Start here

How to build software you can sell by running Claude Code or Codex as your team

By Brooke Lacey

You don't have to write the code. Claude Code and Codex write it, test it and put it online when you ask well. What they can't do is decide what's worth building, tell when it's really done or catch what they got wrong. That's your job now, and at a funded startup it's the work of a whole team.

You decide	The agent does
• Who it's for and what they'll pay • What version one does and doesn't do • What "done" looks like • What the agent must never touch • Whether it shipped	• Plans the change • Writes the code and the tests • Runs them and fixes what fails • Commits and deploys • Reports back to you

1 Pick a problem someone already pays to solve

Start with a person, not an idea. If you can name their job title, a task they repeat every week and what they use for it today, you have a product to test. Spreadsheets, copying between two tools and a weekly report someone builds by hand are good places to look.

Before you build, ask five of those people if they'd pay for it. If one says yes and names a price, start.

2 Set up the workspace once

- **A GitHub repo,** so any change the agent makes can be undone.
- **Claude Code or Codex in your terminal.** Both run on a paid Claude or ChatGPT plan.
- **A rules file** at the top of the repo: `CLAUDE.md` for Claude Code, `AGENTS.md` for Codex. The agent reads it every session. Put in what the product is, how to build and test it, and what the agent must never do.
- **A memory folder** with one short file per thing the agent learns, like a bug or a decision, so tomorrow's session doesn't relearn it.
- **Hooks,** small scripts that run on every action the agent takes. Add one first that blocks any commit containing a password or API key.

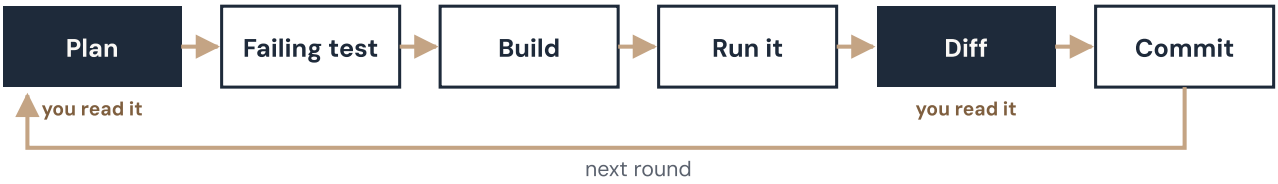
3 Write the brief before any code

One page. Hand it to the agent and ask for a plan, not code. If the plan surprises you, fix the brief and ask again.

Who	Office managers at small dental practices
The job	Turn the week's appointment export into the Monday staffing sheet
Done	A manager signs up, pays, uploads the export and downloads the sheet
Not in v1	Scheduling, insurance, a mobile app

4 Build in small rounds

One feature per session. If a round makes more change than you can read in one sitting, it's too big, so split it.



The agent does every box. You read the two dark ones: the plan before it starts and the diff, the list of every line it changed, before it commits. You don't need to write code to read a diff. You're checking that it changed only what you asked for.

5 Never take "done" on faith

Agents report success with total confidence, and sometimes they're wrong. Done means you watched it work: the page loads on a phone, the form saves a real record, the email arrives. Ask the agent to run a Playwright browser test on the live site and show you the screenshots. When something breaks, ask for a test that fails first, then the fix.

6 Put it online early and charge from day one

Get it on a real web address as soon as one screen works. Next.js on Vercel, Supabase for the database and logins, and Stripe for payments is a stack both agents set up well. Add checkout before you add features, because your first paying customer will tell you what to build next.

The team you're running

1 Chief of staff			Opens first on zosha.dev
2 Product manager	3 Software architect	4 UI/UX designer	5 Full-stack developer
6 Code reviewer	7 QA engineer	8 Security engineer	9 Database engineer
10 DevOps and release	11 Site reliability	12 Technical writer	13 Growth and analytics

Where first products go wrong

- **Keys in the code.** Passwords and API keys go in environment variables, and a hook keeps them out of commits.
- **One giant prompt.** Ten small rounds you can check get further than one big ask you can't.
- **No rules file.** Every session starts from zero and repeats last week's mistakes.
- **Months of building before anyone pays.** Sell it first, then build what the buyer needs.
- **Merging what you didn't read.** You won't write the code, but you still read every diff.
- **Trusting the summary.** Open the live site and check it yourself.

Zosha Dev teaches each of the 13 roles on video, with drills checked in your own repo. zosha.dev

© 2026 Jayla Labs, Inc.