Harness engineering wins today; Sonnet 4.6 rolls into Claude Code, Cursor, and OpenClaw

Coding Agents Alpha Tracker

2026-02-18

Harness engineering wins today; Sonnet 4.6 rolls into Claude Code, Cursor, and OpenClaw

By Coding Agents Alpha Tracker • February 18, 2026

LangChain's deepagents-cli jump on Terminal Bench shows how much mileage you can still get from harness engineering (verification hooks, context injection, loop detection) without switching models. Meanwhile, Sonnet 4.6 lands across Claude Code/Cursor/OpenClaw alongside Cursor's push into long-running agents + plugin marketplaces, plus fresh practitioner workflows for plan mode, subagents, and sustainable pacing.

TOP SIGNAL

LangChain showed a *high-signal* reminder: **you can get big agent gains without swapping models**—they moved `deepagents-cli` **+13.7 points** on Terminal Bench 2.0 (**52.8 → 66.5**) by iterating on the **harness** (context injection, forced self-verification, loop detection, reasoning budget strategy) while keeping the model fixed (**gpt-5.2-codex**) [1]. If you're not running a harness iteration loop, you're leaving performance on the table.

TOOLS & MODELS

- **Claude Sonnet 4.6 (rollout + platform support)**
  - Anthropic: Sonnet 4.6 is a "full upgrade across coding, computer use, long-context reasoning, agent planning…" and includes a **1M token context window (beta)** [2].
  - **Claude Code**: Sonnet 4.6 is now live, **cheaper than Opus 4.6**, "nears Opus-level intelligence," and is now the **default for Pro and Team** [3].
  - **Cursor**: Sonnet 4.6 is now available; Cursor says it's a **notable improvement** over Sonnet 4.5 on longer tasks (but below Opus 4.6

for intelligence) [4].
  - **OpenClaw**: new beta (v2026.2.17) adds **Sonnet 4.6 + 1M context** support (plus "a truckload of fixes") [5].
- **Claude web search/fetch: dynamic filtering via code execution**
  - Claude's web search + fetch tools can now **write and execute code to filter results before they hit the context window** [6].
  - When enabled, Sonnet 4.6 saw **+13% accuracy on BrowseComp** while using **32% fewer input tokens** [6].
  - Anthropic also says **code execution, web fetch, memory, programmatic tool calling, tool search, and tool use examples** are now GA [7].
- **Cursor: "bigger tasks" infrastructure**
  - **Long-running agents**: Cursor says a new harness enables agents to complete "much larger tasks," now available at **http://cursor.com/agents** (Ultra/Teams/Enterprise) [8]. CEO Michael Truell says PRs from these agents have **higher merge rates** while being "~an order of magnitude more ambitious" [9].
  - **Plugins marketplace**: Cursor launched an agent plugin marketplace (blog: https://cursor.com/blog/marketplace) [10], positioning it as easy access to knowledge bases + environments + tooling (examples mentioned: Datadog, Notion runbooks, AWS deploys, Figma mocks) [11].
  - **Async subagents** + subagents spawning subagents were called out as the "biggest W" in a Cursor update by @emilheap (reshared by @jediahkatz) [12].
- **Codex (safety routing + transparency improvements)**
  - OpenAI: some requests may be routed from **GPT-5.3-Codex → GPT-5.2** when systems detect elevated cyber misuse risk; they note legitimate work can be misclassified and users can apply at **http://chatgpt.com/cyber** [13].
  - Follow-up: OpenAI calibrated classifiers/policies to reduce flagged traffic (predicting it will be **well under 1%**), shipped **per-turn downgrade notifications** in **CLI v0.102.0**, fixed Trusted Access regain issues, and published docs: https://developers.openai.com/codex/concepts/cyber-safety [14].
- **Model selection chatter (practitioner comparisons)**
  - Theo's current flow: **Opus via Claude Code in terminal**, **Codex 5.3 via Codex CLI + desktop app** [15]. He frames Codex as a "measure twice cut once" model vs Opus moving faster but missing details [15].
  - DHH: "K2.5 is now my main driver. Opus just backup." [16] A colleague raced them on a bug: **K2.5 fixed it in 21s**; Claude took ~1 min to plan + ~2 min to execute, arriving at the same fix [16].

WORKFLOWS & TRICKS

- **Harness iteration loop (LangChain's deepagents-cli playbook)**
  - **Force a build→verify→fix loop**: plan/discover, build (write tests), verify against the *task spec* (not your code), fix; LangChain added a `PreCompletionChecklistMiddleware` to push agents into verification before exiting [1].
  - **Inject environment context up front**: their `LocalContextMiddleware` maps directory structure + discovers tools; they also inject time budget warnings and testability guidance to reduce "slop buildup" [1].
  - **Detect doom loops**: `LoopDetectionMiddleware` tracks edit counts per file and injects "reconsider your approach" context after N edits [1].
  - **Reasoning budget heuristic**: their baseline is an xhigh-high-xhigh "**reasoning sandwich**" (spend more on planning + verification; keep build cheaper to avoid timeouts) [1].
  - **Use traces as feedback**: they turn trace analysis into an "Agent Skill" that fetches traces, spawns parallel error analysis agents, and synthesizes harness changes [1].
- **Production Claude Code habits from Anthropic (Boris Cherny)**
  - **Keep `.claude.md` small and resettable**: Boris recommends doing the *minimum* to keep the model on track; if it gets bloated, delete it and rebuild incrementally as model capabilities change [17].
  - **Plan mode as a parallel planning engine**: Boris starts ~80% of sessions in plan mode, often running multiple plans in parallel across terminal tabs/desktop tabs, then executing once the plan is good [17].
  - **Scale "research/debug" with subagents**: for harder tasks, he calibrates the number of subagents (3/5/10) to research in parallel and converge on solutions [17].
  - **Swarm execution pattern (spec → tickets → agents)**: plugins were "entirely built by a swarm" over a weekend with minimal human intervention—an engineer gave a spec and told Claude Code to use an Asana board; it created tickets and spawned agents that picked up tasks [17].
- **Agent automation beyond coding (OpenClaw, Matthew Berman's setup)**
  - **Scheduled agent ops**: Berman describes using cron jobs for overnight workflows (security review, log ingestion, morning brief), plus a central cron log DB so the agent can reference logs and fix failures [18].
  - **Nightly AI security review prompt**: run at 3:30am, analyze from offense/defense/privacy/operational realism, deliver numbered findings to Telegram, alert on critical findings, and allow "deeper dives by recommendation number" [18].

