

# Vorwort

Die erste Auflage dieses Buchs hat der Verlag mit der Aussage des NVIDIA-Chefs Jensen Huang beworben, wenn auch verpackt in eine Frage: »Wird in wenigen Jahren niemand mehr Code schreiben?« Wir waren damals skeptisch und hielten das für einen der üblichen Marketing-Sprüche.

Das hat sich geändert. Heute glauben wir tatsächlich, dass das traditionelle Coding keine große Zukunft mehr hat. Noch ist es nicht so weit, keine Frage. Zudem wird es weiterhin Software Developer brauchen, um KI-Tools zu bedienen. Aber so unreal viele Erwartungen in die künstliche Intelligenz sind, so real sind die Auswirkungen im Bereich des Codings: Hier findet gerade ein Umbruch statt, der seinesgleichen sucht.

Dieser Umbruch manifestiert sich auch in diesem Buch: Es kommt selten vor, dass ein IT-Buch innerhalb von 18 Monaten eine zweite Auflage erfährt und dabei in großen Teilen neu geschrieben werden muss. Genau das ist uns mit diesem Buch passiert. Natürlich war uns schon beim Verfassen der ersten Auflage klar, dass die Entwicklung von KI-Tools noch nicht zu Ende ist. Aber selbst wir haben nicht erwartet, dass sich die Werkzeuge zur Software-Entwicklung *so* schnell verbessern würden.

Der KI-Forscher Andrej Karpathy glaubt: »The hottest new programming language is English.« Oder, um den Claude-Code-Erfinder Boris Cherny zu zitieren: »Coding is solved.«

Sie müssen in Zukunft nicht mehr die Syntax von Python, JavaScript oder C beherrschen. Vielmehr müssen Sie in der Lage sein, an KI-Tools die Aufgabenstellung klar weiterzugeben. Dieser Schritt erfordert im Übrigen immer noch eine ganze Menge IT-Fachwissen! Um das eigentliche Coding, um die Dokumentation und um die Fehlersuche kümmert sich ein Sprachmodell zusammen mit einer IDE, einem Editor oder einem Command Line Interface (CLI).

## Traum und Wirklichkeit

Auch wenn die einleitenden Absätze vielleicht das Gegenteil vermuten lassen: Wir versprechen Ihnen nicht das Blaue vom Himmel, wir wollen Ihnen nichts verkaufen! Wir sehen die Probleme der KI und benennen sie in diesem Buch ganz deutlich. Es ist eine wichtige Aufgabe dieses Buchs, klarzustellen, was alles *nicht* funktioniert.

Bei allem Potenzial, das die KI in der IT-Branche hat: Aktuell arbeiten KI-Tools keineswegs fehlerfrei! Wo sind also die Probleme?

- ▶ **KI-Tools eignen sich nicht für jede Art von Code.** Am besten funktioniert KI im Coding-Mainstream, also für weitverbreitete Programmiersprachen (idealerweise *nicht* in der neuesten Version) und für alltägliche Probleme. Das hat damit zu tun, dass es für diese Art von Code das meiste Trainingsmaterial gibt. KI-Sprachmodelle sind nur so gut wie das Wissen, das sie im Rahmen des Trainings erworben haben bzw. das nachträglich mit verschiedenen Mechanismen hinzugefügt wird.

Je mehr Sie außerhalb dieses Mainstreams arbeiten, desto mehr müssen Sie die KI-Tools unterstützen, und sie können sich dann eben nicht blind auf deren Ergebnisse verlassen. Und es muss Ihnen immer bewusst bleiben: KI-Tools können Ihnen viel mechanische Arbeit abnehmen, aber »intelligent« in dem Sinne, dass sie wirklich neue Dinge erfinden, sind sie nicht. Dieser Job bleibt immer noch bei Ihnen.

- ▶ **KI-Tools funktionieren am besten, wenn sie von Menschen bedient werden, die selbst über viel IT-Wissen verfügen.** Ja, KI demokratisiert die Programmierung. Das Budget für ein KI-Abo vorausgesetzt können Personen ohne Coding-Erfahrung einfache (mitunter sogar recht komplexe) Programme selbst erstellen.

