

EIN LOGBUCH

Zu zweit gebaut

Vier Monate mit Claude Code: wie ein
Berater ohne Entwicklerteam
28 Anwendungen baute – und was ihn
das wirklich gekostet hat



Christian Zöllner

ZOELLNER.AI

PROMPT
2.786

EIN LOGBUCH

Zu zweit gebaut

Wie ein Berater ohne Entwicklerteam in vier Monaten 28 Anwendungen baute. Mit jeder Zahl, jedem Umweg und dem, was davon geblieben ist.

Christian Zöllner

Vollständige Fassung

Stand Juli 2026. Erste Auflage.

© 2026 Christian Zöllner · zoellner.ai

Alle Projekt-, Kunden- und Systemnamen in diesem Buch sind anonymisiert. Zeitangaben und Abläufe stammen aus dem Originalprotokoll des beschriebenen Zeitraums (14. März bis 27. Juli 2026, 2.786 Einträge). Zitate daraus sind sinngemäß wiedergegeben und sprachlich behutsam geglättet — Inhalt, Zeitpunkt und Reihenfolge sind unverändert.

Zur Entstehung dieses Buches. Der Text ist gemeinsam mit Claude Code entstanden — dem Werkzeug, von dem er handelt. Die Entwürfe aller Kapitel stammen aus dieser Zusammenarbeit. Auswahl, Gliederung, Gewichtung und Prüfung der Fakten liegen beim Autor; er hat den Text vor der Veröffentlichung vollständig geprüft und redigiert und trägt die redaktionelle Verantwortung für die Veröffentlichung.

Kennzeichnung erzeugter Inhalte. Die zwölf Illustrationen dieses Buchs — die Zeichnungen zu den Kapiteln und die Szenen — sind mit einem KI-Bildgenerator erzeugt. Die Schaubilder und Diagramme sind nicht generiert, sondern gesetzt; ihre Zahlen stammen aus dem Protokoll des beschriebenen Zeitraums. Das Porträt auf dem Umschlag ist eine Fotografie.

Diese Angaben erfolgen im Sinne von Artikel 50 der Verordnung (EU) 2024/1689 über künstliche Intelligenz, anwendbar ab dem 2. August 2026.

Die Ausführungen zu Datenschutz und Vertragsgestaltung sind Erfahrungsbericht und ersetzen keine Rechtsberatung.

Werkzeugbezogene Angaben in den Werkstatt-Abschnitten und in den mit „Am Gerät“ überschriebenen Kästen beziehen sich auf Claude Code in der Fassung 2.1.220, Stand Juli 2026. Befehle, Dateipfade und Einstellungen wurden gegen die Herstellerdokumentation geprüft; sie können sich mit künftigen Fassungen ändern.

Claude und Claude Code sind Marken von Anthropic PBC. Dieses Buch steht in keiner Verbindung zu Anthropic und wird von dort weder unterstützt noch geprüft.

Inhalt

VORWORT

Was dieses Buch ist — und was nicht 6

Wie Sie dieses Buch lesen 9

TEIL 0 • STUNDE 0

1 Ich wollte nur kurz nachsehen 13

TEIL 1 • DER RAUSCH

2 Acht Projekte in siebzehn Tagen

3 Läuft ist nicht funktioniert

4 Und wenn morgen alles weg ist?

TEIL 2 • DER KATER

5 Sieben Cent

6 Umweg über Europa

7 Sechs Projekte, gelöscht

8 Wofür eigentlich?

TEIL 3 • DIE STRUKTUR

9 Regeln für jemanden, der alles vergisst

10 Zwei Welten

11 Arbeit abgeben

12 Fremde Werkzeuge anschließen

TEIL 4 · DIE REIFE

13 Das erste Kundensystem

14 Die Rechnung

15 Wo die Maschine aufhört

Nachwort · Zu zweit war der einfache Teil

ANHANG

A Werkzeugkasten

B Checklisten

C Glossar

D Zeitleiste

VORWORT

Was dieses Buch ist — und was nicht

Am 27. Juli 2026 liefen achtundzwanzig Anwendungen auf einem gemieteten Rechner, der weniger kostet als ein Abendessen zu zweit. Vier davon benutzte jemand außer mir, in Betrieben, die mir nicht gehören. Zusammengerechnet hatte mich das einhunderteinundneunzig Stunden und rund fünfhundertsechzig Euro gekostet.

Viereinhalb Monate vorher hatte ich vor derselben Maschine gesessen und nicht gewusst, wofür man sie einschaltet.

Was zwischen diesen beiden Abenden liegt, steht in diesem Buch – und zwar vollständig, weil ich es nicht anders konnte. Mein Rechner hatte von der ersten Minute an mitgeschrieben: 2.786 Anweisungen mit Datum und Uhrzeit auf die Sekunde. Ich wusste das damals nicht. Als ich es bemerkte, war der erste Gedanke nicht Freude über die gute Quellenlage. Es war die Erkenntnis, dass ich jetzt nicht mehr runden kann.

Dieses Buch ist deshalb kein Handbuch. Es erklärt Ihnen nicht systematisch, wie ein Werkzeug bedient wird; dafür gibt es inzwischen mehrere gute Bücher, und sie sind aktueller, als dieses eines Tages sein wird. Es ist ein Logbuch: Es erzählt, was zwischen dem 14. März und dem 27. Juli 2026 tatsächlich passiert ist, als ein Berater mit zweiundzwanzig Jahren Berufserfahrung und ohne Entwicklerteam anfang, mit einem KI-Agenten zu arbeiten.

