

△ AI is removing the middle class of software engineeringpracticesvibecodingblog.florianherrengt.com

50 via facundoolano 2 days ago | caches | 18 comments

You must be logged in to leave a comment.

PostPreview

bediger4000 2 days ago

The title of the linked-to blog post is true, and the contents are true, they just have no explicit, cause-and-effect relationship. The title has been the desire of the management class for at least 30 years that I'm aware of. Microsoft promised managers that software engineering would pretty much go by the wayside with the introduction of Windows NT circa 1994. There was next to nothing about this in the official, printed, bumpf that went with NT, but in person, Microsoft salespeople promised managers huge programmer productivity gains, 5x-10x, and NT would make all programmers plug-compatible. Managers could just hire and fire downtrodden Windows code monkeys. This turned out to be marketing gibberish, and Microsoft lies, but back then, managers believed it.

In 1998, I worked for US WEST, one of the "Baby Bells". The Communication Workers of America struck, so suddenly, I was in management, and had to Loop Assignment. As soon as the strike was over, I wasn't in management and I had to give back my corporate AmEx card. The manager class is really uneasy about sharing power with a new group, they already have to deal with lawyers, and they hate that. The last 30 years of having to listen to software engineers as a category, and take them seriously, has been traumatic. The title of this article describes the end result of many, many attempts to stuff the genie of a new non-worker category of employees back in the lamp.

You don't have to believe me on any of this, but that doesn't make it less true.

- [−] 🇵🇷 FRIGN 2 days ago

△ This is an extremely interesting perspective, thank you! It still shocks me how addicted companies are to MBAs and other bean counters who are often barely accustomed to the industry, terminology and/or technology the respective company offers. I can give so many examples of well-running companies who succumbed to non-engineering leadership.

22
- [−] 🧑 mattgreenrocks 2 days ago

△ Interesting, it's sort of an evergreen desire: the results without paying so much for it.

9

Of course, that pesky thing called reality has a surprising amount of detail...
- [−] 🌿 alexkorban 2 days ago

△

8

I think there have been two competing trends in terms of making software engineers replaceable, funnily enough driven by software engineers themselves as much as by management.

On one hand, automation, standardisation, and open information (blogs, forums, StackOverflow) have commoditised the profession to a great extent. The web platform and all the cloud stuff have contributed too in some ways. It's never been easier to replace a person with another person. Probably the greatest contributors to commodifying our labour have been free software and its suit-and-tie descendant open source.

On the other hand, software engineers have fought back against this with endless technology churn, incidental complexity, and convoluted legacy systems. (I'm mostly joking here - really it's just an unintentional side effect of the dynamics of software organisations.)

[-]

 kghose 2 days ago

△

4

Would have expected upper management to actively work to eliminate middle management since they are the competition.

Current AI is actually a good candidate to eliminate strategy, effort and finance planning and such roles.

The squishy manage small to medium teams of human roles should still be around for a while and that's basically labor too.

[-]

 mattgreenrocks 2 days ago

△

9

Would have expected upper management to actively work to eliminate middle management since they are the competition.

I suspect the contempt comes from the fact that they *need* developers to do the work, and there's no way around that. They have more levers for middle management.

[-]

 creesch 44 hours ago

△

8

Would have expected upper management to actively work to eliminate middle management since they are the competition.

In my experience middle management is a convenient buffer layer for upper management. Basically they are the punching bag on both sides as they need to carry all the bad news down and also deal with all the sentiment going back up.

[-]

 cajually 35 hours ago

△

~

Every upper manager needs their kennel of middle managers that give them the "data" they want. If you have little alignment in upper management this causes pressure for wider middle management.

[-]

 abeyer 31 hours ago

△

~

Would have expected upper management to actively work to eliminate middle management since they are the competition.

The Gervais Principle is this:

Sociopaths, in their own best interests, knowingly promote over-performing losers into middle-management, groom under-performing losers into sociopaths, and leave the average bare-minimum-effort losers to fend for themselves.

[–]  adrien 32 hours ago

△
~ | *In 1998, I worked for US WEST, one of the "Baby Bells". The Communication Workers of America struck, so suddenly, I was in management, and had to Loop Assignment.*

Can you explain that? It's very american, maybe even american-only for some things.

One way I can read it is that you got into management because your manager was striking and you replaced them?

And what's Loop Assignment?

[–]  bediger4000 30 hours ago

△ Sorry, I did an inadequate proofreading of that.

~

The USA had a giant telephone system, The Bell System, which served the majority of the nation until 1983. The US courts deemed it a monopoly, and caused it to be broken up into regional companies, the "Baby Bells". US WEST was one of those regional, serving as I recall 14 US states from Iowa and Minnesota, to Oregon and Washington. Huge land area, relatively sparse population. It's been merged and re-merged, and telephones are mostly of no use.

The Communication Workers of America was one of two unions that a lot of the non-management US WEST employees belonged to. They went on strike summer of 1998. I was not a unionized employee so by default, I was management, so I did not get to go on strike. Due to what I worked on, I knew something of the intricacies of how telephone lines were laid out, so the strike planners put me in "loop assignment". Phone companies called the circuit from central office to customer a "local loop". We had access to the outside plant inventory, so technicians hooking up or repairing phone service had someplace to get extra help if the physical reality didn't match what their work orders said to do. Loop assignment involved looking up the work order, figuring out which pair of wires (26-gauge twisted pair, denoted by color, brown/brown-and-white for instance) could be used at a given telephone pole, or pylon or cross box, and how they should be jumpered together.

 martinald 2 days ago

I think the article is probably right, but also a lot of very smart people are getting into programming because of 'vibe coding' and learning at a tremendous rate.