Den diesbezüglich geprägten Begriff des »Vibe Codings« sehen wir eher skeptisch (siehe [Kapitel 16](#), »Risiken und Ausblick«). Fest steht: Nur mit IT-Erfahrung sind Sie in der Lage, zielführende Prompts zu formulieren, die KI-Ergebnisse zu beurteilen und nachzubessern. Nur mit IT-Wissen können Sie Verantwortung für den von Ihnen (mit oder ohne KI-Hilfe) produzierten Code übernehmen.

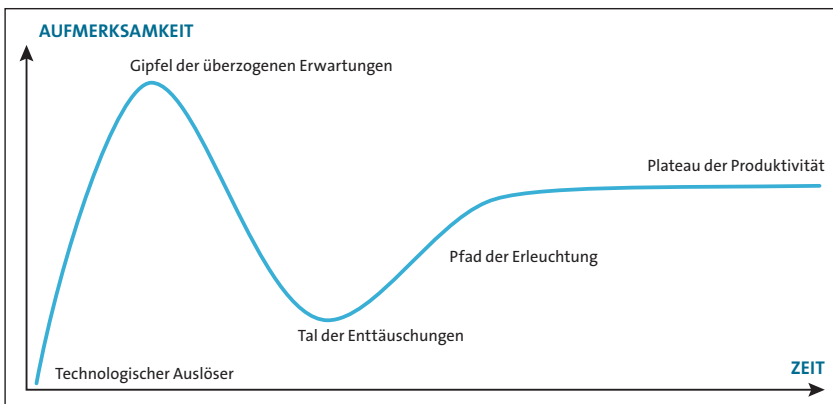
- ▶ **Software-Entwicklung ist mehr als Coding.** Die KI wird uns in Zukunft beim Coding viel Arbeit abnehmen. Aber die Kundenkommunikation, das Abwägen zwischen Technologien, die Koordination von Teams wird unter menschlicher Kontrolle bleiben. Wir glauben, dass die »human skills« wichtiger denn je sind.
- ▶ **Der KI-Markt ist unübersichtlich.** In den letzten drei Jahren wurden praktisch monatlich neue Sprachmodelle, neue Tools und neue Arbeitstechniken vorgestellt. Wenn Sie intensiv an einem Projekt arbeiten, fehlt Ihnen die Zeit, diesem ständigen Wechsel zu folgen. Welches Tool ist ideal für welchen Zweck? Welches Sprachmodell funktioniert am besten? Sind lokale Modelle die Lösung? Ersetzt der Einsatz von CLIs die Arbeit in Entwicklungsumgebungen?
- ▶ **Die Bedienung von KI-Tools ist nicht trivial.** Wenn Sie in den Fachmedien oder in der Werbung über die Nutzung von KI-Tools lesen, erscheint alles ganz einfach. Ein paar Prompts, fertig ist das neue Feature! Diesem Idealfall geht aber meist eine Menge Arbeit voran: Sie müssen die richtigen Tools auswählen, deren Bedienung erlernen, ihre Grenzen erkennen, neue Funktionen richtig anwenden. Die Fülle von Optionen, die Ihnen dabei zur Wahl stehen, ist überwältigend.
- ▶ **KI-Coding ist anstrengend:** Mit KI-Werkzeugen können Sie in kurzer Zeit mehr erledigen. Aber das hat seinen Preis. KI-Editoren können vielleicht asynchron

mit Subagenten mehrere Aufgaben auf einmal verarbeiten. Als Mensch darüber die Kontrolle zu bewahren, ist intellektuell herausfordernd. Viele Entwickler, die intensiv mit KI-Werkzeugen arbeiten und mit deren Geschwindigkeit mitzuhalten versuchen, fühlen sich nach einer Weile ausgebrannt.

## Hype oder Realität?

Wir haben während der Arbeit an diesem Buch nicht nur selbst intensivst mit KI-Tools gearbeitet, wir haben auch mit vielen anderen Entwicklerinnen 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 liefert brauchbaren Code, es löst meine Probleme. 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. Die Ausführungen sind sprachlich überzeugend, aber inhaltlich falsch. Der Code funktioniert vielleicht, ist aber ineffizient und voller Redundanz. Ihre Schlussfolgerung: KI-Tools sind unbrauchbar, ihr Einsatz verantwortungslos etc.
- ▶ **Pragmatischer Einsatz:** Nach ein paar Wochen entwickeln Sie ein gutes Gefühl dafür, welches Tool bei welcher Anwendung die besten Resultate liefert. Sie können die Grenzen besser einschätzen. Übrig bleibt ein Tool, das zwar Fehler macht, das Ihnen aber dennoch eine echte Hilfe ist.



