

AI WORKSHOP · 2026

Coding with AI Agents

One workflow for Claude Code and Copilot

WHAT CHANGED

New models: less prompting, not more

medium

Start at medium effort.
Go higher only with a
measured gain

hours

Long autonomous runs, given
a clear finish line

500K / 1M

Max context, depending on
the model. Thinking built in

Delete from prompts:

~~think carefully~~

~~double-check your answer~~

~~use a subagent to verify~~

TRY THIS FIRST

Three habits for every session

01

Whole task. Clear finish line.

"Done means: every endpoint migrated, old client deleted, tests pass."

02

Turn the effort dial. Skip "think hard".

Modern models always think. Effort decides how much.

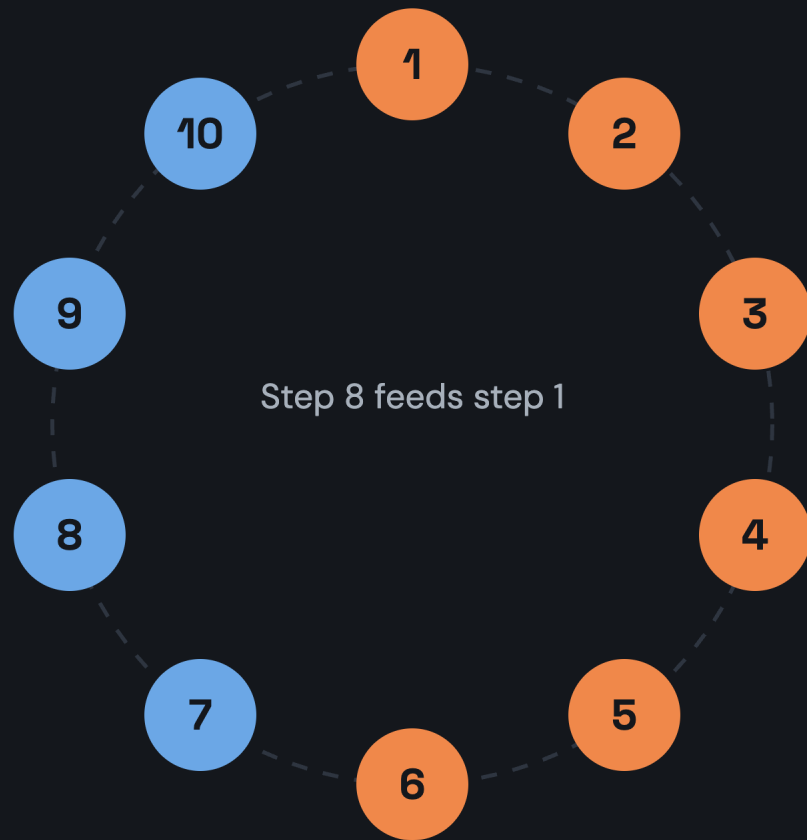
03

Read "needs your decision" first.

Short IDs on every question, so you answer "Q2 yes, OP1 no".

THE WORKFLOW

Ten steps, one loop



- | | | |
|----|-----------------------------|----------------------------------|
| 01 | Context files | short, with a stop rule |
| 02 | Skills, MCP, plugins | on demand, few tools |
| 03 | Jira and GitHub | ticket to PR, least privilege |
| 04 | Spec and review | SPEC.md with a finish line |
| 05 | Agents and advisor | parallel, observer watches |
| 06 | Build | effort dial, tests first |
| 07 | Verify | green ≠ verified, layered review |
| 08 | Learn | repeated fix becomes a rule |
| 09 | Marketplace | owner and version per plugin |
| 10 | Copilot parity | AGENTS.md as the shared base |

Context files: short and sharp

Keep

- ✓ Commands Claude can't guess
- ✓ Rules that differ from defaults
- ✓ Architecture rules and "never do"
- ✓ Test runner and repo etiquette

Cut

- ⚠ What the code already shows
- ⚠ Long docs: link them with @import
- ⚠ "Write clean code"
- ⚠ "Think carefully"

Test each line: **would removing it cause a mistake?** No: cut it.

STEP 01 · THE ONE RULE TO ADD

Tell it when to stop

```
# CLAUDE.md · stop rule
```

When a step doesn't need my input, keep going.

Status notes go with your next action.

Stop and ask only when you can't go on without me,
or before anything destructive: deleting data,
force-pushing, touching files outside this repo.

Report: Needs your decision (Q1, Q2...) · FYI

Progress as % of plan. Commands alone in code blocks.

