Ticket Queues Become the New Agent UI

Coding Agents Alpha Tracker

2026-05-03

Ticket Queues Become the New Agent UI

By Coding Agents Alpha Tracker • May 3, 2026

The strongest signal today is architectural: top practitioners are moving from chatty sessions to ticket queues, narrow agent roles, and repo-native SOPs. This brief covers the best practical setups from Symphony, OpenClaw, Cursor SDK, Codex, and local DeepSeek workflows.

TOP SIGNAL

OpenAI's Symphony / "Symfony" is the clearest sign that coding agents are moving from **session management** to **deliverable management**: a background scheduler polls tickets, opens an isolated workspace per ticket, updates ticket state, and raises a PR when work reaches `Merging` [1]. Jason Zhou says Symphony plus a good codebase harness improved his coding-agent outcomes by **5x**, and the same pattern shows up elsewhere today: Ross Mike gets better results with one orchestrator agent delegating narrow jobs, while swyx frames the role shift as **plan and review** rather than hand-writing every implementation [2, 3, 4]. The edge is increasingly in the state machine around the model—repo-local SOPs, narrow scopes, and explicit review gates—not in babysitting one more chat window [1, 5].

TRY THIS

- **Set up a ticket-native loop in an afternoon.** Jason Zhou's setup is explicit: clone Symphony; if you need custom tooling or language support, point another coding agent at `spec.md`; generate a repo-local `workflow.md`; create a Linear project and save a personal API key with `linear api-token save`; define `To Do`, `In Progress`, `Human Review`, and `Merging`; then run `symphony --workflow path/to/workflow.md --daemon` [1]. The `workflow.md` frontmatter controls ticket filters, polling, hooks, parallelism, and agent settings, while the markdown body is the SOP the agent follows every turn [1]. Add Playwright CLI, a boot

skill, and indexed docs if you want autonomous verification instead of partially autonomous implementation [1, 2].

- **Keep the stack narrow; talk to one orchestrator.** Ross Mike says OpenClaw worked best when one main agent held the full context and delegated bounded jobs to sub-agents; he does **not** talk to the sub-agents directly [3]. He pairs that with a narrow skill surface—few goals, few connectors, domain-specific skills—because broad "do everything" agents with 15-30 skills/connectors made "none of it work" [3]. Good default: one main agent, specialized subs, and human review only at the last consequential step [1].
- **Convert good runs into skills; stop carrying junk context.** Riley Brown and Ross Mike describe the same loop: get one output you actually like, then reverse-engineer that run into a reusable skill with exact structure, examples, and domain rules; that was the difference between garbage reports and one-shot usable output in their demos [3]. Keep only necessary context too—don't tell the model what it can already infer from the repo, and don't expect slangy prompts to produce precise work [3].

  "The value of good instructions has never been higher." [6]

Tibo also calls `/goal` one of Codex's most consequential releases so far, which fits the same pattern: instruction quality is compounding, not getting commoditized [6].

- **If you run local frontier-ish models, compact aggressively.** Salvatore Sanfilippo's `ds4.c` demo on a 128GB M3 Max shows the hidden tax isn't just model size: 32K context adds ~1GB RAM, 250K adds ~7GB, and big tool outputs plus huge system prompts crush real-world latency; a 5K-token tool response after an 11K-token system prompt took 86 seconds to reprocess [7]. His fix is straightforward: after long runs, compact the conversation into a short summary instead of dragging the full transcript forward [7].

