

The Chef's Kitchen Loop for Claude Code

Your 3-Tier System to Stop Burning Your Best Model on Grunt Work

01

The Three Kitchen Tiers

Run your Claude Code setup like a real kitchen: the head chef designs and tastes, the line cooks do the prep. Match the model to the job, not the other way around.

QUICK TIPS

- Head Chef (Opus 4.8): designs the menu — writes the plan, the spec, the acceptance criteria — then tastes every plate by reviewing the final result.
- Sous Chef (Fable 5): does the skilled cooking — the actual implementation and heavy building.
- Line Cooks (Sonnet / Haiku): handle prep work — boilerplate, refactors, renames, tests, repetitive edits.

02

Why This Stretches Your Limits

Sub-agents draw from the same usage meter — they aren't a separate free budget. The savings come from not spending premium-model tokens on work a cheaper model handles just as well. Fewer expensive tokens on prep means your top-tier capacity lasts across far more real work.

03

Setup: The Executor Sub-Agent

Build one sub-agent whose only job is execution — never planning.

QUICK TIPS

- Create a sub-agent whose only job is execution.
- Point it at the cheaper model (Sonnet for skilled work, Haiku for pure prep).
- Give it the plan as input — it never plans, it only builds against the spec your head chef wrote.
- Route the review back to your top model — it checks the output against the acceptance criteria before anything ships.

Setup: The Loop

Once the executor and reviewer roles are wired up, close the loop so the system runs itself.

💡 QUICK TIPS

- Define the goal + criteria up front — "done" means it passes these checks, not "the model stopped talking."
- Loop until criteria are met — wire a check-and-retry so the executor keeps working, hands back to the reviewer, and only exits when the plate passes the taste test.
- Schedule it — once the loop is trustworthy, set it to run on its own: batch overnight, on a trigger, or on a cadence.

Before You Rely On It

Three things to confirm before you trust this loop with real work.

💡 QUICK TIPS

- Confirm the loop/goal mechanism in your current Claude Code version — name the real command or custom slash command you're using; don't assume a built-in exists.
- Watch the reviewer's standards — a weak acceptance spec means the loop "passes" garbage. The criteria are the whole game.
- Check your own usage first before quoting a multiplier — mechanics change, and you want your numbers to hold up.

Ready to go deeper?

Get GrowthOS. The AI Business Operating System You Own. 🏠 Build
Once. Own It Forever.

Want to go deeper? →

<https://builds.digicuratoragency.com/growth-os/>