

Make the agent its own QA: OpenAI's computer-use advice, Anthropic's scaling bottlenecks and Dots for engineering

Coding Agents Alpha Tracker

2026-10-01

Make the agent its own QA: OpenAI's computer-use advice, Anthropic's scaling bottlenecks and Dots for engineering

By Coding Agents Alpha Tracker • October 1, 2026

OpenAI's DevDay team and a summary of Boris Cherny's talk both point to the same next step: have agents test and verify their own work before a human reviews it. Also covered: Dots as a coding orchestrator, Gemini 4 Argon's limited launch, and new friction around MCP.

Have the agent test its work before you look at it

Two sources this period make the same point. Once agents write code quickly, the slow part is manual testing and review, so hand that to agents too.

On Latent Space, OpenAI's Ari Weinstein named this as one of his favorite computer-use cases: the agent tests the software it built, "by the time it comes to me it's already working." Otherwise, he says, you end up as QA for your agent [1]. One of the hosts uses a "visual play test skill" that catches design issues you wouldn't spot by reading code [1]. Concrete details from the interview:

- **Appshots in Codex:** press the Command key twice. Unlike a screenshot, an appshot includes the raw accessibility representation, such as where links go and full calendar event titles, and the format is designed to use few tokens. To see what the model receives, click the attachment and then the small button in its top right [1].
- **Computer use now writes code.** If you expand the tool calls in Codex, you'll see the model running JavaScript that performs several actions at once. It switches between screenshots, accessibility data and Playwright depending on the task [1].

- **Agents API:** Weinstein suggests using OpenAI’s own computer-use harness rather than building one. The models are trained on it, which may help speed, cost and accuracy. Ask for user consent before consequential actions, and limit access to the sites the task needs [1].
- **Harness builders:** Nikunj (API lead) listed several features. The model can keep reasoning while a slow tool call runs (async tool calls). You can inject messages mid-turn. Both run over WebSockets [1]. You can pre-warm the prompt cache and spawn many threads from it [1]. For context, use either server-side auto-compaction at a token threshold or a manual `/compact`. The open-source Codex harness uses `/compact` and is experimenting with file-based approaches [1].

A video summarizing Boris Cherny’s talk lays out the same progression from Anthropic’s side. At 5–10 agents, manual testing becomes impossible, so Claude runs the end-to-end smoke tests. At 10 or more, review becomes the bottleneck, and Anthropic answered with automated code and security review. The video says this catches about 95% of incidents before production [2]. Techniques worth copying: - Give the agent the business problem and success metrics, not a step-by-step plan [2]. - Start a hard task at max effort, then type `/effort` to lower it for the easy parts. You keep the context and the cache [2]. - Have Claude run the tests and add a “Claude verified” commit trailer that links to the session transcript. For big architecture changes, have it generate interactive diagrams for reviewers [2].

Dots for engineering work

Rohan Varma (OpenAI) describes two ways to set up a dot for engineering: - **Connect it to your machine.** It controls the desktop, the Codex app and the in-app browser. - **Point it at a new cloud dev environment.** You define the environment and its dependencies, and the dot spins up instances on demand.

Either way, the dot acts as an orchestrator that hands tasks to other agents and moves them along, much as you push Codex threads forward today [3].

Tibo Sottiaux (OpenAI) describes his own setup in an interview, a dubbed transcript. One dot monitors Twitter. Another finds small documentation issues and opens PRs on its own, which he reviews and merges. Dots also send PRs to each other [4]. His advice is to delegate more as the agent learns your preferences [4], and to compare models on cost per completed task rather than token price [4]. He says the Codex app and harness work with any model, without restrictions [4].

Rollout status: Dots has reached all Pro500 and Pro200 users and is starting on Pro100. Business Premium and Enterprise beta slipped by a day [5]. The EU, UK and Switzerland are excluded for now [6].

Gemini 4 Argon: announced, but limited access

Google DeepMind introduced Gemini 4 Argon for coding, enterprise knowledge work and cyber defense. It is rolling out first to trusted testers in its Fairwind Program [7]. Google claims a new state of the art on real-world long-horizon software engineering tasks [8]. Logan Kilpatrick gives introductory pricing of \$2 in and \$10 out, with wider availability “as soon as possible” [9]. Nicholas Moy, who worked on the model, says he can’t keep it busy with enough work [10]. No independent results have been posted yet.

