O'REILLY®

Für Entwickler, Projekt- und Qualitätsmanager

ChatGPT

in Softwareprojekten

Mit KI Codequalität, Anforderungen und Dokumentation verbessern

Patrick Schnell

ChatGPT in Softwareprojekten

Mit KI Codequalität, Anforderungen und Dokumentation verbessern

Patrick Schnell

O'REILLY®

Patrick Schnell

Lektorat: Ariane Hesse
Fachliche Unterstützung: Andreas Zöller, Nils-Oliver Linden und Eileen Giesel von TNG Technology Consulting sowie Jens Olaf Koch
Korrektorat: Sibylle Feldmann, *www.richtiger-text.de*
Satz: III-satz, *www.drei-satz.de*
Herstellung: Stefanie Weidner, Frank Heidt
Umschlaggestaltung: Karen Montgomery, Michael Oréal, *www.oreal.de*

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über *http://dnb.d-nb.de* abrufbar.

ISBN:
Print 978-3-96009-237-7
PDF 978-3-96010-880-1
ePub 978-3-96010-881-8

1. Auflage 2025

Wieblinger Weg 17
69123 Heidelberg

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«.
O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *kommentar@oreilly.de*.

Inhalt

Vorwort

Die Welt der Softwareprojekte und der Softwareentwicklung ist komplex und äußerst facettenreich. Sie stellt eine einzigartige Herausforderung dar, da sie verschiedene Fachbereiche, organisatorische Anforderungen und technische Details miteinander verknüpft.

Marc Andreessen, Unternehmer und Risikokapitalgeber, formulierte 2011 die viel zitierte Aussage »Software is eating the world«. Er wollte darauf hinweisen, dass Unternehmen, die Software nicht nutzen oder entwickeln, um ihre Geschäftsmodelle zu verbessern oder neu zu definieren, Gefahr laufen, von innovativeren softwaregetriebenen Konkurrenten überholt zu werden.

Seitdem ChatGPT von OpenAI einer breiten Öffentlichkeit zugänglich gemacht wurde, wird es weltweit von Menschen in den unterschiedlichsten Branchen und Arbeitsbereichen genutzt – häufig mit einer Mischung aus Neugier, Faszination, Skepsis und Besorgnis. Gedanklich an Andreessens Ausspruch anknüpfend, fragen sich derzeit sicher viele von uns, ob nun ChatGPT & Co. die Welt verschlingen.

Auch in der Softwareentwicklung stellt sich – wie in den allermeisten Branchen – die Frage, wie Jobs und Tätigkeitsfelder durch die transformative Kraft dieser leistungsfähigen Sprachmodelle verändert werden und wo menschliche Arbeitskraft und Kreativität vielleicht sogar durch sie ersetzt wird. In der Softwareentwicklung werden Sprachmodelle heute bereits in den verschiedensten Bereichen eingesetzt: etwa für die automatisierte Codegenerierung, um Dokumentation zu erstellen, für die Fehlerdiagnose und -behebung, zum Refaktorieren von Code oder im Testing, um beispielsweise Unit- und Integrationstests zu erstellen.

Was bedeuten diese hoch dynamischen Entwicklungen für die Mitarbeiterinnen und Mitarbeiter der Software- und IT-Branche? Ich bin überzeugt, dass wir uns als Softwareentwickler oder IT-Professionals intensiv mit diesen neuen technischen Möglichkeiten beschäftigen müssen. Aber auch dann, wenn

KI-Tools uns effektiv beim Generieren von Codesegmenten und auch in anderen Aufgabenbereichen unterstützen, können sie menschliche Kompetenz und Kreativität nicht ersetzen – vor allem nicht in bestimmten Bereichen. Es bleibt nach meiner Einschätzung eine wesentliche Kompetenz der Developer, Software Engineers sowie Architektinnen und Architekten, die fachlichen Anforderungen der Kunden in eine Softwarearchitektur und Codestruktur zu übersetzen, die flexibel und zukunftsfähig ist.