  - **Defense-in-depth for personal agents**: deterministic sanitization/redaction to defend against prompt injection; restrict permissions (no write access to email/calendar), redact secrets (including in Telegram), and require explicit approval before public actions [18].
  - **Automated backups (copyable prompt)**: auto-discover SQLite DBs, encrypt + archive to Google Drive with retention + restore scripts; hourly Git auto-sync with alerts on failure [18].
- **Observability & evals: stop guessing what your agent did (LangSmith)**
  - LangSmith's primitives: **runs** (single execution step), **traces** (ordered runs), **threads** (multi-turn) [19].
  - Key operational point: with agents, "the source of truth is the traces," and production is where you discover failure modes and what to test offline [19].
- **Anti-burnout guardrail (Salvatore Sanfilippo / Redis creator)**
  - He describes "token fatigue" from async/parallel agent work: anxiety about overnight runs + context switching + chronic fatigue, even if you're shipping more [20].
  - His mitigation: **work one project at a time**, stay close to the code while the model works, alternate "prompt/idea phases" with hands-on coding, and reserve overnight runs for scoped bug hunts/optimizations [20].

PEOPLE TO WATCH

- **Boris Cherny (Anthropic / Claude Code)** — high-quality production detail on plan mode, `.claude.md` discipline, and swarm-style feature building [17].
- **LangChain DeepAgents team** — concrete harness engineering + trace-driven improvement loops with measurable benchmark movement [1].
- **Michael Truell (Cursor CEO)** — shipping the "bigger tasks" stack: long-running agents + plugins, plus merge-rate signal from PRs [8, 9, 11].
- **Alexander Embiricos (OpenAI Codex)** — unusually clear comms on model routing, misclassification, and upcoming UX transparency [13, 14].
- **Salvatore Sanfilippo** — the best articulation today of the *human cost* of vibe-coding-at-scale, plus a practical counter-pattern [20].

WATCH & LISTEN

1) "How to keep flow with coding agents" (Redis creator) — *Fatica da programmazione automatica* (~6:45–10:55)

Hook: a concrete hybrid workflow (agent does the grind; you stay close to the code; avoid context switching) aimed at eliminating "alienation" and burnout.

Fatica da programmazione automatica (6:45)

2) "Why Codex wins on big codebases (and why Claude MD matters)" — Theo: *A realistic comparison of Opus and Codex* (~40:30–42:31)

Hook: Theo's clearest explanation of *why* Codex tends to match existing repo patterns while Opus may patch locally and create inconsistency—plus the practical implication: you'll need steering docs more often with Claude Code in large repos.

A realistic comparison of Opus and Codex (40:30)

3) "Swarm pattern: spec → Asana tickets → agents" — Boris Cherny: *How We Built Claude Code* (~22:06–23:19)

Hook: a minimal recipe for running a weekend-long feature build with multiple agents and minimal human intervention.

Boris Cherny: How We Built Claude Code (22:06)

PROJECTS & REPOS

- **LangChain Deep Agents (open source)**
  - Repos: Python https://github.com/langchain-ai/deepagents and JavaScript https://github.com/langchain-ai/deepagentsjs [1].
  - `deepagents-cli` harness improvements write-up + CLI link: https://github.com/langchain-ai/deepagents/tree/main/libs/cli [1].
  - Public traces dataset: https://smith.langchain.com/public/29393299-8f31-48bb-a949-5a1f5968a744/d?tab=2&ref=blog.langchain.com [1].
- **OpenClaw beta v2026.2.17 (Sonnet 4.6 + 1M context support)**
  - Release: https://github.com/openclaw/openclaw/releases/tag/v2026.2.17 [5].
  - Project velocity signal: "70 Clatributors!" [21].
- **Rodney (browser automation CLI designed for coding agents)**
  - Simon Willison announced a new release with contributions from five people: https://simonwillison.net/2026/Feb/17/rodney/ [22].

— **Editorial take:** The differentiator isn't "which model," it's **harness + infrastructure**: verification loops, trace feedback, subagent orchestration, and real tool access are what turn agent output into something you can ship and trust [1, 23].

Sources

1. Improving Deep Agents with harness engineering
2. X post by @claudeai
3. X post by @bcherny
4. X post by @cursor_ai
5. X post by @steipete
6. X post by @alexalbert___
7. X post by @alexalbert___
8. X post by @cursor_ai
9. X post by @mntruell
10. X post by @cursor_ai
11. X post by @mntruell
12. X post by @emilheap
13. X post by @embirico
14. X post by @embirico
15. A realistic comparison of Opus and Codex
16. X post by @dhh
17. Boris Cherny: How We Built Claude Code
18. 21 INSANE Use Cases For OpenClaw…
19. Observability and Evals for AI Agents: A Simple Breakdown
20. Fatica da programmazione automatica
21. X post by @steipete
22. X post by @simonw
23. X post by @gdb