
ENGINEERING
MANAGEMENT
FOR THE
REST OF US
SARAH
DRASNER

PRAISE FOR ENGINEERING MANAGEMENT FOR THE REST OF US

When you pick up this book, also pick up some sticky notes, bookmarks, and notebooks! Engineering *Management for the Rest of Us* is packed with deeply valuable and practical information that will benefit new and experienced managers and teams. It's rare to find a management book that is compelling, funny, and useful all at once, but Sarah Drasner has nailed it.

—Cassidy Williams

Sarah has written a fantastic modern manual for aspiring and existing technical managers. It provides an excellent hands-on tour of how to navigate prioritization, execution, collaboration and culture in a very approachable way. This book is filled with pragmatism, real-world examples and plenty of engineering management nuggets of wisdom. We can all learn from Sarah's experience and I have no doubt you'll learn something new that you can apply to your own career and team.

—Addy Osmani

When you read between the lines of *Engineering Management for the Rest of Us,* you can feel how empathetic and passionate Sarah is as a leader and how much she believes in the fact that you are cut out for it, too. Regardless of your background you can lead a team and this is a practical book on how to do so.

—Christian Nwamba

Much has been written about how to effectively manage. Jam-packed with insights from an experienced leader, *Engineering Management for the Rest of Us* brings something new to the canon. Sarah shares actionable tips on building trust and psychological safety, and how to scope down PRs. She also offers sage advice on the mindset shifts managers need to make. This is the book every new engineering manager needs.

—Suzan Bond

Engineering Management for the Rest of Us

Sarah Drasner

ISBNs: 979-8-9867693-0-1 (hc); 979-8-9867693-1-8 (pbk); 979-8-9867693-2-5 (ebk)

Library of Congress Control Number: 2022915051

Skill Recordings Inc.
https://skillrecordings.com

CONTENTS

FOREWORD

I'd just become a new manager, and within the first few months I had my first crisis. One of my direct reports had gotten into an altercation and was extremely upset. They demanded that I fix the situation immediately. Having spent my entire career developing software, I was at a loss for how to develop people, and more urgently how to develop an action plan to address their immediate concern. I called my friend Sarah Drasner in a panic, and she calmly walked me through a plan that worked perfectly.

You see, before going into management, I'd reached the level of principal engineer and was starting to contribute more to defining business goals and strategies for the company. At this point in my career, my talents would be better suited guiding other engineers than continuing to write code myself. So I was promoted into management and, like many other newly promoted engineering managers, I was unprepared.

Fortunately, I had Sarah to lean on. This was not the last time I called her for advice or asked her to walk me through how to deal with a situation. Sarah, in human form, was quite literally my handbook to people management. And I must say, she's brilliant at it. I'm probably the reason Sarah wrote this book (lol). Perhaps she'd had enough of my late-night calls, stressing over what to do to ensure I don't totally mess up someone's career.

Since entering the world of engineering management, quite a few things have changed. I have received formal training, grown my teams, and been promoted multiple times. What remains the same are my frequent phone calls to Sarah to bounce ideas off her and get her words of wisdom. She's always open and willing to help, and gives spot-on actionable advice that has helped me grow as a leader.

I'm absolutely thrilled that Sarah decided to encapsulate her wisdom in *Engineering Management for the Rest of Us,* because I can't imagine how new engineering managers would be able to navigate this world without it.

This book captures all of the essentials including career laddering, earning trust, managing conflict, and so much more. It's filled with relatable stories and scenarios to add lively context to the lessons. It prompts thoughtfulness and self-reflection on how you're leading your own teams. Sarah is very careful to preface all her advice with disclaimers about how it's just her take; but trust me, it works across the board.

Engineering Management for the Rest of Us reads as a support group for those of us who are trying to do our best with our newfound powers. Sarah doesn't shy away from sharing her own mistakes, how she overcame them, and the tools, systems, and processes she created to do so.

Sarah, thank you for taking the time to write this amazing guide to engineering management.

Signed,

The Rest of Us

—Angie Jones, Vice President

INTRODUCTION

I'm not the best manager in the world. I've had great teams and great moments, but I've never considered being a manager part of my DNA. I wasn't assigned my first lead position because I was the strongest communicator on the team, but rather because I had the most seniority and knew how to get the project over the line. Being asked to lead at that time had more to do with my engineering skills than it did with actual leadership, which I felt thrown into headfirst.

Many of us started as engineers and were either promoted or moved laterally into management. Sometimes this happened because we were good at motivating a team, sometimes it happened because we showed leadership in strategy, but many didn't necessarily choose this job in management. It chose us.

I've noticed that in my field, software engineering, many people share posts about technical implementations and very few about engineering management.

Management is still related to the code, though. Unless our teams are set up well, have support, and have clear strategy, all the coding best practices and linters in the world won't amount to real outcomes. Management impacts so much of software engineering, so it's worth taking the time to learn and share. **We owe our teams thoughtful leadership.**

I wrote this in the hope that by encapsulating some of the lessons I've learned, this may help others begin with resources and avoid struggles I have had. That said, management isn't one size fits all and some of what I discuss in this book may not be right for your particular situation. Please use your best judgment and find your own path. What I present here are tools for your toolbox and not the only way to approach these subjects.

Though I'm sharing what I've learned so far, I know that I'm not done learning. I don't have it all figured out, and I'm not done making mistakes. I'm still on this journey.

It can be tough for those of us who didn't go into engineering with the distinct goal of becoming managers, but who still want to do our best to support our teams. This book isn't for the "born leaders." This book is for the rest of us.

PART 1

CHAPTER 1

CARING FOR YOUR TEAM

There's a joke in engineering management: "You become really good at bridges . . . so they promote you to be a baker." It doesn't necessarily feel like the type of skills that got you where you are in engineering work will help you on the management path ahead.

Leadership is challenging: where your work used to be about you and what value you brought to a team, **your work is now about enabling everyone around you**. This entails interruption-driven work so that your team can be flow-driven. This can involve a lot of collaboration and communication, which feels strange because it can at times be incredibly isolating as well.

However, each person brings many things to the table as a manager with an engineering background. You've done the work, which makes it easier to understand the tasks at hand. You can use your experience to effect strategy and support your team members (regardless of level) by making sure they understand why their work is important, and what their career trajectory may look like. This is unique to each individual.

The stakes are high with engineering management. Previously, if your code broke, it impacted people indirectly. Now your job affects people's daily lives in a tangible way that you see reflected back on the faces in front of you. Engineering management requires that you understand power imbalances, people structures, and consider strategies that are outside one particular project. For those who may feel mild panic after reading that, **that's actually a good sign**. I would be surprised if you didn't.

My friend Ashley Willis once said:

"The fact that you're worried that you're not a good manager is a key part of being a good manager."

What she means by that is, you should take this job seriously. "Bad" managers are often flippant about the role and the consequences of it. **Caring is vital to doing this job well.**

In PART 1: *Your Team*, I talk through some tools for building productive and driven engineering teams, as your team is the most important part of your work.

CHAPTER 2

THE VALUE OF VALUES

Working with teams made more sense to me when I started to learn about values. People are not pure functions; they have all sorts of interesting side effects. For those just entering management, thinking through values can help provide some clarity. My coach, Jessi Kovalik, spent a lot of time teaching me about this concept when we began working together. I can't overstate how much this fundamental concept helped me have a more solid framework for working with groups of people.

I refer to values a lot throughout this book, and how you can apply some of the base thinking in this chapter. Values work is a tool: it gives us some rubric to understanding where a person might be coming from.

Working to understand values isn't a silver bullet to solving every problem, but it affords us a deeper sense of where to start when building understanding and trust within a group. Unpacking a person's values with them can also help us understand what drives and motivates them. And, the core of working well together is mutual understanding.

INDIVIDUAL VALUES

So . . . what are values?

Values are the fundamental beliefs that guide us, motivate us, and drive our actions. Values describe the qualities we want most to embody. They help us determine what is important, help us understand what we essentially align with. They help us determine what kind of person we want to be.

Individual values can be related to ethics and what we think is moral. Cultural values can also reflect context that's greater than an individual, and relate to what's important to a larger group.

If you pay attention, you can see how a person's values dictate their behavior and ethics. Your values can be formed at a young age, and they can be a reaction to events. They also evolve over time.

Here is a sample of some words that can express values, though this is a condensed list. You may find other values lists that are more comprehensive. Perhaps you may notice a few in here that resonate with you:

Accountability, Advocacy, Autonomy, Compassion, Collaboration, Contribution, Creativity, Curiosity, Dependability, Diversity, Empathy, Ethics, Excellence, Fairness, Family, Friendships, Fun, Growth, Happiness, Health, Honesty, Humility, Humor, Inclusiveness, Independence, Knowledge, Performance, Personal Development, Spirituality, Perfection, Power, Preparedness, Reliability, Success, Teamwork, Traditionalism, Trustworthiness, Versatility, Vision, Warmth, Wealth.

Take a minute or two to examine this list. Which three words resonated with you most? For those values you noticed yourself aligning with, think about why you picked them. Was it a value your parents drove home for you as a kid? Did you overcome adversity and come out the other side, understanding the need for one of these values? Are there some that just feel kind of intrinsic to you, and you're not sure why? Hold that thought.

VALUES AS CONTEXT

It's equally important to think through values as a team. **Values don't offer something to fix, or an action to take; they provide us context so that we can be more understanding of what is happening and why.**

Movies sometimes present problems of conflict in a dangerously simplified way: It's the forces of good versus the forces of evil. Unfortunately, folks can sometimes use that framing when conflicts arise at work, though it's rarely the case. More often, conflicts are the result of a misalignment of values, and neither person is attempting malice against the other. If their needs aren't met and they aren't communicating in a way that someone else with dissimilar values can understand, that may be detrimental to their relationship.

Doing work to understand values as a team can help members see one another as people, and understand where they are coming from. **The act of sharing values can also have a side effect of building trust and vulnerability on a team.**

One way to run a values exercise is to give team members a few minutes to pick five values from a list like what we just saw. Once everyone has picked, go around and ask each individual to discuss why those five values resonate with them. If the group is shy, it may help if the leader goes first.

Seeing the values people choose can be illuminating. Working with individuals for some time often leads to an *aha moment* when you understand a bit more about what drives them and why they work the way they do.

If you don't feel like there's trust on a team yet, and the concept of doing an exercise like this sounds uncomfortable, I completely understand and relate. **However, it's actually most valuable to do something like this when it is uncomfortable.** There have been so many times when I felt like group members weren't vulnerable with one another or understanding each other very well, and doing an exercise like this really helped.

If you can swing it, I suggest doing this values exercise with your peers as well. You need to not only work together across disciplines and departments, but also understand each other in order for the company to operate cohesively as a whole.