Dieses Buch soll genau diese Faktoren im Zyklus eines Softwareprojekts beleuchten und in einzelnen Fallbeispielen aufzeigen, wie diese Herausforderungen und alltäglichen Aufgaben mithilfe von KI-Sprachmodellen (in diesem Fall ChatGPT) unterstützt werden können.

Warum ich dieses Buch geschrieben habe

Das Thema künstliche Intelligenz und die entsprechenden Tools beschäftigen viele von uns nicht erst seit dem Release von ChatGPT im Jahr 2022. Schon vorher konnte man kleine Modelle oder KI as a Service in der Cloud nutzen. Für mich persönlich war es immer interessant, konkrete Anwendungsfälle auszuprobieren, ohne mich selbst um das Training der Modelle kümmern zu müssen. 2008 kam ich zum ersten Mal mit KI-Modellen in Berührung und fand es beschwerlich, ein Modell regelmäßig aufs Neue zu trainieren und anzupassen. Ich hatte mich seinerzeit mit der Klassifizierung von Dokumententypen beschäftigt.

Das Release von GPT und den dazugehörigen APIs markiert einen gewaltigen Schritt in der Entwicklung, KI-Unterstützung einer breiten Öffentlichkeit zugänglich zu machen. Insbesondere durch ChatGPT wurden diese KI-Modelle nun sehr leicht zugänglich, da ChatGPT speziell für Chatbots, virtuelle Assistenten und andere konversationsbasierte Anwendungen entwickelt wurde. Aufgrund ihrer Leistungsfähigkeit und Zugänglichkeit werden ChatGPT & Co. unseren produktiven Alltag stark beeinflussen.

Darüber hinaus sind inzwischen unzählige Tools speziell für Softwareentwickler auf den Markt gekommen, die das Schreiben von Code (teilweise) übernehmen bzw. das Programmieren produktiver gestalten. Beispielhaft seien GitHub Copilot und CodeWhisperer von Amazon genannt. Zum Thema der KI-gestützten Softwareentwicklung und den entsprechenden Tools gibt es viele Fachbeiträge, Videos und Fachbücher. Bei O'Reilly erscheint im Herbst 2024 in deutscher Übersetzung das Buch »Programmieren mit KI-Assistenz« von Tom Taulli.

Ich werde mich in diesem Buch auf ein paar Themen und Anwendungsbereiche konzentrieren, die in Softwareprojekten häufig zu Problemen führen – weil Aufgaben nicht korrekt umgesetzt werden, weil sie vielleicht nicht zu den beliebtesten Aufgaben gehören oder weil sie zu stark von der eigenen Perspektive und den persönlichen Vorlieben geprägt sind. Wie der Titel des Buchs bereits ankündigt, wird es in diesem Buch um das Überprüfen der Codequalität und die Bearbeitung von Anforderungsdefinitionen und Dokumentationen mithilfe von ChatGPT gehen.

Projekte scheitern immer wieder aufgrund von mangelnder Nutzerakzeptanz, verfehlten Zielen oder finanziellen Problemen. Auslöser hierfür sind häufig aber nicht etwa die schlechte technische Umsetzung, fehlende Anforderungsdefinitionen oder zu teure Prozesse. In der Regel werden Softwareprojekte von Expertinnen und Experten mit hoher Fachkompetenz in ihrem Spezialgebiet durchgeführt. Projekte scheitern also nicht an mangelndem Wissen, sondern viel eher, weil unterschiedliche Sichtweisen auf das Projekt bestehen und sich daraus Missverständnisse ergeben. Gerade bei der Definition von Anforderungen oder bei der Einigung auf einzelne Testkriterien konnte ich in Projekten, die ich begleitet habe, diese Abstimmungsschwierigkeiten immer wieder beobachten. Vielfach wird auch nicht ausreichend berücksichtigt, wie wichtig es ist, die Wartbarkeit der Codebasis im Blick zu behalten, um flexible und robuste Software zu entwickeln.

