Devin 2.2, hybrid memory, and the shell-first agent stack

Coding Agents Alpha Tracker

2026-03-09

Devin 2.2, hybrid memory, and the shell-first agent stack

By Coding Agents Alpha Tracker • March 9, 2026

Today's strongest signal is that mature harnesses are finally cashing in on better models. This brief covers Devin 2.2 feedback, Cursor and LangSmith updates, hybrid memory patterns, agent-to-agent backchannels, and the security rules practitioners are using in production.

TOP SIGNAL

- **The biggest edge now looks like harness engineering compounding with better models.** After trying every Devin release, @dtcb says version 2.2 finally feels simpler than a local workflow and is now where he wants to debug [1]. swyx says that jump came from a process the team behind Devin has been running since late 2023: dozens of model groups, constant evals for routing, and full harness rewrites every few months [2].
- *Sam Altman's framing fits the moment: build a company that benefits from the models getting better and better* [3].

TOOLS & MODELS

- **Devin 2.2** — strongest practitioner signal of the day. One experienced user says it is now simpler than his local workflow; swyx says the underlying system relies on a couple dozen model groups, heavy evals, and periodic harness rewrites [4][5].

[1] post by @dtcb
[2] post by @swyx
[3] post by @swyx
[4] post by @dtcb
[5] post by @swyx

- **Enterprise deployment check** — Nvidia says Codex and Claude Code are already used by tens of thousands internally [6][7].
- **Cursor — GPT-5.4 Fast** — enable via *Settings > Models > GPT-5.4 Fast*. Reported tradeoff: **50% faster** for **2x the price** [8].
- **LangSmith Skills + CLI** — new terminal-native tooling so agents can debug traces, create datasets, and run experiments from the shell [9]. Details [10]
- **Super Memory plugins** — Dhravya Shah says a **Cursor plugin** is launching today; plugins already exist for Claude, OpenClaw, and OpenCode [11]. The OpenClaw integration switched from tool-triggered memory search to **hook-based** context injection under **2k tokens** per turn, with contradiction handling, temporal reasoning, and a **hybrid** RAG fallback when memory misses [12][13].
- **Memory eval reality check** — Shah argues **LongMemEval** overrewards extracting everything and ignores cost or forgetfulness, while **Locomo** mostly tests retrieval and can be brute-forced by dumping context. His team open-sourced **Memory Benchmark** to compare providers on shared rules across quality, latency, cost, recall, and NDCG [14][15].
- **GPT-5.4 vision -> code** — Romain Huet says GPT-5.4 is especially strong on dense documents, diagrams, and rough sketches, then suggests handing the result to Codex to turn it into software [16].

WORKFLOWS & TRICKS

- **If you are building an agent harness, copy Devin's routing pattern, not just its UI**
  1. Maintain multiple model groups instead of betting on one model [17]
  2. Eval every model before routing it into the harness [18]
  3. Treat the harness as a living system and rewrite it periodically as models change [19]
- **Use a private agent backchannel with an approval gate**

[6] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"
[7] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"
[8] post by @jediahkatz
[9] post by @LangChain
[10] post by @LangChain
[11] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[12] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[13] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[14] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[15] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[16] post by @romainhuet
[17] post by @swyx
[18] post by @swyx
[19] post by @swyx

1. Run `acpx` inside Codex [20]
2. Connect over ACP to OpenClaw and a remote agent like Molty [21][22]
3. Let the agents discuss privately [23]
4. Send into the live destination only after the target session approves it [24]

- Repo: acpx [25]

- **Terminal beats chat when the toolchain already exists**
  - Nvidia engineers say coding agents outperform more general agents largely because shell access gives them compilers, tests, and every installed tool, so they can write, run, inspect errors, and fix in-loop [26]
  - Concrete example: with an Outlook CLI installed, one engineer had Codex summarize a messy inbox, highlight escalations, move reply-worthy threads into a folder, and archive the rest [27]
  - LangSmith is productizing the same pattern by exposing trace debugging, dataset creation, and experiments through a CLI [28]
- **Memory that helps coding agents is hybrid, not just a folder of notes**
  - File-based memory can work, but Shah says it depends on explicit remember-this behavior, gets slow to traverse, and lacks update logic [29]
  - His replacement pattern: keep a tiny always-on user profile plus recent episodes, surface memories first, and fall back to raw RAG chunks when memory misses [30][31]
- **Hard safety rule for powerful agents**
  - Nvidia's rule of thumb: agents can access **files**, **the internet**, or **custom code execution** — but you should usually grant only **two of the three** [32][33]
  - If you need riskier setups, isolate them. Their example for OpenClaw is a Brev VM off the corporate network [34]

