A fractional CTO's view when the playbook predates AI-native engineering

By Andy Slack | November 10, 2026 | AI Development

How to spot a technical playbook that predates AI-native engineering, raise it without a fight, and choose what to do next.

You have a prototype, an MVP or a live product, and delivery feels slower and more expensive than it should. The question is the playbook, not the person leading the work: how changes are specified, tested, reviewed and released. This page is how to spot a playbook that predates AI-native engineering, how to raise it without a fight, and the options in front of you. It is written for a nontechnical founder.

How a change is specified, and how agents write and review it, already have their own pages: spec-driven development with AI coding agents (/articles/spec-driven-development-ai-coding-agents) and the agentic development loop (/articles/agentic-loop-development). This page does not repeat them. It is about the conversation with the person already leading your product.

How to spot a playbook that is holding delivery back

In our onboarding, the technical assessment covers the product, its code, its infrastructure and how the team works (confirmed with us on 10 November 2026). The same assessment is one outcome we publish on the homepage (/): a technical assessment of your product, code and infrastructure. The signs that the way of working is holding delivery back are the places it falls short of how we work.

- Work does not start as a written issue with acceptance criteria.
- Changes ship without tests that run on every change.
- Decisions live in one person's head, not in the repository.
- Releases come in large, rare batches, not small, frequent ones.

None of these is a verdict on the person. A capable lead can be running a playbook that fitted a time before an agent could write the spec, the tests and the code. The four signs are things you can ask to see.

What you can check without reading the code

Day to day, this is how we work (confirmed with us on 10 November 2026). Every change starts as an issue that states the outcome and three or four acceptance criteria. An agent reads the code and writes the full spec on that issue, with its decisions and reasons. One agent writes failing tests from the spec. A separate agent writes the code to pass them, and it may not rewrite those tests. An AI code review agent reviews the pull request, and work stops whenever a decision needs a person. After the first release, releases go out daily.

You do not need to read the code to see whether this is happening. Check five things: each change has a written issue with acceptance criteria; tests ran and passed on it; there is a record of why each decision was made; releases go out often; and the code and accounts are in your name. The spec, the tests and the review are the parts the two articles above already cover. The last check is ownership, which we also publish on the homepage: your code, accounts, documentation and engineering environment stay in your name.

How to raise it

There is no secret script. Honest communication through trust is what matters, and we like the book Radical Candor (confirmed with us on 10 November 2026). Talk about the playbook, not the person. Take the five checks as the agenda for the conversation: ask to see a recent issue, the tests that ran, the written reason for a decision, and when the last few releases went out.

Say what you noticed, ask what the current way of working is for, and agree one change to try. A written issue for the next piece of work is a small first step. It tells you quickly whether the playbook can move.

Upskill, work alongside, or change who does the work

We normally aim to upskill the existing team and work alongside them, so the required changes improve throughput. Where that is not possible, we look to replace team members with higher performers (confirmed with us on 10 November 2026). Teach and work alongside first. Change who does the work only when that path will not get the throughput.

Working alongside an existing technical lead can happen. It is not normally why we are hired. The published offer is for nontechnical founders who have a prototype, an MVP or an existing product and lack trusted technical leadership, as the homepage (/) says. A founder who trusts their lead and wants that lead to keep the technical decisions is not who the offer is built for.

If two technical leaders are on one product, write down who decides architecture and delivery before the work starts. Two people cannot both own the technical decisions. In our offer, the fractional CTO is accountable for technical strategy, architecture, technical decisions and delivery oversight. You decide customer discovery, demand validation, commercial decisions, final product priorities, scope and budget.

What stays with the product

We have no separate practice for an outgoing lead (confirmed with us on 10 November 2026). What we do have is this. When we take over an existing system, we assess it before we commit. The homepage states the same limit: if you have an existing system, we assess it before we commit to take it over. In our way of working, each change's decisions and reasons are written on its issue, and the specs and tests live in your repository, so what the product depends on stays with the product, not with one person.

If the fractional CTO changes, the handover is managed: you get advance notice, an introduction to your new CTO and documentation of your product and its decisions. That line is on the homepage too.

What our fractional CTO services cover, and when we decline

The offer is one accountable partner: a fractional CTO who sets the technical direction and leads an AI engineering team that builds, tests and ships the product. The public price is USD $9,999 setup and USD $2,999 per month. The fees exclude applicable taxes, AI usage, hosting and third-party tools. There is no minimum term, and you can cancel with 30 days' notice. Any increase to the budget needs your approval, and nothing increases automatically. Output follows the AI budget you agree with us, not a fixed number of features. A named fractional CTO, backed by JuicyLlama founder oversight, is accountable for the technical side. All of that is published on the homepage (/).

If the fractional CTO service is not right for you, we will say so and decline, or offer a tailored scope of the same fractional CTO service. We will not sell you a different service. Before the call we read what you say about your current technical leadership, and whether you can fund the published price. The founder fit form asks for both.

If you want to talk through the playbook on your own product, book a founder fit call (/#lead). There is more in our articles (/articles).