Don’t Vibe. Spec.

*Don’t Vibe Code — **Spec Your Code** for Better AI Results*

Stop wasting time “vibe coding” with AI tools. The future of AI-assisted development belongs to clear specifications, structured planning, and AI-friendly documentation. Learn how to use SPEC.md, DESIGN-SPEC.md, and plan mode effectively with Claude Code and Gemini CLI to get reliable, consistent results.

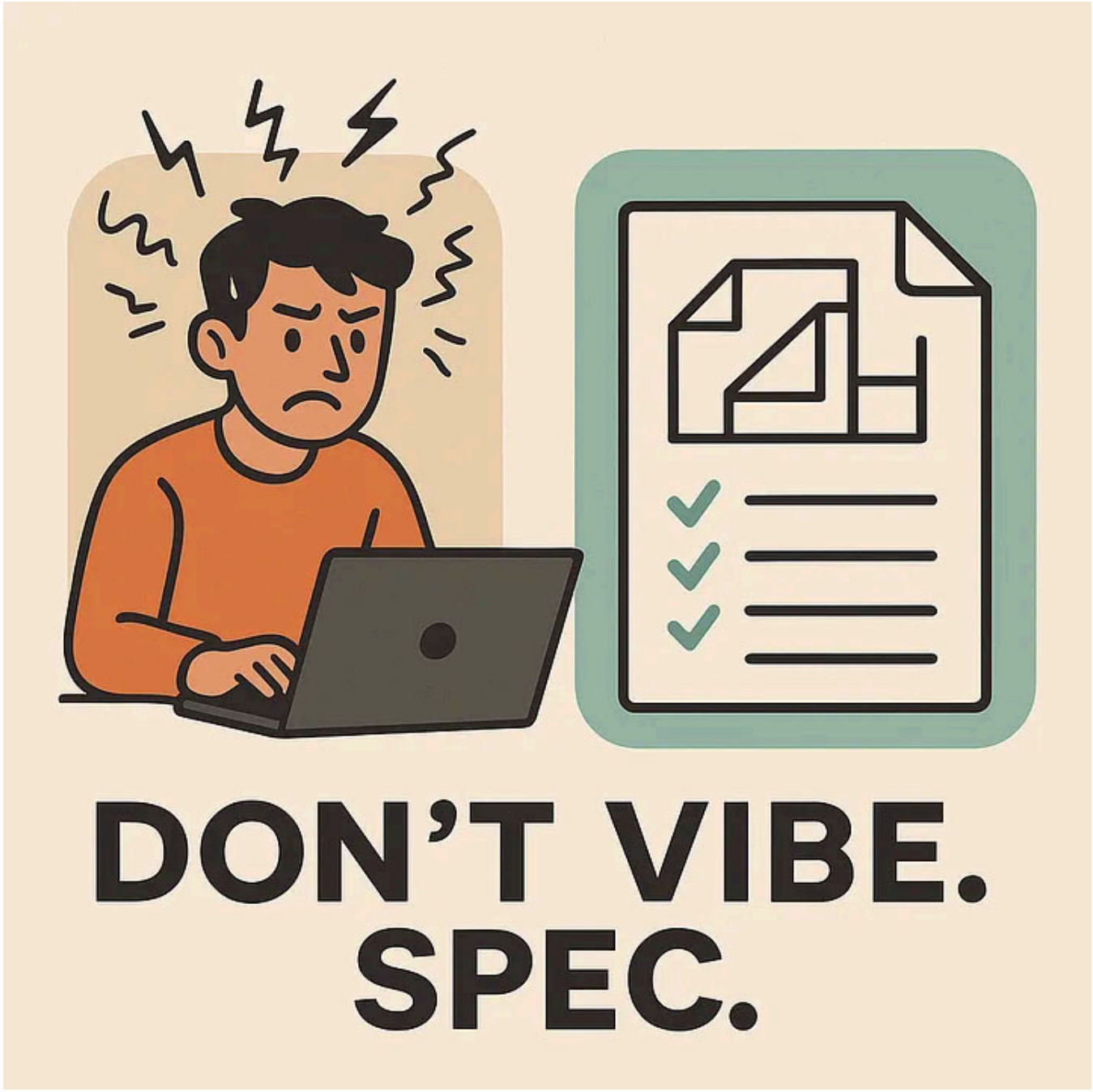
Vibe Coding is Dead ☠️

You may have read my previous article on vibe coding.