Dieses Buch soll genau diese Arbeitsbereiche über den Lebenszyklus eines Softwareprojekts hinweg beleuchten und anhand von Beispielen zeigen, wie Aufgaben mithilfe von KI-Sprachmodellen (in diesem Fall ChatGPT) im Joballtag besser gemeistert werden können.

Wer dieses Buch lesen sollte

Dieses Buch wendet sich an verschiedene Personengruppen, die an Softwareprojekten mitarbeiten: beispielsweise an Entwicklerinnen und Entwickler, die sich mit Sprachmodellen repetitive Aufgaben wie z. B. das Kommentieren von Code etwas erleichtern möchten.

Eine weitere Gruppe sind Projektleiterinnen und Projektleiter, Product Owner und andere Teammitglieder, die eine organisatorische Rolle haben und als Kommunikationsstelle zwischen Kunden/Stakeholdern und der technischen Entwicklung vermitteln. Dieser Personenkreis wird von den Beispielen und Ausführungen zu den Themen Anforderungsdefinition und Use Cases profitieren. Auch das Thema Dokumentation dürfte für sie von Interesse sein.

Darüber hinaus möchte ich die Mitarbeiterinnen und Mitarbeiter im Bereich der Softwarequalitätssicherung und des Testings ansprechen. Die Wartbarkeit von Code ist für sie ausgesprochen wichtig. Auch das Erstellen von Akzeptanzkriterien und Testfällen mittels Sprachmodellen dürfte ihnen die Arbeit erleichtern und dazu beitragen, dass langwierige Rückfragen oder Klärungsrunden an der einen oder anderen Stelle abgekürzt werden können.

Ich hoffe, dass die Beispiele und Erläuterungen die Betroffenen dazu anregen, ChatGPT zu ihrer Entlastung zu nutzen und zugleich erste Erfahrungen mit diesen leistungsfähigen und hochinteressanten Sprachmodellen zu sammeln.

Wer dieses Buch nicht lesen sollte

Zwar werde ich kurz auf die Arbeits- und Funktionsweise dieser KI-Modelle eingehen, das Buch vermittelt aber keine Machine-Learning-Grundlagen. Selbstverständlich werde ich auf einzelne Aspekte der Funktionsweisen von KI-Modellen bzw. deren Lern- und Trainingsprozesse eingehen, dieses Buch wird jedoch keine detaillierte oder gar vollständige Darstellung des Machine Learning geben. Vielmehr geht es mir darum, den Einsatz von Large Language Models möglichst praktisch und anschaulich zu beschreiben. Das mag – aus Sicht von ML-Experten – an der einen oder anderen Stelle vielleicht zu etwas vereinfachenden oder pauschalisierenden Aussagen führen.

Des Weiteren werde ich in diesem Buch das Thema Codegenerierung durch Sprachmodelle nur am Rande beleuchten. Wer sich vor allem für die Codegenerierung mithilfe von Sprachmodellen interessiert, wird sich sicher eher für Fachbücher mit einem anderen inhaltlichen Fokus wie beispielsweise »Programmieren mit KI-Assistenz« von Tom Taulli interessieren.

Überblick über die Buchinhalte

In einem ersten Kapitel sehen wir uns in die grundsätzliche Funktionsweise von Sprachmodellen an und wie diese mithilfe von sogenanntem Prompting gesteuert werden können. Hier erfahren Sie, welche Best Practices und Techniken eingesetzt werden können. Nach der Lektüre des Kapitels wissen Sie, wie man Sprachmodelle in eine gewünschte Richtung lenken kann und Abfragen optimiert ausführt.

Kapitel 2 skizziert verschiedene allgemeine Techniken und Aspekte in Bezug auf die Qualität von Software und ist für Softwareentwickler interessant. Hier

gehen wir insbesondere darauf ein, wie lesbarer und gut strukturierter Code einen Einfluss auf die Softwarequalität sowie die mittel- und langfristige Wartbarkeit von Codebasen hat.