I once worked on a leadership team where we did personality tests that are in some ways similar to the values exercise. We found that another person and I (who sometimes didn't swim in the same direction) had the exact same personality type backed up by the same values. When we realized this, a lot clicked for us. We were both trying to be logical and data-driven, but working toward slightly different objectives and with different data!

We reflected on this together and chuckled a little. From then on, collaborating made more sense as we tried to understand each other's motivations by practicing the values we knew we had in common.

DIVING IN FURTHER

As mentioned before, understanding values in practice isn't a silver bullet to solving a problem. Values provide context to a person's mental state, needs, and motivations. In practice, understanding the values of your team members becomes a useful tool to evaluate and drive your own empathy. You may have different values, so unless you actively seek to understand another person's point of view, it's possible to be bound by your own context and limitations.

The examples I'll discuss next show how thinking through values within a situation can give a little perspective. This is especially important when no one is right or wrong, but when you could use some insight to navigate the conversation.

Example one:

> *Suzy grew up in a house with nine siblings, where sharing food and chatting at dinner was very much a part of how everyone related to each other. Her parents valued happy messes. Rashid grew up as an only child, and his parents valued order and discipline at the dinner table. At a company offsite with "family style" dinners, Suzy reaches over Rashid for some pasta without asking. Rashid finds this rude, and Suzy doesn't think it's a big deal.*

Can you imagine how their individual backgrounds and values might play into this disagreement?

Now when a disagreement comes up, or even when you just sense slight tension, you have a tool for understanding where people are coming from. It's not that anyone in the situation has bad intentions, but rather they have slightly different backgrounds and values. You can also see that if they become aware of the differences in their backgrounds and values, they might be a bit more patient with one another as well.

Example two:

Our values can change over the course of our lifetime. You and I probably don't have the same values today that we did when we were teenagers. What types of experiences have expanded or altered your values over time?

> *Amina grew up hating school. Her high school teachers didn't really care about her education, and as such she didn't really find any joy in learning. However, as she went to college and then grad school, and had more dedicated teachers that taught subjects she elected to take out of interest, her love of learning grew.*

Can you understand how, if you asked Amina if one of her values was learning when she was fifteen years old, and then when she was twenty-five years old, you may get different answers?

Perhaps this personal evolution is meaningful to Amina, and now she wants to be on projects where she can learn as much as she can. Perhaps learning is what's most important to her about her own career growth. That's something you could help facilitate as her manager if you know the full context.

What life events have the people you work with been through that changed the way they think about things? What can you learn about how they've evolved as people by understanding what shaped their values?

Example three:

People can be internally inconsistent. Sometimes it seems that someone values one thing, but they act another way instead. For those around them, this can be disorienting. When you see miscommunication or conflict arise, perhaps dig in to whether this may be the source.

Freddy is very sensitive. He severely dislikes when people judge him and is a bit paranoid that people might be speaking ill of him behind his back. He appeals to his coworkers by asking that they not do this. However, whenever Freddy goes out after work with coworkers, he consistently speaks ill of whoever is not there.

Do you see how, even if Freddy picked integrity as a core value, he might not always act in ways that are congruent with this value? However, this doesn't mean you should call this out. The values workshop is a tool to understand one another better, and not to be weaponized.

When someone is acting in misalignment with their stated values, this is precisely when to be careful. In his book, *Nonviolent Communication*, Marshall B. Rosenberg points out that "All criticism, attack, insults, and judgments vanish when we focus attention on hearing the feelings and needs behind a message."

In this context, there may be a need that Freddy has that's not being met. Rather than responding in an accusatory manner, this is a time to show curiosity about what that need might be. It can also be hard to hear that our personal self-image doesn't align with what others are experiencing of us. We cover this in more depth in the chapters on feedback. Understanding what Freddy *wants* to value is still important though!

YOUR OWN VALUES

We've talked a bit about the values of the folks that surround us, but we should also consider our own values, because they do impact our leadership style. The way you lead a team doesn't need to be the same as other leaders. It can be tempting to "use someone else's map," but ultimately we are each individuals, and we have our own lived experiences, values, and boundaries. It's important to pay attention to these and communicate them to others.

A lot of leadership is selfless, so we may sometimes be tempted to think our own needs take a backseat here. They do to an extent, but you can't care for others if you don't put the oxygen mask on yourself first, so to speak. Communicating our own values also gives our employees a bit of context about our working style and needs, which they'll need in order to best understand us. If you don't have alignment with your values at a base level, you will eventually burn out because, let's face it, management work is very tough. You matter too.

Unpacking our own values allows us to notice patterns in our life where we felt very aligned, and to see patterns where things didn't feel right. That "not feeling right" feeling can be a good indicator of things we need to address. Values misalignment can impact our morale long-term and even lead to burnout if not addressed for a long time.

We can also think about our own values in terms of how we react to other people. If you find someone interesting or, conversely, you don't like how someone communicates, perhaps consider how that is related to what you value?

You may start to notice the interplay between your values and your leadership style. Do you value humor and learning? You may find you start standups with a "dad joke" and something you've been pondering. Do you value family? You may find you put all of your effort into making sure parental leave at your company is equitable for all. Both of these are mine, so of course one of my favorite YouTube videos is this bizarre one by Simone Giertz: "A Drone that Carries Babies."[1]

We can also see how our own values play out in times of conflict. If one of your values is equality, and you find yourself feeling like you need to stand up when an issue around equality arises for your staff, it can be helpful to understand your own motivations. Likewise, if this *isn't* a value of someone you're in conflict with, what is their value structure? How might you communicate in a way that feels true to each of you, so you can overcome the issue and unblock the project?

Understanding the correlation between your values and what you care about as a leader—what you will and will not stand for—can be a great exercise to understand your own motivations, needs, and boundaries. This can help you manage your time as well, as we cover later in this book.

COMPANY VALUES

Companies also have values (explicit or not), and it's pretty crucial that those values align at least in part with your own. There may be printed values in your company handbook that everyone knows, but you can also see expressions of those values that are unwritten and that exist subtly in priorities, communication, and even the way the company organizes itself.

I admit that I used to think company values were hogwash. They felt a little cultish to me, and I couldn't care less about agreeing with or repeating some weird statement like a robot with a huge group of people. It wasn't until pretty recently that I came to see that, when done well at an organizational level, core values help companies to determine their goals by creating a stable direction forward. The skepticism I held was whenever those values seemed empty and didn't match with my reality. When values are aligned well, they are actually a system you can check against.

My former coworker Lauren Sell, who was the vice president of marketing at Netlify, and before that OpenStack and Rackspace, expressed this well one day. She said:

> *"At Rackspace, one of our values was 'fanatical support in all we do.' It was an interesting value because a lot of companies care about their customers, but the word 'fanatical' drew a line in the sand of this being extremely important. People in meetings could say to one another, 'Yes, but is that fanatical? It was a good way of checking that we were actually living up to our values."*

At Netlify, one of the company values was: "The best idea can come from anywhere."

I loved this one, because I do believe that people at all levels and from all parts of an organization can bring fresh ideas to the table. Similarly, we could refer to this if bureaucracy threatens to get in the way of letting through a good idea that was incredibly valuable.

Good company values do this. They convey a viewpoint that you can easily understand and check against. If you notice your company lacks a point of view in their mission statement and you're in the position to push back, I highly suggest asking about it. Leadership is all about driving clarity, and company values should make it clear what the driving forces are in your organization.

TEAM VALUES

I make a list of team values for whatever organization I run. Why do I do this? Because there are certain things I value on my team, things that I don't necessarily need to convince the entire company are important.

Here's an example of values I've put forth on my teams:

- We celebrate everyone's wins as though they're our own.
- No compare and despair.
- Be empowered to forge forward, we hired you because you're an expert.
- Make mistakes.
- Choose impact over butts in seats (We're not counting hours here, focus on making an impact and work when and how you want).

It is not at all important whether you, dear reader, agree with any of these values or use them. However, giving your team space to build values together is important. Doing so helps your team figure out ways to support these things and push back on these things if they find issues with them (better to debate it out now, versus when we're embroiled in something). If their values are upheld, there are no surprises, especially for things like career laddering. You'll note that the supportive and collaborative qualities I encourage here are also how I level my principal engineers.

If people don't align with these values, it may make more sense for them to find another team. That's okay! The interesting thing about values is there's not necessarily a right answer. Rather, there are values that are congruent with the type of team we want to build, and some that are not.

I feel strongly that people should seek out work on teams that don't violate their own value structures, ideally places where they feel the most alignment—perhaps not perfect alignment, as it's also nice to learn from folks who have a different perspective. A few major, fundamental values should be aligned. Otherwise, work can feel even tougher than the job at hand. When you're misaligned, burnout is inevitable.

Working on a team with values that largely align with your own is nice, because you can really dig into the makeup of the culture on your team. **Clarity is what we're aiming for here. Clarity is key.**

VALUES AND BOUNDARIES

At first, values and boundaries may feel like unrelated subject matter, but they are intrinsically related. Values are so key to who you are as a person that it's sometimes not enough to identify them, you also have to protect them. You express what you value in practice by what you let into your life, what actions you actually take, and how you spend your time.

For instance, if I say that family is a value I truly care about, and then let work bleed so much into my life that I have no time for my family, then I'm not living in a way that's true to my values. If I don't create boundaries to protect my core values, I eventually feel misaligned, like my life's work isn't lining up with my needs, and that's not sustainable. As we've mentioned, **misalignment with too many fundamental values can lead to burnout.** And this type of burnout can build slowly over time.

Boundaries are immensely important to a healthy working environment, because everyone has slightly different boundaries with work. Not only should you pay attention to inconsistencies between your values and your work, you should also strive to make sure you're not violating the values of your team in the day-to-day as well.

When you're doing a values exercise with your team, take notes for yourself about your team and what they identify as important for themselves. In the same vein, if they set a boundary at work, note which value might drive this. What's interesting about discussing values as part of a group is to see where they intersect, overlap, or have gaps. Often you'll see a large group where many people share one or two values, but diverge at a few more. It's important to note both.

Conducting a values exercise with your team gives way to tools that are applicable in so many contexts: feedback, managing conflict, 1:1s, prioritization. You'll note in future chapters that I bring up this concept quite a lot.

There are no wrong answers when it comes to values. Everyone comes from different backgrounds and lived experiences. The more we understand one another, the easier it is to work together. When people talk about "good" managers, they are likely referring to people who show care and appreciation for people's values, and also respect their boundaries.

1. https://www.youtube.com/watch?v=735_W3zsIw4

CHAPTER 3

TRUST AND VULNERABILITY

That's a lofty chapter title! Before diving into trust, let's first talk about what we think our work is as managers, and perhaps what it isn't.

