

Turn Your Best Programmers into Managers



MOHAMED ABOELEZ on NOVEMBER 27, 2022 — 2 COMMENTS



Photo by Jonathan Borba on [Pexels.com](https://www.pexels.com)

More ranting and frothing at the mouth from me, I'm afraid.

Well, maybe a bit more positive this time.

As you may have noticed, I'm one of a small, but growing, crowd who believe that People Who Make Software Happen (you may know them as "programmers") should have the option to progress their careers without leaving the code face.

There are a couple of very good reasons for this:

1. If they love to do it (and that's usually what distinguishes the average from the best programmers), then wrenching them away from doing what they love can take a huge toll on their motivation and morale. I'm, sure we've all worked with [development managers who just can't help sticking their noses into the code](#). It's not because they're interfering [busy](#) bodies (well, not usually).

It's because, really, deep down, they'd like to be writing the [code](#) themselves. They're probably writing it in their heads as we discuss it – like a retired chess champion shouting at the TV because they would have played a different move. It's actually a bit cruel to take that away from them. Admiral Kirk needs his starship.

2. If they're good at it, then surely better to let them continue? Especially if less experienced programmers would benefit from their considerable experience. The lack of older faces on [software](#) teams is a problem, in my experience. We may be up on all the latest buzzwords like "test-driven behavior" and "domain-oriented backlogs", but these old hands can still teach us a thing or two about delivering working [software](#). Indeed, they did.

It's worth remembering in this shiny nuclear-powered Agile era that Tom Gilb wrote [Principles of Software Engineering Management](#) over 900 years ago. Or thereabouts. What's new is us young upstarts hearing about it for the first time and thinking we've discovered something new. And I can't help feeling that's partly because we're not exposed to folk who've been around long enough to remember. Teams burn a lot of time relearning the same lessons people were learning decades ago.

I also feel that, even though they may still be at the [code](#) face, it's purely prejudice that suggests that they therefore can't be involved in strategic or executive decision-making. It's

perfectly possible for someone who still performs surgery to play a decisive role in how a hospital is run. Or for someone who still represents clients in court to be a big cheese in a law practice.

So why couldn't a Person Who Still Makes [Software](#) Happen sit on the board of a company and have an equal say in how it's run? Because they're not qualified? Is the Director of HR better qualified? Or the VP of Marketing?

Other, more mature professions have demonstrated that it's quite possible – even desirable – to encourage the best do-ers to carry on doing and reward them for getting better at doing it. They encourage the best do-ers to help less experienced do-ers do it better, too. And they understand that doing is the best qualification for making decisions about how it should be done.

Some enlightened companies have this. They offer unfettered career progression in the technical disciplines and highly value the knowledge, creativity and decision-making ability of their most experienced and capable do-ers.

If attitudes can change, and the prejudice that if someone wants to carry on writing [software](#) then they are not strategic thinkers and not capable of making rational business decisions can be debunked, then maybe we can look forward to a time when people who love to make

[Software](#) Happen have sufficient control over how they get to Make Software Happen, and then, I suspect, better software will happen.

[Share on Facebook](#)

[Tweet](#)

[Save](#)