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

## Agenten und Agentic Coding

Der größte Quantensprung der vergangenen zwei Jahre lässt sich mit »Agentic Coding« zusammenfassen. Dazu müssen wir ein wenig ausholen: Ein *Agent* ist ein Programm, das autonom, von einem KI-Tool gesteuert, Aufgaben erledigt. Sie können also ChatGPT bitten, einen Reiseplan zu erstellen und Flugverbindungen zu suchen. Das ist die traditionelle, nicht-agentische Nutzung von KI-Funktionen.

Oder Sie vertrauen Ihrem KI-Tool so weit, dass Sie ihm die Steuerung über Teile Ihres Computers anvertrauen, ihm Ihre Kreditkartendaten geben (oder Zugang zu einem Tool, das darüber verfügt oder das z. B. *booking.com* steuern darf, wo Ihre persönlichen Daten hinterlegt sind), und beauftragen das KI-Tool: »Buche den Flug, schließe eine Stornoversicherung ab.« Anfang 2026 wurde das Software-Projekt *OpenClaw* populär und zeigte gleichermaßen, welches Potenzial und welche Risiken diese Art der Automatisierung birgt.

In diesem Buch geht es um ganz anderen Agenten – solche, die für das Coding relevante Funktionen übernehmen. Dazu zählen:

- ▶ Dateien lesen, erzeugen und ändern
- ▶ Kommandos lokal ausführen (mittels bash oder PowerShell), gegebenenfalls sogar remote via SSH
- ▶ Webseiten herunterladen, REST-API-Aufrufe durchführen
- ▶ Syntaxtests durchführen (linter, Language Server Protocol)
- ▶ Code (idealerweise in einer Sandbox) ausprobieren, anhand von Unit-Tests oder anderen Test-Frameworks verifizieren, Fehler beheben
- ▶ Datenbanksysteme steuern
- ▶ GitHub, GitLab, Slack etc. administrieren, also Issues auslesen, den Fehler im Code beheben, Commit und Pull Request erstellen

Diese »Agenten« sind auf unterschiedliche Art und Weise realisiert. Moderne KI-Tools verfügen über integrierte Tools, berücksichtigen in Markdown-Dateien formulierte Anleitungen (Skills) oder steuern über eine standardisierte Schnittstelle externe Programme (Model Context Protocol, kurz MCP). Die vielen Aspekte des *Agentic Coding* sind das zentrale Thema dieses Buchs.

## Über dieses Buch

Wir haben dieses Buch in drei Teile gegliedert:

- ▶ **Grundlagen und erste Beispiele:** In Teil I geben wir Ihnen einen Überblick über KI-Grundlagen und -Werkzeuge. Wir zeigen anhand von Praxisbeispielen verschiedene Nutzungsformen von Coding mit KI: Recherche im Chat, automatische Vollständigkeit, Pair Programming und Agentic Coding.

Ganz nebenbei erklären wir eine Fülle von Fachbegriffen und Abkürzungen: CLI-Tools, Inferenz, Kontext, MCP, MoE, Quantisierung, Sandboxing, Skills, Slash-Kommandos, Sprachmodelle (LLMs), Subagenten und viele mehr.

- ▶ **Coding ist nur der Anfang!** KI-Tools helfen auch beim Debugging, bei der Verbesserung vorhandenen Codes (Refactoring), beim Erstellen von Unit-Tests, beim Verfassen von Dokumentation, beim Design von Datenbanken und bei SQL-Kommandos, beim Scripting, bei der Administration und sogar bei der Suche nach Sicherheitslücken (Penetration Testing). Teil II des Buchs beschäftigt sich mit diesen Themen.
- ▶ **KI für Profis:** Teil III geht in voneinander losgelösten Kapiteln auf eine Menge Spezialthemen ein. Anstatt auf kommerzielle KI-Dienste zurückzugreifen, können Sie diese **lokalen Sprachmodelle** auf Ihrem Rechner oder auf einem dedizierten Server in Ihrer Firma ausführen. Wir loten aus, was das bringt und was es kostet.