"Trust is built in drops and lost in buckets"
—Kevin Plank, founder of Under Armour

As the quote implies: Trust takes time. No way around it, trust is the thing you will want most to build with a team. But trust evolves slowly, takes care and attention, and is easy to lose. Trust is delicate and valuable and should be treated that way.

OUR JOBS AS MANAGERS

If I was going to nail down what I think my job is, **it's to enable the people around me to do their best work . . . together**.

Let's break this down a little further:

- **Enable:** This means supporting, growing, and nurturing the development of other people. It means creating conditions for them to do their best work. It means my job is less about me, more about them, and meeting them where they are.
- **The people around me:** A good manager is not just watching their team and shutting out everyone else. A good manager looks at the

wider ecosystem: their team, yes, but also their peers, their peers' teams, the managers above them, the whole company ecosystem.

- **To do their best work:** Their best work is hopefully a synthesis of what they can and like to do, and what the company needs. We're striving to find that bit in the center of the Venn diagram. In order to do this, we talk a bit about drive.
- **Together:** One of the interesting challenges about the role of manager is that as you're looking out for these individuals, as well as the larger company, team dynamics come into play. There is a bit of macro and micro work that intermingles, and it can be challenging to juggle both.

Even in these statements, you can see the two key ingredients that will aid us in working well with the team: trust and vulnerability.

TRUST AND VULNERABILITY

About a decade ago, someone on my team accidentally put in a PHP call that created thousands upon thousands of pages that tanked the company's SEO and greatly impacted our database storage. He called me at 1:00 a.m. Bleary-eyed, he and I and the rest of the team worked well into the morning to fix everything and restore proper functionality to the website. The whole team worked through this together, without complaining or me having to ask them. And they knew I'd give them bonuses and time off when we recovered.

However, they weren't there because they had to be. They were there for their teammate. This was an anomaly—our team didn't typically do this kind of work, but when things got hard, everyone showed up because of trust. We had his back, we didn't want him to have to manage on his own. He trusted he could call me with the problem and I would help. He knew that he wouldn't be made to feel shame, that we would take this as a collective opportunity to improve our system. They also trusted that I would give them compensation for their work and time to recover.

Trust isn't just a fuzzy, feel-good word. In a working environment, trust improves the strength of the systems we build. The book *Accelerate* by Nicole Forsgren, Jez Humble, and Gene Kim addresses this in depth, surveying culture on teams and studying the correlation between high-performing teams and low-performing ones. Surprising no one, the teams with the most generative cultures (mission-driven, trusting) were also those with the least outages and highest performance. (I highly recommend this fascinating book.)

Trust is crucial. Cultivating an environment of trust should be paramount for a manager. Why? Because trust is the basis for a few things:

- **Hearing each other out**: Conflict is not always bad. Sometimes you need to have healthy debates to come up with the best plan and outcome. You can only do this when there's trust in the group and between team members.

- **Finding the human in the mistakes**: We are at our jobs for a large percentage of our waking hours. There's zero chance that you'll make no mistakes with all that time and effort. If a group trusts one another, it recovers from a mistake, acknowledges it, learns, and moves on. People don't belittle or weaponize mistakes. People can show humility about the mistake and learn from it rather than become defensive or pass the blame. Trust helps with all of this.

- **Creativity**: Lack of trust is stifling. It's impossible to be innovative when you feel like every movement is judged. Trust and happiness actually support and develop creativity. It's good for business. More on this in a moment.

- **Making our jobs more enjoyable**: A working environment where we can be ourselves, make jokes, and be within a trusted group can help us feel included. It's hard to stay at a job without that trust; and likewise, you can retain employees by providing it.

So how do we cultivate trust?

MODELING VULNERABILITY

I have a theory that the Spiderman movies keep getting better ratings while the Batman movies are getting worse because of *modeling vulnerability*. Marvel doesn't cover up Spiderman's weaknesses. They lean into them as part of telling the story. Spiderman is deeply relatable because his character isn't perfect. Batman, on the other hand, seems less and less like a human we could understand or be. In my hypothesis, this is why they have to make the movies so dark; it's the only way to suggest there's something compelling to understand beneath the surface. In reality, the darkness just makes me squint a lot.

It's hard to model vulnerability, because we think it makes us weak. In truth, it takes a ton of courage and can embolden courage in others too.

You may notice that I say this again and again: **you as a leader have to go first**. You have to show you can be trusted, and you have to be trusting. This means being okay with being vulnerable with your team. It means explaining to them how and why you care about their well-being, in words *and* in actions.

Some folks think that management looks like a steel fortress of preparedness and authority. I'm not so sure about that.

Let's say something is going wrong. Are you more likely to unpack what's happening with a manager who is always put together, or a manager who owns their mistakes? The truth is, your team needs to know you're human.

You can't fix problems if you don't know about them, and no one will tell you about them unless you make space for them.

You're the manager, so if someone is going to show vulnerability first, it's easiest on team dynamics if it's you. You can do this by admitting you're having a bad day, that you don't understand something, or that you made a mistake. You don't have to do this constantly, a little is helpful.

This is tough if you are an underrepresented minority in tech. Some people may think of an admitted mistake coming from one person as humility and another as failure. I myself struggle with admitting vulnerability—will it make me be seen as weak, given I'm a woman in tech? Some behaviors, when displayed by my male peers, can look like humility. The same behaviors, when displayed by me, could be read as incompetence. This dynamic exists for any in-group/out-group dynamics: race, social class, disability, and so on.

It's okay to admit that modeling vulnerability is complex. Feel out the situation, given your circumstance. Not everyone is on the same playing ground. You may have to test the temperature in a room, and even check whether or not you have the mental cycles for things going sideways, if they do, before acting. Leaning on your lived experiences may help here, as your mileage may vary.

INDIVIDUAL TRUST

There is no single "correct" way to create individual trust with someone. There are so many ways to build bonds. Some of what we mentioned about values in the previous chapter will apply here, as well as what we will explore further in the 1:1 chapter to come.

Unfortunately, humans are bad at judging books by their covers. We are by nature more drawn to people with whom we share similarities. However, in our role as a leader, we have to try our best to connect to folks no matter how similar or different they are to us.

I frequently try to remind myself that everyone I meet has something they know well, and that I would be lucky to learn from them. Every person has a wealth of experience and knowledge. I am grateful that in the time I spend with them, I can learn and grow a bit more. Learning is one of my values.

A 2010 study by Matthias R. Mehl, et al., titled "Eavesdropping on Happiness: Well-Being is Related to Having Less Small Talk and More Substantive Conversations," found a meaningful correlation between the depth of conversation quality and the depth of personal connection: when you force yourself out of small talk and into subjects that matter to you, people tend to build trust faster.

It can be scary to go deep with someone soon after meeting them, but this study shows that small talk feels very ephemeral to people, while diving in and getting to more substantive conversations with folks, even if you don't know them closely, can build important trust and feel better for both parties. Going deep about something very personal can also make a person uncomfortable, so be careful to build this depth over time.

Humor is also a way to build trust, and one that many overlook. If you are able to laugh together, it's much easier to have mutual vulnerability. I remind myself of this all the time to make myself feel better when I subject my team to my terrible puns.

By the way, did I mention what glasses a UX developer uses? Wireframes! Get it?

Moving on . . .

If someone has been burned in the past and you want to support them . . .

Let's say you have someone on your team who was treated poorly at a previous job. Someone like this isn't going to trust right away that you have their best interest at heart. Why would they?

A possible path to supporting this individual and building trust could be as follows:

- Ask questions about their current state, and listen truthfully and with your whole self. Listen to what their everyday job is like, listen to the things they think are going well, listen to what their goals are, listen to their values, and listen to what they think needs fixing on the team.

- Think about the values we discussed in the last chapter. What are their values? What is the personal context in which they're working? How do the individual's values match, or not match, with those of the team, those of the larger organization, and with the work they are doing?

- Notice what good qualities they bring to the table, and let them know you see and value those qualities. Everyone brings something to the table. Pay attention to what they do well. Trust and misalignment may happen when individuals haven't felt seen or valued.

- Ask for feedback. In a situation with a power imbalance, the person may not feel comfortable telling you outright how you could better support them. Again, you have to go first as the leader. Showing vulnerability to them may help bring the barrier down. A note on this: only ask for feedback when you're in a mental state where you can handle it without getting defensive (More on this in a bit).

- Notice their interactions with their peers: Are things lining up? Are there any disconnects? Are there any miscommunications? Sometimes when a person goes through hardship, they can put up walls or be protective. Evolutionarily speaking, that's the right thing to do to protect themselves from experiencing hardship again. However, they may be unaware that they're acting in a way that can feel prickly to folks who may not be aware of the original incident. I have done this too, and I needed someone I trusted to point it out.

- **A larger action item: try to change any processes or patch up misalignments that exist *around the person* that might alleviate the disconnects.** Are they overworked on projects that aren't meaningful to them? If so, perhaps reallocate their time. Is something they're working on lacking clarity? If so, can you provide more clarity? These acts by themselves can help build trust, because you are *showing* (not *telling*) that you care about their well-being.

- Think about a Venn diagram where one circle represents what the person is interested in, and another represents those tasks the company needs them to do. Are the majority of their tasks in the center of that Venn diagram? If not, can you help guide them there?

- Reinforce that you do care about their well-being—that it's important to you that they feel supported.

- If you need to alter their work at all, tell them why: "If we can move you to this project, I believe your work will have more meaning and be less busywork. I want to see you in a role where you can really thrive."

This is just one type of many approaches to building trust. The important thing is that building trust has to matter to you. You must put in the work on your side.

Remember values and boundaries, first and foremost. Where is the person coming from? What boundaries do they need to feel supported? Is the team supporting those values, or is there misalignment? It's difficult to build trust unless you work to understand each other.

This will take a lot of upfront work from you. After doing these exercises, the team will tend to require less from you over time, so you won't always have to be in that initial trust-building and fixing state. Smaller movements will keep your team unblocked. Some of the ways you build trust will produce a more-effective team overall, one that works together more seamlessly.

I can't express how rewarding it is to see someone excel! Especially if things were harder for them in the past.

TEAM TRUST

At one point I took on a team that was full of smart, wonderful people who were all hard workers. Unfortunately, they had not yet really built trust among themselves. I don't mean to say that they disliked each other; it was very clear they all respected one another. But a bit of a wall was between them. The cost of this wall was that small problems arose that they didn't feel comfortable bringing up with one another, so issues just . . . lingered.

How do you build trust, not just between you and each individual, but between team members?

One way is to note how members may be avoiding building trust, even if it's not immediately obvious. In order to evaluate that, let's look at some qualities that a trusting team has:

- They feel comfortable raising issues directly with each other.
- They share personal details with each other, and admit when they are having a bad day—to other team members, this is no big deal.
- They can debate with one another to find good outcomes.
- They are flexible with one another's needs.

