al. 2020, 111ff).

Überprüfbar: Die Erfüllung der Anforderung durch ein implementiertes System kann eindeutig überprüft werden, sodass Stakeholder oder Kundinnen und Kunden entscheiden können, ob die Anforderung erfüllt wurde oder nicht (Glinz et al. 2020, 111ff).

Im Rahmen dieser Arbeit liegt der Fokus insbesondere auf der Untersuchung des Einsatzes von Künstlicher Intelligenz im Kontext der Anforderungen an Überprüfbarkeit, Verständlichkeit und Vollständigkeit. Ziel ist es, zu analysieren, inwieweit KI-gestützte Ansätze zur Unterstützung oder Verbesserung dieser Qualitätsmerkmale von Anforderungen beitragen können.

3.3 Anforderung im agilen Umfeld

Im agilen Anforderungsmanagement werden Anforderungen auf unterschiedlichen Abstraktionsebenen erfasst. Die Begriffe Epic, Feature und User Story strukturieren dabei die Anforderungen von der Grob- zur Feinspezifikation. Dieses Kapitel erläutert deren jeweilige Rolle, Bedeutung und Einsatz im Entwicklungsprozess.

3.3.1 Epic

Ein Epic stellt im agilen Umfeld eine übergeordnete Anforderung dar, aus der in weiterer Folge konkrete User Stories abgeleitet werden. In dieser Anforderungsebene liegt der Fokus ausschließlich auf inhaltlichen Aspekten. Operative Details, sowie Akzeptanzkriterien werden in diesem Stadium noch nicht berücksichtigt. Die weitere Detaillierung erfolgt im Rahmen der Features und User Stories sowie ergänzender Anforderungsartefakte. Aufgrund des hohen Abstraktionsniveaus und den fehlenden Akzeptanzkriterien werden Epics in dieser Arbeit nicht weiter betrachtet (Bergsmann 2018, 75f).

3.3.2 Features

Ein Feature beschreibt eine klar abgrenzbare Eigenschaft oder Funktionalität eines Systems, die für die Stakeholder einen erkennbaren Nutzen oder Mehrwert bietet. Vom Detailgrad her ist ein Feature unter einem Epic anzusiedeln aber über einer User Story. In der Regel umfasst ein Feature mehrere zugrunde liegende Anforderungen und dient insbesondere in der frühen Phase der Anforderungsanalyse sowie in der Kommunikation mit Stakeholdern als abstrahierte Beschreibung von Systemfunktionen. Es bewegt sich damit auf einer höheren Abstraktionsebene als einzelne Anforderungen und unterstützt die strukturierte Planung und Priorisierung im Entwicklungsprozess. Aufgrund des hohen Abstraktionsniveaus werden Akzeptanzkriterien auf der Ebene der Features in dieser Arbeit nicht weiter betrachtet (Bergsmann 2018, 2018, 112ff).

3.3.3 User Story

User Stories dienen, wie der Name bereits andeutet, der Beschreibung von Funktionalitäten, die einen konkreten Mehrwert für eine Nutzerin bzw. einen Nutzer, ein System oder eine Software

schaffen. Sie stellen funktionale Anforderungen dar, die eine geeignete Grundlage für die Sprint-Planung im agilen Entwicklungsprozess bilden. Darüber hinaus dienen User Stories als Ausgangspunkt für die weitere Detaillierung funktionaler Anforderungen sowie für die Ableitung entsprechender Akzeptanztests. User Stories eignen sich nicht zur Darstellung komplexer Abläufe, sondern konzentrieren sich auf die Beschreibung einzelner, in sich geschlossener Funktionalitäten. Für weiterführende Detailinformationen, wie etwa Ablaufspezifikationen, Randbedingungen oder technische Umsetzungen, sollten ergänzende Artefakte wie Use Cases, Nutzungsszenarien, Constraints, UI-Beschreibungen, Schnittstellenspezifikationen oder andere dokumentarische Anhänge erstellt werden (Bergsmann 2018, 115ff).

Da diese Form der abstrahierten Darstellung von Anforderungen im weiteren Verlauf dieser Arbeit eine zentrale Rolle einnimmt, werden im Folgenden noch einige weiterführende Aspekte und Details näher betrachtet.

Beschreibung:

Die Beschreibung einer User Story ist aus Benutzersicht. Sie sollen den Kundinnen und Kunden einen konkreten Mehrwert bieten. Für ein Zeiterfassungssystem könnte das folgendermaßen aussehen (Bergsmann 2018, 115ff).

„Als Mitarbeiter möchte ich meine Tagesarbeitszeit und die Pausen erfassen, damit ich gegenüber dem Arbeitgeber und dem Arbeitszeitgesetz meine Arbeitszeiten nachweisen kann.“ (Bergsmann 2018, S. 118)

Akzeptanzkriterien:

In diesem Abschnitt der User Story sollte der Rahmen der Anforderung in groben Zügen und idealerweise in Stichpunkten dokumentiert werden. Eine detaillierte Testspezifikation ist zu diesem Zeitpunkt weder erforderlich noch effizient. Stattdessen dienen die Akzeptanzkriterien der Eingrenzung der wesentlichen Aspekte, die für die Abnahme der Story entscheidend sind. Auf Basis dieser Kriterien kann anschließend ein geeigneter Abnahmetest für das System entwickelt werden.

Für ein Zeiterfassungssystem könnte dies beispielhaft folgendermaßen dargestellt werden:

„Akzeptanzkriterien:

- *Kommt-/Geht-Zeit (versch. Kombinationen)*
 - *Pauseneingabe*
 - *Tagesarbeitsdauer-Berechnung*
 - *Interne Arbeitszeitregelungen*
- *Grenzen gem. Arbeitszeitgesetz“ (Bergsmann 2018, S. 118)*

Im praktischen Teil dieser Arbeit werden Akzeptanzkriterien zu ausgewählten User Stories mithilfe von KI generiert. Ziel ist es, zu untersuchen, inwiefern KI-basierte Ansätze die Qualität und Vollständigkeit von Akzeptanzkriterien unterstützen und somit zur Effizienz und Nachvollziehbarkeit im Anforderungsmanagement beitragen können.

4 KÜNSTLICHE INTELLIGENZ

KI ist aus dem modernen Alltag sowie aus zahlreichen digitalen Diensten nicht mehr wegzudenken. Anwendungen wie persönliche Sprachassistenten (z. B. Amazon Alexa, Google Assistant oder Apple Siri), Gesichtserkennungssysteme oder maschinelle Übersetzungsdienste veranschaulichen eindrucksvoll die Integration von KI in verschiedenste Lebensbereiche (Kreutzer und Sirrenberg 2019, S. 1).

Zunehmend findet KI auch Anwendung im Bereich der Anforderungsanalyse. Dieses Kapitel erläutert zunächst grundlegende Konzepte der KI sowie relevante Modelle. Anschließend wird der konkrete Einsatz von KI in den Aufgabenbereichen des Requirements Engineering analysiert. Dabei werden sowohl die Potenziale als auch die damit verbundenen Herausforderungen und Risiken systematisch betrachtet.

4.1 Grundlagen Künstliche Intelligenz

Um die Begriffe Künstliche Intelligenz zu verstehen, ist es zunächst notwendig, den Begriff der Intelligenz an sich zu klären. Menschliche Intelligenz wird häufig als „multiple Intelligenz“ beschrieben, da sie sich nicht auf einen einzelnen Bereich beschränkt, sondern verschiedene Fähigkeiten umfasst. Dazu zählen unter anderem sprachliche, musikalische sowie mathematisch-logische Intelligenz. Diese Vielschichtigkeit menschlicher Intelligenz bildet die Grundlage für die Einordnung und Abgrenzung Künstlicher Intelligenz (Sternberg und Kaufman 2011, 2011, 485ff).

Künstliche Intelligenz lässt sich auch definieren als die Fähigkeit von Maschinen, kognitive Aufgaben zu übernehmen, die typischerweise mit dem menschlichen Denkvermögen assoziiert werden (Kreutzer und Sirrenberg 2019, S. 3).

Dieses menschliche Denkvermögen kann auf unterschiedliche Weise durch künstliche Intelligenz nachgeahmt werden, wobei sich die methodischen Ansätze teils erheblich unterscheiden. Beim überwachten Lernen (Supervised Learning) erfolgt das Training eines KI-Modells auf Basis von durch Menschen annotierten, also gelabelten, Daten. Ziel ist es, eine Abbildung von Eingabedaten auf gewünschte Ausgaben zu erlernen, wodurch die KI gezielt auf bestimmte Problemstellungen vorbereitet wird. Im Gegensatz dazu steht das unüberwachte Lernen (Unsupervised Learning), bei dem unstrukturierten oder nicht gelabelten Daten verwendet werden. Hier versucht der Algorithmus selbstständig Muster, Strukturen oder Ähnlichkeiten innerhalb der Daten zu identifizieren, ohne dass eine explizite Zielvorgabe vorliegt. Einen gänzlich anderen Ansatz verfolgt das bestärkende Lernen (Reinforcement Learning). In diesem Ansatz lernt die Künstliche Intelligenz durch Interaktion mit einer Umgebung, indem er für bestimmte Handlungen Belohnungen oder Bestrafungen erhält. Ziel ist es, eine Strategie zu entwickeln, die langfristig die maximale Belohnung erzielt (Kreutzer und Sirrenberg 2019, 7f).

In dieser Arbeit befassen wir uns mit sogenannten Large Language Models (LLMs), welche unter die Kategorie des unüberwachten Lernens (Unsupervised Learning) fallen.

4.1.1 Natural Language Processing

Natural Language Processing (NLP) befasst sich mit der automatisierten Verarbeitung natürlicher Sprache, also jener Sprache, die Menschen im Alltag zur Kommunikation verwenden. Im Gegensatz zu formalen Sprachen, wie etwa Programmiersprachen, die klar definierte syntaktische Regeln aufweisen und eigens für die Mensch-Maschine-Interaktion entwickelt wurden (Kreutzer und Sirrenberg 2019, 28 ff.).

Die Entwicklung von NLP-Technologien lässt sich in drei wesentliche Paradigmen unterteilen. Zunächst dominierten symbolische Ansätze, bei denen sprachliche Informationen durch explizit definierte Regeln und strukturierte, schematische Netzwerke verarbeitet wurden. Diese Methoden arbeiten deterministisch und basieren auf manuell gelabeltem linguistischem Wissen. Mit dem Aufkommen leistungsfähiger Rechenressourcen gewann ab den 1990er-Jahren das statistische NLP an Bedeutung. Hierbei wird Wissen durch Netzwerke miteinander verbundener Wörter modelliert, wobei Wahrscheinlichkeiten und empirisch gewonnene Muster eine zentrale Rolle spielen (Zhao et al. 2022a, 3 f).

Einen weiteren bedeutenden Entwicklungsschub erlebte das Feld ab etwa 2012 mit dem Einsatz tiefer neuronaler Netze (Deep Learning). Deep-Learning-Methoden nutzen große Mengen an Trainingsdaten, um komplexe sprachliche Muster zu erkennen und automatisiert Klassifikationen oder Vorhersagen zu treffen. Diese Ansätze haben sich insbesondere durch ihre hohe Flexibilität und Genauigkeit gegenüber früheren Verfahren durchgesetzt und bilden heute die Basis vieler moderner NLP-Anwendungen (Zhao et al. 2022a, 3 f).

Die Einsatzbereiche von NLP sind vielfältig. Dazu zählen etwa Speech-to-Text-Systeme, wie sie in digitalen Sprachassistenten (z. B. Siri oder Smart-Home-Geräten) verwendet werden, sowie Speech-to-Speech-Übersetzungen (etwa bei Google Translate), Text-to-Speech-Anwendungen (z. B. das Vorlesen von Textdokumenten) oder Text-to-Text-Verarbeitung, etwa im Rahmen der automatisierten Textanalyse (Kreutzer und Sirrenberg 2019, 28 ff).

Trotz erheblicher Fortschritte stellen die Eigenschaften natürlicher Sprache weiterhin große Herausforderungen dar. Dialekte, Akzente, Wortschatzvielfalt, syntaktische Komplexität und semantische Mehrdeutigkeiten erschweren die automatische Interpretation ebenso wie sprachliche Stilmittel wie Ironie, Sarkasmus oder Wortspiele. Letztere setzen häufig kontextuelles und kulturelles Hintergrundwissen voraus, das für Maschinen schwer zugänglich ist (Kreutzer und Sirrenberg 2019, 28 ff.).

Die Relevanz von NLP zeigt sich auch in der Softwareentwicklung, insbesondere im Bereich der Anforderungsanalyse. In einer Umfrage unter 151 Softwareunternehmen aus den frühen 2000er-Jahren gaben 95 % der Befragten an, dass die meisten ihrer Anforderungen in natürlicher Sprache dokumentiert werden (Zhao et al. 2022a, S. 2). Dies verdeutlicht die Notwendigkeit natürliche Sprache nicht nur zu verstehen, sondern auch systematisch verarbeiten zu können, insbesondere im Kontext der Entwicklung komplexer Softwaresysteme.

4.1.2 Large Language Models

Large Language Models (LLMs) stellen eine Teilmenge des Natural Language Processing (NLP) dar und zählen innerhalb dieses Forschungsfeldes zu den leistungsfähigsten Ansätzen (Gao et al. 2023, S. 1). Seit der Veröffentlichung von ChatGPT durch OpenAI im November 2022 hat sich die Verbreitung von LLMs rasant beschleunigt. Sie finden zunehmend Anwendung in vielfältigen Anwendungsbereichen, von der automatisierten Codegenerierung über das Verfassen von Anschreiben bis hin zur Unterstützung bei der Erstellung komplexer Texte (Teubner et al. 2023, S. 1f).

Die Entwicklung von LLMs lässt sich in vier grundlegende Kategorien einteilen: statistische Modelle, Machine-Learning-Modelle, Deep-Learning-Modelle und Transformer basierte Modelle. Ihren Ursprung haben LLMs in statistischen Ansätzen wie dem N-Gram-Modell, das Wahrscheinlichkeiten auf Basis beobachteter Wortfolgen berechnet. Diese frühen Verfahren wurden jedoch im Zuge wachsender Datenverfügbarkeit und gesteigener Rechenkapazitäten zunehmend durch Deep-Learning-Methoden ersetzt. Moderne LLMs wie ChatGPT, BERT oder Claude basieren heute auf der sogenannten Transformer-Architektur, die sich durch eine besonders leistungsfähige Verarbeitung natürlicher Sprache auszeichnet (Hadi et al. 2023, S. 1ff).

Die Transformer-Architektur bildet das technologische Fundament aktueller LLMs. Sie besteht aus zwei zentralen Komponenten: dem Encoder und dem Decoder. Der Eingabetext wird zunächst durch einen oder mehrere Encoder verarbeitet. Im Gegensatz zu früheren neuronalen Netzwerkarchitekturen wie den Recurrent Neural Networks (RNNs), bei denen die Informationsverarbeitung sequenziell erfolgt, erlaubt die Transformer-Struktur eine parallele Verarbeitung. Diese Eigenschaft reduziert die Trainingszeit erheblich und ermöglicht eine effizientere Modellierung komplexer sprachlicher Zusammenhänge (Vaswani et al. 2017, S. 1ff).

4.1.3 Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) ist ein Verfahren, mit dem Large Language Models durch externe Datenquellen ergänzt werden können. Die Einbindung zusätzlicher kontextbezogener Informationen ermöglicht eine Steigerung der Genauigkeit und Relevanz der generierten Antworten. Zudem kann die gezielte Bereitstellung von Datengrundlagen das Auftreten sogenannter Halluzinationen, also faktisch unzutreffender Aussagen, deutlich reduzieren. Ein weiterer wesentlicher Vorteil besteht darin, dass Entwickler*Innen das zugrunde liegende Modell nicht vollständig neu trainieren müssen. Stattdessen genügt es, relevante Daten bereitzustellen, um die Leistung des Modells kontextabhängig zu verbessern (Cheng et al. 2024, 2f).

Das von OpenAI eingesetzte Naive RAG Framework gliedert sich in mehrere aufeinanderfolgende Prozessschritte. Zunächst erfolgt die Extraktion von Informationen aus verschiedenartigen Dateiformaten. In einem zweiten Schritt wird der extrahierte Text in kleinere Abschnitte segmentiert, da Large Language Models aufgrund ihrer Architektur nur mit einer begrenzten Menge an Eingabedaten arbeiten können. Anschließend, im dritten Schritt, werden diese Textabschnitte mithilfe eines Sprachmodells in Vektoren überführt. Im vierten Schritt wird

auch die Nutzereingabe vektorisiert. Daraufhin erfolgt eine semantische Ähnlichkeitssuche zwischen dem Vektor der Anfrage und den Vektoren der gespeicherten Inhalte. Die inhaltlich ähnlichsten Textsegmente werden ausgewählt und für die nachfolgende Generierung verwendet. In der Generierungsphase entscheidet das Sprachmodell selbstständig, ob und in welchem Umfang die bereitgestellten Informationen in die Antwort einfließen. Falls zusätzlich historische Chatverläufe vorliegen, können diese bei der Answererstellung mit den aus dem Retrieval Augmented Generation Prozess gewonnenen Inhalten kombiniert werden (Gao et al. 2023, 6 f).

Retrieval Augmented Generation eignet sich primär für die Beantwortung spezifischer Anfragen und das Auffinden gezielter Informationen. Für das Erlernen neuer Sprachmuster, Formate oder Stile sowie für das tiefgehende Verständnis breit gefächelter Domänen ist dieser Ansatz hingegen nicht geeignet (Cheng et al. 2024, S. 3).

4.1.4 Prompt Optimierung

Die Qualität der von einem Sprachmodell generierten Ausgabe hängt in hohem Maße vom erhaltenen Prompt ab. Unter einem Prompt wird in diesem Zusammenhang beispielsweise ein Satz, eine Frage oder eine Anweisung verstanden, die dem Modell in natürlicher Sprache gegeben wird. Im Bereich des Prompt Engineering wird versucht Anweisungen so zu optimieren, dass der Output eine höhere Qualität aufweist (Liu et al. 2023, S. 2).

Insbesondere in der Phase der Anforderungserstellung kann die Qualität der generativen KI-Ausgaben durch die Einbettung von Kontext verbessert werden. Wird im Prompt entsprechender Kontext integriert, können umfassende Informationen über das System sowie die relevanten Stakeholder berücksichtigt werden (Arvidsson und Axell 2023, 10 ff).

Im Bereich der Anforderungsanalyse haben sich sogenannte Templates und Reasoning-Ansätze als nützlich erwiesen. Templates bezeichnen strukturierte Formulierungen, die einem Prompt vorangestellt werden können, um die Ausgabe zu lenken. Zum Beispiel verfolgt der Reasoning-Ansatz das Ziel, durch schrittweises Denken die Qualität der Ergebnisse zu steigern. Ein typisches Beispiel lautet: „Löse das Problem Schritt für Schritt und achte darauf, dass die Lösung korrekt ist“ (Arvidsson und Axell 2023, 2023, 10 ff).

Auch Prompts aus dem Teilbereich der Analyse lassen sich sinnvoll in der Anforderungsanalyse einsetzen. Diese enthalten häufig Elemente wie: Prompt + Lösung + Prämisse + Hypothese + Bewertung („richtige oder falsche Annahme“) (Arvidsson und Axell 2023, 10ff).

Für die Anforderungsspezifikation ist insbesondere der Bereich der Eindeutigkeit relevant. Ziel ist es, potenziell mehrdeutige oder unklare Anforderungen frühzeitig zu identifizieren. Hierbei kommen Formulierungen wie „Zeige alle Bereiche, die mehrdeutig sein könnten, um unerwünschte Ergebnisse zu vermeiden“ zum Einsatz (Arvidsson und Axell 2023, 10 ff).

Im Rahmen der Anforderungvalidierung haben sich die Verwendung von Personas als hilfreich erwiesen. Diese ermöglichen eine zielgruppenspezifische Überprüfung der Anforderungen. Ein Beispiel für einen entsprechenden Prompt wäre: „Ich möchte, dass du dich wie {Person/Rolle} verhältst (Arvidsson und Axell 2023, 10 ff).“

4.1.5 Fine Tuning

Das Training großer Sprachmodelle (LLMs) erfolgt in vier Phasen: General Pretraining, Domain Adaptive Pretraining (DAPT), Task Specific Pretraining und Fine Tuning. Die ersten drei Phasen basieren auf nicht gelabelten Daten und führen das Modell von allgemeinem Sprachverständnis zu aufgabenspezifischem Wissen. Fine Tuning bildet die letzte Phase und nutzt als einzige gelabelte Daten, die in der Regel durch menschliche Annotation entstehen. Dabei wird das zuvor generalisierte Modell gezielt auf eine konkrete Aufgabe angepasst, wobei menschliche Expertise maßgeblich zur Leistungsverbesserung beiträgt. Während der rechnerische Aufwand mit jeder Stufe abnimmt, steigt der menschliche Aufwand, was jedoch durch deutlich verbesserte Metriken wie Precision und F1 Score gerechtfertigt ist (Norheim et al. 2024, 3ff).

Fine Tuning wurde in der bestehenden Forschung bereits erfolgreich eingesetzt, um unterschiedliche Prozesse innerhalb des Anforderungsmanagements zu unterstützen. So konnte es bei der Generierung von Anforderungen genutzt werden, um auf Basis von Blogeinträgen neue Anforderungen automatisch zu erzeugen. In der Analyse wurde ein Sprachmodell darauf trainiert, Anforderungen in acht inhaltliche Kategorien zu klassifizieren. Im Rahmen der Transformation kam Fine Tuning zum Einsatz, um in natürlicher Sprache formulierte Anforderungen in formale Sprachmodelle wie Gherkin zu übersetzen (Norheim et al. 2024, 5ff). Diese Anwendungen verdeutlichen das Potenzial von Fine Tuning zur gezielten Anpassung von Sprachmodellen auf komplexe und wissensintensive Aufgaben.

Die Qualität der Ergebnisse eines Modells ist in hohem Maße von der Qualität der zugrunde liegenden Trainingsdaten abhängig. Werden einem Basismodell fehlerhafte, unvollständige oder anderweitig ungeeignete Daten bereitgestellt, kann dies nicht nur die Leistungsfähigkeit einschränken, sondern im schlimmsten Fall zu einer Verschlechterung der Modellperformance führen. Zwar ermöglicht Fine Tuning eine Anpassung des Modells an spezifische Anwendungsbereiche, jedoch konnte bislang keine Modellkonfiguration eine inhaltlich präzisere und konsistenter definierte Erzählstruktur erzeugen als ein fachkundiger Domänenexperte (Liu et al., 1ff).

