

THE FREE ONE FROM

THE *FEATURE* LOOP

How to build with an AI coding agent without it spiraling.
A four-line brief, thinnest-slice-first, and six
guardrails that keep you one small step from a working
commit. Steal the whole thing.

What it prevents	the 10-cycle whack-a-mole spiral
Where it came from	a real build that sprawled 8 directions and died
Cost to you	\$0 / free
Time to first use	the next thing you build

MOST AI BUILDS DON'T FAIL BECAUSE THE PROBLEM IS HARD. THEY FAIL BECAUSE THERE WAS NO PROCESS.

A build with no rails sprawls. You start on one feature and end up with eight half-finished directions, no definition of "done," and no stop rule. When one piece breaks you keep hammering the same broken thing, ten cycles deep, instead of stepping back.

That whole spiral is a missing process, not a hard problem. An AI agent will happily follow you into it at full speed because it has no instinct to stop. The Feature Loop is the instinct, written down. Give it to your agent and to yourself.

It is six steps and six guardrails. Every feature, every time. The point is not ceremony. The point is that you are never more than one small, proven step away from something that works.

THE WHOLE IDEA

Build the thinnest slice that proves the core idea. Prove it actually runs. Commit. Then take the next thinnest slice. Everything below is how to hold yourself and your agent to that.

1 BRIEF — BEFORE ANY CODE

Fill four lines and nothing else until they are filled: **Outcome** ("when this works, I can ____"), **Done test** (the one thing you will observe to know it works), **Out of scope** (2 to 3 things you are deliberately NOT doing), **Thinnest slice** (the smallest piece that proves the core idea). Template is on the last page.

2 BUILD

Build only the thinnest slice. One change at a time. Resist every "while I'm here" extra. Your agent will offer them. Say no.

3 PROVE

Actually run it and watch it work. Observe the real result. Never accept "that should work" from yourself or the agent. A claim is not a result.

4 CHECKPOINT

If it works, commit the slice with a clear message. This commit is your safe rollback point. Write one line about what shipped while you are here, that line is your build-in-public log for free.

5 NEXT OR DONE

Hits the Done test from step 1? Ship it, close the build. Not yet? Back to step 2 with the next thinnest slice. If a script will be run more than once, wrap it so it can be invoked by name, an unwrapped script is half-shipped.

6

RETRO — ONCE, WHEN IT CLOSES

Write one line: what did the process teach you this time? Without this step the loop builds features but never improves itself. This is the step that closes the feedback loop.

Six non-negotiables. Each one is a spiral that already happened to someone, turned into a rule so it does not happen to you.

1 · TWO-STRIKE RULE

A step fails twice, STOP digging. Step back, reassess, or roll back to the last green commit. Never keep hammering the same broken thing. This one rule alone kills the whack-a-mole spiral.

2 · SCOPE FREEZE

New ideas mid-build go on a "later" list, never into the current build. Finish the brief, then consider the new idea as its own brief.

3 · CONTROL-TEST FIRST

Anything you depend on but do not control (an OS permission, an external API, a third-party app) gets a five-minute isolated test before you build on top of it. Prove the dependency works alone first.

4 · ALWAYS GREEN

Never be more than one small step from a working commit. If the tree is not in a known-good state, that is the only thing to fix.

5 · PACKAGE OR IT DIDN'T SHIP

A reusable automation is not done until it can be invoked by name. If your system cannot call it, it is not a capability, it is a file you will forget.

6 · A PROCESS SHIPS WITH ITS CHECK

Any recurring job (a cron, a poller, a scheduled task) is not done until it has a check that reads the outcome in the world, never its

own log. A process cannot grade its own homework.

RECEIPTS, NOT HYPE.

04

THE 4 LINES

Paste into your project as BRIEF.md, or into the chat with your agent. No build starts without these.

OUTCOME

When this works, I can ...

DONE TEST

I'll know it works when I see ...

OUT OF SCOPE (NOT THIS BUILD)

THINNEST SLICE

The smallest piece that proves the core idea is ...

DEPENDENCIES I DON'T CONTROL – CONTROL-TEST THESE FIRST

THIS IS PRIMITIVE 1 OF 4

The Feature Loop keeps a build from spiraling. The full **RELIABLE** kit adds the other three primitives that stop AI agents from lying, from grading their own homework, and from losing hard-won judgment: the **Commitment Ledger and Proof-Gate**, the **Process-Check Registry**, and

the **Judgment Canon**. Each one is a doctrine, a copy-paste template, and the real working script.

Built by one person with a full-time job and an AI operating system that does the work. If this doc saved you one spiral, the kit is the rest of them.