Mit **CLI-Tools** (Command Line Interfaces) wie Claude Code oder Codex CLI erledigen Sie die Software-Entwicklung im Terminal.

Mit dem **Model Context Protocol (MCP) und Skills** kann Ihr KI-Tool externe Programme und Werkzeuge steuern und in den Coding-Workflow integrieren.

**Retrieval Augmented Generation (RAG)** ergänzt Sprachmodelle um eigene Daten. Daraus ergibt sich eine Menge zusätzlicher Anwendungsmöglichkeiten, unter anderem die Integration von tagesaktuellen Daten oder die automatisierte Umwandlung von Text-Prompts in SQL-Queries.

Sie können KI-Funktionen nicht nur als Unterstützung beim Programmieren verwenden, Sie können Sprachmodelle auch *in* Ihrem Code nutzen. Wir geben Ihnen eine erste Vorstellung, wie dafür gedachte **APIs** funktionieren.

Das Buch endet mit einem Kapitel, in dem wir auf die **Risiken** der neuen, schönen KI-Welt eingehen. Und deren gibt es viele: Halluzinationen, Datenschutzprobleme, zusätzliche Abhängigkeiten und Kosten, der Verlust der Kontrolle über den eigenen Code, das Verlernen von Fähigkeiten, ethische Bedenken.

Open-Source-Projekte stöhnen über unsinnige, KI-generierte Fehlerberichte. Ebenfalls aus dem Open-Source-Segment kommt der Vorwurf, dass heute zwar Compiler, Bibliotheken usw. kostenlos verfügbar sind, der Zugang zu KI-Tools bzw. zu hochwertigen Sprachmodellen aber nicht bzw. nur in sehr eingeschränktem Umfang. Bei *OpenAI* steht zwar *Open* im Firmennamen, aber in Wirklichkeit sind viele Aspekte der KI-Entwicklung *Closed Source*. Damit besteht die Gefahr einer Zweiklassen-Gesellschaft von Codern: solche, die oder deren Firmen sich KI-Unterstützung leisten können, und alle anderen.

Umgekehrt fragen sich KI-Firmen, wie sie längerfristig Geld verdienen wollen bzw. können. Aktuell profitieren vor allem Rechenzentren und Hardware-Firmen vom KI-Boom, allen voran natürlich NVIDIA. Die Anbieter von KI-Diensten schreiben hin-

gegen unvorstellbare Verluste. Der ganze IT-Markt zeigt unverkennbare Zeichen einer Blase, die jeden Moment platzen kann.

Niemand weiß, wie der KI-Markt nach einer Konsolidierung aussehen wird. Sicher sind wir uns nur in einem Punkt: Coding mit KI ist hier, um zu bleiben. Anders formuliert: Der KI-Geist ist aus der Flasche, er kehrt nicht mehr dorthin zurück.

### Chancen sehen und nutzen

Sie können das Glas als halb leer oder halb voll betrachten. Sie können Angst um Ihren Job in der IT-Branche haben – oder die Chance auf eine vollkommen neue, verblüffend effiziente Art der Software-Entwicklung sehen. Wir raten Ihnen zu Letzterem. Coding ohne KI wird schon bald nicht mehr vorstellbar sein.

Lassen Sie sich von unseren Beispielen inspirieren, und suchen Sie nach Wegen, KI-Werkzeuge sinnvoll in Ihren Arbeitsalltag zu integrieren. Seien Sie dabei, wenn sich hier neue Türen auftun! Werden Sie vom Coder oder Software Developer zum Software Engineer!

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 künstliche Intelligenz umfassende gesellschaftliche Auswirkungen haben wird, positive wie negative. Das ist aber kein Thema in diesem Buch, ebenso wenig wie die Anwendung von KI-Werkzeugen außerhalb der IT.

### Beispieldateien

Zu einigen (wenigen) Kapiteln dieses Buchs gibt es Beispieldateien. Diese können Sie hier herunterladen:

<https://www.rheinwerk-verlag.de/6168>