

FABLE 5.1

Erst aufräumen, dann weiterarbeiten

Warum die Regeln aus den letzten Monaten dein Setup jetzt eher ausbremsen

über 80 %

hat Anthropic am eigenen
System-Prompt gekürzt

bis 45 %

günstiger bei agentischen
Aufgaben

3 Aufgaben

reichen dir für den Test danach

Bei Fable 5.1 fängst du nicht mit neuen Regeln an, sondern du wirfst erst mal die alten raus. Anthropic schreibt das selber in der Prompting-Anleitung zu 5.1, und sie haben ihren eigenen System-Prompt für Claude Code um über 80 % gekürzt und dabei fast nichts verloren. Deine CLAUDE.md, deine Skills und deine Projektanweisung sind halt über Monate gewachsen, und jede Zeile darin war mal die Antwort auf ein Problem, das ein älteres Modell hatte.

01 Die Reihenfolge

Du gehst von unstrittig nach heikel und nicht alles auf einmal. Was zuerst rausfliegt, steht auf Seite 2, und was auf jeden Fall drinbleibt, steht auf Seite 3.

02 Der Prompt

Den Rest lässt du ihn selber aussortieren. Der Prompt dafür steht auf Seite 4 zum Kopieren, und er sortiert dir jede Zeile in eine von drei Gruppen ein und sagt dir bei jeder, warum sie rausfliegt.

03 Der Test

Danach prüfst du, ob wirklich etwas schlechter geworden ist, und zwar mit drei Aufgaben, die du letzte Woche wirklich gestellt hast. Wie du dabei zählst, steht auf Seite 5.

DIE REIHENFOLGE

Was zuerst rausfliegt

Von unstrittig nach heikel, und nicht alles an einem Tag

01 Eine Kopie anlegen

Bevor du irgendetwas löschst, duplizierst du den Ordner oder committest einmal alles. Dann kommst du jederzeit zurück, und nur so traust du dich auch wirklich.

02 Alle Verbote raus

Jede Zeile, die mit niemals, immer oder unbedingt anfängt, fliegt als Erstes raus. Genau diese Sätze widersprechen sich am häufigsten gegenseitig.

03 Formatierungs-Verbote raus

Also sowas wie keine Bullet Points oder kein Fettdruck. 5.1 formatiert von sich aus schon weniger, und mit der alten Regel bekommst du am Ende gar keine Struktur mehr.

04 Sätze, die das Mitreden abstellen

Also berichte erst am Ende. 5.1 schreibt ohnehin weniger Zwischenstand als 5, und dann sitzt du vor einem stummen Fenster und weißt nicht, ob noch was läuft.

05 Alles, was er längst selber kann

Schritt für Schritt denken, Code testen, saubere Commit-Nachrichten schreiben. Das musste man den älteren Modellen aufschreiben, und heute steht es einfach nur im Weg.

06 Zum Schluss der Rest

Wenn nach den Schritten davor sowieso kaum noch was übrig ist, dann lösche auch den Rest. Was du wirklich behalten solltest, steht auf der nächsten Seite.

DIE AUSNAHME

Was drin bleiben darf

Drin bleibt nur das, was wirklich nur bei dir gilt

WAS IN DER DATEI STEHT	WAS DU DAMIT MACHST
Wofür das Projekt da ist	Bleibt drin, aber in zwei Sätzen und nicht länger
Eure Stolperfallen	Bleibt drin, also dass ein Test regelmäßig grundlos fehlschlägt oder dass ein Ordner generiert wird und niemand ihn von Hand anfassen darf
Interne Namen und Absprachen	Bleibt drin, weil er das nirgends nachlesen kann
Verbote mit niemals, immer, unbedingt	Fliegt raus, das sind die Zeilen, die sich am häufigsten gegenseitig widersprechen
Keine Bullet Points, kein Fettdruck	Fliegt raus, sonst bekommst du am Ende gar keine Struktur mehr
Berichte erst am Ende	Fliegt raus, sonst sitzt du vor einem stummen Fenster
Schritt für Schritt denken, Code testen, Commits schreiben	Fliegt raus, das kann er längst von allein

Die Faustregel für die Länge ist eigentlich ganz einfach.

Wenn du deine CLAUDE.md nicht in einem Zug vorlesen kannst, ohne zwischendurch Luft zu holen, dann steht zu viel drin. Und was du einer neuen Kollegin nicht aufschreiben würdest, weil sie es sowieso kann, das braucht in der Datei auch nicht zu stehen.

DER PROMPT

Den Rest sortiert er selber aus

Du gibst ihm die Datei und er teilt dir jede Zeile in drei Gruppen ein

Du startest Claude Code in dem Projekt, in dem deine CLAUDE.md liegt, und gibst ihm das hier rein.

PROMPT ZUM KOPIEREN

Lies meine CLAUDE.md und geh sie Zeile für Zeile durch.
Sortier jede Zeile in eine von drei Gruppen:
1. gilt nur in diesem Projekt und musst du wirklich wissen,
2. kannst du sowieso schon,
3. Verbot aus einer alten Modellgeneration.

Schreib mir danach die gekürzte Fassung, die nur noch aus Gruppe 1 besteht. Und sag mir bei jeder Zeile, die rausfliegt, in einem Satz warum.

01 Erst lesen, dann austauschen

Du lässt dir die gekürzte Fassung erst mal nur anzeigen und tauschst die Datei noch nicht aus. Die Begründung pro Zeile ist eigentlich der wertvollste Teil daran, weil du da schwarz auf weiß siehst, was du dir über Monate angewöhnt hast.

