



# AI4RE Micro-Credential

## Lehrplan und Studienleitfaden

Gunnar Harde, Matthias Herbert,  
Simon Jimenez, Daniela Kreiner,  
Michael Mey, Michael Tesar,  
Franz Zehentner



## Nutzungsbedingungen

1. Einzelpersonen und Seminaranbieter dürfen den Lehrplan und Studienleitfaden als Grundlage für Seminare verwenden, sofern die Inhaber der Urheberrechte als Quelle und Besitzer des Urheberrechts anerkannt und benannt werden. Des Weiteren dürfen der Lehrplan und Studienleitfaden zu Werbungszwecken nur mit Einwilligung des IREB e. V. verwendet werden.
2. Jede Einzelperson oder Gruppe von Einzelpersonen darf den Lehrplan und Studienleitfaden als Grundlage für Artikel, Bücher oder andere abgeleitete Veröffentlichungen verwenden, sofern die Autoren und IREB e. V. als Quelle und Besitzer des Urheberrechts genannt werden.

© IREB e.V.

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Die Verwertung ist – soweit sie nicht ausdrücklich durch das Urheberrechtsgesetz (UrhG) gestattet ist – nur mit Zustimmung der Berechtigten zulässig, dies gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmung, Einspeicherung und Verarbeitung in elektronischen Systemen und öffentliche Zugänglichmachung.

## Danksagung

Dieser Lehrplan und Studienleitfaden wurde verfasst von: Gunnar Harde, Matthias Herbert, Simon Jimenez, Daniela Kreiner, Michael Mey, Michael Tesar, Franz Zehentner.

Reviews wurden durch Rainer Grau und Ana Moreira durchgeführt.

Zusätzliche Review-Kommentare von Gareth Rogers und Christian Ploninger.

Schlussredaktion durch Stan Bühne und Stefan Sturm. Übersetzung ins Deutsche durch Simon Jimenez, Stefan Sturm und Michael Tesar. Anmerkung der Übersetzer: Wenn man sich näher mit künstlicher Intelligenz beschäftigt, dann wird man zwangsläufig mit englischen Schlüsselbegriffen konfrontiert. Wir haben uns daher entschieden diese Schlüsselbegriffe in diesem Dokument auf Englisch zu belassen.

Allen sei für ihr Engagement gedankt.

Die Beta-Version wurde am 23. Oktober 2025 auf Empfehlung von Ana Moreira durch das IREB Council zur Veröffentlichung freigegeben. Version 1.0.0 wurde am XX. Februar 2026 vom IREB-Council auf Empfehlung von Ana Moreira zur Veröffentlichung freigegeben.

Urheberrecht © 2025 – 2026 für diesen Lehrplan und Studienleitfaden besitzen die aufgeführten Autoren. Die Rechte sind übertragen auf das IREB International Requirements Engineering Board e. V.

## Zweck des Dokuments

Dieser Lehrplan und Studienleitfaden definiert die Inhalte für das Micro-Credential **AI4RE (Artificial Intelligence for Requirements Engineering)** des International Requirements Engineering Board (IREB). Er dient zwei Hauptzwecken.

- Lernende können diesen Lehrplan und Studienleitfaden nutzen, um sich auf die Prüfung vorzubereiten und ein strukturiertes Verständnis dafür zu entwickeln, wie Künstliche Intelligenz (KI) im Requirements Engineering (RE) angewendet werden kann.
- Vom IREB anerkannte Schulungsanbieter können diesen Lehrplan und Studienleitfaden als Grundlage für die Entwicklung offizieller Schulungsangebote verwenden, die mit den Lernzielen des Micro-Credentials übereinstimmen.

Der Lehrplan und Studienleitfaden gewährleistet Konsistenz und Transparenz sowohl für Lernende im Selbststudium als auch für die Schulungsformate der vom IREB anerkannten Schulungsanbieter.

## Inhalt dieses Lehrplans und Studienleitfadens

Dieser Lehr- und Studienplan richtet sich an alle, die sich mit dem Thema RE befassen und Unterstützung für ihre Arbeit suchen. Er richtet sich an alle, die an Requirements-Engineering-Aktivitäten beteiligt sind, z. B. Requirements Engineers, Business Analysts, Product Owner und Product Manager.

## Inhaltsabgrenzung

Dieser Lehrplan und Studienleitfaden wurde verfasst, um den zunehmenden Einfluss der künstlichen Intelligenz auf RE zu berücksichtigen. Er spiegelt sowohl die Perspektive des IREB auf verantwortungsvolle RE-Praktiken als auch eine KI-orientierte Sichtweise wider, wie RE von Automatisierung profitieren kann. Der Inhalt umfasst typische KI-Technologien und ihre Anwendungsbereiche, die Verarbeitungspipeline hinter Chatbots sowie die spezifischen Stärken und Grenzen von Ansätzen wie Fine-Tuning und Retrieval-Augmented Generation (RAG). Es werden Möglichkeiten aufgezeigt, wie KI-Tools die Ermittlung, Dokumentation, Validierung und Verwaltung von Anforderungen unterstützen können. Gleichzeitig wird betont, wie wichtig die menschliche Aufsicht ist, um vertrauenswürdige Ergebnisse zu gewährleisten, und es wird auf potenzielle Gefahren der KI-Nutzung hingewiesen.

## Detaillierungsgrad

Der Detaillierungsgrad dieses Lehrplans und Studienleitfadens ermöglicht konsistentes Lehren auf internationaler Ebene. Um dies zu erreichen, umfasst der Lehrplan und Studienleitfaden folgende Punkte:

- Allgemeine Lernziele
- Inhalte mit einer Beschreibung der Lernziele
- Referenzen zu weiterführender Literatur, falls notwendig

## Lernziele/Kognitive Stufen des Wissens

Alle Module und Lernziele in diesem Lehrplan und Studienleitfaden sind einem kognitiven Niveau zugeordnet. Die Stufen sind wie folgt klassifiziert:

- **K1: Kennen** (beschreiben, aufzählen, charakterisieren, erkennen, benennen, erinnern, ...)—Sich an zuvor gelernten Stoff erinnern oder ihn abrufen.
- **K2: Verstehen** (erklären, interpretieren, vervollständigen, zusammenfassen, begründen, klassifizieren, vergleichen, ...)—Bedeutung anhand von gegebenem Inhalt oder Situationen begreifen/erfassen.



Die kognitive Ebene K2 umfasst die kognitive Ebene K1

### Beispiel:

Ein Lernziel des Typs "Verstehen Sie die RE-Technik xyz" gehört zur kognitiven Wissensstufe (K2). Die Fähigkeit zu verstehen setzt jedoch voraus, dass die Lernenden die RE-Technik xyz (K1) kennen.

## Struktur des Lehrplans und Studienleitfadens

Der Lehrplan und Studienleitfaden besteht aus 5 Hauptkapiteln. Ein Kapitel umfasst eine Lerneinheit (LE). Der Titel jedes Hauptkapitels spiegelt die kognitive Stufe wider, die durch die höchste Stufe der Unterkapitel definiert ist.

### Beispiel: LZ 3.1

Das Beispiel zeigt, dass das Lernziel LZ 3.1 im Unterkapitel 3.1 beschrieben wird.

Darüber hinaus gibt es ein Kapitel über die Kernbegriffe, die für die Arbeit mit KI unerlässlich sind. Dies dient als Referenz für den Leser. In den Prüfungen können diese Begriffe im Zusammenhang mit RE-bezogenen Fragen verwendet werden.

## Versions-Historie

Version	Datum	Kommentar
1.0.0	1. März 2026	Erste Version basierend auf der englischen Version 1.0.0.

## Inhaltsverzeichnis

<b>1</b>	<b>KI-Grundlagen</b>	<b>7</b>
1.1	KI-Schlüsseltechnologien und ihre Anwendungsbereiche	7
1.2	Hinter den Kulissen von Chatbots: Vom Prompt zur Antwort und RAG versus Fine-Tuning	8
1.3	Wichtige Erkenntnisse	9
<b>2</b>	<b>Large Language Modelle (LLMs)</b>	<b>10</b>
2.1	Das Grundkonzept der LLMs	10
2.2	Wichtige Erkenntnisse	14
<b>3</b>	<b>Prompt Engineering</b>	<b>15</b>
3.1	Die Bedeutung des Kontexts	15
3.2	Prompt-Pattern und Prompt-Techniken	16
3.3	Wichtige Erkenntnisse	18
<b>4</b>	<b>Risiken und Verantwortlichkeiten in AI4RE</b>	<b>19</b>
4.1	Risiken in AI4RE	20
4.2	Zuständigkeiten in AI4RE	24
4.3	Wichtige Erkenntnisse	26
<b>5</b>	<b>Anwendungsfälle für KI im RE</b>	<b>27</b>
5.1	Anforderungsermittlung	28
5.1.1	Erkundung neuer Domänen	28
5.1.2	Kommunikation mit Stakeholdern transkribieren	30
5.1.3	Extraktion von Anforderungen aus verschiedenen Quellen	31
5.2	Dokumentation	32
5.2.1	Formulierung von Anforderungen	33
5.2.2	Umwandlung von Anforderungen in andere Darstellungen	34
5.3	Validierung	36

5.4	Management.....	38
5.4.1	Zuweisung von Attributen zu Anforderungen.....	38
5.4.2	Priorisierung von Anforderungen.....	38
5.5	Wichtige Erkenntnisse.....	39
6	KI-Terminologie.....	40
7	Referenzen.....	45

# 1 KI-Grundlagen

Stufe: K2

Dauer: 0,5 Stunden

Ziel: Kandidaten können zwischen den verschiedenen Arten von KI-Tools und – Modellen unterscheiden, verstehen, wie sie entwickelt und eingesetzt werden und sind mit Schlüsselkonzepten wie Informationsfluss, RAG und Fine-Tuning vertraut.

## Lernziele

- LZ 1.1 Kennen der Vielfalt der KI-Technologien und ihrer Anwendungsbereiche – nicht jede KI ist ein Chatbot. (K1)
- LZ 1.2 Erklären, was bei der Interaktion mit einem Chatbot hinter den Kulissen passiert, können das Konzept der Retrieval-Augmented Generation (RAG) beschreiben und mit Fine-Tuning vergleichen. (K2)

## 1.1 KI-Schlüsseltechnologien und ihre Anwendungsbereiche

Künstliche Intelligenz umfasst alle Methoden, die es Computern ermöglichen, auf eine Weise wahrzunehmen, zu schlussfolgern oder zu handeln, die wir als intelligent bezeichnen. Je nach Autor gibt es unterschiedliche Einstufungen von KI. Fünf davon tauchen in fast allen Taxonomien auf (ausführliche Definitionen finden Sie in Kapitel 6):

- **Maschinelles Lernen (ML) / Deep Learning:** Algorithmen lernen Muster direkt aus Daten.
- **Computer Vision:** Computer Vision verwandelt rohe Pixel in strukturierte Fakten.
- **Natural language processing (NLP):** NLP umfasst alles, was Sprache oder Text in Bedeutung umwandelt und umgekehrt. Dazu gehören auch Large Language Models (LLMs), die in Kapitel 2 näher erläutert werden.
- **Robotik:** Die Robotik kombiniert die Sensordatenfusion (Integration von Daten mehrerer Sensoren), die Kartierung (Erstellung eines Abbilds der Umgebung), die Wegplanung (Entscheidung, wie etwas sich bewegt) und die Rückkopplungssteuerung (Anpassung von Aktionen auf der Grundlage von Sensoreingaben), damit Maschinen autonom navigieren und die physische Welt manipulieren können.
- **Expertensysteme:** Expertensysteme erfassen menschliches Fachwissen in Form von expliziten Regeln und Ontologien und liefern überprüfbare, deterministische Antworten.

KI-basierte Chatbots sind für Requirements Engineers zwar praktisch, um Anforderungen zu entwerfen oder umzuformulieren, aber sie stellen nur einen Teil des KI-Toolkits dar. Funktionen wie die Transkription von Besprechungen (Sprache-zu-Text) und die Analyse der Benutzeroberfläche (Computer Vision) sind häufig in einer einzigen Schnittstelle integriert. Dies kann durch die Kombination spezialisierter KI-Modelle oder zunehmend durch die Verwendung eines einzigen, multimodalen Modells erreicht werden.

## 1.2 Hinter den Kulissen von Chatbots: Vom Prompt zur Antwort und RAG versus Fine-Tuning

Chatbots wurden zu den bekanntesten Vertretern der modernen KI-Anwendungen. Ziel dieses Kapitels ist es, die Phasen zwischen der Eingabe eines Prompts (Eingabeaufforderung) und der Anzeige einer Antwort zu verstehen sowie Fine-Tuning von RAG zu unterscheiden.

Ein Chatbot sieht aus wie eine Blackbox. Eine Anfrage durchläuft mehrere Bearbeitungsstufen, die alle zur Antwort beitragen, die Sie schließlich erhalten.

Die Kenntnis dieses Ablaufs hinter den Kulissen hilft Ihnen bei der Fehlersuche, bei der Erstellung präziser Prompts und bei der Festlegung realistischer Systemanforderungen.

Die folgende vereinfachte Pipeline kann je nach Tool-Anbieter leicht variieren, fasst aber die wichtigsten Phasen zusammen.

- **Input:** Der Benutzer sendet einen Prompt mit optionalen Kontextelementen wie zusätzlichen Dateien, Bildern oder Dokumenten. Bei der Verwendung in einem Chat wird der gesamte vorherige Austausch als Vortext zum Prompt hinzugefügt. Stellen Sie sich vor, der Chatbot hat kein Gedächtnis und muss bei jeder Eingabe die gesamte Konversation erneut lesen.
- **Pre-processing:** Das System zerlegt Text in sog. Token [SeHB2016], entfernt unsichere Inhalte und normalisiert die Sprache (im Falle einer nicht-textuellen Eingabe, z. B. Sprache, wird das System auch hier zunächst diese Eingabe in Text umwandeln).
- **Context assembly:** Der Systemkontext (siehe 3.1) und der Sitzungskontext (oder Benutzerkontext) werden zusammengeführt und bilden den endgültigen Kontext, der für die aktuelle Konversation gilt.
- **Inference:** Das trainierte System, z. B. ein LLM, verwendet sein erlerntes Wissen, um aus den neuen Eingabedaten eine Ausgabe zu erzeugen oder eine Vorhersage zu treffen. Dies ist die "Laufzeit" Phase in der das Modell anwendet, was es beim Training gelernt hat. Im RE erfolgt eine Inferenz beispielsweise, wenn ein LLM eine Aufforderung verarbeitet, die vorgeschlagene Anforderungen enthält, und dann eine neu formulierte Anforderung erstellt oder auf der Grundlage seines Verständnisses potenzielle Probleme identifiziert.
- **Post processing:** Das System wandelt rohe Token zurück in Text oder andere Ergebnistypen (z. B. Sprache oder Bilder). Optionale Schritte wie Faktenchecker, Profanitätsfilter oder Formatierungsregeln können ebenfalls angewendet werden und variieren stark zwischen den verschiedenen LLM-Modellanbietern.
- **Output:** Das System zeigt dem Benutzer das Ergebnis im Chatfenster an.

