Member-only story

Agent Development Environment: Your Personal Intern

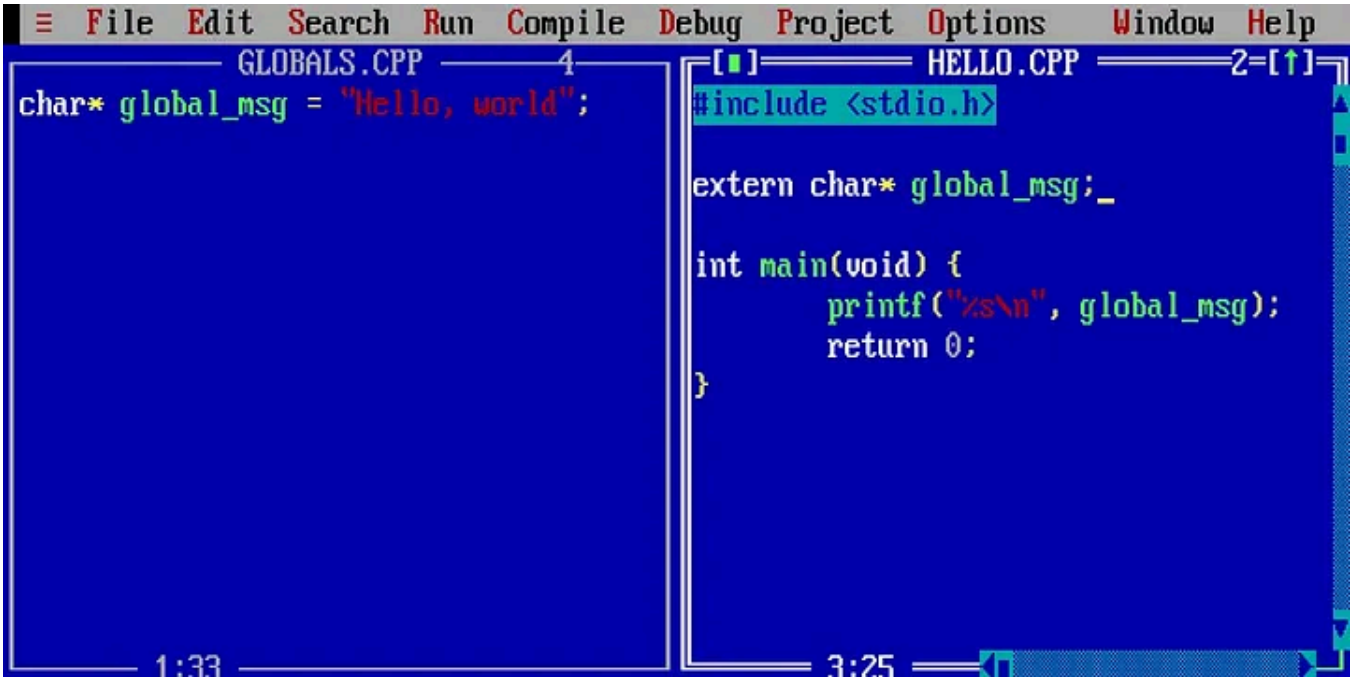
6 min read · Oct 29, 2025

Yashwanth Sai Following

Listen Share More

The first time I opened Turbo C++, I remember staring at it like a kid stepping into a spaceship.
So many buttons. So many panes. So much *power*.

That was the dream of the IDE, your all-in-one command center for software creation. It felt magical. You could build anything from that cockpit.



I remember this felt like some high level stuff back in the day. But all I was doing was writing a function to add 2 numbers XD.

But a decade later, that cockpit feels… old.

Everything takes longer. Things break more often. And somehow, you're doing *more work* just to get *less done*. Somewhere along the way, coding stopped being creative and started being… configurational.

Actually we grew old (Atleast I did, after getting a j*b). Most of the stuff which we have to do, follows some sort of multistep process. "Best practices" we call it. But we all know its boring. We all know it can be automated.

The Old Way

Remember what it took to build a "simple" web app?

You'd open your IDE.
Install dependencies.
Spin up Docker.
Wire up your backend.
Start your local server.

By the time you even *saw* your app, you'd already lost half your day.

The old way rewarded endurance, not imagination. You became a systems janitor before you ever got to be a creator. Now I know all of that is important. Infact you would spend way more time if you didnt follow the "ritual" and jumped straight into coding. But the point is, it can be automated && we can add far more features using our current llm abilities.

Developers juggle a dozen moving parts:

- Multiple APIs
- Half a dozen containers
- Six open terminals
- A browser full of docs
- And an AI assistant running in another tab that still doesn't really get your project

“AI” was supposed to simplify development, but it’s added new layers of chaos. We’re now building half in autocomplete, half in memory and hoping it all somehow runs.

“Vibe Coding” they call it. If you ever worked on a serious codebase, you would know how cringe this sounds.

So what now?

That’s the question, isn’t it? Should we keep up with it, or should we go back to the old painful ways. I believe writing code is the most important and fun activity which I indulge myself in. It still feels the same as it did back when I was in school opening turbo c++ to write some code. It feels magical. I believe the IDEs will evolve into something more like the Jarvis which was shown in the iron man movies.