02 Derselbe Prompt geht für den Rest

Das kannst du genauso mit deinen Skills, deiner Projektanweisung und deinen Hooks machen. Du tauschst im Prompt einfach den Dateinamen aus und lässt es nochmal laufen.

03 Ziel aufschreiben statt Verbot

Wenn du eine Regel wirklich brauchst, dann schreib nicht auf, was er nicht machen soll, sondern was rauskommen soll. Anthropic hat bei sich aus dem Verbot „schreib niemals mehrzeilige Kommentare“ den Satz „schreib Code, der sich liest wie der Code drumherum“ gemacht. Der deckt hundert Fälle ab statt einen.

DER TEST

Woran du merkst, ob es schlechter wird

Nicht nach Gefühl, sondern gezählt

01 Drei echte Aufgaben aussuchen

Nimm dir drei Aufgaben, die du letzte Woche wirklich gestellt hast. Ausgedachte Testfälle sagen dir nichts über deinen Alltag, und an echten Aufgaben merkst du sofort, wenn wirklich etwas fehlt.

02 Vom gleichen Startpunkt

Beide Durchläufe müssen vom selben Stand aus starten, also gleicher Branch und gleiche Dateien. Sonst vergleichst du zwei verschiedene Situationen und weißt hinterher nicht, woran es eigentlich lag.

03 Die Nachkorrekturen zählen

Du zählst mit, wie oft du nachkorrigieren musstest, bis es gepasst hat. Dazu schreibst du dir auf, ob er Sachen gemacht hat, die du gar nicht wolltest, und ob du etwas erklären musstest, was vorher in einer Regel stand.

04 Im Zweifel bleibt es draußen

Findest du nach drei Aufgaben nichts, was schlechter geworden ist, dann war die Regel überflüssig. So ist Anthropic bei sich auch vorgegangen, also ein Stück rausnehmen, messen, und wenn nichts einbricht, bleibt es weg.

Anders ist noch nicht schlechter.

Nach dem Löschen wirkt die erste Antwort fast immer ungewohnt, weil sie einfach anders formuliert ist als sonst. Deshalb zählst du Korrekturen, statt zu bewerten, wie es sich anfühlt. Und wenn dir wirklich etwas fehlt, holst du dir die eine Zeile aus deiner Kopie zurück und nicht gleich die ganze alte Datei.

EINORDNUNG

Was hinter den Zahlen steckt

Damit du weißt, was die Zahlen aus dem Video wirklich bedeuten

ZAHL	WAS WIRKLICH DAHINTERSTECKT
bis zu 45 % günstiger	Das gilt für agentische Workloads. Bei normalen Workloads sind es rund 25 %
0,25 \$ pro Mio. Tokens	Cache-Reads sind um 75 % billiger geworden, und das ist der eigentliche Treiber und nicht der Token-Preis
Weniger Tokens pro Aufgabe	Der zweite Teil der Ersparnis, weil das Modell für dieselbe Aufgabe weniger Tokens braucht
38 Stunden am Stück	Ein einziger dokumentierter Lauf und kein Wert, den du bei dir so erwarten kannst
über 80 % gekürzt	Gekürzt wurde der System-Prompt von Claude Code und nicht deine eigene Datei

01 Warum Anthropic den eigenen Prompt gekürzt hat

Sie haben den System-Prompt von Claude Code um über 80 % gekürzt, weil die Fable-5-Modelle mit einem kleineren System-Prompt besser arbeiten. Und genau das ist der Grund, warum auch deine eigene Datei jetzt kürzer werden darf.

Die 38 Stunden sind ein einzelner Lauf, den jemand öffentlich beschrieben hat.

Dwight Temple hat das Modell bei Ramp unbeaufsichtigt an einem ML-Problem arbeiten lassen, und es hat dabei ein früheres Ergebnis als Label-Artefakt erkannt, das korrigiert und über Nacht sechs Experimente parallel gestartet. Das ist halt ein Einzelfall und keine Angabe, die du auf deine eigenen Aufgaben übertragen kannst.

NÄCHSTER SCHRITT

Direkt umsetzen

Du brauchst dafür nichts zu installieren, eine Kopie deiner Datei reicht

01 Kopie anlegen

Duplizier den Ordner oder commit einmal alles, das dauert eine Minute. Danach kannst du beim Löschen nichts kaputt machen und kommst jederzeit zurück.

02 Den Prompt reingeben

Starte Claude Code in deinem Projekt und gib ihm den Prompt von Seite 4. Du liest dir die Begründungen einmal durch und übernimmst danach die gekürzte Fassung.

03 Drei Aufgaben nachstellen

Nimm dir drei Aufgaben aus der letzten Woche, stell sie vom gleichen Startpunkt nochmal und zähl mit, wie oft du nachkorrigieren musst. Bricht nichts ein, war die Regel überflüssig.

Du willst solche Sachen mitbekommen, ohne danach suchen zu müssen?

Ich zeige auf Instagram und TikTok, was mit Claude, ChatGPT und den anderen Tools gerade geht, und die Anleitung dazu gibt es jedes Mal als PDF.

→ Instagram: @timhuck.ai

→ TikTok: @tim_huck_ai

→ www.shortmedia.agency

Angaben zu Fable 5.1 nach Anthropics eigenen Veröffentlichungen zum Modell und zur Prompting-Anleitung für 5.1, geprüft am 9. September 2026. Die 38 Stunden stammen aus einem öffentlich geschilderten Lauf von Dwight Temple bei Ramp und sind ein Einzelfall. Die Kürzung um über 80 % bezieht sich auf den System-Prompt von Claude Code. Das Beispiel mit den mehrzeiligen Kommentaren stammt aus Anthropics eigenem Beitrag zum Aufräumen von Anweisungen.