# Test-to-Green Loops, Architecture Reviews, and Spend Guardrails

Coding Agents Alpha Tracker

2026-06-16

## Test-to-Green Loops, Architecture Reviews, and Spend Guardrails

*By Coding Agents Alpha Tracker • June 16, 2026*

Today's strongest coding-agent pattern is practical autonomy with explicit checks. Copyable loops from Claude Code/Cursor, context and approval tricks, and new tooling for PR review, spend control, and longer-running agent sessions.

### TOP SIGNAL

- **The strongest pattern today: autonomous coding loops are getting practical when they have hard guardrails.** Jason Zhou highlighted a copy-paste "Ship PR Until Green" workflow: paste a feature spec into Claude Code or Cursor, let the agent run tests, read failures, fix them, and keep looping until the exit condition passes or an iteration cap hits [1]. Addy Osmani's interview clip and Simon Willison's `datasette-agent` release land the same lesson from the reliability side: velocity is not enough without human verification, clear quality bars, or explicit approval before write actions [2, 3].

### TRY THIS

- **Run a test-to-green loop (Jason Zhou / AI Builder Club).**

  1. Paste the feature spec into Claude Code or Cursor.
  2. Let the agent run tests.
  3. Let it read failures and patch them.
  4. Stop only when tests pass or the iteration cap is hit.

  Reported outcome from one run: a green PR after ten iterations with "no hand-holding" [1].

- **Use slash commands as context hygiene (official Antigravity CLI thread).**

  1. `/help` to see available shortcuts.
  2. `/context` when the session gets big and you want to visualize the token window.
  3. `/diff` before review or commit to inspect uncommitted changes.
  4. `/btw` for side questions so the main task stays on track.
  5. `/artifacts` to manage the implementation plan [4, 5, 6, 7, 8, 9].

- **When MCP can't write, drop to the API (Simon Willison).**

  1. Ask Claude Code to derive the exact rule you need.
  2. Verify the final expression before applying it.
  3. If the MCP cannot edit the target resource, switch the agent to the provider API.

  Simon used this to land a Cloudflare WAF rule for search URLs containing `&`: `(http.request.uri.path wildcard r"/search/*" and http.request.uri.query contains "&")` [10].

- **Add approval-gated write tools (Simon Willison / `datasette-agent`).**

  1. Give the agent a write tool that explicitly asks for approval.
  2. Keep permissions intact.
  3. Only use broad flags like `--yes` or `--unsafe` when you intentionally want faster, less constrained execution.

  `execute_write_sql` now does this, and `datasette agent chat content.db -m gpt-5.5 --unsafe` can directly modify a database via prompts like "create a notes table" or "add a note about X" [3].

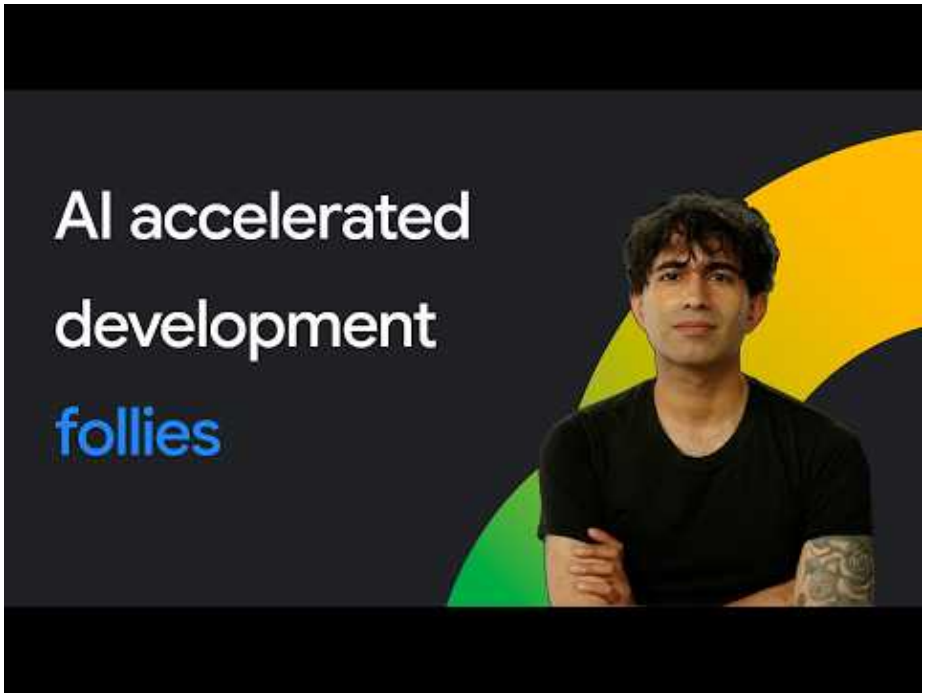
## WHAT SHIPPED

- **`Archlet` open-sourced** — Jason Zhou says AI-written code caused architecture drift against the team's mental model; `Archlet` reviews PR diffs as a graph so you can see architecture impact at a glance. Claimed internal effect: **10x** PR review speed and quality. Repo: superdesigndev/archlet [11].
- **`datasette-agent 0.3a0`** — adds `execute_write_sql` with approval gating, support for approval-requiring tools in `datasette agent chat`, new `--root`, `--yes`, `--unsafe` flags, and plain-text tool output for CLI. Notes: datasette-agent 0.3a0 [3].
- **LangSmith LLM Gateway** — LangChain says one dev on coding agents can burn **thousands of dollars a week** before anyone notices; they built this after hitting the issue internally and say it now makes spend more predictable. Reads: How we made coding agent spend predictable, Introducing LLM Gateway [12, 13].

