

# Fable 5 Pushes Coding Agents Into Project-Sized Work

Coding Agents Alpha Tracker

2026-06-10

## Fable 5 Pushes Coding Agents Into Project-Sized Work

*By Coding Agents Alpha Tracker • June 10, 2026*

Claude Fable 5 dominated the day, but the useful signal is practical: senior engineers are handing coding agents bigger objectives, longer autonomous runs, and cleaner human-approval seams. This brief distills the highest-leverage workflows, what actually shipped, and the clips and code worth studying.

### TOP SIGNAL

- Claude Fable 5 is the clearest step-function signal today: Andrej Karpathy says the jump shows up on long, difficult problem-solving sessions because you can hand the model more ambitious end-to-end tasks and it will run with them; Mike Krieger says it is the first model he hands whole projects to; Simon Willison used it to ship major agent features in LLM 0.32a3 and a 13.9 MB CPython WASM wheel in a day. The practical takeaway is larger task envelopes and longer autonomous runs, not blind trust: Karpathy warns against fully hands-off production use because launch safeguards are still trigger-happy. [1, 2, 3]

### TRY THIS

- **Shift from tasks to objectives in Fable.** Alex Albert's launch checklist: 1) give it bigger tasks, 2) define what done looks like and how to verify it, 3) use `/loop` and `/goal` for open-ended work, and 4) rework old `CLAUDE.mds` or skills that may anchor it to older-model behavior. Karpathy's examples of the bigger task class: explainers, visualizers, dashboards, bespoke single-use apps, 10x test suites, auto-optimization, and custom research HTML. Albert recommends `xhigh/high` by default, while Matthew

Berman found the lowest effort setting sufficient in many of his tests and called extra effort slow or overkill. [4, 1, 5]

- **Use repo + artifacts + iterative follow-ups instead of one giant prompt.** Simon Willison started with `Clone simonw/micropython-wasm` from GitHub and research how this could use a full Python as opposed to MicroPython, attached Brett Cannon's build zips, then followed up with `Try a bit more at the single-zip-stdlib problem and I want a wheel that has the whole system in it....` In his run, that produced a 13.9 MB wheel usable via `uv run --with <wheel> cpython-wasm -c 'code'`. [3]
- **Add a pause/resume seam for human approval.** Simon used Fable to push a clean pattern into llm: let tools accept `llm_tool_call`, raise `llm.PauseChain` when you need approval, and resume from `messages=history` or explicit `tool_calls_list=` so the chain keeps tool-call state. This came out of Datasette Agent's `ask_user()` requirement and is the best public example in today's source set of human-in-the-loop tool execution without brittle placeholder hacks. [3]
- **Run a pre-prod repo review, then override the question treadmill if needed.** Berman's test prompt is `review the entire code base` and ask for security, docs, logical gaps, edge cases, UX/UI workflows, and test coverage; in workflows mode he saw that fan out into 100+ sub-agents. swyx's lighter-weight version is `review my code for issues` as a pre-prod **Fable Check**, and Berman's override when Claude gets stuck in clarification/spec-confirmation loops is `Don't ask more questions. Just build.` [6, 7, 5]

## WHAT SHIPPED

- **Claude Fable 5** — Mythos-class model with added safeguards; Anthropic says its lead grows as tasks get longer and more complex. Shipping today in Claude Code/Cowork, paid Claude plans, the API at \$10/M input and \$50/M output, and Devin/Cognition at 1.4x ACUs; cyber/bio requests fall back to Opus 4.8 and Mike Krieger says 95%+ of sessions never notice. Public benchmark numbers cited across today's coverage: SW Bench Pro 80%, Frontier Code Diamond 29.3%, and computer use 85%. [1, 8, 9, 10, 11, 6]
- **Cursor integration** — Fable 5 is live in Cursor and set a new CursorBench SOTA at 72.9%, 8 points above the previous best; comparison page is [cursor.com/evals](https://cursor.com/evals). Practical warning from Jediah Katz: Cursor users should read the model docs first because Fable has unusual required data-retention policies for safety. [12, 13, 14]
- **Claude Code nested subagents** — shipped today with depth capped at 5. [15]