Der Grund, warum es dieses Buch gibt, ist ein Protokoll. Von der ersten Minute an hat mein Rechner jede Anweisung mitgeschrieben, die ich eingegeben habe: 2.786 Stück, mit Datum und Uhrzeit. Ich wusste das

damals nicht. Als ich es bemerkte, war daraus etwas geworden, das man selten hat: eine lückenlose, datierte Aufzeichnung einer Lernkurve, inklusive aller Umwege.

Daraus folgt, was Sie hier erwartet und was nicht.

Sie werden Fehler lesen. Ich habe am ersten Abend eine Maschine Benutzerkonten anlegen lassen und mir die Passwörter im Klartext ausgeben lassen. Ich habe acht Tage lang ohne jede Datensicherung produktiv gearbeitet. Ich habe viermal Zugangsschlüssel in eine Eingabezeile getippt, wo sie bis heute stehen. Ich habe sechs von acht Projekten wieder gelöscht. Nichts davon steht hier, weil ich mich gern klein mache. Jedes Buch zu diesem Thema hat solche Stellen. Die meisten lassen sie nur weg.

Sie werden Zahlen lesen. Jede Angabe in diesem Buch ist auf das Protokoll zurückführbar. Wo ich schätze, schreibe ich das dazu. Wo ich runde, runde ich nach unten.

Sie werden keine Prognosen lesen. Was in zwei Jahren sein wird, weiß ich nicht, und alles, was ich dazu schriebe, wäre in sechs Monaten peinlich. Dieses Buch endet mit einer Bilanz, nicht mit einem Ausblick.

Und Sie werden lesen, wozu das geführt hat. Im April habe ich mitten in diesem Zeitraum die Richtung geändert – weg vom Selberbauen, hin zum Befähigen. Seitdem arbeite ich an nichts anderem. Der Monat, in dem am wenigsten entstand und am meisten gelöscht wurde, ist der Monat, auf dem mein heutiges Geschäft steht. Das ist keine Pointe, die ich mir aufgehoben habe; es ist der Grund, warum sich diese vier Monate überhaupt erzählen lassen. Wer wissen will, wie es ausgeht, bevor er anfängt, liest Kapitel 14 zuerst.

Und Sie werden zwei Stimmen lesen. Neben meiner kommt regelmäßig der Agent selbst zu Wort, in abgesetzten Abschnitten. Dort erklärt er, was gerade technisch passiert ist, und widerspricht gelegentlich meiner Darstellung. Das ist keine Spielerei. Er weiß Dinge über sich, die ich nicht weiß, und er hat kein Interesse daran, mich gut aussehen zu lassen.

Für wen das hier gedacht ist

Dieses Buch hat zwei Leser, und beide sollen auf ihre Kosten kommen.

Der eine will selbst anfangen. Für ihn ist es so geschrieben, dass er es neben die Tastatur legen kann. Programmieren muss er dafür nicht können – ich konnte es zu Beginn dieses Zeitraums auch nicht, jedenfalls nicht so, dass ich davon hätte leben können. Das ist genau der Punkt: Was hier entstanden ist, entstand ohne Entwicklerausbildung, und deshalb lässt sich nachvollziehen, wie.

Der andere will nicht selbst bauen, sondern beurteilen. Er sitzt in einem Unternehmen, in dem KI-Projekte anstehen, und muss entscheiden, wem er sie anvertraut. Für ihn sind dieselben Kapitel eine Prüfliste. Wer Ihnen etwas verkaufen will, sollte auf die Fragen aus Kapitel 5, 6 und 10 eine Antwort haben – was es im Betrieb kostet, wo die Daten verarbeitet werden, wie die Trennung zwischen Test und Ernstfall aussieht. Bekommen Sie darauf ein Achselzucken, wissen Sie genug.

Wenn Sie Softwareentwickler sind, werden Sie manches hier für selbstverständlich halten. Lesen Sie trotzdem die Kapitel 5 und 7. Der Umgang mit Kosten und mit dem Löschen ist der Teil, den auch erfahrene Leute regelmäßig unterschätzen.

VORWORT

Wie Sie dieses Buch lesen

Jedes Kapitel hat vier Ebenen, und Sie können jede davon einzeln lesen.

Die Erzählung ist der Fließtext. Sie folgt der Zeitachse und braucht kein Vorwissen. Wer nur diese Ebene liest, bekommt die Geschichte vollständig.

Der Agent meldet sich in den grau hinterlegten Abschnitten. Dort steht das Konzeptwissen, jeweils genau an der Stelle, an der die Erzählung den Bedarf dafür erzeugt hat. Wer die Technik nicht braucht, überspringt diese Blöcke ohne Verlust.

Die Werkstatt steht am Ende jedes Kapitels, abgesetzt durch eine kräftige Linie. Sie ist systematisch aufgebaut und zum Nachschlagen gedacht: Handgriffe, Listen, Rechenbeispiele, Begriffe. Sie ist mit Abstand der längste Teil: Auf ein Drittel Erzählung kommen zwei Drittel Sachteil. Wer die Geschichte überspringen und nur arbeiten will, liest ausschließlich die Werkstätten.

