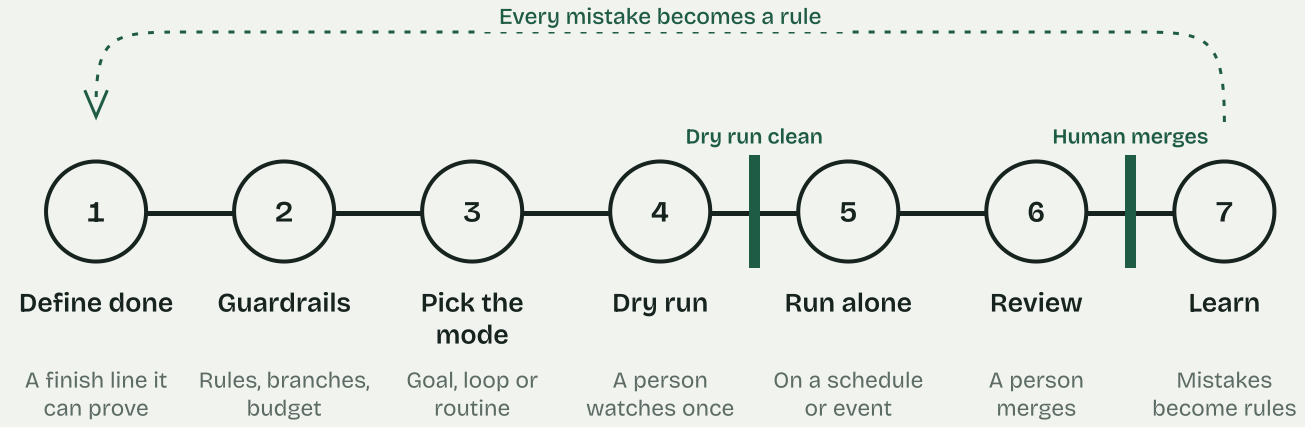


Dominik Heinrich, multi-cloud DevOps & AI solution architect

Let agents work while you sleep: goals, loops and routines

How I move teams from typing prompts one by one to agents that run on their own, with a clear finish line, a schedule and a human review. Works with Claude Code and Codex.



Seven steps, two gates. Agents run on their own; people decide what ships.

From production platforms I run this way, anonymized

5,000+ merged pull requests on two platforms	**Hourly** booking-funnel checks, unattended	**Nightly** restore test of the backups	**Daily** cost report in the team chat

The seven steps

From ping-pong prompting to agents that work on a schedule. Tick them off as they go live.

01 Define done before you start

- [ ] Write the finish line as something the agent can prove: 'npm test exits with 0', not 'make it better'.
- [ ] **Name the scope:** which folders, which tickets, what is off limits.
- [ ] Put a limit into the goal itself, for example 'or stop after 20 turns and report what is left'.

02 Set the guardrails

- [ ] An instruction file per repository with the rules the agent must follow.
- [ ] **Branch-only:** the agent opens pull requests, it never merges or deploys to production.
- [ ] No production credentials, test accounts only, and a token budget per run.

03 Pick the right mode

- [ ] **Clear finish line:** a goal (/goal in Claude Code, goal mode in Codex).
- [ ] **Recurring check while you work:** a loop on an interval, for example /loop 15m.
- [ ] Has to run with your laptop closed: a routine in the cloud, started by a schedule, an API call or a GitHub event.

04 Dry run with a person watching

- [ ] Run it once interactively and read every step.
- [ ] Fix the instructions where the agent went wrong, not just the output.
- [ ] It only runs unattended once a dry run passes cleanly.

05 Run unattended

- [ ] **Start small:** one repository, one task type, such as dependency updates, failing tests or a nightly report.
- [ ] **Every run writes a short summary:** what it did, what it skipped, what needs a person.
- [ ] An alert goes to the team chat when a run fails or takes far longer than usual.

06 Review before anything ships

- [ ] A person reviews and merges, exactly as for human pull requests.
- [ ] **Hard rules:** login, payments, data deletion and permissions always need a human.
- [ ] Staging first, then production, with smoke tests after every deploy.

07 Learn and widen

- [ ] Each mistake becomes a rule in the instruction file, so the next run avoids it.
- [ ] Track the hours saved, the cost per run and the share of runs that needed help.
- [ ] Add the next task type only when the current one has run quietly for two weeks.

Rolling it out, by company size

Startup: one routine for the boring work, such as dependency updates, failing tests and the weekly report. Usually live within a week.

Mid-sized company: a pilot team with shared instruction files and a token budget, then the other teams.

Enterprise: managed settings that users can't override, an approved list of models and tools, audit logs and local models for sensitive code.

Mistakes I see most often

- A goal like 'improve the code' that never ends and burns tokens.
- Letting the agent declare its own work done, without tests that prove it.
- Running unattended before a supervised dry run.
- Agents with merge rights or production credentials.
- Fixing the output by hand instead of the instructions, so the same mistake comes back tomorrow.

Ways to work together

The same quality for a ten-person startup and a global group. The format adapts to you.

Startups

Fractional DevOps & AI lead

A few days a month: architecture, the release pipeline and AI features, so the founders can focus on customers. From prototype to production without a rewrite.

Mid-sized companies

A project with a clear outcome

A defined goal like a lower cloud bill, a migration or the first AI agents in production, with a fixed scope, weekly progress and a handover your team can run.

Enterprises

An architect inside your teams

Embedded with your platform, security and product teams: architecture decisions, governance for AI and regulated environments. For larger scopes I bring a trusted senior team.

How I work

Decisions written down: architecture decision records with the trade-offs, so your team knows why, not just what.

Secure by default: OIDC and managed identities, least privilege, secrets in a vault, an audit trail.

Operated, not just delivered: SLOs, alerts on business outcomes and an incident routine.

Measured: cost, lead time, uptime and hours saved, reported every week.

Knowledge stays with you: runbooks, docs and workshops for your team.

One point of contact who talks to engineering and to management, in German, English and Spanish.



Want your team's agents running on their own, safely?

Book a 30-minute call

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

Available worldwide, remote. On-site on request, depending on the country.