

GUEST ESSAY

The A.I. Disruption We've Been Waiting for Has Arrived

Feb. 18, 2026

Listen to this article · 11:17 min [Learn more](#)**By Paul Ford**

Mr. Ford is an essayist and a technologist.

On weekday evenings, heading home on the subway from Union Square in New York City, I log into an A.I. tool from my phone and write a prompt. “Look at the data in the files I just uploaded,” I tap. “Load it into a database, then make it searchable with a web interface.” Underground in the subway tunnels my internet connection drops, but when my train emerges onto the Manhattan Bridge, I get a few minutes to see all the work my coding agent has done, and if I type fast enough I can issue another prompt. By the time I get home to Brooklyn, my little project tends to be done: a website, a feature in a music app, a complex search tool or some tiny game.

This is called “vibe coding,” a term coined a year ago by the artificial intelligence expert Andrej Karpathy. To vibe code is to make software with prompts sent to a specialized chatbot — not coding, but telling — and letting the bot work out the bugs. Like many other programmers, I use a product called Claude Code from Anthropic, although Codex from OpenAI does about as well, and Google Gemini is not far behind. Claude Code earned \$1 billion for Anthropic in its first six months. It was always a helpful coding assistant, but in November it suddenly got much better, and ever since I’ve been knocking off side projects that had sat in folders for a decade or longer. It’s fun to see old ideas come to life, so I keep a steady flow. Maybe it adds up to a half-hour a day of my time, and an hour of Claude’s.

November was, for me and many others in tech, a great surprise. Before, A.I. coding tools were often useful, but halting and clumsy. Now, the bot can run for a full hour and make whole, designed websites and apps that may be flawed, but credible. I spent an entire session of therapy talking about it.

The tech industry is a global culture — an identity based on craft and skill. Software

development has been a solid middle-class job for a long time. But that may be slipping away. What might the future look like if 100 million, or a billion, people can make any software they desire? Could this be a moment of unparalleled growth and opportunity as people gain access to tech industry power for themselves?

According to the market, the answer is no. Recently, software stocks — Monday.com, Salesforce, Adobe and many others — plummeted all at once; the Nasdaq 100 lost half a trillion dollars in two days. Legal software company stocks slumped recently because Anthropic released tools to automate some legal work. Financial services firms and real estate services — the market keeps devaluing them because traders expect there to be less need for humans at desks in an A.I.-automated future. Why will anyone need all that legacy software when A.I. can code anything up for you in two shakes of a robotic lamb's tail?

Personally this all feels premature, but markets aren't subtle thinkers. And I get it. When you watch a large language model slice through some horrible, expensive problem — like migrating data from an old platform to a modern one — you feel the earth shifting. I was the chief executive of a software services firm, which made me a professional software cost estimator. When I rebooted my messy personal website a few weeks ago, I realized: I would have paid \$25,000 for someone else to do this. When a friend asked me to convert a large, thorny data set, I downloaded it, cleaned it up and made it pretty and easy to explore. In the past I would have charged \$350,000.

That last price is full 2021 retail — it implies a product manager, a designer, two engineers (one senior) and four to six months of design, coding and testing. Plus maintenance. Bespoke software is joltingly expensive. Today, though, when the stars align and my prompts work out, I can do hundreds of thousands of dollars worth of work for fun (fun for me) over weekends and evenings, for the price of the Claude \$200-a-month plan.

That's not an altogether pleasant feeling. The faces of former employees keep flashing before me. All those designers and JavaScript coders. I could not hire the majority of them now, because I would have no idea how to bill for their time. Some companies, including IBM, think A.I. will create tons of new jobs. But no one thinks they'll be the same as the old jobs.

Is the software I'm making for myself on my phone as good as handcrafted, bespoke code? No. But it's immediate and cheap. And the quantities, measured in lines of text, are large. It might fail a company's quality test, but it would meet every deadline. That is what makes A.I. coding such a shock to the system.

An axiom of programming is “real artists ship.” That was something Steve Jobs once said to remind his team that finishing and releasing a product matters more than

endlessly refining it. Much of the software industry is organized around managing ship risk, and the possibility that a product never actually makes it out to the world. A good technology manager assumes that a product will never ship for launch, that every force is arrayed against it and that the devil himself has cursed it — and then the manager works back from that. Even if all these obstacles are surmounted, the software will ship late. Remember, Steve Jobs returned to Apple in 1997 only because Apple couldn't ship a new version of its operating system, so they bought his company, NeXT. And the direct descendant of NeXT's software is what's running on Macs and iPhones in 2026. In software, sharp change is to be avoided at all costs. The risk is just too high.

Except ... what if, going forward, it's not? What if software suddenly wanted to ship? What if all of that immense bureaucracy, the endless processes, the mind-boggling range of costs that you need to make the computer compute, just goes *poof*? That doesn't mean that the software will be good. But most software today is not good. It simply means that products could go to market very quickly.

