

Coding mit KI

Das Praxisbuch für die Softwareentwicklung

» Hier geht's
direkt
zum Buch

DAS VORWORT

Vorwort

Ziemlich sicher haben Sie es schon ausprobiert: Sie stellen ChatGPT oder einem anderen KI-Tool eine Coding-Frage – und das Tool spuckt innerhalb von Sekunden einen (scheinbar) perfekten Code aus.

In diesem Buch loten wir das Potenzial und die Grenzen von KI-Tools bei der Software-Entwicklung aus. Beginnen wir mit dem Positiven: Wir zeigen Ihnen, dass ChatGPT, GitHub Copilot & Co. mehr können, als ein paar Zeilen Code zu erzeugen. Sie können damit auch Unit-Tests erstellen, Fehler suchen, Refactoring-Aufgaben erledigen, Datenbank-Schemas entwickeln, SQL-Abfragen optimieren, Server administrieren und Scripts verfassen.

Die schöne neue KI-Welt hat aber auch Schattenseiten: Selbst wenn der Code plausibel aussieht und der Begleittext elegant formuliert ist, sind manche KI-Vorschläge schlicht falsch. Die Bandbreite ist groß: Logikfehler, Verwendung nicht vorhandener Funktionen oder die Bezugnahme auf Variablen, die in Wirklichkeit anders heißen. Manchmal funktioniert der Code zwar, aber er ist ineffizient oder unsicher.

KI-Tools machen Fehler, so viel ist unumstritten. Richtig eingesetzt sparen diese neuen Werkzeuge aber auch eine Menge Zeit. Darum geht es in diesem Buch.

Lokale Ausführung von Sprachmodellen

Gerade Firmen haben oft Probleme damit, ihre Code-Basis Tools anzuvertrauen, die in der Cloud laufen. Selbst wenn KI-Anbieter bei Unternehmensangeboten versprechen, Ihren Code nicht für das Training zukünftiger Sprachmodelle zu nutzen, stellt sich die Frage, wie weit Sie diesen Versprechen trauen und ob nicht doch Datenschutzprobleme auftreten können.

Der einfachste Ausweg besteht darin, die für KI-Tools erforderlichen Sprachmodelle lokal auszuführen. Das *Large Language Model* (LLM) von ChatGPT können Sie zwar nicht herunterladen, aber es gibt diverse Alternativen, die kostenlos verfügbar sind und ähnlich gut funktionieren.

Die lokale Ausführung von LLMs ist allerdings ressourcenintensiv. Nur leistungsstarke Notebooks oder PCs sind dazu geeignet. Neue CPUs mit *Neuronal Processing Units* (NPUs) versprechen diesbezüglich neue Möglichkeiten. Ein echter Nutzen für die lokale Ausführung von LLMs ergibt sich aber erst in Kombination mit geeigneter Software – und da muss sich noch viel tun.

Es gibt noch eine Option: Sie können in Ihrer Firma oder in einem Rechenzentrum einen eigenen Server einrichten, der allen Mitarbeitenden für KI-Aufgaben zur Verfügung steht. »Lokal« heißt dann zwar nicht auf dem eigenen Notebook, aber doch auf einem firmeneigenen Rechner. Wir zeigen Ihnen in diesem Buch, wie Sie lokale Sprachmodelle einsetzen und sogar mit eigenem Trainingsmaterial optimieren.

In diesem Buch stellen wir Ihnen die Programme *GPT4All*, *Ollama* und *Tabby* vor, um freie LLMs lokal auszuführen. Sie können die lokalen Sprachmodelle dann sowohl im Chat-Betrieb, zum Coding (z. B. mit dem Editor-Plug-in *Continue*) oder über eine API zur Entwicklung eigener KI-Anwendungen nutzen.

Coding mit KI-Unterstützung für Fortgeschrittene

Wenn Sie das erste Mal ChatGPT um Rat bei einem Coding-Problem gefragt haben, wenn Sie dem ersten Vorschlag von GitHub Copilot (oder einem vergleichbaren Tool) gefolgt sind, dann kennen Sie die interaktive KI-Unterstützung beim Coding. Aber da geht noch mehr!

