



Mark's Manuscript on Management

Thoughts on engineering management from my never-to-be-published book

by Mark Kater

First edition, 2026



Contents

I Welcome to the Team

The three things I tell every person who joins my team

II The Flux Capacitor

Why every team gets a leadership triad, and why the name matters

III Teamwork & Transparency

Why psychological safety and real connection beat individual heroics

IV On Being Appreciated

Recognition isn't a nice-to-have. It's one of the job's core duties

V On Distributed Teams FORTHCOMING

Running an engineering org spread across countries and time zones



Introduction

I've been an engineering leader for over 15 years, the better part of the last decade at eero ("an Amazon company"). I joined eero when it was still a small hardware startup with no recurring revenue to speak of, and I've spent the past several years standing up the software side of the business: growing my organization from scratch to 70+ engineers across multiple countries, and taking four subscription products from an idea into one of the company's core revenue lines.

I've been fortunate to hire and work alongside genuinely great people who came together to build strong, successful teams, and it's those people who've told me for years that I should write about how I lead. I've always resisted, mostly because I just don't like the "look at me" attitude so many engineering managers have, unaware that it really isn't about them. On the flip side, maybe sharing my thoughts actually weighs heavier than appearing to "show off", and someone out there (you?) could learn or draw inspiration from it. So here we go, sharing the secret sauce.

"Mark has cultivated an exemplary team culture. His leadership style and willingness to be an equal with the team, rather than just a boss, have enabled him to achieve this. It is a rare quality that I have only observed in a few leaders during my tenure at Amazon."

— Peer, Product Leadership

These are things I genuinely value, shaped by a career that took me from a six-person startup to a company with over a million employees. Each stop in between taught me something different. Some of you will read this and disagree with parts of it. Others might wish their own manager led more like I do. Some of you might take a piece of it and make it your own. All of that's fine. Everyone develops their own style eventually. This one's mine.

manuscript (noun) / 'mæn.jə.skript/

A book or document, typically before publication or official editing.

While I don't intend to ever actually turn this into a book, I did organize things into short "chapters", each a quick read. Most of them relate to people management and building thriving, productive teams, the things I think matter more than most of what gets written about "engineering culture". I don't cover the technical side of things here, mostly because I don't think it's that interesting to write about: technology, especially in this age of AI, evolves so fast that whatever I write today might be outdated six months from now. Until robots actually do take over, the people side of being an engineering manager will hopefully stay relevant a bit longer.



Here's what's written so far. The first one is the conversation everyone who joins my team has with me, so it's the place to start; the rest stand on their own.



CHAPTER I

Welcome to the Team

The three things I tell every person who joins my team

Whenever someone joins my team, whether I hired them or whether they're a skip-level, I always have a casual “welcome to the team” meeting. Keyword: casual. During this first meeting we don't go over the tech stack, or the roadmap. No, the first meeting is to set the tone: we have an amazing team of smart people and we're excited to have you join. In order to keep the team and its culture amazing, I request three things:

- **Take your job seriously. Don't take yourself too seriously.**
- **Speak up. Good or bad.**
- **Have some fun.**
- **Take your job seriously. Don't take yourself too seriously.**

The best engineers I've worked with are the ones who care deeply about their craft and can still laugh at themselves. That combination is rarer than it sounds, but I'm lucky enough to have many of them on my team.

And please don't take me too seriously either. For most engineers on my team, I'm the manager of their manager, and people are just wired to feel some level of intimidation by hierarchy, whether you want to or not (and clearly I do not).

I use my monthly “all hands” meetings to work against it. During these meetings we cover team updates and demo new functionality. You know how when you're on hold, there's the “waiting music”? Well at some point someone proposed we should have waiting music, while we “wait for folks to trickle in”. Great idea! For the past few years, we open each all hands with a random song I pick from YouTube. It sets the tone right away: let's just have a bit of fun and let's not take ourselves too seriously. We even have a playlist of all the songs on Spotify! :)

At the end of each meeting, the final slide opens it up for “Questions, comments, or silly jokes”. It has a dad joke that I've pulled from somewhere, which is often guaranteed to get a few eyerolls. But these things are deliberate and important! I want people to feel comfortable, see me as the leader, but also as a regular person who isn't afraid to laugh at himself and who is approachable.



“Mark is the kind of leader who makes work feel less like work. He's smart, supportive, funny, approachable, and just a great person to have leading the team.”

– Direct Report

The point of all this: there's no room for egos on the team. “We win as a team and we lose as a team” is a cliché, but it's true, and it only holds up if nobody is too invested in being right. Not taking yourself too seriously is what lets you own up to a mistake instead of reaching for something, or someone, to blame. And that's a lot easier when the person above you goes first.

● **Speak up. Good or bad.**

I encourage people to tell me what they want to work on. What excites them. What they want to grow into. What kind of work they want to do more of. Sure, sometimes you'll be told what to work on, but let's make sure it's a 2-way street. People who work on things that excite them simply do better work.