4.2 Künstliche Intelligenz und Anforderung

Das Requirements Engineering (RE) umfasst sämtliche Aktivitäten, die notwendig sind, um Anforderungen an ein Softwaresystem zu erheben, zu analysieren, zu spezifizieren, zu validieren und im weiteren Projektverlauf zu managen. Die Qualität dieser Anforderungen wirkt sich direkt auf den Erfolg des gesamten Entwicklungsprozesses aus. Mit dem Aufkommen leistungsfähiger Methoden der KI eröffnen sich neue Möglichkeiten, diese Phasen effizienter, präziser und teilweise automatisiert zu gestalten. KI-gestützte Verfahren können nicht nur große Mengen unstrukturierter Informationen verarbeiten, sondern auch Muster erkennen, Anforderungen klassifizieren, Inkonsistenzen identifizieren und Priorisierungsempfehlungen ableiten.

In den folgenden Unterkapiteln werden fünf zentrale Phasen des RE, von der Anforderungserhebung über Analyse und Modellierung, Anforderungsspezifikation und

Validierung bis hin zum Anforderungsmanagement, betrachtet. Für jede Phase werden relevante Forschungsansätze vorgestellt, die den Einsatz von KI zur Unterstützung oder Automatisierung spezifischer Aufgaben untersuchen. Dabei wird deutlich, dass KI nicht nur punktuell eingesetzt werden kann, sondern das Potenzial besitzt, den gesamten Requirements-Engineering-Prozess nachhaltig zu verändern.

4.2.1 Anforderungserhebung

Die Anforderungserhebung stellt einen zentralen Prozessschritt bei der Entwicklung eines Produkts dar. Ziel dieses Schritts ist es, die Anforderungen der Endnutzer möglichst vollständig und präzise zu erfassen. Hierzu kommen klassische Erhebungstechniken wie Interviews, Besprechungen, Workshops und Brainstorming zum Einsatz. Der Prozess lässt sich in vier aufeinanderfolgende Teilschritte gliedern: Vorbereitung, Sammlung, Prüfung und Verhandlung. Angesichts der zunehmenden Geschwindigkeit in der Produktentwicklung und der wachsenden Ansprüche der Endkundinnen und Endkunden stoßen traditionelle Methoden jedoch zunehmend an ihre Grenzen (Cheligeer et al. 2022, 1ff).

Im Rahmen der Vorbereitung werden in der aktuellen Forschung verschiedene technologische Ansätze erprobt, um den Erhebungsprozess zu optimieren. So beschreibt eine Studie ein SVM-basiertes System, das auf Basis der Wissensstruktur im Unternehmen relevante Informationen aus wissenschaftlichen Publikationen extrahiert. Eine weitere Untersuchung setzt einen überwachten Machine-Learning-Algorithmus ein, um Nutzerbewertungen in vier Kategorien zu klassifizieren: Nutzungserfahrung, Fehlerberichte (Bugs), Funktionswünsche sowie allgemeine Bewertungen (Cheligeer et al. 2022, 7f).

Der Teilschritt Sammlung befasst sich primär mit der Identifikation und Klassifikation von Anforderungen. Ein Forschungsschwerpunkt liegt auf der automatisierten Erkennung, ob ein gegebener Textausschnitt überhaupt eine Anforderung darstellt. Darüber hinaus werden Verfahren entwickelt, um Anforderungen in funktionale und nicht-funktionale Kategorien zu unterteilen. Künstliche Intelligenz wird zudem eingesetzt, um sicherheitsrelevante Anforderungen zu identifizieren und zu clustern (Cheligeer et al. 2022, 9).

Die beiden abschließenden Schritte, Prüfung und Verhandlung, können in allen Phasen des Requirements-Engineering-Prozesses stattfinden. Die Prüfung umfasst Maßnahmen wie die Sicherstellung eines konsistenten Formats und einer einheitlichen Strukturierung sowie die Bewertung der inhaltlichen Qualität von Anforderungen. Die Verhandlung zielt auf die Optimierung der Entscheidungsgrundlage ab. So zeigt ein Forschungsprojekt, wie Kundenanfragen in technische Änderungen und funktionale Änderungen kategorisiert werden, um Projektleitungen eine fundiertere Verhandlungsposition zu ermöglichen. Ergänzend wurde ein überwachter Machine-Learning-Algorithmus entwickelt, der Kundenanfragen automatisiert der zuständigen Abteilung zuweist (Cheligeer et al. 2022, 9).

4.2.2 Analyse und Modellierung

Die automatische Generierung schematischer Modelle wie Feature-Modelle, Domänenmodelle oder Use-Case-Modelle unter Einsatz von Methoden der natürlichen Sprachverarbeitung (Natural Language Processing, NLP) stellt einen vielversprechenden Ansatz zur Unterstützung des Softwareentwicklungsprozesses dar. Durch die Extraktion relevanter Informationen aus Software-Anforderungsdokumenten und deren Transformation in strukturierte Modellrepräsentationen kann eine konsistentere und zielgerichtetere Umsetzung der geplanten Software erreicht werden. Dieser Ansatz reduziert potenzielle Missverständnisse in frühen Phasen des Entwicklungsprozesses und trägt zur Steigerung der Qualität der resultierenden Softwaresysteme bei (Da Cruz und Cruz 2025, 2ff).

Eine systematische Übersicht über den Forschungsstand zu diesem Themenfeld liefert die Arbeit „The Use of NLP-Based Text Representation Techniques to Support Requirement Engineering Tasks: A Systematic Mapping Review“. Darin wird festgestellt, dass ein Großteil der aktuellen Forschungsschwerpunkte auf der Klassifikation von Anforderungen in funktionale und nicht-funktionale Anforderungen liegt (R. Sonbol et al. 2022, S. 62815). Zur Umsetzung dieser Klassifikationsaufgabe werden unterschiedliche Vektorisierung Methoden wie Bag-of-Words (BoW), Term Frequency, Inverse Document Frequency (TF-IDF) und Chi-Quadrat-Merkmalss Selektion eingesetzt. Diese werden in Kombination mit Klassifikationsalgorithmen wie K-Nearest Neighbor, Multinomial Naive Bayes und logistischer Regression verwendet (Dias Canedo und Cordeiro Mendes 2020, S. 2).

Aufgaben des Anforderungsmanagements, wie z.B. die Sicherstellung der Nachvollziehbarkeit (Traceability), haben in der bisherigen Forschung einen niedrigeren Stellenwert (Cheng et al. 2025, S. 13). Traceability bezieht sich auf die Identifikation zusammenhängender Anforderungen innerhalb eines Anforderungsdokuments oder über verschiedene Dokumente hinweg. In einem Forschungsprojekt wurde diese Aufgabe durch Vektorisierung der Anforderungen und die Nutzung von FrameNet, einem computerlinguistischen Lexikon auf Basis der „Semantic Frame Theory“, umgesetzt. Die so erzeugten Vektoren wurden anschließend mittels Cosine Similarity verglichen, um semantisch verwandte Anforderungen zu identifizieren (Alhoshan et al. 2019, 1ff).

Ein konkretes Anwendungsbeispiel bietet die Arbeit „An Efficient Methodology for the Categorization of Software Requirements Using Natural Language Processing and Similarity Analysis“. Hier wurden TF-IDF und Cosine Similarity zur Gruppierung (Clustering) von Anforderungen eingesetzt. TF-IDF bewertet dabei die Relevanz eines Wortes in Relation zu einem Dokumentenkörper, während die Cosine Similarity misst, wie ähnlich sich Vektoren sind, in der NLP-Praxis typischerweise, wie ähnlich sich Wörter oder Dokumente zueinander verhalten (R. Izhar et al. 2025, 83596ff).

4.2.3 Anforderungsspezifikation

Ein erheblicher Teil der aktuellen Forschung zum Einsatz Künstlicher Intelligenz (KI) im Requirements Engineering (RE) konzentriert sich auf die Phase der Anforderungsspezifikation. Die systematische Literaturübersicht von Cheng et al. (2024) zeigt, dass 14 von 27 analysierten

Primärstudien (also 51,9%) explizit die Unterstützung oder Automatisierung von Aktivitäten der Anforderungsspezifikation thematisieren (Cheng et al. 2024, S. 25). Dies unterstreicht die zentrale Rolle, die der KI in dieser frühen Phase des Softwareentwicklungsprozesses zugeschrieben wird.

Ein wesentlicher Fokus innerhalb dieses Forschungsfeldes liegt auf der Nutzung statischer Datenquellen zur Anforderungsermittlung. Dabei handelt es sich typischerweise um domänenspezifisches Wissen in Form von Dokumentationen, Handbüchern oder unternehmensspezifischen Artefakten wie Pflichten- und Lastenheften. Ergänzend dazu existieren Forschungsarbeiten, die dynamischere, nutzergenerierte Quellen wie App-Bewertungen, Blogs oder Mailinglisten in die Anforderungsgenerierung (Henriksson und Zdravkovic, S. 4). Diese unterschiedlichen Datenquellen eröffnen neue Potenziale zur Ableitung relevanter Anforderungen aus vielfältigen Kontexten.

Zur Extraktion von Anforderungen aus bestehenden Textdokumenten wurden verschiedene Ansätze entwickelt. So ermöglichen KI-gestützte Verfahren beispielsweise die automatische Generierung von UML-Diagrammen aus Domänendokumenten. Ein Beispiel hierfür ist auch die Transformation von natürlichsprachlich formulierten User Storys in visuelle Use-Case-Diagramme (Elallaoui et al. 2018, 1ff).

Neben der Anforderungsermittlung rückt zunehmend auch die Unterstützung bei der Ausarbeitung und Verfeinerung von Anforderungen in den Fokus. Mehrere in Studien befragte Expert*Innen betonen, dass der Einsatz von KI als hilfreich empfunden wird, insbesondere im Hinblick auf die Vollständigkeit und inhaltliche Plausibilität geplanter Anforderungen. Unterstützend wirken dabei generative Rückfragen wie: „Welche Aspekte muss ich bei der Umsetzung dieser Anforderung berücksichtigen?“ oder „Ist diese Art der Umsetzung zielführend?“ (Arvidsson und Axell 2023, 11ff).

Ein weiteres vielversprechendes Anwendungsfeld ist die teilautomatisierte Erstellung von Software Requirements Specifications (SRS). So konnte beispielsweise durch den Einsatz von ChatGPT der Erstellungsaufwand um das 24,6-Fache reduziert werden, bei Einsatz des Modells CodeLlama-13B wurde sogar eine Reduktion um das 29,5-Fache erreicht. Die erzeugten Spezifikationen zeigten vergleichbare oder bessere Ergebnisse hinsichtlich Verständlichkeit und Eindeutigkeit. Gleichzeitig bestehen jedoch weiterhin erhebliche Herausforderungen in Bezug auf Richtigkeit und Nachvollziehbarkeit der Inhalte (M. Krishna et al. 2024, 6ff).

4.2.4 Validierung

Die Validierung von Softwareanforderungen verfolgt das Ziel, früh zu prüfen, ob die Anforderungen korrekt, vollständig und verständlich sind. Dafür stehen verschiedene Methoden zur Verfügung. Formale Verfahren wie Z Notation, Alloy und SDL verwenden mathematische Modelle, um Anforderungen besonders präzise zu beschreiben. Ergänzend dazu bieten modellbasierte und grafische Darstellungen Unterstützung, zum Beispiel Use Case Diagramme zur Analyse der gewünschten Funktionen oder Entity Relationship Diagramme zur Strukturierung der Daten. Eine weitere Möglichkeit ist Behavior Driven Development, bei dem Anforderungen

direkt in ausführbare Testszenarien überführt werden. Zusätzlich kommen Prototypen, Reviews, Inspektionen, Traceability Analysen und automatisierte Tests zum Einsatz. Zusammen tragen diese Ansätze dazu bei, Anforderungen systematisch zu überprüfen und ihre Qualität sicherzustellen (Ahmed et al., 2f).

Aktuelle Large Language Models (LLMs) wie beispielsweise ChatGPT bieten neue Möglichkeiten zur automatisierten Analyse natürlichsprachlicher Anforderungen. Insbesondere eignen sie sich dazu, Inkonsistenzen in der Anforderungsspezifikation zu identifizieren, ohne dass eine formale Modellierung vorausgesetzt wird (Cheng et al. 2024, 6f).

In einer Fallstudie wurde der Ansatz verfolgt, ein LLM mit einer Vielzahl an Inputformaten zu trainieren. Dazu zählten natürlichsprachliche Anforderungen, verschiedene Diagrammtypen (darunter UML, Entity-Relationship-Diagramme, Datenflussdiagramme und Use-Case-Diagramme), annotierte Inkonsistenzen sowie Informationen zur inhaltlichen Domänenabdeckung. Das trainierte Modell erreichte anschließend eine Genauigkeit von 94 Prozent bei der Erkennung von Inkonsistenzen zwischen textuellen Anforderungen und den zugehörigen Diagrammen. Besonders hervorzuheben ist die Effizienzsteigerung, die durch den Einsatz des LLMs erzielt wurde. Der zeitliche Aufwand für die manuelle Analyse konnte von etwa zwei Stunden auf lediglich 7,2 Sekunden reduziert werden. Dies entspricht einer Beschleunigung um den Faktor 100 (Ahmed et al., 76ff).

4.2.5 Anforderungsmanagement

Bei der Priorisierung und Klassifizierung von Anforderungen im Rahmen des Anforderungsmanagements kommen verschiedene Verfahren des maschinellen Lernens (ML) zum Einsatz. Insbesondere Algorithmen wie K-Nearest Neighbors (KNN) und Random Forest werden genutzt, um Entscheidungen im Release Planning datenbasiert zu unterstützen. Diese Modelle ermöglichen es, Anforderungen hinsichtlich ihrer Relevanz und Umsetzungspriorität zu bewerten, indem sie historische Daten und Ähnlichkeitsmaße heranziehen (Da Cruz und Cruz 2025, S. 37).

Darüber hinaus finden zunehmend Methoden Anwendung, die Techniken der natürlichen Sprachverarbeitung (Natural Language Processing, NLP) mit ML-Algorithmen kombinieren. Diese Ansätze analysieren die semantischen Inhalte von Anforderungsbeschreibungen, um daraus Priorisierungsempfehlungen abzuleiten. Durch die automatisierte Verarbeitung sprachlicher Daten können so auch große Mengen an Anforderungen effizient klassifiziert und bewertet werden (Da Cruz und Cruz 2025, S. 38).

Neben überwachten klassifikationsbasierten Verfahren kommen auch unüberwachte Lernmethoden wie das k-Means-Clustering zum Einsatz. Diese dienen beispielsweise der Segmentierung von Stakeholdern anhand marktbezogener Merkmale. Durch die Gruppierung ähnlicher Interessen- und Bedürfnisprofile lassen sich Anforderungen zielgerichteter priorisieren und Entscheidungen an den jeweiligen Stakeholdergruppen ausrichten (Da Cruz und Cruz 2025, S. 38).

Die Kombination dieser datengetriebenen Ansätze ermöglicht eine objektivere und fundiertere Entscheidungsfindung. Stakeholder und Projektverantwortliche profitieren von der strukturierten Analyse, da sie auf dieser Basis Anforderungen zielgerichteter bewerten und priorisieren können. Dadurch steigt die Wahrscheinlichkeit, dass Projekt- und Produktzeile erreicht werden (Da Cruz und Cruz 2025, S. 44).

4.3 Herausforderungen bei der Nutzung Künstlicher Intelligenz

Die Nutzung von Künstlicher Intelligenz zur Generierung von Anforderungen zeigt ein hohes Potenzial und eröffnet vielversprechende Perspektiven für die Anforderungsanalyse. Dennoch bestehen in einigen Bereichen weiterhin Entwicklungsbedarfe, da die derzeitigen Systeme noch bestimmte Limitationen aufweisen (Cheng et al. 2024, S. 12–13).

Dieses Kapitel behandelt zentrale Herausforderungen im Umgang mit Large Language Models, darunter Bias und Fairness, ethische Fragestellungen und Compliance, Sicherheitsaspekte sowie Datenschutz. Darüber hinaus werden Probleme wie Halluzinationen von Modellen sowie Fragen des Urheberrechts und Copyrights thematisiert.

4.3.1 Bias und Fairness

Large Language Models (LLMs) und andere Systeme Künstlicher Intelligenz (KI) können, ähnlich wie der Mensch, Verzerrungen (Bias) aufweisen. Diese Verzerrungen äußern sich häufig in Entscheidungen, die weder fair noch nachvollziehbar sind. Besonders problematisch wird dies, wenn KI-Systeme in sensiblen gesellschaftlichen Bereichen eingesetzt werden, etwa bei der Einschätzung individueller Risiken im Strafjustizsystem. Ein vielbeachtetes Beispiel ist, das in den USA eingesetzte System Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), das die Rückfallwahrscheinlichkeit von Straftäterinnen und Straftätern bewerten soll. Studien zeigen, dass COMPAS systematische Benachteiligungen aufweist, insbesondere gegenüber People of Color, und somit erhebliche ethische wie auch rechtliche Fragestellungen aufwirft (Mehrabi et al. 2021, S. 4).

Um die Ursachen solcher Verzerrungen zu verstehen, ist eine differenzierte Betrachtung notwendig. Grundsätzlich lassen sich zwei zentrale Quellen von Bias in KI-Systemen unterscheiden: einerseits Verzerrungen, die aus voreingenommenen Trainingsdaten resultieren, und andererseits solche, die durch algorithmische Entscheidungen während der Systementwicklung entstehen. Da KI-Modelle auf der Auswertung großer, meist von Menschen erzeugter Datenmengen basieren, spiegeln sie oftmals bestehende gesellschaftliche Vorurteile wider. Diese werden durch den Trainingsprozess in die Modelle übernommen und somit reproduziert. Gleichzeitig können auch Designentscheidungen auf algorithmischer Ebene, etwa die Auswahl von Zielgrößen, Verlustfunktionen oder Modellarchitekturen, zu einer strukturellen Benachteiligung bestimmter Gruppen führen, selbst wenn die zugrunde liegenden Daten neutral erscheinen (Mehrabi et al. 2021, S. 4).

Zur weiteren Systematisierung lassen sich Verzerrungen in KI grob in drei Kategorien einteilen: (1) von den Daten zum Algorithmus, (2) vom Algorithmus zur nutzendem Person und (3) von Nutzenden zu den Daten.

Ein Beispiel aus der Kategorie Daten zum Algorithmus ist der sogenannte Omitted Variable Bias (deutsch: Ausgelassene Variablenverzerrung). Dieser tritt auf, wenn entscheidungsrelevante Informationen in den Trainingsdaten fehlen oder noch nicht vorliegen, was zu fehlerhaften oder halluzinierten Ergebnissen führen kann. So kann ein System etwa eine erhöhte Kündigungsrate von Kund*Innen beobachten und daraus unzutreffend auf eine schlechte Produktqualität schließen, obwohl der eigentliche Grund neue Konkurrenz auf dem Markt ist und dieser Konkurrent noch nicht in den Daten vorliegt (Mehrabi et al. 2021, 4ff).

Im Zusammenhang mit dem Algorithmus im Nutzerkontext lassen sich zwei zentrale Arten von Bias unterscheiden. Zum einen können Verzerrungen durch das Design und die Implementierungsentscheidungen des Algorithmus selbst entstehen. Zum anderen kann auch das Verhalten der Nutzerinnen und Nutzer, insbesondere die Art und Weise ihrer Interaktion und Kommunikation mit dem System, Bias hervorrufen oder verstärken (Mehrabi et al. 2021, 7ff).

Ein weiterer relevanter Verzerrungstyp ist unter der Kategorie Nutzer*Innen zu Daten zu verorten. Hierbei wirkt sich das Verhalten der Nutzer*Innen direkt auf die Datengrundlage aus. Ein bekanntes Phänomen ist der Primacy Effect, bei dem Menschen Informationen, die zuerst präsentiert werden, stärker gewichten als nachfolgende. Dies hat praktische Auswirkungen, etwa bei Suchmaschinen: Webseiten mit hoher Nutzerinteraktion werden algorithmisch bevorzugt und steigen im Ranking, selbst wenn sie qualitativ schlechtere oder weniger korrekte Informationen enthalten. Dadurch entsteht ein selbstverstärkender Kreislauf, in dem Popularität fälschlicherweise mit Relevanz oder Richtigkeit gleichgesetzt wird (Mehrabi et al. 2021, 4ff).

Zusammenfassend lässt sich festhalten, dass Verzerrungen in KI-Systemen tiefgreifende Auswirkungen auf gesellschaftliche Prozesse und den Alltag der meisten Menschen haben können. Sie betreffen nicht nur Anwendungen wie Gesichtserkennung oder Chatbots, sondern finden auch zunehmend Eingang in sicherheitsrelevante und staatliche Entscheidungssysteme, etwa in der Justiz oder Verwaltung. Vor diesem Hintergrund ist es unerlässlich, Fragen der Fairness, Transparenz und Rechenschaftspflicht bei der Entwicklung und Implementierung Künstlicher Intelligenz systematisch zu berücksichtigen.

4.3.2 Ethik und Compliance

Im Zuge der raschen Verbreitung von Systemen der Künstlichen Intelligenz (KI) haben staatliche Institutionen, Nichtregierungsorganisationen (NGOs) und privatwirtschaftliche Unternehmen in den vergangenen Jahren vermehrt ethische Leitlinien formuliert. Ein prominentes Beispiel bildet der EU-Verordnungsentwurf zum „AI-Act“, der einen verbindlichen Rechtsrahmen für vertrauenswürdige KI schaffen soll.

Um die Zielsetzungen solcher Regelwerke einordnen zu können, ist zunächst ein begriffliches Fundament erforderlich. Die Ethik, ein Teilgebiet der Philosophie, untersucht systematisch das menschliche Handeln, indem sie Konzepte wie „gut“ und „böse“ sowie die Begründungen

moralischer Entscheidungen reflektiert. Sie liefert somit den theoretischen Rahmen, auf dem Vorgaben für gesellschaftliche und technologische Kontexte basieren (Ritter et al., 1971, S. 1102).