Since we know the qualities of a trusting team, we can start to think through what it might look like if team members lack trust:

- They raise issues with you about one another rather than with each other.
- They are closed off and don't know much about each other. They don't feel comfortable admitting when they're having a rough time, so no one has context when things go poorly.
- When an issue is raised, they quickly close it, and whoever talks first "wins."
- They think mainly of their own needs without checking in with the group.

The good news is: you now have a path forward! If you see any of these behaviors, you can gently guide your team toward a more positive place, by suggesting they raise issues in the team meeting instead of with you alone, or set up a 1:1 with a peer. You can even call out, "We're trying to build a bit of trust within the group in order to work well together, so let's try to work through this together."

In the next chapter, we discuss the role social connection plays in creating an environment where people feel supported in doing their best work.

WEEKLY TEAM TIME

One major mistake I've seen again and again is when managers don't meet with their teams. I get it; engineering teams like to . . . do engineering work! No one likes a lot of meetings, but getting a group together is valuable time spent. Your team needs to see one another with a regular cadence. They need to talk through their work, discuss issues, and hang out a bit. This is especially true of remote teams, with members who don't cross paths with one another in a work environment. But you still need to create situations where people can meet with one another, remote or not. In *Social Pressures in Informal Groups: A Study of Human Factors in Housing* by authors Festinger, Schachter, and Back studied friendships in apartment buildings and discovered that proximity and exposure contributed to how much trust people developed with one another. People tend to like folks they have more similarities with. The more we spend time with people, the more similarities we may find, even if not immediately obvious or surface level.

Building Trust Across the Team

Sometimes teams will build trust with you individually and not with each other. It feels great to build this kind of trust with someone, but your job is to boost the *entire* team's morale. If you build a team that only trusts you and you alone, they are not set up for success without you or creating systems beyond those that involve you.

Let's say a member of your team has an issue that affects their group or peers. The first thing you can suggest is that they bring this issue to a weekly team meeting, and you can facilitate an open discussion. It's important to reinforce that you'll be there to support them, but also align them with the *why*: In order to be healthy, the team needs to build the muscle of communicating with each other more directly and diving into productive debate instead of dealing with things behind closed doors, or not dealing with them at all.

It can be awkward at first to push some conversations out into the open, but it does become easier with time. A manager should facilitate these discussions (more on this in the Managing Conflicts chapter).

Some general suggestions I make to help build team trust:

- **Depending on the size of the group, a brief standup twice weekly can be productive** (notice I didn't say "biweekly" because it's the most confusing word ever). I've managed groups where the standup entails people talking through a kanban board, and other groups where the standups alter: one for triaging issues, one for discussion. Do what works for your team. But I suggest having your team meet on a regular cadence to foster that togetherness.

- **Let people chat.** It's probably not wise to just chitchat the entire meeting, but some natural conversation is great for keeping people from feeling so isolated.
- **Make dumb jokes, if you feel comfortable doing so.** Humor can be incredibly bonding. Teams that can joke around together are often a bit better at admitting mistakes and disagreeing without taking offense. Besides, it makes work more fun.

TEAM-ONLY SPACES

Consider: if you're working in an in-person environment and your team goes to lunch, they don't take the whole company with them. They have time and space to chat among themselves. **This gives a certain psychological safety that's tough to have if they feel the whole company is watching them.** This psychological safety is important in building an atmosphere of trust.

Sometimes conflict and issues arise in a team that are really no big deal, but they have to be sorted out by talking through them. It's incredibly disruptive to an organization if a team member tries to talk through such conflict or issues out in the open. All of the sudden, someone from another group or someone who has a ton of status may "weigh in" to influence a decision that doesn't affect them.

In a situation like this, you will likely lose the voice of the shyest people on your team. And I can tell you right now, shy folks often have keen observations you don't want to miss out on. This type of open forum also prohibits the team from feeling inclusive, trusting, and autonomous.

There is, of course, a time and place to have more open conversations. I'm not saying to do away with all open conversations or written documents. I am saying that it's a good practice to balance open conversations with a place where team members can relax a bit. I recommend making a private chat group for the team. A lot of companies want to have every chat out in the open, but I've managed a lot of remote teams, and I think it's important for a team to have a place of its own.

When you first start a team chat, you may be the only one talking for a bit. That feels awkward. It passes, I promise. Like I said before, you have to go first. I tend to start things out a bit lighter: some dorky memes, a "Have a great weekend!" here and there, celebrating some small team wins or an individual's hard work. After a while, other folks begin to follow suit.

I once took on a team that didn't chat with each other much. I started the chat channel, and for a while it was just me (this usually happens). Slowly, the team started chatting with each other—celebrating one another, joking, working through problems. On one night in particular, they all chimed in and joked for a few hours while I stayed quiet.

That night, I danced around my kitchen in glee and celebration. Hey, in management, the wins can be strange. We take them when we can.

CHAPTER 4

YOUR TEAM IS NOT "THEM." YOUR TEAM IS "WE."

When I consider good managers whose teams deliver well, I have noticed some similarities. I've also noticed some similarities in those managers at the other end of the spectrum, whose teams do not deliver well. It starts with how the managers communicate about their teams. The way that we refer to our teams sends signals: to stakeholders, to your peers, to the team itself, and even to ourselves.

The way you talk about your team and the work you're doing can build trust or break it. Owning your mistakes as the leader accountable can take vulnerability. Being mindful about how you communicate about your team is one way we can put trust and vulnerability to work in practice.

There can be a perception that as a manager that you are in control at all times. That control can be reflected in how you appear to be in charge, in how competent you are, or how you personally perform. Due to our concern for how others perceive our level of control, some bad behaviors can arise, not due to malice, but perhaps due to fear or caution. For this reason, it can be tempting to take credit for success and avoid taking accountability for failure.

The irony is that the more you try to hold on to these external perceptions, the more it will slip away. Why? **Because the problems you are solving as a manager really aren't about *you*.**

Your team is "we." You are a driving force of that team, no matter how high up the hierarchy chain. What happens on that team is your responsibility. When you speak about your team, include yourself in the statement.

When your team succeeds in something, though, praise *them* and leave yourself out of it. By leaving yourself out, you give the team the credit they deserve. If you include yourself, people can take that to mean you're inflating the story for your own sense of self-worth (and thus diminish the attribution you're trying to give them for their work).

Here's an example:

> *"They really pulled this project over the line, despite the incredibly tight project timeline. Everyone showed up and was driven throughout the engagement. They did a fantastic job."*

However, you're the responsible and accountable party for setting the team up appropriately for the success of the project, so if the team failed at something, the pronoun is then "I.":

> *"I didn't recognize how tight this turnaround was, and I failed to prioritize the team's time well. I need to reconvene with everyone so we can come up with a better plan."*

And never, ever "them." As in:

> *"They didn't adhere to this tight timeline. They just weren't able to get this project over the line."*

Do you see how the last example shirks responsibility for what occurred? Too often I will hear managers relieve themselves of their duties when shit hits the fan, and that is exactly when a manager needs to step up and dive into the problems that **are their responsibility**.

THE WIDER ORGANIZATION

Another piece of this is how it impacts the way your team operates. Your job is not to be the ambassador of who you manage. You cannot think of every other group as separate. You're part of a larger system. A company is composed of groups, but those groups can only be successful if they're working together, not if they are protecting their own group at all costs.

I admit, I didn't fully understand the depth of this until I read Patrick Lencioni's great book *The Advantage*, thanks to Dalia Havens, a peer at Netlify. Lencioni writes about organizational health (not "being smart") as the biggest key to success. Plenty of smart people with good ideas build companies and see them fail. **Organizational health lies in being able to work together.**

YES, THE LEADERSHIP TEAM AS WELL

Fundamentally, all groups within a company are part of one whole. **The leadership team is also a team, and should also be treated as *your team*. How you speak about this team is equally important.**

As such, when we talk about the successes and failures of any groups, remain mindful about how we refer to them. There should be a sense that you're all working toward a common goal together, and every group contributes to the overall success. Within a leadership team, there should be trust and vulnerability to own their part so that the whole organization can operate at its best.

You may see where I'm going with this: When you talk about the leadership team, this is **"we"** too. You can't speak to your team about decisions that were made at a table with your peers and boss and say, "They decided [something you don't agree with]" even if you don't agree. You were there. Ideally, you took part in that decision. So when you talk about that team, presenting them as "we" is important as well. ***Why?*** **Because as a manager, our job is to try as much as we can to drive balance and clarity.**

It's confusing and disorienting to hear a manager talk about a leadership team they're on as though they're not a part of it and as though they don't take accountability for what's happening there. Your reports can't effect change at that level; if you don't own your involvement in the leadership group, you can demoralize your team and make them feel distrustful of other parts of the company. This will then cause folks to demonize other teams and their initiatives, which (as we've discussed) is ultimately unhealthy.

Saying "we" holds you accountable to your team for leadership decisions that you are a part of, which is how it should be. If people on your team have issues with the direction in which the leadership team is headed, it's also your responsibility to own that conversation and next steps, as a liaison to the leadership team.

In some instances however, defending the decisions of leadership might not be appropriate, i.e., if the leadership team made a decision that really goes against your core values, that you fought strongly against. Those instances should ideally be infrequent, otherwise you may unfortunately need to pursue a job in a different organization.

Here's how this works in practice, using an example of conveying a decision at the leadership level to the people who report to you:

> *"The leadership team decided that we need to ship at least three features this quarter, so I guess that's what we have to figure out how to do."*

versus:

> *"One of the key OKRs (objectives and key results) this quarter is that we as a company need to double the signups to our platform. We've done some calculations that show we can almost certainly get there by shipping three features, so let's all talk about what we can do within our group to make that possible. If you're curious, we can chat through what initiatives other groups are doing to support this as well."*

The first is not just passive, but demotivating. I have made the mistake of using this approach when I wanted to be liked by my employees and for them to think of me as a peer. But we're not peers. I have a responsibility to them.

You'll note that in the second approach we also explained the reasoning behind the decision. I've noticed that when I have to hold myself accountable to the decision, I care more that people understand the reasoning behind it. This is a very good thing for boosting the morale on your team, which is arguably one of your most important jobs!

The last line in the second approach also opens up discussion. Since you're taking ownership of the decision, you're also owning that you know about other pieces of the puzzle, and are showing a willingness to dive in with your team.

WHAT IF YOU MAKE A MISTAKE?

We all make mistakes! Management can be difficult and it's impossible to be perfect all the time. I've made lots of mistakes as well. Think of a mistake not as a stick to beat yourself (or others) up with, but as a lesson learned to be as mindful as possible in the future, to promote a better working environment.

If you realize you've made a mistake while in a meeting, it's possible to backtrack and clarify, revisiting the issue and taking ownership of it. Or if the conversation has moved on, you can address this when you speak about next steps. In situations where things didn't go well, clarify that you are ultimately the accountable party, and you will take responsibility for outcomes.