The flip side is equally important: tell me when something isn't working. But here's the thing: speaking up isn't easy. It requires trust. And trust has to be earned.

The way I earn it is through transparency. Some call it my Dutch bluntness, because I grew up in the Netherlands. When something isn't working, I call it out and make it clear we're going to try to fix it. The message is always the same: it's fine to say what sucks, as long as we stay constructive about it. We see it, we name it, we fix it.

There's a specific reaction I've come to expect the first time a new team member hears me say, out loud, that something sucks. You can almost see the thought forming: “Wait, aren't you leadership? Aren't you supposed to tell us everything is exactly how we planned it?” That moment of surprise is usually when the trust starts.

“He stays true to deadlines and calls out things he doesn't agree with, which really earns trust. He's the kind of leader you want on your side when you're working on big, change-the-paradigm initiatives.”

– Peer, Product Leadership

When they see me call something out openly, it signals that it's safe for them to do the same. As most managers, I was an engineer before this. And what I learned was that venting to your coworkers about things that aren't going well feels good for about five minutes and often changes nothing. Bringing it to your manager is its own kind of scary. I get that. So I try hard to build an environment where people feel safe saying the thing out loud. There's a third thing, though, that ties the first two together.



● Have some fun.

Yes, we have plenty of serious conversations. But there should also be room to be direct and poke a little fun at each other. To tell the person who's consistently longwinded that they need to get to the point already. Or to ask your Brazilian coworker how their soccer team did, knowing full well they already lost. You have a laugh. It creates a bond. It makes the team real.

That bond matters most when things aren't going well. When you need to stay late to fix an incident, it's a lot easier to grind through the hard stuff when you actually like the people next to you.

If you enjoy what you work on and have an intrinsic sense of motivation, you'll do your best work. So I try to give people work they actually want to do, not just work that needs doing. When someone's genuinely into a project, it shows: the pace picks up, the quality goes up, and I don't need to hover over them checking in. My job at that point is mostly to get out of the way.

"He lets people work on what they enjoy most and where they have the most potential to thrive, while still holding us to high standards."

— Direct Report, Engineering

It works!

Culture isn't a set of values on a wall. It's the specific things you say and do when new people arrive. If you're consistent, it compounds.

What surprised me was how quickly a team culture spreads by reputation. When I inherited a team of 11 engineers from a different part of the org, several of them told me they'd already heard about my team's culture and were excited to finally be part of it. I hadn't advertised anything.

But I guess it's not complicated: we treat people like grownups, and word gets around. Voluntary attrition across my whole org has stayed consistently low, well below the industry norm. Meanwhile, I regularly get approached by engineers from other teams, asking if there's a possibility to switch teams to my org. What better compliment than that!?

"Under his leadership, the team has expanded in both charter and size while maintaining velocity and quality."

— Peer



CHAPTER II

The Flux Capacitor

Why every team gets a leadership triad, and why the name matters

In the early days of starting the team at eero, its leadership structure was very straightforward: our head of product, our most senior engineer, and myself. Three people, complementary skills, fast decisions. Where a lot of teams have some level of contention especially between product and engineering, we had deliberate shared ownership instead, and it worked really well.

As the org grew into what became the Software Services organization, spanning multiple teams and dozens of engineers across several countries, we started bringing on new leaders and standing up new teams from scratch. Rather than reinvent the leadership structure each time, I modeled every new team after what had already worked so well in that first small one: an engineering manager, a product manager, and one or more senior engineers, all with real, shared ownership. The senior engineers weren't just there for technical input either. They got a formal voice in roadmap planning, in raising concerns about the team, in keeping decisions grounded in what was actually happening on the ground, not just what looked good in a planning doc.

As for the name: I don't like to take things too seriously, myself included. Given that the flux capacitor is shaped like a triad (go watch *Back to the Future* if you don't get the reference), I decided that's how we should label our teams' leadership groups: the Flux Capacitor. To this day, each of my teams has a Flux Capacitor meeting on their calendars.

The structure landed well. Engineering managers had a clearer operating model. Senior engineers had a real seat at the table, which helped both retention and the quality of roadmap decisions. And the leaders one layer below me got genuine autonomy: they weren't waiting on me for permission, they had what they needed to run their team well.

It scaled cleanly too. As we added teams, the model just replicated. Every new team knew exactly what their leadership structure looked like from day one, no reinvention required.

Good org design isn't about control. It's about giving the right people the right context and the right authority to make good decisions without you. The flux capacitor worked because it balanced three things that are usually in tension: engineering judgment, product direction, and technical ground truth. Put those three in a room together with real authority, and things move faster, and the decisions are better.

And naming it something fun didn't hurt. People remember the Flux Capacitor. Nobody remembers "leadership triad".



“The way I see it, if you're gonna build a time machine into a car, why not do it with some style?”

– Doc Brown



CHAPTER III

Teamwork & Transparency

Why psychological safety and real connection beat individual heroics

When people feel like they're actually part of a team, not just a set of individuals who happen to share a manager, they go the extra mile for each other. And just as importantly, when something goes wrong, nobody's first instinct is to figure out who to blame.

