The Four Horsemen of Agentic Coding

Slop, Alienation, Deskilling, and Team Fallout


ALEX MARTSINOVICH
SEP 29, 2026

35 4 9 Share

Agentic coding is undeniably very useful. It's also very bad. Or rather, it's having some horrible effects on us, our craft, and our relationships with each other. I can feel it in my bones, and a lot of others can too. But whenever I try to explain what *exactly* is wrong, I find myself waving my hands wildly and jumping from point to point. What do you mean, what's wrong? So many things are wrong!

Fine, here's a list. 4 problems with no solution in sight–or, as I call them, the four horsemen of agentic coding.

Slop

LLM-generated code has a strong smell that makes codebases repulsive to humans.

LLMs aren't humans, and the way they write code is... different. Not strictly worse, but different for sure. We quickly came up with a name for it—slop—and it undoubtedly carries a negative connotation.

For plain prose, humanity seems to be converging on the opinion that AI prose is bland at best and an insult at worst. For code, due to its instrumental nature, the debate is far from over. A lot of people latch on to the idea that very soon we won't have to read the code at all. "Remember, it's the worst the models will ever be," they say.

And yet, the latest generation of models is surprisingly underwhelming. They're certainly smart, but they seem to be moving towards "unemployable smart" rather than "inspiring smart." Claude now famously communicates entirely through word salad, and Astra writes in a bizarre competitive code-golfy style, incomprehensible to normal humans. Sloppiness turned out to be a surprisingly persistent property of LLM output and at this point looks like a signature move rather than a growing pain.

Which is very bad news for people who still want to poke around in the codebase and maintain some familiarity with it, especially in a team setting. Once agents are allowed, they very quickly take over. What used to be a shared space for humans where each team member contributed became an AI wasteland where people don't want to spend time.

But what if we don't have to spend time in these slop caves? Why spend your time there when you can be a design mastermind and reign over a swarm of loyal agents from the comfort of the

Cookie Policy

We use cookies to improve your experience, for analytics, and for marketing. You can accept, reject, or manage your preferences. See our privacy policy.

Manage Reject Accept

Engineers become alienated from code and, as a result, care less.

Software engineering used to be a fairly... kinesthetic field of work. Maybe not to the same degree as woodworking, but we loved our tools too, both real and virtual. Te
cult-like following. Popular color themes and programming fonts had their o
keyboards were a hot thing once, and there was no shortage of keyboard sou

We produced code directly, by hand. It's only natural that this resulted in a st
ownership of the fruits of our labor. Sure, this feeling varied depending on th
if you just assembled something from a set of building blocks, you would fee
IKEA certainly knows that!

Agentic coding has dramatically increased the distance between engineers ar
their work. This gap is so large that none of our kinesthetic tools can cross it.
experience code directly anymore: we send an agent with a vague set of instr
skim through its report. Hell, even typing is optional—you can just yap at yo
driving.

So naturally, we become more detached. We care less. We lose touch with our craft. And it shows. It shows in poorly thought-out features that receive no pushback. It shows in band-aid bugfixes that don't address the root cause. It shows in how unfulfilled we feel at the end of the day. For software to spark joy, it has to be imbued with its creator's love.





Discover more from Distant

I can't shut up about AI

Enter your email...

Subscribe

By subscribing, you agree Substack's Terr
acknowledge its Information Collection Notice

Already have an account? Sign

Cookie Policy

We use cookies to improve your experience, for analytics, and for marketing. You can accept, reject, or manage your preferences. See our privacy policy.

AI erodes existing skills, impedes learning, and doesn't offer any meaningful skill progression.

I think at this point it shouldn't be controversial to say that deskilling is real. There is no shortage of people reporting their wits slipping away after prolonged use of AI. Which makes sense: a lot of things in life are use-it-or-lose-it, and professional skills are no exception. Yes, you'll always know how to ride a bike, and I myself will die before I forget how to exit Vim, but anything beyond that is fair game.

I've made the point before that we're currently experiencing a historical anomaly. We have all those seasoned engineers who spent years doing things the hard way and can now hold LLMs exactly the right way to be insanely productive. But unless we figure out how *learning* works in the age of AI, this supply won't last long.

