

Kontext-Faustregel: der Blast-Radius-Spickzettel

SPICKZETTEL · STAND: JULI 2026

Die Faustregel

Gib dem Agenten so viel Kontext, wie die Änderung Schaden anrichten kann. Kontext folgt Blast-Radius — nicht der Grösse des Fensters.

Die Stufen: Änderung → Kontext

Änderung	Was du übergibst
Tippfehler, Ein-Datei-Fix	Die Datei. Fertig.
Feature innerhalb eines Moduls	Das Modul + seine Tests + die Interfaces, die es berührt
Querschnitt (Auth, Logging, Datenmodell)	Die betroffenen Subsysteme + Architektur-Doku + ein, zwei Referenz-Implementierungen der Konvention
Architektur-Entscheid oder Review	So viel echte Codebase wie möglich, plus ADRs — hier zählt sich die Million aus

Nebeneffekt: Die Regel zwingt dich, den Blast-Radius *vor* dem Prompt zu bestimmen — und genau diese Überlegung ist die halbe Spezifikation.

Wann grosser Kontext gewinnt

- **Subsysteme statt Dateien übergeben** — der Agent sieht die Beziehungen, statt sie zu raten; die «kompiliert, passt aber nicht zur Architektur»-Fehler schrumpfen.
- **Architektur-Review über die echte Codebase** — «Wo verletzen wir unsere Schichten?» wird vom Sampling-Spiel zur ehrlichen Frage.
- **Migrationsarbeit mit Gedächtnis** — der Agent kennt Konvention und Sonderfälle aus dem ganzen bisherigen Diff, statt sie pro Datei neu zu erfinden.
- **Dokumente und Code zusammen** — Spezifikation, ADRs, Ticket und Code in einem Kontext; die Übersetzungsverluste schrumpfen messbar.

Aber: Retrieval ist nicht tot. Fürs Nachschlagen bleibt gezielte Suche billiger und oft präziser — grosser Kontext gewinnt, wo *Zusammenhang* die Aufgabe ist.

Die 3 Gegenkräfte

- **Verwässerung** — Relevanz ist keine Funktion der Menge: Wer wahllos das Repo hineinkippt, macht die entscheidenden zwanzig Zeilen zur Nadel im Heuhaufen.
- **Kosten und Latenz** — grosse Kontexte kosten linear Geld und Zeit, bei jedem Aufruf; ein 800k-Kontext für eine Ein-Zeilen-Änderung ist Verschwendung mit Ansage.
- **Falsche Sicherheit** — der Agent mit vollem Kontext macht *plausiblere* Fehler, die schwerer auffallen; Review wird wichtiger, nicht unwichtiger.

Der komplette Deepdive: <https://halvic.ch/de/blog/1-million-token-kontext>