Sandboxes, Self-Summarization, and TDD Loops Tighten the Coding-Agent Stack

Coding Agents Alpha Tracker

2026-03-18

Sandboxes, Self-Summarization, and TDD Loops Tighten the Coding-Agent Stack

By Coding Agents Alpha Tracker • March 18, 2026

The useful signal today was harness quality, not just model churn. New sandboxed execution layers, better long-horizon context handling, and concrete test/manual-test habits from experienced practitioners point to what actually improves coding-agent reliability.

TOP SIGNAL

Today's clearest pattern: **the harness is becoming the product**. LangChain launched **LangSmith Sandboxes** and **Open SWE** around isolated execution, persistent sandboxes, curated toolsets, and workflow-native triggers, while Cursor said RL-based **self-summarization** cut compaction error by **50%** on coding tasks that require **hundreds of actions** [1, 2, 3].

The practical takeaway is straightforward: safer execution plus better context compression is where reliability is improving right now—not just raw model swaps [1, 2, 3].

TOOLS & MODELS

- **GPT-5.4 mini** — now available in **ChatGPT, Codex, and the API**. OpenAI says it is optimized for **coding, computer use, multimodal understanding, and subagents**, and is **2x faster** than GPT-5 mini [4].
- **Cursor Composer** — now trained to **self-summarize via RL** instead of a prompt. Cursor says this cuts compaction error by **50%** and improves success on long coding tasks with **hundreds of actions** [3].
- **LangSmith Sandboxes** — now in **private preview**. Key pieces: **MicroVM isolation**, an **auth proxy** so secrets never touch the runtime,

persistent long-running sessions, state carryover, tunnels, and direct integrations with **Deep Agents** and **Open SWE** [1].

- **Open SWE** — new open-source framework for internal coding agents built on **Deep Agents** and **LangGraph**. It packages patterns LangChain says it observed across Stripe, Ramp, and Coinbase: isolated sandboxes, curated tools, Slack/Linear/GitHub invocation, `AGENTS.md` startup context, subagents, and middleware safety nets [2].
- **Operator comparison: Codex vs. Claude Code** — Theo said **GPT-5.4** in Codex/T3 Code quickly diagnosed mixed TanStack versions and fixed a Vite+ migration, while his Claude Code run sat for **15+ minutes** without changing code [5].

WORKFLOWS & TRICKS

- **Simon Willison's low-drama loop**: start every session by telling the agent how to run the tests, then add **"use red-green TDD."** After tests pass, make it boot the server and hit the API with `curl`, because green tests still miss runtime failures. If you want an artifact, **Showboat** turns the manual test into a markdown log with commands and outputs [6].

  "Tests are no longer even remotely optional." [6]

- **Conformance-first implementation**: have the agent build a test suite from multiple working implementations, then code against that suite. Simon used behavior from **Go, Node.js, Django, and Starlette** to generate multipart upload tests first, then implemented the feature in Datasette [6].
- **Keep `AGENTS.md` lean**: Open SWE injects a root `AGENTS.md` into the system prompt for conventions, testing rules, and team patterns. Theo's live Vite+ run shows the failure mode: bloated agent files packed with scaffold commands and irrelevant noise hurt the model; move bulky details to docs or skills instead [2, 5].
- **Async bug-fix fanout**: Felix Rieseberg's internal Cowork loop is copyable:
  1. Point the agent at the crash dashboard.
  2. Have it separate fixable bugs from OS/kernel noise.
  3. Write **one markdown prompt per fixable bug**.
  4. Launch a remote Claude Code task for each prompt and let them run while you're in meetings [7, 8].
- **Sandbox rule of thumb**: isolate first, then allow full permissions inside the boundary. Open SWE and LangSmith both follow this pattern, and LangSmith adds proxy-based access so credentials stay off the sandbox entirely [2, 1].

PEOPLE TO WATCH

- **Simon Willison** — shared concrete operator playbooks today: Pragmatic Summit highlights plus new chapters on **how coding agents work** and **subagents**. Useful because they include reusable prompts, TDD/manual-test loops, and context tactics [6, 9, 10].
- **Felix Rieseberg** — useful voice on VM-based agent harnesses. The Cowork interviews connect VM isolation, markdown skills, Chrome integration, and internal bug-triage orchestration in one coherent workflow model [7].
- **Theo** — worth watching when you want an unpolished tool comparison instead of a vendor benchmark. Today he showed both a practical **Codex/GPT-5.4** win and a sharp critique of noisy `AGENTS.md` files [5].
- **Logan Kilpatrick** — strong big-company signal: better models and harnesses let him get back into shipping production code at Google, but humans still own review, prioritization, and the "what should we build?" decision [11].
- **DHH** — notable because he was publicly skeptical for a long time. His shift from using AI as a better search/pairing tool to daily agent use is meaningful, and his framing is useful: agents amplify output without reducing the programmer to a project manager [12].

WATCH & LISTEN

- **2:39-3:37 — LangSmith Sandboxes as a tool**: a short demo of the pattern. A deployed agent spins up a sandbox, generates HTML, renders it with a headless browser, and sends back a screenshot [13].

Introducing: LangSmith Sandboxes (Now in Private Preview) (2:38)

- **15:35-17:25 — Felix's async bug-fix loop**: Cowork reads a crash dashboard, filters fixable issues, writes per-bug markdown prompts, and fans out remote Claude Code runs [8].

Why Anthropic Thinks AI Should Have Its Own Computer — Felix Rieseberg of Claude Cowork/Code (15:34)

- **44:29-46:40 — DHH on the flip**: worth the segment for the mental-model update. He explains why late-2025 agents stopped feeling like bad autocomplete and started feeling like parallel cognitive leverage [12].

  “It is more like I’ve grown 18 arms and seven more brains.” [12]

PROJECTS & REPOS

- **Open SWE** — new open-source foundation for internal coding agents. The adoption signal here is architecture: LangChain says it packages the same core patterns seen in Stripe’s Minions, Ramp’s Inspect, and Coinbase’s Cloudbot [2].
- **pi-autoresearch** — worth watching because it was used in Shopify’s Liquid optimization run. That effort produced **93 commits** from around **120 automated experiments** and landed a **53%** parse+render improvement on Liquid [6].
- **Shopify/liquid PR #2056** — a strong proof artifact for autonomous optimization: the PR headline claims **53% faster parse+render** and **61% fewer allocations** after agent-driven micro-optimization work [6].
- **multipart-form-data-conformance** — small repo, clear pattern. It shows how to turn multiple existing implementations into a conformance suite the agent can target for a new implementation [6].

Editorial take: the durable edge right now is not one model release; it's the harness—sandboxed execution, lean context, and ruthless verification. [1, 3, 6]

Sources

1. Introducing LangSmith Sandboxes: Secure Code Execution for Agents
2. Open SWE: An Open-Source Framework for Internal Coding Agents
3. X post by @cursor_ai
4. X post by @OpenAI
5. They fixed Vite
6. Fireside chat about agentic engineering at the Pragmatic Summit
7. Why Anthropic Thinks AI Should Have Its Own Computer — Felix Rieseberg of Claude Cowork & Claude Code Desktop
8. Why Anthropic Thinks AI Should Have Its Own Computer — Felix Rieseberg of Claude Cowork/Code
9. X post by @simonw
10. X post by @simonw
11. Navigating the Future of AI with Logan Kilpatrick | Google Deepmind
12. $100M+ Advice That'll Piss Off Every Business Guru (ft. DHH)
13. Introducing: LangSmith Sandboxes (Now in Private Preview)