Die Anbieter können Schleifen durch die Schritte der Kontextzusammenstellung und -ableitung durchlaufen, zusätzliche Tools aufrufen oder Chain-of-Thought (siehe Kapitel 3.2) Prompts durchlaufen, bis Qualitätskriterien erfüllt sind.

In der Phase *Context assembly* gibt es zwei Möglichkeiten, zusätzliches Wissen einzubeziehen:

1. **Fine-Tuning** bezieht das zusätzliche Wissen direkt in das Modell ein und macht es so für alle künftigen Interaktionen verfügbar. Ein Nachteil dieser Methode ist, dass das Modell jedes Mal neu trainiert werden muss, wenn Sie neue Informationen hinzufügen, was sehr ressourcenintensiv sein kann. Außerdem ist es schwieriger, die Quelle des Wissens zu bestimmen, da es nach dem Fine-Tuning Teil des Modells wird.
2. **RAG** nutzt einen separaten Wissensspeicher, auf den mit einer vom Modell ausgelösten Suchmaschine zugegriffen werden kann. Neu hinzugefügte Daten/Informationen sind sofort verfügbar, ohne dass das Modell neu trainiert werden muss, und die Quelle kann bis zum Wissensspeicher zurückverfolgt werden.

Fine-Tuning eignet sich hervorragend, wenn Sie sich wiederkehrendes Training leisten können und Antworten mit geringer Latenzzeit ohne separaten Wissensspeicher benötigen, aber es läuft Gefahr, zu veralten und bietet keine integrierten Quellenangaben. RAG hält Inhalte frisch und nachvollziehbar.

Ein Nachteil ist, dass RAG eine zusätzliche Suchlatenz mit sich bringt und von der Qualität Ihres Abrufindex abhängt, d.h. die abgerufenen Informationen sind nur wertvoll, wenn sie von guter Qualität sind.

### 1.3 Wichtige Erkenntnisse

KI hat viel mehr zu bieten als Chatbots. Für einige Anwendungsfälle sind LLMs nicht das richtige Werkzeug. Die richtige KI-Technologie für die Aufgabe auszuwählen ist der Schlüssel zum bestmöglichen Ergebnis; die Kenntnis der verfügbaren Möglichkeiten ist der Schlüssel zur Nutzung des vollen Potenzials der KI.

Das Hinzufügen von Wissen zu Ihrem aktuellen Chatbot kann entweder durch Fine-Tuning und Einbindung des Wissens in das Modell selbst oder durch die Verwendung von RAG erfolgen, um flexibler zu sein und Wissen aus externen Quellen bei Bedarf hinzuzufügen.

## 2 Large Language Modelle (LLMs)

Stufe: K1

Dauer: 0,5 Stunden

Ziel: Kandidaten können darlegen, dass es sich bei LLMs (Große Sprachmodelle) um probabilistische Textvorhersagesysteme handelt und ihre gemeinsamen Fähigkeiten und Grenzen aufzählen. Kandidaten wissen, worauf sie bei der Interaktion mit LLMs achten müssen, können Kriterien für die Überprüfung ihres Outputs erinnern und Aspekte nennen, die bei ihrer verantwortungsvollen Integration in Arbeitsabläufe zu berücksichtigen sind.

### Lernziele

LZ 2.1 Kennen des grundlegenden Konzepts, wie ein LLM grob funktioniert. (K1)

LZ 2.2 Beschreiben der Funktionsweise von LLMs, die auf einer probabilistischen Vorhersage des nächsten Wortes und nicht auf Schlussfolgerungen oder Verständnis beruht. (K2)

### 2.1 Das Grundkonzept der LLMs

#### Der Kernmechanismus: Vorhersage, nicht Verständnis

In den letzten Jahren haben sich Large Language Modelle [ZhWX2023] als eine Klasse von KI-Systemen herausgebildet, die in der Lage sind, menschenähnliche Texte zu verarbeiten und zu erzeugen. Sie werden auf riesigen Mengen von Textdaten trainiert und nutzen statistische Muster in der Sprache, um Aufgaben wie das Verfassen von Dokumenten, das Zusammenfassen von Informationen oder das Generieren von Testfällen durchzuführen [VoFi2025]. Für einen Requirements Engineer ist es wichtig, die Fähigkeiten und Grenzen von LLMs zu verstehen, da diese Systeme ein breites Spektrum von RE-Aktivitäten unterstützen können—von der Ermittlung und Dokumentation von Anforderungen bis hin zur Erstellung von Domänenglossaren—und gleichzeitig neue Risiken und Überlegungen einbringen [NoeA2024], [ZaDA2025].

Wenn Menschen ein Large Language Modell zum ersten Mal ausprobieren, können sich die Antworten menschlich anfühlen. Der Text ist flüssig, der Tonfall passt zur Frage, und das Ergebnis scheint oft Verständnis zu zeigen. Die wahre Geschichte ist einfacher: LLMs sind statistische Textvorhersagesysteme, die auf der Grundlage von Mustern, die aus Trainingsdaten gelernt wurden, ein Wort nach dem anderen erzeugen. Sie verstehen nicht und denken nicht nach—sie suchen nach Mustern und schreiben weiter. Auf Basis weniger Worte wählen sie das nächste kleine Stück aus, das am wahrscheinlichsten ist. Man kann es sich als eine sehr leistungsfähige Autovervollständigung vorstellen. Diese Idee erklärt den größten Teil ihres Verhaltens.

## Probabilistische Generierung und ihre Folgen

Das Modell wird mit einer riesigen Menge an Text trainiert. Während des Trainings sieht es viele Beispiele dafür, wie Wörter in verschiedenen Situationen aufeinander folgen, vom lockeren Gespräch bis zum technischen Schreiben. Das Ziel ändert sich nie: für jeden gegebenen Kontext das nächste Token zu erraten, das ein kleines Stück Text ist, das ein Wort, ein Teil eines Wortes oder eine Zeichensetzung sein kann. Wenn Sie später eine Frage stellen, verwendet das Modell das Gelernte, um Ihren Text Token für Token fortzusetzen, wobei jede Auswahl auf dem Vorangegangenen basiert.

Das funktioniert gut, weil Sprache voller Muster ist. Grammatik schafft Struktur, gebräuchliche Ausdrücke wiederholen sich in vielen Themenbereichen, und viele Dokumente folgen einer vorhersehbaren Struktur. Wenn ein System sehr gut darin ist, diese Muster fortzuführen, wirkt die Ausgabe kohärent und relevant. Moderne Modelle verwenden eine Architektur, die als Transformer (Transformator) [VaeA2017] bezeichnet wird.

Ein wichtiger Teil davon, die Self-Attention, hilft dem Modell zu entscheiden, welche Teile Ihrer Eingaben für den nächsten Schritt am wichtigsten sind. Deshalb kann das Modell ein Pronomen mit dem richtigen Namen verbinden oder ein Thema verfolgen, das viele Sätze zuvor eingeführt wurde.

## Implikationen und mentale Modelle für das Requirements Engineering

Der wichtige Punkt ist, was das System nicht tut. Es denkt nicht wie ein Mensch und hat kein inneres Bild von der Welt. Hinter den Sätzen steckt keine Absicht und kein Verständnis. Das Modell erkennt und reproduziert Muster, die es in vielen Texten gesehen hat. Da diese Muster reichhaltig und vielfältig sind, kann das Ergebnis wie Verständnis aussehen, aber der Mechanismus bleibt eine statistische Vorhersage.

Zum Zeitpunkt der Generierung erzeugt das Modell eine Wahrscheinlichkeit für jedes mögliche nächste Token unter Berücksichtigung des gesamten bisherigen Kontextes. Es wählt dann eines aus und fügt es der Ausgabe hinzu. Manchmal wird die oberste Option gewählt, die in der Regel einen sicheren, gleichmäßigen Text ergibt. Manchmal werden Daten aus der Verteilung genommen, um für Abwechslung zu sorgen. Eine kleine Portion Zufälligkeit kann helfen, langweilige oder sich wiederholende Antworten zu vermeiden. In jedem Fall ist der sichtbare Text das Ergebnis vieler solcher kleinen Entscheidungen, wobei jede einzelne eine Auswahl des nächsten Tokens auf der Grundlage des bereits Vorhandenen ist.

Da es sich dabei um einen probabilistischen Prozess handelt, kann das Modell Fehler machen, die dennoch überzeugend klingen. Wenn Sie es z. B. bitten, eine lange Multiplikation durchzuführen, kann es eine Zahl ausgeben, die zwar vernünftig aussieht, aber falsch ist. Es hat sie nicht berechnet. Es sagte einfach eine Zahl voraus, die statistisch gesehen dem Muster einer Multiplikationsaufgabe entspricht, ohne die eigentliche Berechnung durchzuführen. Um deterministische Aufgaben, d. h. Aufgaben, die einem eindeutigen Algorithmus folgen, korrekt zu berechnen, verwenden bestimmte Modelle ein Werkzeug im Hintergrund [SceA2023], z. B. einen Kalkulator oder einen Code Runner, während das Sprachmodell die Anweisungen und die Erklärung verarbeitet. Das Modell fährt mit der Vorhersage des Textes fort und verwendet das Werkzeug für die deterministische Aufgabe.

Manchmal wird ein Modell gebeten, seine Schritte zu zeigen [WeeA2022]. Dies kann die Ergebnisse bei Aufgaben verbessern, die von einer Strukturierung profitieren. Der Grund dafür ist einfach: Das Modell hat in seinen Trainingsdaten viele Beispiele für schrittweise Erklärungen gesehen (aus Lehrbüchern, Tutorien, Fragen und Antworten), so dass es dieses Format imitieren kann. Reasoning-Systeme (Schlussfolgerungssysteme) führen sogar interne Planungsschritte durch, bevor sie eine Antwort geben. Sie entwerfen und überarbeiten intern und präsentieren dann ein Ergebnis. Dies kann die Qualität verbessern, ändert aber nicht den Kern. Das System sagt den Text immer noch voraus, nur eben in mehreren Runden, bevor Sie ihn sehen.

Es ist hilfreich, zwei Ebenen (Oberfläche und Fakten) zu trennen, wenn Sie die Modellausgabe lesen. Die *Oberflächen-Ebene* umfasst Sprachgewandtheit, Tonfall und Struktur und ist in der Regel sehr stark. Diese Merkmale sind in den Daten häufig anzutreffen und leicht zu reproduzieren. Die *Fakten-Ebene* umfasst Daten, Mengen, Namen und Randfälle. Hier kann der reine Mustervergleich scheitern. Wenn die Trainingsdaten in irgendeinem Punkt dürftig oder inkonsistent sind, kann die wahrscheinlichste Fortführung von der Wahrheit abweichen. Die Verbindung des Modells mit Suchmaschinen, Datenbanken oder Funktionsaufrufen verbessert diese Ebene, da die Vorhersage des nächsten Tokens auf frischen oder berechneten Informationen beruht [LeeA2020]. Das Modell macht immer noch Vorhersagen, aber auf Basis besserer Eingaben.

Für einen Requirements Engineer ist das effektivste mentale Modell für eine KI eine leistungsstarke Autovervollständigung. Damit ist seine Rolle sofort klar: Er ist ein Entwurfsassistent, kein Fachexperte.

Das Modell ist sehr empfänglich für Muster, daher sollten Sie ein klares Beispiel für die gewünschte Ausgabe bereitstellen [BreA2020]. Wenn Sie eine Anforderung in einem bestimmten Stil benötigen, fügen Sie ein kurzes Beispiel bei und bitten das Modell, es für neue Inhalte zu übernehmen. Dadurch verschieben sich die Wahrscheinlichkeiten zugunsten von Versionen, die zu Ihrer Vorlage und Ihrem Stil passen. Sie können sogar mehrere User Stories übergeben, die Sie in der Vergangenheit geschrieben haben, um den Ton und die Struktur festzulegen. Wenn Sie Wert auf Qualitätskriterien wie Testbarkeit, eindeutige Formulierung oder messbare Kriterien legen, geben Sie diese direkt an. Das Modell wird diese Merkmale oft einbeziehen, weil ähnliche Anweisungen in seinen Trainingsdaten vorkommen.

Im Requirements Engineering ist dies sowohl für die Erstellung als auch für die Überprüfung von Bedeutung. Eine vage Anweisung wie "Schreibe eine Anforderung aus diesem Text" lädt zu einer vagen Antwort ein. Eine präzise Anweisung, die eine Vorlage und eine klare Definition des erwarteten Detailgrads enthält, lenkt die Ausgabe in Richtung Ihrer Standards. Sie können das Modell auch auffordern, mehrdeutige Begriffe zu markieren und messbare Alternativen vorzuschlagen. Es wird in der Regel einen nützlichen ersten Entwurf produzieren, da es viele Beispiele dafür gesehen hat, wie Menschen über Klarheit und Messbarkeit diskutieren.

## Ein Blick auf die technischen Grundlagen: Embeddings und Vektoren

Diese Modelle sind per se nicht deterministisch. Wenn Sie dieselbe Frage zweimal stellen, erhalten Sie möglicherweise unterschiedliche Antworten, da die Texterstellung auf Wahrscheinlichkeiten beruht. Schon ein einziges Wort in Ihrem Prompt kann das Modell auf einen anderen Weg bringen und zu einem ganz anderen Ergebnis führen. Diese Empfindlichkeit bedeutet, dass das Modell durch einen einzelnen Begriff oder ein Beispiel abgelenkt werden kann und von dem, was Sie wollen, abweicht. Mehrere verschiedene Antworten können allerdings alle richtige und gültige Reaktionen auf Ihren Prompt sein.

Eine kurze Erklärung, warum das so ist. Wenn das Modell schreibt, wählt es nicht immer das wahrscheinlichste nächste Token aus. Es wählt aus einer Bandbreite wahrscheinlicher Optionen aus. Die Einstellung, die diese Zufälligkeit steuert, wird als Temperatur bezeichnet: Eine niedrige Temperatur führt dazu, dass sich das Modell an sicherere, naheliegendere Entscheidungen hält, was oft zu konsistenteren Antworten führt; eine höhere Temperatur lässt es mehr erkunden und variieren, was für Brainstorming nützlich sein kann, aber weniger stabil ist.

Unabhängig von dieser Variabilität sollten Sie den KI-Output immer als einen Entwurf behandeln, der überprüft werden muss. Diese Überprüfung dient zwei wichtigen Zwecken: erstens, um Halluzinationen (sachlich falsche Informationen, die das Modell als wahr darstellt [YeeA2023]) zu erkennen und auszuschließen, und zweitens, um das Ergebnis für Ihren spezifischen Kontext und Ihre Anforderungen zu optimieren. Das Modell ist nicht so zuverlässig wie ein Taschenrechner – selbst wenn es konsistente Ergebnisse liefert, können diese Ergebnisse noch Fehler enthalten oder nicht perfekt Ihren Anforderungen entsprechen.

Diese Perspektive verdeutlicht auch das Risiko. Wenn das Modell eine falsche Zahl angibt oder sich auf die falsche Norm beruft, ist das kein Fehler in der Logik. Es ist bei der Vorhersage am falschen Ort erfolgreich. Abhilfe schafft die Überprüfung und Verankerung. Geben Sie in Ihrem Prompt den richtigen Kontext an, verknüpfen Sie das Modell mit Abfrage- oder Berechnungswerkzeugen, wenn es auf Präzision ankommt, und behandeln Sie das Ergebnis als einen Entwurf, der überprüft werden muss. Das Modell muss nichts davon verstehen. Es muss lediglich zum richtigen Muster geführt werden, und Sie müssen sich auf Tools und Quellen verlassen, wenn Muster allein nicht ausreichen.