Vibe Coding: The Good, Bad, & Fixes

Can an LLM Build an App from Scratch?

medium.com



Forget it. Still vibe coding? That's ancient history — like 20 minutes ago. That's a long time at the pace AI is advancing!

It used to feel like you were coding with "vibes" because, let's face it — you had no idea what was happening. AI coding even a few months ago is very different from what you can do today.

AI assisted coding has come a long way in just a few months. The application "Sark" that I was trying to build back then now exists. And the Anthropic team that built it did a better job than I could have imagined. It's called Claude Code.

I'm not a big fan of using an IDE with AI, mainly because I expect AIs to advance fast enough to make IDEs less necessary. So, I've been getting comfortable with terminal-

based AIs like **Claude Code** and **Gemini CLI.** Yeah, yeah terminal-based will quickly be replaced by something else. But for techies like me, I'm deeply comfortable staying within the command line even if something "better" comes along.

Here's some advice on how to think of these tools — and how to play with them effectively.

⚔️ Make Specs, Not War

If you've ever fought with your AI to understand the problem or track down a bug, there's a better way:

👉 Write clear specifications.

Instead of treating your AI like a coding partner, think of it as an **outsourced development team with high employee churn.** Every new session is a brand-new employee who doesn't know what was discussed before.

To solve this, both Claude Code and Gemini add Markdown files (**CLAUDE.md** and **GEMINI.md**) to your project to preserve context across sessions.

But don't stop there. Give your constantly rotating "team" a hand by providing a **detailed SPEC.md file.**

You can write this manually if you're precise. Or, have a chat AI help draft it.

Clearly explain:

- What you're building
- Who it's for
- What you need it to do
- What its limitations should be

Even if you write this yourself, paste it into a chat AI to polish it. After all, an AI's output is shaped by its training — so it will likely understand structured AI-formatted input better than freeform human notes.

🗓️ Short-Term & Mid-Term Planning

In addition to specs, keep **short-term and mid-term planning files.** These steer your AI where you want to go. Without them, the AI will choose its own "best path

forward" — which may not be what you intended.

With **Claude Code,** I'm fine letting it decide most next steps — it's been great!

With **Gemini CLI,** I've found it often wanders off, forcing rollbacks.

🧭 The Magic of Plan Mode (Claude)

Claude Code has an excellent **plan mode** (Shift-Tab twice). You'll see `# plan mode on` at the bottom left of the terminal.

Get Sevak Avakians's stories in your inbox

Join Medium for free to get updates from this writer.

Enter your email

Subscribe

Here's how to use it:

1. Ask Claude what to work on next — or give it a goal.
2. Claude will think holistically and generate a **step-by-step plan.**
3. You then choose:

- **1)** Accept the plan and let it auto-execute
- **2)** Accept the plan but approve each step
- **3)** Reject and adjust

I often choose **3** — because after seeing the plan, I realize I forgot to mention something. I'll say: *"Yes, execute that plan, but also [insert what I forgot here]."*

Claude will create additional plan files when necessary but usually integrates progress into **CLAUDE.md** or even **README.md.**

🎨 DESIGN-SPEC.md & README.md

Consider adding a **DESIGN-SPEC.md** file to guide your AI:

- Branding
- Colors
- Layouts
- Look & feel

The more detail, the better your output.

Don't forget the **README.md** — the foundation of your project docs. Include:

- Project description
- Purpose
- Usage instructions
- Whether it's a web app, SaaS, mobile app, etc.

✅ Final Word: Specs Save Time

Specs don't have to be hard — especially since AI can help generate the boring parts. But including them will:

- Save you from AI "going rogue"
- Keep the AI on track over many sessions
- Reduce wasted cycles
- Keep your project aligned with your vision

Claude Code has blown my mind. If you plan well and learn to become an AI whisperer, you can have it help you build full production ready applications. Yes, back in April of this year (2025), my final thoughts on this were:

The hype about AI replacing software engineers is (still) premature. But it's not fantasy. AI has been able to make me at least 60% more efficient. This will improve with newer tools wrapped around AI models, like Cursor and Manus.

We're edging toward a future where coding becomes collaboration — between humans and agents. And yes, you'll still need humans, at the very least to provide intent. But fewer of them. More focused. More strategic. And, with the help of AI, much faster.

Now, August 2025, I believe we're "there" with AI coding. I've given Claude Code some complex, nuanced tasks and it has handled most of them completely without my help. Only a few times have I had to help it understand my intent. But this is less times than I've had to do the same with professional developers.

Your turn. Go build something awesome without putting in all that effort.

👉 **Don't vibe code. Spec your code.**