

Ran Aroussi: How I Work

The blueprint - setup, systems, and the decision everything hangs on

Companion to the Old School New Tech solo episode · aroussi.com

People keep asking how I get through it all - the agency, the books, the podcast, the side projects. This is the honest answer. Not a productivity system you can install, just the setup I've arrived at and the reasoning underneath it.

My job used to be doing the work. Now it's deciding which bucket the work goes in.

1. The one idea

I make 20 to 30 decisions a day and almost all of them are classifications. Two tracks, over and over:

- Rails or agents
- Bespoke or general
- Human, automated, somewhere in between - or time to kill the process entirely

Bezos said he gets paid to make a small number of high-quality decisions. I'm not running Amazon, but that's the target: a small number of good calls, and the rest of my time spent on the part I actually enjoy, which is building things. Including plenty that will never ship.

Everything below - the machine, the lists, the pipeline - exists to protect the context in my head so I can keep classifying all day. Everything else is scaffolding.

2. The automation rule

Anything I've done more than a few times, and have a process for, gets automated. Automated does not mean AI. It sorts into three buckets:

BUCKET	WHEN	WHAT IT LOOKS LIKE
Deterministic	Known inputs, known outputs	Plain automation. No model anywhere in the loop.
Hybrid	Mostly fixed flow, one fuzzy step	Deterministic pipeline with an agent dropped in at the judgment point - write the email, make the call on this branch. Everything before and after stays dumb.
Agentic	Genuinely fuzzy, needs judgment	An agent owns the task end to end.

Most of what I run is the middle one. The instinct to make the whole workflow intelligent is usually wrong - it's slower, more expensive, and harder to debug than a fixed pipeline with one smart step in it.

Why almost everything is gated

Most of my automations require human approval at some stage. That's on purpose, and it isn't caution about the technology - I'm all in on the technology. It's accuracy about the domain. Every client we serve is an edge case. The processes can look nearly identical between them and still diverge in ways that matter.

The dial between fully automated and human-gated is technically a setting. In practice it's a judgment about how we want to work. AI should take the dirty work off my plate. It shouldn't replace me in communication, especially not with clients and partners.

3. The setup

The environment question

Local or cloud is a real debate and both sides have a point. My answer is that my local environment *is* the cloud: a dedicated machine in the home office, accessible to me and only me, from anywhere. Fully configured the way I want it, safe to run experiments on, and reachable from a laptop or a phone. That's what both camps are actually asking for.

Two modes of coding

Exploratory - still figuring out the shape of the thing. Droid as the harness, driven through a modified Paseo build. This is also what lets me work from mobile, which matters most for long-running sessions that don't need much back-and-forth.

Mature - the product is cooked, now it's features and fixes. That runs on Cloop, our internal Devin-style environment, with a brainstorming module for new features and Ladybug for hunting production bugs end to end, from discovery through PR. The human is always in the loop at the PR.

Everything funnels into Droid underneath, so sessions stay consistent no matter where I started them.

The stack

TOOL	USED FOR	LINK
Droid	Primary coding harness, runs all models	factory.ai/product/droids
Paseo	Exploratory and mobile coding, orchestrates agents across devices	paseo.sh
Cloop	Mature-codebase work (internal, - shipping externally)	

MUXI	Personal assistant, briefings, email triage and drafts	-
Claude	Brainstorming, speccing, research; also the MCP client for MUXI	claude.ai
ChatGPT	Second opinion during brainstorming loops	chatgpt.com
Perplexity	Research, freshest sources	perplexity.ai
VS Code	Writing environment for AsciiDoc book work	code.visualstudio.com
Superhuman	Email	superhuman.com
Apple Reminders	To-do list	apple.com/apps
Apple Notes	Quick notes	apple.com/apps
Slack Connect	All client communication	slack.com/connect
Discord	Internal team comms at Automaze	discord.com
Cloudflare	Hosting and edge	cloudflare.com
Hetzner	Servers	hetzner.com
DigitalOcean	Servers	digitalocean.com
Envapor	Commit env files safely instead of maintaining templates	-

For client work we use whatever fits the client. The above is what I reach for on my own projects.

Self-hosting

I self-host almost everything I can. It has nothing to do with cost - I spend tens of thousands on tokens, so saving thirty dollars a month on a subscription isn't the point. It's data ownership and governance. I want to reach the database directly when I need to. I'm happy to pay for software. Just let me host it.

4. Where the time goes

Thinking, not typing

Calls, clients, team syncs - all real, all less frequent than you'd guess. Most of my time goes into thinking, and a lot of that thinking is about shrinking the codebase, because a smaller codebase is a smaller maintenance surface.

The recurring exercise is first principles. Even on products that have been live for years and have real users, I'll ask what the thing actually *is* and strip it to the core. What is a coding agent? The Kanban boards and workflows are secondary to a system that can code. Strip to essentials, then add features back deliberately. The result is less complexity, and code that's easier for both people and agents to maintain.

Which means a lot of my output is decisions about what *not* to build. I'll spend hours, sometimes days, occasionally weeks on brainstorming and spec'ing before a line gets written.

The brainstorming loop

Claude for hypothesizing and thinking out loud - not because it's smarter, but because I find it easier to talk to. Then I take the output to GPT and ask what it thinks, feed that back to Claude, and go around again. Perplexity for research, where being current matters.

I'm deliberately the human in the middle of that loop. It's fuzzy, it's the part I enjoy, and it's where judgment actually shows up. I don't plan to automate it.

Reading the code

I read everything the agents produce, and I comment on it heavily. Generation is cheap now. Judgment is still expensive. What I bring isn't typing Python faster than anyone else - it's seeing problems in a codebase or a production environment before we get there.

Email

Open twice a day: once after the morning briefing, once at the end. There's a short VIP list that gets a fast reply, and it isn't ranked by importance - it's about not being the bottleneck. An accountant with something urgent uses a specific address. Routine bookkeeping goes in the normal pile.

5. Running the business

Lean founder, not solo founder

Automaze is close to thirty people and I mostly don't run it. Anton and Alex run the agency - team, pipeline, builds. My job is winning and keeping customers, writing the PRDs, and deciding how products get built. Then the team takes over and I come back for the last ten percent.

Where leads come from

Ninety-nine percent inbound. Open source people find and trace back to me, the podcast, the newsletter, the books. None of it was built as lead gen - it accumulated organically over fifteen-plus years. That's the unsatisfying part of the answer: it isn't a strategy you can run next quarter. It's a byproduct of giving things away for a long time.

The FDE decision

Forward-deployed engineering is having a moment, but embedding directly in the client's team is what we've always done. What I do specifically is act as the FDE myself for every new client, and stay in the room for every new process we implement.

People ask whether that makes us look small. I don't think it does - it's a design decision. I can't confidently talk about something I haven't experienced first-hand, and I can't improve a process I don't understand. Small number of high-quality decisions, and there's no high-quality anything without understanding first.

So I'm there at the start, the team takes over, and I come back at every fork in the road. Clients tend to like having the founder in the room working it out with them. For a company our size, I think that's the founder's job.

More at **aroussi.com** - newsletter, podcast, and open source.