We communicate to our teams, peers, and stakeholders, whether or not we're taking responsibility as a true leader in these moments. We communicate whether we approach a problem with humility, and a desire to collaborate and improve. This humility is a powerful piece of leading an organization

CHAPTER 5

HAPPINESS AND DRIVE

On a project our team worked on, we set up a little cartoon space hamster bot (because every team needs a space hamster bot) to alert us when a particular service had completed running a transformation. The night before a big release, someone on the team pushed a change that created thousands of calls to that service, which in turn thought it was the target of a DDoS attack, and shut down our access. Hundreds upon hundreds of space hamsters appeared in our communication channel, with intermittent cries from our team of, “Oh no!” and “I have regrets!”

We only had a handful of hours before the release, and everything was broken.

Here’s a moment when a lot of folks could have panicked and blamed one another. Instead we giggled wildly, jumped into chats and onto calls, fixed it, and kept going. I couldn’t have been more proud of the team that day. Their response was wonderful. And it makes all the difference in how we work together, recover, and iterate.

Wouldn’t it be nice if everyone on our team felt like they were doing their best work possible? That they avoided cognitive overhead when they could, and felt an innate drive to complete their work instead of trudging through for a carrot on a stick? Our job as engineering managers is to create an environment where people can feel like they’re getting their best work done.

The culture and morale on your team is a large part of your responsibility.

I've long felt that happiness wasn't just "nice to have" at a job. I've noticed over many years that the teams that are able to express joy and humor together are often the most productive, both in good times and when things get tough. It wasn't until I read *Drive* by Daniel Pink and *The Happiness Advantage* by Shawn Achor that I realized science backs up my observation, and I couldn't have been more thrilled. Happiness at work isn't some "hippie" concept. It's also connected to the bottom line.

TOXIC POSITIVITY AND FALSE HARMONY

"Buck up, little camper! It's not so bad!"

Is there anything worse than hearing those words when you're sad or upset? Probably not.

When I refer to "happiness" in this book, I'm not talking about smiling and saying things are all right when they aren't. You can feel down and be honest about that while still being connected to those around you. You can feel safe and trusted enough to tell the truth. You can smile when you mean it, not smile when you don't feel it, and be around people who care enough for you to see you through that gamut of emotion.

A strange kind of happiness comes with being part of a team that fosters trust and vulnerability, but that's only part of the story.

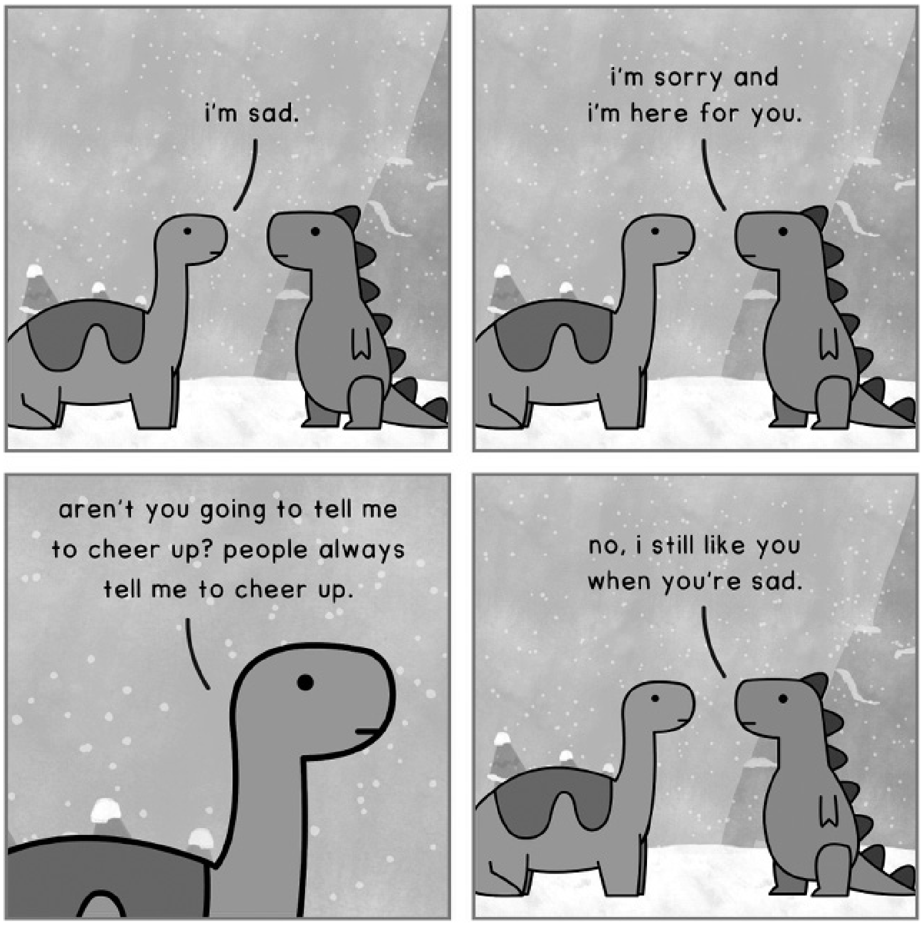

Figure 1: Dinosaurs that still like each other when they're sad. *(Used with permission @dinosandcomics)*

FLOW AND PURPOSE

It took my husband five hours to convince me to leave the house so that he could propose to me in the park.

"Five more minutes!" I kept shouting across the room at him, for hours on end while I worked on a coding project that definitely did not only have five minutes left to go. The poor guy had champagne in his bag that he had to keep returning to the fridge each time he realized I wasn't budging.

I had no idea what his intentions were at the park that day, but I was so heads-down on what I was doing, I didn't notice he was a little more nervous than usual, or that he had a huge bag packed for a pretty simple excursion.

I was in flow state. I love coding. It's truly hard for me to rip myself away when I get going on a project.

Can you think of the last time you felt truly immersed in a task? The last time you forgot about time and space and even yourself. Some call it *flow state*, moments where you're so involved in what you're doing, you feel like every tool you need is at your fingertips, and what you're making is bigger than you. When your sense of purpose is at the forefront of your mind. This sense of autonomy and mastery is something I wish for and try to cultivate in my teams, in part because I myself feel so much joy when connecting to my work in such a productive way.

Coined in 1975 by psychologist Mihály Csíkszentmihályi, *flow state* is the phenomenon in which a person is fully immersed in an activity that leads to focus, energized involvement, and enjoyment. A person in flow state is no longer thinking of multiple things, or even their sense of self, but is singularly focused on a task or challenge. Many people report it is the happiest feeling in their lives.

Another wonderful benefit of flow state is that tolerance for frustration goes up. This goes back to the concern I hear around the "passion" argument: How do folks stick with it when it gets tough? In flow state, obstacles that would have otherwise caused anxiety and burden are perceived differently, and people have more resolve to push through a task.

So I'll ask you: When are the people on your team feeling this sense of purpose? What work allows them to feel flow? Does that mean a manager owns their employees' flow state? *Sort of.*

As managers, we can't actually create flow state. That's up to the individual. But we as managers should create an environment where flow state can exist as much as possible. A few conditions need to be met for your engineers to get into flow state in their work:

- You are aligned on the base premises of the work.

- Your work is challenging, but not impossible.
- You feel a sense of togetherness with your team and peers, that you're all building something together and have each other's backs. Your moral values are not at odds with the work at hand.
- You feel respected.
- You get fair and timely feedback on your tasks. This does not necessarily have to be human feedback. It can also come in the form of compilation success, tests passing, or PRs (pull requests) going through.
- Upon completing tasks and working hard for the company, you get fair pay, bonuses, promotions, and raises.
- You feel a loss of sense of self when performing your work. You are doing something for the sake of the task, not for politics or to be right. Rather you are doing the best work possible for the best possible outcomes. As such, you care more about continuous improvements than your own ego.
- You feel like people believe in you and what you're working on.

If we can cultivate environments where people can achieve flow state with little distractions, so much of our work can be removed of unnecessary friction. A person who can be in flow state is also capable of seeing perspectives other than their own.

Can you trust the teams you lead to have autonomy to drive outcomes? You hired them for their expertise. Are there ways you can allow them to bring that out?

Ultimately, a happy, driven team sees the wider purpose of their work and also feels empowered to execute tasks individually. Remember to celebrate your team members when they do well, as soon as possible. Engineering is tough. It's critical to be there, cheering for them at that finish line.

ACHIEVEMENTS AND REWARDS

If people are bringing forth their own sense of reward in their work, how do you as a manager incorporate rewards? Due to the creative problem-solving nature of the work, dangling carrots as rewards is not necessarily productive.

Intrinsic and extrinsic motivation

Let's talk about motivation for a moment. In the times that you felt completely connected to your work, where everything else floated away, what were some commonalities of those events that you might be able to isolate?

- Your work was challenging, but not impossible.
- You were likely connected, in part, to the goal of the task.
- You did the work regardless of outside influence or reward.
- The act of doing the work was a reward unto itself.

In recent years, a lot of studies have been done about intrinsic and extrinsic rewards. Intrinsic rewards come when you do something for the sake of doing it, and extrinsic rewards come when you complete a task to get outside validation. The book *Drive* isolates each type of reward and shows how those motivated by intrinsic working habits and desires are more productive over the long haul.

Of course, this hinges on whether or not the person's basic needs are met: they're compensated fairly, often better than average. A person will struggle to reach flow state if they are not compensated fairly. Otherwise they will be consumed with the inequity, and it will be challenging for them to focus on the task. I've seen this time and time again. Part of our duty as managers is to strive to ensure that our staff are compensated fairly for their work. This can at times mean working with the company's people

(Human Resources) team, or with other stakeholders to understand the process for raises or promotions.

In the following chapters, we talk more about extrinsic rewards and how to set up career progress and promotion.

SOCIAL CONNECTION

In stressful moments, we're not usually thinking about cultivating our relationships. However, the more connected you feel to other people in times of stress, the happier you are, and the less stress you report. Isolation can actually hurt productivity. Social support can allow people to more easily recover from work-related stress. People even report that strong social connections with peers makes work feel less like work overall, even when they're tasked with longer hours and difficult conditions.

Positive social interactions raise oxytocin, which lowers anxiety. Social support has even been shown to extend one's lifespan.

Working to create an inclusive environment is quite literally connected to doing great work. This also means encouraging bonds between workers, even when you're not around.

Let's say Kiran says he has felt lonely lately and that he really doesn't feel connected to his colleagues. This is a time when it may seem like individual trust needs to be worked on, but maybe Kiran is giving you a hint that team trust, for everyone, needs to be worked on. Perhaps it's time for a team off-site? It might be nice to spend a little time getting to know one another. You can do these remotely as well. Some activities that I've done during an off-site gathering that have worked well:

- Have everyone on the team do a short (five-minute) presentation on something that interests them, not work-related.
- Do an escape room or play a game.
- Do a values exercise, and discuss how you like to receive feedback.