Das dritte Kapitel wird hierzu einige Beispiele aufgreifen, und wir werden versuchen, existierenden Code mithilfe von Sprachmodellen zu optimieren, den Code lesbarer zu gestalten und ihn besser zu strukturieren. Die durchgespielten Beispiele versetzen Sie in die Lage, in eigenen Projekten Refactorings mittels ChatGPT durchzuführen und dabei sowohl den Aufbau als auch die Formatierung Ihres Codes in wenigen Schritten zu überarbeiten. Mit recht einfachen Maßnahmen erhalten Sie so einen wesentlich einfacher zu wartenden und, z. B. für neue Teammitglieder, verständlicheren Code.

In Kapitel 4 beleuchten wir einen Teil der überaus komplexen und oftmals als große Herausforderung empfundenen Arbeitsschritte, die im Rahmen der Anforderungsdefinition anfallen. Hier werden Sie sehen, wie grundlegende Techniken Anforderungen verbessern, wie diese im Rahmen der agilen Entwicklungsmethoden eingesetzt werden können und wie wichtig die korrekte und umfassende Definition der Anforderungen zu Beginn eines Softwareprojekts sind.

Insbesondere in Kapitel 5 werden einzelne Anwendungsfälle beleuchtet, und wir sehen uns an, wie mittels ChatGPT sowohl Anforderungen analysiert und validiert als auch User Stories oder Testfälle bzw. Akzeptanzkriterien generiert werden können. Wir stellen passende Vorlagen und Beispiel-Prompts vor, um zu demonstrieren, wie Sie die Definition von Anforderungen, sei es fachlicher oder technischer Natur, wesentlich beschleunigen und Fehler vermeiden können.

Ergänzend gehen wir in Kapitel 6 auf die immense Bedeutung von aussagekräftiger Dokumentation ein und beschreiben, wie man sie mit den Akzeptanzkriterien abgleicht. Hierzu stellen wir verschiedene Techniken zur Beschreibung von Code und Anwendungen vor. Ziel ist es, anhand von Fallbeispielen sowohl Datenmodelle als auch Anwendungslogik mittels ChatGPT zu beschreiben und zu dokumentieren. Zusätzlich wird demonstriert, wie z. B. unleserlicher oder schwer verständlicher Code mittels ChatGPT über einfache Prompts verständlicher gemacht werden kann. Das Sprachmodell kann dabei helfen, unbekannten Code zu durchleuchten und dessen Zweck und Ablauf schneller zu verstehen. Ein weiterer Aspekt ist die Verwendung der standardisierten HTTP-Schnittstellendokumentation OpenAPI, die wir nutzen, um ChatGPT-gestützt passenden Clientcode nach unseren Vorstellungen zu generieren.

Die im Buch befindlichen Fallbeispiele sollen die Themen Dokumentation, Codegenerierung und Anforderungsdefinition zusammenbringen und aufzeigen, wie eine integrierte KI-Strategie allein schon auf Basis von frei verfügbaren Sprachmodellen eine erhebliche Arbeitserleichterung darstellen kann.

Das letzte Kapitel befasst sich mit den Themen Ethik sowie Datenschutz und geht kurz darauf ein, wie sich die Arbeit an Softwareprojekten zukünftig verändern könnte, welche Rolle Sprachmodelle hierbei spielen und welche Chancen sich aus ihrem Einsatz ergeben können.

Warum ChatGPT, und welche Alternativen gibt es?

Die in diesem Buch vorgestellten Prompts und Ergebnisse im Rahmen unserer Beispiele wurden mit den kostenfreien Versionen von ChatGPT (ChatGPT-3.5 und ChatGPT 4o) generiert.