Auf dieser Grundlage entstanden etwa die AI4People-Leitlinien, die ihrerseits den AI-Act maßgeblich beeinflusst haben. Sie beruhen auf Schlüsselprinzipien wie Autonomie, Fürsorge und Gerechtigkeit; ein besonderer Stellenwert kommt der Transparenz einschließlich der Erklärbarkeit zu. Ziel ist es, den verantwortungsvollen Einsatz von KI in sensiblen Bereichen wie Gesundheitswesen oder Personalwesen sicherzustellen (Akbarighatar, 2025, 7ff).

Die gegenwärtige Fachdiskussion konzentriert sich insbesondere auf fünf Problemfelder: (1) Bias und Fairness, (2) Privatsphäre und Datensicherheit, (3) Falschinformationen, (4) Transparenz und Rechenschaftspflicht sowie (5) Urheberrechtsfragen und Plagiate. Einige dieser Themen werden in den nachfolgenden Kapiteln vertieft, die übrigen werden im vorliegenden Abschnitt behandelt (Jiao et al. 2024, 2024, S. 8).

Unter Fairness bei Large-Language-Models (LLMs) versteht man die Abwesenheit systematischer Leistungsunterschiede zulasten bestimmter Gruppen, etwa bei Diagnosen im Gesundheitswesen in Abhängigkeit von Hautfarbe, Lebensstil oder Geschlecht. Die aktuelle Forschung zielt vor allem darauf ab, solche Abweichungen zu messen und durch technische wie organisatorische Maßnahmen zu mindern (Jiao et al., 2024, 7ff).

Ein zweites zentrales Risiko liegt in der möglichen Verstärkung von Falschinformationen: Da LLM-generierte Texte meist kohärent und plausibel wirken, können sie gezielt für massenhafte Desinformation in sozialen Medien missbraucht werden. Parallel dazu entstehen jedoch auch Verfahren, um derartige Inhalte automatisiert zu erkennen und einzuordnen (Jiao et al., 2024, 15f).

Durch ethische Betrachtungen lässt sich ein Rahmen entwickeln, der sowohl die Chancen als auch die Risiken von KI-Systemen adressiert und damit zu ihrer verantwortungsvollen Gestaltung beiträgt.

4.3.3 Sicherheit und Datensicherheit

Ein wesentlicher Schwerpunkt aktueller wissenschaftlicher Auseinandersetzungen mit dem Thema Datensicherheit liegt im Schutz sensibler Informationen im Zusammenhang mit modernen KI-Systemen, insbesondere großen Sprachmodellen. Diese Modelle zeichnen sich durch eine außergewöhnliche Fähigkeit zur Informationsverarbeitung und -wiedergabe aus. Ihre hohe Reproduktionsfähigkeit birgt jedoch das Risiko, dass vertrauliche Daten ungewollt wiedergegeben oder erschlossen werden. Dabei handelt es sich unter anderem um personenbezogene Angaben wie Namen, Adressen, Telefonnummern sowie um finanzielle, medizinische oder andere besonders schützenswerte Informationen.

Die besondere Relevanz dieser Problematik zeigt sich in der zunehmenden Häufigkeit sogenannter Extraction Attacks. Dabei handelt es sich um Angriffe, bei denen systematische Abfragen dazu genutzt werden, um Informationen, die das Modell während des Trainings

verarbeitet hat, teilweise zu rekonstruieren. Wie aktuelle Forschungsergebnisse zeigen, sind LLMs trotz implementierter Schutzmechanismen nicht immun gegenüber solchen Angriffen (Jiao et al. 2024, 14f).

Am Beispiel von ChatGPT lässt sich die praktische Relevanz dieses Themas verdeutlichen. Eingaben von Nutzerinnen und Nutzern werden bis zu 30 Tage lang gespeichert und können zur Weiterentwicklung des Modells verwendet werden. Zwar betont OpenAI, dass branchenübliche Sicherheitsstandards wie die Verschlüsselung von Daten bei Übertragung und Speicherung zum Einsatz kommen, doch verbleibt ein Restrisiko. Eine Weitergabe an Dritte erfolgt laut Unternehmensangaben nur mit ausdrücklicher Einwilligung oder in gesetzlich definierten Ausnahmefällen. Darüber hinaus versichert OpenAI, dass auch externe Partnerunternehmen vergleichbare Datenschutzstandards einhalten (Sebastian 2023, 2ff).

Hinzu kommen gezielte Angriffe auf das Modellverhalten. Adversariale Eingaben können ein Modell dazu bringen, sicherheitskritische oder unerwünschte Ausgaben zu erzeugen. Ebenso besteht das Risiko von Modellextraktionsangriffen, bei denen durch systematische Abfragen versucht wird, das Modellverhalten zu replizieren. Darüber hinaus können Datenvergiftungsangriffe das Trainingsmaterial gezielt manipulieren, was langfristige Auswirkungen auf die Ausgaben des Modells haben kann (Sebastian, 2023, 5f).

Diese Bedrohungslage verdeutlicht die Notwendigkeit umfassender technischer und rechtlicher Schutzmaßnahmen. In der Europäischen Union gelten mit der Datenschutz-Grundverordnung (DSGVO), der ePrivacy-Richtlinie sowie der geplanten KI-Verordnung bereits verbindliche Regelwerke, die den Umgang mit personenbezogenen Daten regeln. Bezugsquellen wie OpenAI sind verpflichtet, diese Standards einzuhalten und Maßnahmen zur Sicherstellung der Datensicherheit umzusetzen (Lami et al. 2024, 3ff).

Besonders in cloudbasierten Systemen, bei denen Nutzereingaben zentral gespeichert und zur Weiterentwicklung der Modelle genutzt werden, ist die Sensibilität gegenüber Datenschutzfragen hoch. Um Vertrauen in KI-Systeme langfristig zu gewährleisten und ihre gesellschaftliche Akzeptanz zu fördern, ist die Integration robuster Sicherheits- und Datenschutzmaßnahmen unerlässlich.

4.3.4 Halluzinationen

Halluzinationen im Kontext der künstlichen Intelligenz bezeichnen die Generierung von Inhalten durch Sprachmodelle, die entweder inhaltlich falsch oder semantisch unplausibel sind. In der wissenschaftlichen Auseinandersetzung werden dabei zwei grundlegende Formen unterschieden: Intrinsische Halluzinationen treten auf, wenn der generierte Text den Aussagen der zugrunde liegenden Quelle widerspricht. Bei extrinsischen Halluzinationen hingegen handelt es sich um Inhalte, die anhand der verwendeten Quellen weder bestätigt noch widerlegt werden können. Besonders problematisch ist dabei, dass solche Inhalte für den Nutzer glaubhaft erscheinen, obwohl ihre faktische Grundlage fehlt. Dies kann zu erheblichen Fehlinterpretationen führen, insbesondere in wissensintensiven Anwendungsbereichen (Huang et al., 2025, 2 ff).

Die Ursachen für Halluzinationen lassen sich auf drei zentrale Faktoren zurückführen: Daten, Trainingsverfahren und Inferenzstrategien. Auf der Ebene der Daten zeigt sich, dass LLMs (Large Language Models) so konstruiert sind, dass sie in der Lage sind, große Mengen an Informationen zu speichern. Mit zunehmender Modellgröße verbessert sich zwar die Fähigkeit, komplexes Wissen zu erfassen, zugleich steigt jedoch auch die Gefahr, dass falsche oder verzerrte Informationen aus den Trainingsdaten reproduziert werden. Besonders kritisch ist dies in Fällen, in denen Modelle mit Anfragen konfrontiert werden, die außerhalb der gelernten Wissensdomäne liegen. Da LLMs nicht in der Lage sind, sämtliche Trainingsdaten exakt zu memorieren, insbesondere seltene Inhalte, neigen sie in solchen Fällen verstärkt zur Halluzination (Huang et al. 2025, 7ff).

Auch das Training selbst kann Halluzinationen begünstigen. Schon im Pre-Training arbeitet das Modell meist mit kausaler Sprachmodellierung, bei der es nur den vorherigen Textteil sieht. Dadurch ist der verfügbare Kontext eingeschränkt, und ein sogenannter Exposure Bias kann entstehen: Fehler pflanzen sich im weiteren Verlauf fort. Im Supervised Fine Tuning (SFT) lernt das Modell, auf konkrete Anweisungen zu reagieren. Bekommt es jedoch Fragen, zu denen ihm verlässliches Wissen fehlt, liefert es trotzdem Antworten, weil es keine Unsicherheit ausdrücken kann. Für Nutzende wirkt die Information dadurch fälschlich gesichert. Beim Reinforcement Learning from Human Feedback (RLHF) kommt ein weiteres Risiko hinzu. Um positive Bewertungen zu erhalten, passt sich das Modell den erwarteten Nutzerpräferenzen an und wird dabei zum „Ja-Sager“. Es erklärt etwas für wahr, nur weil es gut ankommt, selbst wenn es objektiv falsch ist. So entstehen falsche, aber sehr überzeugende Ausgaben (Huang et al. 2025, 10 f).

Auch die Dekodierungsstrategie, also die Art, wie das Modell beim Schreiben das nächste Wort wählt, beeinflusst die Zuverlässigkeit. Zufallsbasierte Verfahren wie stochastisches Sampling machen den Text zwar abwechslungsreicher, erhöhen aber auch die Chance, dass seltene und fehlerhafte Wörter erscheinen. Je höher man dabei die sogenannte Temperatur einstellt, desto stärker fällt dieser Effekt aus. Ein weiteres Problem ist Selbstüberschätzung: Das Modell verlässt sich zu sehr auf das, was es gerade selbst geschrieben hat, und kann so allmählich den ursprünglichen Kontext verlieren. Hinzu kommt das Softmax-Bottleneck, eine mathematische Begrenzung, die verhindert, dass das Modell Wahrscheinlichkeiten sehr fein abbildet. Dadurch sinkt die Genauigkeit bei der Wortwahl. Schließlich scheitern große Sprachmodelle bisweilen an bestimmten logischen Aufgaben, obwohl das nötige Wissen vorhanden ist. Ein Beispiel ist der Reversal Curse: Wird eine Frage in umgekehrter Form gestellt, antwortet das Modell häufig falsch (Huang et al. 2025, 11f).

Trotz erheblicher Fortschritte in der Entwicklung und Feinjustierung großer Sprachmodelle bleibt das Phänomen der Halluzination eine zentrale Herausforderung. Fehlerhafte oder erfundene Inhalte können erhebliche Konsequenzen nach sich ziehen, insbesondere wenn LLMs in sensiblen oder sicherheitskritischen Anwendungsfeldern eingesetzt werden. Die Fähigkeit zur präzisen, verlässlichen und überprüfbaren Informationsgenerierung ist daher ein wesentliches Qualitätsmerkmal solcher Systeme. Es bedarf integrierter Ansätze, um Halluzinationen systematisch zu minimieren, angefangen bei der Auswahl der Trainingsdaten über angepasste Trainingsmethoden bis hin zur sorgfältigen Abstimmung der Dekodierungsmechanismen (Huang et al. 2025, 1 ff).

4.3.5 Urheberrecht

Ein leistungsfähiges KI-Modell ist nur so gut wie die Daten, mit denen es trainiert wurde. Ohne qualitativ hochwertige und ausreichend umfangreiche Trainingsdaten ist selbst das beste Modell nicht einsetzbar. Daraus ergibt sich die zentrale Frage, woher diese Daten stammen und ob bei ihrer Nutzung urheberrechtlich relevante Handlungen vorgenommen werden (Käde 2021, S. 61).

Maschinelles Lernen folgt in der Regel einem mehrstufigen Prozess, der die Sammlung, Aufbereitung, das Training und die Anwendung von Daten umfasst. Die erste Phase, also die Erhebung und Organisation der Daten, wird in diesem Zusammenhang nicht weiter vertieft, da sie im deutschen Recht insbesondere durch den Schutz von Datenbanken gemäß § 4 Absatz 2 UrhG sowie den §§ 87a ff. UrhG geregelt ist. Wesentlich komplexer ist hingegen die urheberrechtliche Bewertung der nachfolgenden Schritte, insbesondere der Datenverarbeitung im Rahmen des Modelltrainings. Je nach technischer Ausgestaltung erfolgt die Aufbereitung der Daten entweder separat vor dem Training oder direkt innerhalb des Trainingsprozesses. Erfolgt eine vorherige und dauerhafte Speicherung der aufbereiteten Daten, etwa in Form transformierter Text- oder Bilddaten, kann dies eine Vervielfältigung im Sinne des § 23 UrhG darstellen. In solchen Fällen ist eine rechtliche Grundlage erforderlich, beispielsweise durch eine Lizenz oder eine gesetzlich vorgesehene Schrankenregelung. Anders ist die Lage, wenn die Daten lediglich flüchtig verarbeitet werden. Das ist der Fall, wenn sie nicht dauerhaft gespeichert, sondern direkt im Arbeitsspeicher verarbeitet und lediglich zur Anpassung der Modellparameter genutzt werden. In solchen Konstellationen liegt in der Regel keine urheberrechtlich relevante Vervielfältigung vor, da keine dauerhafte Kopie des Werkes entsteht. Auch im letzten Schritt des Prozesses, also bei der Anwendung des trainierten Modells, besteht in der Regel kein urheberrechtlicher Verstoß, sofern das Modell keine Trainingsdaten in identischer Form wiedergibt. Die von der KI generierten Ausgaben beruhen typischerweise auf generalisierten Mustern und stellen somit keine exakte Reproduktion dar. Ein Sonderfall liegt jedoch vor, wenn ein Modell so trainiert oder manipuliert wurde, dass es geschützte Inhalte, wie etwa ein Bild oder einen Text, exakt reproduziert. In einem solchen Fall können selbst die gespeicherten Modellparameter als eine Form der Vervielfältigung betrachtet werden, wodurch eine Anwendung des § 23 UrhG gerechtfertigt sein kann (Käde 2021, 61 ff).

Gerade im Kontext des Requirements Engineering ergeben sich daraus nicht nur technische, sondern auch rechtliche und ethische Herausforderungen. Der Einsatz KI-gestützter Verfahren zur Anforderungsanalyse wirft Fragen nach der Verantwortlichkeit, der Nachvollziehbarkeit der Datenverarbeitung sowie der rechtlichen Zuweisung geistigen Eigentums auf. Diese Aspekte dürfen im Entwicklungsprozess nicht außer Acht gelassen werden (Ihalainen 2018, 1ff).

5 UMSETZUNG

Die Umsetzung der Forschungsfrage erfolgt im Rahmen eines Design-Science-Research-Ansatzes, der die Entwicklung eines Artefakts zur Verbesserung von Akzeptanzkriterien mittels KI vorsieht. Dieses Kapitel beschreibt den methodischen Rahmen, das Umsetzungskonzept und die technologischen Grundlagen.

5.1 Design Science Research

Ein geeigneter methodischer Rahmen für die Umsetzung dieser Forschung ist der Design-Science-Ansatz. Dieser wurde auch in vergleichbaren Arbeiten, wie beispielsweise in der Studie zu ReqGen (Zhao et al. 2023, 1ff), erfolgreich angewendet. Zur Evaluation des entwickelten Artefakts wird eine quantitative Umfrage durchgeführt. In dieser vergleichen Stakeholder die Qualität von automatisiert generierten Akzeptanzkriterien mit jener von manuell erstellten. Ziel ist es, Rückschlüsse auf die praktische Anwendbarkeit und Akzeptanz der automatisierten Lösung zu ziehen.

Als Design-Science-Ansatz dient das Vorgehensmodell nach Jan vom Brocke, Alan Hevner und Alexander Maedche. Diese besteht aus sechs aufeinander aufbauenden Aktivitäten:

1. **Identifikation und Definition des relevanten Problems (Problem Identification and Motivation):** In dieser Phase wird das Forschungsproblem klar definiert und seine Relevanz für die Praxis und Forschung aufgezeigt. Hier wird die Notwendigkeit der Verbesserung von Akzeptanzkriterien durch KI basierend auf den in der Einleitung dargelegten Problemen (mangelhafte Software, wirtschaftlicher Schaden, manuelle Prozesse) begründet.
2. **Festlegung der Zielsetzungen für eine potenzielle Lösung (Objective of a Solution):** Es werden die spezifischen Ziele für das zu entwickelnde Artefakt formuliert. Dies beinhaltet die Definition, was das Artefakt erreichen soll, z. B. die Generierung qualitativ hochwertigerer Akzeptanzkriterien aus Produktdaten.
3. **Entwurf und Entwicklung eines Artefakts (Design and Development):** Basierend auf den Zielen und dem theoretischen Wissen (aus dem Literaturreview) wird das Artefakt konzipiert und implementiert. Dies umfasst die Auswahl der KI-Technologien, die Gestaltung der Systemarchitektur und die Implementierung des Prototyps.
4. **Demonstration des Artefakts in einem Anwendungskontext (Demonstration):** Das entwickelte Artefakt wird in einem relevanten Anwendungskontext demonstriert, um seine Funktionsweise und Machbarkeit zu zeigen. Dies kann durch Fallstudien, Beispiele oder Pilotanwendungen geschehen.
5. **Evaluation der Lösung hinsichtlich der Zielerreichung (Evaluation):** Das Artefakt wird systematisch evaluiert, um zu beurteilen, inwieweit es die zuvor definierten Ziele erreicht.

Dies kann durch quantitative oder qualitative Methoden erfolgen, wie beispielsweise eine Umfrage zur Qualität der generierten Akzeptanzkriterien.

6. **Kommunikation der Ergebnisse (Communication):** Die Forschungsergebnisse, das Artefakt und die Erkenntnisse aus der Evaluation werden der wissenschaftlichen Gemeinschaft und der Praxis kommuniziert. Dies geschieht durch die vorliegende Masterarbeit (vom Brocke et al. 2020, 4ff.).

5.2 Umsetzungskonzept

Das entwickelte System verfolgt das Ziel, User Stories automatisiert mit relevantem Kontext aus bestehenden Dokumentationen anzureichern und daraus präzise, testbare Akzeptanzkriterien zu generieren. Um dieses Ziel zu erreichen, werden im Folgenden die zentralen funktionalen und nicht-funktionalen Anforderungen beschrieben, die die Grundlage für Konzeption, Implementierung und Evaluation des Prototyps bilden.

Die Anforderungen orientieren sich an den typischen Verarbeitungsschritten innerhalb des Gesamtprozesses von der Dokumentaufnahme über die semantische Analyse bis hin zur Generierung der Ergebnisse. Sie definieren somit, welche Fähigkeiten das System besitzen muss, um eine robuste, nachvollziehbare und erweiterbare Verarbeitungspipeline sicherzustellen.

5.2.1 Anforderungen

Nr.	Erklärung
1	Dokumentaufnahme: Das System kann unterschiedliche Dokumentformate (z. B. PDF, Markdown, HTML) einlesen und deren Inhalte in ein einheitliches Textformat überführen.
2	Fehlertoleranz: Fehlerhafte oder nicht lesbare Dokumente werden erkannt, protokolliert und blockieren nicht den Gesamtprozess.
3	Vektorisierung: Eingelesene Texte werden in dichte Vektorrepräsentationen überführt, um semantische Ähnlichkeiten vergleichbar zu machen.
4	Indizierung: Die erzeugten Vektoren werden in einem Index gespeichert, der eine schnelle semantische Suche ermöglicht.
5	Kontextsensitive Suche: Für eine User Story werden über den Index relevante Dokumentpassagen identifiziert und zurückgegeben.
6	Prompt-Generierung: User Story und gefundener Kontext werden zu einer Eingabeaufforderung für ein Sprachmodell kombiniert.
7	Generierung von Akzeptanzkriterien: Das Sprachmodell erstellt aus User Story und Kontext eine Liste klarer, testbarer Akzeptanzkriterien.
9	Benutzereingabe: Das System nimmt User Stories über eine interaktive Eingabe entgegen.
10	Konfigurierbarkeit: Zugangsdaten und Konfigurationswerte werden über Umgebungsvariablen verwaltet und auf Vollständigkeit überprüft.

Tabelle 2: DSR-Anforderungen

5.2.2 Prozess der Datenverarbeitung



Abbildung 9: Prozess der Datenverarbeitung

Der dargestellte Prozess beschreibt einen methodischen Ansatz zur automatisierten Anreicherung von User Stories mit domänenspezifischem Wissen aus vorhandener Dokumentation. Ziel ist es, durch eine Kombination aus semantischer Textverarbeitung und Large-Language-Models (LLMs) eine inhaltlich präzisere Unterstützung für die Softwareentwicklung zu gewährleisten. Der Prozess umfasst die folgenden Schritte:

1. Einlesen der Dokumente

Zunächst werden vorhandene Dokumente, in diesem Fall die Nutzerdokumentation, eingelesen. Diese Dokumente bilden den Wissenspool, auf dem im weiteren Verlauf zurückgegriffen wird. Da die Dokumente häufig in unterschiedlichen Formaten (z. B. PDF, DOCX, Markdown) vorliegen, ist eine Vereinheitlichung in ein maschinenlesbares Textformat erforderlich.

2. Input der User Story

Im nächsten Schritt erfolgt die Eingabe einer User Story. Diese dient als Ausgangspunkt für die Kontextanreicherung und beschreibt das Anliegen der Kunden in strukturierter Form. Diese Eingabe geschieht über die Kommandozeile.

3. Vektorisierung der Texte

Sowohl die eingelesenen Dokumente als auch die User Story werden in Vektorrepräsentationen überführt. Dabei kommen sogenannte Embeddings zum Einsatz, die semantische Eigenschaften von Texten im hochdimensionalen Raum abbilden. Durch diese Transformation wird es möglich, inhaltliche Ähnlichkeiten zwischen Textfragmenten zu berechnen und nicht nur auf identische Schlüsselwörter zurückzugreifen.

4. Suche relevanter Dokumentation

Auf Basis der vektorisierten User Story erfolgt eine semantische Suche innerhalb der Dokumente. Hierbei werden diejenigen Textstellen identifiziert, die inhaltlich am nächsten an der User Story liegen. Im gegebenen Beispiel könnten dies Abschnitte der Dokumentation sein, die Methoden zur Ladezeitmessung oder vorhandene Funktionen für das Monitoring beschreiben. Das Ergebnis ist eine Menge an relevanten Dokumenten oder Dokumentausschnitten, die zur Bearbeitung der User Story herangezogen werden können.