- **LLM 0.32a3 / Datasette Agent** — Fable pushed out `llm.PauseChain`, `llm_tool_call`, guaranteed unique `tool_call_id`, and resuming from unresolved tool calls; Simon says the release was almost entirely written by Fable. Adoption signal: he spent \$110.42 of tokens in one day on his Max plan, with one `prod_datasette_agent` session accounting for 78.2M tokens and \$99.26. [3]
- **Anecdotal side-project ceiling** — Riley Brown says Fable one-shotted a Replit-like mobile app builder from the prompt `build an app like Replit, that uses @daytonaio for sandboxing, and convex for DB`, and later extended a Lovable-like builder to web/mobile preview, editing, sandboxes, DB/auth, voice-to-text, and Swift previews in 2-5 prompts. Treat this as anecdotal, but it is a useful snapshot of what fast-moving builders tried on day one. [16, 17, 18, 19]

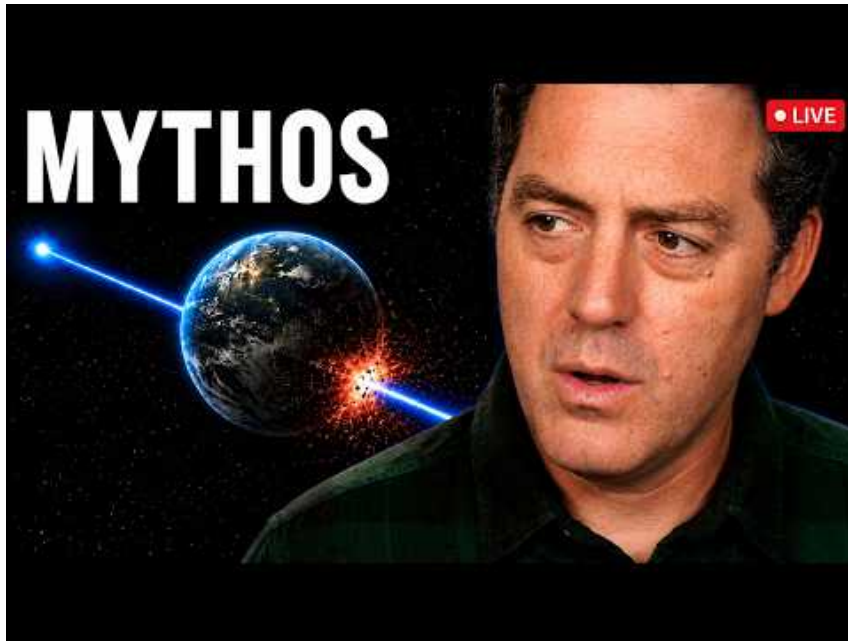
## GO DEEPER

- **Matthew Berman — 13:05-13:31: workflows fan-out.** Best short demo of what repo-wide orchestration looks like in practice: a planning agent that can spawn 100+ subagents after a few minutes of planning. Good watch before you hand a big review or migration to workflows mode.



[6]  
 MYTHOS MYTHOS MYTHOS (13:04)

- **Matthew Berman — 23:36-24:22: model routing for cost discipline.** His takeaway after a week with Fable: reserve it for the hardest long-horizon problems and route the rest to Sonnet/Haiku. [5]



*MYTHOS is LIVE!!!! (23:35)*

- **Matthew Berman — 48:51-50:12: the Just build override.** Worth watching if you keep hitting clarification, spec confirmation, and approach approval loops before the model starts coding. [5]



*MYTHOS is LIVE!!!! (48:51)*

- **Read these diffs:** Datasette Agent PR #20 and LLM PR #1482. They show a real human-approval workflow getting turned from local hacks into framework primitives. [3]
- **Tool worth adding:** AgentsView for local agent transcript and cost analysis. Simon uses it to attribute spend across projects and models; when Fable pricing was missing, he used Fable itself to reverse-engineer a custom price config. [20]

*Editorial take: the bottleneck just moved up a level — from getting a model to write code, to deciding how much work to hand it, where to force approval, and when frontier-model tokens are actually worth spending. [1, 4, 3, 5]*

---

## Sources

1. X post by @karpathy
2. X post by @mikeyk
3. Initial impressions of Claude Fable 5
4. X post by @alexalbert\_\_
5. MYTHOS is LIVE!!!!
6. MYTHOS MYTHOS MYTHOS
7. X post by @swyx
8. X post by @claudeai
9. X post by @bcherny
10. X post by @mikeyk
11. X post by @swyx
12. X post by @cursor\_ai
13. X post by @cursor\_ai
14. X post by @jediahkatz
15. X post by @bcherny
16. X post by @rileybrown
17. X post by @rileybrown
18. X post by @rileybrown
19. X post by @rileybrown
20. Setting a custom price for a model in AgentsView