A BEGINNER'S FIELD GUIDE

14 STEPS · 4 PHASES

◆ START HERE

The AI Developer Workflow

A calm, step-by-step path from your very first prompt to a workflow the AI can actually run for you.

Decide → Set Up → Build → Ship & Automate

Johannes Rabauer · July 2026

BEFORE YOU BEGIN · THE GATE

First: are you allowed to use AI?

Four honest checks that decide whether AI belongs in this project at all.

01 · COPYRIGHT

It can copy code verbatim

Models can reproduce licensed code word-for-word. Know what you're legally allowed to ship.

02 · ENVIRONMENT & ETHICS

It burns real power

Heavy use has a genuine energy footprint. Weigh whether the task justifies it.

03 · OTHERS' WORK

Don't ship stolen code

Training data includes other people's work. Don't pass off what isn't yours to use.

04 · DATA SECURITY

Mind what you share

Your prompts and code may leave your machine. Know what you're sending, and to whom.

Johannes Rabauer · July 2026

THE JOURNEY

Four phases, fourteen steps

PHASE 01

Decide

The four choices that shape everything else.

Steps 1–4

PHASE 02

Set Up

A workspace the AI understands from day one.

Steps 5–7

PHASE 03

Build

The daily loop: plan, generate, review, repeat.

Steps 8–13

PHASE 04

Ship & Automate

Turn a project into a system that improves itself.

Step 14

PHASE 01 · STEPS 1–4

Decide

Before you write a line, make the four choices that shape everything else.

Johannes Rabauer · July 2026

01 Full AI or AI-enhanced development?

Decide up front: is the AI driving, or riding shotgun? Both are valid — pick based on how much is at stake.

02 Find the cheapest AI provider

Prices vary wildly for similar quality. A little shopping now saves a lot over the months ahead.

COMPARE → pricepertoken.com/coding-assistants — *handy, but sometimes outdated.*

DECIDE · STEP 3

03 Pick an AI harness

This is the app you'll live in all day. Try a few before you commit to one.

CHOICE A

CLI or desktop?

Terminal power vs. a friendlier window.

CHOICE B

Cloud or local?

Local runs on your own machine and tools; cloud frees you from your device — start a job and let it run overnight.

DECIDE · STEP 4

04 Choose the right model

There's no single "best" model — match it to the task in front of you.

FIRST DECISION

Local or cloud LLM?

On-device for privacy and control, or a hosted model for the most power.

HABIT

Switch models freely

A quick model for small edits, a strong one for hard problems.

TRADE-OFF

Quality · Speed · Cost

You rarely get all three. Pick the two that matter today.

COST STRATEGY → Default to a cheap, fast model — escalate to a bigger, pricier one only for the hard parts, or when you get stuck.

PHASE 02 · STEPS 5–7

Set Up

A comfortable editor, plus a project the AI actually understands.

Johannes Rabauer · July 2026

SET UP · STEP 5

05 Tell the AI about your project — once

One file in the repo — `Agents.md` or similar — carries your context and rules into every prompt. No more re-explaining.

WHAT IT IS

Project facts & structure

HOW YOU WORK

Conventions & skills

GUARDRAILS

Rules it must follow

CONTEXT ENGINEERING → Give it only what it needs — curate, don't dump. Keep the context lean; more isn't better, but do think about what belongs.

SKILL LIBRARIES → `mattpocock/skills` `obra/superpowers` `antonarhipov/agentskills`

06 Add MCP servers

Give the AI safe, direct access to your tools — files, databases, APIs — no more copy-paste.

TRY `IntelliJ` `javadocs.dev` `Maven (Java)`

07 Choose a language and framework

Pick what you know, or what fits the job. **It matters much less than it used to.**

The right tools matter more than the right model.

PHASE 03 · STEPS 8–13

Build

The daily loop: plan, generate, review, repeat — six habits that keep you in control.