**Wenn Sie sich dafür interessieren, warum das funktionieren kann:**

Das Modell stellt auch Token als Vektoren dar, die lange Listen von Zahlen sind. Token, die in ähnlichen Kontexten auftreten, erhalten ähnliche Vektoren. Auf diese Weise entsteht eine Art Geometrie, die sich mit der Bedeutung in der Praxis deckt. Ein einfaches Beispiel aus früheren Methoden verdeutlicht die Idee. Der Unterschied zwischen den Vektoren für König und Königin ähnelt tendenziell dem Unterschied zwischen Mann und Frau [MiSC2013]. Das System speichert keine Wörterbucheinträge, sondern lernt numerische Muster, die widerspiegeln, wie Menschen Wörter verwenden.

Deshalb kann das Modell die semantische Bedeutung erfassen. Token, die in ähnlichen Kontexten vorkommen, erhalten ähnliche numerische Repräsentationen, sogenannte Embeddings (Einbettungen). In der Praxis bedeutet dies, dass der Vektor für "Benutzeranforderung" näher an "Stakeholder Bedarf" liegt als an "Datenbankschema". Dies ermöglicht es dem LLM, ähnliche Anforderungen zu identifizieren oder Abhängigkeiten zu finden, selbst wenn eine andere Terminologie verwendet wird.

## 2.2 Wichtige Erkenntnisse

Zusammenfassend beschreibt dieses Kapitel die grundlegenden Verhaltensweisen und die Terminologie von LLMs, die ihre Anwendung in RE beeinflussen. Ein Requirements Engineer, der Prompts, Tokens, Halluzinationen, Fine-Tuning und verwandte Konzepte wie Embeddings und RAG versteht, kann besser einschätzen, wann und wie LLMs effektiv eingesetzt werden können, um sicherzustellen, dass die generierten Ergebnisse einen Mehrwert darstellen und die Qualität der Anforderungen nicht beeinträchtigen.

# 3 Prompt Engineering

Stufe: K2

Dauer: 0,5 Stunden

Ziel: Kandidaten verstehen die Grundlagen des Promptens, einschließlich Qualitätskriterien, Techniken und Mustern, und können erklären, wie diese Elemente die Qualität von KI-generierten Ergebnissen im RE beeinflussen.

## Lernziele

- LZ 3.1 Erläutern der Bedeutung des Kontexts um bei der Verwendung von KI-basierten Tools akkurate Ergebnisse zu erzielen. (K2)
- LZ 3.2 Benennen des allgemeinen Zwecks von Prompt-Pattern im RE und erkennen wie sie sich von Prompt-Techniken unterscheiden. (K1)

## 3.1 Die Bedeutung des Kontexts

LLMs wie GPT, Claude, Gemini oder ähnliche generieren Text, indem sie das wahrscheinlichste nächste Output-Token vorhersagen. Dies deutet nicht auf ein "Verstehen" des Inputs und der aktuellen Situation in einer Konversation hin. Ein klar definierter Kontext grenzt den Lösungsraum ein und lenkt das Modell in Richtung Domänenwissen, was zu präziseren und umsetzbaren Ergebnissen führt.

Es gibt zwei Kontextebenen, die die Qualität der Ergebnisse beeinflussen:

- **Systemkontext:** Der Systemkontext wird für jede Sitzung festgelegt. Er ergibt sich aus dem System oder dem ersten Prompt durch den Hersteller des Modells, den Trainingsdaten des Modells und den integrierten Einstellungen wie der Temperatur, die die Zufälligkeit der Ausgabe steuert. Diese Faktoren bestimmen Aspekte wie die Standardsprache, den Ton der Antworten und den Stichtag für das Hintergrundwissen. Ein Modell kann zum Beispiel immer auf Englisch antworten, einen neutralen Stil verwenden und nur Wissen bis zu einem bestimmten Datum einbeziehen. Der Systemkontext ist in der Regel nicht veränderbar, kann aber durch Benutzereingaben ergänzt werden. Die Fähigkeit, Kontext hinzuzufügen, wird durch die Größe des Kontextfensters (context window) begrenzt, das die maximale Anzahl von Token – Eingabe und Ausgabe zusammen – definiert, die ein Modell in einer einzigen Interaktion verarbeiten kann.
- **Benutzerkontext:** Der Benutzerkontext ist dynamisch und ändert sich mit jeder Konversation oder sogar mit jeder Nachricht, die Sie mit einem LLM austauschen. Dies kann Einzelheiten über das Projekt oder das Produkt enthalten, z. B. ein medizinisches Gerät nach ISO 13485, ein PSD2-konformes Bankensystem oder eine E-Learning-Plattform. Er deckt auch die Perspektive von Stakeholdern wie Pflegepersonal, Finanzanalysten oder Studenten ab, sowie anzuwendende Normen und Vorschriften wie IEC 62304 [IEC62304], GDPR [GDPR2016] oder FDA 21 CFR Part 11 [FDA1997].

Darüber hinaus kann der Benutzerkontext die Rolle definieren, die die KI einnehmen soll (z. B. "Sie sind ein erfahrener Requirements Engineer"), den Zweck der Abfrage angeben (z. B. die Erstellung von Testfällen oder die Validierung von Anforderungen) und unterstützendes Material wie Vorlagen oder Beispiele bereitstellen.

Wenn der Kontext fehlt, verlässt sich das Modell ausschließlich auf seine Trainingsdaten und gibt Antworten auf der Grundlage statistischer Wahrscheinlichkeiten. Solche Ergebnisse mögen zwar plausibel klingen, haben aber oft keinen Bezug zur Domäne und keinen praktischen Wert. Bei RE bedeutet dies in der Regel, dass das Ergebnis weder umsetzbar noch von ausreichender Qualität ist.

## 3.2 Prompt-Pattern und Prompt-Techniken

Bei der Interaktion mit großen LLMs hängt die Qualität der Ausgabe nicht nur vom Kontext ab (was das Modell wissen sollte), sondern auch davon, wie die Anweisungen formuliert sind.

### Prompt-Pattern

Prompt-Pattern sind wiederverwendbare Strukturen, die helfen, Prompts konsistent und effektiv zu formulieren.

Ein gängiges Pattern ist das Role-Task-Format (RTF):

- **Rolle:** Eine Beschreibung der Rolle, die die KI übernehmen soll.
- **Task:** Die spezifische Aufgabe, die zu erfüllen ist.
- **Format:** Die erwartete Struktur des Outputs.

#### Beispiel:

Du bist ein professioneller Business Analyst, der klare Zusammenfassungen von Workshop-Ergebnissen erstellt.

Lies die folgenden Workshop-Notizen und erstelle eine strukturierte Zusammenfassung, die Ziele, Erkenntnisse, Entscheidungen und nächste Schritte hervorhebt.

Gib die Ergebnisse in Form von kurzen Aufzählungspunkten unter klaren Abschnittsüberschriften an.

Durch die Kombination dieser drei Elemente können die Benutzer das Modell von allgemeinen Antworten weg und hin zu präziseren und umsetzbaren Ergebnissen führen.

Es gibt viele solcher Pattern (siehe [WheA2023]). Einige wichtige Beispiele sind:

- **Chain-of-Thought:** Fordern Sie die KI auf, ihre Überlegungen Schritt für Schritt zu erläutern, bevor sie das Endergebnis angibt.
- **CRISPE:** Detailliertes Pattern mit Kontext, Rolle, Anweisungen, Schritten, Parametern und Beispielen.

## Vorteile von Prompt-Pattern

Prompt-Pattern bieten mehrere Vorteile, die sowohl die Qualität der KI-Ergebnisse als auch die Effizienz des RE-Prozesses verbessern.

- **Praktische Vorteile:** Prompt-Pattern erhöhen die Verwendbarkeit von Antworten, indem sie im Vorfeld klare Anweisungen geben. Die Festlegung des gewünschten Formats reduziert den Bedarf an Nachbearbeitung und spart Zeit. Gut strukturierte Prompts verkürzen auch die Anzahl der Iterationen, die erforderlich sind, um zu brauchbaren Ergebnissen zu gelangen.
- **Kognitive Vorteile:** Die Verwendung von Prompt-Pattern zwingt die Nutzer, ihre Bedürfnisse genau zu überdenken, bevor sie die KI fragen. Indem die Rolle, die Aufgabe und das Format explizit gemacht werden, wird die Mehrdeutigkeit minimiert und die Gedanken hinter einem Prompt werden transparent und reproduzierbar.
- **Methodische Vorteile:** Prompt-Pattern können für verschiedene Aufgaben wiederverwendet werden, was Konsistenz gewährleistet und die Lernkurve für neue Benutzer senkt. Innerhalb eines Teams schaffen sie einen gemeinsamen Standard für die Arbeit mit KI. In Kombination mit Prompt-Techniken (siehe unten) bieten sie sowohl eine stabile Struktur als auch situative Flexibilität, was zu zuverlässigeren und anpassungsfähigeren Ergebnissen führt.

## Prompt-Techniken

Prompt-Techniken sind situationsabhängige Formulierungen von Prompts. Anders als Pattern schreiben sie keine Struktur vor, sondern beeinflussen, wie das Modell Anfragen interpretiert:

- **Zero-Shot:** eine Anfrage ohne Angabe von Beispielen

### Beispiel:

Bitte lies das folgende Interviewtranskript und erstelle eine kurze Zusammenfassung, in der die Hauptthemen, Schlüsselargumente und wichtigen Aussagen hervorgehoben werden. Vermeide persönliche Interpretationen.

Transkript: [hier eine Abschrift des Interviews einfügen]\*

\* Text in eckigen Klammern [] ist ein Platzhalter für Eingaben des Benutzers

- **Few-Shot:** Geben Sie ein paar Beispiele an und stellen Sie dann Ihre Anfrage

**Beispiel:**

Hier sind zwei Beispiele für die Umwandlung von Anforderungen in User Stories

Anforderung: "Aufgaben suchen" → User Story: Als Benutzer möchte ich Aufgaben suchen, damit ich Elemente schnell finden kann.

Anforderung: "Passwort zurücksetzen" → User Story: Als Benutzer möchte ich mein Passwort zurücksetzen, um wieder Zugang zu erhalten.

Nun dein Aufgabe:

Konvertiere die folgenden Anforderungen in User Stories:

- Prioritäten einer Aufgabe zuordnen
- Kommentare zu einer Aufgabe hinzufügen
- Aufgaben nach Status filtern

- **Vorlagenbasiert:** Bereitstellen einer Struktur, die die KI für ihre Ergebnisse verwenden soll

**Beispiel:**

Lies die folgenden Workshop-Notizen und fasse die Ergebnisse zusammen. Verwende genau diese Struktur:

1. Titel des Workshops:
2. Datum und Teilnehmer:
3. Hauptziele: 3-5 Aufzählungspunkte
4. Wichtige Einblicke: Aufzählungsliste
5. Getroffene Entscheidungen: Aufzählungsliste
6. Offene Fragen / Nächste Schritte: Aufzählungsliste

Workshop-Notizen: [hier Workshop-Notizen hinzufügen]\*

\* Text in eckigen Klammern [] ist ein Platzhalter für Eingaben des Benutzers

### 3.3 Wichtige Erkenntnisse

LLMs treffen Voraussagen, sie verstehen nicht. Im RE hängen zuverlässige KI-Ergebnisse von einem präzisen und vollständigen Kontext ab. Ohne Kontext fallen die Ergebnisse auf Klischees und Annahmen zurück; mit reichhaltigem Kontext spiegeln sie Fachwissen wider. Ein guter Kontext allein reicht jedoch nicht aus: Die Art und Weise, wie die Prompts gestaltet sind, bestimmt, wie effektiv die KI diesen Kontext nutzen kann. Prompt-Pattern bieten wiederverwendbare Strukturen, während Prompt-Techniken Interpretation und Stil verfeinern. Gemeinsam unterstützen sie konsistente, kontextbezogene Ergebnisse, die auf die Bedürfnisse der Stakeholder zugeschnitten sind.

## 4 Risiken und Verantwortlichkeiten in AI4RE

Stufe: K2

Dauer: 1 Stunde

Ziel: Kandidaten können die Risiken, Beschränkungen und Datenschutzanforderungen beim Einsatz von KI im Bereich RE erklären und die wichtigsten Aspekte eines verantwortungsvollen und gesetzeskonformen Einsatzes beschreiben.

### Lernziele

LZ 4.1 Nennen der typische Risiken, die mit dem Einsatz von KI im RE verbunden sind. (K1)

LZ 4.2 Erläutern der sich entwickelnden Rolle des Requirements Engineers in KI-gestützten Prozessen. (K2)

Die wirksame Anwendung von KI im RE erfordert eine umfassende menschliche Überprüfung, Validierung und Kuratierung aller unterstützten Aktivitäten. KI-generierte Fragen, Glossareinträge und extrahierte Inhalte müssen sorgfältig beurteilt werden, um sicherzustellen, dass sie mit dem Projektkontext und den Bedürfnissen der Stakeholder übereinstimmen. KI kann zwar Terminologiemuster erkennen und Interviewtranskripte verarbeiten, doch können automatische Ergebnisse domänenspezifische Konzepte falsch interpretieren oder Mehrdeutigkeiten einführen, die das gemeinsame Verständnis beeinträchtigen (CPRE-Prinzip 3, [IREB2024]). Besonders kritisch wird der Validierungsprozess, wenn KI schriftliche oder gesprochene Kommunikation verarbeitet. Zufällige Beobachtungen und spekulative Diskussionen erfordern eine menschliche Interpretation, um echte Anforderungen von vorläufigen Ideen zu unterscheiden, die möglicherweise keine formale Dokumentation durch etablierte Erhebungsverfahren rechtfertigen (siehe Kapitel 4.2 in [GleA2024]).

Template-basierte KI-Outputs, einschließlich formatierter Anforderungen und User Stories, erfordern eine strenge Überprüfung, um die semantische Genauigkeit und die Zustimmung der Stakeholdern zu gewährleisten. KI-generierte Anforderungen müssen ihre ursprüngliche Bedeutung beibehalten und gleichzeitig Qualitätskriterien erfüllen (siehe Kapitel 3.8 in [GleA2024]), um die Wertorientierung zu unterstützen (CPRE-Prinzip 1, [IREB2024]) und nicht nur die formale Erfüllung zu erreichen. In ähnlicher Weise erfordern von der KI erstellte visuelle Darstellungen wie Prototypen und UML-Diagramme eine Validierung, um zu bestätigen, dass sie sich auf einer angemessenen Abstraktionsebene befinden (siehe Kapitel 3.1.2 in [GleA2024]), mit dem beabsichtigten Zweck der modellbasierten Arbeitsprodukte übereinstimmen (siehe Kapitel 3.4 in [GleA2024]) und beispielsweise keine voreiligen Entwurfsannahmen enthalten. Wenn Anforderungsattribute und Metadaten mit Hilfe von KI zugewiesen werden, ist menschliches Urteilsvermögen erforderlich, um zu bestätigen, dass kritische Faktoren wie der Geschäftswert und der Genehmigungsstatus der Stakeholder korrekt bewertet wurden. Die vorgeschlagenen Zuweisungen müssen den Projektstatus genau widerspiegeln und ein effektives Anforderungsmanagement unterstützen (siehe Kapitel 6.5 in [GleA2024]) und sollten nicht zu irreführenden Informationen führen.

