

How not to be an AI slop programmer.

We asked our engineers for one habit each. Forty answers came back, and **almost none were about prompts**. They were about what happens before and after the code. These are the 12 habits they share.

BEFORE CODE

01–04

01 Know what done looks like

Understand the task, the domain and the expected output before you start. If you can't describe the result, you can't judge it. Research first.

02 Plan first, and do it yourself

Draft your own approach, then have the agent critique it, find edge cases and suggest alternatives. Use your strongest model here.

03 Make it ask, not assume

Have the agent list open questions, unclear requirements and its assumptions. Resolve them yourself. Important decisions stay with you.

04 Ground it in what exists

It reads the real ticket, code, data and docs first. It searches for existing implementations, follows local patterns and cites sources.

WHILE IT BUILDS

05–07

05 Work in small slices

One feature or step per run, reviewed before the next. Size each iteration to what you can verify. Shorter loops when you're unsure.

06 Push for the minimal diff

Assume the first draft is overbuilt. Challenge the amount of code, ban unrequested changes and rewrites, stop scope creep as you see it.

07 Write tests that can fail

TDD where it fits, real test databases over mocks. Revert the fix and confirm the test fails. Never let the agent edit tests to pass.

BEFORE YOU SHIP

08–11

08 Ask for evidence, not reports

The end-of-task summary is the least reliable output. Read the diff, logs and touched data. Run the feature live, then try it yourself.

09 Get an independent review

Review in a fresh session against the original task, with a second model and several review skills. Each catches different things.

10 Defend it without AI

If you can't explain and justify the PR on your own, it isn't ready. Ask small "how does X work?" questions until you understand it.

11 Review your own PR first

Before anyone else sees it. Obvious, and the step people skip most.

OVER TIME

12

12 Turn corrections into rules

When the agent repeats a mistake, write the fix and the reason into the repo's instruction file or a review skill, so the next session doesn't repeat it.

Slop is code accepted without judgment. Writing can be delegated. Judgment can't.

THE THREAD'S WORKING DEFINITION

MAIN TAKEAWAYS · JUDGMENT IS THE JOB NOW

01

The work moved to before and after the code.

Specify and plan up front. Verify and review at the end.

02

Trust what you can check.

Failing tests, live runs, diffs and independent reviewers. Never the agent's own claim of success.

03

Verification capacity sets the pace.

Accept only as much code as you can actually check.

04

Magic phrases are the weakest advice.

"Make no mistakes" doesn't replace a process.