Grundsätzlich sind aber alle Anwendungsfälle und Beispiele auf andere KI-basierte generative Sprachmodelle anwendbar. In diesem Buch habe ich mich jedoch bewusst dafür entschieden, ein Sprachmodell zu verwenden, das von allen Leserinnen und Lesern kostenfrei genutzt werden kann und möglichst niedrigschwellig einsetzbar ist (z. B. ohne eigene Installation).

Wer möchte, kann natürlich auch die kostenpflichtigen (oder neuen kostenfreien Modelle) von OpenAI nutzen, um auf ein optimiertes Modell zuzugreifen. Darüber hinaus stehen inzwischen einige Sprachmodelle zur Verfügung. Wer sich einen Überblick darüber verschaffen möchte, kann auf online verfügbare Listen zurückgreifen, die regelmäßig aktualisiert werden, wie z. B. die Übersicht von Steven Van Vaerenbergh auf GitHub (*https://github.com/steven2358/awesome-generative-ai*).

Modelle, die wie ChatGPT für unsere Beispiele genutzt werden können, sind unter anderem (Stand Mitte 2024, alphabetisch sortiert):

Tabelle 0-1: Übersicht über weitere Sprachmodelle

Name des Modells	Unternehmen	Open Source?	Link
Claude	Anthropic	Nein	*https://claude.ai*
Llama	Meta (USA)	Ja	*https://llama.meta.com*
Luminous	Aleph Alpha (Deutschland)	Nein	*https://app.aleph-alpha.com*
Mixtral	Mistral AI (Frankreich)	Ja	*https://mistral.ai*

Es ist wichtig zu wissen, dass wir bei der Nutzung von KI-basierten Modellen nie repetitiv exakt gleiche Antworten erhalten werden. In Kapitel 1, »Mit Sprachmodellen reden: das Prompting«, gehe ich darauf näher ein. Wenn Sie eins der vorhandenen Beispiele nutzen und einen gleichlautenden Prompt in

ChatGPT und in ein anderes Modell eingeben, können und werden sich Qualität, Umfang und Korrektheit der Antwort teils gering, manchmal aber auch drastisch unterscheiden. Details dazu und welche Techniken man zum Gegensteuern einsetzen kann, erfahren Sie im nachfolgenden Kapitel.

Noch eine Anmerkung, bevor wir einsteigen: Wir wenden uns mit diesem Fachbuch selbstverständlich an alle Menschen, unabhängig von ihrem Geschlecht und Pronomen. Wir haben in diesem Buch versucht, eine geschlechtergerechtere Sprache umzusetzen: durch Nennung der weiblichen und männlichen Form, durch einen abwechselnden Gebrauch der weiblichen und der männlichen Form, durch – sofern passend – Partizipformen wie »die Mitarbeitenden« oder textliche Umschreibungen wie »alle, die an Softwareprojekten mitarbeiten«. Beschreibt eine Textpassage aus unserer Sicht eher eine berufliche Rolle wie beispielsweise einen »Tester«, haben wir auf das Gendern verzichtet.

In diesem Buch verwendete Konventionen

Die folgenden typografischen Konventionen werden in diesem Buch verwendet:

Kursiv
: Kennzeichnet neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateiendungen.

`Konstante Zeichenbreite`
: Wird für Programmlistings und für Programmelemente in Textabschnitten wie Namen von Variablen und Funktionen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter verwendet.

`Konstante Zeichenbreite, fett`
: Kennzeichnet Befehle oder anderen Text, den der Nutzer wörtlich eingeben sollte oder wenn etwas hervorgehoben werden soll.

`Konstante Zeichenbreite, kursiv`
: Kennzeichnet Text, den der Nutzer je nach Kontext durch entsprechende Werte ersetzen soll.

Dieses Symbol steht für einen Tipp oder eine Empfehlung.

Dieses Symbol steht für einen allgemeinen Hinweis.

Dieses Symbol warnt oder mahnt zur Vorsicht.