And for lots of users, that's going to be fine. People don't judge A.I. code the same way they judge slop articles or glazed videos. They're not looking for the human connection of art. They're looking to achieve a goal. Code just has to work.

There are many arguments against vibe coding through A.I. It is an ecological disaster, with data centers consuming billions of gallons of water for cooling each year; it can generate bad, insecure code; it creates cookie-cutter apps instead of real, thoughtful solutions; the real value is in people, not software. All of these are true and valid. But I've been around too long. The web wasn't "real" software until it was. Blogging wasn't publishing. Big, serious companies weren't going to migrate to the cloud, and then one day they did.

But right now, excited developers are overextending themselves to the point of burnout, obsessively coding all the time. Open source projects are deluged by A.I. submissions, often from bots pretending to have found a security bug requesting a payment. People trumpet the Jevons paradox, which points out that greater efficiency often leads to more consumption — but at the same time, would it surprise you to find out tomorrow that large technology consulting firms had just laid off 10,000 people? A hundred thousand? A million?

The market keeps convulsing, and I wish we could hit the brakes. But we live in a brakeless era.

No matter where you work, my hunch is this is coming for you. Have you noticed the software you use every day adding "A.I. features"? That's the top of the slippery slope. Whatever unifying principle equates to ship risk in your industry, people are trying to

mitigate it with A.I. Insurance, finance, architecture, manufacturing, textiles, every kind of project management — they want to automate it all through A.I.

When large language models aren't enough, companies use "world models," which simulate physical reality, not just language. One of the best-known users of that technology is Waymo, Alphabet's self-driving taxi company. In the perfect Silicon Valley system, bots would write code to run the bots that drive the taxis, with new code every minute. Every app creates itself. Who will be able to afford the taxi rides offered by this system? That's for a different department to answer.

I've spent my last few years working with a team to build an A.I. software platform, trying to help clients and customers navigate all of these changes. That sounds like the perfect job for the moment, right? It's not. Every six months, some new A.I. bomb goes off in our industry, and we have to metabolize the change, reset our product, change our strategy and marketing and adapt, at great expense. Our road map keeps getting pushed back as a result of all this "progress." Everyone is fried.

This is all exacerbated by how much of the A.I. industry is led by people who see human thought as raw material, like a steel manufacturer sees ore. The industry is arranged into an ouroboros of mutual investments, with the world economy teetering on their sweetest dreams. Social change at this level needs careful, federal governance and thoughtful regulation. But we're being handed the opposite: racist A.I. video slop shared on Truth Social, Grok doing who-knows-what inside the Pentagon, and a White House policy that would give the U.S. attorney general the power to challenge any state's attempt to regulate A.I. No brakes.

All of the people I love hate this stuff, and all the people I hate love it. And yet, likely because of the same personality flaws that drew me to technology in the first place, I am annoyingly excited.

Here is why: I collect stories of software woe. I think of the friend at an immigration nonprofit who needs to click countless times, in mounting frustration, to generate critical reports. Or the small-business owners trying to operate everything with email and losing orders as a result. Or my doctor, whose time with patients is eaten up by having to tap furiously into the hospital's electronic health record system.

After decades of stories like those, I believe there are millions, maybe billions, of software products that don't exist but should: dashboards, reports, apps, project trackers and countless others. People want these things to do their jobs, or to help others, but they can't find the budget. They make do with spreadsheets and to-do lists.

My industry is famous for saying no, or selling you something you don't need. We have an earned reputation as a lot of really tiresome dudes. But I think if vibe coding gets a little bit better, a little more accessible and a little more reliable, people won't have to

wait on us. They can just watch some how-to videos and learn, and then they can have the power of these tools for themselves. I could teach you now to make a complex web app in a few weeks. In about six months you could do a lot of things that took me 20 years to learn. I'm writing all kinds of code I never could before — but you can, too. If we can't stop the freight train, we can at least hop on for a ride.

The simple truth is that I am less valuable than I used to be. It stings to be made obsolete, but it's fun to code on the train, too. And if this technology keeps improving, then all of the people who tell me how hard it is to make a report, place an order, upgrade an app or update a record — they could get the software they deserve, too. That might be a good trade, long term.

Paul Ford is an essayist and a technologist. He is a co-founder and the president of Aboard.com, an A.I.-powered software acceleration platform.

The Times is committed to publishing a diversity of letters to the editor. We'd like to hear what you think about this or any of our articles. Here are some tips. And here's our email: letters@nytimes.com.

Follow the New York Times Opinion section on Facebook, Instagram, TikTok, Bluesky, WhatsApp and Threads.

A version of this article appears in print on , Section SR, Page 8 of the New York edition with the headline: A.I. Is Making Me Less Valuable. Hooray.