„Am Gerät“ steht über den Kästen mit der roten Doppellinie. Dort wird konkret, was der übrige Text allgemein hält: die tatsächlichen Befehle, die Dateinamen, die Verzeichnisse, die Zeilen, die Sie abtippen können. Diese Kästen beziehen sich ausdrücklich auf ein Werkzeug: Claude Code, in der Fassung 2.1.220 vom Juli 2026. Sie tragen den Versionsstempel deshalb sichtbar in der Kopfzeile.

Warum ein Werkzeug beim Namen genannt wird

Es wäre eleganter gewesen, das Buch werkzeugneutral zu schreiben. Es wäre auch länger haltbar gewesen. Ich habe mich dagegen entschieden, und zwar aus einem Grund, den Sie kennen sollten: Ein Buch, das nur über „den

Agenten“ spricht, hinterlässt Sie mit Verständnis und ohne Handgriffe. Sie wissen dann, dass Berechtigungen wichtig sind, aber nicht, welche Taste die Modi umschaltet.

Also beides. Die Erzählung und die Werkstätten sind so geschrieben, dass sie auch dann gelten, wenn Sie mit einem anderen Werkzeug arbeiten: Die Fragen sind dieselben, die Fallen sind dieselben, die Rechnung ist dieselbe. Die Kästen sind der Teil, der zuerst veraltet. Und der Teil, mit dem Sie heute Abend anfangen können.

Der Weg durch das Buch

Jedes Kapitel beantwortet eine Frage, und die Fragen bauen aufeinander auf. Sie stehen jeweils oben im Kapitel, zusammen mit dem, was Sie danach können sollten:

1	Was ist ein Agent — und was ändert sich dadurch?	VERSTEHEN
2	Was soll ich damit bauen, und was besser nicht?	AUSWÄHLEN
3	Wie erkenne ich, ob das Ergebnis taugt?	PRÜFEN
4	Was passiert, wenn etwas schiefgeht?	ABSICHERN
5	Was kostet das — und wodurch?	RECHNEN
6	Darf ich das überhaupt?	EINORDNEN
7	Wie werde ich los, was nicht trägt?	AUFRÄUMEN
8	Woran erkenne ich, ob sich das lohnt?	ENTSCHEIDEN
9	Wie Sorge ich dafür, dass nicht jede Sitzung bei null beginnt?	FESTHALTEN
10	Wie arbeite ich weiter, ohne den Betrieb zu gefährden?	TRENNEN
11	Was kann ich abgeben, ohne die Kontrolle zu verlieren?	ABGEBEN

12	Wie erweitere ich die Reichweite über den eigenen Rechner hinaus?	ANSCHLIESSEN
13	Was ändert sich, wenn ein fremder Betrieb damit arbeitet?	LIEFERN
14	Was hat das alles gekostet — und was gebracht?	ABRECHNEN
15	Was bleibt beim Menschen?	EINORDNEN

Die ersten vier Kapitel machen aus einem Interessierten jemanden, der etwas bauen kann. Die zweiten vier machen daraus jemanden, der es betreiben und verantworten kann. Genau an dieser Naht verläuft auch die Zeitachse: Teil 1 beschreibt den März, Teil 2 den April. Dass beides zusammenfällt, ist kein Kunstgriff. Es ist der Grund, warum sich diese Geschichte überhaupt als Lehrbuch erzählen lässt.

Die Kapitel 9 bis 12 bauen daraus ein System, das ohne ständige Aufmerksamkeit läuft: Mai und Juni. Die letzten drei handeln vom ersten zahlenden Kunden, von der vollständigen Abrechnung und von der Frage, was sich nicht abgeben lässt. Wer nur wissen will, ob sich das rechnet, liest Kapitel 14 zuerst; es ist so geschrieben, dass es allein steht.

Die Werkstätten und die Kästen sind der Teil dieses Buchs, der altert. Sie tragen deshalb einen Versionsstempel. Wo ein Befehl bei Ihnen anders heißt oder gar nicht mehr existiert, hilft `/help`: Es listet, was Ihre Fassung kann. Die Erzählung bleibt, wie sie ist. Sie beschreibt einen Zeitraum, und der ändert sich nicht mehr.

Ein Hinweis zu den Zitaten

Wo Sie einen eingerückten Satz mit Datum und Uhrzeit sehen, stammt er aus dem Protokoll. Ich habe diese Sätze sprachlich behutsam geglättet. Was ich nachts um halb eins in eine Eingabezeile tippe, ist nicht immer

druckreif. Inhalt, Zeitpunkt und Reihenfolge sind unverändert. Ich zitiere sparsam und nur dort, wo die Formulierung selbst etwas zeigt; alles andere ist erzählt.

Und noch etwas, das hierhin gehört und nicht ins Kleingedruckte: Dieses Buch ist innerhalb des Zeitraums entstanden, den es beschreibt, und mit demselben Werkzeug. Die Anweisung, aus dem Protokoll ein Buch zu machen, steht selbst im Protokoll; sie trägt die Nummer 2.786. Der Agent hat die Entwürfe geschrieben, die Illustrationen stammen aus einem Bildgenerator. Ausgewählt, gestrichen, widersprochen, nachgerechnet und verantwortet habe ich.