KEEP GOING

When nothing needs you

STOP AND ASK

Blocked, or destructive

UNATTENDED

**Prod, deploy, push: your
explicit yes**

Context is the scarce resource



CLAUDE.md

Every session



Skills

Loaded when relevant



Hooks

Every time, enforced



Subagents

Their own context

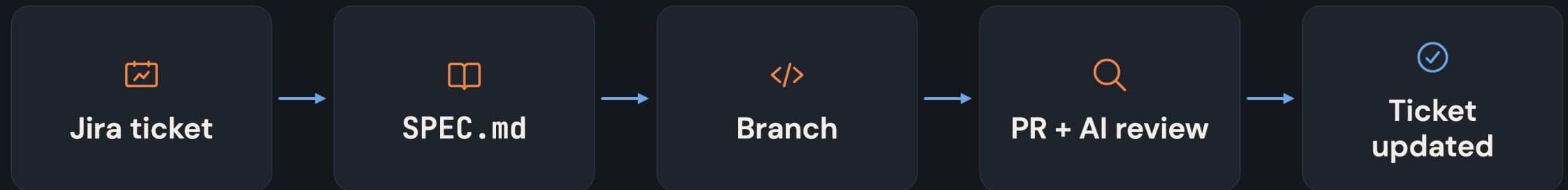


MCP

Tool defs use context

Prefer CLI tools (gh, gcloud) · **Vet every plugin** like a package

From ticket to pull request



AGENT MAY

Read tickets and code, comment, draft the PR

CLI first (gh). MCP where there is no CLI.

HUMAN APPROVES

Merge, status changes, anything destructive

Least-privilege tokens. Read-only by default.

Spec before code

1 • Explore

2 • Plan

3 • Implement

4 • Commit

Plan mode: reads, asks, changes nothing

1 • Claude interviews you



2 • It writes **SPEC.md**



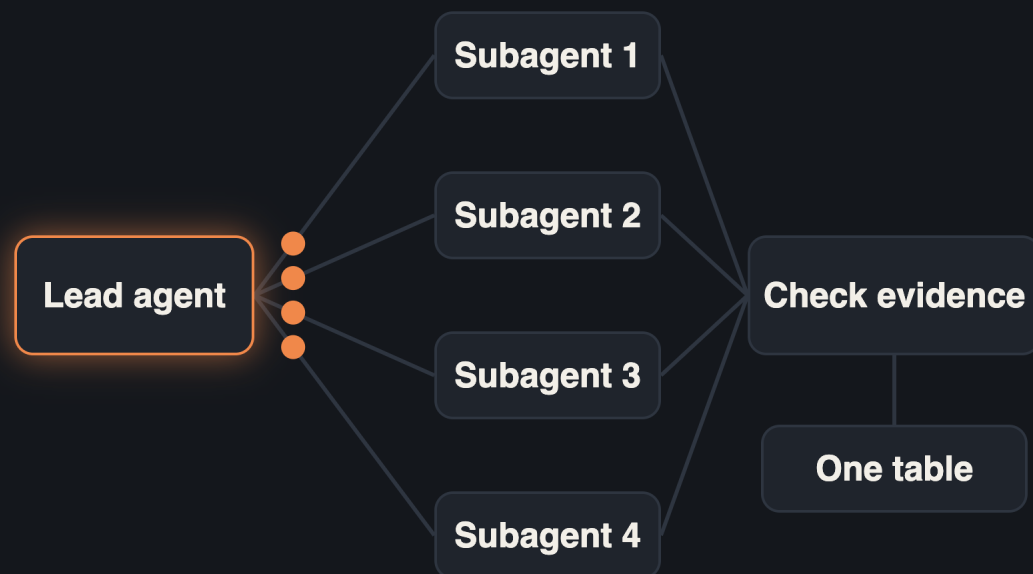
3 • A fresh session builds it

One-sentence diff? Skip the plan.

SPEC.md

- ✓ Goal and finish line: "Done means..."
- ✓ Files, interfaces, layers involved
- ✓ Out of scope
- ✓ Tests and an end-to-end check
- ✓ When to stop and ask

Delegate big work, check evidence



Independent tasks in parallel · small ones stay local

ADVISOR

Second opinion at decisions

`/advisor opus`

Fast main model + stronger advisor · same-tier advisor for high stakes

OBSERVER

A second agent watches

Catches silent decisions and false success

Unasked behavior change = a question

Effort dial, not “think hard”

Level	Use it for
low	Quick edits, first review pass
medium	Default. Most coding work
high	Hard, ambiguous changes
xhigh / max	Only with a measured gain

Less thinking? Lower effort first.