Sol: harness dispute continues, and capacity is coming

Artificial Analysis says it saw no significant gain from the Codex harness over mini-swe-agent at any effort level except low. It runs 3 repeats and asked Theo whether his results fall within confidence intervals [11]. Theo asked whether their mini-swe runs have vision/image tools, which he says “made a huge difference” in his runs [12]. If you benchmark models yourself, check which tools the harness gives the model, not just which harness it is.

Tibo calls Sol OpenAI’s most-demanded model “pretty much ever.” He expects ChatGPT and Codex to run close to twice as fast as the day before [13].

MCP friction

- **Allowlists:** Figma’s remote MCP server accepts only clients on its approved list, and Pi isn’t on it [14]. Armin Ronacher called this a failure to understand open protocols [15]. Peter Steinberger says it is “incredibly easy to work around” [16]. A reply in the thread said “We plan to support Pi”; Ronacher still wants it “open indiscriminately” [17, 18].
- **Ignored instructions:** Kent C. Dodds says Bot and Cursor don’t inject MCP server instructions, according to the agents themselves. If you rely on those instructions, check that your client actually passes them to the model [19].
- **Hosted MCP in ChatGPT:** ChatGPT Sites can now host MCP servers. A prompt like “@sites create a todo list that I can use in ChatGPT” creates the server, deploys it, turns it into a plugin and installs it on web, mobile and desktop [20]. You can restrict access to specific people or share it publicly [21].

Smaller tool notes

- **Codex:** Brent Traut suggests dropping Worktree threads. Normal threads are much faster, and the model creates worktrees when it needs them [22].
- **T3 Code nightly** adds “Start thread with no project,” which creates a new directory under `~/t3` for each thread [23, 24].

- **OpenClaw** now shows messages between agents as one expandable line in the chat stream (PR #161656) [25].
- **LangChain:**
 - Managed Deep Agents 0.8 adds HTTP channels, so any webhook can trigger an agent [26]. You supply your own functions to verify and parse incoming requests [27].
 - LangGraph interrupts can now take typed response schemas. Human responses are validated before they reach the agent, and you can render a real form in your UI [28].
 - LangSmith Engine v2 reads an agent’s traces and repos, then red-teams it for verified weaknesses [29].
- **OpenAppa (Archestra)**, from a sponsored Fireship segment: a policy layer that sits between Claude Code and its tools. Once a session reads a file marked private, it blocks outbound actions such as opening a public GitHub issue until you approve [30]. Fireship reports it finished 75% of jobs against up to 96% for Claude Code auto mode, and used more tokens. It is in preview and MIT-licensed [30].

Practitioner views

DHH says every new piece of code he has written since December starts as a prompt, and he reads the code only when an agent drifts [31]. He says agent work comes back ready for review in about seven minutes, a feedback loop that made delegating something he actually enjoys [31]. Theo argues that working with only partial knowledge of a large codebase is normal, with or without AI [32]. Salvatore Sanfilippo says benchmark scores have little to do with real agentic coding. He prefers capped flat subscriptions because the fixed ceiling covers concurrent sessions across several projects [33].

Sources

1. Inside OpenAI DevDay: Superhuman Computer Use, Decisions API, and the AI Cloud — Ari & Nikunj
2. Boris Cherny on future of Coding
3. X post by @TheRohanVarma
4. AGI Is Here. I Asked OpenAI What Happens to Our Jobs.
5. X post by @embirico
6. X post by @embirico
7. X post by @GoogleDeepMind
8. X post by @Google
9. X post by @OfficialLoganK
10. X post by @thenickmoy
11. X post by @ArtificialAnlys
12. X post by @theo
13. X post by @thsottiaux

14. X post by @GayaniFigma
15. X post by @mitsuhiko
16. X post by @steipete
17. X post by @zoink
18. X post by @mitsuhiko
19. X post by @kentcdodds
20. X post by @mxstbr
21. X post by @thsottiaux
22. X post by @btraut
23. X post by @theo
24. X post by @theo
25. X post by @steipete
26. X post by @LangChain
27. HTTP Channels for Managed Deep Agents
28. X post by @sydneyrunkle
29. X post by @LangChain
30. Did a 50 year old military secret just solve agent prompt injection?
31. Tech Genius Reveals Why The AI Revolution is Inevitable | David Heine-meier Hansson
32. X post by @theo
33. Opus 5.5 e alcune tristi considerazioni sul costo dei token