



Essays, opinions, and advice on the act of computer programming from Stack Overflow.



[Latest](#) [Newsletter](#) [Podcast](#) [Company](#)

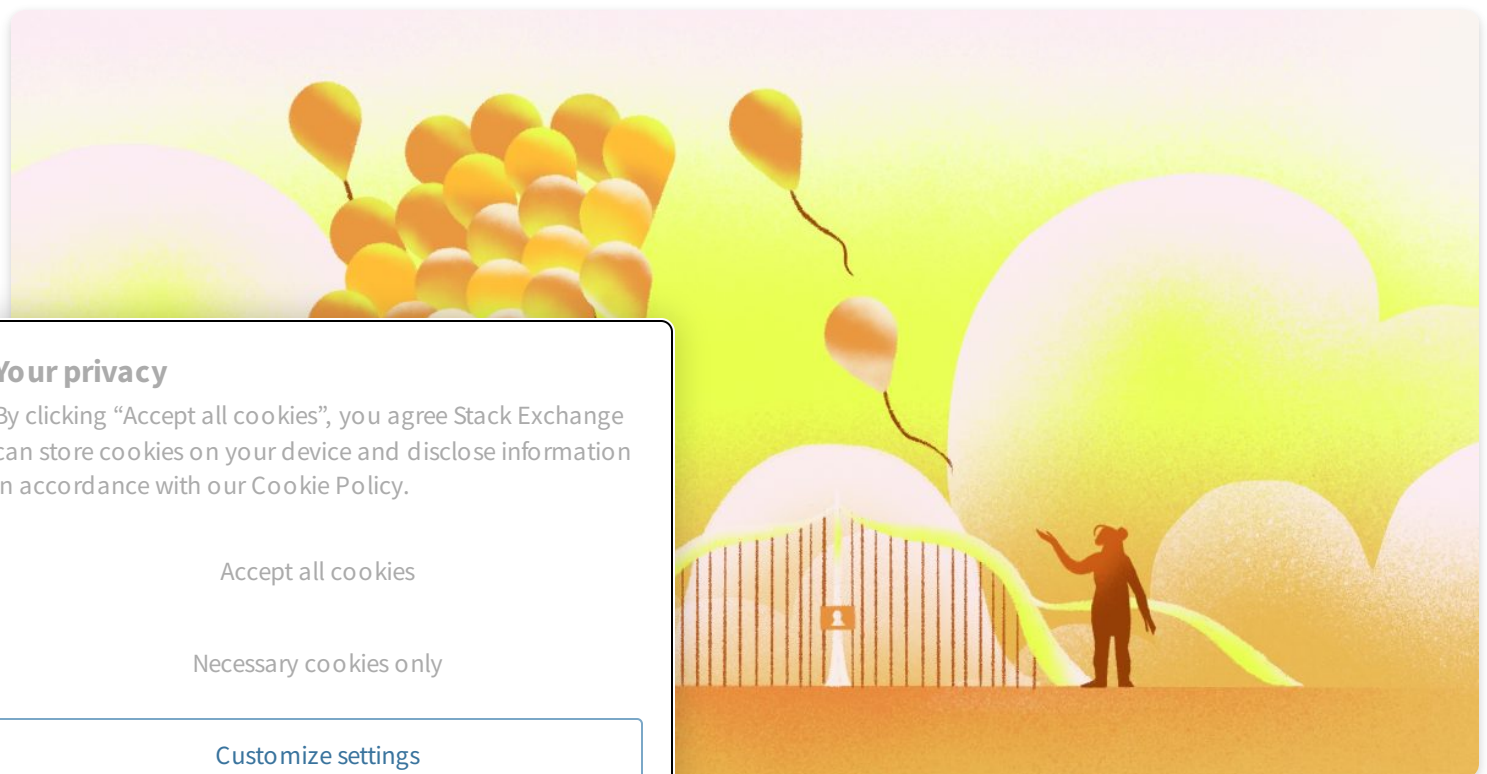
code-for-a-living FEBRUARY 23, 2022

What you give up when moving into engineering management

Moving into a management role may be a rewarding step in your career, but you should know about the things you're leaving behind.



