Qwen's 17GB Local Coding Agent Is Real—After You Kill xhigh

Coding Agents Alpha Tracker

2026-08-17

Qwen's 17GB Local Coding Agent Is Real—After You Kill xhigh

By Coding Agents Alpha Tracker • August 17, 2026

Simon Willison's Qwen 3.8 27B tests turn a 17GB open-weight model into a real Pi coding loop, with speed and reasoning defaults as the remaining constraints.

TOP SIGNAL

The local coding-agent baseline moved up, but the default configuration is actively bad. Simon Willison ran Qwen 3.8 27B as a 17GB local model and, through Pi, got it to answer a multi-file auth question and write and test `pi_jsonl_to_md.py` from a prompt. [1] The catch is `xhigh`: a simple SVG consumed 22,276 reasoning tokens and took 21 minutes, versus 137 seconds with reasoning off; Willison's recommendation is low or no reasoning first, with the full 262,144-token context. [1] The remaining gap is speed, not basic capability: he reports 15–30 tokens per second locally and says performance is what keeps it from daily-driver status. [1]

TRY THIS

- **Tune Qwen before you evaluate it, then give it a thin harness.** In LM Studio, load the full 262,144-token context and start at low or no reasoning; only turn reasoning up when a task actually needs it. [1] For Pi, Simon's working pattern is an OpenAI-compatible provider in `~/.pi/agent/models.json`—replace the endpoint with your own LM Studio host:

```
{
  "providers": {
    "spark": {
      "baseUrl": "https://YOUR-LM-STUDIO-ENDPOINT/v1",
```

```
      "api": "openai-responses",
      "apiKey": "dummy",
      "models": [{"id": "qwen3.8-27b", "reasoning": true}]
    }
  }
}
```

Run `pi --provider spark --model qwen3.8-27b` in the repo. Willison's useful smoke tests were `how does auth work?` followed by `Write Python code to convert this jsonl to markdown`; the agent inspected multiple files, then built and tested the utility. [1]

- **Make 1M context an opt-in long-session mode.** At the top level of `~/.codex/config.toml`, before any section headers, use:

  ```
  model = "gpt-5.6-sol"
  model_context_window = 1000000
  model_auto_compact_token_limit = 900000
  ```

  Restart Codex and start a new session. For a one-off CLI test: `codex -m gpt-5.6-sol -c model_context_window=1000000 -c model_auto_compact_token_limit=900000`. The Codex maintainer's warning is worth keeping: the smaller default was tuned for performance and cost, so treat 1M as an escape hatch for unusually long code, tool-output, or history-heavy sessions. [2]

- **Use `AGENTS.md` as the lightweight instruction layer.** Armin Ronacher says he removed most `CLAUDE.md` files, then found that explicitly telling Claude Code to read `AGENTS.md` worked well enough when he returned to debug a regression. Put the repo's durable rules in `AGENTS.md` and make "Read `AGENTS.md` before changing anything" the first instruction in a Claude Code session. [3]

- **Put a control plane in front of agent fan-out and external writes.** Kent C. Dodds's Kody package fingerprints run errors and triages them without spawning a thousand agents, then creates Cursor cloud agents to fix the affected packages. Copy the pattern: deduplicate by error fingerprint, dispatch one repair per unique failure, and queue or rate-limit writes. [4, 5] DHH's Omabot filed 128 legitimate QA issues in about a minute and still tripped GitHub's spam protection; his separate rule is that increasingly automated development ends with a human merge decision. [6, 7, 8]

WHAT SHIPPED

- **Qwen 3.8 27B** — Apache-2 licensed, 27B, and vision-capable. Simon tested the 17GB Q4 build on an M5 Max MacBook Pro and an NVIDIA DGX Spark; its benchmark lead over Qwen 3.6 27B and closed-weight

Qwen 3.7-Plus is self-reported, with independent benchmarks still pending. [1]

- **GPT-5.6 Sol 1M in Codex** — the 1M context option, previously limited to API-key usage, now works through ChatGPT accounts too. The documented model window is 1,050,000 tokens, but the maintainer repeats that the current default was tuned deliberately for performance and cost. [9, 2]
- **Kody issue triage** — Kent C. Dodds shipped a package that subscribes to error events and creates a Cursor cloud agent to fix errors in the affected packages automatically; the companion description emphasizes fingerprinting and bounded triage rather than unbounded agent spawning. [5, 4]
- **Coming in Omarchy: voice-driven OS changes.** DHH says the next version will integrate Voxtype with the default agent so users can speak requests for widgets, panels, and apps. This is an announcement, not a demonstrated release or benchmark. [10]

GO DEEPER

- **Bilawal Sidhu on OpenClaw → Codex** — Start with Sidhu's migration story: six persona agents on an M1 Max and WhatsApp gave way to Codex as a connected daily driver that he can control remotely from the ChatGPT app without tunneling; he still uses Claude for many coding tasks. [11]

Codex Just Replaced All His Apps | Bilawal Sidhu (3:03)

Continue into the browser-as-shared-canvas workflow: Sidhu triggers YouTube A/B-test monitoring from his phone, while detailed Google Docs comments become Codex's review input and he implements the final fixes himself. [11]

- **Study Simon's `pi_jsonl_to_md.py`.** It is a small, inspectable artifact of the local-agent loop above: Qwen received a single conversion request, wrote the Python utility, tested it, and the resulting tool was used to publish the transcript. [1]

Editorial take: The durable edge today is controlled delegation: tune model behavior, keep context explicit, deduplicate before fan-out, and make every external write or merge pass through a human-controlled boundary. [1, 4, 8]

Sources

1. Qwen 3.8 27B is excellent, but it defaults to wildly overthinking things
2. X post by @thsottiaux
3. X post by @mitsuhiko
4. X post by @kodykoala
5. X post by @kentcdodds
6. X post by @dhh
7. X post by @dhh
8. X post by @dhh
9. X post by @thsottiaux
10. X post by @dhh
11. Codex Just Replaced All His Apps | Bilawal Sidhu