Programming had a pretty high bar of entry. I certainly remember many decades ago wanting to give in so much fighting basic syntax errors on my projects (self teaching myself from books).

It took a lot of persistence to break through that. "Vibe coding" lets you learn very differently, everything is "too easy" until you end up with a slop mountain. But I have noticed curious and smart people really picking up the concepts quickly. It's like learning the opposite way down.

I've no real idea how this will turn out long term though.

[–]  trenchant 2 days ago

△ You are right on the money.

2 I think it will turn out fine. *Digital* software enabled >10-1000x the developers vs punch-cards per expensive machine. FORTRAN/COBOL/BASIC/C/Java enabled another >10-1000x growth in programmers vs the assembly population. JavaScript and the rise of the Internet and web browser + SaaS enabled another 10-1000x more programs than we were capable of deploying before, as a species.

For proof, see my PS5 unable to play Fortnite because the devs released yet another 100GB update. Pay no mind to how much RAM we went to the moon with, and how no "engineer" we think of when reading these articles has read NASA's coding guide from then. Look instead forward: the update's contents could have been streamed to me as needed, while I was playing, saving Epic cloud costs and giving me another evening in which to buy skins. Why haven't they built it?

Yes, right now AI turns out horribly wasteful and bad results. Every leap forward in abstraction has, when used by the clueless/useless. But the bottleneck is still the human mind, still the interface from the computer to the real world, still the same as it always was.

Pushing programming forward has often been about being the most hard-headed and motivated to build something that you believed possible. The bar just lowered a *bit* on the two-decade/computing-era scale of things, but it will still take a lifetime for the majority of the working population to catch on and be presented with the advantage, by which time the leading edge will have moved. The majority of the programming population...well...we've always had to keep up.

It will turn out the same way it always has. With lots of awesome advancements, and some disasters along the way thanks to man's hubris, see Therac-25's software cost-savings strategy. As individual engineers who don't take on further responsibility, we can only control our own contributions to each category. And man, I have seen some really cool things built and used daily by guys who would in the past never dreamed of even *talking* to a programmer, in some cases even using software instead of paper!

Personally, in the long term, I worry about how things turn out if articles like this are the reading material of today's up and coming. I remember as a high school and college student reading HN and Reddit pre lobste.rs. I would have fostered the same sort of frustrated helplessness that it seems inspired the author to tell this tale, on my school projects when everyone else was cheating, at my internships when products failed. What I have to ask is: where is the adult in the room? Why does no one stand and take accountability, responsibility, and a little bit of control? Plenty of organizations lack this (by design?), but you can only fully control your own actions as the "most senior member on the team." For everyone else there's influence: firing, stern talking to, removing merge rights, admitting that you're not the most senior, venting online, or confronting your own role in your own life.

I know which I would have chosen before I had a year to meditate on it.

--

Further reading, for posterity (feel no need to reply please sir, I am also venting into the Internet, not trying to offend or goad you. Only support, agree, and talk too much by sharing in return :-))

michiel 2 days ago

You were never expected to understand every service and every database. But at least someone did and would explain it to you.

I think to companies, this is a potentially desirable outcome of LLM-driven software development; instead of knowledge, understanding and accountability about the software system being located in one or more employees, who have to be retained, listened to, and/or have replacements trained, LLMs *potentially* offer the perspective of storing this knowledge digitally, turning it into intellectual property that is entirely owned by the company.

The main problem with this idea is that coding tools *as they exist today* are sycophantic lackeys entirely under the control of developers who, as the author points out, are incentivized to game the worst kind of productivity metrics.

This is just another example of the optimism bias that causes people to construct a comforting narrative where AI takes everybody else's job, but not their own.

Loup-Vaillant 42 hours ago

You are reading an article. Or watching a video. The beginning is a classic setup, second person, very first word is "You". Then comes the disruptive element... Wait a minute, you *know* this writing style! Where it comes from, *what* it comes from.

Look, I don't mind the occasional assistance, but when it's that visible it drives me off pretty quick.

[–] greysonp 28 hours ago

△ Honestly, I'm usually pretty turned off of AI writing styles and have become pretty sensitive to it, but ~ IMO this isn't it. This seems human written. I also ran a couple scans and nothing about it seems generated. Writing takes a lot of effort, and I think readers owe writers some amount of due diligence before throwing around accusations.

[–] Loup-Vaillant 27 hours ago

△ The well has been poisoned I guess. Every single YouTube video I have seen so far that began like this ~ turned out to be slop. I'm out of patience now, and I confess, on a hair trigger.

[–] bmo 28 hours ago

△ It's possible this is an example of the "proto-LLM slop" that people actually used to write by hand and ~ AI was trained on and adopted. One of the tells is the constant 1-2 sentence paragraphs, which I don't see as much in slop blog posts. This is common on LinkedIn, and while LinkedIn is all AI now, it was common there 5 years ago too.

Don't be too fast to ascribe to AI the horrors we're capable of creating ourselves.