In einigen Prompts und Aufforderungen, die an das Sprachmodell gesendet werden, kommen Tabellen zum Einsatz. Üblicherweise werden diese Tabellen mit einer Markdown-Formatierung in den Text des Prompts eingebettet. Da solche per Markdown erstellten Tabellen viel Platz in Anspruch nehmen und in diesem Buch nicht gut dargestellt werden können, wird in den Prompts lediglich auf diese Tabellen verwiesen. Wenn Sie Prompts mit inkludierten Tabellen ausprobieren möchten, können Sie diese Verweise durch eine passende Tabelle im Markdown-Format ersetzen. Im Folgenden sehen Sie zwei Screenshots solcher Tabellen im Markdown-Format.

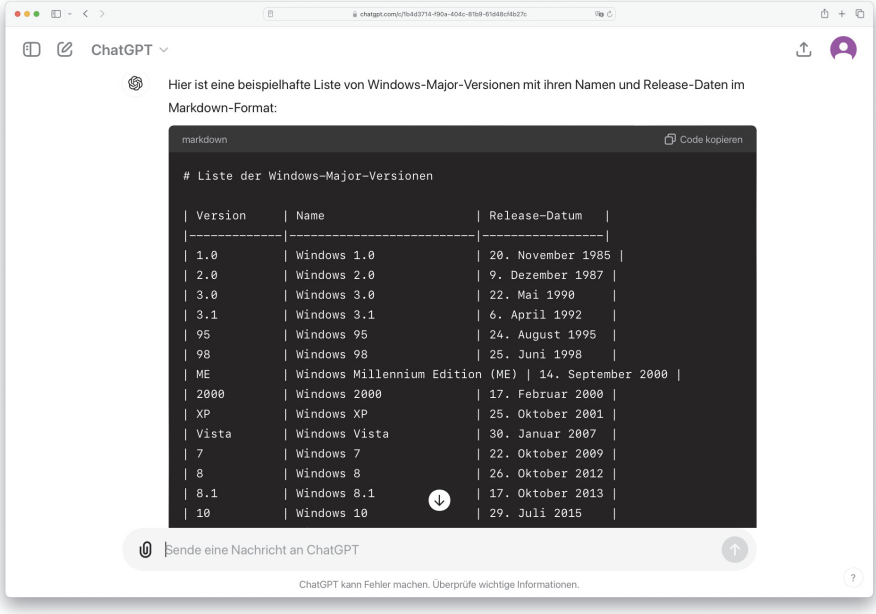

Abbildung 0-1: Tabelle im Markdown-Format, Beispiel 1

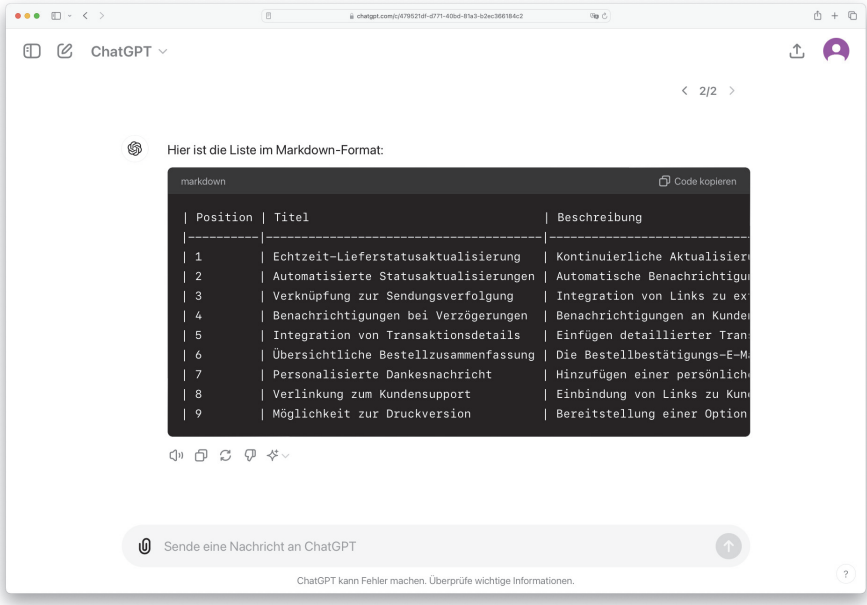

Abbildung 0-2: Tabelle im Markdown-Format, Beispiel 2

Danksagung

Ein Fachbuch zu schreiben, kostet viel Zeit – insbesondere dann, wenn es sich mit einer neuen und hoch dynamischen Technologie beschäftigt.