5. Erstellung des Prompts für die OpenAI API

Im letzten Schritt wird ein zusammengesetzter Prompt generiert, der drei Komponenten umfasst: die Instruktion an das Modell (näher beschrieben: 5.3.6 Prompts) den extrahierten Kontext aus der Dokumentation, sowie die ursprüngliche User Story. Diese kombinierte Eingabe wird an die OpenAI API übergeben, sodass eine kontextualisierte Antwort entsteht, die sowohl die

Anforderungen der User Story berücksichtigt als auch konsistent mit der vorhandenen Dokumentation ist.

Die verwendeten Technologien und Verfahren sind in nachfolgendem Kapitel genauer beschreiben.

5.3 Technologie

In der vorliegenden Arbeit wird das technische Fundament für den Einsatz von KI im Requirements Engineering in drei Schritten dargelegt. Zunächst gehe ich auf das verwendete KI-Modell ein und erläutere, welche Ansätze in der Literatur dominieren und welche Modelle für diese Arbeit relevant sind. Darauf aufbauend beschreibe ich die notwendigen Vektorisierungsverfahren, mit denen Textdaten in eine für KI-Modelle verarbeitbare Form überführt werden. Im dritten Schritt stelle ich die Suchverfahren vor, die es ermöglichen, in den erzeugten Vektorräumen effizient nach inhaltlich passenden Informationen zu suchen.

5.3.1 Programmiersprache

Als Programmiersprache wird Python gewählt, da sie eine gut lesbare und übersichtliche Syntax bietet, was die Verständlichkeit und Pflege des Codes erleichtert. Außerdem steht eine große Anzahl an Open-Source-Bibliotheken zur Verfügung, die viele Aufgaben bereits abdecken und so die Entwicklung vereinfachen und beschleunigen.

5.3.2 Einlesen von Dokumenten

Für den Import der Quelldokumente wird ein mehrstufiger Prozess angewendet. Die Verarbeitung der Dateien erfolgt abhängig vom jeweiligen Format: PDF-Dokumente werden mithilfe der Bibliothek PyPDF2 eingelesen, während HTML-Dateien unter Verwendung von BeautifulSoup verarbeitet werden. Markdown-Dateien werden ohne externe Bibliotheken geladen und anschließend mithilfe von regulären Ausdrücken (Regex) systematisch bereinigt, um eine konsistente Textbasis sicherzustellen.

Dateien, deren Format nicht den unterstützten Typen (PDF, Markdown oder HTML) entspricht, werden nicht verarbeitet. Stattdessen wird ein entsprechender Fehlerhinweis in der Kommandozeile ausgegeben, wodurch eine transparente Fehlerbehandlung gewährleistet wird.

Alle validen Dokumente werden rekursiv aus dem angegebenen Verzeichnis geladen, in reinen Text konvertiert und für die nachfolgende Vektorisierung gespeichert. Auf diese Weise entsteht eine einheitliche und strukturierte Textbasis, die als Grundlage für weitere Verarbeitungsschritte dient.

5.3.3 Verwendete KI

In der systematischen Literaturübersicht *Generative AI for Requirements Engineering: A Systematic Literature Review* wurde herausgearbeitet, dass ein Großteil der analysierten Arbeiten den Fokus auf die GPT-Modellreihe legte, insbesondere auf GPT-3.5 und GPT-4. Konkret entfielen 67,3 % der untersuchten Beiträge auf diese Modelle. Diese Dominanz ist vor allem durch das umfassende Pretraining erklärbar, das eine enge Passung zu den typischen Aktivitäten im Requirements Engineering aufweist. Andere Modelle, wie Claude oder PaLM, fanden mit 18,4 % deutlich weniger Beachtung, während Open-Source-Modelle wie LLaMA mit lediglich 7,1 % stark unterrepräsentiert waren. Zudem wurde festgestellt, dass 90 % der Projekte vortrainierte Modelle ohne zusätzliches Fine Tuning für spezifische Anwendungsfälle einsetzten (Cheng et al. 2024, 11ff).

Eine weitere Studie verglich die manuelle Erstellung von *Software Requirements Specifications* (SRS) mit der KI-basierten Generierung. Dabei wurden die Modelle CodeLlama-34B und ChatGPT-4 untersucht. Die Ergebnisse zeigten, dass CodeLlama-34B detailliertere Ausgaben lieferte, jedoch eine erhöhte Anfälligkeit für Halluzinationen aufwies. Im Gegensatz dazu erzeugte ChatGPT-4 präzisere und kürzere Antworten, die zwar weniger fehleranfällig waren, jedoch aufgrund ihrer Kürze teilweise unvollständig blieben (M. Krishna et al. 2024, 7f).

Vor dem Hintergrund dieser Befunde wurde in der vorliegenden Arbeit das Modell *gpt-5-mini-2025-08-07* gewählt. Ausschlaggebend hierfür war, dass die in der Literatur beschriebenen Charakteristika auch auf diese Modellversion zutreffen. Darüber hinaus zeichnet sich *gpt-5-mini-2025-08-07* durch kurze Antwortzeiten bei gleichzeitig hohen Reasoning-Fähigkeiten aus, was eine hohe Passgenauigkeit für den zugrunde liegenden Use Case sicherstellt. Ein weiterer Vorteil liegt im erweiterten Kontextfenster von 400.000 Zeichen (OpenAI 2025).

Neben den technischen Eigenschaften ist auch die Marktstellung von Relevanz. In Österreich zeigt sich eine deutliche Dominanz von ChatGPT: 86 % der Befragten nutzen die Plattform. Deutlich abgeschlagen folgen Perplexity mit 6 %, Microsoft Copilot mit 5,5 % und Gemini mit lediglich 1 % Marktanteil (AI Chatbot Market Share Austria 2025).

5.3.4 Vektorisierungsverfahren

Ein zentraler Schritt dabei ist die Vorverarbeitung der Daten, die unter anderem das Entfernen von Rauschen umfasst. Dazu zählen beispielsweise Absatzzeichen sowie inkonsistente Groß- und Kleinschreibung (Abbas et al., 2025, S. 64). Erst auf dieser Grundlage lassen sich die Texte in numerische Repräsentationen überführen, die für maschinelles Lernen nutzbar sind.

Grundsätzlich lassen sich vier Ansätze der Vektorisierung unterscheiden: kategorische, gewichtete, verteilte und kontextuelle Repräsentationen.

Die kategorische Repräsentation stellt Wörter in Form binärer Vektoren dar, wobei die Position eines Wortes im Text ausschlaggebend ist. Bekannte Verfahren dieser Art sind das *One-Hot-Encoding* sowie das *Bag-of-Words-Modell* (Abbas et al. 2025, S. 65).

Die gewichtete Repräsentation geht darüber hinaus, indem sie Wörter anhand ihrer Häufigkeit im jeweiligen Korpus unterschiedlich stark gewichtet. Ein prominentes Verfahren ist hier *Term Frequency–Inverse Document Frequency* (TF-IDF). Dabei wird durch die Berechnung relativer Worthäufigkeiten ein Maß für die Relevanz einzelner Wörter innerhalb eines Textes bestimmt (Abbas et al. 2025, 65f).

Einen deutlichen Fortschritt stellen die verteilten Repräsentationen dar. Sie erfassen erstmals semantische und syntaktische Beziehungen zwischen Wörtern. Das zentrale Konzept besteht darin, dass Wörter mit ähnlicher Bedeutung im Vektorraum dichter beieinanderliegen als Wörter ohne Zusammenhang (Abbas et al. 2025, 66ff).

Die kontextuellen Repräsentationen sind ein Spezialfall der verteilten Verfahren, werden jedoch gesondert betrachtet. Anders als klassische Methoden wie *FastText* oder *GloVe* basieren sie auf tiefen neuronalen Netzen und *Attention-Mechanismen*. Dadurch werden Wörter dynamisch im Kontext der umgebenden Begriffe eingebettet, was eine differenzierte Verarbeitung sprachlicher Besonderheiten ermöglicht (Abbas et al. 2025, 68f).

Für die vorliegende Arbeit wird konkret der Transformer *Sentence Bidirectional Encoder Representations from Transformers* (SBERT) eingesetzt. Dieses vortrainierte Netzwerk basiert auf einem großen Korpus, das unter anderem die englischsprachige Wikipedia und zahlreiche Bücher mit insgesamt rund 2,5 Milliarden Wörtern umfasst. Gegenüber dem ursprünglichen *BERT*-Modell wurde SBERT gewählt, da es durch zusätzliches Fine Tuning ein verbessertes Verständnis auf Satzebene bietet (Abbas et al. 2025, S. 69).

5.3.5 Suchverfahren

Im vorherigen Abschnitt wurde die Vektorisierung der Produktdokumentation beschrieben. Aufbauend darauf erfolgt nun eine Clustersuche, die mithilfe der Python-Bibliothek Faiss umgesetzt wird. Faiss ist eine spezialisierte Bibliothek zur effizienten Einbettung und Durchsuchung hochdimensionaler Vektorräume. Sie zählt zu den am weitesten verbreiteten Implementierungen im Bereich der Vektorsuche (Douze et al. 2024, 1f).

Die Stärke von Faiss liegt in ihrer Flexibilität: Die Bibliothek unterstützt unterschiedliche Distanzmaße, Suchverfahren und Komprimierungsmethoden, wodurch sich das Vorgehen präzise an die jeweilige Anwendungssituation anpassen lässt. In der vorliegenden Arbeit wird für die Distanzberechnung die *Cosine-Similarity* verwendet. Dieses Maß erweist sich als besonders robust gegenüber sowohl sehr nah beieinanderliegenden als auch stark voneinander entfernten Vektoren. Damit eignet es sich in besonderem Maße für die Einbettung von Produktdokumentationen, da terminologische Feinabstufungen häufig sehr ähnliche Vektorrepräsentationen erzeugen, während unterschiedliche Themengebiete deutlich weiter voneinander entfernt liegen (Douze et al. 2024, 3f).

Hinsichtlich der Suchstrategie wurde eine Brute-Force-Suche gewählt. Diese Vorgehensweise liefert die qualitativ besten Ergebnisse, während mögliche Performanceeinbußen aufgrund der überschaubaren Datenmenge in diesem Anwendungsfall von geringer Relevanz sind (Douze et al. 2024, S. 4). Auf Komprimierungsverfahren wird bewusst verzichtet, da bei der gegebenen

Datenmenge wenig Nutzen hinsichtlich Speicherverbrauch oder Geschwindigkeit zu erwarten ist und ein möglicher Arbeitsspeichergewinn als vernachlässigbar gilt (Douze et al. 2024, 6f).

Als Suchverfahren wird schließlich die Methode Inverted Files eingesetzt, die in den Bereich der non-exhaustive searches fällt. Im Gegensatz zu exhaustiven Verfahren, bei denen sämtliche Vektoren berücksichtigt werden, fokussiert sich die Inverted Files-Methode auf relevante Teilbereiche. Dies wird dadurch erreicht, dass die Vektoren bereits während der Indexierung in Cluster organisiert werden. Bei einer anschließenden Anfrage werden nur die Cluster durchsucht, die als relevant identifiziert wurden (Douze et al. 2024, 9f).

5.3.6 Prompts

Im Rahmen des Prompting-Prozesses wurde die Methode des Zero-Shot Prompting angewendet. Die nachfolgende Prompt-Schablone wurde unter Verwendung des Platzhalters {Story} mit der jeweils vorliegenden User Story befüllt. Auf Grundlage dieser User Story konnten mithilfe der in den vorherigen Unterkapiteln beschriebenen Verfahren relevante Funktionen identifiziert und im Abschnitt {context} bereitgestellt werden. Durch die gezielten Anweisungen „Liste nicht im Format given when then“ sowie „Liste sollte sich auf das Wesentliche beschränken“ wurde das Ausgabeformat der generierten Liste entsprechend angepasst. Auf diese Weise wurde sichergestellt, dass die Ergebnisse in einer für den jeweiligen Use Case geeigneten und fokussierten Form vorliegen. Darüber hinaus wurde die zusätzliche Einschränkung „Liste nicht im Format given when then“ integriert, um das Format der in der Firma evon gebräuchlichen Akzeptanzkriterien möglichst genau abzubilden.

"Du bist ein erfahrener Product Owner. "

"Basierend auf der folgenden User Story und Projektdokumentation, formuliere klare und testbare Akzeptanzkriterien.\n"

"User Story:\n{story}\n\n"

"Kontext (relevante Dokumente):\n{context}\n\n"

"Akzeptanzkriterien (als stichpunktartige Liste):\n"

"Liste nicht Format given when then \n"

"Liste sollte sich auf das wesentliche Beschränken"

6 EVALUIERUNG DES ARTEFAKTS

Das folgende Kapitel präsentiert die empirischen Ergebnisse der Untersuchung, in der die Qualität von durch künstliche Intelligenz generierten Akzeptanzkriterien mit von Menschen formulierten, verglichen wird. Ziel ist es, herauszufinden, inwiefern KI-gestützte Ansätze dazu beitragen können, die Formulierung und Bewertung von User Stories im agilen Entwicklungsprozess zu verbessern.

Zuerst zeigt Abschnitt 6.1 konkrete Beispiele aus dem XAMControl-Projekt. Jede User Story gibt es in zwei Varianten: einmal menschlich formuliert und einmal von einer KI erstellt. Der Vergleich macht sichtbar, wo sich Struktur, Sprache und Detailgrad ähneln oder unterscheiden. In Abschnitt 6.2 wird anschließend erklärt, wie die Qualität der Akzeptanzkriterien bewertet wurde. Dazu gehören eine Befragung von Entwickler*Innen sowie automatische Verfahren zur Auswertung.

Die darauffolgenden Abschnitte (6.3 bis 6.5) widmen sich der Auswertung der empirischen Ergebnisse, der maschinellen Analyse der semantischen Übereinstimmung sowie der Interpretation der Freitextkommentare der Teilnehmenden. Abschließend werden in Abschnitt 6.6 die zentralen Erkenntnisse zusammengefasst und im Hinblick auf die Forschungsfrage interpretiert.

6.1 Ergebnisse

Im folgenden Abschnitt werden ausgewählte User Stories aus dem XAMControl-Projekt dargestellt, die sowohl in menschlicher Formulierung als auch in einer durch KI generierten Variante vorliegen. Ziel dieser Gegenüberstellung ist es, Unterschiede und Gemeinsamkeiten in der Ausdrucksweise, Struktur und Detaillierung zu analysieren.

Um die Vergleichbarkeit zu gewährleisten, werden die Akzeptanzkriterien in tabellarischer Form aufbereitet. Dabei werden jeweils die menschlich formulierten Kriterien den maschinell erzeugten Kriterien gegenübergestellt, sofern sie dasselbe Themengebiet behandeln. Auf diese Weise lassen sich Unterschiede in Präzision, Vollständigkeit und Klarheit systematisch herausarbeiten.

Die ausgewählten User Stories (6.1.1 bis 6.1.3) dienen als Fallbeispiele, die unterschiedliche technische Anforderungen abbilden – von der Ablösung einer Rendering-Engine über die Integration und Umbenennung von Komponenten bis hin zur Umsetzung neuer Kommunikationsmechanismen. Sie bilden die Grundlage für die spätere Diskussion darüber, in welchen Bereichen KI-gestützte Formulierungen Vorteile oder Einschränkungen gegenüber menschlich erstellten Anforderungen aufweisen.

6.1.1 User Story 1

Beschreibung:

Als XAMControl Entwickler möchte ich, dass CefSharp durch WebView2 ersetzt wird.

CefSharp wird als Preview und als Code-Editor der Web-Panels verwendet dass soll durch WebView2 ersetzt werden.

Der *XAMBrowser* in der *XAMStandardLibrary* verwendet auch CefSharp dort soll das auch ersetzt werden.

- Voranalyse wo CefSharp überall verwendet wird.-->Kommentar in Item
- Im Nuget *XAMControl.XAMWeb.WebBrowser* CefSharp durch WebView2 ersetzen
- In XAMControl Repo: Das neue *XAMControl.XAMWeb.WebBrowser* Nuget verwenden und CefSharp Referenzen löschen
- PluginManager

Akzeptanzkriterien:

Menschlich	Künstliche Intelligenz
Preview und Code-Editor für Web funktionieren weiterhin	Der Code-Editor der Web-Panels verwendet WebView2 und grundlegende Editor-Funktionen (Rendern, Editieren, Syntax-Anzeige, Suche) funktionieren wie zuvor.
XAMBrowser von XAMStandardLibrary funktioniert weiterhin	Der XAMBrowser in der XAMStandardLibrary nutzt WebView2 und verhält sich funktional ident zu vorherigem CefSharp-Browser.
PluginManager funktioniert weiterhin	PluginManager und betroffene Plugins referenzieren das neue NuGet und enthalten keine CefSharp-Referenzen mehr.
#36674 tritt nicht mehr auf	
	Voranalyse abgeschlossen und alle Stellen, an denen CefSharp verwendet wird, in den jeweiligen Items als Kommentar dokumentiert.
	Das NuGet-Paket XAMControl.XAMWeb.WebBrowser enthält WebView2 als Rendering-Engine und wurde veröffentlicht.
	Im XAMControl-Repo ist das neue NuGet eingebunden und alle CefSharp-Referenzen wurden entfernt.

	Web-Panels Preview verwendet WebView2 und stellt Inhalte visuell gleichwertig zur früheren CefSharp-Darstellung dar.
	Automatisierte Tests und manuelle Smoke-Checks für Preview, Editor und Browser laufen erfolgreich (kein Regressionsverhalten).
	Die gesamte Solution kompiliert erfolgreich ohne CefSharp-Abhängigkeiten.
	Keine CefSharp-DLLs oder NuGet-Pakete verbleiben in Projekt-Referenzen, Build-Outputs oder Deployment-Artefakten.
	Automatisierte Tests und manuelle Smoke-Checks für Preview, Editor und Browser laufen erfolgreich (kein Regressionsverhalten)
	Der Web-Deploy/Run-Workflow (Deploy auf XAMWebServer und Anzeige im Browser) funktioniert weiterhin mit WebView2.
	Alle zuvor vorhandenen Dokumentationen/Kommentare, die auf CefSharp verwiesen haben, wurden angepasst und den Wechsel zu WebView2 vermerkt.

Tabelle 3: User Story Akzeptanzkriterien 1

6.1.2 User Story 2

Beschreibung

Als XAMControl Entwickler möchte ich den XAMirisStarter von VMIS für evo4 übernehmen und damit den XAMirisProxy ablösen. Dazu soll auch der Name in XAMirisLauncher geändert werden.

Folgende Punkte müssen dazu durchgeführt werden:

- Merge einer älteren Version des XAMirisStarter von VMIS (Projekt) nach XAMControl (nur evo4) -->ältere Version deshalb, da der Area Service dort noch nicht enthalten ist (prüfen ob es von dieser Version zur neuesten viele Änderungen (Bugfixes) gegeben hat)
 - Muss als Autostart eingetragen werden
- Alle Codeteile müssen auf den neuen Namen (XAMirisLauncher) umbenannt werden

- Features des XAMirisStarter sollen alle übernommen werden
- Bei XAMiris Start prüfen ob Alarme direkt abgerufen werden (IsProxy)
- XAMSpeech muss dazu auch übernommen werden (wird in eigener US durchgeführt - >XAMSynthesizer)
- Anpassen des Installers (wird mit eigener US durchgeführt)

Voraussetzung: US User Story 38309: XAMServer.Api: IdentityService - Erweiterung um den temporären Token für das Login nutzen zu können ist bereits durchgeführt.

Akzeptanzkriterien:

Menschlich	Künstliche Intelligenz
XAMirisLauncher (in VMIS XAMirisStarter) ersetzt XAMirisProxy und alle Funktionalitäten des XAMirisProxy	Alle Funktionalitäten des ursprünglichen XAMirisStarter sind im XAMirisLauncher vollständig übernommen.
Alle Codeteile wurden auf den neuen Namen XAMirisLauncher geändert	Alle Code-, Dienst- und Konfigurationsnamen wurden konsistent in XAMirisLauncher umbenannt.
Launcher mit Startargument "xamserver=ServerIp" starten	Der bisherige XAMirisProxy ist durch den XAMirisLauncher ersetzt; entsprechende Proxy-Start-/Steuerfunktionen sind erhalten.
Launcher meldet sich beim Start mit "Guest" user bei Server an	
Launcher holt sich die ClientConfig	
Launcher meldet sich mit dem in der ClientConfig definierten StartupUser an	
Launcher generiert für jede Iris einen temporären Key und startet Iris mit diesem Key in den Startargumenten	
Iris startet und meldet sich mit diesem Key an	Der XAMirisLauncher verwendet das temporäre Login-Token aus dem erweiterten IdentityService (US 38309).
	XAMirisStarter aus dem VMIS-Projekt ist in XAMControl (evo4) gemerged und als Windows-Autostart-Dienst eingetragen.
	Beim Start führt der XAMirisLauncher eine IsProxy-Prüfung durch und ruft bei IsProxy=false Alarme direkt ab.

	Schnittstellen zur Sprachausgabe (XAMSpeech / Vorbereitung für XAMSynthesizer) sind im XAMirisLauncher vorhanden.
	Alle Konfigurationsdateien und Einträge verweisen ausschließlich auf XAMirisLauncher und nicht mehr auf XAMirisProxy.
	Es treten keine funktionalen Regressionen gegenüber dem bisherigen XAMirisStarter/XAMirisProxy-Verhalten auf.

Tabelle 4: User Story Akzeptanzkriterien 2

6.1.3 User Story 3

Beschreibung

Nachfolgeitem von: #31843.

Es soll die zyklische Kommunikation mit dem neuen symbolische Zugriff (Deltalogic) umgesetzt werden. Auf Basis der Konfiguration sollen die entsprechenden Datenpunkte gelesen, geschrieben und R/W Zugriff ermöglicht werden.

Der Zugriff soll optimiert nach den Vorschlägen von deltalogic umgesetzt werden (ganze Mail im Anhang):

Die ganze Prozedur in einer Schleife zu durchlaufen ist nicht notwendig, wiederholt werden muss eigentlich nur die Lesefunktion Symbolic_ReadMixEx wenn sie zyklisch lesen wollen.

