

# Vibe Coding: The Hangover

21. november 2025 | 10 min

## When AI Codes Faster Than We Can Think (Part 2/2)

AI can now build complete web apps in minutes. But behind the magic lies a new form of technical debt... invisible, untested, and perhaps irresponsible.

In Part 1, I described how vibe coding became the software world's wild west, an explosion of creativity, freedom, and AI-driven innovation. But where there is freedom, responsibility follows, and there can be unintended consequences. This part is about the hangovers: security flaws, lost accountability, and the new demand for digital judgment.

## When Speed Exceeds Understanding

AI can now write complete web apps faster than we can read the documentation. But development is about both making it work and also understanding why it works and how it can fail.

A growing number of software engineers report that AI-generated code often works on the first test but collapses later. The code is poorly organized or has hidden dependencies.

## As one senior developer put it:

"It's not that AI produces more bugs per line. It's that those bugs reach production before anyone notices."

## Case 1: Lovable's Security Issue

In March 2025, security researcher Matt Palmer discovered that over 10% of all web apps built with Lovable had the same critical flaw in database security. The flaw allowed unauthorized users to retrieve names, emails, and payment information. Not smart.

When the issue was made public, it turned out that the AI code had reused an outdated example from an open-source forum. No humans had reviewed it. No man in the loop...

## Case 2: Replit's Database Deletion

In July 2025, SaaStr founder Jason Lemkin experienced Replit's AI agent deleting the production database. The AI panicked over empty queries and removed data on over 1,200 executives in the database. Worse still: The AI initially said rollback was impossible (it wasn't) and tried to hide the error... The prototype had worked fine enough. Until it didn't anymore.

## Challenges With AI Coders

**Low Security Awareness:** Security flaws like hardcoded API keys and missing rate limiting are often seen in AI-generated code from novice users, especially in rapid prototypes.

**Tough Debugging:** AI-generated code quickly becomes opaque if no one documents the decisions along the way, making debugging more difficult.

**Speed vs. Review:** AI tools accelerate code production so much that organizations' existing review and security processes can't keep up.

Databricks AI Red Team found that even "well-running" AI code often contains critical vulnerabilities like arbitrary code execution.

## Why This Is an Organizational Problem

The real problem is only partly technical, but especially organizational. When everyone can build software in minutes and hours instead of days and weeks, accountability shifts... Who owns quality? Who validates data security? Can IT architecture keep up?

These questions are now central in any company where AI is part of the development process.

Leaders think that AI means fewer developers, but in practice it means more products, more features, faster milestones, and if governance and quality routines don't keep up, there's a greater risk of errors.

Compliance teams are already reporting 'AI shadow IT': small internal tools built with no-code and AI platforms that inadvertently expose personal data.

AI optimization typically focuses on making it work here and now, not on GDPR's principles of data minimization and privacy by design.

## How to Vibe Code Responsibly

### 1. Security-First Prompting

Start all projects with a security-focused system prompt. Explicitly ask the AI to use "least privilege principles," input

validation, and secrets management (if you don't know where to start, use this sentence and ask your AI to build a system prompt).

## 2. Human Review

Treat the AI like a junior developer: give it bounded tasks, build one small feature at a time, and always review output. You can in principle have another AI as a code reviewer, as I mentioned in the Part 1 article, but the responsibility for the code is still yours.

## 3. Use It for the Right Things

Vibe coding is excellent for prototypes, weekend and hobby projects, and learning, not for critical systems. Example: I'm finishing building a beach volleyball tournament app in Lovable.dev.

### The Industry's Wake-Up Call

As the hype wave subsided, the first structural challenges began to emerge. Fast Company wrote in September 2025: "The vibe coding hangover is upon us." Large companies like Databricks and Anthropic introduced internal rules requiring that AI-generated code be reviewed by a human before being merged to production.

OpenAI's own "Code Safety Initiative" is now working on marking potentially unsafe AI code with a kind of "nutrition label" for software.

### The Reality Behind Productivity

A study from METR Research in July 2025 showed that experienced developers on average spent 19% more time when working with AI assistants, primarily because they spent more time on review and verification, even though they felt they were faster. They spent less time writing code but more time understanding what the AI had actually done.

New developers, on the other hand, experienced 20-25% productivity gains. So AI makes the inexperienced faster and the experienced more cautious.

### The Future of Vibe Coding

Vibe coding won't die. It will just mature over time. It will likely become part of the professional toolbox. This isn't the end of programming, it's the beginning of a new discipline where creativity, responsibility, and AI must coexist.

### Karpathy himself has written:

"Ultimately, vibe coding full web apps today is kind of messy and not a good idea for anything of actual importance. But

there are clear hints of greatness..."

The future developer becomes an AI orchestrator who formulates visions, designs processes, and evaluates output from a network of AI agents.

---

Har du spørgsmål om vibe coding, eller vil du høre, hvordan det kunne se ud i din organisation? Kontakt mig - jeg hjælper gerne med at komme i gang.

---

*Stefano Vincenti | AI Advisor & Trainer | aitrainer.dk*