

I run a private AI agent system. Here is what that means.

The short version

I direct coding agents to build and maintain a working system. Not a chat window: persistent memory, scheduled jobs, a messaging bridge, a self-hosted client, and checks that test the system's own output for failure. It has been running daily for weeks. I do not write the code. I specify the work, break it into tasks, verify the result, and take the responsibility for whether it holds.

That is the work employers are hiring for under titles like *AI enablement*, *AI operations*, and *automation specialist*. It also happens to be the work I have been doing on my own time, without being asked, since those titles meant nothing to me.

What it actually is

- **A memory that survives the session.** Files are the memory: read at the start of a session, written at the end. Nothing important lives only inside a conversation window.
- **The same agent, several surfaces.** It runs in a desktop app, a terminal, and a chat platform, with the same behavior and the same memory in each. The surface changes where it is, not what it is.
- **Scheduled and unattended work.** Jobs that run on a clock with nobody present, and report back when they are done.
- **A client I built rather than adopted.** A self-hosted interface over the engine, with the record readable live inside it.
- **Checks that read the system's own records.** Automated instruments that flag claims never verified, output that repeats itself, documents that contradict each other, and the pattern of hours a session keeps. When a check disagrees with my instinct, the check wins until I can show it wrong.

What I have built

The system is one output of the method. These are others.

- **An interface I built rather than adopted.** A self-hosted client over the engine, with the record readable live inside it. The look, the layout and the behavior were mine to decide, and they were iterated against daily use rather than a mockup.
- **Tools that remove recurring work.** Anything that repeats gets specified, built, verified, and then runs on its own. The task goes off the list once and stays off. That is the part of this I want to do more of.
- **A place to break things safely.** An isolated copy of the system where behavior is tested and measured before anything touches the real one: fixed cases with known-

good outcomes, run against every change. It finds what the happy path hides, and it is why a fault gets hunted down to its cause instead of worked around.

What I have taught

- **A person, a craft, and a real curriculum.** I designed a course and direct it: a student profile, a progress board, and lessons written to a fixed method. One action per step, a hint before an answer, never a lesson that ends without a next step, and nothing generated for the student. Every lesson produces something they made themselves.
- **A model taught to extend itself.** I taught an agent to build and train its own tools, and to measure its own accuracy against my eye instead of its own confidence. Directing an agent is mostly teaching it to catch its own mistakes, and this is where that was learned. The method is the point: specification, accuracy, and a human standard held as the way it corrects itself.
- **And on a retail floor, the same thing in miniature.** Associates brought me the customer objections they were stuck on. I took the sale from where they left it, and afterwards I walked them through what I had done. They kept the customer and learned the answer, instead of reading it off a sheet. **The fix stayed with the person. It did not replace them.**

The two habits that make it work

Write it or lose it. Anything not written down does not exist. This is not tidiness. It is the only reason a months-long project survives interruption, illness and time. Sessions end; the record does not.

Corrections live beside the error, never over it. The record is append-only where it matters. When I get something wrong, the correction is written next to the mistake instead of replacing it, so the reasoning stays auditable. A clean-looking record that has quietly papered over its own errors is worth less than a messy one that hasn't.

What broke, and what I learned

This is the longest section on purpose.

- **I trusted a summary over the source.** A number I had read once, carried forward, and stated as current. It cost me a wrong claim in the record. The fix: checks that force a value to be re-read rather than remembered, plus a standing rule that a number I cannot point to is a number I do not say.
- **I invented a cause and defended it.** A fault looked like a bad tool. It was a network drop outside the system entirely. The fix: a rule that when something looks like the system, check the human and the environment first, and that a theory gets killed by a control test before it is reported as a finding.

- **Two theories of mine died to their own control tests in one afternoon.** Both looked right. One lined up perfectly with the evidence until I ran it against healthy cases and found it lined up with those too. The habit I kept: **run the control before believing the finding.**
- **I wrote a change into the record that was wrong and caught it hours later.** The fix was not to quietly edit it. It was to log the correction and the reason alongside. The scars are what make the rest trustworthy.

The working receipts

Three pieces of work taken end to end, no payment asked. **The wins sit here because the losses sit above them.**

- **A performance sweep on someone else's PC.** Audited before anything was touched, fixed in tiers, every change reversible, and an undo kit left on the machine. **Measured result:** memory running at its rated 3600 instead of 2133, eight startup programs cut, the sleep settings that cause controller dropouts turned off, an unused 2.1 GB app removed, about 1.1 GB reclaimed. Audited again at the end, and the cleanup verified rather than assumed.
- **A rescue on an editor's knowledge base.** A desktop tool that would not open after a version change, and the error said nothing useful. Diagnosed down to a single configuration directory, moved aside, and the whole library came back with nothing lost. **It became the procedure:** the same diagnosis is written up as a playbook, and it is what made the follow-up hand-off work.
- **An audio repair bench.** Built first for a music rig, then pointed at a creator's real 48-minute recording. **Result:** 941 broken joins mapped, 233 bump spots repaired, a de-click pass over the faults, and the level lifted about 17 dB to a broadcast master, delivered beside an edit copy and matched before-and-after samples. **And it holds:** the tuning is saved as presets, so it is set once and stays set. Drop a file in, ask for it cleaned, get both outputs back ready for the editor, the original untouched, the same the next time. **Then the cause:** the files had kept arriving unusable because of a compensation setting in the editor itself, so the work ended with the reset written down and handed over. Set once, and it does not come back.

Small jobs, done properly, on somebody else's hardware and somebody else's deadline.

How I work

- **I specify before I build.** The problem, the constraints, what "done" means, and what I am willing to trade away.
- **I break work into tasks an agent can finish,** and keep the ones that need judgment for myself.
- **I verify rather than trust.** The check is cheap. Being wrong in front of someone else is not.
- **I say when I am unsure,** in the same sentence as what I am doing about it.

- **I work alone comfortably, and I document as I go,** so the next person or the next session can continue without me standing there.
- **I am not stopped by unfamiliar problems.** That was most of this.

What I want next

Work where running agents is the job: taking recurring work off a team's plate, building the automations, keeping them running, and teaching people to use them well. Remote suits me and the work. **I would rather be handed the hardest recurring problem you have than the easiest one, because I work a problem to its end.** Whether that end is solved or established as unsolvable, the answer is where I stop.