Sie stellen sich vorher die Variablen zusammen und schicken diese gemeinsam an Symbolic_ReadMixEx.

Wie sie die Werte nur einmal lesen wollen, genügt auch ein Aufruf.

Beim Zusammenstellen der Variablen, gibt es mehrere Möglichkeiten, wenn sie z.B. ein Array der 4000 DINT erstellen, können sie dieses Array mit einer Variablen lesen, dann geht das schnell.

Ansonsten stellen sie sich nur einmal alle gewünschten Variablen zusammen und schicken diese gemeinsam an Symbolic_ReadMixEx. Das Zusammenstellen der Variablen sollte nicht zyklisch erfolgen.

Akzeptanzkriterien:

Menschlich	Künstliche Intelligenz
Lesend/Schreibend/ReadWrite funktioniert	R/W-Berechtigungen werden bei Lese- und Schreiboperationen eingehalten

<p>Vergleichsmessungen durchgeführt und dokumentiert (CPU, Zykluszeiten)</p> <ul style="list-style-type: none"> • Werte: ~30.000 Datenpunkte (wie diese aufgeteilt wurden -> GBR/BDI) <ul style="list-style-type: none"> ○ multiple Channel: <ul style="list-style-type: none"> ▪ xxx Datenpunkte xx ms ▪ xxx Datenpunkte xx ms ▪ xxx Datenpunkte xx ms • nachhaltiger Aufbau in QS (damit diese Messungen bei Anpassungen im Treiber immer wieder gestartet werden können) MAF kann unterstützen 	
<p>Empfehlung für "high Performance" Dokumentieren in Treiberdoku (Arrays...)</p>	
<p>Kombination prüfen (Verbindungen Put/Get & Symbolic in unterschiedlichen Verbindungen)</p>	
<p>Direct RW ist f. symbolic Zugriff nicht relevant</p>	
<p>Datenpunkte müssen auch auf der richtigen XAMVariable gemappt werden (Werte prüfen)</p>	<p>Schreibzugriffe (Write) funktionieren für konfigurierte Datenpunkte und setzen Werte korrekt.</p> <p>Konfigurationsgemäße Lesezugriffe liefern alle erwarteten Datenpunkte korrekt.</p>
<p>Testaufbau/Vergleichsmessungen Basis</p>	
<p>Verbindungsausfälle werden richtig gehandelt</p>	
<p>Mischbetrieb (symbolic und wie bisher) ist möglich</p>	
<p>Computerressourcenverbrauch zeigt keine negativen Auffälligkeiten (CPU, RAM, Handles)</p>	

	Bulk-Reads (z. B. große DINT-Arrays) werden unterstützt und liefern korrekte Werte.
	Die Variablenliste wird einmalig vor dem Zyklus zusammengestellt und nicht zyklisch neu erzeugt.
	Zyklische Leseoperationen erfolgen durch einen einzigen gebündelten Aufruf von <code>Symbolic_ReadMixEx</code> pro Zyklus.
	Bulk-Reads (z. B. große DINT-Arrays) werden unterstützt und liefern korrekte Werte.
	Einmaliges Lesen (nicht zyklisch) liefert korrekte Ergebnisse bei nur einem Aufruf von <code>Symbolic_ReadMixEx</code> .
	Lese-/Schreibperformance ist optimiert durch gebündelte Aufrufe statt Einzelaufrufen pro Datenpunkt.
	Die Prozedur zum Zusammenstellen der Variablen erfolgt außerhalb der Zyklus-Schleife.
	Fehlerhafte oder nicht vorhandene Variablen werden erkannt und verursachen definierte Fehlermeldungen statt Systemabsturz.

Tabelle 5: User Story Akzeptanzkriterien 3

6.2 Methoden der Evaluierung

In meiner Arbeit betrachte ich die Qualität von User Stories als zentralen Faktor im Kontext agiler Softwareentwicklung. Um diese Qualität messbar zu machen, gehe ich zweistufig vor. Zunächst analysiere ich die Dimensionen von User Stories anhand der Kriterien Lesbarkeit, Verständlichkeit, Abgrenzbarkeit sowie technischer Richtigkeit. Diese Dimensionen dienen als theoretische Grundlage für die Bewertung und bilden die Basis des Fragebogens, der in den Entwicklungsteams der *evon* GmbH. eingesetzt wird. Im Anschluss daran werden die Methoden zur maschinellen Validierung vorgestellt, mit denen generierte Akzeptanzkriterien automatisch bewertet werden können.

6.2.1 Fragebogen

Zunächst analysiere ich die Dimensionen von User Stories Anhand Lesbarkeit, Verständlichkeit und Abgrenzbarkeit. Lesbarkeit beschreibt, inwieweit ein Entwicklungsteam in der Lage ist, auf Grundlage einer User Story eine fundierte Aufwandsabschätzung vorzunehmen. Eine gut lesbare Story ist klar formuliert, strukturiert und vermeidet Unklarheiten. Verständlichkeit bedeutet, dass die User Story so verfasst ist, dass auch nicht-technische Stakeholder den erwarteten Mehrwert nachvollziehen können. Dies fördert den interdisziplinären Austausch und erleichtert die gemeinsame Formulierung von Akzeptanzkriterien und Testfällen. Abgrenzbarkeit bezieht sich darauf, dass sich eine Story auf eine klar abgegrenzte Funktionalität konzentriert, die innerhalb einer Iteration umgesetzt werden kann. Jede Story sollte zudem in sich geschlossen sein und einen erkennbaren Nutzen liefern. Darüber hinaus sind technische Details ein weiterer wesentlicher Aspekt. Eine qualitativ hochwertige User Story sollte dem Entwicklungsteam ausreichende technische Informationen bieten, um das zugrunde liegende Problem zu verstehen und geeignete Lösungsansätze zu entwickeln (Rahman und Zhu 2024, 2f).

Die zugrunde liegenden Bewertungsdimensionen wurden aus dem Beitrag Automated user story generation with test case specification using large language model von Tajmilur Rahman und Yuecai Zhu übernommen und in Teilen konzeptuell angepasst. Die ursprüngliche Dimension der Spezifizierbarkeit wurde in Abgrenzbarkeit umbenannt, da diese Bezeichnung die intendierte Bedeutung der zugrunde liegenden Bewertungsfrage präziser widerspiegelt.

Der Fragebogen besteht aus 50 Fragen. Die Audienz besteht aus den MitarbeiterInnen des Softwareentwicklungsteams der Firma evon. Der Fragebogen startet mit einigen Hintergrundfragen. Als Erstes wird die Rolle abgefragt. Hier wird zwischen Entwickler*Innen, Architekt*Innen, Tester*Innen und Sonstigen unterschieden. Die Unterscheidung der Rolle soll Einblick in die Wichtigkeit von Teilaspekten in den verschiedenen Gruppen geben. Als Nächstes wird die Berufserfahrung abgefragt hier wird zwischen 0-1 Jahren, 2-4 Jahren, 5-9 Jahren oder über 10 Jahren unterschieden. Dass soll Einblick geben, wie der Zusammenhang von Domänenwissen zu der Bewertung ist.

Im Anschluss bekommen die Teilnehmer*Innen 3 Storys vorgelegt. Zu diesen Storys gibt es jeweils generierte Akzeptanzkriterien und von Domänen Expert*Innen erarbeitete in unserem Fall nach den Refinement des gesamten Teams. Beide Akzeptanzkriterien werden danach nach den RUdT-Faktoren mit der Likert Skala bewertet.

Nr.	Frage	Rudt Teilbereich
1	Die Kriterien sind sprachlich gut lesbar.	Lesbarkeit
2	Syntax und Formatierung sind konsistent.	Lesbarkeit
3	Die verwendete Terminologie ist einheitlich.	Lesbarkeit
4	Die Kriterien machen die Intention der Story klar.	Verständlichkeit
5	Die Aussagen sind eindeutig und vermeiden Mehrdeutigkeit.	Verständlichkeit
6	Wichtige Begriffe sind verständlich oder aus dem Kontext ableitbar.	Verständlichkeit
7	Die Kriterien sind so präzise, dass sie als Testfälle implementierbar sind.	Abgrenzbarkeit
9	Die Kriterien sind mit der beschriebenen Systemumgebung vereinbar.	Technische Richtigkeit
10	Es werden keine technisch falschen Annahmen getroffen.	Technische Richtigkeit
11	Gesamtqualität der Kriterien Tests.	Gesamturteil
12	Würden Sie diese Kriterien Tests in ein Backlog übernehmen.	Entscheidung
13	Freitextkommentar.	Kommentar

Tabelle 6: Fragenkatalog

6.2.2 Maschinelle Validierung

Im Rahmen der Literaturrecherche wurden verschiedene Methoden zur Validierung maschinell generierter Texte identifiziert. Besonders häufig kamen drei Verfahren zum Einsatz: BLEU, ROUGE und BERT-Score. Im Folgenden werden diese Ansätze erläutert und hinsichtlich ihrer Eignung zur Bewertung generierter Akzeptanzkriterien eingeordnet.

BLEU (Bilingual Evaluation Understudy) ist eine der frühesten und bekanntesten Metriken zur Bewertung der Qualität maschinell erzeugter Texte, insbesondere im Bereich der maschinellen Übersetzung. Der BLEU-Score basiert auf dem Vergleich von n-Grammen zwischen dem generierten Text und einem oder mehreren Referenztexten, etwa von professioneller Übersetzungssoftware. Dabei gilt: Je höher die Übereinstimmung der n-Gramme, desto höher der Score. Der Score selbst liegt zwischen 0 (keine Übereinstimmung) und 1 (vollständige Übereinstimmung). Als Maß verwendet BLEU die Precision, also den Anteil der n-Gramme im generierten Text, die auch im Referenztext vorkommen. Obwohl BLEU in der maschinellen Übersetzung etabliert ist, ist seine Aussagekraft im Kontext der automatisierten Generierung von Akzeptanzkriterien eingeschränkt, da hierbei oft keine wortwörtliche Übereinstimmung, sondern inhaltliche Ähnlichkeit entscheidend ist (Sun et al. 2023, S. 7f).

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) wurde ursprünglich zur Bewertung automatischer Textzusammenfassungen entwickelt und findet ebenfalls Anwendung in der maschinellen Übersetzung. Wie BLEU basiert ROUGE auf der Überlappung von n-Grammen zwischen Referenz- und Zieltext. Anders als BLEU verwendet ROUGE jedoch den Recall als Metrik, also den Anteil der n-Gramme aus dem Referenztext, die im generierten Text enthalten sind. ROUGE umfasst mehrere Varianten: ROUGE-N misst die Überlappung von n-Grammen, ROUGE-L berücksichtigt die längste gemeinsame Teilfolge, und ROUGE-S verwendet sogenannte „Skip-Bigramme“, die auch dann Übereinstimmung erlauben, wenn zwischen zwei Wörtern weitere Wörter eingefügt wurden. Aufgrund dieser differenzierten Betrachtung inhaltlicher Überlappungen ist ROUGE besser geeignet als BLEU, um die Qualität generierter Akzeptanzkriterien zu beurteilen (Sun et al. 2023, S. 7f).

BERT-Score unterscheidet sich grundlegend von den n-grammbasierten Ansätzen. Statt auf Oberflächenähnlichkeit zu setzen, nutzt BERT-Score kontextuelle Wortrepräsentationen, die mit Hilfe vortrainierter Sprachmodelle wie BERT erzeugt werden. Jedes Wort oder Wortteil wird dabei als Vektor (Token) im semantischen Raum abgebildet. Die Ähnlichkeit zwischen Referenz- und Zieltext wird anhand der Cosine Similarity dieser Vektoren berechnet. Diese Metrik gibt an, wie ähnlich sich die Bedeutungen zweier Wörter oder Sätze im Kontext sind, unabhängig von ihrer konkreten Formulierung. BERT-Score erlaubt damit eine semantische Bewertung der Textqualität und eignet sich besonders für Anwendungsfälle, in denen inhaltliche, nicht jedoch formale Übereinstimmung entscheidend ist (Zhang et al. 2019, S. 3ff).

Zusammenfassend lässt sich feststellen, dass BLEU zwar eine etablierte Metrik darstellt, jedoch vorwiegend für wortwörtliche Übersetzungsaufgaben geeignet ist. ROUGE bietet durch seine Varianten eine flexiblere Bewertung inhaltlicher Übereinstimmungen. BERT-Score schließlich geht über die reine Textoberfläche hinaus und ermöglicht eine semantisch fundierte Qualitätsbewertung, was für die Analyse generierter Akzeptanzkriterien besonders relevant ist.

6.3 Auswertung Fragebogen

Im Rahmen dieser Untersuchung wurde eine Umfrage unter den Entwicklungsmitarbeitenden der Firma evon durchgeführt, um die Wahrnehmung und Bewertung von Akzeptanzkriterien zu erfassen, die entweder manuell oder mithilfe künstlicher Intelligenz erstellt wurden. Ziel der Erhebung war es, Unterschiede in den Dimensionen Lesbarkeit, Verständlichkeit, Abgrenzbarkeit und technischer Richtigkeit zu identifizieren sowie die generelle Einschätzung der Teilnehmenden hinsichtlich der Praxistauglichkeit der Kriterien zu analysieren.

Die Umfrageergebnisse bilden die Grundlage für die Bewertung, inwiefern KI-gestützte Formulierungen als qualitativ gleichwertig oder sogar überlegen gegenüber menschlich erstellten Akzeptanzkriterien eingeschätzt werden. Zunächst wird die Stichprobe der Teilnehmenden beschrieben, um ein besseres Verständnis für die Zusammensetzung der Befragten und deren Erfahrungshintergrund zu schaffen (Abschnitt 6.2.1). Anschließend werden die zentralen Ergebnisse der Umfrage im Detail dargestellt und interpretiert (Abschnitt 6.2.2).

6.3.1 Beschreibung der Stichprobe

Die Umfrage wurde innerhalb der Entwicklung von XAMControl durchgeführt. Von insgesamt 42 Entwicklungsmitarbeitenden haben 19 Personen an der Umfrage teilgenommen. Dabei handelte es sich um 13 Entwicklerinnen und Entwickler, zwei Softwarearchitekt*Innen sowie eine Person aus dem Bereich Testing/Quality Assurance.

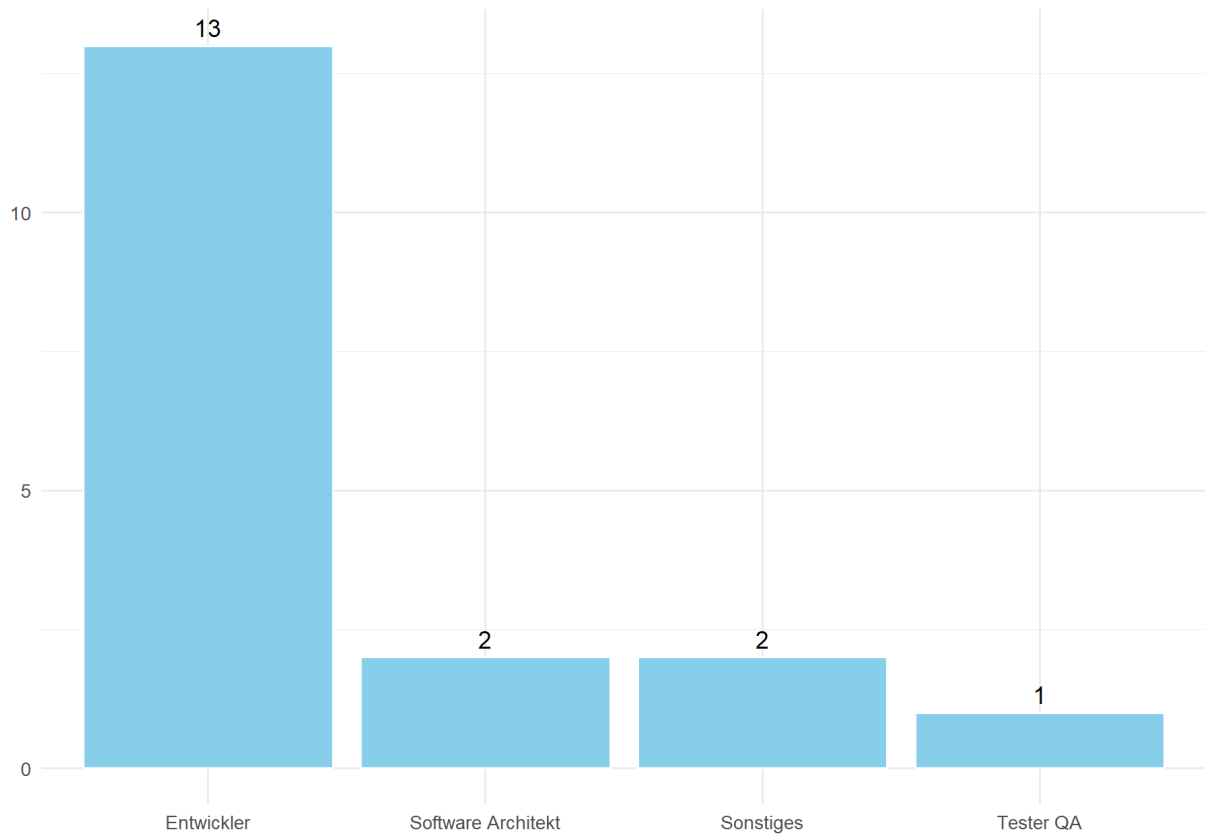


Abbildung 10: Rolle der Umfrage Teilnehmer*Innen

Die Auswertung des Dienstalters zeigt, dass eine Person über ein Dienstalter von null bis einem Jahr verfügt. Sechs Personen gehören der Kategorie mit zwei bis vier Jahren an, während fünf Personen ein Dienstalter von fünf bis neun Jahren aufweisen. In der Gruppe mit zehn oder mehr Jahren Betriebszugehörigkeit befinden sich ebenfalls sechs Personen.

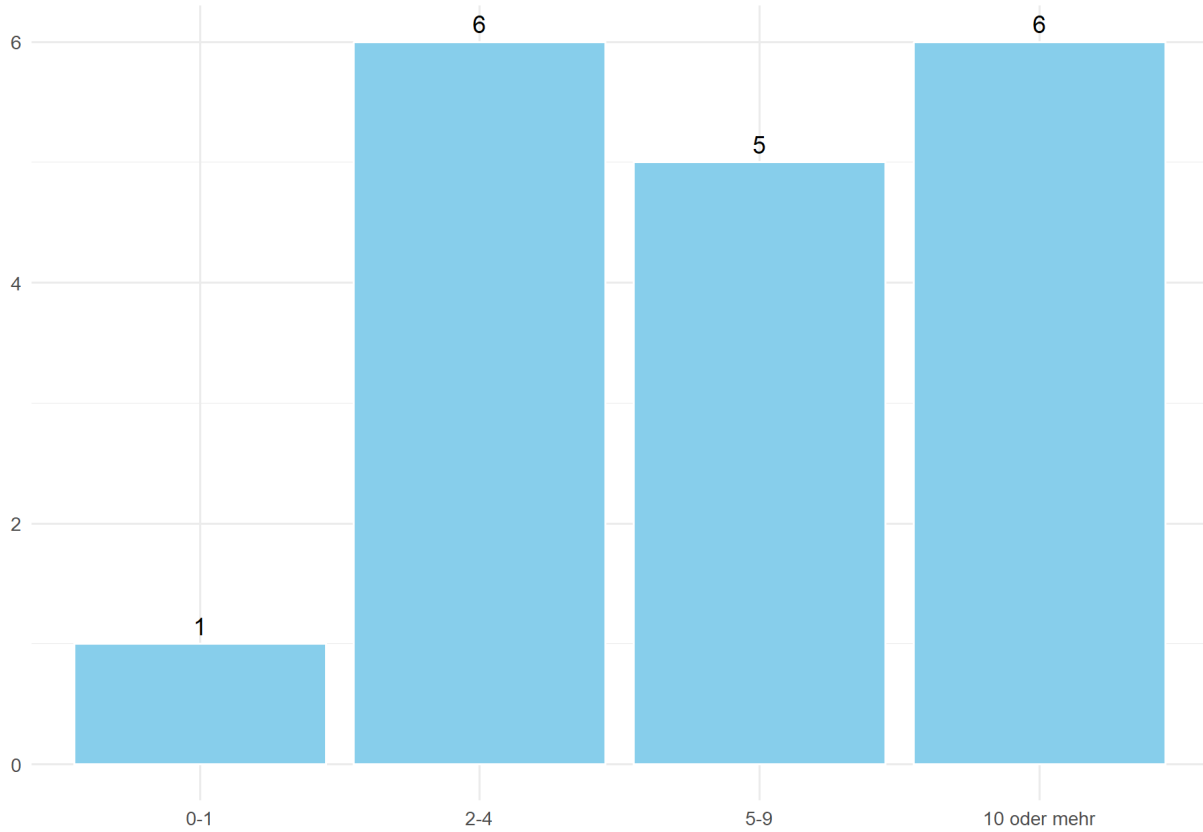


Abbildung 11: Dienstalter Umfrage Teilnehmer*Innen

Die Auswertung zur Erfahrung mit User Storys zeigt, dass drei Personen angeben, sehr erfahren zu sein. Die größte Gruppe bilden mit acht Personen jene, die sich als erfahren einschätzen. Fünf Personen ordnen sich in die Kategorie „Weder unerfahren noch erfahren“ ein. Zwei Personen bezeichnen sich als unerfahren im Umgang mit User Storys. Keine Person bezeichnet sich als sehr unerfahren im Umgang mit User Storys

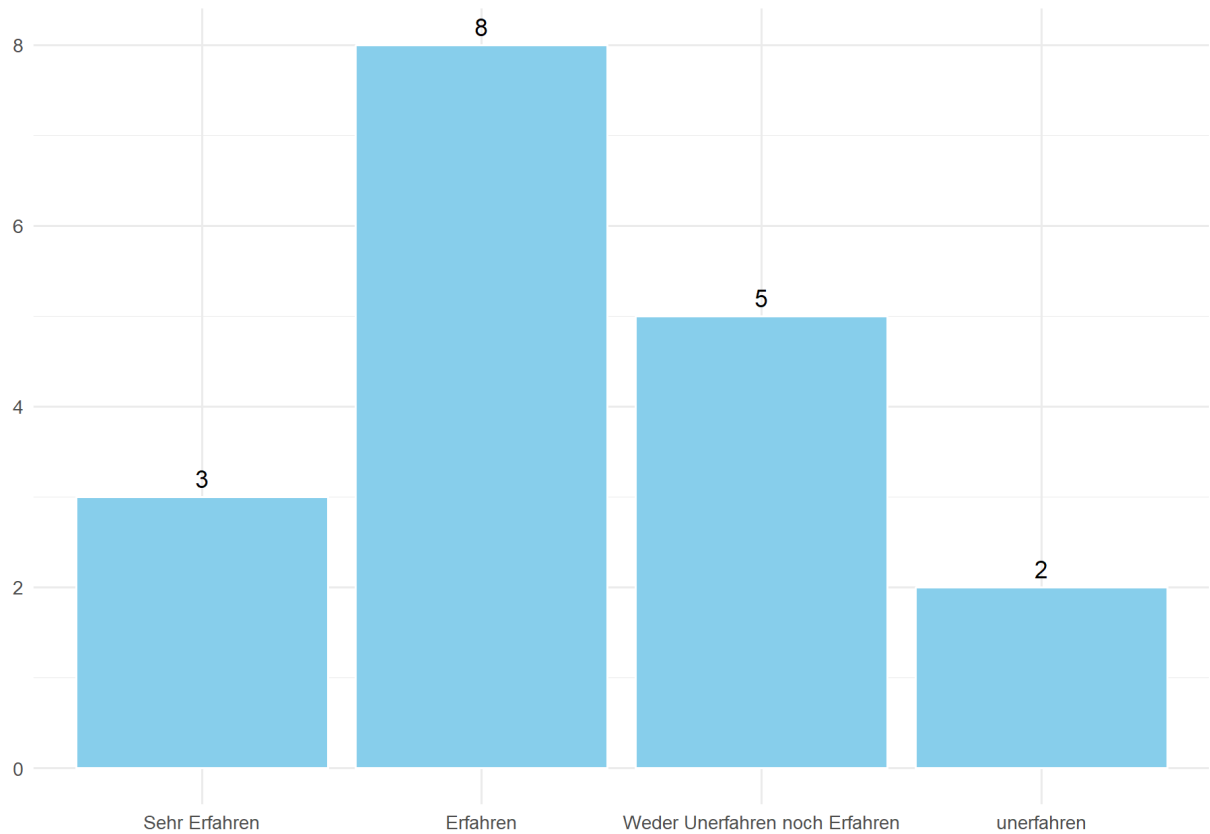


Abbildung 12: Erfahrung mit User Story Umfrage Auswertung

6.3.2 Beschreibung der Ergebnisse

Wie in Abschnitt 6.1.1 beschrieben, werden die Fragen des Fragebogens in vier Kategorien unterteilt: Readability (Lesbarkeit), Understandability (Verständlichkeit), Definability (Abgrenzbarkeit) und Technical Accuracy (Technische Richtigkeit). Jede Frage wird anhand einer fünfstufigen Likert-Skala bewertet. Die Antwortmöglichkeiten werden in numerische Werte von -2 bis $+2$ überführt:

-2 = stimme überhaupt nicht zu,

-1 = stimme nicht zu,

0 = neutral,

$+1$ = stimme zu,

$+2$ = stimme voll zu.