Model by task: strongest for hard code and review, cheaper for tests and scripts

Give it a check it can run



Tests first

No test edits without approval



TASKS.md

Survives compaction. No TODO without evidence



Hooks and /goal

Per change: lint and touched tests. Full suite per batch



Fresh session

After two misses or a full context. Point to docs

Review in layers, not one gate

2×

More merged pull requests

-40%

Share of PRs reverted

Claude Code team, June to September 2026

People where the risk is

0

HUMANS, ONCE CHECKS
ARE GREEN

Low risk

Docs, copy, additive changes behind
a flag

1

HUMAN APPROVAL

Medium risk

Ordinary product changes

2

HUMAN APPROVALS

High risk

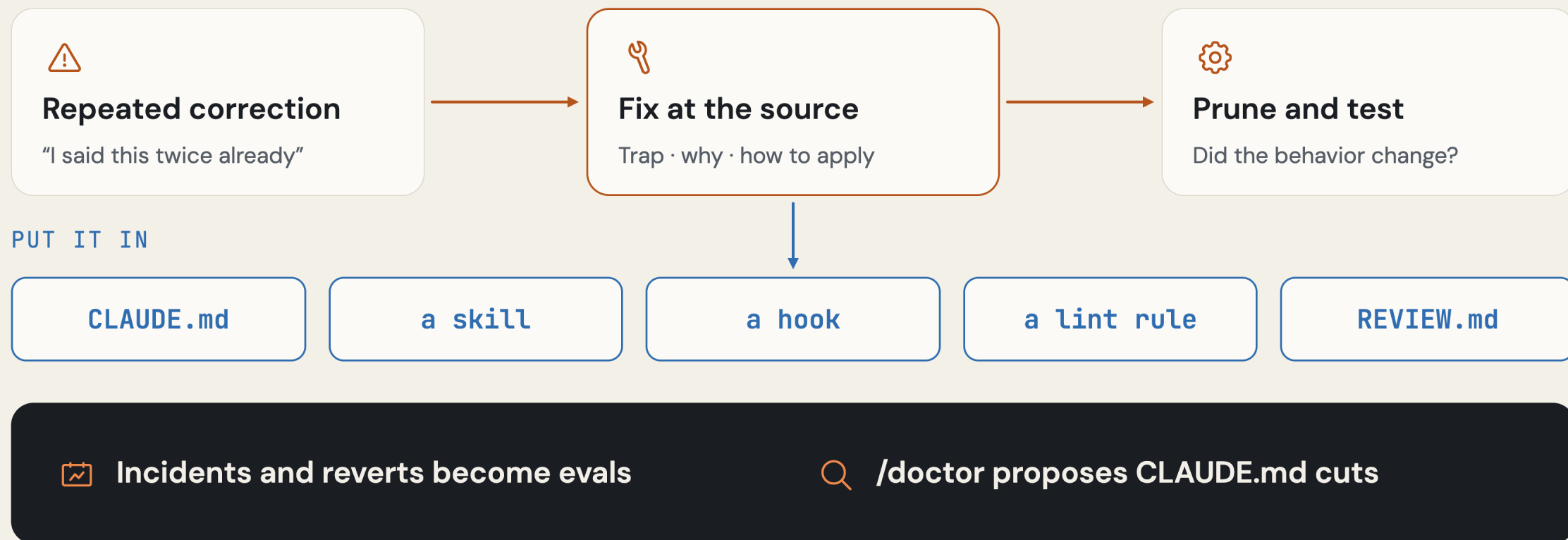
The riskiest paths you name: auth,
billing, CI

