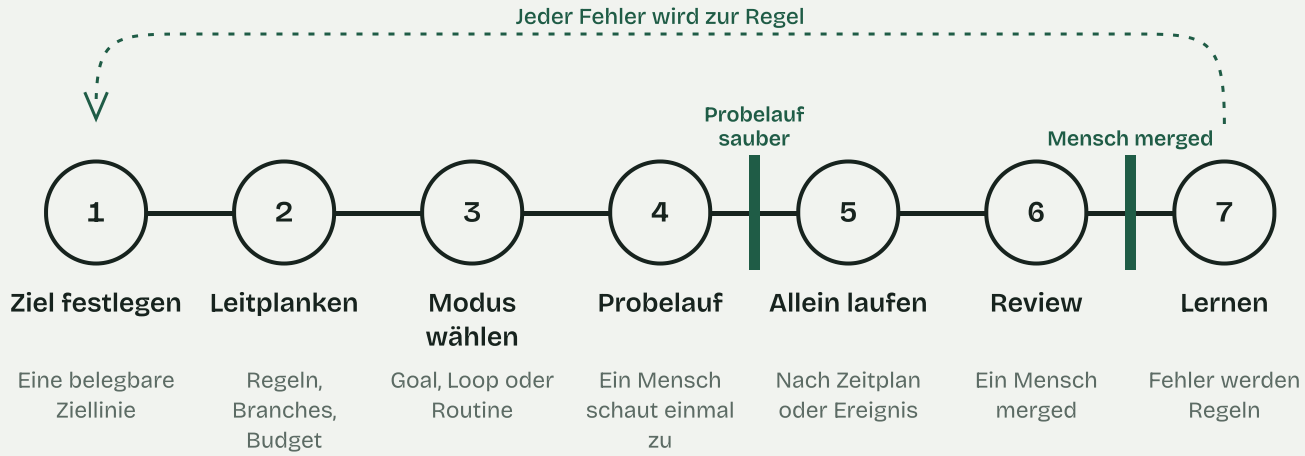


Dominik Heinrich, Multi-Cloud DevOps & AI Solution Architect

Agenten arbeiten lassen, während Sie schlafen: Goals, Loops und Routines

Wie ich Teams vom Prompt-Ping-Pong zu Agenten bringe, die selbstständig laufen, mit klarer Ziellinie, Zeitplan und menschlichem Review. Funktioniert mit Claude Code und Codex.



Sieben Schritte, zwei Gates. Agenten laufen selbstständig; Menschen entscheiden, was live geht.

Aus produktiven Plattformen, die ich so betreibe, anonymisiert

5'000+	Stündlich	Nächtlich	Täglich
gemergte Pull Requests auf zwei Plattformen	Checks der Buchungsstrecken, unbeaufsichtigt	Restore-Test der Backups	Kostenbericht im Team-Chat

Die sieben Schritte

Vom Prompt-Ping-Pong zu Agenten, die nach Zeitplan arbeiten. Abhaken, sobald sie live sind.

01 Vorher definieren, was fertig heisst

- [ ] Die Ziellinie so formulieren, dass der Agent sie belegen kann: „npm test endet mit 0“, nicht „mach es besser“.
- [ ] **Den Rahmen nennen:** welche Ordner, welche Tickets, was tabu ist.
- [ ] Eine Grenze ins Ziel schreiben, etwa „oder nach 20 Runden stoppen und berichten, was offen ist“.

02 Leitplanken setzen

- [ ] Eine Anweisungsdatei pro Repository mit den Regeln, die der Agent einhalten muss.
- [ ] **Nur Branches:** Der Agent öffnet Pull Requests, er merged nie und deployt nie in die Produktion.
- [ ] Keine Produktions-Zugangsdaten, nur Testkonten, und ein Token-Budget pro Lauf.

03 Den passenden Modus wählen

- [ ] **Klare Ziellinie:** ein Goal (/goal in Claude Code, Goal Mode in Codex).
- [ ] Wiederkehrende Prüfung, während Sie arbeiten: ein Loop im Intervall, etwa /loop 15m.
- [ ] Muss bei zugeklapptem Laptop laufen: eine Routine in der Cloud, gestartet per Zeitplan, API-Aufruf oder GitHub-Ereignis.

04 Probelauf unter Aufsicht

- [ ] Einmal interaktiv laufen lassen und jeden Schritt lesen.
- [ ] Die Anweisungen korrigieren, wo der Agent danebenlag, nicht nur das Ergebnis.
- [ ] Unbeaufsichtigt läuft es erst, wenn ein Probelauf sauber durchgeht.

