Software that gets built and tested while I sleep

I leave software to build and test overnight, and nothing lands unless the tests and a review agree. Routine steps go to cheap and free models, including a large one on my own laptop, and Claude does the design and the review. Measured figures from 24 October and 3 to 4 November 2026.



Published
4 November 2026

Reading time
1 minute

About
Mullet

Tags
AI, Testing

More on: Mullet

In the evening I plan a piece of work, approve the plan and leave it running. In the morning I read what came back. Nothing has been merged without me.

The job runner is my own. It's called Mullet, and I'm not going to describe how it works. Here is what it does.

On the night of Saturday 24 October 2026, during a weekend of focused work on my personal projects, it built three of its own features. Each was a plan of five to seven steps, and I merged each one only after reading the diff.

Who does what

Claude does the design and the review. The routine steps go to cheap and free models, and one of those is a large coding model that runs on my own laptop's graphics card.

On 3 and 4 November 2026 that local model did its first real job for me: two small fixes across three files. It passed every check on the first attempt in about three and a half minutes, and it cost nothing. That is one job, a sample of one.

In the same long session, the outside models I used for delegated work cost about $0.34 in total. That figure is from Mullet's own record, and I haven't checked it against the providers' bills.

Why it's safe to use cheap models

The saving isn't the point by itself. A cheaper model is only worth using if something independent checks what it did, and every change here faces the same checks whoever wrote it. A wrong attempt costs time and tokens, and it doesn't reach my code.

That is the rule I hold the whole setup to, and it has its own post: the rule that lets me trust work I'm not watching.