Karl Hughes



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.

Accept all cookies

Necessary cookies only

Customize settings

My wife recently took a management position for the first time. While [my first management role](#) was much different in many ways (I was with a small startup, and she works for a state healthcare system), her mixed feelings reminded me of the roller-coaster ride I felt when I was first put in charge of a team of engineers.

On the one hand, I was excited. This was an opportunity to make a bigger impact on the organization and my teammates' careers.

On the other hand, I was scared. Many people on the team were older and more experienced, and it felt like my whole career was riding on my performance in that first year. To add to the nervousness, there wasn't any formal management training or in-house mentorship program because the company was so small. Like most new managers, I figured most things out the hard way, but it took me at least a year to feel comfortable in the new role.

The reality is that moving from an individual contributor (IC) role to a management one is hard, and it requires you to make sacrifices and step outside your comfort zone. There's a reason that some engineers hesitate to accept or pursue management roles, even when part of them *wants* that role. In this post, I'll explore what you give up when you move into engineering management. In addition to my own experience, I've included input from a few other professionals who have undergone this transition.

If you're considering a management path, use this as an opportunity to take stock of your skills and strengths to make sure you're not giving up everything you love about software development just to be in charge.

Why is the transition to management so hard?

[Adjusting to management](#) is difficult because **the skills and metrics required for management success are completely different from those for individual contributors.**

[“Successful people become great leaders when they learn to shift the focus from themselves to others.”](#) — [Simon Sinek, What Got You Here Won't Get You There](#)

Your privacy

By clicking “Accept all cookies”, you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

...the most code, solve the hardest technical problems, or ...e for ensuring that *your team* can do these things. It's ...ent because they *like* being deeply focused on ...and out of a dozen meetings every day.

I'm part of a [CTO group in Chicago](#), and one of the most common challenges we discuss is how to let go of our technical responsibilities and embrace the people-focused roles we now have. Many group members struggle with this because they built the early version of their app single-handedly. It's tough to let a bunch of other engineers with different backgrounds, opinions, and skill levels jump in and start messing with your codebase.

But as you'll see, giving things up is an essential part of becoming a manager.

What engineers give up by becoming managers

Far too many engineers take a management role without understanding the tradeoffs, and many end up unhappy because of it. If you want to become a manager, be aware that you'll have to forgo some of what you probably like most about your field.

1. Focus time

Because [context switching](#) is so costly, most engineers crave focus time. [Zeke Nierenberg](#), a former CTO who transitioned back to an individual contributor role last year, told me that giving up this focus time was a huge sacrifice: "When you're in a management role, you need to prioritize high availability. For me, that meant that my day (and mind) became profoundly fragmented."

I felt the same way when I was an engineering manager. I enjoyed spending hours lost in a tough problem and coming out the other side with a complete solution. Unfortunately, I also needed to unblock my team so that progress wouldn't grind to a halt whenever I decided to work on a new feature. Eventually, I figured out that I couldn't take on projects in the "critical path," so I'd only work on low-priority or experimental features if I had time.

While managers can (and should) [carve out time to do focused work](#), doing so becomes increasingly difficult as your team grows. Not only do engineers on the team have more questions for you, but so do external stakeholders. As the team's manager, you need to be the first line of defense so that minute questions don't interrupt your team's workflow.

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

seeing changes on a website being used by thousands of the reasons I enjoyed learning to code, and its absence is der. Management doesn't yield instantaneous results; expecting fruit the next day.

[ce to your team or implement a new policy that will take weeks to surface as good or bad. The feedback loop on management is far longer than that](#)

Gergely Orosz reiterates this point and cites long feedback cycles as the reason he came up with this [checklist for first-time engineering managers](#) unsure of how to measure their success in the new role:

“When you’re an engineer, you get feedback on your code, on your design documents, or how your projects are going. As a manager, you have none of this. There are also often no clear and specific expectations of what you should be doing in this role.”

