Harness Design, Review Gates, and Always-On CI Agents

Coding Agents Alpha Tracker

2026-05-06

Harness Design, Review Gates, and Always-On CI Agents

By Coding Agents Alpha Tracker • May 6, 2026

Practitioners converged on a durable lesson: the edge in coding agents is shifting from model choice to harness design—shared specs, context trimming, portable interfaces, and explicit review gates. Also in this brief: Cursor's CI autofix, fs-safe, CodexBar 0.24, and the clips worth stealing from.

TOP SIGNAL

The durable edge is moving from **model picking** to **harness design**. PI maintainers say tool-call and system-prompt work can move a model's score by ~30-40% [1], LangChain is pushing ACP so the same agent can survive CLI/TUI/IDE changes [2], and Harrison Chase argues that the state wrapped around the model—not the model itself—is now the bigger lock-in risk [3].

Simon Willison's day-to-day workflow is the operational version of that thesis: agents can be black-box reliable for routine tasks, but humans still own security-adjacent review and higher-order judgment [4].

> "the model is yours to pick. the interface is yours to pick. the harness shouldn't be the thing that locks you in." [2]

TRY THIS

- **Black-box the boring path; hand-review the risky path.** Give the agent a bounded task like: *"build a JSON API endpoint that runs a SQL query and outputs the results as JSON; add automated tests and documentation."* Simon Willison says that class of work is now reliable enough to treat as a semi-black box—but he still manually reviews anything security-adjacent [4].

- **Run parallel spikes, not parallel production merges.** Simon's current workflow: fire off a Claude Code web task for one spike, run a second spike in Codex, keep doing other work, then come back and review the prototypes. He says this only became practical once reliability improved enough to reduce review overhead [4].
- **Use one shared spec, many fresh subagents, and aggressive context trimming.** Max's PI setup starts each subagent from a fresh session with a common-ground plan/spec and a manager session id; the main session surfaces blockers, and `Reduce` strips tool calls/thinking so the active context keeps only user + assistant finals [1].
- **If the code is wrong, rewrite the spec—not just the prompt.** Salvatore Sanfilippo's Redis-arrays loop: write the spec in Markdown, improve the spec with GPT, generate an implementation, go back to the spec if tests are unsatisfying, then do a manual line-by-line review of the core code [5].

WHAT SHIPPED

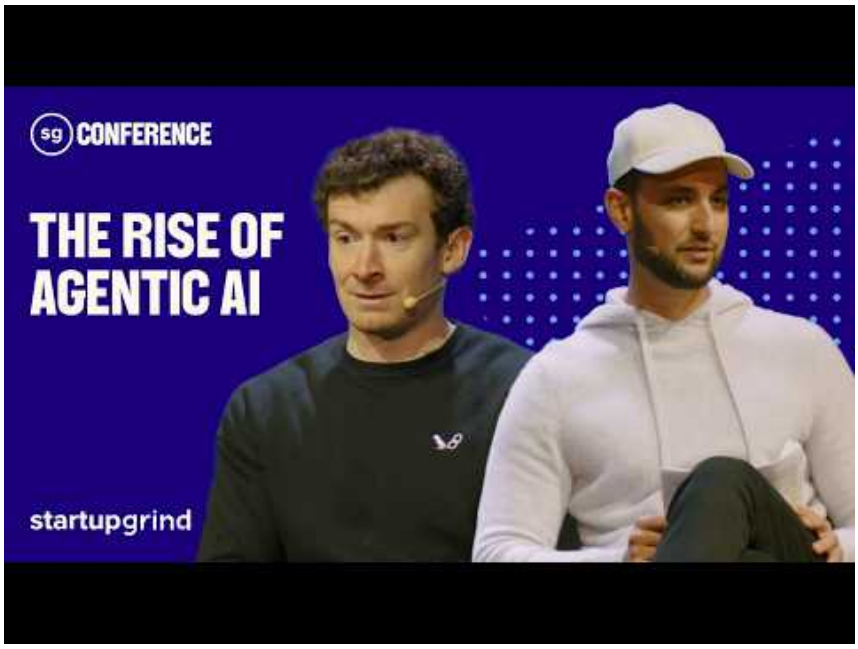
- **Cursor CI autofix** — Cursor now offers always-on agents that monitor GitHub, investigate CI root causes, and open PRs with fixes. Setup template: cursor.com/marketplace/automations/ci-autofix [6, 7]
- **openclaw/fs-safe** — Peter Steinberger shipped a reusable filesystem safety primitive extracted from OpenClaw. The guidance is practical: if your Node app accepts paths from agents, plugins, uploads, configs, or users, use a **root handle** instead of treating string normalization as the security boundary. Docs: fs-safe.io [8]
- **CodexBar 0.24** — New Windsurf, Codebuff, and DeepSeek providers; Copilot multi-account switching; opt-in local storage breakdowns; fixes for hung Codex RPC and redraw battery drain. Release: github.com/steipete/CodexBar/releases/tag/v0.24 [9]
- **Deep Agents + ACP** — LangChain says `deepagents-acp` can serve any agent and `deepagents-cli --acp` exposes the same harness over ACP, with working frontends like toad and JetBrains IDE integration via this blog post [2, 10].
- **Current model/tool preference snapshot** — Simon Willison says Codex has replaced Claude Code for most of his daily use because the latest version is "outstanding" and Claude Code pricing is a trust issue for him [4]. His current favorite local model runs in about **20GB RAM** on a laptop and feels roughly like frontier capability from **6-12 months ago** [4]. Harrison Chase adds that **GLM5** feels close enough to Sonnet/Opus for a lot of prototyping that product taste now matters more than squeezing out the absolute best model [3].

GO DEEPER

- **10:26-11:55 — Simon Willison on vibe coding vs agentic engineering.** Best short reset on where agents belong in real software work: personal tools are one thing; production systems touching other people's data need a stricter bar. Watch: YouTube [4]

High Leverage - Ep. #9, The AI Coding Paradigm Shift with Simon Willison (10:25)

- **12:35-13:50 — PI's shared-spec + blocker handoff pattern.** Best concrete demo in today's pile of a main session steering fresh-context subagents: every worker reads the same plan/spec, and the main session surfaces blockers so the human can drop straight into the right subagent. Watch: YouTube [1]
- **8:16-9:34 — Harrison Chase on memory as the real lock-in.** If your stack is quietly starting to depend on provider-managed memory, this is the clip to watch before that hardens into architecture. Watch: YouTube



[3] *The Rise of Agentic AI with Harrison Chase (LangChain) + Rajeev Dham (Sapphire Ventures) (8:15)*

- **Study these artifacts, not just the takes.** Cursor's CI autofix template is the most copyable always-on GitHub agent setup from today [7]. fs-safe.io is the cleaner reference if any part of your stack lets agents touch the filesystem through generated or user-supplied paths [8].

Editorial take: model choice still matters, but today's durable edge is harness design—portable interfaces, owned memory, trimmed context, and explicit review gates. [2, 3, 1, 4]

Sources

1. Pi.dev explained by its creators | AI Agents Under the Hood
2. X post by @masondrxy
3. The Rise of Agentic AI with Harrison Chase (LangChain) + Rajeev Dham (Sapphire Ventures)
4. High Leverage - Ep. #9, The AI Coding Paradigm Shift with Simon Willison
5. Redis ora ha gli array. Con una sorpresa.
6. X post by @cursor_ai
7. X post by @cursor_ai
8. X post by @steipete
9. X post by @steipete

10. X post by @LangChain