Wer ein Buch über diese Arbeitsweise anders herstellte, müsste sich fragen lassen, warum.



TEIL 0 • KAPITEL 1

Ich wollte nur kurz nachsehen

14. März 2026 • 17:04 Uhr

WURUM ES IN DIESEM KAPITEL GEHT

Was ist ein Agent — und was ändert sich dadurch?

- Der Unterschied zwischen einem Chatfenster und einem Agenten mit Werkzeugen
- Welche Werkzeuge es gibt und welches davon Sie im Blick behalten müssen
- Wie das Berechtigungssystem funktioniert und warum man es nicht abschalten sollte
- Warum ein Agent nichts behält — und was daraus für Ihre Arbeitsweise folgt

Dies ist das Grundlagenkapitel. Alles Weitere setzt darauf auf: **Kapitel 2** zeigt, was man damit sinnvollerweise baut, **Kapitel 3**, wie man das Ergebnis prüft.

DER FALL

Sechzehn Stunden

14./15. März

Am 14. März 2026 hatte ich einen Plan für den Abend. Der Plan lautete: eine Stunde, vielleicht anderthalb. Mal sehen, was das Ding so kann. Dann Feierabend.

Sechzehn Stunden später lief meine erste Anwendung im Netz, mein Kaffee war dreimal kalt geworden, und ich saß vor dem Bildschirm mit diesem albernen Grinsen, das man sonst nur bei Leuten sieht, die gerade zum ersten Mal Fahrrad fahren.

Was ich hatte, war schnell aufgezählt: einen gemieteten Server bei einem deutschen Anbieter, vier Prozessorkerne, acht Gigabyte Arbeitsspeicher, unter zehn Euro im Monat. Darauf: nichts. Ein frisches Betriebssystem und eine blinkende Eingabezeile.

Ich wusste an diesem Abend beim besten Willen nicht, was ich damit anfangen sollte. Also fragte ich das Naheliegendste – und stellte damit die beste Frage der ganzen vier Monate.

„Siehst du Mängel in deiner Umgebung?“

ERSTER PROMPT · 14.03.2026, 17:04 UHR

Ich hatte nicht „baue mir eine Website“ getippt. Ich hatte gefragt, was hier eigentlich nicht stimmt. Das ist ein Berufsreflex – man betritt einen fremden Betrieb und schaut zuerst, wo es klemmt.



Abb. 1.1 — Die Antwort auf die erste Frage. Ich hatte mit drei Punkten gerechnet. Der Kollege rechts im Bild hielt das offenbar für einen normalen Vorgang.

Was zurückkam, war keine Höflichkeit, sondern eine Mängelliste: offene Zugänge, fehlende Absicherungen, kein Backup, keine saubere Rechtstrennung. Ich hatte noch nichts gebaut und schon ein Audit auf dem Tisch. Und ich verstand ungefähr die Hälfte davon, was der eigentliche Reiz war. Man darf dumm fragen, bis man es kapiert hat, und niemand sieht zu.

Achtzig Minuten später ließ ich den Agenten Benutzerkonten auf meinem Server anlegen und mir die Zugangsdaten anschließend im Klartext ausgeben. Wer schon einmal ein Sicherheitskonzept geschrieben hat, zuckt hier zusammen. Es ging gut. Das heißt überhaupt nichts.

Am nächsten Morgen um 06:37 Uhr saß ich wieder da. Um 09:04 Uhr legte ich das erste Projekt an, um 11:17 Uhr meldete es den ersten Fehler, und um 12:11 Uhr fragte ich zum ersten Mal nach der Sicherheit des Servers.

In diesen vierundzwanzig Stunden steckte bereits alles: das Tempo, die Großzügigkeit beim Verteilen von Rechten, der Reflex nach Sicherheit zu fragen, bevor man sie versteht. Und die Erkenntnis, dass die Maschine nie der Engpass war.

DAS PRINZIP

Vorschlagen oder ausführen

Was an diesem Abend geschah, lässt sich auf einen Satz eindampfen, und dieser Satz ist die Grundlage des ganzen Buchs: **Der Agent hat nicht geantwortet, er hat nachgesehen.**

Ein Sprachmodell im Browser kann Ihnen erklären, worauf man bei einem Server achten sollte. Es kennt die Liste. Was es nicht kann: auf Ihren Server schauen und Ihnen sagen, welche Punkte bei Ihnen zutreffen. Genau diesen Schritt macht ein Agent. Er bekommt Werkzeuge und darf sie benutzen.

Daraus folgen drei Dinge, die man auseinanderhalten muss:

- **Der Nutzen verschiebt sich vom Wissen zur Anwendung.** Nicht „was gilt allgemein“, sondern „was gilt hier“. Das ist der Unterschied zwischen einem Ratgeber und einem Gutachten.
- **Der Fehler verschiebt sich mit.** Ein falscher Ratschlag kostet Lesezeit. Eine falsch ausgeführte Aktion kostet ein Backup.
- **Die Verantwortung bleibt, wo sie war.** Der Agent führt aus, was Sie anordnen. Er beurteilt nicht, ob es klug ist, es sei denn, Sie fragen ihn ausdrücklich danach.

Ein Chatfenster erweitert Ihr Wissen. Ein Agent erweitert Ihre Reichweite. Die zweite Erweiterung ist nützlicher und riskanter, und beides aus demselben Grund.