One common piece of advice is that if you're at the beginning of your career, you simply shouldn't use AI. And I completely agree, but it's important to understand that this isn't a viable strategy for humanity as a whole. Everyone will not just.

We can't be blind to the fact that the introduction of an "easy button" has led to a tectonic shift in incentives to learn. And if I know one thing about humans, it's that they follow incentives.

Another common response to the deskilling argument is that LLM usage comes with its own learning curve. You just need to upskill and learn the intricacies of downloading markdown files! I'm not buying it. The learning curve is basically non-existent.

A printing press or a chainsaw at least requires some skill to operate, but the fancy new magic box? You don't have to know how to read or write. A caveman could build a B2B SaaS and be on time for his weekly podcast appearance. And if you think your custom agent orchestration workflow is an important feat of engineering, just you wait.

Team Fallout

Agents weaken the social fabric of engineering teams by reducing interpersonal communication, increasing self-reliance, and hindering skill expression.

Slack channels used to be crowded with people. Anything that your rubber duck couldn't answer would go into a team chat. Now the chat is a pale shadow of its former self. Everyone is too busy talking to their agents. Each of us has received a magic familiar that follows us around like we're some kind of Disney protagonists. It's very smart, available 24/7, and will never report you to HR. The rubber duck can finally talk back!

In a certain sense, being more autonomous is virtuous. Why distract colleagues with your silly nonsense? The problem is, helping other people with silly nonsense is *our thing*. It's how we build and maintain relationships. You need to walk the path; otherwise, it will be overtaken by grass.

Cookie Policy

We use cookies to improve your experience, for analytics, and for marketing. You can accept, reject, or manage your preferences. See our privacy policy.

Agentic coding also changes the way we see other people. Teams used to be sort of like parties of adventurers: a Git wizard, a Rust witch, a bard who is really into mechanical keyboards. There was plenty of room for skill expression, and the diversity of expertise was celebrated.

Now, an average team consists of a Claude operator, a Codex whisperer, and a guy with AI psychosis who insists on committing .md files. “This is John—you’ll want to see how he *prompts*,” said nobody ever.

And that’s true: nobody wants to see how John prompts. There is something repulsive and fundamentally unsexy about talking to an agent. It *feels* fine when *you* are doing it, but others... ugh, get a ~~chat~~room. Their prompts are cringe; their agents are weird. Like looking at someone else’s TikTok feed.

So here we are, still peers in the org chart, but growing more and more distant over time. I used to praise your Vim skills and Advent of Code solutions. These days you prompt Claude all day, and I do too. Are we even a team anymore?

So that’s my list. And I’ll be honest with you, I have no idea what to do with any of this. I can only hope that eventually we’ll move forward in this cursed discussion. However beneficial AI is, we pay dearly for it with our curiosity, craftsmanship, and social connection.

AI discussion will continue until the morale improves

Type your email... Subscribe



35 Likes · 9 Restacks

Discussion about this post

Comments Restacks



Write a comment...



Clinton Blackburn 11h

I've struggled with these feelings over the past week as I take some time away from work and the LLM madness. I miss discussions and debates about architecture and solutions to problems. I

Cookie Policy

We use cookies to improve your experience, for analytics, and for marketing. You can accept, reject, or manage your preferences. See our privacy policy.

article and comments. Perhaps we were the ones hallucinating the values existed in the first place?

I don't begrudge folks who want to solve problems by any means, and not worry about the alienation or deskilling. That's their right. However, it's now clear to me that I have to make a conscious effort to express what my own values are and to build/join and organization that shares them.

LIKE (2) REPLY SHARE



DamienClem 7h

"learn the intricacies of downloading markdown files!"

I love this.

I often feel like they are not even necessary because the existing codebase is the harness. But we still keep a lot of control in my team.

I am tasked to define how we will further use all this and adapt AI to our work, and I consider applying a strong "AI in the loop, humans everywhere" principle, instead of the "looping AI where humans were" dogma.

LIKE REPLY SHARE

2 more comments...

© 2026 Alex Martsinovich · Privacy · Terms · Collection notice
Substack is the home for great culture

Cookie Policy

We use cookies to improve your experience, for analytics, and for marketing. You can accept, reject, or manage your preferences. See our privacy policy.