Für jede der vier Kategorien wird anschließend der arithmetische Mittelwert der zugehörigen Fragen berechnet. Dadurch ergibt sich ein Mittelwert pro Kategorie, der das durchschnittliche Urteil der Teilnehmenden widerspiegelt. Diese Mittelwerte werden anschließend zwischen den manuell erstellten und den mithilfe von künstlicher Intelligenz generierten Akzeptanzkriterien verglichen, um mögliche Unterschiede in den Dimensionen zu identifizieren.

Readability:

Aus der Auswertung geht hervor, dass die von der KI generierten Akzeptanzkriterien mit einem Mittelwert von 1,14 deutlich höher bewertet wurden als die von Menschen formulierten Kriterien, die einen Mittelwert von 0,35 erreichten. Dies deutet darauf hin, dass die Teilnehmenden die KI-generierten Kriterien im Durchschnitt als lesbarer empfanden. Die rote Linie bei null markiert den neutralen Bereich der Skala und verdeutlicht, dass beide Mittelwerte im positiven Bereich liegen, also insgesamt eine tendenziell gute Lesbarkeit attestiert wird.

Diese Ergebnisse legen nahe, dass KI-gestützte Formulierungen in Bezug auf die Lesbarkeit einen Vorteil gegenüber menschlich verfassten Akzeptanzkriterien bieten könnten.

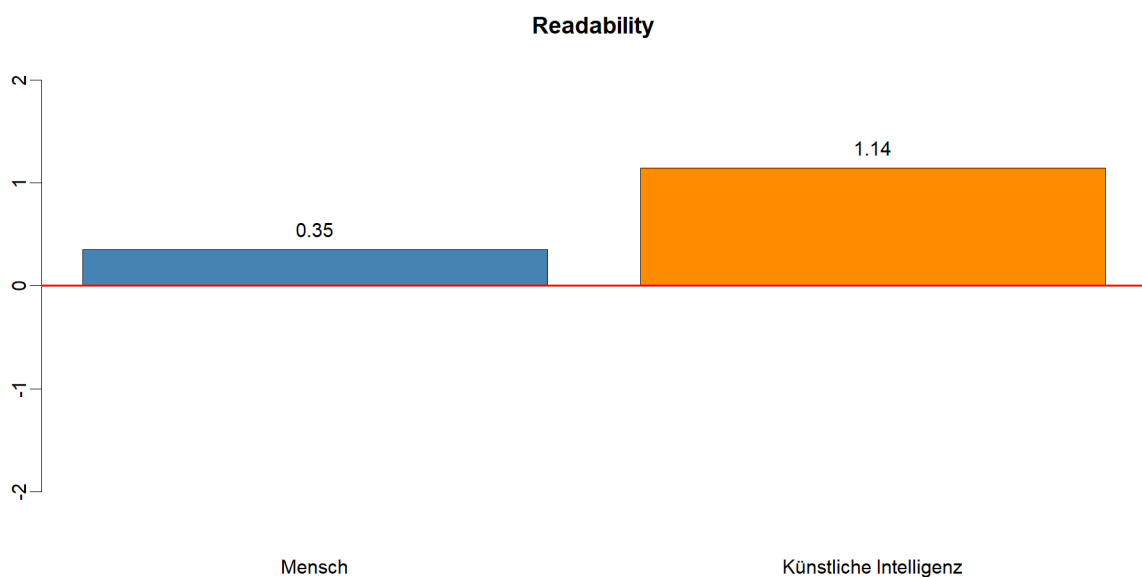


Abbildung 13: Auswertung Readability

Understandability:

Im Diagramm sind die Durchschnittswerte für die manuell formulierten Akzeptanzkriterien (Mensch) und die mithilfe künstlicher Intelligenz generierten Akzeptanzkriterien (Künstliche Intelligenz) dargestellt. Die von Menschen erstellten Kriterien erzielten dabei einen Mittelwert von 0,5 während die KI-generierten Kriterien mit 0,72 leicht höher bewertet wurden.

Beide Werte liegen im positiven Bereich oberhalb der neutralen Referenzlinie (rot bei 0), was darauf hindeutet, dass die Teilnehmenden sowohl die menschlich als auch die KI-erstellten Akzeptanzkriterien als insgesamt verständlich empfanden. Der etwas höhere Wert der KI-Version deutet jedoch darauf hin, dass die Teilnehmenden die Verständlichkeit der KI-generierten Kriterien tendenziell als etwas besser einschätzten.

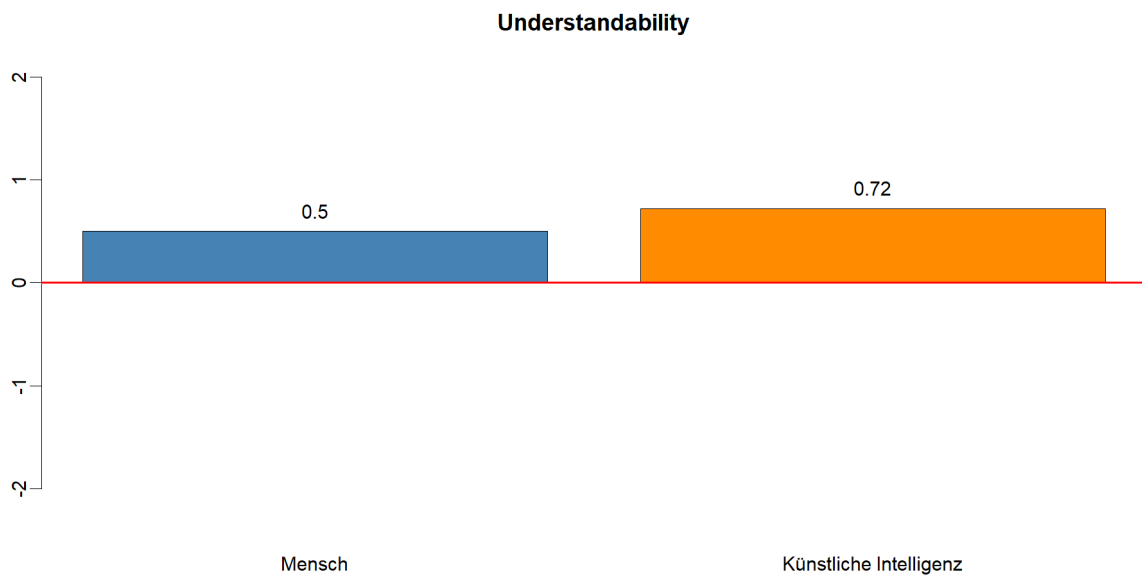


Abbildung 14: Auswertung Understandability

Definability:

In der Auswertung sind die Durchschnittswerte für die von Menschen formulierten Akzeptanzkriterien (Mensch) sowie für die mithilfe künstlicher Intelligenz generierten Akzeptanzkriterien (Künstliche Intelligenz) dargestellt. Die manuell erstellten Kriterien erzielten einen nahezu neutralen Mittelwert von 0,04, während die KI-generierten Kriterien mit einem deutlich höheren Mittelwert von 0,81 bewertet wurden.

Beide Werte liegen oberhalb der roten Referenzlinie bei null, die den neutralen Bereich der Skala kennzeichnet. Dies deutet darauf hin, dass die Teilnehmenden insgesamt eine eher positive Einschätzung zur Abgrenzbarkeit der Kriterien hatten. Der klare Unterschied zwischen den beiden Mittelwerten zeigt jedoch, dass die KI-generierten Akzeptanzkriterien als deutlich Abgrenzbarer wahrgenommen wurden als die von Menschen formulierten.

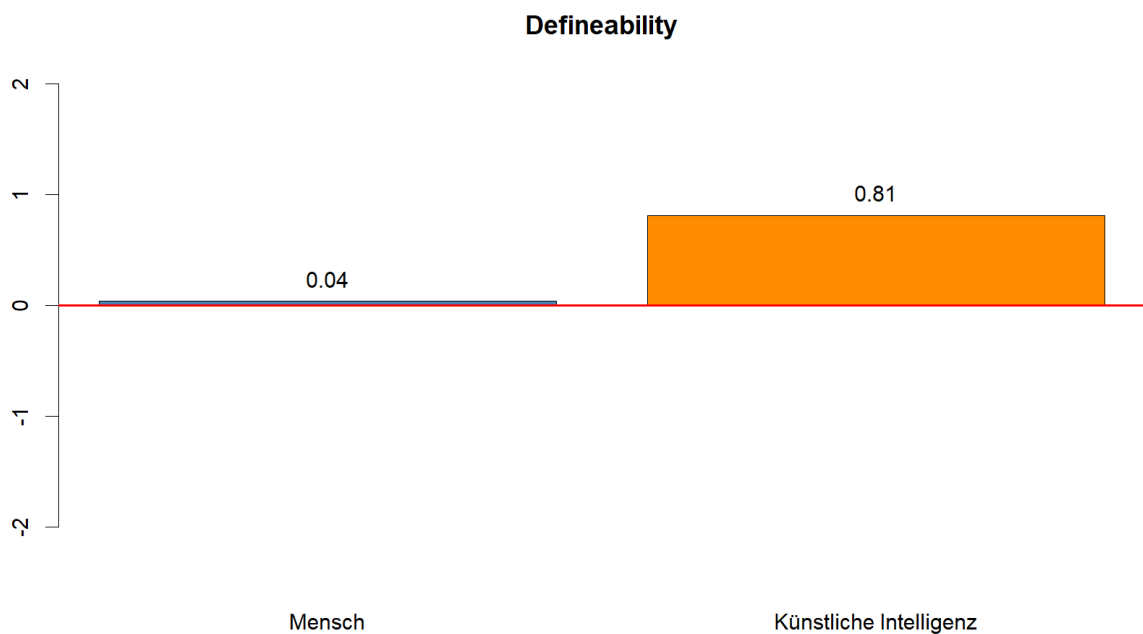


Abbildung 15: Auswertung Definability

Technical Accuracy:

In der Abbildung sind die durchschnittlichen Bewertungen für die manuell erstellten Akzeptanzkriterien (Mensch) sowie für die mithilfe künstlicher Intelligenz generierten Kriterien (Künstliche Intelligenz) dargestellt. Die menschlich formulierten Kriterien erzielten einen Mittelwert von 0,34, während die KI-generierten Kriterien mit 1,04 deutlich höher bewertet wurden.

Beide Mittelwerte liegen oberhalb der neutralen Referenzlinie bei null, was darauf hinweist, dass die Teilnehmenden insgesamt eine positive Einschätzung der technischen Richtigkeit beider Varianten hatten. Der deutliche Unterschied zwischen den beiden Balken verdeutlicht jedoch, dass die KI-generierten Akzeptanzkriterien als technisch präziser und korrekter wahrgenommen wurden als die von Menschen verfassten.

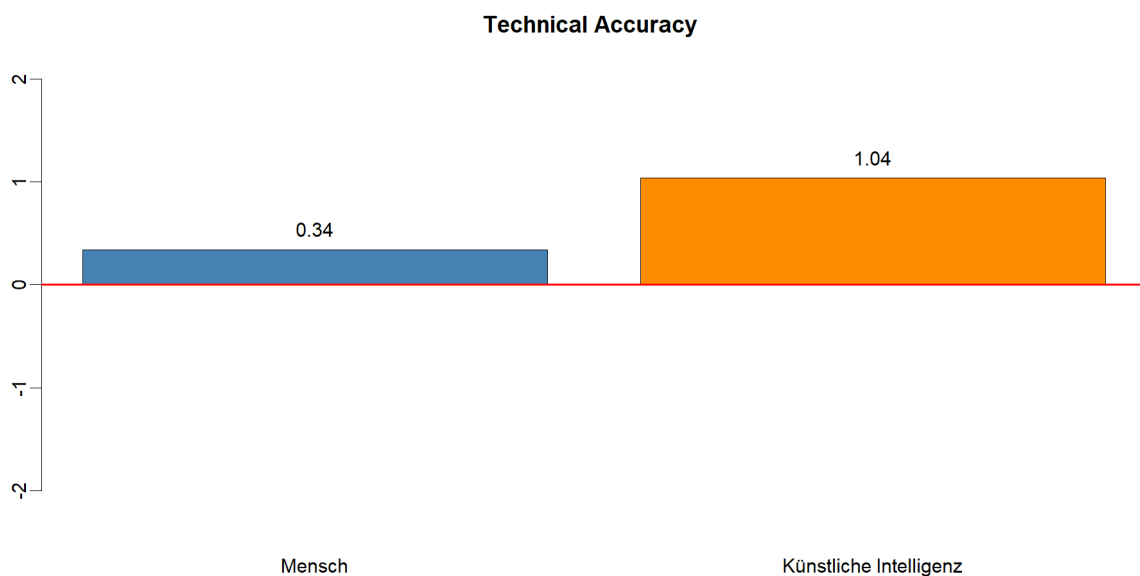


Abbildung 16: Auswertung Technical Accuracy

Gesamtbewertung:

Die von Menschen formulierten Akzeptanzkriterien erreichten einen durchschnittlichen Wert von 0,19, während die mithilfe künstlicher Intelligenz generierten Kriterien mit einem Mittelwert von 1,04 deutlich besser bewertet wurden.

Beide Werte liegen über der neutralen Referenzlinie bei null, was zeigt, dass die Teilnehmenden beide Varianten insgesamt positiv einschätzten. Der Unterschied in den Mittelwerten verdeutlicht jedoch, dass die KI-generierten Akzeptanzkriterien als qualitativ hochwertiger wahrgenommen wurden als die manuell erstellten. Dies bestätigt den in den vorherigen Kategorien beobachteten Trend, wonach die künstliche Intelligenz in mehreren Dimensionen eine konsistent bessere Bewertung erzielte.

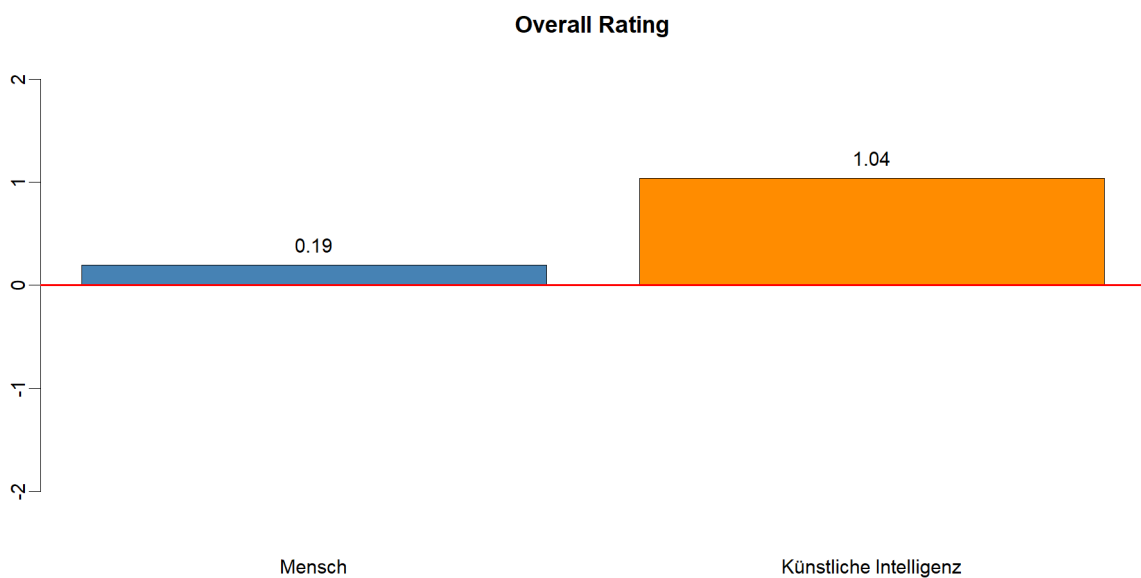


Abbildung 17: Overall Rating

Übernahme in den Backlog:

Der Graph in Abbildung 17 zeigt die Ergebnisse der zusätzlichen Frage, ob die jeweilige User Story in den *Product Backlog* aufgenommen werden würde. Diese Frage wurde binär beantwortet („Ja“ oder „Nein“) und der dargestellte Anteil gibt den Prozentsatz der Teilnehmenden an, die sich für eine Aufnahme ausgesprochen haben.

Für die von Menschen formulierten Akzeptanzkriterien gaben 54 % der Teilnehmenden an, dass sie die zugehörige User Story in den Backlog aufnehmen würden. Bei den mithilfe künstlicher Intelligenz generierten Akzeptanzkriterien lag dieser Anteil mit 80 % deutlich höher.

Dieses Ergebnis verdeutlicht, dass die KI-generierten User Stories häufiger als umsetzungswürdig eingestuft wurden. In Kombination mit den zuvor präsentierten Ergebnissen zu Lesbarkeit, Verständlichkeit, Abgrenzbarkeit und technischer Richtigkeit lässt sich daraus schließen, dass die Teilnehmenden die von der KI erstellten Kriterien nicht nur qualitativ höher bewerteten, sondern auch als praxisnäher und produktionsreifer wahrnahmen.

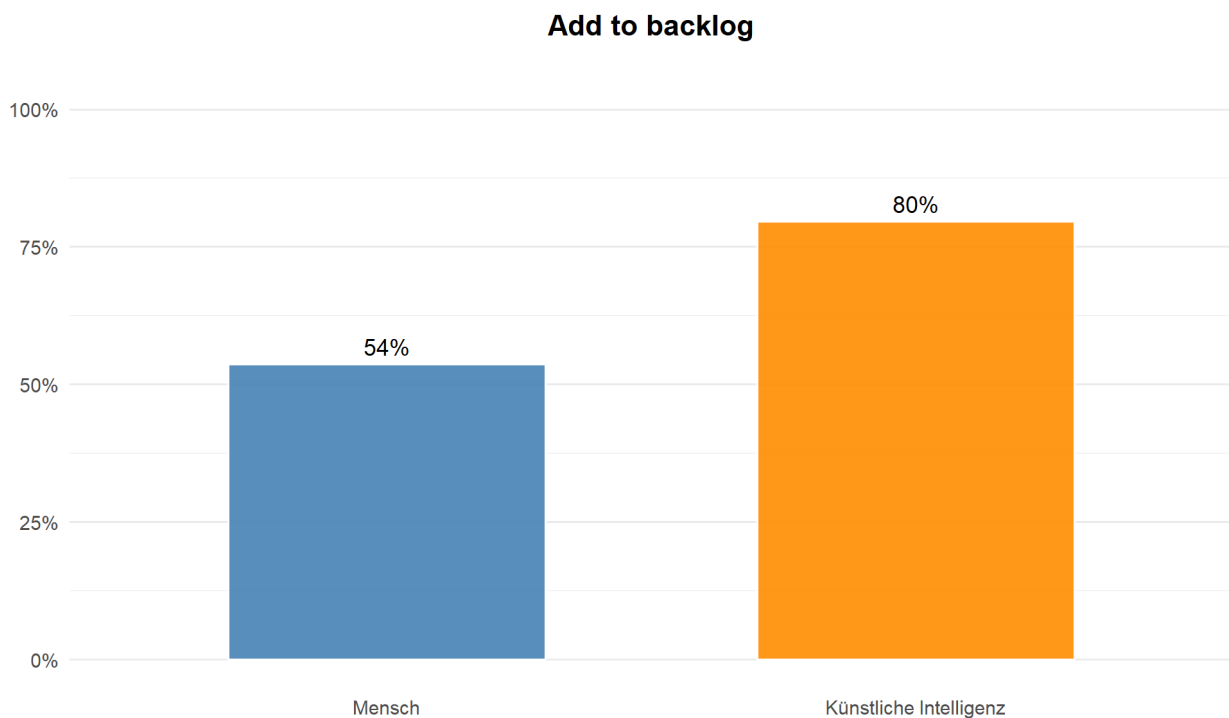


Abbildung 18: Backlog übernehmen

6.4 Maschinelle Auswertung

Für die maschinelle Auswertung der semantischen Übereinstimmung zwischen den von der KI generierten und den von Menschen formulierten Akzeptanzkriterien wurde, wie in 6.2.2 Maschinelle Validierung beschrieben, der BERTScore verwendet. Dieses Maß ermöglicht eine inhaltlich-semantische Bewertung, indem es auf Kontextrepräsentationen basiert und somit über eine rein oberflächliche Wortübereinstimmung hinausgeht.

