



Menu ▾



| 31 January 2026 | 6 min read

# AI Makes the Easy Part Easier and the Hard Part Harder

#ai-workflows

#developer-experience

A friend of mine recently attended an open forum panel about how engineering orgs can better support their engineers. The themes that came up were not surprising:

*Sacrificing quality makes it hard to feel proud of the work. No acknowledgement of current velocity. If we sprint to deliver, the expectation becomes to keep sprinting, forever.*

I've been hearing variations of this for a while now, but now I'm also hearing and agreeing with "AI doesn't always speed us up".

## "AI did it for me"

Developers used to google things. You'd read a StackOverflow answer, or an article, or a GitHub issue. You did some research, verified it against your own context, and came to your own conclusion. Nobody said "Google did it for me" or "it was the top result so it must be true."

**Now I'm starting to hear "AI did it for me."**

That's either overhyping what happened, or it means the developer didn't come to their own conclusion. Both are bad. If someone on my team ever did say Google wrote their code because they copied a StackOverflow answer, I'd be worried about the same things I'm worried about now with AI: did you actually understand what you pasted?

## Vibe coding has a ceiling

Vibe coding is fun. At first. For prototyping or low-stakes personal projects, it's useful. But when the stakes are real, every line of code has consequences.

On a personal project, I asked an AI agent to add a test to a specific file. The file was 500 lines before the request and 100 lines after. I asked why it deleted all the other content. It said it didn't. Then it said the file didn't exist before. I showed it the git history and it apologised, said it should have checked whether the file existed first. (Thank you git).

### **Now imagine that in a healthcare codebase instead of a side project.**

AI assistance can cost more time than it saves. That sounds backwards, but it's what happened here. I spent longer arguing with the agent and recovering the file than I would have spent writing the test myself.

Using AI as an investigation tool, and not jumping straight to AI as solution provider, is a step that some people skip. AI-assisted investigation is an underrated skill that's not easy, and it takes practice to know when AI is wrong. Using AI-generated code can be effective, but if we give AI more of the easy code-writing tasks, we can fall into the trap where AI assistance costs more time than it saves.

## Hard part gets harder

Most people miss this about AI-assisted development. Writing code is the easy part of the job. It always has been. The hard part is investigation, understanding context, validating assumptions, and knowing why a particular approach is the right one for this situation. When you hand the easy part to AI, you're not left with less work. You're left with only the hard work. And if you skipped the investigation because AI already gave you an answer, you don't have the context to evaluate what it gave you.

Reading and understanding other people's code is much harder than writing code. AI-generated code is other people's code. So we've taken the part developers are good at (writing), offloaded it to a machine, and left ourselves with the part that's harder (reading and reviewing), but without the context we'd normally build up by doing the writing ourselves.

## Sprint expectations and burnout

My friend's panel raised a point I keep coming back to: if we sprint to deliver something, the expectation becomes to keep sprinting. Always. Tired engineers miss edge cases, skip tests, ship bugs. More incidents, more pressure, more sprinting. It feeds itself.

This is a management problem, not an engineering one. When leadership sees a team deliver fast once (maybe with AI help, maybe not), that becomes the new baseline. The conversation shifts from "how did they do that?" to "why can't they do that every time?"

### **My friend was saying:**

*When people claim AI makes them 10x more productive, maybe it's turning them from a 0.1x engineer to a 1x engineer. So technically yes, they've been 10x'd. The question is whether that's a productivity gain or an exposure of how little investigating they were doing before.*

Burnout and shipping slop will eat whatever productivity gains AI gives you. You can't optimise your way out of people being too tired to think clearly.

## Senior skill, junior trust

I've used the phrase "AI is senior skill, junior trust" to explain how AI coding agents work in practice. They're highly skilled at writing code but we have to trust their output like we would a junior engineer. The code looks good and probably works, but we should check more carefully because they don't have the experience.

Another way to look at it: an AI coding agent is like a brilliant person who reads really fast and just walked in off the street. They can help with investigations and could write some code, but they didn't go to that meeting last week to discuss important background and context.

## Ownership still matters

Developers need to take responsible ownership of every line of code they ship. Not just the lines they wrote, the AI-generated ones too.

If you're cutting and pasting AI output because someone set an unrealistic velocity target, you've got a problem 6 months from now when a new team member is trying to understand what that code does. Or at 2am when it breaks. "AI wrote it" isn't going to help you in either situation.

## How can AI make the hard part easier?

The other day there was a production bug. A user sent an enquiry to the service team a couple of hours after a big release. There was an edge case timezone display bug. The developer who made the change had 30 minutes before they had to leave to teach a class, and it was late enough for me to already be at home. So I used AI to help investigate, letting it know the bug must be based on recent changes and explaining how we could reproduce. Turned out some deprecated methods were taking priority over the current timezone-aware ones, so the timezone was never converting correctly. Within 15 minutes I had the root cause, a solution idea, and investigation notes in the

GitHub issue. The developer confirmed the fix, others tested and deployed, and I went downstairs to grab my DoorDash dinner.

No fire drill. No staying late. AI did the investigation grunt work, I provided the context and verified, the developer confirmed the solution. That's AI helping with the hard part.

---

 28

Share:   

---

## Related Articles

| 7 February 2026

### Claude Code /insights Roasted My AI Workflow (It Wasn't Wrong)

What is Claude Insights: The /insights command in Claude Code generates an HTML report analysing your usage pattern...

[Read article >](#)

| 13 December 2025

### When to Ask a Second AI: Signal-Based Adaptive Orchestration

With chatbots and coding agents we've all experienced those moments that make us stop trusting the first answer. From...

[Read article >](#)

| 8 December 2025

### Are You an Idiot Sandwich? Building an SEO Self-Sabotage Scanner in 5 Hours

At work someone asked how to rank higher. I said wrong question, first check if you're sabotaging yourself. The most common...

[Read article >](#)

[About](#) [Articles](#) [Books](#) [Mental Models](#) [Projects](#) [Uses](#)

© 2026 Matthew Hansen. All rights reserved.