## 4.1 Risiken in AI4RE

Der Einsatz von KI im Bereich RE bietet erhebliche Vorteile, birgt aber auch spezifische Risiken, die sich auf die Genauigkeit, Relevanz und Vertrauenswürdigkeit der Ergebnisse auswirken können. Diese Risiken ergeben sich sowohl aus den technischen Beschränkungen von KI-Modellen als auch aus der Art und Weise, wie sie in RE-Kontexten angewendet werden. Das Verständnis der im Folgenden beschriebenen Risiken ist für Requirements Engineers von entscheidender Bedeutung, da sie dadurch in die Lage versetzt werden, potenzielle Probleme zu antizipieren, Strategien zur Risikominderung zu implementieren und die Integrität des Anforderungsprozesses zu wahren.

### Halluzinationen

LLMs können plausible, aber sachlich falsche Ergebnisse produzieren, indem sie Details oder Referenzen erfinden, die einfach nicht existieren. Im RE kann dies dazu führen, dass erfundene Normen, Aussagen von Stakeholdern oder Systemverhalten als gültige Anforderungen behandelt werden. Das Modell hat kein internes Konzept von "Wissen", so dass es nicht angeben kann, wann es rät oder nicht genügend Daten hat.

#### Beispiel:

Die KI zitiert "ISO 20258p" als Sicherheitsrichtlinie, obwohl es keine solche Norm gibt.

### Falsche Präzision

Manche KI-Ergebnisse erscheinen formal korrekt, sind aber semantisch bedeutungslos oder nicht überprüfbar. Technisch präzise formulierte Anforderungen können in der Praxis dennoch unprüfbar oder mehrdeutig sein.

#### Beispiel:

"Das System muss jederzeit optimal funktionieren" mag plausibel klingen, aber "optimal" ist nicht definiert und "jederzeit" ist unrealistisch und kann nicht getestet werden.

### Verstärkung von Voreingenommenheit

KI-Modelle können in ihren Trainingsdaten vorhandene Verzerrungen (Voreingenommenheiten, engl. Bias) verstärken oder ausbauen, was zu systematisch verzerrten Anforderungen oder Personas führt. Dies kann zu Ungerechtigkeit oder Diskriminierung in der Systemgestaltung führen, wenn es unentdeckt bleibt.

#### Beispiel:

In einem Projekt für HR-Systemanforderungen wird die KI mit der Erstellung von Personas für Stellenbewerber beauftragt. Auf der Grundlage voreingenommener historischer Rekrutierungsdaten beschreibt sie die Kandidaten für Führungspositionen durchweg als männlich, 35–50 Jahre alt und mit einem Hintergrund im Finanzwesen. Wenn diese Voreingenommenheit nicht korrigiert wird, besteht die Gefahr, dass diskriminierende Annahmen in automatisierte Einstellungsabläufe eingebettet werden und sogar Gesetze zur Chancengleichheit verletzt werden.

## Datenverlust

Sensible oder vertrauliche Informationen können bei der Nutzung von externen KI-Diensten ungewollt preisgegeben werden. Selbst anonymisierte Daten können identifizierbar sein, wenn sie mit anderen Quellen kombiniert werden.

### Beispiel:

In einem Gesundheitsprojekt werden anonymisierte Transkripte von Patientenbefragungen zur Zusammenfassung an einen Cloud-basierten KI-Dienst außerhalb der Europäischen Union<sup>1</sup> gesendet. Die zuständige Datenschutzbehörde stellt später fest, dass eindeutige Kombinationen von Symptomen und Behandlungsdaten in den Protokollen ausreichen, um einzelne Patienten zu identifizieren. Dies stellt einen Verstoß gegen die DSGVO [GDPR2016] dar und führt zu einer sehr hohen Geldstrafe.

## Urheberrechts-Verstöße

KI-Ausgaben können geschütztes oder vertrauliches Material aus Trainingsdaten oder früheren Aufforderungen wiedergeben, was rechtliche und vertragliche Risiken birgt.

### Beispiel:

Die KI erstellt eine Anforderungsspezifikation, die wortwörtlich den Text aus dem Produkthandbuch eines Wettbewerbers enthält.

## Veraltetes Wissen

Modelle, die auf der Grundlage historischer Daten trainiert wurden, können bereits veraltet sein, weil z. B. neuere Normen, Technologien oder Vorschriften fehlen. Dies ist eine statische Einschränkung, die von der Einführung an besteht und besonders in regulierten Bereichen kritisch ist, wo veraltete Anforderungen zur Nichtkonformität führen können.

### Beispiel:

Die KI empfiehlt eine Randbedingung, die ein Netzwerkprotokoll einbezieht, das seit zwei Jahren veraltet ist.

## Modell Drift

Im Laufe der Zeit kann die Genauigkeit der Ergebnisse eines KI-Systems aufgrund von sich entwickelndem Domänenwissen, Datenmustern oder Betriebsbedingungen abnehmen. Es handelt sich dabei um ein dynamisches Risiko, das während der Nutzung auftritt. Ohne Überwachung kann diese sog. Drift zu unauffälligen, aber signifikanten Fehlern führen.

---

<sup>1</sup> Die Beispiele in diesem Kapitel beziehen sich zwar auf EU-Verordnungen, dienen aber nur der Veranschaulichung und sollten an den rechtlichen Rahmen und den kulturellen Kontext des jeweiligen Projekts angepasst werden.

**Beispiel:**

Eine KI, die im Jahr 2023 für die Einhaltung von Bankvorschriften optimiert wurde, gibt nach Änderungen der Vorschriften im Jahr 2024 veraltete Empfehlungen ab.

### Überlauf des Context Windows

Wenn die Menge der Eingaben die Kontextkapazität eines Modells übersteigt, können frühere Informationen weggelassen werden, was zu unvollständigen oder inkonsistenten Ergebnissen führt.

**Beispiel:**

Während eines groß angelegten Workshops zur Anforderungserhebung verarbeitet das KI-Modell (mit einer Kontextgrenze von 16.000 Token) Besprechungsnotizen in Echtzeit. Frühe Äußerungen von Stakeholdern zu Sicherheitsanforderungen werden stillschweigend aus der Ausgabe gestrichen, weil sie die Kontextkapazität des Modells überschreiten, was zu unvollständigen Spezifikationen und dem Wegfall kritischer Sicherheitsfunktionen führt.

### Fehlausrichtung des Umfangs

Die KI kann Anforderungen generieren, die den Rahmen des Projekts sprengen oder den vereinbarten Zielen widersprechen, insbesondere wenn die Prompts vage oder unvollständig sind.

**Beispiel:**

Beim Entwurf von Anforderungen für eine mobile App beginnt die KI, Funktionen für eine Desktop-Version vorzuschlagen, die nie geplant war. Das Entwicklungsteam verbringt mehrere Tage damit, diese irrelevanten Features zu analysieren, bevor es feststellt, dass sie den Umfang sprengen und die vereinbarte Entwurfsphase verzögern.

### Übermäßiges Vertrauen in KI

Wenn man sich zu sehr auf KI-Ergebnisse ohne angemessene menschliche Validierung verlässt, kann dies dazu führen, dass fehlerhafte oder voreingenommene Anforderungen akzeptiert werden, was die Gesamtqualität verringert und das Projektrisiko erhöht.

**Beispiel:**

Ein Projektteam im Automobilssektor übernimmt KI-generierte Sicherheitsanforderungen für eine Funktion des autonomen Fahrens ohne angemessene menschliche Überprüfung. Monate später stellen die Prüfer bei einer behördlichen Inspektion fest, dass mehrere Anforderungen im Widerspruch zu vorgeschriebenen Sicherheitsstandards stehen, so dass ein kostspieliges Redesign erforderlich wird und sich die Produkteinführung verzögert.

## Intransparenz

Viele KI-Modelle funktionieren wie Blackboxen, so dass es schwierig ist, nachzuvollziehen, wie bestimmte Ergebnisse zustande gekommen sind. In RE kann dieser Mangel an Transparenz das Vertrauen der Stakeholder untergraben und die Validierung von Anforderungen erschweren.

### **Beispiel:**

Die KI schlägt vor, bestimmte Anforderungen nach Prioritäten zu ordnen, kann aber die Logik der Rangfolge nicht erklären, was die Überprüfung erschwert.

## Transparenzverpflichtungen

Regulierungen wie der EU AI Act [EUAI2026] sehen eine gesetzliche Verpflichtung zur Offenlegung von Inhalten vor, die von einer KI generiert wurden, auch wenn diese Regeln recht spezifisch sind und nicht für alle Ergebnisse gelten. Dadurch entsteht ein Compliance-Risiko, wenn Teams bestimmte Arten von KI-Output ohne die erforderlichen Kennzeichnungen oder Kontrollen veröffentlichen.

Für Textausgaben, wie sie im RE üblich sind, gilt die Verpflichtung, den Inhalt als KI-generiert zu kennzeichnen insbesondere dann, wenn der Text dazu bestimmt ist, die Öffentlichkeit über Angelegenheiten von öffentlichem Interesse zu informieren. Die meisten typischen RE-Artefakte, wie z. B. interne Spezifikationen, User Stories oder Backlog-Elemente, fallen nicht in diese Kategorie.

Eine Offenlegung ist auch dann nicht erforderlich, wenn der von der KI erzeugte Output von Menschen überprüft wurde und eine Person oder Organisation die redaktionelle Verantwortung für die Veröffentlichung übernimmt. In einem RE-Kontext entspricht dies dem Prinzip, dass ein Requirements Engineer, Product Owner oder ein anderer Stakeholder die Anforderungen immer validieren und übernehmen muss, bevor sie akzeptiert werden.

### **Beispiel:**

Ein Team setzt eine KI ein, um aus einer Reihe genehmigter Anforderungen einen veröffentlichten Bericht zu erstellen, in dem detailliert dargelegt wird, wie die neue Finanzsoftware mit den staatlichen Transparenzvorschriften übereinstimmt. Dieser Bericht ist von öffentlichem Interesse. Würde der Bericht automatisch veröffentlicht, ohne dass ein Compliance-Beauftragter oder ein Projektleiter ihn überprüft und die offizielle Verantwortung für seinen Inhalt übernommen hat, würde er wahrscheinlich gegen die Gesetze zur Transparenz von KI verstoßen. Die Organisation könnte rechtlich belangt werden, wenn sie KI-generierte Analysen zu einer öffentlichen Angelegenheit ohne die erforderliche Offenlegung oder menschliche Aufsicht vorlegt.

## 4.2 Zuständigkeiten in AI4RE

Die Einführung von KI in RE-Prozesse ersetzt den Requirements Engineer nicht, sondern *erweitert und vertieft* seine Rolle.

Der Requirements Engineer fungiert nun nicht nur als Ermittler, Dokumentator und Validierer von Anforderungen, sondern auch als *Lenker von KI-gestützten Prozessen*, der sicherstellt, dass die Ergebnisse relevant, vertrauenswürdig, erklärbar [DDOS2024] und mit den geltenden Regeln und ethischen Standards vereinbar sind.

Zu den wichtigsten Aspekten der erweiterten Rolle gehören:

### KI-Prozessdesigner, Integrator und technischer Kooperationspartner

Der Requirements Engineer trägt zur Auswahl von KI-Tools bei, definiert deren Integration in den RE-Prozess und arbeitet eng mit technischen Teams (z. B. KI-Entwicklern, Datenwissenschaftlern, IT) zusammen, um sicherzustellen, dass die Tools richtig konfiguriert, geschult und gewartet werden. Dazu gehört die Bewertung ihrer Eignung für bestimmte RE-Aufgaben (z. B. Klassifizierung, Zusammenfassung, Qualitätsprüfung) und die Anpassung der Eingabe-/Ausgabeformate an Modellierungs- und Dokumentationsstandards.

#### **Beispiel:**

Auswahl eines LLM zur Entfernung von Anforderungsduplikaten, Integration mit dem Repository des Teams und Festlegung von Regeln für die Kennzeichnung und Überprüfung von Duplikaten.

### Prompt-Designer und Qualitätsprüfer

Der Requirements Engineer erstellt und verfeinert Prompts, die die KI zu relevanten, präzisen und kontextbezogenen Ergebnissen führen, und bewertet diese Ergebnisse dann kritisch anhand von Domänenwissen, Stakeholder-Anforderungen und Qualitätskriterien. Dadurch wird sichergestellt, dass KI-Vorschläge niemals ohne menschliche Überprüfung angenommen werden.

#### **Beispiel:**

Entwurf einer Prompt-Vorlage für die Erstellung von Akzeptanzkriterien, die ausdrücklich Performance-Schwellwerte enthält, und Zurückweisung von KI-generierten Anforderungen, die nicht testbar sind.

### Risiko- und Compliance-Manager

Der Requirements Engineer identifiziert und entschärft rechtliche, ethische und betriebliche Risiken der KI-Nutzung – vom Schutz personenbezogener Daten und der Vermeidung von Voreingenommenheit bis hin zur Sicherstellung, dass die Ergebnisse den gesetzlichen und vertraglichen Anforderungen entsprechen. Dazu gehören die Wahl der Verarbeitungsoptionen (z. B. lokal oder in der Cloud), die Sicherstellung der Datenminimierung und das Erkennen von Modell Drift.

**Beispiel:**

Pseudonymisierung des Stakeholder-Feedbacks vor dem Hochladen in einen KI-Dienst, der außerhalb der Gerichtsbarkeit gehostet wird, und routinemäßige Überprüfung, ob die KI immer noch regelkonforme Vorschläge liefert.

### Umweltauswirkungen der Nutzung von KI

Generative KI wirkt sich auf die Umwelt aus, vor allem aufgrund des Energiebedarfs für das Training und den Betrieb großer Modelle. Zwar liegen viele dieser Faktoren außerhalb der direkten Kontrolle von Requirements Engineers, doch kann die Art und Weise, wie KI im Rahmen des RE-Prozesses ausgewählt und eingesetzt wird, den Energieverbrauch beeinflussen. Ein unreflektierter oder unnötig intensiver Einsatz von KI-Tools birgt daher Umwelt-, Reputations- und Governance-Risiken für Projekte und Organisationen.

**Beispiel:**

Ein Projektteam plant, ein LLM kontinuierlich im Hintergrund laufen zu lassen, um alle Anforderungen in Echtzeit zu überwachen und neu zu klassifizieren. Der Requirements Engineer stellt fest, dass dieses "always-on"-Nutzungsverhalten deutlich mehr Energie verbrauchen würde als ein planmäßiger Batch-Verarbeitungsansatz, bei geringem Zusatznutzen. Durch die Empfehlung, stattdessen eine Stapelverarbeitung durchzuführen, reduziert der Requirements Engineer die Umweltbelastung und erfüllt gleichzeitig die analytischen Anforderungen des Teams.