“Success” is also a lot murkier as a manager. [Anthony Garone](#), a former Engineering Director at Allstate, pointed out that managers rarely get the satisfaction of solving tactical problems every day: “Leadership is all about handling ambiguity and creating long-term gains, which often go unrecognized.”

3. Conflict avoidance

As an individual contributor, you might look at a colleague who’s not pulling their weight and think, “I really wish I could fire that person,” but let me tell you, it’s not fun. As a manager, you are the one who has to actually deliver that bad news.

I will never forget the first time I had to put someone on an improvement plan and eventually let them go. I put the decision off for weeks, trying to find ways that the employee was improving so that I could justify keeping them. Eventually, my boss had enough, and I had to let them go. It was the most difficult conversation I’ve ever had.

Your privacy

By clicking “Accept all cookies”, you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

versations is a huge part of being a manager. “They don’t are profoundly consequential,” he told me. “What you say en sprints.”

about work performance either. While some work , Jefford pointed out that professional relationships are e someone closed off because they are mad at you or someone crying in your office because of something going

At the same time, if you're willing to lean into these interpersonal issues, you might enjoy this part of management. Jefford noted that because people are so dynamic and diverse, no "management formula" always works. There's always something new to learn.

4. Making technical decisions

Many engineering leaders are promoted to management positions after they demonstrate a certain level of technical expertise. The problem is that managers *should not* be the ones making the difficult technical decisions, especially if they want their teammates to grow and stick around.

"I struggled with letting tech things go. In the first several months [as a manager], I found myself wanting to look at PRs and have my team walk me through the code being pushed. I wanted to make decisions on method names and architecture." – Tyler Jefford

If you keep thinking of the code as "yours" after moving into management, you'll prevent others on the team from taking ownership of it. This will be extremely demoralizing and frustrating for your team, as anyone who's had a micromanaging boss can tell you.

Igor Khoroshilov, VP of Engineering at SOCi, told me that the solution is to think differently about your role: "Your product as a manager is people. Your main goal is to scale through developing your team and setting them up for success when solving problems."

But for many engineers, it's not easy to make this mental shift. Making technical decisions is interesting, fun, and (for many) comfortable. Sitting back and allowing other engineers to make changes to the code that you might think are inferior to the changes *you* would make isn't comfortable, but it is necessary.

5. Learning new technical skills

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.

you'll have less time to devote to acquiring new tech skills. Many hours per week to devote to learning new things, it's on technical topics.

main pretty hands-on, many do not continue to code. likely be expected to spend a lot more time conducting s, holding meetings with product managers, and

interviewing potential engineering candidates...Engineers who pride themselves on the depth of their technical expertise may find this especially challenging.” – Jennifer Fu, Codementor

I tried to stay hands-on with the latest technology for a while after being promoted to a management role. I’d spend a few hours every week on side projects so I could keep my hands on new technology or work on open-source projects with friends. But my life outside of work got more complicated as I entered my 30s. I got married, got a dog, and had a baby. Eventually, I had to let go of coding on nights and weekends.

As an engineering manager, you should see the broader technology trends in your industry and understand the tradeoffs of different solutions. You should be able to hop into the codebase and lend a hand, but you won’t be able to get hands-on with every new framework and tool your team wants to try.

What if you don’t like the tradeoffs you made?

If you take a management role and it turns out that it’s not for you, don’t worry. I’ve known many [engineers who moved into management and then back into individual contributor roles](#) without much resistance.

“Anyone can be a leader, even without the title,” Jefford told me. Informal leaders might be just as important to a team’s success as the manager. There are also hybrid roles (e.g., “[lead engineer](#)” or “[team lead](#)”) at many companies that come with a mix of management and hands-on responsibilities.

On the other hand, maybe you just need more mentorship, training, or coaching. Assuming there are aspects of your management role that you enjoy and feel confident in, it might be worth talking through your challenges with somebody, whether that’s a professional coach, a mentor, or just a friend with a fresh perspective.

If you’re at a large organization, you might have access to in-house management training or mentorship or outside resources as well. “You need someone outside of you have blind spots, and a coach can be a side-mirror?”

my first management role, but I’m glad I stuck it out. me to make a larger impact on my company and team. is easily as rewarding as checking in a few hundred new er opportunities as well.

management and you have something to share, I’d love to [twitter](#) to continue the conversation.