CATEGORIES

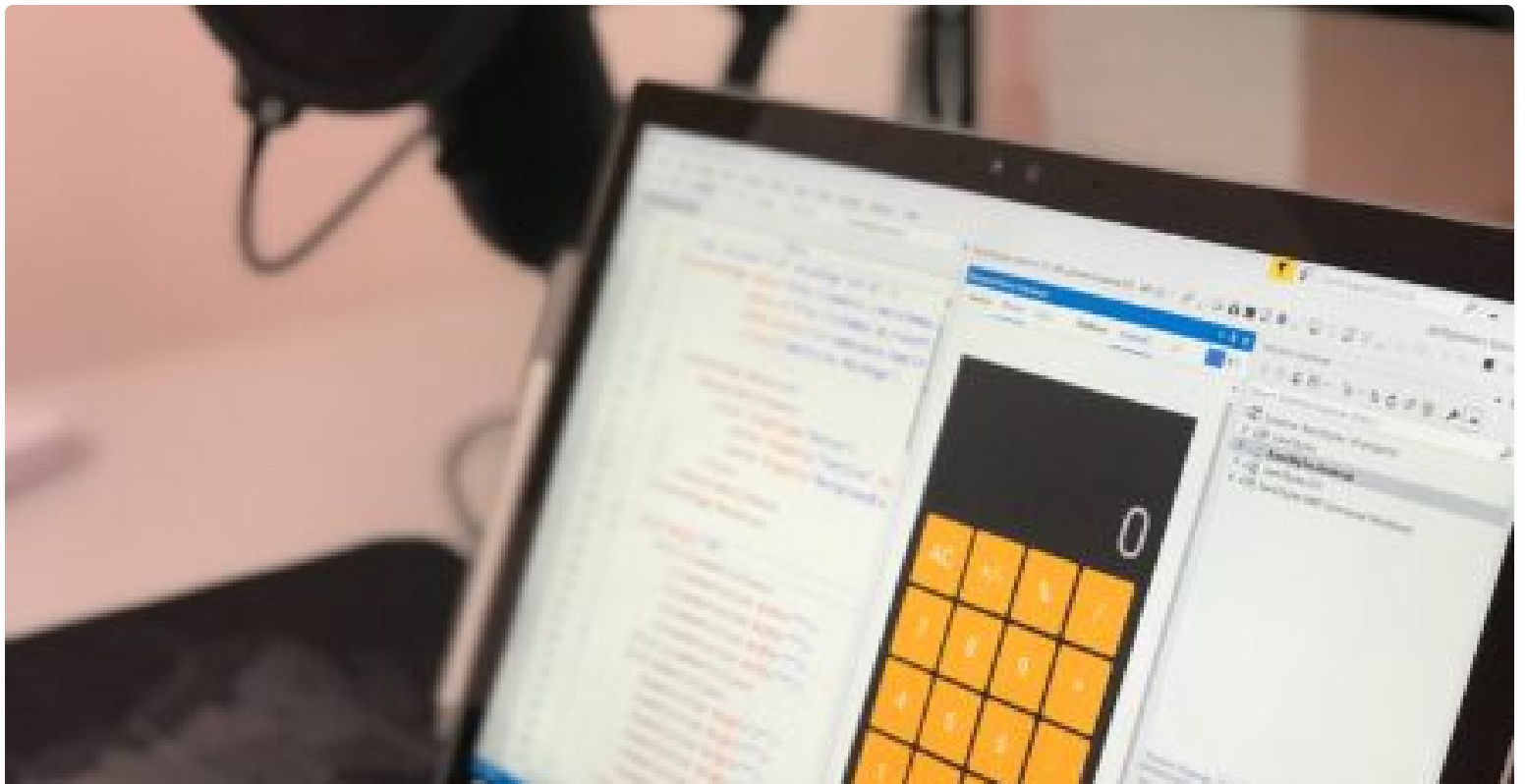
[Programming](#), [Software Development](#)



PROGRAMMING

Should I Run All Tests After Every Refactoring?

When we run our suites of automated tests, we do it to reduce risk. Specifically, to reduce the risk that...



What If We're Building The Wrong Thing?

I had a very interesting couple of days in Finland, hosted by those lovely folk who organize the Scandinavian Agile...

2 thoughts on “Turn Your Best Programmers into Managers”

desireesmithies

December 24, 2022 at 7:28 am

Very energetic blog, I loved that a lot. Will there be a part 2?

Reply

Kent Pitman

December 24, 2022 at 9:04 am

Good essay.

It is lamentable in the extreme that after decades of programming, career programmers of excellent caliber are left reporting to first-level line managers, who often treat them with disdain as people who must somehow be inferior, as illustrated by their position in the reporting tree.

A manager's day job is to manage people and coordinate schedules, but nothing about that makes them uniquely placed to assess strategic direction any more than a programmer could. Programmers are given leadership roles, just not leadership titles.

The “agile” process compounds this 1000% by making programmers bid for work, as if any old job could be done by any old person. In a quest to robustify skills, there’s a pressure to declare that any programmer is as good as any other that further undercuts the dignity of programmers and robs them of any ability to feel like they have special knowledge or insight.

The pressure for the “full stack” developer is yet another nail in the coffin of programmer dignity, the skill of good UX work being so very different than the skills of robust application-implementation-level work. Although I feel equally bad about the hiring requirement of intermediate managers as “hands-on managers”. In small startups, these designations may be essential to get things going, but in robust companies, each seems more a desire for a twofer, a way to avoid paying for two distinct roles to be what they are good at, than a good theory of what makes solid product.

Programmers must profess mastery of all things, must be subject to endless competitions and treadmills, and yet at the end of the day never be recognized as being important enough to include in strategic conversation, often left to be second-guessed by people with far less experience but better positioning in a reporting tree.

So of course they become managers, not because it makes best use of their skills, but because it finally says “I am a person”.

None of this makes sense. Let programmers be programmers and managers be managers. Those are different skills. But they all need a seat at the table in strategy meetings. With rank and pay should come the respect to sit alongside those of peer rank and pay. And if senior programmers are not the peers of senior managers, something in the pay or rank structure needs reconsidering.

I think the summing it up as a prejudice is a good one, and like all prejudices, it comes with the cost of lost opportunity. We have yet to see the public resonate fully to nerds in public office either. The world is full of complex systems problems that require the kinds of skills programmers have, the ability to understand how parts interact and how to do complex refactorings. It’s a pity there isn’t a programmer or two seated in Presidential cabinets. Public policy is a lot like programming.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

[Contact Us](#)

[Terms and Conditions](#)

[Privacy Policy](#)