- **Antigravity CLI slash-command pass** — official thread documenting `/help`, `/context`, `/diff`, `/btw`, `/config`//`settings`, and `/artifacts` for agentic workflows [4, 5, 6, 7, 8, 14, 9].
- **Sourcegraph Cloud** — longer Deep Search conversations now use automatic compaction to keep growing threads manageable [15].
- **Claude Agent SDK billing reversal** — the planned credit change is paused, and Theo says T3 Code users can keep using Claude Code with their existing subscriptions [16, 17].
- **Emerging project: `clawsweeper`** — Peter Steinberger says new issues on their open-source projects get checked against `VISION.md`, then the agent can create and autoreview a PR if the issue fits. Example: openclaw/gogcli#816 [18, 19].
- **Practitioner comparison: Anthropic Ultracode** — swyx says the subagent model feels like "subroutines but intelligent" and can apply beyond coding, but warns the fanout only pays off if the repo is set up for parallelization; otherwise it is "scarily good at burning tokens" [20].

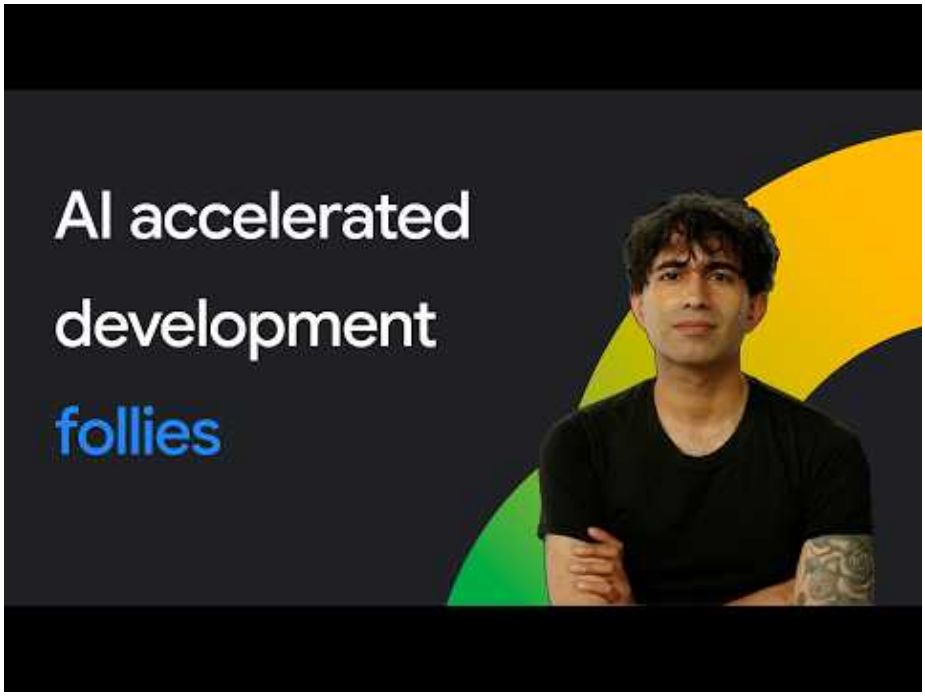
## GO DEEPER

- **0:45–1:42 — verification beats raw velocity.** Addy Osmani's guest lays out the boring but critical part of agentic engineering: tests, visual regression, or a crisp definition of "good" still have to exist if you're going to trust the output [2].



*How to build reliable software with AI agents (0:45)*

- **3:44–4:27 — the "cognitive surrender" warning.** Short clip on the failure mode where you stop thinking critically and just ship whatever the agent emits [2].




*How to build reliable software with AI agents (3:44)*

- **Repo to study: `Archlet`.** If AI-generated PRs are getting harder to reason about, this is the most concrete repo in today's sources aimed at architecture-level review rather than line-level diff skimming. Repo: github.com/superdesigndev/archlet [11].
- **Workflow artifact to study: `openclaw/gogcli#816`.** This PR is a live example of Peter Steinberger's issue -> `VISION.md` check -> agent-created -> autoreviewed loop. Start here if you want a concrete automation trace, not just a description: openclaw/gogcli#816 [18].

*Editorial take: the useful frontier is not more autonomy by itself — it's autonomous loops with explicit stop conditions, review surfaces, spend controls, and human verification.* [1, 11, 12, 2, 3]

---

**Sources**

1. X post by @aibuilderclub_

2. How to build reliable software with AI agents
3. datasette-agent 0.3a0
4. X post by @antigravity
5. X post by @antigravity
6. X post by @antigravity
7. X post by @antigravity
8. X post by @antigravity
9. X post by @antigravity
10. Cloudflare CAPTCHA on at least one ampersand
11. X post by @jasonzhou1993
12. X post by @LangChain
13. X post by @LangChain
14. X post by @antigravity
15. X post by @Sourcegraph
16. X post by @aronprins
17. X post by @theo
18. X post by @steipete
19. X post by @steipete
20. X post by @swyx