Use a Second Model to Steer Your Coding Agent

Coding Agents Alpha Tracker

2026-07-12

Use a Second Model to Steer Your Coding Agent

By Coding Agents Alpha Tracker • July 12, 2026

Today's strongest field signal is a practical multi-model pattern: let one coding agent execute while another provides fresh-context steering at blockers. Also included: a Claude Code-to-GPT routing setup, a second-instance planning loop, and a clear boundary between agent-assisted exploration and hard design work.

TOP SIGNAL

Use a second model as a steering layer—not just a fallback. Salvatore Sanfilippo's production workflow for Dwarfstar CUDA kernels keeps GPT-5.6 as the executor but tells it to consult Claude through the terminal whenever it is blocked; he reports this unlocked optimizations neither model reached alone. The useful mechanism is fresh context: the adviser repeatedly examines the problem without inheriting the primary agent's tunnel. [1]

TRY THIS

- **Give Claude Code a GPT-5.6-Sol route via CLIProxyAPI** *(Tibo, with a setup summary from Theo).* Install CLIProxyAPI, configure Claude and Codex authentication, connect Claude Code, then create a `claudex` alias. Tibo's configuration sets the subagent model, enables effort, caps tool concurrency at three, and disables tool search: [2, 3]

  ```
  alias claudex='CLAUDE_CODE_SUBAGENT_MODEL=gpt-5.6-sol
  CLAUDE_CODE_ALWAYS_ENABLE_EFFORT=1
  CLAUDE_CODE_MAX_TOOL_USE_CONCURRENCY=3
  ENABLE_TOOL_SEARCH=false
  claude -model gpt-5.6-sol'
  ```

  [2]

Tibo describes this as a roughly five-minute path that avoids installing the Codex app; Theo says it took him about two prompts after the proxy was already configured. [2, 3]

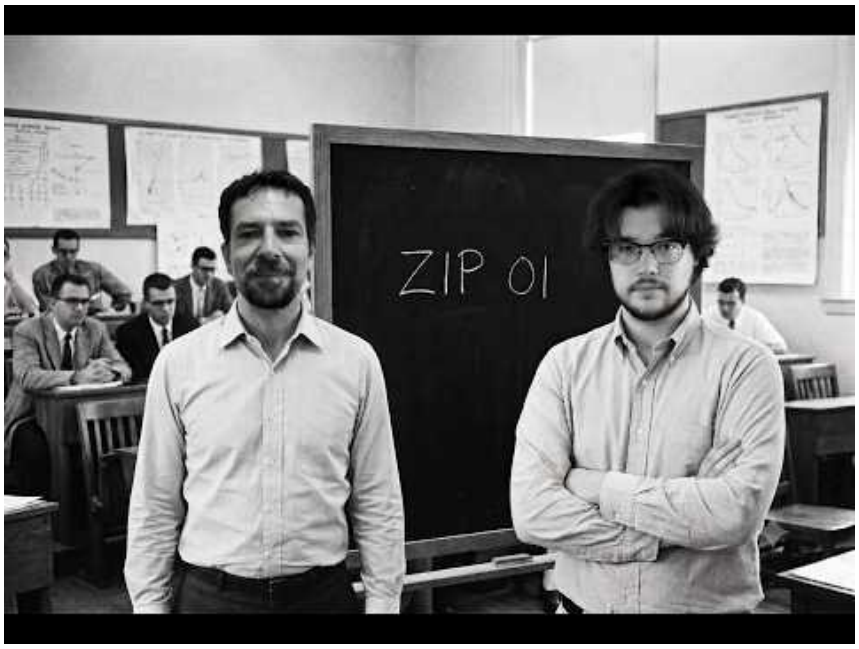
- **Add an explicit "consult another model when blocked" rule.** Start the primary coding agent on the implementation, then instruct it to use a second model through the terminal for suggestions at blockers. Sanfilippo used this GPT-to-Claude pattern on CUDA-kernel work; the secondary model's clean context is the point, not parallel generation for its own sake. [1]
- **Run a planning critique before implementation.** Give a second model the desired outcome plus your proposed approaches, and ask it to surface risks and constraints before transferring the discussion to the primary agent. Sanfilippo specifically recommends this two-instance handoff; he also cites OpenAI guidance that excessive implementation detail can degrade output, so lead with the outcome rather than a prescriptive build plan. [1]
- **Use agents for probes; retain ownership of the hard abstraction.** Armin Ronacher finds Pi useful for quickly building POCs and probing APIs, but not for speeding up cross-provider abstraction design. Treat that split as a task-routing rule: delegate exploration, then keep critical thinking engaged for the design decision—otherwise, in his words, the result is "slop." [4]

WHAT SHIPPED

No additional release with enough new, practical implementation detail passed today's filter; the strongest material was practitioner workflow evidence rather than release notes.

GO DEEPER

- **13:35–14:58 — Sanfilippo on using Claude as GPT-5.6's terminal adviser.** A concise walkthrough of the multi-model steering pattern and why a fresh-context reviewer can help recover an agent that is stuck. [1]



GPT 5.6, Fable e i moscerini della frutta | Zip episodio 01 (13:35)

- **23:52–24:25 — Pre-plan with a second instance.** Watch for the actionable sequence: state the target, offer candidate strategies, ask for failure modes, then bring that critique to the implementation agent. [1]

Editorial take: the durable workflow is not "pick the winning model"—it is using a second model to break local reasoning loops while keeping human judgment on the abstractions that remain hard. [1, 4]

Sources

1. GPT 5.6, Fable e i moscerini della frutta | Zip episodio 01
2. X post by @thsottiaux
3. X post by @theo
4. X post by @mitsuhiko