- Eat the same food together. If you're remote, you can order something frozen that's delivered in advance to everyone. This gives the team the opportunity to talk about the tastes and smells. (While being mindful of food preferences and allergies.)
- Have everyone talk about something a bit personal: When they were small, what did they want to be when they grew up? If they could have any superpower, what would it be and why? What was their greatest challenge growing up?

You may be surprised how much these small exercises really open up a team and help you all get to know one another a bit better. I used to create happy hour/social hangout meetings (not necessarily with alcohol). Sometimes I attended, and sometimes I did not. Aside from needing a break from so many meetings, I wanted the team members to have their own space and time to bond.

PEER AND MOB PROGRAMMING

Peer programming, or mob programming among many people on a team can be a good bonding exercise when done well, but can also have detrimental effects if executed poorly. When we speak about trust and vulnerability, and drive and happiness, peer or mob programming can be a way for the team to build all of these muscles. In peer and mob programming, team members share their thinking through a problem, as well as wins and mistakes.

A key to this practice is making sure that **the person with the most knowledge remains patient.** If you can find someone who can help guide a team, and be respectful and kind, it can create an open environment where people don't feel guarded. Someone who is good at separating their self-worth from their work will be a good person to lead pairing sessions. The key is to find someone who derives value from supporting others, not in looking smart themselves.

Peer and mob programming can also help a team **share context when a project is in a 0-1** (from nothing to productive) state. If you're just getting a project off the ground and there's discussion about architecture, working through it together can help all subsequent PRs. There are fewer surprises when everyone starts together. Code review also goes quicker, with less chance for passive aggression, or time wasted trying to figure out intention. The team is starting from a place of mutual understanding, and that continues on to subsequent processes.

Finally, peer and mob programming **helps remote teams feel connected**. When you don't see one another in the halls or at the coffee machine, there can be a tendency to only see one another during meetings. Then you're heads down alone for most of the day. This can feel a little bleak if the work is not also balanced with things like team happy hours, but even then a bit of collaboration is missing. Peer and mob programming can get people thinking together on hard problems and feel more connected in the process.

NEGATIVE BIAS

As a manager, you can never guarantee an outcome with people, but you can make sure you're doing what you can to give your team an environment where they feel safe and intrinsically motivated about their work, with little distraction.

Our brains have a negative bias. This seems like a bad thing, but *negative bias* is actually a very strong, protective evolutionary mechanism. You burn your hand on the stove, your brain forces that memory on you so that you know not to do that again. Given two different memories (a bad one and a good one) your brain will remind you of the bad memory more often, when left to its own devices. When you feel fear from something, like when your brain remembers that burn, you try to avoid that danger in the future. "Good job, brain." Pat, pat.

Unfortunately, the mechanism of negative bias doesn't serve us so well in our day-to-day work. Not every fear reaction is an endangerment to our livelihood or well-being. We're no longer running from predators.

What does this have to do with happiness? Well, while you or your team members are sorting out a problem, your brains might parse and hold more of the threat related to that problem than is real or helpful.

This means that you have to be proactive and do some work with yourself and your team to unravel this phenomenon. This is called the *undoing effect*, and it can help mitigate stress when there's conflict.

Here are some ways I've implemented the undoing effect:

- **Check in with facts**. Sometimes we react to an event without checking that we understand exactly what's going on. Sometimes our brain protects us by leading us to a negative outcome so that we might prepare for it. This is very unhelpful when trying to get a team on the same page, because some members may feel misrepresented or attacked. My coach, Jessi, often reminds me of this as a tool. It can be so helpful to revisit the literal facts involved in a given circumstance.

- **Find and rally around the positive.** Since your brain is actively pulling you and others into fear mode, looking for the positive elements of a situation and saying them out loud can help balance what's naturally imbalanced. Neurobiologists find that because of our brain's mirror neurons, even introducing one positive person to a group can raise the overall happiness of the team. When hiring, keep in mind that not everyone you hire has to be positive all the time, but it's nice to have at least one person on the team to balance things, especially if others tend toward the negative.

- **Reject negative premises.** As a manager, you will sometimes have to counter paranoid or negative views on a situation.
- **Review consequences.** Most of the time, the consequences of an action are not as dire as people make them out to be. Announcing that "Everyone will be fired!" or "The system will implode!" is not very productive. What are the real risks? State them clearly, or let someone else voice them, so that everyone is on the same page about what's at stake. Then you can sort out what the biggest risks are and what aren't really risks at all.
- **In dire situations, step away.** If you have extremely volatile people on your team who are stuck in a bad loop, you may have to reconvene at a later time, and meet with them 1:1 to get on the same page. Conversations should remain as open as they can be, but there can be times where meeting as a group is no longer productive, and everyone needs time away to cool off. Use this tool infrequently and not as a default practice, but use in situations where things feel especially off track.

Psychologists suggest a ratio of negative to positive interaction that humans can tolerate before they feel demoralized. This ratio, called the "Losada coefficient,"[2] suggests a 2.9 times positive-to-negative interaction (i.e. praises: reproach), with a 6:1 ratio for optimal happiness. The ratio may differ depending on the group or individual, but it's worth keeping in mind with our teams.

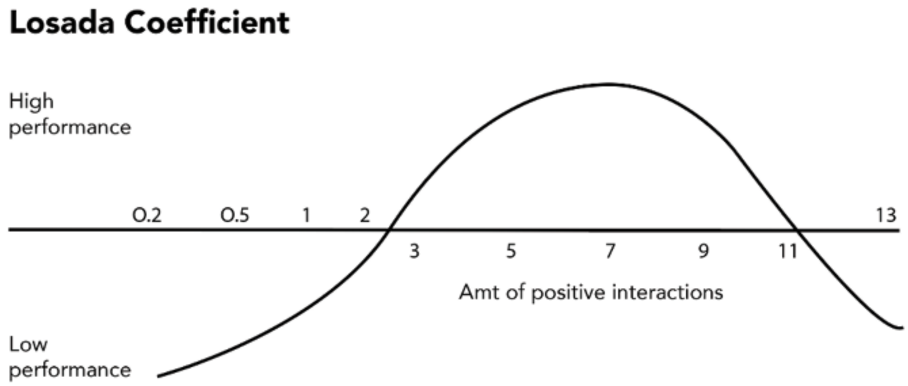

Figure 2: Curve showing the Losada coefficient—how the amount of positive interactions can affect performance, according to the hypothesis.

The concept of the Losada Coefficient in Figure 2 is heavily debated[3], but the concept is an intriguing one: that to be truly rational, creative, and thoughtful, we have to be mindful to resist our instincts to seek out and remember the negative. As Shawn Achor says in *The Happiness Advantage*: "Students who were told to think about the happiest day of their lives right before taking a standardized math test outperformed their peers. And people who expressed more positive emotions while negotiating business deals did so more efficiently and successfully than those who were more neutral or negative."

One more time for the people in the back: *You need to actively create a good culture and team morale; it's part of your job!*

This means diving in when there's trouble and asking questions so that you can understand an individual's perspective. A section on soliciting feedback is included in the Giving Feedback chapter. Feedback, as well as trust and understanding each other's values, is connected to the happiness of the team.

LEVELING UP

How do you increase focus and drive when a task is very difficult and/or out of your comfort zone? It's unrealistic to expect that you, or members of your team, will change your behavior or output, or grow by leaps of bounds in a matter of days. But we can level up slowly by working on incremental improvements.

Incremental Habits

At one time, I managed a woman who was incredibly ambitious. At every 1:1, she outlined all the things she wanted to get accomplished. Her goals were triple those of anyone else on my staff. They were all huge projects. Her ideas were sound and strategic. If she could get any one of them over the line, it would mean something incredible for the company.

At first she did terrific work. Time after time, I cautioned her that she might be trying to do too much, but I would stand back and be amazed at what she did. But after a while, her progress slowed. The boulders left on her plate were big ones. She started to freeze at the enormity of the task.

For a little while, that was okay. After all, she had consistently been bringing in incredible work, and no one should be expected to keep up such a pace forever. But after a little while, I began to get worried. She seemed a bit more shy than usual, and I could tell her lack of progress was getting to her.

"You know we don't have to get all of this done, right?" I said.

"Yeah, I know," she said. But I could tell it wasn't really me she was trying to make amends with. It was herself and her own ambition.

"Okay, how about this?" I offered. "Some of these projects are fairly large in scope. How about we break these down?" I saw a glimmer in her eye that I'd not seen in a while.

For the next hour, we worked to break down one of her larger tasks into smaller bits. We gave her the tiniest pieces to focus on that week. Lo and behold, she completed all of these small parts and came back the next week rejuvenated. Shipping some small parts helped her feel that the rest was doable. Which of course it was. For someone who might be trying to grow, breaking tasks down and focusing on the smallest, most doable bits first can help a ton.

Habits and new tasks take up different parts of the brain. New tasks take cognitive effort. We have to parse every bit of what we're doing. **Habits, on the other hand, take very little effort for our mind.**

Habits don't take much cognitive function to put into action. You can easily accomplish them without creating new memories and putting such an exacting load on your brain. If we alter our habits slightly, day after day, we can steadily work toward larger goals instead of taking everything on at once.

James Clear, author of *Atomic Habits*, wrote:

> *"Every action you take is a vote for the type of person you wish to become. No single instance will transform your beliefs, but as the votes build up, so does the evidence of your new identity. This is one reason why meaningful change does not require radical change. Small habits can make a meaningful difference by providing evidence of a new identity. And if a change is meaningful, it is actually big. That's the paradox of making small improvements."*

Think of programming and driving a car. When you were first learning to drive, you had to think of every movement, every act of coordination, all of the signals and what they might mean. Over time you could likely do all of these things without much thought.

The first time you wrote an async/await function or some other task fundamental to your job, it probably took a lot of cognitive awareness and trial and error. Over time, it became more natural. The more we can break things down into incremental habits for our employees, the higher the chance that they can get into "flow," and the better chance they have of truly enjoying their work and moving past small goalposts to get to further and further destinations.

People Can Grow

Mastery has no fixed state or prerequisite. Your brain can literally grow and adapt with challenges. In *The Happiness Advantage*, Shawn Achor refers to a study that showed that London cab drivers had a larger hippocampus (the part of the brain that's devoted to spatial memory) due to the complexity of London's streets. Achor writes:

"Who cares? While a larger hippocampus may not seem exciting to you, it forced scientists to confront the 'myth' of neuroplasticity, that brain change is possible depending on how you live your life. . . . The answer is obvious and inescapable. Brain change, once thought impossible, is now a well-known fact, one that is supported by some of the most rigorous and cutting-edge research in neuroscience."