Für die erste User Story ergibt sich ein Recall-Wert von 0,53, was bedeutet, dass rund 53 % der Inhalte der menschlichen Akzeptanzkriterien auch in den KI-generierten Kriterien enthalten sind. Der Precision-Wert liegt bei 0,74, sodass etwa drei Viertel der von der KI erzeugten Inhalte inhaltlich mit den menschlichen Akzeptanzkriterien übereinstimmen. Der daraus resultierende F1-Wert von 0,62 weist auf eine mittlere semantische Übereinstimmung hin. Die KI greift somit viele relevante Themen der menschlichen Kriterien auf, erzeugt jedoch auch zusätzliche oder leicht abweichende Inhalte.

Die zweite User Story zeigt mit einem Precision-Wert von 0,67, einem Recall von 0,65 und einem F1-Wert von 0,66 ein insgesamt konsistenteres Ergebnis. Etwa zwei Drittel der von der KI erzeugten Inhalte stimmen semantisch mit den menschlichen Akzeptanzkriterien überein, während zugleich etwa zwei Drittel der menschlichen Inhalte von der KI wiedergegeben werden. Dies deutet auf eine mittlere bis gute semantische Übereinstimmung zwischen den beiden Texten hin.

Für eine weitere Auswertung ergeben sich ähnliche Werte (Precision: 0,64, Recall: 0,65, F1: 0,65). Diese liegen nur geringfügig unter den Ergebnissen der zweiten User Story und bestätigen die Beobachtung, dass die KI einen Großteil der menschlichen Inhalte erfasst. Gleichzeitig ergänzt sie weiterhin eigene Formulierungen oder zusätzliche Themen, was auf eine gewisse kreative Varianz in der Textgenerierung hinweist.

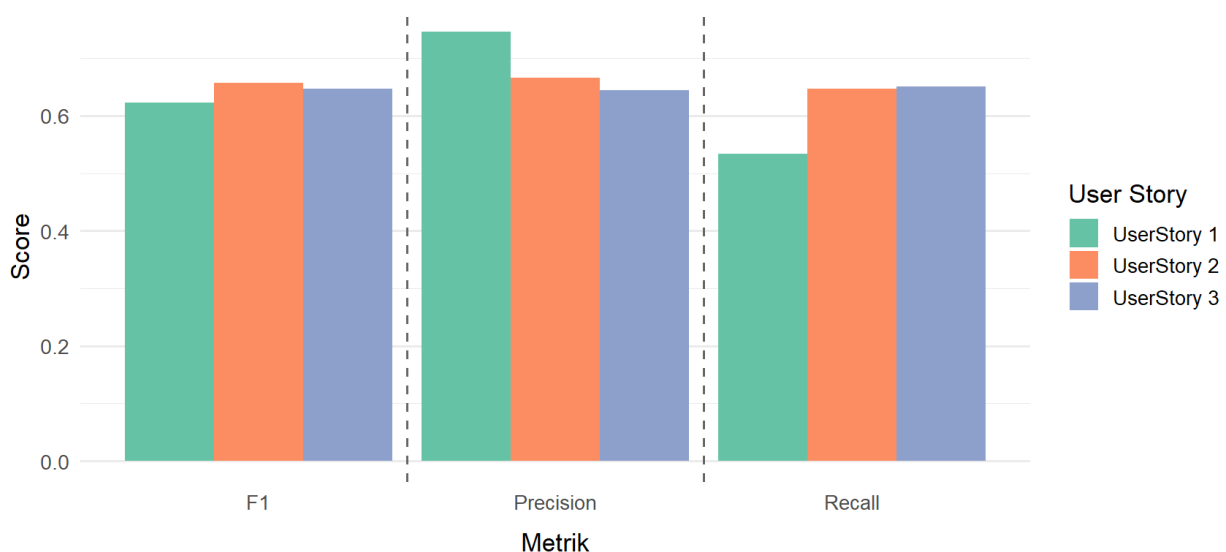


Abbildung 19: Auswertung BERTScore

Insgesamt zeigen die Ergebnisse, dass die KI inhaltlich relevante und thematisch passende Akzeptanzkriterien generiert, jedoch nicht ausschließlich die menschlichen Inhalte reproduziert. Vielmehr lässt sich eine Tendenz erkennen, dass die KI vorhandene Strukturen aufgreift und erweitert, wodurch eine moderate, aber konsistente semantische Übereinstimmung erzielt wird.

6.5 Interpretation Freitextkommentare

In den folgenden Unterkapiteln (6.5.1 bis 6.5.3) werden die Rückmeldungen zu jeweils einer menschlich und einer KI-erstellten User Story zusammengefasst und verglichen. Dabei geht es vor allem darum, wie beide Varianten wahrgenommen. In Bezug auf Verständlichkeit, Struktur, Umfang und die Qualität der Akzeptanzkriterien. Die Auswertung zeigt, dass KI-generierte User Stories oft strukturierter und ausführlicher wirken, dafür aber manchmal zu komplex oder widersprüchlich sind. Menschlich verfasste Stories erscheinen dagegen natürlicher und technisch besser eingeordnet, weisen jedoch häufiger formale Schwächen auf. Durch den Vergleich soll deutlich werden, welche Stärken und Schwächen beide Ansätze haben und wie sie sich gegenseitig ergänzen können, um künftig bessere User Stories zu erstellen.

6.5.1 User Story 1

In den Freitextkommentaren zur menschlich erstellten User Story wird vor allem auf sprachliche und strukturelle Aspekte eingegangen. Mehrere Teilnehmende bemängeln inkonsistente Formatierungen sowie die Sprachwahl und empfehlen eine englische Version. Inhaltlich wird die User Story als zu umfangreich beschrieben, wodurch sowohl die Entwicklung als auch das Testen erheblich aufwändiger werden. Die Kommentierenden schlagen vor, die Story in kleinere, klar abgegrenzte Items zu unterteilen – beispielsweise nach Komponenten wie Analyse, Code-Editor oder XAMBrowser. Auch die Grammatik und der Schreibstil werden als durchschnittlich und teilweise holprig empfunden. Einzelne Akzeptanzkriterien, etwa zum „PluginManager“, werden zudem als unklar oder unzureichend beschrieben.

Die Freitextkommentare zur KI-generierten User Story betonen dagegen die deutlich höhere Ausführlichkeit und den Mehrwert durch enthaltene Testhinweise. Gleichzeitig werden jedoch auch hier Verbesserungspotenziale benannt: So wird auf inhaltliche Widersprüche und unklare Bezüge zwischen Beschreibung und Akzeptanzkriterien hingewiesen, etwa bei Begriffen wie „Solution“ oder „Preview“. Auch hier wird die Story als zu umfangreich und aufwendig wahrgenommen. Obwohl die Akzeptanzkriterien als sehr detailliert und grundsätzlich testbar gelten, wird der Gesamtumfang als überfordernd beschrieben, weshalb eine Aufteilung in kleinere User Stories empfohlen wird. Zudem wird angeregt, die Kriterien prägnanter zu formulieren und die Formatierung zu vereinheitlichen.

Im direkten Vergleich zeigt sich, dass die KI-generierte User Story als strukturierter, vollständiger und stärker testorientiert wahrgenommen wird, während die menschlich verfasste Story sprachlich einfacher, aber weniger formal konsistent ist. Beide Varianten werden als zu umfangreich eingeschätzt, wobei die KI-Version tendenziell einen höheren Detaillierungsgrad und

damit auch einen größeren wahrgenommenen Aufwand aufweist. Insgesamt verdeutlichen die Kommentare, dass die KI zwar zu einer präziseren und umfassenderen Formulierung beitragen kann, dies jedoch mit dem Risiko einer inhaltlichen Überladung einhergeht.

6.5.2 User Story 2

In den Freitextkommentaren zur menschlich erstellten User Story wird vor allem auf den inhaltlichen Detaillierungsgrad eingegangen. Mehrere Anmerkungen heben hervor, dass die Beschreibung bereits technische Details enthält, die eher in ein technisches Dokument (z. B. ein Technical Design) gehören würden. Gleichzeitig wird positiv hervorgehoben, dass erkennbar ist, dass sich die erstellende Person bereits intensiv mit der Implementierung auseinandergesetzt hat. Der Umfang der Story wird als potenziell problematisch bewertet, insbesondere die vorgeschlagene Voranalyse, die sich mit Kompatibilitätsfragen und Aufwandsschätzungen beschäftigt. Diese sollte nach Ansicht der Befragten als eigenständiges Item behandelt werden. Der restliche Teil der User Story wird jedoch überwiegend als sinnvoll und klar abgegrenzt wahrgenommen. Besonders positiv wird hervorgehoben, dass die Erwartungshaltung – auch auf technischer Ebene – eindeutig ist. Einzelne Kommentare weisen darauf hin, dass die Akzeptanzkriterien möglicherweise zu technisch formuliert sind.

Die Freitextkommentare zur KI-generierten User Story betonen zunächst formale Auffälligkeiten, wie doppelte oder redundante Formulierungen im Fragebogen („2b: Syntax und Formatierung sind konsistent“) sowie inhaltliche Unstimmigkeiten in den Akzeptanzkriterien (z. B. fehlerhafte Logik bei „IsProxy=false“). Zudem wird bemängelt, dass einzelne Punkte zusammengefasst oder präzisiert werden könnten. Auch hier wird der Umfang der Story als kritisch bewertet und vorgeschlagen, die Voranalyse als separates Item auszulagern. Positiv hervorgehoben wird, dass die Akzeptanzkriterien sehr präzise und gut zur Ableitung von Testfällen geeignet sind, jedoch wird eine gewisse inhaltliche Unklarheit bei einzelnen Features („Alle Features des XAMirisStarter übernehmen“) bemängelt, da diese ohne weitere Beschreibung nicht ausreichend einschätzbar seien.

Im Vergleich zeigt sich, dass die menschlich verfasste User Story stärker auf praktische Umsetzbarkeit und technische Nachvollziehbarkeit fokussiert ist, während die KI-generierte Version zwar formal strukturierter, aber mit Redundanzen und kleineren logischen Fehlern behaftet ist. Beide Varianten werden hinsichtlich ihres Umfangs als zu groß empfunden, wobei sich in beiden Fällen die Empfehlung zur Aufteilung in kleinere Einheiten wiederfindet. Die KI-Version zeichnet sich durch präzise, testorientierte Akzeptanzkriterien, die menschliche Version durch klar verständliche technische Abgrenzungen aus. Insgesamt wird deutlich, dass die KI tendenziell zu einer formalen Vollständigkeit neigt, während menschlich verfasste Stories eine höhere inhaltliche Kohärenz und Kontextsensitivität aufweisen.

6.5.3 User Story 3

In den Freitextkommentaren zur menschlich erstellten User Story wird vor allem auf Unklarheiten und Umfang hingewiesen. Die Akzeptanzkriterien werden als verwirrend beschrieben, und eine

Neuformulierung oder Unterkategorisierung wird als hilfreich angesehen. Zudem zeigt sich, dass Abkürzungen oder Fachbegriffe (z. B. „QS“) nicht eindeutig verständlich sind. Einzelne Formulierungen wie „Direct RW ist für symbolic Zugriff nicht relevant“ werden hinterfragt, da der genaue Zweck und die technische Begründung unklar bleiben. Die Story wird insgesamt als zu umfangreich wahrgenommen; insbesondere ist nicht klar, welche Anpassungen an welchen Stellen vorgenommen werden sollen und welcher Aufwand damit verbunden ist. Positiv wird die beigefügte Hilfemail bewertet, da sie Kontext liefert, jedoch wird angemerkt, dass eine technische Vorabanalyse oder Demonstration notwendig wäre. Es wird vorgeschlagen, die Analyse und den Aufbau eines Testsystems (etwa mit 30.000 Datenpunkten) als separate Items zu behandeln, bevor eine Übernahme in die Anwendung erfolgen kann.

Die Freitextkommentare zur KI-generierten User Story zeichnen hingegen ein insgesamt positiveres Bild. Die Akzeptanzkriterien werden als verständlich beschrieben, wenngleich es an konkreten Zielwerten fehlt (etwa zur Leistungsfähigkeit). Einzelne technische Hinweise deuten darauf hin, dass die Story fachlich nachvollziehbar ist, jedoch Optimierungsmöglichkeiten im Design bestehen (z. B. Nutzung von Bulk-Read-Funktionen im Treiber). Kritisch angemerkt wird, dass die Story ohne genauere Kenntnisse der Implementierung nicht innerhalb eines Sprints umsetzbar erscheint und daher in mehrere kleinere User Stories aufgeteilt werden sollte. Insgesamt wird sie jedoch als greifbar und gut vorbereitet beschrieben. Es scheint bereits Vordenken und Struktur in die Erstellung eingeflossen zu sein.

Im Vergleich zeigt sich, dass die menschlich formulierte User Story stärker unter Unklarheit, Überkomplexität und fehlender Strukturierung leidet, während die KI-generierte Story als klarer, verständlicher und methodisch durchdachter wahrgenommen wird. Gleichzeitig weist die KI-Version formale und technische Lücken auf, etwa das Fehlen messbarer Zielgrößen. Beide Varianten verdeutlichen, dass eine präzise Strukturierung und schrittweise Aufteilung der Anforderungen essenziell sind, um die Umsetzbarkeit sicherzustellen. Insgesamt tendiert die KI dazu, klarere, aber abstraktere Beschreibungen zu erzeugen, während menschlich verfasste Stories mehr Kontext und technische Tiefe, jedoch weniger Struktur und Klarheit aufweisen.

6.5.4 Zusammenfassung

Über alle analysierten Beispiele hinweg zeigen sich konsistente Unterschiede zwischen den menschlich und den KI-generierten User Stories.

Die menschlich verfassten User Stories werden in den Kommentaren häufig als verständlich, aber unstrukturiert und teilweise überladen beschrieben. Sie enthalten oft technische Details, die über das für eine User Story übliche Maß hinausgehen, was einerseits auf praktische Erfahrung und Kontextwissen der Erstellenden hinweist, andererseits aber zu Unklarheiten, Redundanzen und hohem Testaufwand führt. Zudem werden sprachliche Inkonsistenzen, holprige Formulierungen und fehlende Strukturierung (z. B. bei Akzeptanzkriterien) mehrfach kritisiert. Die Stories wirken dadurch zwar nah an der Umsetzungspraxis, aber häufig zu komplex und schwer prüfbar.

Die KI-generierten User Stories hingegen werden überwiegend als strukturiert, ausführlich und testorientiert wahrgenommen. Ihre Akzeptanzkriterien sind meist präzise formuliert und gut zur

Ableitung von Testfällen geeignet. Gleichzeitig neigen sie jedoch dazu, zu detailliert oder überladen zu sein. Kommentierende bemängeln teilweise Widersprüche oder fehlende Zielgrößen, was auf eine gewisse inhaltliche Inkonsequenz hinweist. Die KI-Stories wirken dadurch formal korrekt, aber weniger kontextsensitiv und mitunter realitätsfern im Aufwand.

Zusammenfassend lässt sich festhalten, dass die menschlichen User Stories durch praktische Relevanz und Erfahrungstiefe, die KI-generierten hingegen durch Struktur, Vollständigkeit und Prüfbarkeit überzeugen. Während menschliche Stories tendenziell verständlicher, aber unpräziser sind, erscheinen KI-Stories präziser, aber kognitiv überlastend. Beide Varianten werden häufig als zu umfangreich empfunden, wobei die KI-Versionen insgesamt einen höheren Detaillierungsgrad und damit größeren wahrgenommenen Aufwand aufweisen.

6.6 Interpretation der Ergebnisse

Die Ergebnisse der Umfrage sowie der maschinellen Analyse zeigen ein konsistentes Gesamtbild: Die von der KI generierten Akzeptanzkriterien wurden in nahezu allen untersuchten Dimensionen besser bewertet als die manuell erstellten. Dieses Ergebnis deutet darauf hin, dass die KI in der Lage ist, qualitativ hochwertige und praxisnahe Formulierungen zu erzeugen, die von den Teilnehmenden als klarer und strukturierter wahrgenommen werden.

In der Kategorie Lesbarkeit erreichten die KI-generierten Akzeptanzkriterien einen deutlich höheren Mittelwert (1,14) als die menschlich formulierten (0,35). Dies zeigt, dass die KI-Texte sprachlich flüssiger, konsistenter und damit angenehmer zu lesen sind. Auch in der Kategorie Verständlichkeit wurde ein leichtes, aber konsistentes Plus für die KI festgestellt (0,72 gegenüber 0,5), was darauf hindeutet, dass die Teilnehmenden die durch KI erzeugten Kriterien als klarer und nachvollziehbarer einstufen.

Besonders deutlich fällt der Unterschied in den Dimensionen Abgrenzbarkeit und technische Richtigkeit aus. Hier wurden die KI-generierten Akzeptanzkriterien mit Mittelwerten von 0,81 bzw. 1,04 deutlich höher bewertet als die manuell erstellten (0,04 bzw. 0,34). Diese Ergebnisse legen nahe, dass die KI in der Lage ist, präziser und strukturierter zu formulieren, was insbesondere für die Weiterverarbeitung in Entwicklungsprozessen von Vorteil ist. Die höhere Bewertung in der technischen Richtigkeit deutet zudem darauf hin, dass die KI sprachliche und logische Konsistenz gut wahren kann, wodurch die erzeugten Kriterien insgesamt professioneller und vollständiger wirken.

Auch in der Gesamtbewertung bestätigt sich dieses Muster: Während die manuell formulierten Akzeptanzkriterien einen durchschnittlichen Wert von 0,19 erreichten, erzielten die KI-generierten Kriterien mit 1,04 eine deutlich höhere Bewertung. Beide Werte liegen oberhalb der neutralen Schwelle, was zeigt, dass beide Varianten grundsätzlich als brauchbar eingeschätzt werden. Der Unterschied verdeutlicht jedoch, dass die KI-generierten Texte als qualitativ hochwertiger, einheitlicher und potenziell produktionsreifer wahrgenommen wurden. Diese Einschätzung spiegelt sich auch in der zusätzlichen Frage zur Übernahme in den Product Backlog wider: 80 % der Teilnehmenden würden eine User Story mit KI-generierten Akzeptanzkriterien aufnehmen,

gegenüber 54 % bei den menschlich formulierten. Damit werden die KI-Ergebnisse nicht nur als sprachlich überzeugend, sondern auch als praktisch umsetzungswürdig bewertet

Die maschinelle Auswertung mithilfe des BERTScore ergänzt diese Ergebnisse um eine inhaltliche Perspektive. Die F1-Werte zwischen 0,62 und 0,66 zeigen eine mittlere semantische Übereinstimmung zwischen den KI-generierten und den menschlich formulierten Akzeptanzkriterien. Dies bedeutet, dass die KI einen großen Teil der thematischen Inhalte der menschlichen Formulierungen aufgreift. Gleichzeitig wird jedoch deutlich, dass nicht alle relevanten Aspekte aus den menschlichen Akzeptanzkriterien in den KI-generierten Texten wiederzufinden sind. Der moderate Recall-Wert weist darauf hin, dass teilweise wichtige inhaltliche Punkte ausgelassen werden. Diese Lücken können im praktischen Einsatz problematisch sein, da sie dazu führen könnten, dass bestimmte fachliche Anforderungen oder Randbedingungen unzureichend berücksichtigt werden. Die KI zeigt somit zwar ein gutes Verständnis der Struktur und Sprache von Akzeptanzkriterien, erfasst jedoch nicht immer die gesamte inhaltliche Tiefe menschlicher Formulierungen.

Insgesamt lässt sich festhalten, dass die KI-generierten Akzeptanzkriterien von den Teilnehmenden durchgehend positiver bewertet wurden als die menschlich formulierten, insbesondere hinsichtlich Lesbarkeit, Abgrenzbarkeit und technischer Richtigkeit. Die Ergebnisse des BERTScore verdeutlichen jedoch, dass diese höhere sprachliche Qualität nicht zwingend mit vollständiger inhaltlicher Deckungsgleichheit einhergeht. Für die Praxis bedeutet dies, dass KI-gestützte Generierung von Akzeptanzkriterien ein hohes Potenzial zur Qualitätssteigerung und Effizienzverbesserung bietet, jedoch weiterhin eine menschliche Validierung und fachliche Kontrolle erforderlich bleibt, um den vollständigen Informationsgehalt sicherzustellen. Die KI kann somit, als unterstützendes Werkzeug betrachtet werden, das die Erstellung von Akzeptanzkriterien beschleunigt und vereinheitlicht, jedoch nicht die fachliche Verantwortung der Entwickler*Innen ersetzt.

6.7 Eingrenzung der Erkenntnisse

Die Ergebnisse dieser Untersuchung unterliegen mehreren Einschränkungen, die bei der Interpretation berücksichtigt werden müssen. Eine geografische Einschränkung ergibt sich aus dem Kontext der Datenerhebung: Sämtliche Teilnehmer*Innen stammen aus einem österreichischen Unternehmen innerhalb des deutschsprachigen DACH-Raums. Damit sind sowohl Arbeitskultur als auch Kommunikationsgewohnheiten, insbesondere im Rahmen agiler Entwicklungsprozesse, regional geprägt. Praktiken in Meetings, die Auslegung von Rollen wie Product Owner oder Entwickler sowie die Einschätzung von Lesbarkeit und Verständlichkeit können in anderen Ländern und Unternehmen abweichen. Die Generalisierbarkeit der Ergebnisse auf internationale oder kulturell unterschiedliche Teams ist daher nur eingeschränkt gegeben.

