



Dipl.-Ing. Karl-Heinz Haas, MA

Gerichtssachverständiger für Informationstechnik

Unternehmensberater für digitale Transformation und KI

Über 35 Jahre Erfahrung in Analyse, Planung und Begutachtung komplexer IT-Systeme

# KI-ENTWICKLUNG MIT PYTHON & CLAUDE CODE FÜR SACHVERSTÄNDIGE

BESCHREIBEN. ERZEUGEN. PRÜFEN.

MODUL 6 VON 7 – LEHRGANG «KI FÜR SACHVERSTÄNDIGE ALLER DISZIPLINEN»

## Typische Probleme, die wir lösen

### Echtes Programmieren schien außer Reichweite

Low-/No-Code stoßen an Grenzen, doch ein Programmier-Werkzeug verlangte bisher einen Entwickler. Claude Code erzeugt den Python-Code Zeile für Zeile – Sie beschreiben und prüfen, statt zu tippen.

### KI-generierter Code ist ohne Prüfung eine Black Box

Eine App per Knopfdruck, die niemand liest, ist nicht erklärbar. Claude Code arbeitet diff-basiert: Sie geben jede Änderung frei – so erfüllen Sie die menschliche Aufsicht, die Art. 14 KI-VO verlangt.

### Das mächtigste Werkzeug läuft in fremder Cloud

Claude Code ist proprietär und cloudgebunden, doch sensible Akten gehören nicht dorthin. Hybrid löst das: Sensibles bleibt lokal über Ollama, der fertige Code läuft auf Ihrem Rechner.

## Professionelle KI-Kompetenz

Mit Low-/No-Code-Tools klicken Sie Werkzeuge in fertigen Baukästen zusammen. Hier entsteht echter Python-Code – frei geformt um genau Ihren Fall, ohne die Grenzen eines Frameworks. Ob Sie selbst schreiben, ein Mitarbeiter es übernimmt oder Sie es extern vergeben: Sie beschreiben das Ziel, Claude Code erzeugt, und Sie lesen jede Änderung als Diff, bevor sie zählt. Das Schwere übernimmt Claude, Sensibles bleibt lokal über Ollama – und das fertige Werkzeug läuft auf Ihrem Rechner.

Nicht der Code prüft sich.

Sie prüfen ihn – und verantworten, was Ihr Werkzeug erzeugt.

## Kompetenzfelder

- ✓ Gerichtsgutachten
- ✓ Seminare
- ✓ Privatgutachten
- ✓ Fachvorträge
- ✓ IT- und KI-Planung
- ✓ Autorentätigkeit



### Eigenbau und Fertiglösung abgrenzen

Standardsoftware und No-Code-Werkzeuge lösen das Allgemeine, nicht jeden Fall. Sie lernen zu entscheiden, wann sich ein selbstgebautes Python-Werkzeug lohnt – und wann das fertige Tool reicht.



### Die Werkbank aufsetzen

VS Code, Miniconda mit Python-Umgebungen, dazu Claude Code als agentischer Motor. Hybrid wie in der ganzen Reihe: das Schwere über Claude, Sensibles lokal über Ollama.



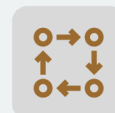
### Vom Prompt zum Programm erzeugen

Sie beschreiben, was das Werkzeug können soll; Claude Code erzeugt den Python-Code, Schritt für Schritt und nachvollziehbar. Kein Blindflug: jede Änderung als Diff, das Sie freigeben.



### Lokal ausliefern und betreiben

Das fertige Werkzeug läuft auf Ihrem Rechner, als Web-App im Docker-Container oder als Desktop-App. Keine fremde Cloud, kein Datenabfluss – Sie betreiben es selbst.



### Recht & Haftung verantworten

Ein selbstgebautes Werkzeug bleibt Werkzeug, kein Gutachter. DSGVO und KI-VO, geprüfte Logik, dokumentierte Methode – und die Git-Historie als Nachweis, was wann entstanden ist.



### Live am Fall prüfen

Ein Tool, vor Ihren Augen von Claude Code gebaut: Sie lesen die Diffs, geben frei, am Ende ist es einsatzbereit. Nicht der Code entscheidet – Sie prüfen ihn und verantworten das Ergebnis.

Rufen Sie  
mich an



+ 43 664 34 10 309