Gerade vor diesem Hintergrund bin ich den engagierten Reviewern, die sich intensiv mit meinem Manuskript befasst haben, sehr dankbar. Thomas Endres von TNG Technology Consulting war so freundlich, den Kontakt zu interessierten Kolleginnen und Kollegen bei TNG Technology Consulting herzustellen. Ich bedanke mich insbesondere für das kompetente und differenzierte Feedback von Andreas Zöller, Nils-Oliver Linden und Eileen Giesel.

Eine weitere Durchsicht des Manuskripts hat Jens Olaf Koch vorgenommen – ebenfalls fachlich sehr kompetent und mit vielen konkreten Vorschlägen. Jens Olaf Koch hat übrigens auch ein Buch über ChatGPT geschrieben: »Herr Tschie und ich« – allerdings als literarisch-spielerische Entdeckungsreise, also für eine andere Zielgruppe.

Selbstverständlich wäre ein solches Fachbuch nicht ohne eine motivierte Lektorin möglich. Ich danke ganz besonders Ariane Hesse von O'Reilly. Sie hat mich von Anfang an bei diesem Projekt begleitet und mir in zahlreichen Mee-

tings, durch die Auswahl der Reviewer und die Arbeit am Manuskript sowie mithilfe von Brainstormings unschätzbar wertvollen Input zu diesem Buch gegeben.

Besonders liegt mir aber am Herzen, meiner Familie zu danken, die mein Interesse für Technik, IT und die Softwareentwicklung von Kindheit an unterstützt hat. Immer hat sie mir den erforderlichen Freiraum gewährt und toleriert, dass ich bei meinen Versuchen, Software zu verstehen, den Familien-PC regelmäßig »zerschossen« habe.

Inzwischen knüpft daran meine wundervolle Frau an. Dank ihr kann ich auch nach Feierabend oder sogar mal am Wochenende meiner Arbeit nachgehen, die zugleich (und glücklicherweise) mein Hobby ist. Auch sie begeistert sich sehr für verschiedene Themen, und gemeinsam sind wir ein starkes Team.

Ich danke meiner Frau, meinen Eltern und meiner Schwester, die all die Jahre meine Begeisterung unterstützt und regelmäßig meine ausufernden Ausführungen zum Thema Softwareentwicklung ertragen haben.

KAPITEL 1

Mit Sprachmodellen reden: das Prompting

Die Nutzung von Sprachmodellen hat seit Ende 2022 und dem Release von ChatGPT dramatisch zugenommen. Gerade durch ChatGPT sind solche Sprachmodelle (im Englischen als *Large Language Models*, kurz *LLMs*, bezeichnet) schlagartig bekannt geworden. In diesem Buch möchte ich einen Einblick darin geben, wie Entwicklungsteams diese Technologie einsetzen können, um ihre Arbeitsabläufe effizienter zu gestalten und Projekte schneller und effektiver zu realisieren.

Ein Schlüsselelement in der Interaktion mit Sprachmodellen ist das *Prompting*. Doch was genau ist Prompting, und wie kann es für Fragen, die die Entwicklungsarbeit und das Managen von Softwareprojekten betreffen, optimiert werden? In diesem Kapitel beschäftigen wir uns mit dem Konzept des Promptings, untersuchen verschiedene Arten von Prompts und beleuchten mögliche Fallstricke, die alle, die LLMs wie ChatGPT nutzen, im Hinblick auf die sogenannten *Halluzinationen* stets im Hinterkopf haben müssen. Zudem erhalten Sie praktische Tipps und Tricks für das Schreiben von Prompts.