### Kommunikator und Vertrauensbildner für die Stakeholder

Vertrauen zu schaffen bedeutet nicht nur, KI-gestützte Ergebnisse weiterzugeben, sondern aktiv transparent zu machen, was erklärt werden kann und was nicht, und die Gründe für diese Grenzen zu nennen. Der Requirements Engineer stellt sicher, dass die Stakeholder transparent über den Einsatz von KI, ihre Rolle im RE-Prozess, ihre Vorteile und ihre Grenzen informiert werden.

Dazu gehört auch die Erleichterung des Lernens, der Verhandlung und der Konsensbildung zwischen den Stakeholdern. Diese menschenzentrierten Aktivitäten sind für jeden RE-Prozess unerlässlich und bedürfen besonderer Aufmerksamkeit, wenn Teile des Prozesses durch KI automatisiert oder beschleunigt werden.

Ein wichtiger Aspekt dieser Rolle ist es, KI-gestützte Ergebnisse erklärbar [DDOS2024] und nachvollziehbar zu machen und Raum für Diskussionen zu schaffen, wenn ethische Bedenken oder Vorbehalte gegen den Einsatz von KI aufkommen. Vertrauen aufzubauen bedeutet, diese Fragen ausdrücklich anzusprechen, anstatt sie zu vermeiden.

**Beispiel:**

In einem Workshop zur Anforderungserhebung erklärt der Requirements Engineer, welche Schritte KI-gestützt sind, erläutert die Gründe für die KI-generierten Vorschläge und weist auf etwaige Einschränkungen bei der Erklärbarkeit hin. Die Stakeholder sind eingeladen, Bedenken zu äußern, KI-gestützte Ergebnisse in Frage zu stellen und Strategien für den Umgang mit sensiblen Daten und möglichen KI-Einschränkungen im Projekt zu diskutieren.

### 4.3 Wichtige Erkenntnisse

KI verstärkt das kritische Denken im Bereich des RE, anstatt es zu ersetzen. Der Requirements Engineer muss Risiken erkennen und mindern, KI-Ergebnisse kritisch bewerten und sicherstellen, dass nur validierte Ergebnisse in die Anforderungs-Basislinie einfließen. Jenseits der Überwachung übernimmt der Requirements Engineer eine strategische Rolle: Er entwirft KI-gestützte Prozesse, stellt die Einhaltung von Vorschriften sicher und dokumentiert Entscheidungen für die Nachvollziehbarkeit. Technologische Kompetenz, Domänenwissen und menschliches Urteilsvermögen sind nach wie vor entscheidend, um sicherzustellen, dass KI Wert steigert und das Vertrauen der Stakeholder fördert.

## 5 Anwendungsfälle für KI im RE

Stufe: K2

Dauer: 2,5 Stunden

Ziel: Kandidaten können erklären, wie KI-Tools RE-Aktivitäten unterstützen können, ihre Grenzen beschreiben und die Notwendigkeit der Validierung von KI-generierten Ergebnissen erläutern.

Begriffe: Ermittlung (von Anforderungen), Dokumentation, Validierung, Management, Exploration, Transkription, Extraktion, Systemkontext, Formulierung, Transformation, semantische Genauigkeit, Qualitätskriterien

### Lernziele

LZ 5.1 Nennen typischer Anwendungsfälle für KI bei der Ermittlung, Dokumentation, Validierung und Verwaltung von Anforderungen. (K1)

LZ 5.2 Erläutern der Bedeutung der menschlichen Kuratation von KI-Ergebnissen im RE und beschreiben der Rolle des Requirements Engineers bei der Sicherstellung valider Ergebnisse. (K2)

Das Aufkommen von KI-Technologien eröffnet neue Möglichkeiten zur Unterstützung von Requirements Engineers bei ihren Kernaufgaben. Es ist jedoch von entscheidender Bedeutung zu verstehen, dass KI bei der Arbeit im RE als unterstützendes Werkzeug dient und nicht als Ersatz für menschliches Fachwissen und Urteilsvermögen.

KI-Technologien können Requirements Engineers bei den vier zentralen RE-Aktivitäten unterstützen: Ermittlung, Dokumentation, Validierung und Management von Anforderungen [IREB2024]. Sie bieten das Potenzial, Aufgaben zu beschleunigen, Perspektiven zu erweitern und Konsistenzprüfungen zu verbessern. In diesem Kapitel werden anschauliche Anwendungsfälle für jede Aktivität untersucht, wobei auch betont wird, dass KI-generierte Ergebnisse einer sorgfältigen menschlichen Überprüfung unterzogen werden müssen. Letztendlich bleibt der Requirements Engineer dafür verantwortlich, dass diese Ergebnisse zweckmäßig und methodisch fundiert sind und mit den Projektzielen und Qualitätsstandards übereinstimmen.

Die korrekte Anwendung von KI im Bereich des RE erfordert ein gründliches Verständnis sowohl der Möglichkeiten als auch der Grenzen dieser Technologien. Requirements Engineers müssen die Kompetenz entwickeln, KI-Ergebnisse kritisch zu bewerten und zu erkennen, wo sie einen Mehrwert schaffen, aber auch, wo sie Risiken oder Ungenauigkeiten mit sich bringen können. Bei der praktischen Arbeit mit KI-Tools sind alle in den vorangegangenen Kapiteln dieses Dokuments festgelegten Grundsätze, Praktiken und Qualitätskriterien weiterhin uneingeschränkt anwendbar und müssen sorgfältig berücksichtigt werden. Umfassende Informationen zu grundlegenden Konzepten und Praktiken des RE finden Sie im CPRE Foundation Level Lehrplan [IREB2024] und im dazugehörigen Handbuch [GleA2024].

Weitere praktische Beispiele für die Umsetzung von KI in der RE-Praxis finden Sie in der Broschüre der Special Interest Group #AIREB [IREB2025] und im AI4RE Prompt Guide [IREB2026].

## 5.1 Anforderungsermittlung

Die Anforderungsermittlung ist das Herzstück von RE. Dazu gehört das Erheben, Interpretieren und Strukturieren von Informationen aus Quellen wie Stakeholdern, Dokumenten und Systemen, um zu verstehen, was ein System erreichen muss. Dieser Prozess beschränkt sich nicht auf das Stellen von Fragen: Er erfordert aktives Zuhören, kontextbezogenes Verständnis und die Fähigkeit, informellen Input in formale, umsetzbare Anforderungen zu übersetzen.

Es gibt drei Bereiche innerhalb der Ermittlung, in denen KI wirksame Unterstützung bieten kann:

- **Erkundung:** Bevor Anforderungen erhoben werden können, müssen Requirements Engineers zunächst die Domäne, die Ziele, die Stakeholder und die Randbedingungen verstehen. KI-Tools können dies unterstützen, indem sie Domänenwissen zusammenfassen, Stakeholdergruppen identifizieren und bei der Vorbereitung strukturierter Interviewleitfäden und Workshops helfen. Diese erste Erkundung stellt sicher, dass die Ermittlungsaufwände fundiert und zielgerichtet sind.
- **Transkription:** Interviews von Stakeholdern und Workshops liefern reichhaltige, qualitative Daten, die genau erfasst werden müssen. KI-gestützte Transkription wandelt gesprochene Dialoge und visuelle Artefakte in strukturierte, durchsuchbare Formate um und reduziert gleichzeitig den manuellen Aufwand und verbessert die Nachvollziehbarkeit. Dies trägt dazu bei, dass wertvolle Erkenntnisse aufgezeichnet werden und für weitere Analysen zur Verfügung stehen.
- **Extraktion:** Sobald der Input der Stakeholder dokumentiert ist, besteht der nächste Schritt darin, klare und umsetzbare Anforderungen zu extrahieren. KI kann bei der Identifizierung und Kategorisierung von funktionalen Anforderungen, Qualitätsattributen und Randbedingungen anhand von Rohdaten wie Besprechungsnotizen, E-Mails oder Protokollen helfen. Die Gruppierung dieser Anforderungen nach Themen- oder Funktionsbereichen hilft weiter bei der Klärung von Prioritäten und Abhängigkeiten.

Zusammen bilden diese Aktivitäten einen umfassenden Ansatz für die Ermittlung, bei dem menschliches Fachwissen mit KI-Unterstützung kombiniert wird, um die Effizienz, die Abdeckung und die Qualität zu verbessern.

### 5.1.1 Erkundung neuer Domänen

Die explorative Phase des RE schafft die Grundlage für alle nachfolgenden RE-Aktivitäten, indem ein umfassendes Verständnis der Domäne, der Stakeholder und des betrieblichen Kontextes entwickelt wird, bevor die formale Anforderungserhebung beginnt. KI kann diese kritische Phase beschleunigen, indem sie Rechercheaktivitäten unterstützt, analytische Perspektiven erweitert und Ansätze zur Einbindung von Stakeholdern strukturiert. Ihre Ergebnisse dienen jedoch nur als Ausgangspunkt und müssen durch menschliche Expertise und direkte Interaktion mit den Stakeholdern validiert werden.

KI kann interne Unterlagen und öffentlich zugängliche Quellen zusammenfassen, um einen Überblick über die Ziele, die Terminologie, die Herausforderungen und den rechtlichen Kontext zu geben und gleichzeitig die organisatorischen Strukturen für potenzielle Stakeholder zu analysieren, die sonst vielleicht übersehen würden. Sie kann strukturierte Interviewleitfäden mit offenen Fragen erstellen, die auf Sondierungsgespräche zugeschnitten sind, und bei der Organisation von Anforderungsworkshops helfen, indem sie Schlüsselthemen, Stakeholder-Rollen und potenzielle Konfliktbereiche oder Übereinstimmungen identifiziert. Diese Fähigkeiten zeigen, wie KI die explorative Arbeit von Requirements Engineers verbessern kann, indem sie die Effizienz der Recherche und die Abdeckung der Stakeholder erhöht. Der wahre Wert hängt jedoch davon ab, dass diese Instrumente als Katalysatoren für eine tiefere Untersuchung eingesetzt werden und nicht als Ersatz für kritisches Denken und die Einbeziehung von Stakeholdern, die für die Schaffung eines gemeinsamen Verständnisses unerlässlich sind (CPRE-Prinzip 3, [IREB2024]).

### Identifizierung des Systemkontextes

KI kann die Identifizierung der im CPRE Foundation Level Syllabus beschriebenen Elemente des Systemkontextes erheblich verbessern (CPRE-Prinzip 4, [IREB2024] und [GleA2024]). Wenn Artefakte wie Stakeholder-Listen, Diagramme der Systemlandschaft, Prozessbeschreibungen oder Organigramme verfügbar sind, kann KI die relevanten Informationen extrahieren und diese nach den fünf Kernaspekten (Stakeholder, Dokumente, Systeme, Prozesse und Ereignisse) klassifizieren sowie Beziehungen und Abhängigkeiten zwischen den Quellen aufdecken. Dies unterstützt die Requirements Engineers dabei, sich ein umfassendes Bild von den System- und Kontextgrenzen zu machen und verringert das Risiko, kritische Elemente zu übersehen.

Alternativ kann KI strukturierte Fragelisten und Checklisten erstellen, die speziell darauf ausgerichtet sind, von den Stakeholdern Informationen zu jedem der fünf Aspekte des Systemkontexts zu erhalten. Diese KI-generierten Instrumente können als umfassende Erhebungsinstrumente bei Befragungen von Stakeholdern oder in Workshops dienen und gewährleisten eine systematische Erfassung aller Kontextdimensionen. Die KI kann Fragen auf bestimmte Domänen oder Systemtypen zuschneiden und dabei bewährte Erhebungstechniken für Anforderungen (siehe Kapitel 4.2 in [GleA2024]) einbeziehen, wobei die Konsistenz mit den etablierten Qualitätskriterien für Arbeitsprodukte (siehe Kapitel 3.8 in [GleA2024]) gewahrt bleibt.

### Gemeinsames Verständnis schaffen

Das Schaffen eines gemeinsamen Verständnisses (CPRE-Prinzip 3, [IREB2024]) hängt im Wesentlichen von der einheitlichen Verwendung der Terminologie zwischen allen Projektbeteiligten ab. Glossare (siehe Kapitel 3.5 in [GleA2024]) dienen als zentrale Definitionssammlungen, die das Risiko von Fehlinterpretationen vermindern und sicherstellen, dass Stakeholder, Requirements Engineers und Entwickler auf der gleichen konzeptionellen Grundlage arbeiten.

KI kann die Erstellung und Pflege umfassender Glossare erheblich beschleunigen, indem sie vorhandene Projektdokumentation, Kommunikation mit Stakeholdern und bereichsspezifische Materialien analysiert, um Begriffe zu identifizieren, die definiert werden müssen. KI kann eine inkonsistente Verwendung von Begriffen in verschiedenen Dokumenten erkennen, potenzielle Homonyme (gleiche Begriffe für verschiedene Konzepte) kennzeichnen und Synonyme vorschlagen, die standardisiert werden sollten. Darüber hinaus kann KI erste Definitionen vorschlagen, die auf einer kontextuellen Analyse der Verwendung von Begriffen in der Projektumgebung beruhen, so dass Requirements Engineers einen soliden Ausgangspunkt für die Entwicklung eines Glossars erhalten.

### 5.1.2 Kommunikation mit Stakeholdern transkribieren

Eine wirksame Kommunikation mit den Stakeholdern bildet die Grundlage für erfolgreiches RE. Das Erfassen und Bewahren von Erkenntnissen aus Interviews und Workshops kann jedoch eine Herausforderung darstellen, wenn wertvolle Informationen aus informellen, unstrukturierten Formaten hervorgehen. Die Umwandlung von gesprochenen Dialogen und visuellen Artefakten in eine strukturierte, bearbeitbare Dokumentation trägt dazu bei, dass wichtige Erkenntnisse erhalten bleiben und für die Analyse, Validierung und Integration in formale RE-Prozesse verfügbar sind. Dies unterstützt die Verfolgbarkeit und die Abstimmung im Team, die für den Erfolg eines Projekts unerlässlich sind.

KI-basierte Transkriptionstools können diesen Konvertierungsprozess erheblich verbessern, indem sie aus den Aufzeichnungen der Stakeholder-Interviews Transkripte erstellen, den Rohtext bereinigen und die wichtigsten Diskussionspunkte zusammenfassen.

Dies kann den Übergang vom Gespräch zur formellen Anforderungsdokumentation beschleunigen. Für kollaborative Workshops kann KI Whiteboard-Inhalte, Haftnotizen und Skizzen in strukturierte Formate transkribieren, die den Austausch und die kollaborative Verfeinerung erleichtern, was besonders in agilen oder verteilten Entwicklungsumgebungen nützlich ist. Diese Fähigkeiten unterstützen die Erstellung von transparenten und nachverfolgbaren Anforderungen und reduzieren den manuellen Dokumentationsaufwand. Gleichzeitig müssen die von der KI generierten Ergebnisse sorgfältig von Requirements Engineers überprüft und verfeinert werden, um Genauigkeit, Vollständigkeit und Übereinstimmung mit den Absichten der Stakeholder und den Projektzielen sicherzustellen.

#### Verarbeitung von Stakeholder-Interviews und Audioaufnahmen

Generative KI [FHJZ2024] kann die Effizienz der Anforderungserhebung durch die automatische Transkription von Stakeholder-Interviews substantiell steigern, indem sie Audioaufnahmen in strukturierten Text umwandelt, der eine systematische Analyse und Dokumentation erleichtert. Dank dieser Fähigkeit können sich Requirements Engineers während der Ermittlungssitzungen voll und ganz auf die Interviewdynamik und das Engagement der Stakeholder konzentrieren (siehe Kapitel 4.2 in [GleA2024]), anstatt ihre Aufmerksamkeit zwischen aktivem Zuhören und Notizen machen aufzuteilen.

