Verifiable Loops Win: Codex Testing, GLM 5.2, and Claude Artifacts

Coding Agents Alpha Tracker

2026-06-22

Verifiable Loops Win: Codex Testing, GLM 5.2, and Claude Artifacts

By Coding Agents Alpha Tracker • June 22, 2026

The strongest signal today is that coding-agent loops only become useful when the task is verifiable. This brief covers copyable Codex workflows, Riley Brown's GLM 5.2 and Record & Replay demos, Claude Code Artifacts, and a same-day hype check on Sakana Fugu.

TOP SIGNAL

Today's clearest pattern: **agent loops only get reliable when the task is verifiable**. Romain Huet says coding is the right proving ground because long tasks can be checked with tests [1], ThePrimeagen's checklist for successful loops is defined inputs/outputs, clear success/failure, repeatability, and observability [2], and Armin Ronacher says that without that structure loops still mostly hold up for review/research rather than medium-sized implementation [3]. Tom Osman and Greg Brockman's Codex workflow is the practical template: generate canonical user stories for every feature, test them, log errors, fix them, then retest—with a human still reviewing PRs before merge [4, 5, 1].

TRY THIS

- **Run Codex as a full feature-coverage loop (Tom Osman via Greg Brockman).** Point it at an existing app and give it an explicit end state:

```
/goal go over every single feature in this app
create a user story with expected behaviour based
on the code keep a single canonical spreadsheet
tracking the features status  - when done switch loop
to testing every user story and documenting all
```

```
errors  - when done fix every logistical error or ux
error - test every user behaviour again post fix
```
[4]

Greg Brockman highlighted this as Codex for testing every feature, and Tom says it can work through hundreds of user stories automatically [5, 4]. It also fits ThePrimeagen's loop criteria: defined outcome, clear success/failure, repeatable, observable [2].

- **Force a second opinion after API design (Theo).** Add this to your Codex first loop:

  ```
  When you are done designing the API, get a second
  opinion from Opus with 'claude -p'
  ```
  [6]

  Theo says this has significantly improved the code quality he gets from OpenAI models [6]. Good default whenever the agent is making architectural calls.

- **Turn a manual browser task into a reusable skill (Riley Brown).** In Codex, use the Record and Replay plugin, say `Please make a skill called [Name]`, perform the workflow on screen, stop recording, and Codex turns it into a slash command you can invoke later like `/manual tweet draft` [7]. Riley's demo shows recordings up to 30 minutes, which makes this useful for real UI chores, not just toy clicks [7].

- **Use an agent as a backlog analyst, not just a coder (Geoffrey Huntley).** Give the agent `gh cli` access and ask it to generate a markdown report of the top unresolved issues, with columns for problem description, platform, upvotes, and age, and a linked LLM summary plus proposed resolution for each row [8]. Huntley's concrete example targets the top 250 unresolved NixOS/nix issues in a file called `nixos-nix.md` [8].

WHAT SHIPPED

- **GLM 5.2 is now a serious Cursor candidate via OpenRouter.** Riley Brown's setup: in Cursor go to Settings → Models → API keys, enable custom, override the OpenAI base URL with the OpenRouter endpoint, then add `z-ai/glm-5.2` as a custom model [7]. In Riley's own tests, GLM 5.2 one-shotted a Trello-style app with DB/auth via Convex, built and ran a landing page locally, and handled Notion/Slack agent tasks comparably to Opus 4.8; he also says it feels close to GPT 5.5 / Opus 4.8 overall [7].

- **Claude Code Artifacts.** Claude Code can now generate shareable interactive mini-apps/artifacts with their own links, giving teams something concrete to review and pass around [7].

- **Codex stack openness got clearer.** Romain Huet says the Codex CLI, full harness, and server are open source on GitHub; the Codex app can also run open-source models, and he says OpenAI uses Codex across the company, including non-engineers [1].

- **Temporary deploys for AI-built apps are practical now.** Simon Willison had GPT-5.5 xhigh in Codex Desktop build cloudflare-redirect-resolver, then deployed it with `npx wrangler deploy --temporary`; Cloudflare kept the ephemeral Workers project live for 60 minutes, and Simon says the temporary deployment worked as advertised [9].
- **Sakana Fugu launched with an immediate reality check.** Sakana introduced Fugu as a full multi-agent orchestration system behind a single model API and says Fugu Ultra matches Fable and Mythos; it is available at sakana.ai/fugu [10]. Riley Brown's first design-task test did not finish before daily limits kicked in [11].

GO DEEPER

- **8:41–12:11 — Riley Brown on Record & Replay → Codex skill.** The most copyable walkthrough in today's batch: record a real browser workflow, stop capture, then call it later as a slash command [7].

AI Agents Just Changed Forever: GLM 5.2, Codex Skills, Claude & Cursor (8:40)

- **3:34–4:29 — Romain Huet on why coding is the first real agent harness.** Short clip, big idea: long-running agents improve fastest where work can be verified by tests and tools [1].

Quel est le futur de l'IA ? (3:34)

- **Repo study — Simon Willison's cloudflare-redirect-resolver and build gist.** Small, concrete, and deployed: a good example of using Codex Desktop to build a utility app and ship it to a temporary environment for real validation [9].

Editorial take: the durable edge right now is not 'more agents'—it is better harnesses: explicit goals, clear success criteria, repeatable environments, verifiable tests, and human review at merge time [4, 2, 1].

Sources

1. Quel est le futur de l'IA ?
2. X post by @ThePrimeagen
3. X post by @mitsuhiko
4. X post by @tomosman
5. X post by @gdb
6. X post by @theo
7. AI Agents Just Changed Forever: GLM 5.2, Codex Skills, Claude & Cursor
8. X post by @GeoffreyHuntley
9. Temporary Cloudflare Accounts for AI agents
10. X post by @SakanaAILabs

11. X post by @rileybrown