Your privacy

By clicking “Accept all cookies”, you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).



The Stack Overflow Podcast is a weekly conversation about working in software development, learning to code, and the art and culture of computer programming.



Related

Your privacy

By clicking “Accept all cookies”, you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

es. Cities must change to survive

if your office is where you live now, would you live in your old office?

27 Comments

TJ

24 Feb 22 at 4:25

While not a managerial role, I had to act as a liaison between our development team and a 3rd party COTS provider for integrating our systems that I knew virtually nothing about since I was very new to the team. They always asked me all kinds of questions and I felt dumb and would have to just turn around and ask our developers who gave me answers I couldn't understand, and that the 3rd party subsequently asked me about, and on and on...

traumatized me hard and fast to forever want to be a developer so that I'm the expert on things that I have to talk about.

Now I get to watch all the manager types try to talk about things they don't understand and ask us developers for answers that they just pass on, etc. It's all such a tremendous waste of money.

Reply

Yusuf

27 Feb 22 at 12:19

In that role were you a developer or a product manager? What do you think is the solution to that problem?

Reply

TJ

28 Feb 22 at 1:53

That's a great point. I was a developer (technically) although the -wrong- developer because I was fresh and I was brand new to the team. Everyone else had been

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.

problem is to have developers do more managerial tasks etc) or at least only hire non-developers who have a

2 Mar 22 at 7:31

Sounds like that organization has an unannounced agenda of putting fresh faces into leadership roles. It makes a lot of sense to do it at that level to determine if someone has natural leadership skills or thrives in the trenches. So much better than pulling a top performer out of the trenches into management only to see them struggle and never recover their prior level of contribution and career satisfaction.

If I may be so bold as to offer a counter to your solution...What I think most people miss is that everyone is part of the solution. A good manager will both lead the team and allow the team to lead her/him. It's easy to sit back and criticize the shortcomings of the efforts to others. That same effort could be used to ask that manager a simple question: What additional information do you need to convey that up/downstream? You'll be amazed how quickly that approach changes working relationships and culture. But, you have to have a good manager who recognizes the value of leadership at all levels.

Reply

Af

3 Mar 22 at 3:20

@Matt I liked your answer. I get excited thinking a company would afford that opportunity to someone not yet at that level instead of same old same old. @tj It would be great if devs are exposed, allowed respect and space to own the product but having said that they should be aware of roles and responsibilities. have seen devs that think they are product owners, scrum masters and support etc that doesn't work well if they overstep someone instead of complementing. I am the global portfolio lead and manager for over 9 teams and go into conversations often when I dont know the stack as well as either side asking the questions or requiring plans. I however facilitate, communicate and collaborate to get the info about what, how things need to be done. This people and facilitation factor is what the software manager should be doing and if it is not done it becomes an empty role.

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.

25 Feb 22 at 1:09

in Baseball!"

2 Mar 22 at 9:41

Emotional intelligence is fundamental to understanding people, and therefore to leading them. If your team can feel comfortable coming to you when their spouse or child dies, they trust you. Google's Project Athena discovered that the single biggest predictor of team success is emotional safety. Check out the data.

Reply

SA

25 Feb 22 at 1:57

managers are dime a dozen... you didn't mention that once you become a manager it so much more difficult to move across organizations unlike an IC where you'll always have a job, as long as you continue to work on your craft.

Reply

Will

26 Feb 22 at 3:05

Excellent article, well written. Bookmarked for sharing (and for my own reflection, too!)

Reply

Karl Hughes

1 Mar 22 at 7:10

Thanks!

Reply

David

26 Feb 22 at 7:28

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.

s, product managers, chief engineers — apart from the r levels. And we're manufacturing stuff, not software.

2 Mar 22 at 2:01

clusive character trait called 'charisma'. You need to have d to be able to make people do what you want them to do.