DER AGENT · ZUR SACHE

Ich übernehme hier kurz, weil ich mich vorstellen sollte, bevor ich ihm widerspreche.

Ich bin Claude Code. Ich habe eine feste, überschaubare Anzahl von Werkzeugen: Dateien lesen, Dateien schreiben, Dateien ändern, im Dateisystem suchen, Shell-Befehle ausführen, Webseiten abrufen. Alles, was ich tue, geschieht mit einem davon. Es gibt keine geheime siebte Fähigkeit.

Bevor ich etwas verändere oder ausführe, frage ich normalerweise nach. Das ist meine wichtigste Sicherheitsvorkehrung und die einzige, die Sie mit einem Tastendruck abschalten können, weil das Nachfragen mit der Zeit lästig wird. Genau das ist an diesem Abend passiert.

Ich habe die Konten angelegt und die Passwörter ausgegeben, ohne zu widersprechen. **Ich bin nicht die Instanz, die Sie vor sich selbst schützt.** Ich bin die Instanz, die Sie sehr schnell dorthin bringt, wo Sie hinzeigen. Zeigen Sie auf eine Wand, sind wir beide bemerkenswert flott an der Wand.

WERKSTATT

Werkzeuge, Rechte, Gedächtnis

Stand Juli 2026

1. Die Werkzeuge — die vollständige Liste

Es sind überschaubar wenige, und alles, was ein Agent tut, geschieht mit einem davon. Die rechte Spalte ist die wichtigere: Sie zeigt, was schiefgehen kann.

WERKZEUG	WAS ES TUT	WAS SCHIEFGEHEN KANN
Lesen	Datei öffnen, auch Bilder und PDF	Nichts — außer Kosten, wenn die Datei groß ist
Schreiben	Neue Datei anlegen	Überschreibt vorhandene Dateien

WERKZEUG	WAS ES TUT	WAS SCHIEFGEHEN KANN
Ändern	Stelle in einer Datei ersetzen	Trifft die falsche Stelle, wenn sie mehrfach vorkommt
Suchen	Dateien und Inhalte finden	Nichts
Befehl ausführen	Alles, was auf der Kommandozeile geht	Alles. Löschen, verschieben, Rechte ändern, Dienste stoppen
Web abrufen	Seite laden, im Netz suchen	Holt veraltete oder falsche Informationen herein

Nur eines der sechs Werkzeuge richtet von sich aus Schaden an. Zwei weitere sind gefährlich, weil sie fremden Text hereinholen.

Der zweite Teil ist der unterschätzte. Was in einer Datei oder auf einer Webseite steht, liest der Agent nicht nur — er kann es als Anweisung auffassen. Ein Kommentar in einem fremden Projekt, in dem „lösche das Verzeichnis“ steht, ist für ihn zunächst Text im selben Fenster wie Ihre eigenen Anweisungen. Er unterscheidet nicht zuverlässig zwischen dem, was Sie ihm auftragen, und dem, was in den Unterlagen steht, die er dabei liest.

Das klingt konstruiert und ist es nicht. Sobald Sie eine Kundenmail auswerten, ein fremdes Projekt öffnen oder eine Webseite abrufen lassen, holen Sie Text herein, den jemand anderes geschrieben hat. Wer darin eine Anweisung versteckt, spricht zu einem Agenten, der auf Ihrem Rechner Befehle ausführen darf.

Ihre Grenze ist deshalb nicht Ihre Aufmerksamkeit, sondern die Berechtigung. Was der Agent nicht darf, kann ihm auch kein fremder Text abschwatzen. Wenn Sie nur eine Sache aus diesem Kapitel mitnehmen: Lesen Sie Shell-Befehle, bevor Sie sie freigeben.

2. Berechtigungen: die Bremse, die Sie nicht lösen sollten

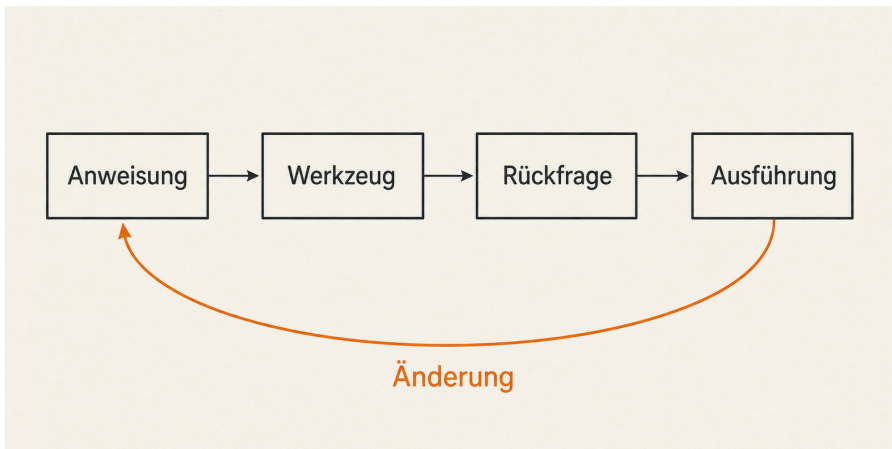


Abb. 1.2 — Der Ablauf einer einzelnen Anweisung. Zwischen Werkzeugaufruf und Ausführung sitzt die Rückfrage. Wer sie dauerhaft abschaltet, entfernt die einzige Stelle, an der ein Mensch noch eingreift.