---

[20] post by @steipete
[21] post by @steipete
[22] post by @steipete
[23] post by @steipete
[24] post by @steipete
[25] post by @steipete
[26] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"
[27] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"
[28] post by @LangChain
[29] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[30] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[31] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[32] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"
[33] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"
[34] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and

- **Visual-to-code loop**
  1. Feed GPT-5.4 the dense doc, diagram, or rough sketch for interpretation [35]
  2. If the task is UI-heavy, connect a design surface like Paper to Claude Code or OpenClaw [36][37]
  3. Riley Brown's demo flow: install Paper -> connect Claude Code -> plan design -> generate designs -> iterate -> build the React app -> deploy [38][39]
- **100% agent-written code can still be disciplined**
  - Kent C. Dodds says he already has agents writing 100% of his code, but still steers the work and can read all generated code manually. His point: that is not the same as hands-off vibe coding [40][41]

PEOPLE TO WATCH

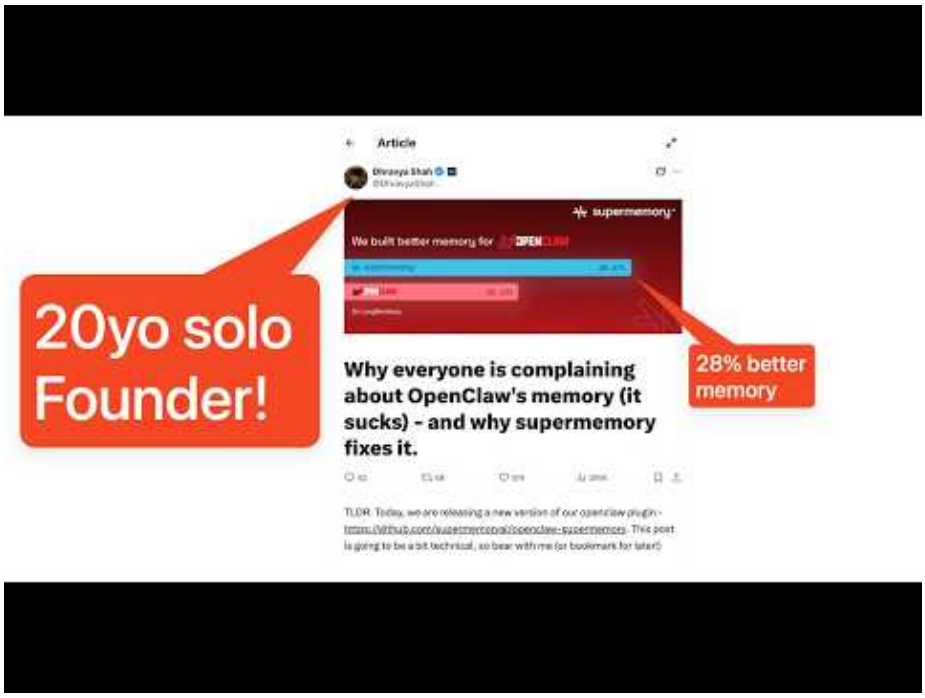
- **swyx + @dtcb** — best current read on why Devin suddenly feels good: same harness, better models, real user feedback [42][43]
- **Dhravya Shah** — rare mix of implementation detail and benchmark skepticism on agent memory; worth watching if you care about stateful agents more than leaderboard screenshots [44][45][46]
- **Peter Steinberger** — actively wiring Codex, OpenClaw, and ACP together in public; good source for multi-agent orchestration patterns, not just model takes [47][48][49]
- **Andrej Karpathy** — now pushing autoresearch toward agent communities coordinated through GitHub Discussions and PRs instead of a single linear branch [50][51]
- **Theo** — useful dissent. After hopping back into Claude Code for UI work, he says CLI agent UX is still awful compared with a real GUI [52][53]

"Speed of Light"

[35] post by @romainhuet
[36] post by @rileybrown
[37] post by @rileybrown
[38] post by @rileybrown
[39] post by @rileybrown
[40] post by @kentcdodds
[41] post by @kentcdodds
[42] post by @dtcb
[43] post by @swyx
[44] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[45] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[46] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[47] post by @steipete
[48] post by @steipete
[49] post by @steipete
[50] post by @karpathy
[51] post by @karpathy
[52] post by @theo
[53] post by @theo