KI-Transkriptionsdienste können verschiedene Sprecher identifizieren, Gespräche mit einem Zeitstempel versehen und sogar potenzielle Anforderung oder Schlüsselterminologie hervorheben, die in Projektglossare aufgenommen werden sollten (siehe Kapitel 3.5 in [GleA2024]).

### Analyse transkribierter Interviews

Darüber hinaus kann KI transkribierte Interviews analysieren, um vorläufige Anforderungen zu extrahieren, Bedenken der Stakeholder zu identifizieren und Folgefragen vorzuschlagen, die sich an systematischen Erhebungsmethoden orientieren, wie z.B. den in [GleA2024] und Kapitel 0 beschriebenen Befragungsmethoden oder Kollaborationstechniken.

## 5.1.3 Extraktion von Anforderungen aus verschiedenen Quellen

### Analyse von E-Mails, Chats, Besprechungsprotokollen und Diskussionsbeiträgen

KI kann systematisch E-Mails, Chats, Besprechungsprotokolle und Kollaborations-Threads analysieren, um anforderungsrelevante Inhalte zu ermitteln, die in solchen informellen Kanälen oft übersehen werden. Diese automatische Extraktion stärkt die Identifizierung von Anforderungsquellen (siehe Kapitel 4.1 in [GleA2024]), indem sie implizite Anforderungen, Bedenken der Stakeholder und kontextuelle Hinweise aus der laufenden Kommunikation erfasst. KI kann Elemente nach Typ klassifizieren (funktionale Anforderungen, Qualitätsanforderungen und Randbedingungen), Konflikte zwischen formellen und informellen Aussagen erkennen und Inkonsistenzen mit offiziellen Dokumenten markieren. Dies ist besonders in agilen Umgebungen wertvoll, in denen sich Anforderungen durch kontinuierliche Interaktion mit den Stakeholdern und informelles Feedback weiterentwickeln.

### Analyse vorhandener Dokumente und Querverweise zwischen Anforderungen

Darüber hinaus kann generative KI [FHJZ2024] vorhandene Dokumente aus verschiedenen Quellen systematisch analysieren (siehe Kapitel 4.1 in [GleA2024]), darunter Geschäftsprozessdokumente, regulatorische Dokumente, Spezifikationen von Altsystemen und Marktanalyseberichte.

Sie kann dann sowohl explizite als auch implizite Anforderungen ermitteln, die in diesem Material enthalten sein können. KI kann mehrere Dokumentenformate gleichzeitig verarbeiten und potenzielle funktionale Anforderungen, Qualitätsanforderungen und Randbedingungen extrahieren und sie gemäß dem in [GleA2024], Kapitel 1.1, beschriebenen Klassifizierungsrahmen für Anforderungen kategorisieren. KI kann auch extrahierte Anforderungen mit dem Projektumfang und den Systemgrenzen abgleichen, um ihre Relevanz sicherzustellen. Außerdem kann sie Beziehungen zwischen Anforderungen in verschiedenen Quelldokumenten erkennen. Diese automatisierte Extraktion unterstützt eine umfassende Abdeckung von Anforderungsquellen, die über den direkten Stakeholder-Input hinausgehen, und trägt dazu bei, dass das dokumentierte organisatorische Wissen und die gesetzlichen Verpflichtungen in der Anforderungsspezifikation korrekt erfasst werden.

## Analyse von Mustern und Vorschläge für weitere Untersuchungen

KI kann Muster im extrahierten Stakeholder-Input analysieren, um potenzielle Anforderungslücken, implizite Annahmen oder übersehene Systemaspekte zu identifizieren, die die Stakeholder vielleicht für selbstverständlich hielten, aber nicht explizit formuliert haben. Durch den Vergleich von Stakeholder-Aussagen mit umfassenden Anforderungsframeworks und domänenspezifischen Checklisten kann KI Bereiche für weitere Untersuchungen vorschlagen und Fragen stellen, die dabei helfen, unbewusste Anforderungen zu ermitteln, wie im Kano-Modell beschrieben (siehe Kapitel 4.2 in [GleA2024]). KI kann auch Unstimmigkeiten zwischen verschiedenen Stakeholder-Perspektiven aufdecken: Diese können auf fehlende Anforderungen oder ungelöste Konflikte hinweisen, die durch Konfliktlösungstechniken behoben werden müssen (siehe Kapitel 4.3, in [GleA2024]). Der Requirements Engineer muss jedoch alle von der KI generierten Erkenntnisse durch systematische Einbeziehung der Stakeholder und Verifizierungsprozesse validieren. Sie müssen sicherstellen, dass es sich bei den festgestellten Lücken um echte Anforderungen und nicht um Annahmen der KI handelt; dass diese Anforderungen tatsächlich den Wünschen und Bedürfnissen der Stakeholder entsprechen (CPRE-Prinzip 2, [IREB2024]) und nicht nur eine theoretische Vollständigkeit darstellen.

## 5.2 Dokumentation

Nach der Erkundung der Domäne, der Transkription und Extraktion von Anforderungen besteht die nächste Herausforderung darin, diese präzise und klar auszudrücken. Dies beinhaltet zwei eng miteinander verbundene Aktivitäten:

- **Formulierung:** Die Formulierung konzentriert sich auf die Strukturierung von Anforderungen unter Verwendung standardisierter Formate und Qualitätskriterien. Ob in Form von Statements gemäß IEEE, Use Cases oder User Stories, gut formulierte Anforderungen verbessern die Klarheit, Vergleichbarkeit und Verfolgbarkeit. KI-Tools können bei der Auswahl eines geeigneten Formats, bei der Verfeinerung der Sprache und bei der Zerlegung komplexer Anforderungen in umsetzbare Teilelemente helfen. Darüber hinaus können von der KI erstellte Synonymtabellen dazu beitragen, die Konsistenz der Dokumentation zu wahren, insbesondere in Domänen mit spezieller Terminologie.

- **Transformation:** Durch die Transformation wird sichergestellt, dass die Anforderungen für verschiedene Stakeholdergruppen und technische Kontexte verständlich und nutzbar sind. Dazu gehören die Umformulierung von Anforderungen in einfache Sprache, die Anpassung von Zusammenfassungen an bestimmte Zielgruppen und die Konvertierung zwischen Formaten, z. B. die Erstellung von Modellen aus Text oder umgekehrt. KI unterstützt auch die Erstellung von Personas auf der Grundlage von Stakeholder-Input, die Generierung von Testfällen auf der Grundlage von Anforderungen und sogar den Vergleich von Spezifikationen mit Quellcode zur Validierung der Implementierung. Umgekehrt kann KI dabei helfen, Anforderungen aus bestehenden Codebasen zu extrahieren, um die Dokumentation und Modernisierungsbemühungen zu unterstützen.

Gemeinsam überbrücken Formulierung und Transformation die Kluft zwischen technischer Präzision und dem Verständnis der Stakeholder. Sie stellen sicher, dass Anforderungen nicht nur richtig und vollständig sind, sondern auch nutzbar, testbar und auf die Unternehmensziele abgestimmt.

### 5.2.1 Formulierung von Anforderungen

In der Formulierungsphase werden die Rohdaten der Stakeholder in klare, strukturierte Anforderungsstatements umgewandelt, die sich für die Umsetzung und Validierung eignen. KI-Tools können diesen Prozess erheblich verbessern, indem sie die Standardisierung unterstützen, die terminologische Konsistenz wahren und komplexe Anforderungen in handhabbare Komponenten zerlegen, die mit den etablierten RE-Praktiken übereinstimmen.

#### Anwendung von standardisierten IEEE-Vorlagen und Satzschablonen

Generative KI [FHJZ2024] kann Requirements Engineers bei der Verwendung von standardisierten IEEE-Vorlagen und Satzschablonen (siehe Kapitel 3.3 in [GleA2024]) unterstützen, indem sie Anforderungsstatements automatisch nach festgelegten syntaktischen Mustern und Formatierungsrichtlinien strukturiert. KI kann informellen Stakeholder-Input oder vorläufige Anforderungsbeschreibungen analysieren und in korrekt formatierte Einzelanforderungen umwandeln, die IEEE-Standards wie ISO/IEC/IEEE 29148 folgen, was eine konsistente Struktur gewährleistet und die Gesamtqualität der Spezifikation verbessert.

#### Häufigen Fallstricke vermeiden

KI kann geeignete Konditionalsätze vorschlagen, messbare Kriterien spezifizieren und beim technischen Schreiben (siehe Kapitel 3.2 in [GleA2024]) helfen, häufige Fallstricke wie Passiv, Universalquantoren oder unvollständige Beschreibungen zu vermeiden. Dank dieser automatischen Formatierungsunterstützung können sich die Requirements Engineers auf die Validierung des Inhalts und die Abstimmung mit den Stakeholdern konzentrieren, anstatt Zeit mit Stilkorrekturen zu verbringen. Das stellt sicher, dass alle Anforderungen den etablierten Vorlagenrichtlinien entsprechen, die Klarheit und Konsistenz in der gesamten Spezifikation fördern.

### Formulierung des Stakeholder-Inputs als User Stories

KI kann bei der Formulierung von Stakeholder-Eingaben in Form von gut strukturierten User Stories helfen, die bewährten Formulierungsvorlagen folgen, wie z. B. dem weit verbreiteten Format "Als [Benutzertyp] möchte ich [Ziel], damit [Nutzen]" ([GleA2024], Kapitel 3.3). KI kann Beschreibungen von Stakeholdern, Interviewprotokolle oder informelle Anforderungsstatements analysieren, um Benutzerrollen, gewünschte Funktionen und den Geschäftswert zu ermitteln, und diese Informationen dann gemäß den Konventionen für User Stories strukturieren.

### Vorschlagen von Akzeptanzkriterien

KI kann auch geeignete Akzeptanzkriterien vorschlagen, fehlende Story-Elemente identifizieren und sicherstellen, dass sich die User Stories auf den Wert für den Nutzer und nicht auf Implementierungsdetails konzentrieren. Diese Fähigkeit erweist sich als besonders wertvoll in agilen Entwicklungskontexten, in denen User Stories als primäre Anforderungsartefakte in Product Backlogs dienen (siehe Kapitel 3.6 in [GleA2024]). Auf diese Weise trägt KI dazu bei, dass das Format und die Vollständigkeit der User Stories in großen Katalogen konsistent bleiben.

### Erstellung von Domänenglossaren und Synonymtabellen

Durch die Unterstützung der Erstellung von Domänenglossaren und Synonymtabellen trägt KI zur Klärung der domänenspezifischen Terminologie und zur Förderung eines konsistenten Sprachgebrauchs in den Teams bei, wodurch das Problem der Mehrdeutigkeit natürlicher sprachlicher Arbeitsprodukte gelöst wird (siehe Kapitel 3.2 in [GleA2024]).

### Anforderungen zerlegen

Darüber hinaus kann KI logische Unterteilungen innerhalb von High-Level-Anforderungen identifizieren und sie in umsetzbare Teilanforderungen, Spezifikationen auf Systemebene oder Akzeptanzkriterien zerlegen, die die Ausrichtung der Implementierung auf die Unternehmensziele erleichtern. Diese Formulierungsarbeit stützt sich auf das während der Explorationsaktivitäten erarbeitete Domänenverständnis, auf dokumentierte Erkenntnisse aus Transkriptionsprozessen und auf strukturierte Inhalte aus Extraktionsphasen. Es stellt sicher, dass Anforderungen in Formaten ausgedrückt werden, die eine effektive Zusammenarbeit, Entwicklung und Validierung während des gesamten RE-Prozesses unterstützen.

## 5.2.2 Umwandlung von Anforderungen in andere Darstellungen

Im modernen RE geht es bei der Transformation um mehr als nur um die Übersetzung von Text aus einer Sprache in eine andere: Es geht um die Transformation von Anforderungen über Formate und Stakeholder-Perspektiven hinweg. Da Systeme immer komplexer und die Teams immer interdisziplinärer werden, wird die Fähigkeit, Anforderungen effektiv zu transformieren, für Klarheit, Abstimmung und Verfolgbarkeit unerlässlich.

## Umwandlung natürlichsprachlicher Anforderungen in visuelle Modelle

KI-Tools unterstützen Requirements Engineers, indem sie einen fließenden Übergang zwischen Darstellungen und Stakeholder-Anforderungen ermöglichen. Sie können Anforderungen von natürlicher Sprache in visuelle Modelle wie UML- oder BPMN-Diagramme umwandeln (siehe Kapitel 3.4 in [GleA2024]), so dass Systemstruktur, Arbeitsabläufe und Beziehungen leichter zu verstehen sind als über reinen Text. Umgekehrt kann KI visuelle Modelle in strukturiertem Text beschreiben, um die Zugänglichkeit für nichttechnische Stakeholder zu verbessern und die Dokumentation zu unterstützen. Neben allgemeinen Diagrammen kann KI auch spezifische UML-Artefakte wie Klassen-, Aktivitäts- oder Use-Case-Diagramme sowie BPMN-Workflows generieren, die Entscheidungen und Prozesse visualisieren, wobei der Schwerpunkt stets auf der Spezifikation von Anforderungen und nicht auf dem detaillierten Design liegt.

## Erstellen von Prototypen

Generative KI [FHJZ2024] kann Requirements Engineers bei der Erstellung explorativer Prototypen unterstützen (siehe Kapitel 3.7 in [GleA2024]), indem sie automatisch Wireframes, Mock-ups oder sogar funktionale Prototypen auf der Grundlage von Anforderungsspezifikationen oder User Stories generiert. KI kann funktionale Anforderungen und Beschreibungen der Benutzeroberfläche analysieren, um visuelle Darstellungen zu erstellen, die den Stakeholdern helfen, das vorgeschlagene Systemverhalten besser zu verstehen und die Anforderungen anhand konkreter Beispiele zu validieren. KI kann Prototypen in verschiedenen Detaillierungsgraden erstellen, von einfachen Wireframes zur frühen Konzeptvalidierung bis hin zu anspruchsvolleren Mock-ups, die Abläufe der Benutzerinteraktion und Schnittstellenkonzepte demonstrieren. Diese automatisierte Prototypenerstellung ermöglicht eine schnelle Iteration und das Sammeln von Stakeholder-Feedback und unterstützt den Validierungsprozess (siehe Kapitel 4.4 in [GleA2024]), indem sie greifbare Darstellungen abstrakter Anforderungen liefert, die von den Stakeholdern bewertet und verfeinert werden können.

## Erstellen dynamischer visueller Inhalte

KI bietet vielfältige weitere Anwendungen zur Verbesserung der Kommunikation und Validierung im Bereich des RE, die über die Kernaktivitäten des RE hinausgehen. Diese Fähigkeiten können bei richtigem Einsatz das Engagement der Stakeholder und die Qualität der Anforderungen verbessern. Von der Umwandlung der Anforderungsdokumentation in dynamische visuelle Inhalte wie Erklärvideos über die Übersetzung technischer Anforderungen in eine vereinfachte Sprache für nichttechnische Stakeholder bis hin zur Erstellung gezielter Zusammenfassungen für bestimmte Zielgruppen (z. B. Führungskräfte, Endbenutzer, Entwickler) kann KI dazu beitragen, dass die Informationen für die Zielgruppe relevant sind und gleichzeitig die Konsistenz der Spezifikationen gewahrt bleibt.