Mit geeigneten Bibliotheken können Sie noch einen Schritt weiter gehen:

- ▶ **Viele Code-Dateien automatisiert verarbeiten:** Sie wollen in 100 Dateien alle Kommentare von Deutsch auf Englisch umstellen? Ein Projekt von Python 2 nach Python 3 migrieren? Anstatt interaktiv Datei für Datei zu verarbeiten – ist es da nicht naheliegend, ein Script zu verfassen, das diese Aufgabe automatisiert? Wir haben es ausprobiert und teilen unsere Erfahrungen mit Ihnen.
- ▶ **Level-3-Tools:** Ähnlich wie bei selbststeuernden Autos gibt es auch für KI-Coding-Tools Kategorien der Selbstständigkeit. ChatGPT, Claude, GitHub Copilot etc. werden je nach Anwendung den Levels 1 und 2 zugeordnet (*Code Completion* und *Code Creation*). Es existieren aber schon erste Level-3-Tools (*Supervised Automation*), die darüber hinausgehen und auf Ihre Anweisung hin Code-Dateien selbstständig erzeugen bzw. verändern. Wir haben mit *OpenHands* und *Aider* experimentiert und waren insbesondere vom Konzept von *Aider* begeistert.
- ▶ **Retrieval Augmented Generation (RAG):** Für Privatanwender oder kleine Firmen ist es aktuell unmöglich, eigene Sprachmodelle zu trainieren. Mit RAG besteht aber die Möglichkeit, Sprachmodelle um eigene Daten zu ergänzen. Daraus ergeben sich eine Menge zusätzlicher Anwendungsmöglichkeiten, unter anderem die Integration von tagesaktuellen Daten oder die automatisierte Umwandlung von Text-Prompts in SQL-Queries. Wir konzentrieren uns in diesem Kapitel vor allem auf die Bibliothek *LlamaIndex*.

Um es gleich vorwegzunehmen: Theoretisch zeichnet sich mit den hier skizzierten Tools der nächste Entwicklungsschritt im Einsatz von KI beim Coding ab. Praktisch haben wir aber durchwachsene Erfahrungen gemacht. Nicht alles hat so funktioniert,

wie wir uns das erhofft haben. Manche Aufgaben kann die KI in dieser Form schon übernehmen. Bei komplizierteren Wünschen ist aber noch Geduld notwendig, bis diese Tools alltagstauglich werden.

KI macht Fehler und halluziniert

Wenn Sie mit anderen Entwicklern sprechen, werden Sie viele Anekdoten hören, welche kapitalen Fehler KI-Tools zuletzt gemacht haben. Auch in diesem Buch gibt es viele Kapitel, in denen wir Beispiele mit fehlerhaftem KI-Code, falschen Schlussfolgerungen oder schlichtweg erfundene Funktionen oder Optionen zeigen. (Erfundene Funktionen haben mit einer unangenehmen Eigenschaft der aktuell üblichen Sprachmodelle zu tun: Wenn die KI aufgrund unzureichenden Trainingsmaterials keine sichere Antwort geben kann, erfindet sie einen möglichst plausibel klingenden Text. Die KI »halluziniert«.)

Je nachdem, wie Sie KI-Werkzeuge einsetzen, pendelt die Fehlerquote zwischen eher klein (weitverbreitete Programmiersprachen, Alltagsprobleme) und ziemlich hoch (exotische Sprachen, neue Bibliotheken/APIs, sehr spezifische Aufgabenstellung).