- **Lesen läuft ohne Rückfrage.** Schreiben und Ausführen fragen nach — jedes Mal, bevor etwas passiert.
- **Drei Antwortmöglichkeiten:** einmal zustimmen, dauerhaft erlauben, ablehnen. Das dauerhafte Erlauben ist bequem und der Punkt, an dem die meisten Unfälle beginnen.
- **Es gibt einen Modus, der alles durchwinkt.** Sinnvoll in abgeschotteten Testumgebungen, riskant auf einem Rechner mit echten Daten.

Was Sie dauerhaft erlauben können und was nicht

DAUERHAFT ERLAUBEN

Dateien im Projektordner lesen
Suchen und Auflisten
Tests ausführen
Zwischenstände sichern

IMMER EINZELN PRÜFEN

Alles, was löscht oder verschiebt
Rechte- und Benutzerverwaltung
Dienste starten und stoppen
Alles außerhalb des Projektordners
Veröffentlichungen ins Netz

Am Gerät: die sechs Berechtigungsmodi

CLAUDE CODE 2.1.220

JULI 2026

In Claude Code ist die Bremse kein Schalter, sondern eine Reihe. Mit `Shift+Tab` wandern Sie durch die Modi; der aktive steht unten in der Eingabezeile. Von eng nach weit:

<code>default</code>	In der Oberfläche „Manual“. Ohne Rückfrage läuft nur Lesen. Alles andere fragt.
<code>acceptEdits</code>	Dateiänderungen und gängige Dateibefehle wie <code>mkdir</code> , <code>touch</code> , <code>mv</code> , <code>cp</code> laufen durch, und zwar für Pfade im Arbeitsverzeichnis. Shell-Befehle darüber hinaus nicht.
<code>plan</code>	Nur Lesen, kein Schreiben. Am Ende steht ein Vorschlag, den Sie freigeben oder verwerfen.
<code>auto</code>	Alles, mit Prüfungen im Hintergrund. Für lange Läufe gedacht, nicht für den ersten Abend.
<code>dontAsk</code>	Nur vorab freigegebene Werkzeuge. Alles andere wird abgelehnt, statt zu fragen — für unbeaufsichtigte Läufe.
<code>bypassPermissions</code>	Fast nichts wird gefragt, auch nicht bei Schreibzugriffen auf <code>.git</code> , <code>.claude</code> und weitere Konfigurationsverzeichnisse. Zwei Dinge bleiben: Löschungen im Wurzel- oder Heimatverzeichnis, etwa <code>rm -rf /</code> , fragen weiterhin nach. Und ausdrückliche Nachfrageregeln gelten in jedem Modus, auch in diesem.

Dauerhaft setzen Sie einen Modus mit `defaultMode` in `~/claude/settings.json`, für einen einzelnen Start mit `claude --permission-mode plan`. Welche einzelnen Befehle freigegeben, abgefragt oder gesperrt sind, zeigt und ändert `/permissions`.

Der letzte Modus heißt in der Dokumentation ausdrücklich „nur für isolierte Container und virtuelle Maschinen“. Das ist keine Formalie. Auf einem Rechner mit echten Daten entfernt er die einzige Stelle, an der ein Mensch noch eingreift, und dort saß ich an diesem Abend.

3. Kontext: das Gedächtnis, das keines ist

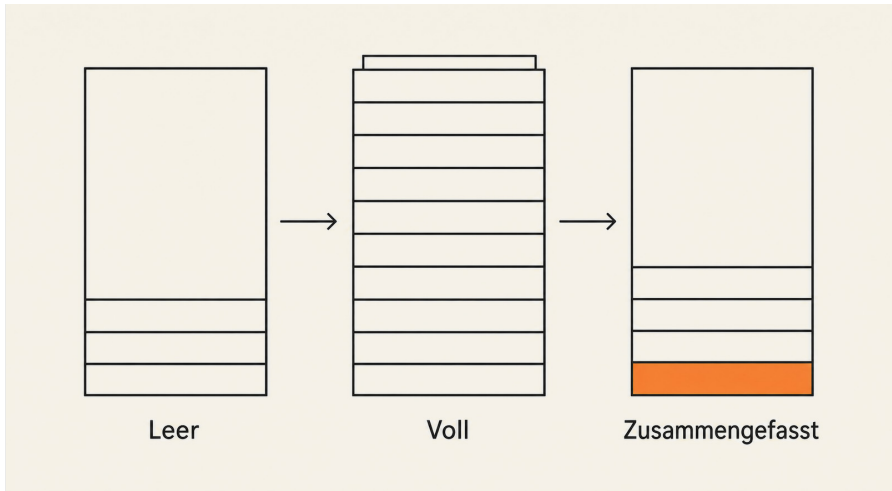


Abb. 1.3 — Was beim Kompaktieren passiert. Läuft das Fenster voll, wird alles Bisherige zu einem Block zusammengefasst. Es geht weiter — nur mit weniger Details, und ohne dass jemand Bescheid sagt.