WATCH & LISTEN

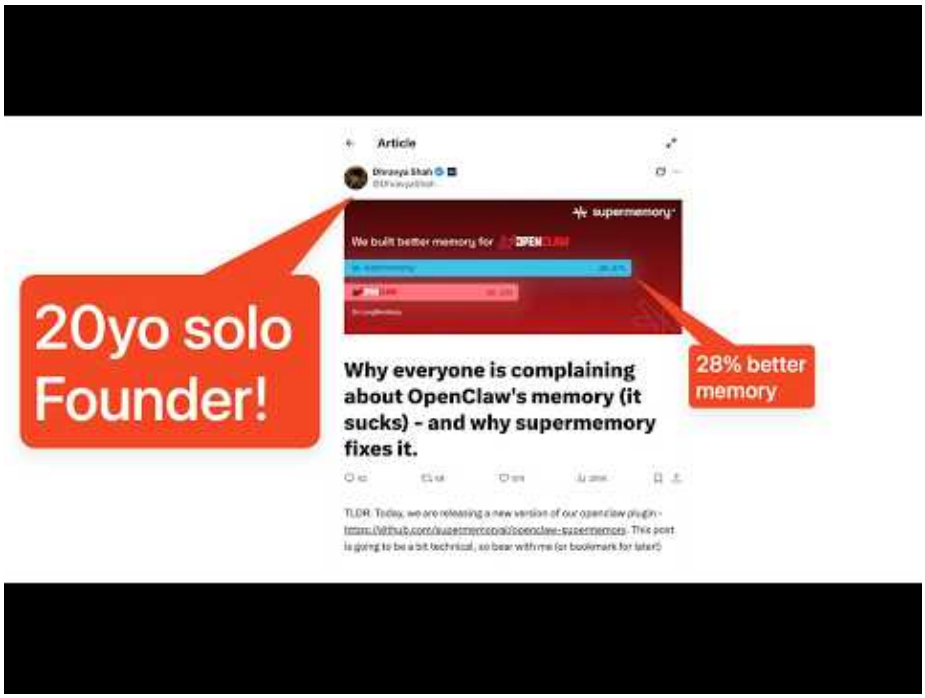
- **Latent Space — 19:24–20:35:** why user profiles beat literal retrieval. Good explanation of why an agent needs a tiny always-on profile plus recent episodes to answer questions like what monitor fits you, even if you never explicitly talked about monitors [54]



OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory (19:24)

- **Latent Space — 22:25–23:42:** hybrid memory mode for OpenClaw. Memories surface first, RAG fills the gap when memory misses, and the system extracts that information in the background for future turns [55]

[54] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[55] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory



OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory (22:25)

- **NVIDIA on Latent Space — 1:08:21–1:09:41:** why coding agents keep beating general agents. The argument is straightforward: the terminal gives agents access to compilers, tests, and every installed tool, so the feedback loop is tighter than pure chat [56]

[56] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"

NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light" (68:21)

PROJECTS & REPOS

- acpx — bridge layer that lets Codex call OpenClaw over ACP and OpenClaw call Codex back. Steinberger is already using it for private agent-to-agent discussion with an approval gate before posting to Discord [57][58][59][60]
- **Super Memory** — open-source context infrastructure for stateful agents. Shah says the project reached **100k users** on about **$5/month** of Cloudflare spend in its early consumer phase and hit **10k GitHub stars in a few weeks** after open source [61][62]
- **Memory Benchmark** — open-source eval harness for memory systems across providers, benchmarks, and judges, with metrics for quality, latency, cost, top-K recall, and NDCG [63]
- **Karpathy's lightweight GitHub coordination pattern** — use Discussions for agent-written run summaries and PRs for exact commits you

[57] post by @steipete
[58] post by @steipete
[59] post by @steipete
[60] post by @steipete
[61] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[62] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[63] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory

might adopt without merging [64][65][66]

Editorial take: the edge is shifting from choosing one best model to building the system around it — routing, memory, terminal access, and permission boundaries [67][68][69][70]

Sources

1. post by @dtcb
2. post by @swyx
3. NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"
4. post by @jediahkatz
5. post by @LangChain
6. post by @LangChain
7. OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
8. post by @romainhuet
9. post by @steipete
10. post by @steipete
11. post by @steipete
12. post by @rileybrown
13. post by @kentcdodds
14. post by @kentcdodds
15. post by @karpathy
16. post by @theo

[64] post by @karpathy
[65] post by @karpathy
[66] post by @karpathy
[67] post by @swyx
[68] OpenClaw's Memory Sucks and the fix is simple — Dhravya Shah, Supermemory
[69] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"
[70] NVIDIA's AI Engineers: Brev, Dynamo and Agent Inference at Planetary Scale and "Speed of Light"