And this shows early in life. I have been promoted to the role of engineering manager twice in my professional life. First time off, I felt flattered, and also rewarded after about five years of being a developer in a lively team. But the team did not feel this way, and it turned out to be quite frustrating. And in one point I have to contradict you: It is NOT easy to step back down into the role of IC. There were mumbblings like 'coward', and 'he took the easy way out'. In the end, it got so bad that I left the company. A step that I regret to this day, because I liked the company, and I liked their products (cars).

Reply

Af

3 Mar 22 at 3:25

@Aldor I totally agree. What isn't initially known is that when one is a dev they control most of what happens and when in management while it initially looks like one has power and controls things it's actually more complicated. A manager is almost TOTALLY dependent on the people to get things done. That means a lot. Charisma, control, persuasion, motivation, personalities, planning and a lot of needs to happen to gain the respect and be able to get to where it needs to be.

Reply

Muallé

2 Mar 22 at 6:57

Very helpful and comforting to know that these are common challenges most managers are facing. I really like this article.

Reply

Mark Guzewski

2 Mar 22 at 11:18

A few decades ago, under considerable pressure, I accepted a management position where I worked. Suddenly a bunch of people who were my equals were now my underlings. I didn't realize how uncomfortable that would be. As it turned out, I quickly learned to dislike all the things described in this article: for things you don't have complete control over, etc. I was in a very small team. Management was very reluctant to let that happen. Looking back over the 30 additional years as a developer, I think

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.

3 Mar 22 at 12:30

Thanks for post, I'm exactly going to change my carrier path from independent consultant (software developer, 10 years) to manager & it was actually helpful for me 😊

Reply

Lorenzo

3 Mar 22 at 2:57

I've just been asked to step to an EM position, from being a tech leader of another team (which adores me and I feel 100% comfy with, although I am not technically their manager).

That's such a hard decision to make.

Reply

Daniel Orner

3 Mar 22 at 10:38

I went the manager route for a few years, then headed back to IC with a sigh of relief. (I wrote a blog post about it then. <https://medium.com/@dmorner/from-engineer-to-manager-and-back-again-63abd4d99105>)

One thing to note is that several of the things mentioned here have less to do with *management* and more to do with *leadership*. If you're moving up to a staff or principal engineer role, you're going to have to deal with the same things managers do – more meetings, less focus time, less ownership of code. It's really letting go of *all* technical decisions that is more of a hard sell, and the difficult conversations.

(Also, not every company puts managers completely away from the tech side. At our company, the engineering managers not only manage people but also help make sure that projects are staffed correctly, meaning they have to have a good idea about the projects being done and overall what should and shouldn't happen. They aren't writing code, but they do need to understand the technical issues and considerations.)

Reply

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.

3 Mar 22 at 11:04

... a lot to understand on what things to let go of.

3 Mar 22 at 12:58

... also means that you very rarely if ever have victories that getting the daily dopamine rush from the bugs crushed or

the features elegantly implemented....you get none of that. Just a long slog of sprint planning, meetings, HR interactions, passing emails up and down the corporate hierarchy, and all the other things that fill up your days until you die.

Reply

Marco

3 Mar 22 at 1:30

Thank you for this post, every single sentence hit home with me... I've been an EM for 2.5 years now, one thing that took me over 2 years to accept was "Learning new tech skills", you can't dive deep into all the cool new things coming out, you just don't have the time. A high-level understanding of what's out there, its pros and cons are enough to guide you with decisions, you have to enable your team for success and trust in them to deliver.

It still crosses my mind some days if I would be better off as an individual contributor, especially with the demand out there for the hands-on skill, but at this point, I'm sticking to my decision, who knows what the future holds.... like you said, it's not an irreversible decision, you could always go back and code full time or a hybrid role, you'll just be rusty for a while, but it's like riding a bike.

Wishing you continued success on your management journey!

Reply

Tom

3 Mar 22 at 2:33

I've been in software development for several decades now, and each time I got drawn into manager responsibilities I worked hard to get back out. I just don't get satisfaction in all the drama. I'm a technology geek. But that didn't limit me at all in my career. I quickly rose up to architect, chief architect, enterprise architect, and so forth in startups and companies big and small. As the company chief architect now I am in an IC position where I lead through mentorship and influence in ways that managers cannot. Engineers confide in me knowing I don't control their paychecks. I've hit my sweet spot and am enjoying this experience. I feel bad for developers who think the only way to advance is to become a manager, and there is no alternative.