KI und Machine Learning: Was ist das?

Diese Frage detailliert zu beantworten, würde Erklärungen im Umfang mehrerer Fachbücher erfordern. Grundsätzlich reicht die Geschichte der künstlichen Intelligenz ziemlich weit zurück, und eine klare Unterscheidung zwischen modernen Methoden und den ersten Anfängen ist nicht so einfach zu treffen, wie es auf den ersten Blick vielleicht scheint.

KI und das damit verbundene Machine Learning lassen sich darauf zurückführen, dass aus vorgefertigten und teils eigens für den Anwendungsfall aufbereiteten Daten ein *Modell* angefertigt wird. Ein solches Modell beinhaltet je nach Funktionsweise sehr vereinfacht ausgedrückt die statistischen Merkmale dieser

Rohdaten, um, ebenfalls äußerst vereinfacht dargestellt, darauf Bezug nehmend unter Berücksichtigung von Gewichtungen aus den Eingabedaten ein Ergebnis zu kreieren.

Moderne KI-Modelle basieren hauptsächlich auf neuronalen Netzen, die durch Machine Learning und Deep Learning trainiert werden. Während in der Vergangenheit einige KI-Methoden auf Fuzzy-Logik basierten, nutzten die meisten regelbasierte Systeme oder einfache statistische Methoden. Heutzutage ermöglichen maschinelles Lernen und insbesondere Deep Learning eine viel größere Flexibilität und Leistungsfähigkeit, indem sie aus großen Datenmengen lernen und komplexe Muster erkennen. Diese Methoden finden Anwendung in einer Vielzahl von Bereichen, von Empfehlungssystemen in Onlineshops (»Kunden, die dies kauften, kauften auch ...«) bis hin zu Verhaltensmodellen in Computerspielen. Der Begriff KI umfasst somit eine Vielzahl unterschiedlicher Technologien und Verfahren.

Grundsätzlich beschreibt KI Systeme, die autonom auf Eingaben reagieren und Entscheidungen treffen können, ohne dass jede mögliche Reaktion vorab explizit programmiert wurde.

Praxisbeispiel Machine Learning

Wir starten zunächst mit einem kurzen Exkurs, um die grundlegende Arbeitsweise des Machine Learning anhand eines Praxisbeispiels nachzuvollziehen. Ein stark vereinfachtes und häufig im Zusammenhang mit Datenanalyse und Machine Learning genutztes Beispiel ist die Identifizierung von Pflanzentypen, genauer die Identifizierung verschiedener Arten von Schwertlilien. Diese Pflanzenspezies besitzt sogenannte Kronblätter (Petal) sowie sogenannte Kelchblätter (Sepal). Schwertlilien werden in drei Untergruppen unterschieden: »Iris setosa«, »Iris versicolor« und »Iris virginica«. Man kann diese Arten identifizieren, indem man das Verhältnis von Länge und Breite der jeweiligen Kron- und Kelchblätter vergleicht und daraus die Pflanzenspezies ableitet.

Diese verschiedenen Arten von Schwertlilien zu unterscheiden, soll die Aufgabe unseres Modells sein. Eingabewerte sind die Größen der Blätter. Als Ergebnis hätten wir gern die Nennung der jeweiligen Untergruppe der Lilien.

Für dieses Beispiel bedient man sich des sogenannten *Supervised Training*: Hierfür werden große Mengen an Daten sowie Endergebnisse vorgegeben. Das Machine-Learning-Training versucht dann anhand dieser Daten, Rückschlüsse zu ziehen und für neue Eingabedaten die fehlende Information – in diesem Fall die Pflanzengruppe – zu ermitteln. Dieses Verfahren wird bei den meisten KI-