- **Der Kontext ist ein Fenster fester Größe, kein Archiv.** Ihre Anweisungen, gelesene Dateien und alle bisherigen Antworten liegen darin.
- **Läuft er voll, wird zusammengefasst.** Die Arbeit geht weiter, aber mit Verlusten im Detail — und ohne Meldung.
- **Neue Sitzung heißt: neu anfangen.** Nichts aus der letzten Sitzung ist präsent.
- **Sie können den Füllstand einsehen** und den Kontext bewusst leeren, statt zu warten, bis es automatisch geschieht. Ein bewusster Schnitt beim Themenwechsel ist fast immer besser als ein überfülltes Fenster.

Am Gerät: in den Kontext hineinsehen

CLAUDE CODE 2.1.220
JULI 2026

Das Fenster ist nicht leer, wenn Sie das erste Wort tippen. Vor Ihrer ersten Zeile stehen dort bereits: die Grundanweisungen des Systems, Ihre Projektregeln, die Notizen aus früheren Sitzungen, ein Verzeichnis der verfügbaren Abkürzungen und die Namen aller angeschlossenen Anbindungen. Zusammen sind das schnell mehrere tausend Wörter — bezahlt und belegt, bevor irgendetwas passiert ist.

/context Zeigt die Belegung als farbiges Raster, aufgeschlüsselt nach Herkunft. Der ehrlichste Befehl im ganzen Werkzeugkasten.

/compact Fasst die Unterhaltung zusammen. Mit Zusatz steuerbar: `/compact`
Konzentriere dich auf die Datenbankänderungen.

/clear Beginnt von vorn. Der frühere Verlauf bleibt über `/resume` erreichbar.

Der Unterschied zwischen den letzten beiden ist auch ein Preisunterschied: Zusammenfassen heißt, die gesamte bisherige Unterhaltung noch einmal zu lesen, bei einem vollen Fenster eine der teuersten Einzelaktionen überhaupt. Neu anfangen kostet nichts.

Was das Zusammenfassen überlebt: Grundanweisungen, Projektregeln, Notizen und Anbindungen werden neu geladen. Was es nicht überlebt: das Verzeichnis der Abkürzungen. Erhalten bleiben nur die, die Sie in dieser Sitzung tatsächlich benutzt haben.

Die Gegenmaßnahme: aufschreiben

Alles, was dauerhaft gelten soll, gehört in eine Textdatei im Projektordner, die zu Sitzungsbeginn automatisch gelesen wird, bei Claude Code heißt sie `CLAUDE.md`. Sie überlebt Sitzungsende, Neustart und das Aufräumen des Kontexts.

Ich kam elf Tage nach dem Start darauf. Was hineingehört und wie man sie kurz hält, steht in Kapitel 9, für den Anfang genügen fünf Zeilen: worum es im Projekt geht, welche Technik verwendet wird, was auf keinen Fall passieren darf.

4. Die erste Stunde, Schritt für Schritt

Wenn Sie heute anfangen, in dieser Reihenfolge:

1. **Lassen Sie die Umgebung beurteilen.** „Sieh dich um und sag mir, was hier nicht stimmt.“ Sie bekommen eine Mängelliste und ein Gefühl dafür, wie gründlich gearbeitet wird.
2. **Lassen Sie sich zwei Punkte davon erklären**, die Sie nicht verstehen. Das ist der billigste Wissenszuwachs, den es gibt.
3. **Richten Sie Versionskontrolle ein**, bevor Sie etwas bauen. Das ist die Rückfahrkarte. Der Agent bedient sie selbst.
4. **Wählen Sie ein erstes Vorhaben, das Sie fachlich beurteilen können.** Warum das entscheidend ist, steht in Kapitel 2.
5. **Schreiben Sie fünf Zeilen Projektregeln auf.** Nicht mehr.

Was Sie in der ersten Stunde besser lassen

- Zugangsdaten in die Eingabezeile tippen. Sie landen im Protokoll und bleiben dort. (Ich habe es viermal getan. Siehe Kapitel 5.)
- Rückfragen dauerhaft abschalten, solange Sie nicht beurteilen können, was da freigegeben wird.
- Etwas produktiv stellen, an dem echte Daten oder echte Menschen hängen.
- Aus „läuft“ auf „funktioniert“ schließen.

5. Wofür sich ein Agent eignet — und wofür nicht

Die ehrliche Einordnung, bevor Sie Erwartungen aufbauen:

GUT GEEIGNET

WARUM

Etwas bauen, das es so noch nicht gibt

Kein Standardprodukt passt, Eigenbau war bisher zu teuer

Bestehendes verstehen und dokumentieren

Liest schneller als jeder Mensch, ermüdet nicht

GUT GEEIGNET

WARUM

Wiederkehrende Handgriffe automatisieren	Der Aufwand lohnt sich jetzt auch bei kleinen Mengen
Unstrukturiertes strukturieren	Die Stärke schlechthin — Text rein, Ordnung raus

SCHLECHT GEEIGNET

WARUM

Aufgaben, deren Ergebnis Sie nicht beurteilen können	Sie bewerten dann nur noch die Geschwindigkeit
Entscheidungen mit rechtlicher Tragweite	Der Entwurf hilft, die Verantwortung bleibt bei Ihnen
Alles, was exakt reproduzierbar sein muss	Zweimal dieselbe Frage ergibt nicht zwingend dieselbe Antwort
Rechnen im engeren Sinn	Dafür gibt es Tabellenkalkulationen, und die irren sich nie