These London cab drivers didn't have a passion for driving, or a particular passion for roads and spatial awareness. Their capabilities grew in response to the task.

Your team can grow to address challenges, they are not fixed. As their manager, it's important to remind yourself of this when the going gets tough. Believe in your team. As you set higher goals, let them know that you believe they are more than capable of accomplishing them.

2. https://en.wikipedia.org/wiki/Critical_positivity_ratio [this one may have been eliminated]
3. This wikipedia article runs down the critical feedback to the Losada Coefficient: https://en.wikipedia.org/wiki/Critical_positivity_ratio

CHAPTER 6

LONG-TERM CARE OF EMPLOYEES

"You can't call yourself a leader by coming into a situation that is by nature uncertain, ambiguous—and create confusion. You have to create clarity where none exists."

—Satya Nadella, CEO of Microsoft

Previously, we introduced the concept of intrinsic motivation. You can have an employee who cares intrinsically about their work and still feels misaligned with the overall impact that the company sees in it. That's often when the disconnect can feel the most hurtful. If you have an employee working extremely hard and doing absolutely everything they can, the feeling of being undervalued can be heartbreaking. If we don't help teams find a good career path, it can be demotivating.

When a person feels that their work is valued, that they have a North Star purpose both for their personal growth and for wider impact with the people and industry, it's an incredibly powerful thing. Knowing that what you're doing day to day is meaningful, that small tasks accumulate to a larger goal, and that you have a path forward can be extremely motivating and drive us with resilience past obstacles, into a greater outcome.

On the other hand, I have never seen employees more demoralized than when they're unsure where their career is headed and whether their title or compensation is fair. It's frustrating, exhausting, and can lead to burnout.

It's also incredibly distracting. *Who can get their job done when they have no clue if what they're doing is valued?*

Our collective aim as managers should be taking the careers of our employees as seriously as we do the team's technical/work processes. Promotions ideally come exactly when and how everyone thinks they will. The goal is to set your team up for success: everyone has a good path forward and they can focus on doing work that is both impactful and rewarding.

TRYING TO DO EVERYTHING YOURSELF AS A MANAGER IS NOT THE BEST WAY TO HELP

Years ago, I managed a woman who was bright, talented, capable, and an all-around pleasure. She was sort of new to our industry and could come across as timid, so I did my best to be a "poop umbrella" for her, fighting battles behind the scenes to set her up for success. She was on a steady track to land a senior role. Even after I decided to leave the company, I let her next manager know that she was on track for a senior position in the next few months.

Then I moved to another city. Years later, I met up with the woman and was shocked to learn she never got the position.

Here's what I learned: Her promotion wasn't the same high priority for the capable hands I left her in as it had been for me. The team was challenged with a million other things that took center stage, to the extent that her promotion fell off the radar. But even more than that, what became very clear to me was that all of that "protection" I thought I had set up for her didn't really serve her well for the long haul. For example, I didn't teach her how to advocate for herself or how to navigate the system. I vowed never to make that mistake again.

This is tough! If you're strong and care about your team as people, it can feel unnatural to teach someone to advocate for themselves instead of moving things out of their way.

The point is not to throw that person into the fire. The point is to care. Are you teaching them the things they need to learn? Are they really growing under you? Feeling like you're protecting someone at all costs can also lead to your own ego trip, which threatens progress.

This works for a manager wanting to code all the time, as well. It's a paradigm shift to realize that your primary focus is no longer to do all the programming for your team. It really isn't. You can impede your team's growth, make them feel that you don't trust them, and worse, block them because you have knowledge of the project they don't have and now you're stuck in meetings. I have 100 percent made this mistake, sometimes because I miscalculated a deadline and felt I needed to swoop in and "help," and sometimes because I didn't yet understand how to delegate. These are skills you *have* to develop as a manager. Take it from someone who messed this up.

Try to think through what skills someone needs to succeed without you. Teach those skills incrementally. Sure, this advice is easy to say, but it's really hard to do when you're in the thick of things. Spend some time thinking through ways you can inject that teaching into everyday work and interactions.

Try to work on aligning people instead to the outcomes you are looking for from them. You don't have to code it all yourself, you need to articulate why the code is necessary, and what it will need to do in order to accomplish this. Trust in them to figure out the hows and consult when they need guidance.

BIAS

This is a good time to talk through bias. You have to work to create an inclusive culture on your team. Building a trusting team where everyone can contribute is not possible if people feel excluded. Issues of bias will always be part of your role, there beneath the waterline. There's no ignoring it. It's your job to be thoughtful about this.

Building a team where a group of people from diverse backgrounds, cultures, and lived experiences can thrive is how we create healthy working environments. We talked about enabling people, all people, around you to work well together. This is part of that mission.

The first part of building healthy teams is admitting you won't know everything, and this may be a time where you need to *listen* more than you talk. Everyone's experience is going to be a little different, and likely different from yours. Opening yourself up to understanding is crucial. It may feel very uncomfortable at times.

If you are part of an in-group, it may feel like an attack to hear from someone in an out-group about how we foster cultures that exclude people. If you are white like me, you may not be as attuned to how you might be participating in a racist culture, but that is because much of the dominant (white) culture is geared toward making white people feel comfortable and orienting perspectives around white experiences.

People in an out-group are constantly living through experiences that you as a leader may not understand. In *Whistling Vivaldi: How Stereotypes Affect Us and What We Can Do*, a book I highly recommend, author Claude M. Steele writes:

> *"The problem is that the pressure to disprove a stereotype changes what you are about in a situation. It gives you an additional task. In addition to learning new skills, knowledge, and ways of thinking in a schooling situation, or in addition to trying to perform well in a workplace like the women in the high-tech firms, you are also trying to slay a ghost in the room, the negative stereotype and its allegation about you and your group. You are multitasking, and because the stakes involved are high—survival and success versus failure in an area that is important to you—this multitasking is stressful and distracting. . . . And when you realize that this stressful experience is probably a chronic feature of the setting for you, it can be difficult for you to stay in the setting, to sustain your motivation to succeed*

there. Disproving a stereotype is a Sisyphean task; something you have to do over and over again as long as you are in the domain where the stereotype applies."

I have made mistakes when trying to understand things from a different person's lived experience, and you may have too. As leaders, we must first acknowledge that there is something to learn here. We won't know everything. But we do have to make space for this.

In *How to be an Inclusive Leader: Your Role in Creating Cultures of Belonging Where Everyone Can Thrive*, author Jennifer Brown describes **Stages of Inclusive Leadership Continuum** as:

1. Unaware
2. Aware
3. Active
4. Advocate

You may acknowledge that going from unaware to aware is a major process in and of itself. And helping to advocate for people is something that happens down the line, even if it takes some courage. *Listening* is still a big part of all stages of this spectrum.

This spectrum can be nonlinear for a leader. We may be further down a path at different times in life, or on different dimensions of inclusivity. Perhaps you affiliate yourself with one group, so you find it easier to advocate in that area than in others. However, we can't stop there- ideally you are always working toward making your unconscious biases known to yourself, and eventually moving groups of people away from prejudiced behavior, not just for your group.

You are a leader. It is your responsibility to work toward and demonstrate an inclusive culture. You cannot passively watch and hope it emerges.

This is hard, of course, and will take effort to do right, but trust is also built this way. You can't have a group built on trust if people feel that they are segmented into in-groups and out-groups. **Trust is inclusion.**

Another suggestion: Please don't try to force a conversation about race, LGBTQ+, or any other in-group/out-group dynamic if the person doesn't have the energy for it, or if you haven't built enough trust for them to talk through it safely with you. It's truly not about you. A person has to devote enough mental cycles for this all of the time, sometimes they can feel like they now have to navigate how to care for you too.

TRUST AND BIAS IN ACTION

Consider the following scenario: Your team is hanging out, perhaps chatting in the team chat, and one person makes a homophobic remark. The conversation halts. What do you do?

It can be tempting to hide in these moments. You may think of all sorts of excuses to give yourself: "I wasn't even really participating in the conversation, it would be weird if I said something now." Or, "I'm not gay, I probably shouldn't speak for other people."

The thing is, you are the leader. **The culture of your team is only as strong as the worst behavior it tolerates. It's your job to speak up.** You may not be gay and, you're right, you shouldn't speak from knowledge you don't have. But putting the onus on gay people to stand up for themselves and creating a culture where everyone is on their own is detrimental to people's psychological well-being.

In the above scenario, my suggestion is that you swiftly state very clearly, "Hey, this was a homophobic remark, and we don't do that here. This is a place for everyone to feel included. I know that was probably not your intent, but just want to call out that it is an outcome of these words. We all have bias and work to do, but it's also important to call it out so that we can be a strong team."

Everyone is going to have a different method of approaching a situation like this. I usually try to unpack intent and impact in these moments. Usually, the person doesn't intend harm, and this allows us to have that discussion.

Why did I suggest calling it out in the chat room? Couldn't I have pulled the person aside privately? I have taken that approach at times, but in this situation it's important that everyone sees that this behavior isn't okay.

I mentioned above "We all have bias," because it's true. Even though this person demonstrated behavior that wasn't great, we want to create an environment where it is okay to make our unconscious biases, well, conscious. I believe the more you make understanding bias a part of everyday life, the more you create a culture that can reflect and grow too.

Further Resources

Bias is truly a topic that is worthy of a whole book in and of itself. A number of good resources that cover bias and other related topics in depth:

- *Blind Spot: The Hidden Biases of Good People* (Delacorte Press, 2013), Mahzarin R. Banaji and Anthony G. Greenwald
- *How to be an Inclusive Leader: Your Role in Creating Cultures of Belonging Where Everyone Can Thrive* (Berrett-Koehler Publishers, 2019), Jennifer Brown
- *Whistling Vivaldi: How Stereotypes Affect Us and What We Can Do* (W.W. Norton & Company, 2011), Claude M. Steele
- *The Inclusion Dividend: Why Investing in Diversity & Inclusion Pays Off* (Bibliomotion, 2013), Mark Kaplan and Mason Donovan
- *Subtle Acts of Exclusion: How to Understand, Identify, and Stop Microaggressions* (Berrett-Koehler Publishers, 2020), Tiffany Jana and Michael Baran
- *Diversity in the Workplace: Eye Opening Interviews to Jumpstart Conversations about Identity, Privilege, and Bias* (Rockridge Press, 2020),

I'm writing this from the perspective of someone who is still learning and growing daily in this area. I'm also writing the chapter I wish I had long ago.

EMPLOYEE GOALS

In my experience, it's best if a manager learns an employee's Ultimate Goals early on. Where do they see themselves in five years? What kind of work do they like to do most? What environments do they work in best and which ones are the most difficult? A manager can't always facilitate the ideal situation, but having this information is still extremely valuable for cultivating a person's career trajectory, for the work that needs to be done, and for a general understanding of what will keep people working well together.