Part of building that starts with something simple: know your teammates as people, not just names on an org chart. With roughly half my team based in Brazil, I've spent time learning Portuguese myself (unsuccessfully, btw), and it's led to some great conversations along the way, from regional accents to just learning more about the country. It was more just for fun, but it signals that you care. Not an act or a strategy, in my opinion a good people manager should always care about their people, and not just about the quality of their work.

"He cultivates a culture of camaraderie where people willingly step in to help each other out even if doing so is not part of their job."

— Direct Report

I also think I'm more transparent than most engineering managers I've worked alongside. Most managers, understandably, filter what they share upward and downward: soften the bad news, hedge the uncertain stuff, wait for the "right time" to say something hard. I don't operate that way. If a decision is going to be unpopular, I say so before anyone asks. If I don't have an answer yet, I say that too, instead of pretending I do. People don't need every problem solved for them. They need to trust that you're not hiding the ball.

That combination, real connection plus real transparency, is what actually gets tested when you have to ask a team for more than seems fair.

A few years back, a partner deal got signed with a launch date nobody had cleared with me first, one that our team had no realistic way of hitting with our existing plan. Rather than downplay it, I told the team the situation was, frankly, kind of ridiculous (my Dutch bluntness showing again), and that I'd understand if they felt the same. I also told them exactly what the new plan was and what I was asking of them, and that I believed we could pull it off. We shipped one product early in a leaner form, moved everyone onto the deadline that mattered most, and hit it.

What stuck with me wasn't the deadline. One of my peers on the product side put it well afterward: "every time we find issues in solutions we've built, Mark's first response is let's fix it vs. a debate or a blame game on what caused it." That's exactly the value of transparency. I've seen plenty of managers hide things when it isn't necessary, in an attempt to look like they're in



control of the situation. And sometimes that really is the right call. But often it isn't, and you actually get more buy-in by just acknowledging it plainly: yes, someone messed up, it doesn't matter who, let's move forward and get this done, together. When a team feels like you're in it together with them, they take that extra step to make it a success.

“His attitude not only allows him to build trust with the team, but also to keep the team motivated and willing to always go the extra mile to achieve business goals.”

– Direct Report



CHAPTER IV

On Being Appreciated

Recognition isn't a nice-to-have. It's one of the job's core duties

A person who feels appreciated will always do more than what's expected of them.

Many managers think their job is to extract work out of people. I think the job is to make people want to give it. Feeling like what you do matters is incredibly powerful, and plenty of managers underestimate how valuable it is. As a manager, you play two roles here: you show genuine appreciation yourself, and you build an environment where people value each other too, not just where recognition flows down from you.

That second part comes back to everything else I've written about here: no egos, real trust, a genuine sense of team. Get those things right, and appreciation almost takes care of itself. People who trust each other and see themselves as one team don't need to be told to notice each other's work. They just do.

One nuance I actively coach my managers on: appreciation done wrong can backfire. Never single out one person, and never create awards for individuals. That person you're praising almost certainly had help, someone who unblocked them, reviewed their work at 9pm, covered for them so they could focus. Praise only the name on the ticket, and you've told everyone else who contributed that none of it counted. Real appreciation has to include the people who don't show up in the commit history.

"Mark's greatest strength, as a manager, is to take care of his direct reports and not leave a single comment unattended."

— Direct Report

I think the word "appreciation" is more interesting than people give it credit for. It doesn't just mean recognition. It also means to gain in value, the way a house or a stock appreciates. That's not a coincidence, it's exactly what happens when you show it. People become more confident that what they're doing matters, and once they feel that, they'll go looking for that feeling again.

People don't leave good teams over one bad sprint or one hard deadline. They leave because, over time, they stop believing anyone actually sees what they're doing.



Not because it's nice. Because I'm constantly aware of the importance of feeling appreciated. Of course, it does need to be real, and it's more difficult with underperformers, but even then I still believe that showing appreciation for trying to do the right thing helps get them back on track.

And this applies to any role, of course, and ideally bubbles up throughout the organization. Guess what: I could have the most productive team in the world, with people I genuinely love working with, but if I don't feel like my own leadership appreciates my work, I'm going to have a difficult time staying motivated. It's human nature.



A bit more about me

I originally moved to San Francisco about 20 years ago, from the Netherlands. Started off as a Frontend Engineer, literally turning my hobby into a job. I'm happily married and have two kids who I keep reminding: it doesn't matter what you want to do when you grow up, as long as you're passionate about it.

Being a Senior Engineering Manager at Amazon unfortunately doesn't leave a lot of time to code, but I still love to dabble in side projects. The most recent one is JukeKeeper, a web app I initially built for my dad to manage his vinyl jukebox collection, and then decided to expand to a larger audience.

One more note: all the quotes throughout these pages are real, unless they're a Back to the Future reference, of which you'll find several (a bit of a BTTF nerd, sorry). Every real quote was pulled from annual review feedback I've received over the past few years.