### Erstellen von Benutzer-Personas

KI-Technologie kann den Input der Stakeholder synthetisieren, um Benutzer-Personas zu erstellen, die die traditionellen Methoden zur Identifizierung der Stakeholder ergänzen (siehe Kapitel 4.1 in [GleA2024]).

### Ableiten von Testfällen

KI kann Testfälle aus Anforderungen ableiten, um Validierungsaktivitäten zu unterstützen (siehe Kapitel 4.4 in [GleA2024]).

### Analyse von Legacy-Code

KI kann Legacy-Code analysieren, um implizite Anforderungen für die Dokumentation zu extrahieren.

### Vergleich von Spezifikationen

KI kann auch Spezifikationen mit implementiertem Code vergleichen, um die Einhaltung von Vorschriften zu gewährleisten oder Diskrepanzen zu erkennen, was besonders in regulierten Branchen, die eine strenge Rückverfolgbarkeit von Anforderungen verlangen, von großem Nutzen ist.

## 5.3 Validierung

Generative KI [FHJZ2024] stößt bei der Erstellung von Anforderungen auf natürliche Grenzen, die Requirements Engineers erkennen und berücksichtigen müssen. KI kann Anforderungen generieren, die syntaktisch korrekt zu sein scheinen und etablierten Vorlagen folgen (siehe Kapitel 3.3 in [GleA2024]), die aber in Wirklichkeit nicht den Wünschen und Bedürfnissen der Stakeholder entsprechen (CPRE-Prinzip 2, [IREB2024], Kapitel 2) oder die nicht im tatsächlichen Systemkontext verankert sind (CPRE-Prinzip 4, [IREB2024]). Darüber hinaus mangelt es KI-generierten Anforderungen oft an der notwendigen Präzision und Überprüfbarkeit, die qualitativ hochwertige Anforderungen kennzeichnen, wodurch Mehrdeutigkeiten oder unvollständige Spezifikationen entstehen können, die grundlegende Qualitätskriterien verletzen. Requirements Engineers müssen auf KI-Ergebnisse achten, die zwar umfassend zu sein scheinen, aber in Wirklichkeit kritische funktionale Anforderungen, Qualitätsanforderungen oder Randbedingungen nicht berücksichtigen, oder die Lösungen vorschlagen, anstatt die tatsächlichen Anforderungen gemäß dem Problem-Anforderung-Lösungs-Prinzip (CPRE-Prinzip 5, [IREB2024]) zu erfassen.

### Wahrung der Konsistenz

KI kann Requirements Engineers bei der Wahrung der Konsistenz von Anforderungsspezifikationen und Product Backlogs unterstützen, indem sie automatisch Widersprüche, sich überschneidende Funktionen und terminologische Inkonsistenzen erkennt, die im Laufe der Entwicklung von Arbeitsprodukten entstehen können. KI kann Anforderungen mit etablierten Glossaren abgleichen (siehe Kapitel 3.5 in [GleA2024]) und Fälle identifizieren, in denen dieselben Konzepte mit unterschiedlichen Begriffen beschrieben werden.

Dies unterstützt die einheitliche Verwendung der Terminologie, die für ein gemeinsames Verständnis unerlässlich ist (CPRE-Prinzip 3, [IREB2024]). KI-Tools können auch Abhängigkeiten zwischen Anforderungen analysieren und potenzielle Konflikte aufzeigen, die die Gesamtkohärenz der Spezifikation beeinträchtigen könnten. Das Sicherstellen echter Konsistenz erfordert jedoch mehr als nur einen automatischen Musterabgleich, sondern auch ein Verständnis der zugrunde liegenden Geschäftslogik, der Prioritäten der Stakeholder und der Systemarchitektur. Der Requirements Engineer muss validieren, dass die von der KI erkannten Inkonsistenzen echte Probleme und keine akzeptablen Variationen darstellen, und er muss sicherstellen, dass die von der KI vorgeschlagenen Konsistenzverbesserungen mit den Projektzielen übereinstimmen und nicht versehentlich neue Konflikte einführen oder die Angemessenheit einzelner Anforderungen beeinträchtigen.

### Wertorientierung, Angemessenheit und Notwendigkeit

Die Validierung von KI-generierten Anforderungen erfordert eine strenge Bewertung nach dem Grundsatz der Wertorientierung (CPRE-Prinzip 1, [IREB2024]), um sicherzustellen, dass jede vorgeschlagene Anforderung sinnvoll zur Erfüllung der Wünsche und Bedürfnisse der Stakeholder beiträgt und nicht nur isoliert sinnvoll erscheint. Requirements Engineers müssen bewerten, ob die von der KI generierten Anforderungen wirklich notwendig sind, indem sie sie bis zur Quelle der Stakeholder zurückverfolgen (siehe Kapitel 4.1 in [GleA2024]) und ihre Übereinstimmung mit den identifizierten Geschäfts- und Benutzerzielen überprüfen. Wie bei von Menschen erstellten Anforderungen sollte der Schwerpunkt auf Angemessenheit und Notwendigkeit liegen (siehe Kapitel 3.8 in [GleA2024]): Es ist zu prüfen, ob die KI keine Anforderungen eingeführt hat, die den tatsächlichen Umfang der Bedürfnisse der Stakeholder oder der Projektbeschränkungen überschreiten.

### Abstraktions-Ebene

Der Requirements Engineer muss auch beurteilen, ob die von der KI generierten Anforderungen zu den gewählten Abstraktionsebenen (siehe Kapitel 3.1.2 in [GleA2024]) passen und einen für die aktuelle Projektphase angemessenen Detaillierungsgrad aufweisen (siehe Kapitel 3.1.3 in [GleA2024]). Es sollte sichergestellt werden, dass die Anforderungen dem beabsichtigten Zweck dienen und nicht zu unnötiger Komplexität oder schleichender Ausweitung des Umfangs führen, die den Projekterfolg gefährden könnten.

## 5.4 Management

KI kann zwei wichtige Aufgaben des Anforderungsmanagements unterstützen: die *Zuordnung von Attributen zu Anforderungen* und die *Priorisierung von Anforderungen*.

### 5.4.1 Zuweisung von Attributen zu Anforderungen

KI kann die Zuordnung von Attributen zu Anforderungen optimieren, indem sie deren Inhalt analysiert und Werte auf der Grundlage etablierter Muster und projektspezifischer Kriterien vorschlägt. Wie in [GleA2024] Kapitel 6.5 beschrieben, dokumentieren Attribute wichtige Metadaten, die es den Stakeholdern ermöglichen, während des gesamten Projektlebenszyklus auf relevante Informationen zuzugreifen. Durch Textanalyse und Vergleich mit ähnlichen Anforderungen aus aktuellen oder früheren Projekten kann KI auch Werte für gängige Attribute wie Priorität, Komplexität, Quelle oder verantwortliche Person vorschlagen. Außerdem können fehlende Attributzuordnungen identifiziert und die Vollständigkeit über große Anforderungsmengen hinweg sichergestellt werden, wodurch die für ein effektives Anforderungsmanagement unerlässliche systematische Dokumentationspraxis unterstützt wird (siehe Kapitel 6.1 in [GleA2024]).

### 5.4.2 Priorisierung von Anforderungen

Generative KI [FHJZ2024] kann Requirements Engineers bei der Priorisierung von Anforderungen unterstützen (siehe Kapitel 6.8 in [GleA2024]), indem sie mehrere Kriterien gleichzeitig analysiert und eine Rangfolge vorschlägt, die auf der Bewertung von Geschäftswert, Dringlichkeit, Aufwand, Abhängigkeiten und anderen relevanten Faktoren basiert.

KI kann eine große Menge an Anforderungen und Stakeholder-Input verarbeiten, um Muster und Abwägungen zu erkennen, die bei einer manuellen Priorisierung übersehen werden könnten, und gleichzeitig die systematische Berücksichtigung aller in [GleA2024] Kapitel 6.8 beschriebenen Priorisierungsschritte sicherstellen. KI kann auch vergleichende Analysen zwischen Anforderungen erstellen und so potenzielle Konflikte oder Abhängigkeiten aufzeigen, die Priorisierungsentscheidungen beeinflussen könnten. Darüber hinaus kann es die Priorisierungsvorschläge auf der Grundlage unterschiedlicher Stakeholder-Perspektiven oder sich ändernder Projektbeschränkungen anpassen.

Eine effektive Priorisierung geht weit über eine algorithmische Analyse hinaus und erfordert ein tiefes Verständnis der Bedürfnisse der Stakeholder, der Geschäftsstrategie und des Projektkontextes – Elemente, die KI nicht vollständig erfassen kann. Die Einbeziehung der entsprechenden Stakeholder in Prioritätsentscheidungen ist nach wie vor unerlässlich (siehe Kapitel 6.8 in [GleA2024]), und KI-Vorschläge müssen anhand dieser Einschätzungen und strategischen Ziele validiert werden. Um die Übereinstimmung mit dem Grundsatz der Wertorientierung (CPRE-Prinzip 1, [IREB2024]) zu wahren, müssen die Prioritäten echte Bedürfnisse der Stakeholder widerspiegeln und nicht abstrakte Maßstäbe.

Bei der iterativen Entwicklung erfordert die dynamische Natur der Prioritätensetzung außerdem eine ständige menschliche Aufsicht, um sicherzustellen, dass die KI-Empfehlungen relevant bleiben, wenn sich die Projektbedingungen weiterentwickeln.

## 5.5 Wichtige Erkenntnisse

KI kann alle zentralen RE-Aktivitäten – Ermittlung, Dokumentation, Validierung und Anforderungsmanagement – verbessern, indem sie die Effizienz, Struktur und Konsistenz erhöht. Sie sollte als unterstützendes Werkzeug gesehen werden, das Aufgaben beschleunigt, aber nicht das menschliche Fachwissen ersetzt. Requirements Engineers müssen die KI-Ergebnisse prüfen, um sicherzustellen, dass sie korrekt, relevant und auf die Projektziele abgestimmt sind. Letztendlich dient KI als Hilfsmittel für das menschliche Urteilsvermögen, während die Verantwortung für die Qualität der Anforderungen beim Requirements Engineer verbleibt.

## 6 KI-Terminologie

Stufe: K1

Dauer: 0,5 Stunden

Ziel: Kandidaten kennen die Terminologie, die für die Arbeit mit KI unerlässlich ist.

### Lernziele

LZ 6.1      Erinnern und identifizieren von Kernterminologie, die für die Arbeit mit KI wichtig ist. (K1)

In diesem Kapitel wird die Schlüsselterminologie erklärt, die für das Verständnis und die Arbeit mit KI, insbesondere mit LLMs, unerlässlich ist.

### Chatbot

Ein Chatbot ist ein KI-System, das dafür entwickelt wurde, menschliche Unterhaltungen durch Text oder Sprache zu simulieren. [ZhWX2023]. Moderne Chatbots, die auf LLMs basieren, können den Kontext verstehen, den Dialogverlauf aufrechterhalten und kohärente, kontextbewusste Antworten generieren. Im RE können Chatbots Requirements Engineers durch interaktive Dialoge unterstützen, indem sie Interviews mit Stakeholdern transkribieren, Informationen aus Dokumenten abrufen oder Dokumentationsaufgaben durch interaktive Dialoge unterstützen.

### Chain-of-thought Begründung

Chain-of-Thought-Begründung ist ein Ansatz, bei dem das LLM aufgefordert wird, seine Argumentation Schritt für Schritt zu erklären, bevor es die endgültige Antwort [WeeA2022] gibt. Im RE kann dies die Transparenz und Nachvollziehbarkeit unterstützen, zum Beispiel bei der Analyse widersprüchlicher Anforderungen oder der Bewertung von Entwurfsalternativen.

### Computer Vision

Computer Vision verwandelt rohe Pixel in strukturierte Fakten [Szel2022]. Moderne Bildverarbeitungsnetzwerke erkennen Objekte, lesen Text aus einem Bild und schätzen die räumliche Tiefe. Das bekannteste Beispiel ist wohl die Gesichtserkennung zum Entsperren moderner Handys.

### Context Window

Das Context Window definiert die maximale Anzahl von Token, die ein LLM auf einmal berücksichtigen kann (einschließlich des Prompts und der generierten Ausgabe) [BreA2020]. Diese Grenze schränkt ein, wie viel von einer Anforderungsspezifikation oder Dialoghistorie bei der Generierung von Antworten berücksichtigt werden kann. Bei RE-Aufgaben wie der Validierung von Anforderungen oder der Analyse von Stakeholder-Feedback hilft das Wissen um die Größe des Context Windows bei der effizienten Strukturierung der Eingaben. Dies ist jedoch nicht immer eine "größer ist besser"-Sache: Je mehr Input Sie einem LLM geben, desto besser ist es am Anfang, aber zu viel Input kann auch dazu führen, dass das Modell abgelenkt wird.

## Deep Learning

Deep Learning ist ein Teilbereich des maschinellen Lernens, bei dem neuronale Netze (z. B. Transformer für Sprache und gefaltete neuronale Netzwerke – convolutional networks – für Bilder) verwendet werden, um komplexe Datentypen und Aufgaben zu bearbeiten, einschließlich verstärkendem Lernen [GoBC2016].

## Embeddings

Embeddings sind numerische Vektordarstellungen von Text, die die semantische Bedeutung erfassen [VoFi2025]. Im RE ermöglichen Embeddings die semantische Suche, das Erkennen von Duplikaten, das Gruppieren und die Rückverfolgbarkeit zwischen Anforderungen und zugehörigen Artefakten – selbst wenn der Wortlaut unterschiedlich ist. Embeddings bilden die Grundlage für viele Abfrage- und Analysefunktionen, erzeugen aber selbst keinen Text.

## Expertensysteme

Expertensysteme erfassen menschliches Fachwissen in Form von expliziten Regeln und Ontologien und liefern überprüfbare, deterministische Antworten. [Bari2025]. Sie werden nach wie vor bevorzugt in Bereichen wie dem Steuerrecht oder der medizinischen Triage eingesetzt, wo Transparenz über Wahrscheinlichkeit geht und absolut kein Spielraum für Fehler erlaubt ist. Moderne Steuersoftware kann z. B. mithilfe von KI Ihre Rechnungen, Einkommensnachweise und andere Daten analysieren, um die Höhe Ihrer Steuerschuld zu berechnen.

## Fine-tuning

Unter Fine-Tuning versteht man den Prozess der Anpassung eines vortrainierten LLM an eine bestimmte Domäne oder Aufgabe durch Training mit zusätzlichen, relevanten Daten [HoRu2018]. Für RE-Aufgaben kann Fine-Tuning das Modell an die Terminologie des Fachgebiets, an die Sprache der Vorschriften oder an organisatorische Standards anpassen und so die Genauigkeit der generierten Anforderungen oder der Konformitätsprüfungen verbessern. Fine-Tuning erfordert Aufwand, Fachwissen und qualitätsgeprüfte Daten, um effektiv zu sein.