Das größte Problem ist gar nicht, dass Fehler passieren. Menschen machen auch Fehler, vermutlich sogar mehr. Ärgerlich ist, dass KI-Tools auch den größten Blödsinn höflich und sprachlich auf hohem Niveau argumentieren. Das macht es so schwierig, richtige Antworten von falschen zu unterscheiden. Wirklich »intelligent« werden KI-Tools aus unserer Sicht erst sein, wenn sie auch einmal antworten: »Das weiß ich nicht.« Oder: »Bei diesem Lösungsvorschlag bin ich mir unsicher, aber so könnte es funktionieren.« So weit sind wir im Herbst 2024 noch nicht, auch wenn es Bestrebungen gibt, dass die Herleitung der Antworten gezeigt wird.

Sie sehen schon: Wir wollen Ihnen kein KI-Tool verkaufen! Im Gegenteil, wir nennen die Probleme, auf die wir gestoßen sind, beim Namen – in allen Kapiteln und insbesondere in Kapitel 13, »Risiken und Ausblick«.

Die drei Phasen der KI-Anwendung

Wir haben während der Arbeit an diesem Buch nicht nur selbst intensivst mit KI-Tools gearbeitet, wir haben auch mit vielen anderen Entwicklern gesprochen. Wir glauben, dass wir dabei ein Muster in der Anwendung von KI-Tools beim Coding erkannt haben. Sie entspricht ziemlich genau dem »Hype-Zyklus«, einem von der Beraterin Jackie Fenn geprägten Modell, wie neue Technologien wahrgenommen und genutzt werden (siehe Abbildung 1).

- **Euphorie:** Die ersten Experimente lassen die meisten Entwickler sprachlos vor Begeisterung zurück. Dieses »Ding« versteht meine Fragen, es gibt sinnvolle Antworten, es erklärt sogar den Code. Wahnsinn!

- ▶ **Ernüchterung:** Dann kommt der Moment, wenn dem KI-Tool Ihrer Wahl die ersten Fehler unterlaufen. Es geht nicht mehr um Hello-World-Funktionen, sondern um echten, komplexen Code. Das KI-Tool erklärt Ihnen mit größter Selbstsicherheit die Welt – und Sie erkennen, dass die Ausführungen sprachlich überzeugend sind, aber inhaltlich komplett falsch. Ihre Schlussfolgerung: KI-Tools sind unbrauchbar, ihr Einsatz verantwortungslos etc.
- ▶ **Pragmatischer Einsatz:** Nach ein paar Wochen entwickeln Sie ein gutes Gefühl dafür, wann KI-Resultate höchstwahrscheinlich korrekt sind und wo Sie nachbessern müssen. Sie können die Grenzen immer besser einschätzen. Übrig bleibt dann ein Tool, das zwar Fehler macht, das Ihnen im Rahmen seiner Möglichkeiten jedoch immer noch eine echte Hilfe ist. Sie hinterfragen jede KI-Antwort, rechnen mit einer Fehlerquote von vielleicht 10 Prozent. Aber Sie nutzen das, was funktioniert. Und das ist eine ganze Menge.

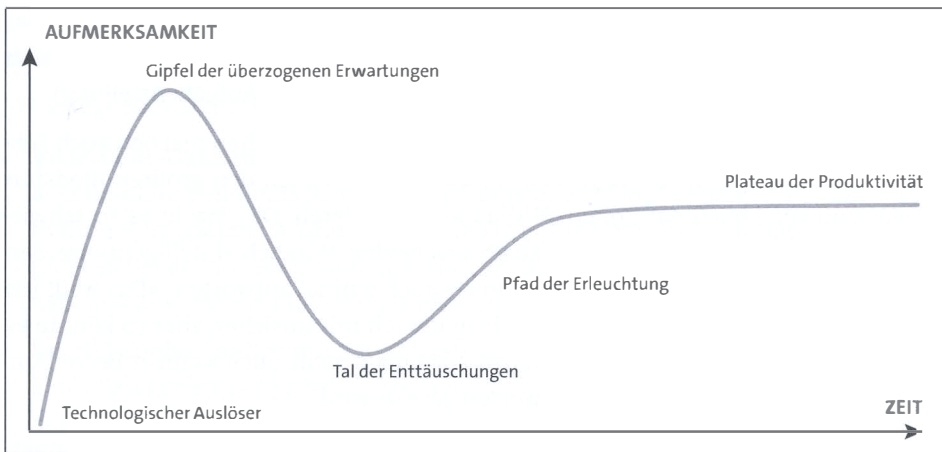


Abbildung 1 Die Nutzung von KI-Tools durchläuft aktuell die Phasen des »Hype-Zyklus«