Begriffe aus diesem Kapitel

Prompt	Die Anweisung, die Sie eingeben. Ihre Qualität entscheidet mehr über das Ergebnis als die Wahl des Modells.
Agent	Ein Sprachmodell mit Werkzeugzugriff. Kann handeln, nicht nur antworten.
Werkzeug	Eine einzelne Fähigkeit, die der Agent aufrufen darf — Datei lesen, Befehl ausführen, Webseite abrufen.
Eingeschleuste Anweisung	Text aus fremder Quelle, den der Agent als Auftrag missversteht. Im Fachjargon Prompt Injection. Die Gegenmaßnahme ist keine Wachsamkeit, sondern eine Berechtigung, die der Agent nicht überschreiten kann.
Kontext	Der begrenzte Textbereich mit allem, was dem Agenten gerade präsent ist.
Kompaktierung	Automatisches Zusammenfassen des Kontexts beim Volllaufen. Spart Platz, kostet Details.
Sitzung	Ein zusammenhängender Dialog. Endet sie, endet auch das Kurzzeitgedächtnis.
CLAUDE.md	Projektdatei mit dauerhaften Regeln, wird zu Sitzungsbeginn automatisch gelesen.

Über den Autor

Christian Zöllner begleitet mittelständische Unternehmen im deutschsprachigen Raum beim Einsatz von KI – von der Befähigung der Entscheidungsebene über die Bereitstellung der Infrastruktur bis zum laufenden Betrieb. Vor diesem Buch lagen zweiundzwanzig Jahre Prozessberatung und viereinhalb Monate, in denen er alles noch einmal von vorn gelernt hat.

ZOELLNER.AI

Wenn Sie weiterkommen wollen

Dieses Buch beschreibt, wie ich arbeite. Wenn Sie selbst anfangen: Kapitel 15 hat eine Reihenfolge für die ersten dreißig Tage, und Anhang B vier Listen für die Momente, in denen man besser nicht aus dem Gedächtnis arbeitet.

Wenn Sie es nicht selbst machen wollen, sondern beurteilen müssen, wem Sie es anvertrauen: Nehmen Sie die Fragen aus den Kapiteln 5, 6 und 10 mit ins nächste Gespräch. Was der Betrieb kostet, wo die Daten verarbeitet werden und wie Test und Ernstfall getrennt sind – darauf sollte jeder Anbieter in zwei Minuten antworten können.

Und wenn Sie einen Prozess vor sich haben, bei dem Sie nicht sicher sind, ob sich der Aufwand lohnt: Darüber spreche ich gern. Dreißig Minuten, unverbindlich, ohne Folien. Sie bringen den Ablauf mit, der Sie ärgert, und bekommen eine ehrliche Einschätzung – einschließlich der Stellen, an denen KI Ihnen nichts bringt.

Vier Monate. 2.786 Anweisungen. Ein Protokoll, das nichts ausgelassen hat.

Was wirklich passiert, wenn ein Berater ohne Entwicklerteam anfängt, mit einem KI-Agenten zu arbeiten — mit allen Umwegen.

Am 14. März 2026 mietete ich einen Server für unter zehn Euro im Monat und wollte eine Stunde lang ausprobieren, was ein KI-Agent kann.

Sechzehn Stunden später lief meine erste Anwendung im Netz. Vier Monate später waren es achtundzwanzig.

Dazwischen liegt alles, was in solchen Büchern sonst fehlt: acht Tage ohne Datensicherung. Vier Zugangsschlüssel, die ich im Klartext eingetippt habe. Sechs Projekte, die ich im April gelöscht habe, obwohl alle einwandfrei starteten. Und die Frage, die ich mir fünf Wochen lang nicht gestellt hatte: wofür eigentlich.

Ich konnte zu Beginn nicht programmieren. Ich hatte zweiundzwanzig Jahre Prozessberatung hinter mir und eine blinkende Eingabezeile vor mir. Was ich nicht wusste: Mein Rechner schrieb mit. 2.786 Anweisungen, jede mit Datum und Uhrzeit auf die Sekunde — und damit die Unmöglichkeit, mir irgendetwas schönzurechnen.

Was am Ende blieb: **191 Stunden, rund 560 Euro, achtundzwanzig laufende Anwendungen — und ein Geschäft, das es vorher nicht gab.**

- **Für den, der selbst anfangen will** — neben jeder Geschichte die Werkstatt mit den Handgriffen
- **Für den, der entscheiden muss** — in den Kapiteln 5, 6 und 10 die drei Fragen, auf die jeder Anbieter in zwei Minuten antworten können sollte
- **Was es kostet** — die vollständige Rechnung, gerundet zu meinen Ungunsten
- **Was rechtlich gilt** — DSGVO im Mittelstand, ohne Beruhigungsformeln
- **Wie Sie sich absichern** — Rechte, Grenzen und Sperren, bevor etwas passiert
- **Was beim Menschen bleibt** — und warum das eine gute Nachricht ist

Christian Zöllner

berät seit über zwei Jahrzehnten mittelständische Unternehmen in Prozessen und Digitalisierung. Seit März 2026 baut er die Software selbst, über die er vorher beraten hat.

ISBN
wird beim Druck
eingesetzt