Lately I am seeing warp push these ideas with some cool features. At first glance, it looks like a terminal. But it’s not. It’s more like a co-pilot that actually *understands* your codebase. I have partnered with warp to test some of their features and present a neat writeup on the same.

I have been using warp from like last year for free

It feels less like typing into a shell and more like talking to your development environment. Its as if my terminal has its own intelligence. This is what I imagined when I said that these IDEs will evolve into something. You can depend on it as if it was a junior intern rather than seeing them as tools to use.

Let me break down what makes it different.

Context-Aware Commands

You know that feeling when you forget a CLI command you've used a hundred times? Yeah, me too. Every. Single. Day. Especially the ffmpeg commands.

This is what makes Warp special: it *gets it*. You don't have to remember or type the perfect thing anymore.

I open my terminal and want to convert a small video into a gif. I can simply type and warp gets it neat!

I can literally just say

> *I wanna convert this mkv video to mp4 format*

or

> *Kill port 8080 and restart the npm app*

Saves a lot of time from endless stack overflow scrolling. After using this, typing raw commands honestly feels like sending Morse code.

Intent-Driven Development

ADE is the here. Warp's agent mode is cool. And in my experience its better than the others which I have used. Just for fun, I have it the Tiny Recursive Model paper and asked it to code it out. Note that this is a project which I myself I am currently going to do. I just wanted to see how well the agent mode keeps up.

Note: This is just a test from my side. I dont really recommend oflloading a task like this to an ai agent. Learning is most important and it happens when you use your brain.

Of course, the goal of this test was to see how far can we go from just a single prompt, and I can say that I am impressed with this. The codebase is very informational, although not comparable to the original code.

This is super powerful, because most papers dont come with code that often. This helps a lot in the process of learning and keeping up with the AI space.

This is just one prompt. You can achieve a lot more with multiple back and forth discussions

The Code Review Panel

When you're working solo, managing your changes is easy. But the moment two or three people join the same repo, it becomes chaotic.

This is a very intuitive way to review changes. This is my personal website codebase

Juggling branches, pull requests, merge conflicts, and half-written commits. Its hellish.

Warp's **Code Review Panel** hits different. You can **review all your open changes across files in one simple view**. No need to bounce between Git diffs, terminals, and editors. Want to see what's changed on your current branch? It's there. Want to compare against `master`? One quick toggle

It's like having a bird's-eye view of your entire codebase, all the moving parts, all the edits, all in one single place. Super powerful feature

Dream of a Sentient Environment

I open my environment, and it already knows what I am working on. It understands the dependencies, stack, and style. I say, "Deploy the staging build," and it just does it with all the right configs.

You don't control the machine anymore. You collaborate with it like its your junior intern.

You can give various tools to the warp agent. There are 100s of 3rd party MCP servers which you can add.

Conclusion

The IDE was built for an era where humans controlled machines. The ADE is built for an era where humans *collaborate* with them.

Tools that win next won't just autocomplete, rather they'll *co-create.* We're entering the **Agentic Development Era,** a shift from syntax to **semantics**, from instructions to **intention**. You don't code anymore. You orchestrate and manage context.

The IDE had a good run, But it's time to evolve our ways.

Thumbnail

AI | Vibe Coding | Ai Agent | Programming | Software Development

Following

Written by Yashwanth Sai

3.7K followers · 32 following

AI @ Vuhosi, Building turilabs.tech, Author of an AI Agents book @ManningBooks. Mail: taddishetty34@gmail.com

Responses (2)

Bgerby

What are your thoughts?

Pannag Kumaar
6 days ago

getting dejavu right now, Man Alt + F9 hitting me harder ....

Waro seems cooler too ....

16 Reply

The Informed Mind
Oct 29

beautifully Written !This brought back memories of those Turbo C++ days . Warp seems really cool — definitely gonna check it out!

Reply

More from Yashwanth Sai

Yashwanth Sai

The DSPy Playbook: A Humble Introduction

A small part from my upcoming book on AI Agents. I'm building in public and sharing it as we go. Stay tuned!

Jul 23 480 5

Yashwanth Sai

A Prompting Playbook for Building Multi-Agent AI Systems

Lessons from Anthropic's Latest Study

Jul 13 128 1

Yashwanth Sai

This AI Tool Can Help you Build a Powerful Network

A career in tech isn't just built on what you know, but who you know. For students, recent graduates, and lifelong learners, this advice...

Oct 17 115

Yashwanth Sai

Why Start a Research Lab? 1 year of Building my Indie AI Lab

One year, 5 research papers, ~10 part time researchers, neurips workshop, and lots of learning.

5d ago 152 3

See all from Yashwanth Sai

Recommended from Medium

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 3.4K 116

Aakash Gupta

How Solo Founders Are Building $1M+ SaaS Businesses Using Only AI (Complete Playbook)

Maor Shlomo sold his company for $80 million.

Oct 28 198 4

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...

Oct 29 777 18

In Data Science Collective by Ida Silfverskiöld

Agentic AI: Single vs Multi-Agent Systems

Building with a structured data source in LangGraph

Oct 28 720 15

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

Oct 31 2.3K 104

In The Generator by Thomas Smith

The Right Way to Use AI At Work

An experiment from Stanford shows why we're all doing it wrong

Oct 28 1.1K 26