**The change sets the tier, not
the author**

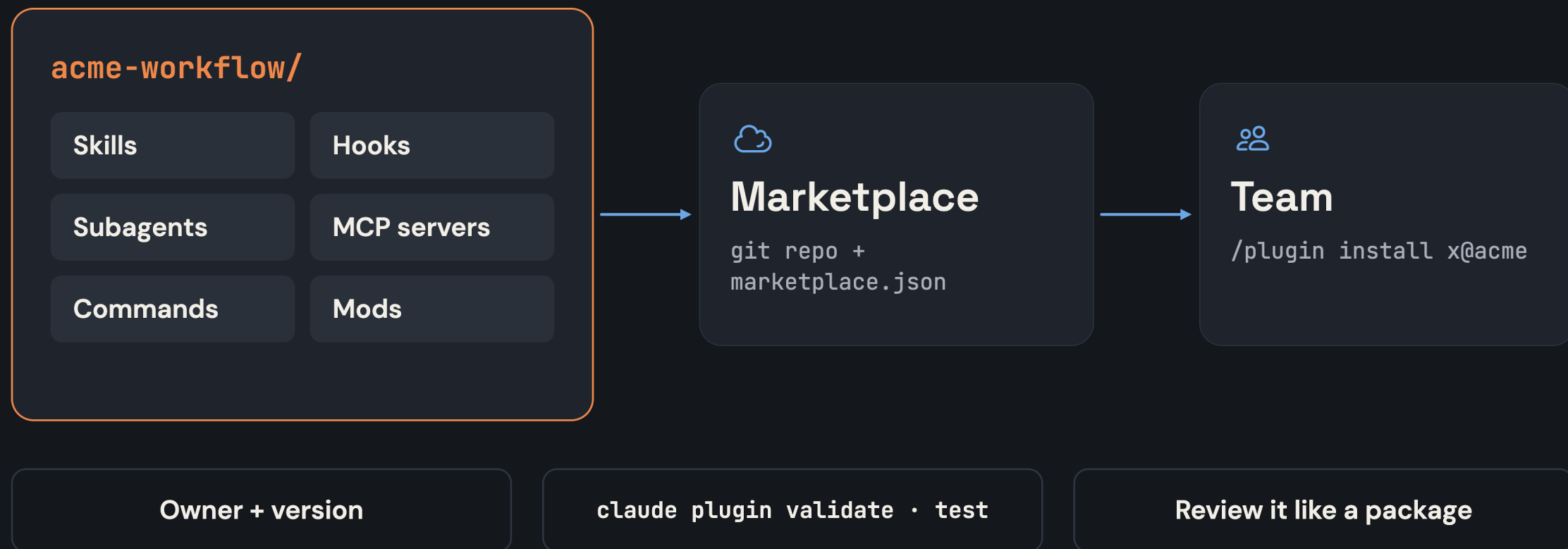
**Whoever prompted the change
owns it**

**A bot approval never replaces a
human one**

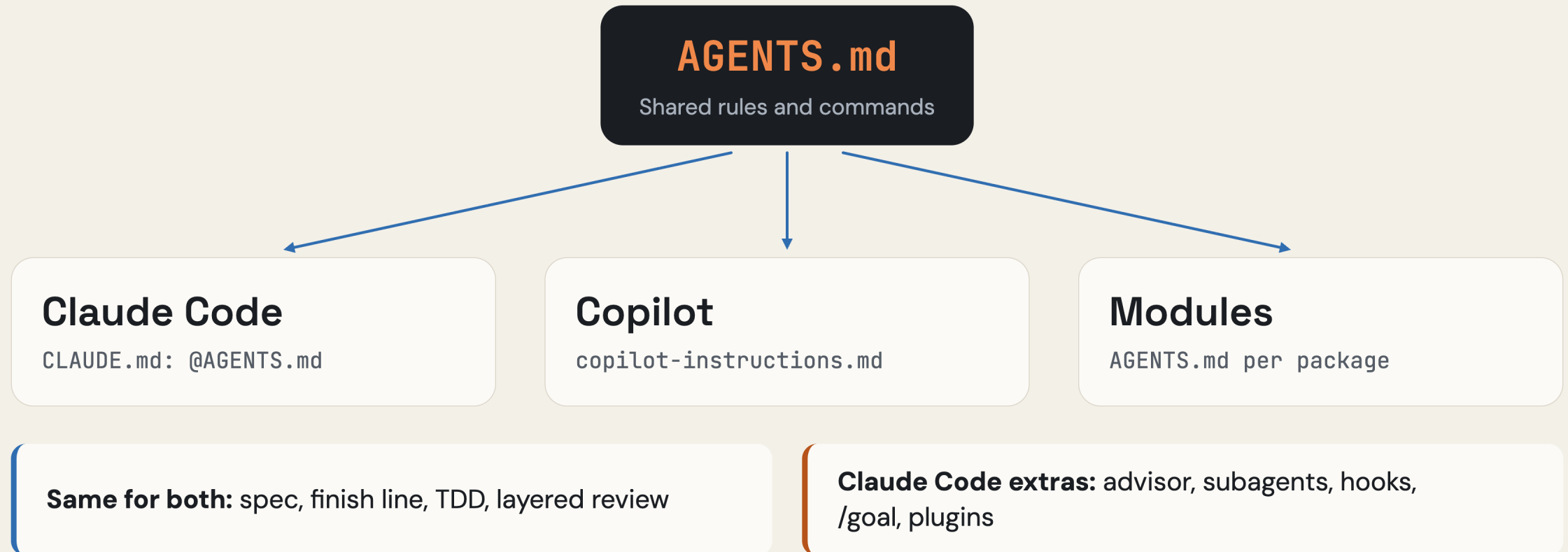
Fix it once, at the source



Package what works, share it



One base for every tool



Your checklist

Ask

- ✓ Whole task with a finish line
- ✓ No “think hard” lines
- ✓ Model and effort set up front

Long runs

- ✓ Stop rule in CLAUDE.md
- ✓ Context and permissions set before you leave
- ✓ Parallel by default, full suite per batch

Check

- ✓ A check it can run
- ✓ An observer agent beside the main one
- ✓ Green ≠ verified

Team

- ✓ Repeated asks become written rules
- ✓ No TODO without evidence
- ✓ AGENTS.md as the shared base

IN ONE LINE

**Hand over the whole task. Give it
a way to check. Own what ships.**

Questions?