The Hacker Ethic and the Vibe Coder

April 2026

4 min read • 987 words

In 1984, Steven Levy published Hackers: Heroes of the Computer Revolution and formalized something the early computing community already knew in their bones: that building software carries ethical weight. The hackers at MIT in the 1960s, the hardware tinkerers in Silicon Valley in the 1970s, the open source movement that followed — they all operated under a shared set of principles that nobody had to enforce because the people writing code understood the systems well enough to feel responsible for them.

The principles were simple:

- Access to computers should be unlimited and total.
- Information should be free.
- Mistrust authority — promote decentralization.
- Judge people by their work, not their credentials.
- You can create art and beauty on a computer.
- Computers can change your life for the better.

These weren't idealistic platitudes. They were load-bearing ethical commitments that shaped how the internet was built, how open source works, how the free software movement thinks about human freedom. Richard Stallman put it

plainly: the hacker ethic is about "the feelings of right and wrong" in a community — "that knowledge should be shared with other people who can benefit from it."

This ethic assumed something important: that the people building software understood what they were building.

The New Developer

We're now producing a new kind of developer at scale. The vibe coder.

Vibe coding is what happens when AI handles the implementation and the human handles the intent. You describe what you want. The model writes the code. You run it, see if it works, adjust. The feedback loop is fast and the results can be impressive. People are shipping real products this way. I'm not here to say it doesn't work.

But something gets lost in the translation.

When you learn to code the traditional way — when you fight with memory allocation, trace through stack frames, debug race conditions at 2 AM — you develop an intuition for how systems actually behave. You learn that software is not magic. It's infrastructure. It runs on other people's machines. It touches other people's data. It makes promises through its interfaces that other people depend on.

> Levy called this the "hands-on imperative" — the belief that essential lessons about systems and the world come from taking things apart, seeing how they work, and using that knowledge to build something better. You can't develop that understanding by prompting alone.

The hacker ethic emerged from that understanding. You can't feel responsible for a system you don't understand. You can't mistrust authority if you can't read the code the authority is running. You can't value information freedom if you've never thought about what information is at the infrastructure level.

Vibe coding skips the part where you develop the intuition that makes the ethics feel obvious.

The Power Problem

Here's what worries me. The barrier to building software has never been lower. That's genuinely good — more people creating things is a net positive for the world. The early hackers believed in universal access, and AI-assisted coding is the closest we've come to realizing that vision.

But the early hackers also understood that access without understanding is dangerous.

A vibe coder can ship a web app in an afternoon. That app might collect user data, make API calls to third-party services, store credentials, handle payments, run on shared infrastructure. Every one of those is an ethical surface. Every one of those can harm real people if handled carelessly.

The hacker ethic wasn't just about what you could build. It was about what you should build, what you shouldn't build, and for whom. The "hands-on imperative" — learning by taking things apart — wasn't just a learning style. It was the foundation of ethical responsibility. You understand the weight of the thing because you've held it in your hands.

When the AI holds it for you, where does the weight go?

What We Owe the Next Generation

I'm not arguing against AI-assisted development. I use it every day. It's the most significant shift in my creative process since learning to code. But I came to it with twenty years of understanding what code actually does, how systems fail, where the sharp edges are. The AI amplifies my judgment. It doesn't replace it.

For someone whose entire experience of programming is describing what they want and watching it appear — that person needs the hacker ethic more than the rest of us, not less. They have the power without the instinct for responsibility that traditionally came bundled with the craft.

We need to teach it explicitly now. It used to be ambient — you absorbed it through years of open source contribution, through debugging other people's code, through the slow accumulation of understanding that makes you careful. That pipeline is being bypassed.

So here's what I think we owe the vibe coders:

- **Teach them where the code runs.** Not abstractly. Concretely. Whose server. Whose data. Whose trust.
- **Teach them what information is.** Not as product feature but as responsibility. Every database is someone's life reduced to rows.
- **Teach them to read code, not just generate it.** You don't need to write every line. But you need to be able to read what you're shipping. If you can't audit it, you shouldn't deploy it.
- **Teach them the history.** The hacker ethic exists because people thought carefully about the power they were wielding. That thinking isn't optional just because the tooling changed.

With great power comes great responsibility. That's not a comic book line. It's the entire moral architecture of computing, and we're handing the power to people who've never heard of the architecture.

The hacker ethic wasn't about gatekeeping. It was the opposite — it was about making sure that as access expanded, understanding expanded with it. We're at the moment where access has leapt ahead of understanding, and the gap is growing.

Closing it is the most important thing the existing developer community can do right now.