## Few-shot / Zero-shot learning

Few-Shot- und Zero-Shot-Learning sind Techniken, bei denen das Modell eine Aufgabe mit wenigen oder keinen Beispielen im Prompt [BreA2020], [DAIR2024] ausführt. Im RE können sie verwendet werden, um Anforderungen zu klassifizieren, Akzeptanzkriterien zu erstellen oder Spezifikationen ohne umfangreiche aufgabenspezifische Trainingsdaten umzuformulieren.

## Halluzination

Halluzinationen sind eine direkte Folge des in Kapitel 1.1 [FHJZ2024] beschriebenen probabilistischen Mechanismus. Eine Halluzination tritt auf, wenn ein LLM einen statistisch wahrscheinlichen Text produziert, der faktisch falsch ist oder nicht durch Quelldaten gestützt wird.

Im RE können Halluzinationen besonders problematisch sein, wenn erfundene Details mit Aussagen von Stakeholdern oder Fakten aus der Domäne verwechselt werden. Requirements Engineers, die LLMs verwenden, müssen daher KI generierte Inhalte anhand maßgeblicher Quellen validieren, in Übereinstimmung mit dem CPRE-Prinzip 6, dass nicht validierte Anforderungen nutzlos sind [IREB2024]. Alle von KI erzeugten Ergebnisse müssen als ungeprüfte Entwürfe behandelt werden.

### Maschinelles Lernen (ML)

Algorithmen des maschinellen Lernens lernen Muster direkt aus Daten [Bish2006]. Klassisches ML sagt Zahlen oder Kategorien voraus; Deep-Learning-Netze wie Transformer und Neuronale Netze verarbeiten Sprache, Bilder und Strategien für Reinforcement Learning (verstärkendes Lernen). Die Qualität der Ergebnisse steigt mit der Größe und Relevanz der Trainingsmenge. Ein Modell könnte zum Beispiel Daten über vergangene Hausverkäufe verwenden und die Preise für zu verkaufende Immobilien vorhersagen.

### Natural language processing (NLP)

NLP umfasst alles, was Sprache oder Text in Bedeutung umwandelt und umgekehrt [QieA2020]. Das reicht von regelbasierten Parsern hin zu den heutigen LLM-Chatbots sowie Speech-to-Text- und Text-to-Speech-Modulen.

### Prompt

Ein Prompt ist die Eingabe, die einem LLM zur Verfügung gestellt wird, um seine Ausgabe zu steuern [Kuka2024]. Es kann sich um eine Frage, eine Anweisung oder einen Teil des Textes handeln. In der RE-Praxis bestimmen Aufforderungen die Relevanz und die Genauigkeit der generierten Ergebnisse, z. B. wenn ein LLM aufgefordert wird, alternative Formulierungen einer Anforderung vorzuschlagen oder funktionale Anforderungen aus Interviews mit Stakeholdern zu extrahieren. Die Qualität des Prompts hat einen direkten Einfluss auf die Nützlichkeit der Antwort.

### Prompt engineering

Prompt-Engineering ist die Praxis der Entwicklung und Verfeinerung von Prompts, um zuverlässig die gewünschten Ergebnisse von einem LLM [CZLZ2025] zu erhalten. Dies ist vergleichbar mit der Ausarbeitung von präzisen Interviewfragen oder Vorlagenstrukturen in RE: Klarheit und Struktur verbessern die Qualität der gewonnenen Informationen.

### Retrieval-augmented Generation (RAG)

Retrieval-augmented generation (RAG) ist ein Prozess, der Embeddings verwendet, um relevante Dokumente zu finden, und diese dann an ein LLM weitergibt, um eine Antwort zu generieren [LeeA2020]. Im RE stellt RAG sicher, dass KI-Ergebnisse auf tatsächlichen Spezifikationen, Notizen von Stakeholdern oder regulatorischen Dokumenten basieren, wodurch das Risiko von Halluzinationen verringert wird und die Ergebnisse projektspezifisch bleiben, anstatt nur auf den internen Trainingsdaten des Modells zu basieren.

## Robotik

Die Robotik kombiniert die Sensordatenfusion (Integration von Daten mehrerer Sensoren), die Kartierung (Erstellung eines Abbilds der Umgebung), die Wegplanung (Entscheidung, wie etwas sich bewegt) und die Rückkopplungssteuerung (Anpassung von Aktionen auf der Grundlage von Sensoreingaben), damit Maschinen autonom navigieren und die physische Welt manipulieren können. [Szel2022]. Anwendungen reichen von Kommissionierern und autonomen Drohnen bis hin zu Staubsaugern, die Ihr Wohnzimmer kartografisch erfassen und Hindernissen ausweichen können.

## Systemprompt / -anweisung

Ein Systemprompt ist eine spezielle Art von Prompt, der das übergreifende Verhalten oder die Rolle des LLM festlegt (z.B. "Sie sind ein Assistenz-Requirements Engineer...") [CZLZ2025]. Dadurch wird sichergestellt, dass die erzeugten Ergebnisse den Stilrichtlinien der Organisation oder bewährten RE-Verfahren entsprechen.

## Temperature

Die Temperatur ist ein Parameter, der die Zufälligkeit der Ausgabe eines LLM steuert [BreA2020]. Niedrige Werte machen das Modell deterministischer und führen zu konsistenten Formulierungen, während höhere Werte zu vielfältigeren und kreativeren Ergebnissen führen. Im RE sind niedrigere Temperaturen oft vorzuziehen, um eine einheitliche Formulierung der Anforderungen zu erreichen, während höhere Werte nützlich sein können, um verschiedene Lösungsideen zu entwickeln. Bei vielen LLMs kann die Temperatur jedoch nicht von den Benutzern eingestellt werden.

## Token

Ein Token ist eine Texteinheit – oft ein Wort, ein Teil eines Wortes oder eine Zeichensetzung –, die das Modell [SeHB2016] verarbeitet. LLMs arbeiten mit Token und nicht mit ganzen Sätzen, und ihre Leistung und Kosten werden teilweise durch die Anzahl der verarbeiteten Token bestimmt. In RE-Kontexten können Token-Limits beeinflussen, wie viel Spezifikationstext, Stakeholder-Input oder Glossarinhalte in einer einzigen Interaktion verarbeitet werden können.

## Transformer

Transformer sind die neuronale Netzwerkarchitektur hinter den meisten modernen LLMs [VaeA2017]. Sie nutzt einen Aufmerksamkeitsmechanismus, um die Bedeutung von Wörtern im Verhältnis zueinander zu bewerten, unabhängig von ihrer Position, was eine effiziente Handhabung langer Sequenzen und komplexer Sprachmuster ermöglicht. Im RE ermöglichen Transformer KI-Tools die Interpretation und Generierung von Anforderungstexten mit einem hohen Maß an Kontextbewusstsein, z. B. die Identifizierung von Abhängigkeiten oder die Aufrechterhaltung einer konsistenten Terminologie über verschiedene Dokumente hinweg.

## Vektor

Eine mathematische Darstellung von Daten als geordnete Liste von Zahlen [MCCD2013]. Im Zusammenhang mit Embeddings erfassen Vektoren die semantische Bedeutung von Texten, indem sie ähnliche Wörter, Sätze oder Anforderungen in einem mehrdimensionalen Raum nahe beieinander platzieren. Im RE ermöglichen Vektoren Operationen wie die Ähnlichkeitssuche, das Clustern verwandter Anforderungen oder die Verknüpfung von Stakeholder-Bedürfnissen mit vorhandenen Unterlagen, selbst wenn unterschiedliche Formulierungen verwendet werden.

## 7 Referenzen

- [Bari2025] Barisic, Davor: Exploring the Landscape of Expert Systems: A Review. International Journal of Management Trends: Key Concepts and Research, 4(1), 58–68, 2025.
- [Bish2006] Bishop, Christopher M.: Pattern Recognition and Machine Learning. Springer, 2006.
- [BreA2020] Brown, Tom et al.: "Language models are few-shot learners." Advances in Neural Information Processing Systems 33 (2020): 1877–1901.
- [CZLZ2025] Chen, Banghao, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu: Unleashing the Potential of Prompt Engineering for Large Language Models. Patterns 6, no. 6 (2025).
- [DAIR2024] DAIR.AI. Zero-Shot Prompting. Prompting Guide, 2024. <https://www.promptingguide.ai/de/techniques/zeroshot>. Zuletzt besucht im Oktober 2025.
- [DDOS2024] Deters, Hannah, Droste, Jakob, Obaidi, Marti, Schneider, Kurt: How Explainable Is Your System? Towards a Quality Model for Explainability, Springer Nature Switzerland, 2024.
- [EUAI2026] The EU Artificial Intelligence Act (AIA), 2026. <https://artificialintelligenceact.eu/de/> Zuletzt besucht im Februar 2026.
- [FDA1997] FDA 21 CFR Part 11: U.S. Food and Drug Administration. Title 21 Code of Federal Regulations Part 11: Electronic Records; Electronic Signatures. U.S. Government Publishing Office, 1997. <https://www.ecfr.gov/current/title-21/chapter-I/subchapter-A/part-11> Zuletzt besucht im November 2025.
- [FHJZ2024] Feuerriegel, Stefan, Jonas Hartmann, Christian Janiesch, and Patrick Zschech: Generative AI. Business & Information Systems Engineering vol. 66(1), pages 111–126, February 2024.
- [GDPR2016] European Union. Datenschutz–Grundverordnung (EU) 2016/679. VERORDNUNG DES EUROPÄISCHEN PARLAMENTS UND DES RATES, 2016. <https://eur-lex.europa.eu/eli/reg/2016/679/oj?locale=de> Zuletzt besucht im November 2025.
- [GleA2024] M. Glinz et al.: Handbook for the CPRE Foundation Level according to the IREB Standard (Version 1.2.0). IREB e.V., Karlsruhe, 2024. <https://cpre.ireb.org/de/downloads-and-resources/downloads#cpre-foundation-level-handbook>. Zuletzt besucht im September 2025.
- [Glin2024] M. Glinz: A Glossary of Requirements Engineering Terminology (Version 2.1.0). IREB e.V., Karlsruhe, 2024. <https://cpre.ireb.org/de/downloads-and-resources/downloads#cpre-glossary>. Zuletzt besucht im September 2025.
- [GoBC2016] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville: Deep Learning. MIT Press, 2016.

- [HoRu2018] Howard, Jeremy, and Sebastian Ruder. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), 2018.
- [IEC62304] International Electrotechnical Commission. IEC 62304: Medical Device Software—Software Life Cycle Processes. IEC, 2006.  
<https://webstore.iec.ch/en/publication/6792> (Anmerkung: 2021 als aktuell bestätigt). Zuletzt besucht im Oktober 2025.
- [IREB2025] #AIREB Booklet zu RE und AI. <https://cpre.ireb.org/de/downloads-and-resources/downloads#ai-snippets>. Zuletzt besucht im Oktober 2025.
- [IREB2024] IREB Certified Professional for Requirements Engineering Foundation Level Syllabus (version 3.2.0). IREB e.V., Karlsruhe, 2024.  
<https://cpre.ireb.org/de/downloads-and-resources/downloads#cpre-foundation-level-syllabus>. Zuletzt besucht im September 2025.
- [IREB2026] AI4RE Prompt Guide. Prompt-Vorlagen für verschiedene RE-Anwendungsfälle von IREB. <https://ireb.atlassian.net/wiki/spaces/airebpromptguide/overview>. Zuletzt besucht im Februar 2026.
- [Kuka2024] Kuka, V. (2024, October 22). Prompt Engineering, [https://learnprompting.org/docs/basics/prompt\\_engineering](https://learnprompting.org/docs/basics/prompt_engineering). Zuletzt besucht im Oktober 2025.
- [LeeA2020] Lewis, Patrick et al.: "Retrieval-augmented generation for knowledge-intensive NLP tasks." Advances in Neural Information Processing Systems 33 (2020): 9459–9474.
- [MiSC2013] Mikolov, T., Sutskever, I., Chen, K., et al. (2013). "Distributed representations of words and phrases and their compositionality." Advances in Neural Information Processing. <https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html>. Zuletzt besucht im Oktober 2025.
- [MCCD2013] T. Mikolov, K. Chen, G. Corrado, J. Dean: Efficient Estimation of Word Representations in Vector Space. 2013. arXiv:1301.3781 (2013).
- [NoeA2024] Norheim, J.J., Rebentisch, E., Xiao, D., Draeger, L., et al. (2024). "Challenges in applying large language models to Requirements Engineering tasks." Design Science <https://www.cambridge.org/core/journals/design-science/article/challenges-in-applying-large-language-models-to-requirements-engineering-tasks/1FC7666F0A0B4E7091D2D4B2D46321B5>. Zuletzt besucht im Oktober 2025.
- [QieA2020] Qiu, Xipeng et al.: Pre-trained Models for Natural Language Processing: A Survey. Science China Technological Sciences vol. 63, no. 10 (2020): 1872–1897.

- [RaeA2019] Radford, Alec e. al: "Language models are unsupervised multitask learners." OpenAI blog 1, no. 8 (2019): 9. [https://cdn.openai.com/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf](https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf). Zuletzt besucht im Oktober 2025.
- [SceA2023] Schick, T., Dwivedi-Yu, J., Dessì, R., et al. (2023). "Toolformer: Language models can teach themselves to use tools." Advances in Neural Information Processing Systems. [https://proceedings.neurips.cc/paper\\_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html](https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html). Zuletzt besucht im Oktober 2025.
- [SeHB2016] Sennrich, Rico, Barry Haddow, and Alexandra Birch: Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1715–1725. Association for Computational Linguistics, Berlin, Germany, 2016.
- [Szel2022] Szeliski, Richard: Computer Vision: Algorithms and Applications (2nd ed.). Springer Nature, 2022.
- [VaeA2017] Vaswani, Ashish et al.: "Attention is All you Need." In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (eds.), Advances in Neural Information Processing Systems 30 (NeurIPS 2017). Curran Associates, 2017.
- [VoFi2025] A. Vogelsang, J. Fischbach (2025). Using Large Language Models for Natural Language Processing Tasks in Requirements Engineering: A Systematic Guideline. In: A. Ferrari, G. Ginde (eds.) Handbook on Natural Language Processing for Requirements Engineering. Cham: Springer Nature. Preprint available for free at <https://arxiv.org/pdf/2402.13823>. Zuletzt besucht im Februar 2026.
- [WeeA2022] Wei, Jason et al.: "Chain-of-thought prompting elicits reasoning in large language models." Advances in Neural Information Processing Systems 35 (2022): 24824–24837.
- [WheA2023] White, Jules et al.: "A prompt pattern catalog to enhance prompt engineering with chatgpt." arXiv preprint arXiv:2302.11382 (2023).
- [YeeA2023] Ye, H., Liu, T., Zhang, A., Hua, W., Jia, W. (2023). "Cognitive mirage: A review of hallucinations in large language models." arXiv preprint arXiv:2309.06794
- [ZaDA2025] Zadenoori, M.A., Dąbrowski, J., Alhoshan, W., et al. (2025). "Large Language Models (LLMs) for Requirements Engineering (RE): A Systematic Literature Review." arXiv preprint arXiv:2509.11446
- [ZhWX2023] Zhao, Wayne, Yuan Wang, Xing Xie, et al.: A Survey of Large Language Models. 2023. arXiv:2303.18223 (2023).