

Skill-Audit: Was abwertet, was aufwertet

SPICKZETTEL · STAND: JULI 2026

Dein Wert wandert von *Code produzieren* zu *Systeme verantworten*. Wenn Code beliebig billig wird, zählt nicht, wie du das Tool bedienst — sondern welche deiner Fähigkeiten an Wert gewinnen und welche verlieren.

Die Bilanz

Wertet ab	Wertet auf
Boilerplate und CRUD aus dem Kopf tippen — der Agent ist schneller, jeden Tag	Spezifizieren statt prompten — ein präziser, durchdachter Auftrag ist das neue Kern-Artefakt
Syntax und API-Signaturen auswendig können — das war nie Kompetenz, nur Übung	Urteilen unter Volumen — generiertem Code misstrauen und ihn überführen
Der schnellste Tipper im Team sein — Schreibgeschwindigkeit ist kein Engpass mehr	Systeme denken, nicht Dateien — Architektur, Datenflüsse, Schicht-Grenzen
Framework-Trivia — «Wie hiess nochmal der Parameter?» ist eine gelöste Frage	Übersetzen zwischen Business und Technik — was soll es leisten, für wen, woran misst man Erfolg?

Nichts davon ist neu: Es ist das klassische Software-Handwerk — befreit von der Tipparbeit, die es jahrelang verdeckt hat.

Selbst-Audit

- ☐ Kann ich ein Problem in eine präzise Spec zerlegen — mit Inputs, Edge Cases und erwartetem Fehlverhalten — bevor ich prompte?
- ☐ Misstraue ich generiertem Code auch in Menge — und kann ich ihn überführen, statt jeden Entwurf durchzuwinken?
- ☐ Denke ich in Systemen statt in Dateien — sehe ich Architektur, Datenflüsse und Schicht-Grenzen, wo der Agent nur den lokalen Ausschnitt sieht?
- ☐ Kann ich übersetzen, was das Ding fürs Business leisten soll, für wen — und woran man Erfolg erkennt?

Das Experiment diese Woche

Kein Kurs, keine Zertifizierung — ein Experiment:

- ☐ Nimm deine nächste Aufgabe und schreib die Spezifikation, **bevor** du den Agenten anwirfst.

- ☐ Halte alles vor dem ersten Prompt fest: Inputs, Edge Cases, erwartetes Fehlverhalten — und die Tests, die es beweisen müssen.
- ☐ Miss danach: Wie viele Korrektur-Prompts hast du gespart?
- ☐ Miss danach: Wie viel besser war der erste Entwurf?

Du wirst merken: Die Arbeit, die zählt, passiert *vor* und *nach* der Generierung — genau da wächst der KI-System-Experte.

Der ganze Artikel: <https://halvic.ch/de/blog/vom-entwickler-zum-ki-experten> · Der Deep-Dive: das Hörbuch *Vom Entwickler zum KI-System-Experten*.