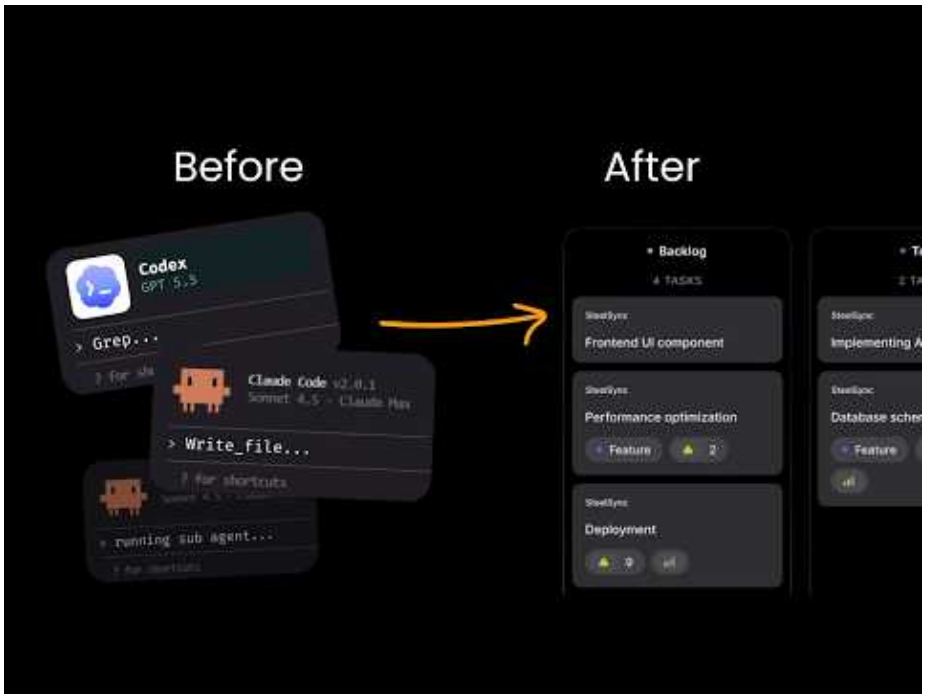
WHAT SHIPPED

- **OpenAI Symphony / "Symfony"** — open-source ticket-driven coding-agent orchestrator. Core pieces: a background scheduler plus repo-local `workflow.md` that acts as config + SOP; default flow polls Linear every 30s, creates one workspace per ticket, and auto-PRs on `Merging` [1]. Jason Zhou's field report: `Playwright CLI` + boot skill + `WORKFLOW.md` + good harness = **5x** better outcomes [2].
- **Cursor SDK** — official SDK for building agents with the same runtime, harness, and models as Cursor. Use cases: CI/CD jobs, end-to-end automations, and embedding agents in products. Announcement [8, 9]. Composer 2 is 50% off in the SDK this weekend [8].

- **Petdex** — RaillyHugo's public gallery for discovering, sharing, and installing Codex pets "with one curl"; Greg Brockman amplified it and submissions are open. Link [10, 11].
- **`ds4.c` / `ds4 server`** — Salvatore Sanfilippo's mostly GPT-5.5-built local inference engine for DeepSeek v4 Flash 270B in 2-bit GGUF (~81GB), with an OpenAI-compatible server for coding agents like OpenCode. Current state: working locally on a MacBook Pro M3 Max 128GB, not on GitHub yet [7].
- **Codex vs Anthropic: the practitioner split is getting sharper.** Riley Brown and Ross Mike argue Codex is winning the "super app" lane because coding + knowledge work live behind one interface toggle, while Anthropic's Cowork / Code / Dispatch / Remote stack still feels fragmented [3]. Their parallel claim is useful: a great coding model is increasingly a great general-purpose knowledge-work model because everything reduces to files, tools, and GUI wrappers [3].
- **Claude Code for web, real build** — Simon Willison shipped iNaturalist syndication from his phone and wired it into homepage, archives, and search. Good proof that browser-native coding agents are now viable for real feature work, not just edits. PR [12].

GO DEEPER

- **1:45-3:38 — Symphony's mental model.** Jason Zhou explains the scheduler + `workflow.md` split: frontmatter handles polling, workspace hooks, and agent settings; the markdown body holds the SOP. This is the shortest clean explanation of why the ticket tracker becomes the state machine [1].



New AI coding paradiagm - OpenAI Symphony (2:31)

- **23:53-25:04 — A proactive agent loop that actually saves time.** Ross Mike walks through an OpenClaw workflow that vets sponsor emails, researches companies, sends the first pricing email, tracks negotiations in Notion, and hands off only the finals. Even if you don't do sponsorships, the reusable pattern is heartbeat + database + human-at-the-end [3].
- **15:09-20:58 — Local frontier model, real latency pain.** Salvatore's ds4/OpenCode demo is worth watching because it shows the practical bottleneck nobody advertises: tool output size and system-prompt bulk, not just raw tok/s. You'll leave with a much better feel for when local agents are viable and when compaction is mandatory [7].
- **25:26-33:24 — Steering vs queueing in Codex.** Riley Brown's beginner guide is one of the better current overviews of projects, plugins, custom skills, and automations; skip straight to the steering/queueing section if you already know the UI. Tutorial [13].
- **Study this PR, not just the screenshot.** Simon Willison's PR #668 is a clean example of shipping a real feature with Claude Code for web, while the Codebase Context Specification is still a useful artifact if you want a shared language for persistent context layout [12, 5].

Editorial take: the durable edge is moving out of the model picker and into ticket queues, repo-native SOPs, narrow agent roles, and ruthless context hygiene [1,

3, 5].

Sources

1. New AI coding paradiagm - OpenAI Symphony
2. X post by @jasonzhou1993
3. Every AI Company Is Building the Same App (Here's Why)
4. X post by @aiDotEngineer
5. Agentic Engineering isn't a Vibe Coding
6. X post by @thsottiaux
7. Dimostrazione pratica di DeepSeek v4 Flash in locale con 128GB di RAM
8. X post by @cursor_ai
9. X post by @cursor_ai
10. X post by @RaillyHugo
11. X post by @gdb
12. Sightings
13. X post by @rileybrown