An KI-Tools geht kein Weg vorbei

Trotz aller Probleme sind wir überzeugt, dass KI-Tools gekommen sind, um zu bleiben. Einer Untersuchung zufolge glauben Manager, dass dieser Zeitgewinn enorm ist, während Entwickler, die KI-Tools tatsächlich anwenden, den Nutzen wesentlich vorsichtiger (und realistischer) abschätzen!

Das Ausmaß des Effizienzgewinns hängt stark von der Art des Codes ab, den Sie entwickeln. Aber ein gewisses Maß an Zeitersparnis, ein höherer Arbeitskomfort bleibt praktisch immer übrig. Ganz egal, ob Sie das Glas halb voll oder halb leer sehen: Coding mit KI-Unterstützung steigert die Effizienz heute schon und morgen noch mehr.

Der Chemiker Derek Lowe hat es im Februar 2023 so formuliert: *I don't think AI and automation will replace chemists, but rather that the human chemists who use them well will replace the ones who don't.* Wir sind davon überzeugt, dass diese seither oft zitierte Aussage auch auf die IT-Welt zutrifft. Künstliche Intelligenz wird Software-Entwickler also nicht ersetzen, zumindest nicht in naher Zukunft. Aber Entwicklerinnen, die KI-Tools vernünftig einsetzen, werden ihre Kollegen ersetzen, die dies nicht tun.

Der richtige Zeitpunkt, sich intensiv mit KI-Tools auseinanderzusetzen, ist jetzt! Ja, KI-Werkzeuge werden in den kommenden Jahren noch besser werden. Aber schon jetzt ist das Potenzial riesig. Und das Wissen und die Erfahrung, die Sie heute gewinnen, werden Ihnen morgen helfen, neue KI-Techniken besser zu verstehen und schneller in Ihren Arbeitsalltag zu integrieren.

Laut <https://survey.stackoverflow.co/2024/ai> verwenden bereits jetzt 63 % aller Entwickler und Entwicklerinnen KI-Tools. Weitere 13 % planen den Einsatz in der nahen Zukunft. Ganz offensichtlich stehen wir am Beginn eines neuen Zeitalters, was das Software-Development betrifft. Bei allen Ängsten, die mit solchen Umbrüchen verbunden sind: Wir haben den Einsatz von KI-Tools in unserem Coding-Alltag als sehr befriedigend empfunden. KI-Tools haben uns nicht das Denken abgenommen, wohl aber eine Menge langweilige Tipparbeit. Die Konzepte der von uns entwickelten Software kamen immer noch von uns; aber die oft öde Arbeit bei der Implementierung ist uns leichter von der Hand gegangen.

In diesem Sinne wünschen wir Ihnen viel Vergnügen bei der Lektüre dieses Buchs. Lassen Sie sich von unseren Beispielen inspirieren, und suchen Sie nach Wegen, KI-Werkzeuge sinnvoll in Ihren Arbeitsalltag zu integrieren. Nutzen Sie die Chancen, die sich hier auftun!

Michael Kofler (<https://kofler.info>)

Bernd Öggl (<https://webman.at>)

Sebastian Springer (<https://sebastian-springer.com>)

PS: In diesem Buch geht es ausschließlich um den Einsatz von KI-Tools zur Software-Entwicklung und zur Administration. Es ist uns natürlich klar, dass die neuen Entwicklungen in der KI-Branche umfassende gesellschaftliche Auswirkungen haben werden, positive wie negative. Das ist aber kein Thema in diesem Buch, ebenso wenig wie die Anwendung von KI-Werkzeugen außerhalb der IT.