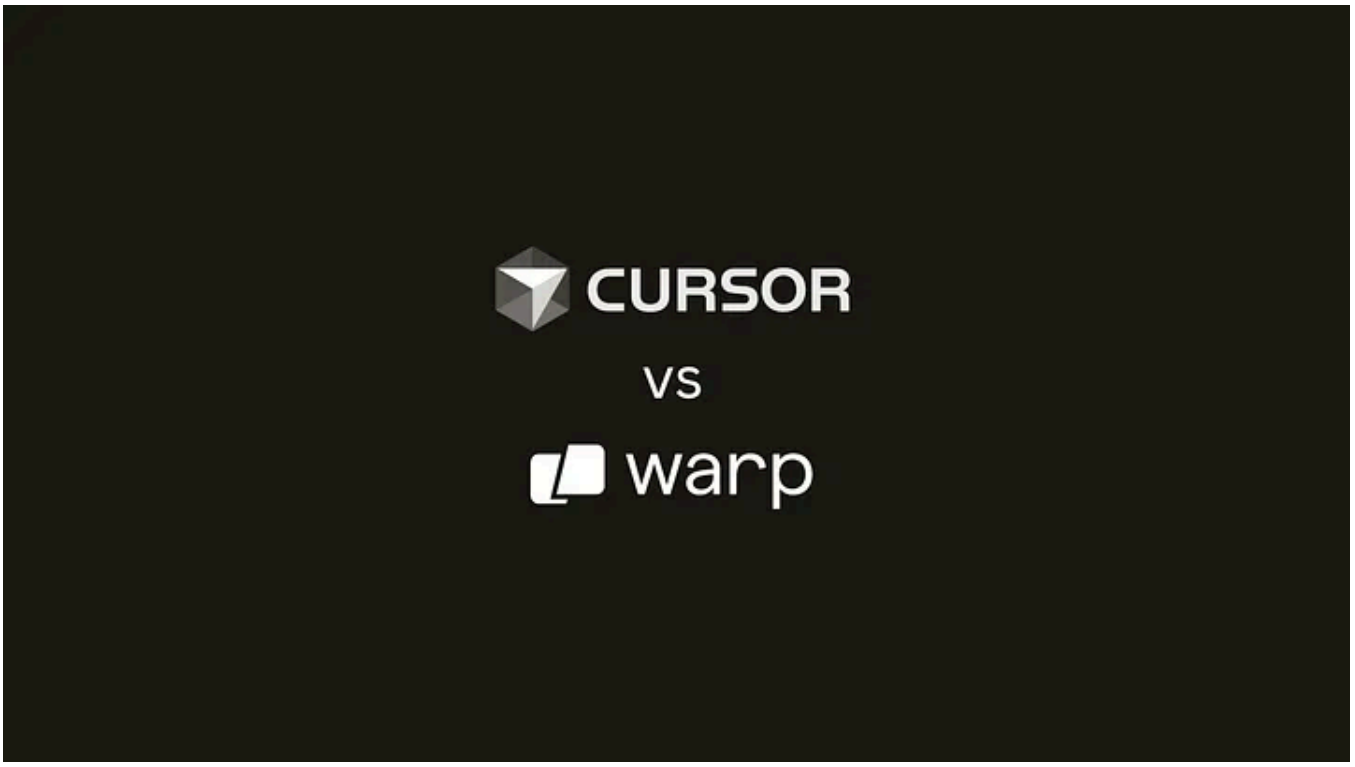


Member-only story

# Cursor vs Warp Review: Warp 3x Faster & 1/6th of the cost of Cursor

8 min read · 2 days ago

Chris Dunlop Follow

Listen Share More

I've run these AI coding tool comparisons before. 3 months ago, I compared Windsurf to Cursor, I built the same Kanban board in both to see which was faster and better.

Cursor demolished Windsurf. It was 3.3x faster, 5x cheaper, and the output was way better.

But then I discovered Warp.

So I thought it was a good chance for another comparison, this time between Cursor and Warp.

*Not a Medium member? Keep reading for free by clicking here.*

## Why Kanban Boards Are The Perfect Test

I find that Kanban boards are the perfect comparison project for AI tools.

Here's why:

**They test everything:**

- Database schema (boards, columns, cards)
- UI components (drag and drop)
- State management
- Real-time updates
- Package installation (drag and drop libraries)

In my Windsurf vs Cursor test, this is exactly what exposed Windsurf's weaknesses. It struggled with migrations, the drag and drop didn't work properly, and it kept stopping every 15–25 calls.

## The Requirements (Same For Both)

I wanted:

- A Kanban board where I could add cards
- Ability to create new boards
- Ability to add cards to columns
- Drag and drop that actually works
- Clean UI using ShadCN

Let's see how they both did.

### Warp — Everything Happens In The Terminal

I open Warp. I'm staring at my terminal.

First, I run my workflow. These workflows are absolute game changers and speed up development immensely. They run little scripts for you.

In one click this workflow:

- Creates the T3 app with all my settings
- Adds my folder structure
- Downloads my design system
- Pulls in my prompt templates

This takes 1 minute.

