# THE CODEX APP CHANGES EVERYTHING.

No, it doesn’t. The Codex desktop app dropped yesterday. You'll see breathless Twitter posts and YouTube videos about how it changes everything. It doesn't. But it is pretty cool, and it's part of a larger trend worth paying attention to. I'm going to talk briefly about how it's changing my workflow, and then zoom out to what it means that this app exists at all.

## My Workflow (For Now)

I'll write a longer post on this, but the quick version:

* My primary driver is Claude Code in the terminal. I think it has the best features, the most hooks, and the most ability to create a clean development workflow with all the checks I want.
* The [Codex app](https://openai.com/index/introducing-the-codex-app/) is my parallelization layer. The thing that's cool about it (and [Conductor](https://www.conductor.build/), which is similar) is that it makes Git worktrees easy to use. That means real parallelization.

Here's how I'm experimenting with it:

1. I have my main feature or project running in Claude Code in a terminal window
2. Whenever I come up with changes, bug fixes, or investigations outside the scope of that feature, I spin up a worktree in the Codex app. I can chat with it separately. It lets me know when it needs input. It's totally isolated, and I can merge it back whenever I want.

The TLDR: Codex app is OpenAI's supported UI for multi-agent parallelized development. In my workflow, I use it to develop small features in parallel while I'm working on the main thing in Claude Code.

## The Bigger Picture

The reason I find this interesting isn't the app itself, but what it says about where things are headed. I think about IDEs a lot because they're a lens into where software development is going. I've said this before: software development will be unrecognizable in two to three years. And what's happening with IDEs is proof.

"IDE" stands for integrated development environment. The name doesn't imply it has to be about reading and writing code - but that's what it's always been. That's changing.

Here's the thing: **I don't read code anymore.** I used to write code and read code. Now when something isn't working, I don't go look at the code. I don't question the code. I either ask one of my coding agents, or - more often - I ask myself: what happened with my system? What can I improve about the inputs that led to that code being generated?

The code isn't the thing I'm debugging. The system that produced the code is. The people really leading AI coding right now (and I'd put myself near the front, though not all the way there) don't read code. They manage the things that produce code.



## The Continuum

The image above illustrates how I think about this landscape. There's a spectrum with three major zones: Code, Agents, and Specs. The further left you move, the higher up the stack you get.

**Code (right side)**: Traditional IDEs. VS Code, JetBrains. You read code, you write code.

**Code + AI**: AI-assisted features. Autocomplete, inline suggestions. GitHub Copilot lives here. The human is still driving.

**Agentic IDEs**: Cursor, Windsurf. Code and agents combined. The AI makes autonomous multi-file edits, runs terminal commands, iterates on its own work. But you're still looking at code.

**Multi-Agent Orchestration**: Claude Code, Codex CLI, Codex app, Conductor. The whole interface is about managing agents. You're not staring at code - you're dispatching tasks, watching progress, reviewing PRs. Agent inbox.

**Specs (left side)**: Kiro, GitHub Spec Kit, [Vibe Scaffold](https://vibescaffold.dev/). The spec is the primary artifact. Requirements → design → tasks → implementation. Code is an output, not the thing you manage.

## Where This Is Going

I think the industry is moving left. Toward specs. The code is becoming an implementation detail. What matters is the system that produces it - the requirements, the constraints, the architecture. Get those right, and the code follows. I’m actually building something in this area, focused on specs (not Vibe Scaffold 🙂). Hopefully I have some details in the next few weeks.