Auch sprachliche Aspekte stellen eine relevante Limitation dar. Sowohl die User Stories als auch die Akzeptanzkriterien und Bewertungen wurden vollständig in deutscher Sprache durchgeführt. Sprachmodelle verhalten sich jedoch in verschiedenen Sprachen unterschiedlich, insbesondere

hinsichtlich Ausdrucksstärke, Syntax und Wortwahl. Die festgestellten Vorteile der KI-generierten Akzeptanzkriterien bei Lesbarkeit und Abgrenzbarkeit könnten somit sprachspezifisch sein und sich in englischen oder mehrsprachigen Projekten anders darstellen. Für eine vollständige Übertragbarkeit wäre eine zusätzliche Untersuchung in englischer Sprache erforderlich.

Darüber hinaus ist die Branchenspezifität zu berücksichtigen. Die erhobenen Daten stammen aus einem industriellen B2B-Softwarekontext (Industrial Automation), in dem technische Präzision und Integrationsaspekte eine zentrale Rolle spielen. In stärker nutzungsorientierten Domänen wie E-Commerce, MedTech oder Consumer-Software, in denen Anforderungen häufiger auf Benutzerinteraktionen und Nutzererlebnisse ausgerichtet sind, könnten andere Qualitätsdimensionen im Vordergrund stehen. Die Ergebnisse sind daher primär für technisch geprägte Softwareprojekte mit hohem Anteil an Systemintegration und Schnittstellenentwicklung relevant.

Zusammenfassend lässt sich festhalten, dass die Untersuchung wertvolle Einblicke in die Qualität KI-generierter Akzeptanzkriterien im industriellen Kontext liefert, ihre Übertragbarkeit auf andere Sprachen, Regionen und Branchen jedoch nur eingeschränkt gegeben ist. Zukünftige Studien sollten diese Faktoren gezielt variieren, um die externe Validität der Ergebnisse weiter zu stärken.

7 ZUSAMMENFASSUNG

Dieses Kapitel fasst die wichtigsten Ergebnisse der Arbeit zusammen, reflektiert die identifizierten Limitationen und Herausforderungen und gibt einen Ausblick auf zukünftige Forschungs- und Entwicklungsmöglichkeiten. Ziel ist es, die zentralen Erkenntnisse einzuordnen, die Aussagekraft der Resultate zu bewerten und potenzielle Weiterführungen des vorgestellten Ansatzes aufzuzeigen.

7.1 Zusammenfassung

In dieser Arbeit wird ein Ansatz vorgestellt, der Methoden der Künstlichen Intelligenz (KI) zur automatisierten Erstellung von Akzeptanzkriterien für User Storys nutzt. Dazu wurde ein Prototyp entwickelt, der die Beschreibung einer User Story sowie zugehörige Dokumentationsinhalte in Vektoren überführt und mittels Clustering inhaltlich relevante Abschnitte identifiziert. Die gefundenen Dokumentationssegmente werden anschließend gemeinsam mit der User Story in einem Prompt an das Sprachmodell ChatGPT-5-mini übergeben, das darauf basierend geeignete Akzeptanzkriterien generiert.

Die Literaturrecherche umfasste verschiedene Aspekte des Requirements-Engineering-Prozesses und deren Verbindung zu KI-basierten Methoden. Diese Analyse bildete die Grundlage für die Auswahl der eingesetzten Technologien. Zudem wurden potenzielle Herausforderungen bei der Anwendung von KI im Requirements Engineering diskutiert. Besonders Themen wie Halluzinationen, Urheberrecht, ethische Fragestellungen und Compliance-Anforderungen erweisen sich als zentrale Aspekte, die zwingend eine menschliche Kontrolle und Bewertung erfordern.

Die Evaluation des entwickelten Prototyps zeigte, dass KI-basierte Ansätze die Erstellung von Akzeptanzkriterien in mehrfacher Hinsicht unterstützen und verbessern können. Nutzerbefragungen sowie maschinelle Analysen mittels BERTScore belegten eine Steigerung der Verständlichkeit, Einheitlichkeit und Prüfbarkeit der generierten Kriterien. Somit kann KI nicht nur als Werkzeug zur Qualitätssicherung, sondern auch als methodische Unterstützung für Requirements Engineers dienen. Gleichzeitig wurde deutlich, dass die fachliche Bewertung und Kontextinterpretation durch menschliche Experten unverzichtbar bleibt, um die technische Relevanz und organisatorische Passung der Ergebnisse sicherzustellen.

Zusammenfassend belegt die Arbeit, dass der Einsatz von Künstlicher Intelligenz, insbesondere in Kombination mit domänenspezifischen Dokumentationen, einen Beitrag zur Steigerung der Anforderungsqualität leisten kann. KI-Systeme ermöglichen die Automatisierung von Routineaufgaben im Requirements Engineering, erhöhen die Nachvollziehbarkeit von Anforderungen und fördern die objektivere sowie reproduzierbarere Erstellung von Akzeptanzkriterien. Damit entsteht eine neue Schnittstelle zwischen datengetriebener Analyse und menschlicher Expertise, die langfristig zu einer höheren Reife und Effizienz in der Softwareentwicklung führen kann.

7.2 Kritische Reflexion

Obwohl die vorliegende Arbeit vielversprechende Ergebnisse liefert, ist eine kritische Reflexion der Limitationen und Herausforderungen unerlässlich.

Eine wesentliche Limitation liegt in der Qualität und Verfügbarkeit der Produktdaten. Die Leistungsfähigkeit des KI-Systems hängt direkt von der Vollständigkeit, Aktualität und Repräsentativität der zugrunde liegenden Daten ab. Unvollständige oder veraltete Dokumentationen können zu suboptimalen oder sogar fehlerhaften Akzeptanzkriterien führen.

Eine Limitation der vorliegenden Arbeit besteht darin, dass eine Frage im Fragebogen versehentlich doppelt gestellt wurde. Dieses Versehen könnte potenziell zu einer Verzerrung der Ergebnisse geführt haben. Das Problem wurde jedoch gemindert, indem die redundante Frage aus der Auswertung entfernt wurde. Da davon auszugehen ist, dass die Teilnehmenden den Fragebogen von oben nach unten beantwortet haben, ist anzunehmen, dass sich der Einfluss dieser Doppelung auf die Ergebnisse in Grenzen hält.

Eine weitere Limitation ergibt sich daraus, dass im Fragebogen die Konsistenz der Formatierung bewertet werden sollte. Die ursprüngliche Formatierung der User Stories aus dem Portal Azure DevOps konnte jedoch nicht vollständig in das Umfragetool Google Forms übertragen werden. Zudem traten technische Probleme auf, die je nach Endgerät zu Darstellungsunterschieden bei einzelnen Fragen führten. Die Auswirkung dieser Limitation dürfte jedoch gering sein, da es sich lediglich um einen Teilaspekt einer Frage handelt und insgesamt drei Fragen zur Lesbarkeit pro User Story sowie drei User Stories bewertet wurden.

In der Rückschau wäre es sinnvoll gewesen, anstelle des selbst entwickelten Bewertungsmodells RUST (Readability, Understandability, Specifiability und Technical Accuracy) das etablierte INVEST-Modell zu verwenden. Dadurch hätte eine bessere Vergleichbarkeit mit bestehenden Arbeiten gewährleistet werden können, da INVEST in der Forschung und Praxis zur Bewertung von User Stories bereits weit verbreitet ist. Der Einsatz dieses Modells hätte somit eine stärkere Einbettung der Ergebnisse in den wissenschaftlichen Diskurs ermöglicht und die Nachvollziehbarkeit für andere Forschende erhöht.

7.3 Ausblick

Basierend auf den Erkenntnissen dieser Masterarbeit ergeben sich vielfältige Ansätze für zukünftige Forschungsarbeiten und Weiterentwicklungen:

Erweiterung der Produktdatenquellen: Zukünftige Forschung könnte die Integration weiterer Produktdatenquellen wie Nutzerinteraktionen in der Software (z. B. Klickpfade, Nutzungsstatistiken), Fehlerberichte aus dem Bug-Tracking-System oder sogar Videos von Usability-Tests untersuchen. Dies würde die Wissensbasis weiter anreichern und ein noch umfassenderes Verständnis der Nutzerbedürfnisse ermöglichen.

Verbesserung der RAG-Pipeline: Die Optimierung der Retrieval-Komponente, beispielsweise durch fortgeschrittenere Ranking-Algorithmen oder die Integration von Graphen-Datenbanken

zur Abbildung komplexer Beziehungen in den Produktdaten, könnte die Relevanz und Qualität der abgerufenen Informationen weiter steigern.

Integration in CI/CD-Pipelines: Die nahtlose Integration des Artefakts in bestehende Continuous Integration/Continuous Delivery (CI/CD) -Pipelines könnte die Automatisierung des Testprozesses weiter vorantreiben, indem Akzeptanzkriterien direkt in Testfälle überführt werden.

ABKÜRZUNGSVERZEICHNIS

LLM	Large Language Model
NLP	Natural Language Processing
RLHF	Reinforced Learning from Human Feedback
ML	Machine Learning
BLEU	Bilingual evaluation understudy
NGO	Non-Governmental Organization
TFIDF	Term Frequency-Inverse Document Frequency

ABBILDUNGSVERZEICHNIS

Abbildung 1: Literatur Recherche Prozess.....	4
Abbildung 2: Clusterung Literaturrecherche.....	5
Abbildung 3: Vorwärts- und Rückwärtssuche 1.....	8
Abbildung 4: Vorwärts und Rückwärtssuche 2.....	8
Abbildung 5: Evaluierung der Literatur	9
Abbildung 6: Art der Literatur.....	10
Abbildung 7: Altersverteilung Literatur	11
Abbildung 8: PRISMA Flowchart	12
Abbildung 9: Prozess der Datenverarbeitung.....	35
Abbildung 10: Rolle der Umfrage Teilnehmer*Innen.....	50
Abbildung 11: Dienstalter Umfrage Teilnehmer*Innen	51
Abbildung 12: Erfahrung mit User Story Umfrage Auswertung.....	52
Abbildung 13: Auswertung Readability.....	54
Abbildung 14: Auswertung Understandability.....	55
Abbildung 15: Auswertung Definability	56
Abbildung 16: Auswertung Technical Accuracy	57
Abbildung 17: Overall Rating.....	58
Abbildung 18: Backlog Übernehmen.....	59
Abbildung 19: Auswertung BERTScore	60

TABELLENVERZEICHNIS

Tabelle 1: Taxonomie nach Cooper	5
Tabelle 2: DSR-Anforderungen	34
Tabelle 3: Userstory Akzeptanzkriterien 1.....	42
Tabelle 4: Userstory Akzeptanzkriterien 2.....	44
Tabelle 5: Userstory Akzeptanzkriterien 3.....	46
Tabelle 6: Fragenkatalog.....	48

LITERATURVERZEICHNIS

- Abbas, Muhammad; Bashir, Sarmad; Saadatmand, Mehrdad; Enoiu, Eduard Paul; Sundmark, Daniel (2025): Requirements similarity and retrieval. In: Handbook on Natural Language Processing for Requirements Engineering: Springer, S. 61–88.
- Ahmed, A.; Nasr, M.; Abd Elhamid, L.: Multi-Modal AI for Software Requirements Validation. AI Chatbot Market Share Austria (2025). Online verfügbar unter <https://gs.statcounter.com/ai-chatbot-market-share/all/austria>.
- Akbarighatar, Pouria (2025): Operationalizing responsible AI principles through responsible AI capabilities. In: *AI Ethics* 5 (DOI: 10.1007/s43681-024-00524-4), S. 1801. DOI: 10.1007/s43681-024-00524-4.
- Alhoshan, Waad; Batista-Navarro, Riza; Zhao, Liping (Hg.) (2019): Using Frame Embeddings to Identify Semantically Related Software Requirements.
- Arvidsson, Simon; Axell, Johan (2023): Prompt engineering guidelines for LLMs in Requirements Engineering.
- Berger, Philipp (2023): Applying software model checking: experiences and advancements: Dissertation, RWTH Aachen University, 2023.
- Bergsmann, Johannes (2018): Requirements Engineering für die agile Softwareentwicklung : Methoden, Techniken und Strategien. Unter Mitarbeit von Unterauer, Markus, 1979-. 2., überarbeitete und aktualisierte Auflage: Heidelberg : dpunkt.verlag; [Ann Arbor] : ProQuest eBook Central. Online verfügbar unter <https://permalink.obvsg.at/fwg/AC15292877>.
- Cheligeer, Cheligeer; Huang, Jingwei; Wu, Guosong; Bhuiyan, Nadia; Xu, Yuan; Zeng, Yong (2022): Machine learning in requirements elicitation: A literature review. In: *AI EDAM* 36, e32.
- Cheng, Haowei; Husen, Jati H.; Lu, Yijun; Racharak, Teeradaj; Yoshioka, Nobukazu; Ubayashi, Naoyasu; Washizaki, Hironori (2024): Generative AI for Requirements Engineering: A Systematic Literature Review. In: *arXiv preprint arXiv:2409.06741*.
- Cheng, Haowei; Husen, Jati H.; Lu, Yijun; Racharak, Teeradaj; Yoshioka, Nobukazu; Ubayashi, Naoyasu; Washizaki, Hironori (2025): Generative ai for requirements engineering: A systematic literature review. In: *Software: Practice and Experience*.
- Chris Rupp, Die SOPHISTen (2014): Requirements-Engineering und -Management. Aus der Praxis von klassisch bis agil. Extra: Wissenstest a. d. ILIAS-Lernplattform u. E-Book inside. In Zus.-Arb. m. d. SOPHISTen. 6. Aufl.
- Da Cruz, António M. Rosado; Cruz, Estrela Ferreira (2025): Artificial Intelligence Techniques for Requirements Engineering: A Comprehensive Literature Review.

- Dias Canedo, Edna; Cordeiro Mendes, Bruno (2020): Software requirements classification using machine learning algorithms. In: *Entropy* 22 (9), S. 1057.
- Douze, Matthijs; Guzhva, Alexandr; Deng, Chengqi; Johnson, Jeff; Szilvasy, Gergely; Mazaré, Pierre-Emmanuel et al. (2024): The faiss library. In: *arXiv preprint arXiv:2401.08281*.
- Elallaoui, Meryem; Nafil, Khalid; Touahni, Raja (2018): Automatic transformation of user stories into UML use case diagrams using NLP techniques. In: *Procedia computer science* 130, S. 42–49.
- Gao, Yunfan; Xiong, Yun; Gao, Xinyu; Jia, Kangxiang; Pan, Jinliu; Bi, Yuxi et al. (2023): Retrieval-augmented generation for large language models: A survey. In: *arXiv preprint arXiv:2312.10997* 2 (1).
- Glinz, Martin; van Loenhoud, Hans; Staal, Stefan; Bühne, Stan (2020): Handbook for the CPRE Foundation Level according to the IREB Standard. In: *International Requirements Engineering Board*.
- Henriksson, Aron; Zdravkovic, Jelena: A data-driven framework for automated requirements elicitation from heterogeneous digital sources. In: *IFIP Working Conference on The Practice of Enterprise Modeling*: Springer, S. 351–365.
- Huang, Lei; Yu, Weijiang; Ma, Weitao; Zhong, Weihong; Feng, Zhangyin; Wang, Haotian et al. (2025): A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. In: *ACM Transactions on Information Systems* 43 (2), S. 1–55.
- Ihalainen, Jani (2018): Computer creativity: artificial intelligence and copyright. In: *Journal of Intellectual Property Law & Practice* 13 (9), S. 724–728.
- ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes (2015). In: *ISO/IEC/IEEE 15288 First edition 2015-05-15*.
- Jiao, Junfeng; Afroogh, Saleh; Xu, Yiming; Phillips, Connor (2024): Navigating llm ethics: Advancements, challenges, and future directions. In: *arXiv preprint arXiv:2406.18841*.
- Käde, Lisa. (2021): *Kreative Maschinen und Urheberrecht : Die Machine Learning-Werkschöpfungskette vom Training über Modellschutz bis zu Computational Creativity*. 1st ed.: Baden-Baden : Nomos Verlagsgesellschaft (Band 2).
- Knuplesch, Stefan (2024): *Mit künstlicher Intelligenz auf dem Weg zu effektiven und effizienten Softwareangeboten*: Graz. Online verfügbar unter <https://permalink.obvsg.at/fwg/AC17302710>.
- Kreutzer, Ralf T.; Sirrenberg, Marie (2019): Grundlagen und Treiber der Künstlichen Intelligenz. In: *Künstliche Intelligenz verstehen: Grundlagen–Use-Cases–unternehmenseigene KI-Journey*, S. 73–105.
- Lami, Bareq; Mohd. Hussein, Safinaz; Rajamanickam, Ramalinggam; Emmanuel, Grace Kaka (2024): The role of artificial intelligence (AI) in shaping data privacy. In: *International*

- journal of law and management* (DOI: 10.1108/IJLMA-07-2024-0242). DOI: 10.1108/IJLMA-07-2024-0242.
- Liu, Hanyue; García, Marina Bueno; Korkakakis, Nikolaos: Exploring Multi-Label Data Augmentation for LLM Fine-Tuning and Inference in Requirements Engineering: A Study with Domain Expert Evaluation. In: 2024 International Conference on Machine Learning and Applications (ICMLA): IEEE, S. 432–439.
- Liu, Pengfei; Yuan, Weizhe; Fu, Jinlan; Jiang, Zhengbao; Hayashi, Hiroaki; Neubig, Graham (2023): Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. In: *ACM computing surveys* 55 (9). DOI: 10.1145/3560815.
- M. Krishna; B. Gaur; A. Verma; P. Jalote (2024): Using LLMs in Software Requirements Specifications: An Empirical Evaluation. In: 2024 IEEE 32nd International Requirements Engineering Conference (RE). 2024 IEEE 32nd International Requirements Engineering Conference (RE), S. 475–483.
- Mehrabi, Ninareh; Morstatter, Fred; Saxena, Nripsuta; Lerman, Kristina; Galstyan, Aram (2021): A survey on bias and fairness in machine learning. In: *ACM computing surveys* 54 (6), S. 1–35.
- Norheim, Johannes J.; Rebentisch, Eric; Xiao, Dekai; Draeger, Lorenz; Kerbrat, Alain; Weck, Olivier L. de (2024): Challenges in applying large language models to requirements engineering tasks. In: *Design Science* 10, e16. DOI: 10.1017/dsj.2024.8.
- OpenAI (2025): GPT-5 mini. Unter Mitarbeit von OpenAI. Hg. v. OpenAI. Online verfügbar unter <https://platform.openai.com/docs/models/gpt-5-mini>.
- Pohl, Klaus; Rupp, Chris (2021): Basiswissen requirements engineering: Aus-und Weiterbildung nach IREB-Standard zum certified professional for requirements engineering foundation level: dpunkt. verlag.
- R. Izhar; K. Cosh; L. Ramingwon; S. Ramingwong; S. N. Bhatti (2025): An Efficient Methodology for the Categorization of Software Requirements Using Natural Language Processing and Similarity Analysis. In: *IEEE Access* 13, S. 83591–83606. DOI: 10.1109/ACCESS.2025.3568504.
- R. Sonbol; G. Rebdawi; N. Ghneim (2022): The Use of NLP-Based Text Representation Techniques to Support Requirement Engineering Tasks: A Systematic Mapping Review. In: *IEEE Access* 10, S. 62811–62830. DOI: 10.1109/ACCESS.2022.3182372.
- Rahman, Tajmilur; Zhu, Yuecai (2024): Automated user story generation with test case specification using large language model. In: *arXiv preprint arXiv:2404.01558*.
- Ritter, Joachim; Gründer, Karlfried; Gabriel, Gottfried; Weber, Marcel (1971): Historisches wörterbuch der philosophie: Schwabe Basel (13).

- Sebastian, Glorin (2023): Privacy and data protection in chatgpt and other ai chatbots: Strategies for securing user information. In: *International Journal of Security and Privacy in Pervasive Computing (IJSPPC)* 15 (1), S. 1–14.
- Sharma, Shreta; Pandey, S. K. (2013): Revisiting requirements elicitation techniques. In: *International Journal of Computer Applications* 75 (12).
- Statistik Austria (16.10.2024): Nutzung von künstlicher Intelligenz in Unternehmen innerhalb eines Jahres fast verdoppelt. 20 % der Unternehmen nutzen bereits künstliche Intelligenz. Online verfügbar unter <https://www.statistik.at/fileadmin/announcement/2024/10/20241016IKTU2024.pdf>.
- Sternberg, Robert J.; Kaufman, Scott Barry (2011): *The Cambridge handbook of intelligence*: Cambridge University Press.
- vom Brocke, Jan; Hevner, Alan; Maedche, Alexander (2020): Introduction to design science research. In: *Design science research. Cases*, S. 1–13.
- Wei, Xiaoyang; Wang, Zhengdi; Yang, Shuangyuan (2023): An automatic generation and verification method of software requirements specification. In: *Electronics* 12 (12), S. 2734.
- Zhao, Liping; Alhoshan, Waad; Ferrari, Alessio; Letsholo, Keletso J.; Ajagbe, Muideen A.; Chioasca, Erol-Valeriu; Batista-Navarro, Riza T. (2022a): Natural Language Processing for Requirements Engineering: A Systematic Mapping Study. In: *ACM computing surveys* 54 (DOI: 10.1145/3444689), S. 41. DOI: 10.1145/3444689.
- Zhao, Ziyang; Zhang, Li; Gao, Xiaoyun; Lian, Xiaoli; Lv, Heyang; Shi, Lin (2022b): Automated Generating Natural Language Requirements based on Domain Ontology. In: *arXiv preprint arXiv:2211.16716*.
- Zhao, Ziyang; Zhang, Li; Lian, Xiaoli; Gao, Xiaoyun; Lv, Heyang; Shi, Lin (2023): ReqGen: Keywords-Driven Software Requirements Generation. In: *Mathematics* 11 (2). DOI: 10.3390/math11020332.