**Then I just start typing my prompt:**

"Create a kanban board with columns and cards"

It creates a todo list and gets on with it.

Warp starts with a bunch of tasks first

The thing I love about Warp is I never leave the terminal. It's showing me code inline, it's installing packages, it's running migrations. It's the fastest way to setup a project and it's not close.

**The speed is immediately noticeable**

Within 10 minutes I have:

- Working database schema
- List screen for boards
- Kanban board component
- Drag and drop working
- ShadCN components installed and working

## Cursor

With Cursor, I open the app. I have to manually create my T3 app in the terminal.

I have to remember:

- TypeScript? Yes
- Tailwind? Yes
- App router? Yes

- What database? MySQL

Then I create my folders manually.

Then I open Cursor's chat window and paste my requirements.

Already at this stage, I've lost about 5 minutes relative to the Warp workflow and cognitively had to remember more things.

Finally, after a while, I turn on Cursor plan mode.

## Cursor's plan mode is great

Cursor plan mode is a recent feature that you can activate by pressing the button on the bottom left

You write in the prompt and then you get back a little plan with the build button.

All in all it’s a great developer experience.

A nice clean plan with everything laid out nicely. Note you should copy this style of making a plan in all your builds

I found as I was editing the plan, I would constantly get this error.

So I ended up having to press build, despite the fact that the plan wasn't quite right.

By the time Cursor was up to this step below...

Warp had already finished!

That’s huge points to Warp. Speed and iteration loops are everything in AI development.

Meanwhile, Cursor got stuck in the loop on this

Basically, I've noticed this more with Cursor, it gets itself stuck a lot. In the end it just said, cool I won't worry about the database until the end.

Meanwhile on Warp, I had the ability to make multiple boards.

All of the lanes had cards and all the CRUD operations worked.

We are 15 minutes into the test at this point. Warp has finished after 8 minutes.

Cursor meanwhile was really struggling.

These errors have really started to creep up in Cursor since earlier versions.

This is quite strange to be honest as Cursor performed much better under this test 6 months ago, and on the Claude 3.5 model as well.

Here are my results from April 2025.

These were the results from 6 months ago

6 months ago, Cursor completed the test in 16 minutes off 4 requests.

In the test today, the first pnpm dev on Cusor failed

Points go to Warp for the build itself. Warp worked first go. One prompt from Cursor and it fixed the build issue.

Cursor does have a number of great quality of life features. I love the little desktop popup notifications when things are finished.

Cursor delivered a nice clean interface

The actual product itself was really good to use.

The drag and drop on both Warp and Cursor worked but not smoothly. Warp was hard to hit the target and Cursor was more reliable but there was a delay. I'd score Cursor the win on that.

By the time it was done Cursor took 23 minutes in total. Remember Warp only took 8.

### I can tell the difference from 6 months ago.

One cool thing about both products is that you can now see that a lot of work has gone into lowering the prompting time and the general error rates of the code that is produced is far lower than before.

The tradeoff is that there is a speed difference from before, but that's probably because Cursor is wanting to prioritise more reliable code.

### The Results

Let me break down what happened:

**Time:**

- Warp: 8 minutes
- Cursor: 23 minutes

**Output Quality:**

- Warp: 9/10 (everything worked, drag and drop smooth, UI clean)
- Cursor: 9/10 (drag and drop glitchy, had to manually fix migrations)

Both products still struggle with the drag and drop.

**Experience:**

- Warp: Never left the terminal, everything flowed
- Cursor: Constantly switching between terminal and composer

Cost — Warp

Warp summaries are better than Cursor for seeing your usage in the product

Here is my summary, Total cost for Warp is $0.21 USD

I was surprised that Cursor defaulted to the thinking model — I swapped Warp and Cursor both to auto for this test

Total cost is $1.36 USD for Cursor — I think this has been inflated due to going on the thinking model.

## Where Warp Really Shines

The workflows are a game changer.

In my Cursor workflow, I have to:

1. Open terminal
2. Run create t3-app
3. Answer all the prompts
4. Create folders manually
5. Copy design system files
6. Open Cursor
7. Start prompting

In my Warp workflow, I:

1. Open Warp
2. Run workflow with project name
3. Start prompting

That's it.

**The terminal context advantage**

Because Warp is in the terminal, it just understands things better.

When I say "install shadcn components for the kanban board", it:

- Installs the packages
- Adds the components
- Updates the imports
- Runs the dev server to check

All in one flow.

With Cursor, I have to constantly check the terminal, manually run commands, and switch contexts.

I find that it gets lost in the terminal.

Like in the above example, it's like cool let me wait a bit and check if the server starts successfully. It's like Cursor is blind to the terminal itself and that ends up in making everything slower.

## The Things Cursor Does Better

Cursor and Warp are both great tools, they both have different things. I prefer Warp. But here is some things that I like better about Cursor.

**The diff view is nice**

When Cursor proposes changes, you get a nice visual diff. With Warp, you're just reading the code in the terminal.

**The plan mode is better**

I like the plan mode with the build button and there's nice quality of life UI features like the desktop notifications.