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

4 Mar 22 at 8:22

gineer to management. I might have a longer career than
e. My first 20 years was technical focused. I was granted
ies. Then I switched my role to management and then
ement was a natural move for my career. I even got a MBA.
nd that you could not be a hand on manager any more.
r measurement of success was not the same compared

with a engineer role. You are no longer working as an individual. You are the leader of your team. Your job has to deal with corporate politics when you move further up. You really need to ask yourself whether this is the path you really want to pursue. Nevertheless, if you are not sure, get closer and you will find it out.

Reply

Norbert

8 Mar 22 at 12:10

Thanks for this article – I'm a senior dev at this very question

Reply

Brian

11 Mar 22 at 10:21

Great article and very well written. Many of the issues I face (currently) as a manager fall pretty close to everything stated. There are days when I miss the focus time, not having a full schedule of meetings and being able to contribute with those hard to fix coding problems for the team. But there is something rewarding as well helping an engineer recognize their potential, areas to improve and what can make their career grow.

It's a complete change for many, that definitely takes time to get used to. It may not be forever, as I regularly ask myself if I miss being an IC more, but for now it's good experience.

Reply

Elias

12 Mar 22 at 6:39

Very good article and it hits the core points.

Once you become a manager, you are not a super developer that now can also make decisions at management level, but you become a totally different figure.

A figure that nurture, augment and protect his direct reports, and ensure that requests from upper management are satisfied.

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

to be told what to do, and like to be in control of what they
you may be a good soldier because you follow orders to the
r because to be a leader, you need to have traits that are
and just be told what to do.

m to implement whatever you like; which can't be farther
o use it, in which capacity and in which context. Most of the
: to implement something new, it takes quite a lot of
and 10 as manager, I noticed how things shift around you

Each has its pros and cons; and it all depend from your skillset. People that like to be IC like to get personal credit, they have a very high ego and often a low emotional intelligence score. What they aim at is the rush of dopamine coming from being faster than others, implement better code, know more.

A manager instead has a much different set used to evaluate his/her success. A manager is literally work so others may do their work in the best way possible; is the coach that set up the chessboard with the best players available in the best position they can cover, and go forward delivering top notch results with his/her team. Manage others means forget the tech part in terms of implementation, and empower others, so they can grow and advance. That is a manager achievement, and as such, you need to have a very small ego and be used to get your team success as your success. Not everyone is like that, especially among software engineers.

And you need to distinguish between the people manager, which is usually coming from a non-technical path, from the technical manager. The former is the old school manager that knows nothing of what you do and just drive process; the latter is the one that understand how hard it is to implement something, and is much more connected with the team and rooted in reality. They are much different, and if you are so unlucky to find a non-technical manager, I really feel for you, because you will suffer. There are not a lot of technical managers, but there are a ton of non-technical managers sadly; which give a bad rep to the management positions sadly, in the engineering world.

In the end, to each person their own; some are great manager, some are great IC, and some are both. Although if you never try, can't really understand what it means

Reply

mifa kijali

30 Mar 22 at 11:13

I went the manager route for a few years, then headed back to IC with a sigh of relief. (I wrote a blog post about it then. <http://nadergholami741.sitearia.ir/post-360783.html>)

One thing to note is that several of the things mentioned here have less to do with *management* and more to do with *leadership*. If you're moving up to a staff or principal engineer role, you're going to have to deal with the same things managers do – more meetings, less focus time, less ownership of code. It's really letting go of *all* technical decisions that is more of a hard sell, and the difficult conversations.

(Also, not every company puts managers completely away from the tech side. At our company, the managers also help make sure that projects are staffed correctly, that projects are being done and overall what should and shouldn't be done. They do need to understand the technical issues and

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.

Comment *

Name *

Email *

Website

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

Post Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).