05 Unbeaufsichtigt laufen lassen

- [ ] **Klein anfangen:** ein Repository, eine Aufgabenart, etwa Dependency-Updates, fehlschlagende Tests oder ein nächtlicher Bericht.
- [ ] Jeder Lauf schreibt eine kurze Zusammenfassung: was er getan, was er ausgelassen hat, was einen Menschen braucht.
- [ ] Ein Alert geht in den Team-Chat, wenn ein Lauf fehlschlägt oder deutlich länger dauert als sonst.

06 Review, bevor etwas live geht

- [ ] Ein Mensch prüft und merged, genau wie bei Pull Requests von Menschen.
- [ ] **Feste Regeln:** Login, Zahlungen, Datenlöschung und Berechtigungen brauchen immer einen Menschen.
- [ ] Erst Staging, dann Produktion, mit Smoke-Tests nach jedem Deployment.

07 Lernen und ausweiten

- [ ] Jeder Fehler wird zur Regel in der Anweisungsdatei, damit der nächste Lauf ihn vermeidet.
- [ ] Gesparte Stunden, Kosten pro Lauf und den Anteil der Läufe messen, die Hilfe brauchten.
- [ ] Die nächste Aufgabenart erst dazunehmen, wenn die aktuelle zwei Wochen ruhig gelaufen ist.

Einführung nach Unternehmensgrösse

Startup: eine Routine für die lästige Arbeit, etwa Dependency-Updates, fehlschlagende Tests und den Wochenbericht. Meist innerhalb einer Woche live.

Mittelstand: ein Pilotteam mit gemeinsamen Anweisungsdateien und Token-Budget, danach die anderen Teams.

Konzern: Managed Settings, die Nutzer nicht überschreiben können, eine freigegebene Liste von Modellen und Tools, Audit-Logs und lokale Modelle für sensiblen Code.

Die häufigsten Fehler

- Ein Ziel wie „verbessere den Code“, das nie endet und Tokens verbrennt.
- Den Agenten seine Arbeit selbst für fertig erklären lassen, ohne Tests als Beleg.
- Unbeaufsichtigt laufen lassen, bevor es einen Probelauf unter Aufsicht gab.
- Agenten mit Merge-Rechten oder Produktions-Zugangsdaten.
- Das Ergebnis von Hand korrigieren statt der Anweisungen, und morgen kommt derselbe Fehler wieder.

So arbeiten wir zusammen

Dieselbe Qualität für ein Startup mit zehn Leuten und einen globalen Konzern. Das Format passt sich Ihnen an.

Startups **Fractional DevOps- & KI-Lead**	Ein paar Tage im Monat: Architektur, Release-Pipeline und KI-Funktionen, damit sich die Gründer auf Kunden konzentrieren können. Vom Prototyp in die Produktion, ohne Neubau.
KMU und Mittelstand **Ein Projekt mit klarem Ergebnis**	Ein klares Ziel wie tiefere Cloud-Kosten, eine Migration oder die ersten KI-Agenten in Produktion, mit festem Umfang, wöchentlichem Fortschritt und einer Übergabe, die Ihr Team betreiben kann.
Konzerne **Ein Architekt in Ihren Teams**	Eingebettet in Plattform-, Security- und Produktteams: Architekturentscheidungen, Governance für KI und regulierte Umgebungen. Für grössere Vorhaben bringe ich ein erfahrenes Senior-Team mit.

Wie ich arbeite

Entscheidungen schriftlich: Architecture Decision Records mit den Abwägungen, damit Ihr Team weiss, warum, nicht nur was.

Sicher ab Werk: OIDC und Managed Identities, Least Privilege, Secrets im Vault, ein Audit-Trail.

Betrieben, nicht nur geliefert: SLOs, Alerts auf Geschäftsergebnisse und eine Incident-Routine.

Gemessen: Kosten, Durchlaufzeit, Verfügbarkeit und gesparte Stunden, jede Woche berichtet.

Das Wissen bleibt bei Ihnen: Runbooks, Dokumentation und Workshops für Ihr Team.

Ein Ansprechpartner, der mit Engineering und Management spricht, auf Deutsch, Englisch und Spanisch.



Sollen die Agenten Ihres Teams sicher selbstständig laufen?

30-Minuten-Call buchen

E-mail mail@dominikheinrich.ch
Web dominikheinrich.ch

Weltweit verfügbar, remote. Vor Ort auf Anfrage, je nach Land.