**The terminal UI is great**

It's just little things, but I enjoy the file tree and the general VS Code layout.

## The Things Warp Does Better

- **The workflows.** They make everything so fast and especially as you build up a library with them. In this experiment that's why I was able to get Warp to build everything in 8 minutes. Add this up overtime and you can't compare.
- **That it's just one terminal.** This means that if I want to chat or run pnpm dev I just do everything from one place.
- **The overall value ratio and time to benefit.** I find that there is less stuffing around with terminal commands because Warp is a terminal first coding tool. This is a really big advantage when you are doing web development as you constantly require terminal commands.

### The Real Difference: Flow State

Here's what the numbers can't capture.

With Warp, I never left my flow state.

I was in the terminal. Typing. Building. Every command, every prompt, every piece of code — all in one place.

Because it's a terminal I also find it way easier to multi tab and I often do.

With Cursor, I was constantly context switching:

- Terminal for commands
- Cursor for code
- Back to terminal to check
- Back to Cursor to fix
- Terminal again for migrations

This context switching is exhausting. Plus there is no workflows, which at this point for me are basically a non negotiable.

### Putting it all together

Here is a comparison of the two.

. . .

## Before you go

Subscribe to my Substack newsletter **because** you get the following:

- A brand-new article, newsletter-only
- Links to every Medium post I've written in the past week
- Book recommendations every week for you to spend your Audible credits on
- Plus more to come.

AI Cursor Warp Software Development Ai Coding

Follow

## Published in Realworld AI Use Cases

2.4K followers · Last published 2 days ago

I wanted to have a publication that helped share tips and tricks from my consultancy where we implement ai for companies like the Olympic team, the Stock Exchange and other B2C companies.

Follow

## Written by Chris Dunlop

5.7K followers · 39 following

I'll help you code with AI. Cursor tips & Business Strategy. I run a company doing AI for the All Blacks, Olympic Team & the Stock Exchange www.cubdigital.co.nz

## Responses (5)

Bgerby

What are your thoughts?

Xuan
2 days ago

Hey, thanks for the nice comparison 👍 . Do you plan to compare claude code and warp?

6 Reply

Chris Howard
5 hours ago

I love Warp. It is my favourite terminal. But a terminal is not a code editor. I've tried to love Warp for AI code editing, but just can't click with the UI/UX. It's just too different to working in an editor.

If only Warp could be the native terminal in my editor. Then i'd be over the moon!

Reply

Andrew Walsh MD. PhD.
12 hours ago

I love Warp and have used them all. They are changing the pricing though. Heavy users like me will have to pay much more.

Reply

See all responses

## More from Chris Dunlop and Realworld AI Use Cases

In Realworld AI Use Cases by Chris Dunlop

### I thought making money online was fake —then I earned $30,000

I've been trying to earn money online for 10 years and in the past 4 months, I finally did it. This is what I wished I had known.

Sep 4 737 24

In Realworld AI Use Cases by Chris Dunlop

## Why clients pay me 10x more than developers who are better at coding than me

Last week I charged $15,000 for work a better coder would do for $1,500 and I think you should learn these skills now that we have AI

Jun 1 8K 155

In Realworld AI Use Cases by Chris Dunlop

## The 1940s Math Book That Changed How I use Claude Code

The timeless problem-solving method that makes today's AI coding finally make sense.

Oct 9 308 9

In Realworld AI Use Cases by Chris Dunlop

## Finding Clients with Expensive Problems: A Practical Guide to Identifying High-Value Prospects

The difference between a $5,000 project and a $50,000 project isn't the code—it's the cost of the problem you're solving.

Jun 14 1.6K 47

See all from Chris Dunlop

See all from Realworld AI Use Cases

## Recommended from Medium

In AI Software Engineer by Joe Njenga

### Cursor 2.0 Has Arrived — And Agentic AI Coding Just Got Wild

Cursor has released version 2.0 , bringing the most powerful agentic AI we have seen yet, more autonomous than ever before,here's what's...

3d ago 325 6

In Towards AI by Teja Kusireddy

## We Spent $47,000 Running AI Agents in Production. Here's What Nobody Tells You About A2A and MCP.

Multi-agent systems are the future. Agent-to-Agent (A2A) communication and Anthropic's Model Context Protocol (MCP) are revolutionary. But...

Oct 16 1.8K 49

Agen.cy

## 20+ Genius Ways Power Users Are Using Claude Code Right Now

Here are 10+ ways power users are using Claude Code🧵

Oct 23 33 1

ZIRU

## Why Spec-Driven Development Made Me 5x More Productive

And How GitHub Spec Kit Changed Everything

Oct 16 34 2

In Entrepreneurship Handbook by Joe Procopio

## AI Porn Is Here—and We're All Pretending It'll Be Fine

Sam Altman has decided that five years is enough of a head start for the porn companies

2d ago 744 23

In Data Science Collective by Ida Silfverskiöld

## Agentic AI: Single vs Multi-Agent Systems

Building with a structured data source in LangGraph

4d ago 450 9