Johannes Rabauer · July 2026

08 Plan before you prompt

Five minutes of thinking beats fifty lines of wrong code. Say what "done" looks like first.

DECIDE → Keep durable plans **in the repo**; treat quick ones as **throwaway**.

09 Build in small iterations

Small asks, small diffs. Easy to check, easy to undo when something goes sideways.

10 Review with judgment

You don't have to read every line — but you do decide what needs a close look. Own the result.

11 Write tests & automated checks

Give the AI deterministic ways to verify its own code — not just unit tests, but linting, style, and security scans. Machines catch breakage the instant it happens.

- Unit tests
- Linting
- Checkstyle
- OWASP / security

BUILD · DEEPER DIVE

Give the AI signals it can trust

The faster and more objective the feedback, the better the AI self-corrects. Deterministic checks turn "looks right" into pass or fail.

DETERMINISTIC FEEDBACK LOOPS

The AI checks its own work

Machine-readable pass/fail beats opinion. Give it a command to run and exact output to read — then it corrects and repeats on its own.

EVALUATION

Measure quality over time

Acceptance criteria
Define what "done" actually means, up front.

Regression tests
What you fixed once must never break again.

Benchmarks
Track quality, speed and cost across changes.

12 Use Git aggressively

Commit often. Every good state is a safe place you can return to.

13 Know when to code yourself

Ideally, rarely. Hand-editing can quietly break the AI's understanding — reach for it as a last resort.

PHASE 04 · STEP 14

Ship & Automate

Turn a working project into a system that keeps improving itself.

Johannes Rabauer · July 2026

SHIP & AUTOMATE · STEP 14

14 Automate everything — then improve

Hand the repetitive work to the machine and let every release make the next one better.

FIRST

Set up CI/CD early

THEN

Let the AI run builds, tests & deploys

RESULT

A workflow that runs like a software factory

Ship → Measure → Improve ↺

∞ SHIP & AUTOMATE · DEEPER DIVE

Let agents run long

Loop engineering, aka goal-oriented engineering: set the destination and the guardrails, then let the agent work its way there — unattended.

GOAL-ORIENTED

Aim at an outcome, not a step

Define the goal and how success is measured; the agent loops toward it on its own.

LONG-RUNNING

Kick it off before you log off

Big tasks can run for hours. Start one at night, review the results in the morning.

MIND THE COST → Autonomous loops burn tokens fast. Set a budget and stop conditions before you walk away.

Johannes Rabauer · July 2026

THE WHOLE WORKFLOW

Everything, one screen

Decide → Set Up → Build → Ship & Automate → repeat

BEFORE YOU BEGIN Copyright · Environment & ethics · Others' work · Data security

Decide 1–4

01 Full AI or AI-enhanced?
Driving, or riding shotgun.

02 Cheapest provider
Prices vary wildly.

03 Pick a harness
CLI or desktop · cloud or local.

04 Choose the model
Switch freely · quality/speed/cost.

Set Up 5–7

05 Tell the AI once
Context + rules in Agents.md.

06 Add MCP servers
Direct access to your tools.

07 Language & framework
Matters less than it used to.

One-time setup — pays off on every prompt.

Build 8–13

08 Plan before you prompt

09 Build in small iterations

10 Review with judgment

11 Tests & automated checks
Lint · style · security scans.

12 Use Git aggressively

13 Know when to code yourself
Last resort.

Ship & Automate 14

14 Automate everything — then improve

- Set up CI/CD early
- AI runs builds, tests & deploys
- Workflow becomes a software factory

Ship → Measure → Improve ↺

Johannes Rabauer · July 2026

THE REAL SHIFT

It isn't about writing code faster.

It's about building a delivery pipeline the AI can actively operate — and improving it, release after release. That's the move from **using AI** to **running an engineering system.**

Decide → Set Up → Build → Ship & Automate → repeat

Johannes Rabauer · July 2026