Let's say you have two employees: one wants to be a principal architect someday and another tells you that they love refactoring. That actually gives you pretty good insight for a project that requires one person to drive direction and another to refactor the legacy code in preparation for a new feature!

Or let's say you have an engineer on your team who wants to be director someday, but they rarely help others, and you concurrently get an intern. This is your chance to develop one's mentoring skills and scale the other's engineering skills!

CHAPTER 7

CAREER LADDERING

Career laddering is a system that is used to show what the expectations are to develop and progress at different levels of a role. One purpose of career laddering is to define how one might be promoted, but it is ultimately used to drive clarity. This system can take different forms, but tends to be an internal document that states the expectations of a staff member at any given stage of their career.

Laddering is a tool in our toolbox. ***Career ladders alone will not create clarity—as managers, we need to put them into action.*** What's covered in this section is a process that can be used as a framework for putting career laddering into action. Remember though, you can have formalized career laddering in place and still mislead your staff, so it's critical that career laddering documents are just one tool embedded in a deeper process.

Trust factors here too. If you are working with someone on their growth path, and they trust that you will honor it, you enter into a sort of partnership. The qualities that you work on developing together should ideally be things that expand their skill set in the long term and help them grow as an individual as much as it helps the company. Work that's well aligned in this manner has the potential to feel rewarding because there's a larger purpose.

Here is a microsite I set up where I open sourced all of my career ladders: career-ladders.dev

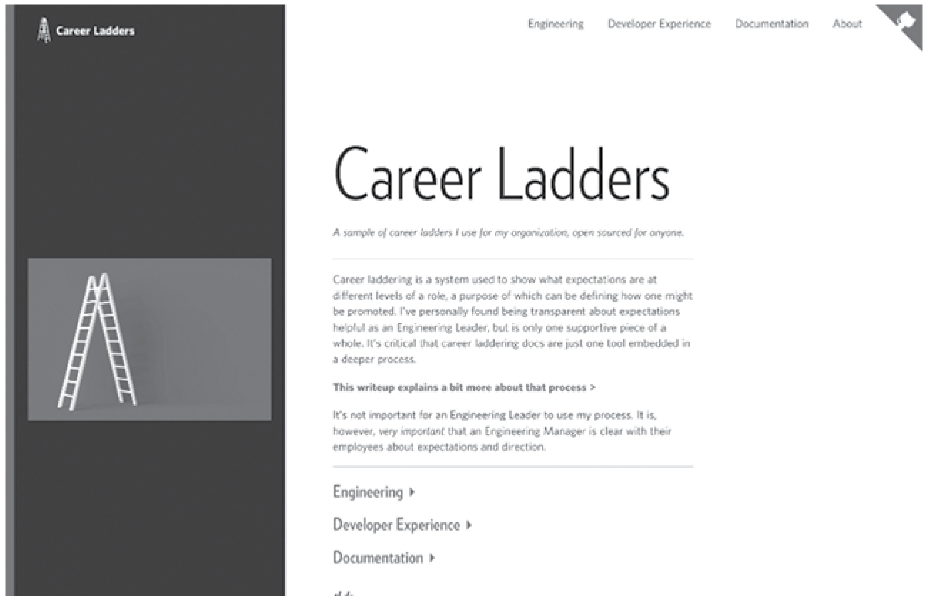

Figure 3: The index page of the Career Ladders microsite.

The site outlines each of the different levels of the engineer, as well as the roles and responsibilities expected at that level. In this particular example, there is a basic concept that ties the whole thing together:

- To get to **"senior,"** you're the best "you" you can be — you perform your role exceedingly well, and you've reached a high potential for your own output.
- To get to **"staff,"** your focus is really to expand beyond yourself. You start teaching people the great things you've learned and help serve their needs.
- To get to **"principal,"** you're creating systems that scale beyond yourself. You're no longer helping folks operate like you — *you're helping them where they are.* A lot of your activities are related to enabling the success of everyone around you.

What I like about this system is that the job of the most advanced folks becomes helping to support and grow other people or systems in such a way that benefits everyone. Principal-level folks don't lord knowledge over others; they work to put the knowledge into practice in a way that's truly helpful.

It's not important that you use *my* exact system, but having clarity about the roles and expectations of each team member can really go a long way. Why? Let's dive in.

PUTTING CAREER LADDERING TO WORK

I mentioned that a career laddering document alone will not help drive an employee's career growth, and I also stressed the importance of clarity. So let's tie this all together and talk through how to use career laddering in practice.

Step One: The Big Picture

I am one of those annoying managers who asks people where they'd like to be in five years. I call it "annoying" because it's a lot to think about. I still ask. I'm not looking for a perfect answer, but hearing this question gives a person an opportunity to consider their future. Usually they tell me *something* I should know.

Here are some examples:

- *"I don't really know what I want to be doing, but I know that I don't want to still be focused on build systems in five years."*
- *"I'm not really sure, because I think I might want to be a manager, even though I haven't tried it yet."*
- *"I want to be able to go camping whenever I feel like it, take my family with me, and work on the road."*
- *"I want to be sure that fellow developers in Africa have every opportunity they want."*

Notice that these answers are not formal outlines for the next five years.

But they give you a sense of people's values, their boundaries, and what we may want to incorporate as part of their working environment. I love it when the qualities you're working on developing with an employee are those that will serve them anywhere, not just within the company. These should be traits that expand their skill set. These types of tasks typically take some long term work, but it can be very rewarding to work together on because there's a larger purpose.

Step Two: Career Laddering Review

> *"If you don't know where you are going, you might wind up someplace else."*
>
> — Yogi Berra, MLB catcher, coach, and manager

In this step, we go through the career laddering doc. I typically have an employee read aloud every list item in their current role to me, then **self-assess the progress they've made on each item**. I sound off a bit. We generally align; people tend to be fair and honest about their progress. I personally think it's important that the employee read their list to me instead of the other way around. There's a sense of ownership that way.

We also go over the next stage in their career and the list items for it. At the end of the process, we break down any common themes. For example:

> *"You're solidly a senior, and doing quite well in your role. To get to staff, you need to be helping others a bit more. Let's make sure you have the time to be available for more PR reviews and pairing in the next few weeks. Let's also talk about getting that internal tool you were building over the line, that will likely help the team move faster."*

Step three: 30/60/90

The next step is called a 30/60/90. The concept is that your employee breaks down the work they want to be doing in thirty, sixty, and ninety days.

I tend to do this with a bit of a twist: **We start with ninety days.** I ask, *"What would you like to accomplish here within the next three months?"* Since the career laddering is fresh in their minds, there's already some guidance on what their focus should include. It's safe at this point to let them drive, and tell you what they should be doing instead of the other way around.

Sometimes this can be quantifiable:

- **Engineering**: *"I would like to close five issues each week, ideally with at least two PRs."*
- **Docs**: *"I would like to address content gaps on two features."*
- **Anyone**: *"I would like to pair with at least two people."*

It can be expressed as a metric:

- *"I would like to help increase adoption of our npm package by 10 percent."*

Or it can be less measurable:

- *"I would like to try to understand our component library a little more as a newcomer."*
- *"I would like to try to interrupt other people less in meetings."*

All of these are valid.

Now that we've taken the time to define the ninety-day plan, we figure out what's doable in a thirty-day period. Again it's up to the employee and what they think they can accomplish in this time. I only chime in if I think they're being too ambitious, or they are missing something that the company needs.

It can also be helpful to state that things change, and nothing in this plan is set in stone — other things may come up that need attention. We adjust, it's no big deal. I honestly don't find the sixty-day piece to be very useful because a lot changes in a month. I usually skip it, but you can absolutely use it if you find it helpful.

During the career laddering process, we also get to talk through what the employee *shouldn't* be doing. If you find that something they are spending time on is not useful, for the employee or the company, you may be able to remove the task and clarify this to other stakeholders. The employees themselves may not be able to have that conversation.

Clarifying the value of an employee's tasks can be incredibly useful for someone who may be an over-performer but are burning out. It helps us align on the tasks that are overextending them, so that they can properly prioritize and focus. It's tempting to think that over-performers need less guidance, but I've found that they tend to need *more* clarity on tasks, not less, in order to define scope and help point them in a good direction.

I've also seen underperformers turn around after the career laddering process. What one might see as a lazy quality in a person might actually be a symptom of misalignment with the purpose of the tasks. A career ladder helps them recognize what, when, why, and how the things they're working on fit into the bigger picture.

One point is critical: If you successfully guide employees on this journey, you need to give them the promotion at the end. The promotion is a change in title and compensation of course. You break the bond of that partnership if you don't follow through. Always give the promotion if that person has earned it on their end.

ITERATION AND REFLECTION

From here, it's probably pretty clear what you do: keep revisiting the list! I try to set a reminder in our 1:1 doc to revisit the 30/90 plan in about a month. When we check in, we see how each task is coming along, putting little checkmarks next to what's done. I'll sometimes put a celebration emoji on something they did particularly well; I believe it's important to celebrate those as successes, even if it makes me sound like Mr. Rogers. Show folks that you appreciate their work and how far they've come.

Then you can carve out another block after the ninety-day mark, so they have direction for the next thirty-to-sixty days. If they didn't finish something, carry it over to the next month. Every few months, we go back and do the career laddering exercise again, but this time denoting the progress that's been made in every area. This is not the only way to provide direction and clarity in a person's work—the sky's the limit. **Anything that provides clarity for your staff can be helpful.**

What I've liked about the career laddering process is that people know where they are, and what it will take to get to the next level. There are no surprises in 360 reviews. The progress is tangible, as everything is documented, and that forms a partnership between both of you. The work is just a unit within something measurable on that path.

When they've fulfilled their end of the deal and completed the things you asked them to work on, it's time to promote them! 🎉 Don't forget to celebrate that as well!

When You Need to Adjust

This process also gives you a framework for working with people if something *doesn't* go well. If you have set clear expectations, both tactically and strategically, you can revisit these expectations down the line if things aren't going well. Again it's best if the employee owns this, so as you talk through line items of what's expected and where they are, ask them about this delta, and see what they say.

Typically I have found that when I build trust with a person, and they are able to own their responsibilities, they are pretty honest about what they are doing well and what they can be doing better, as well as their challenges.

If you see a large enough delta between what you both committed to and what work is being done, it can be tempting to attribute this delta to laziness. A lot of things are misattributed to laziness that have some other root cause, and yet they appear externally as laziness. If someone isn't getting their work done, please do your best to explore further with them before writing them off. If you say someone is "lazy," that is an immovable character assessment and absolves you as a manager from figuring out their motivations, potential misalignments, or challenging external factors that might be impacting their work.

Instead, get curious. Pull back from the tactical and talk about bigger outcomes. The chapter on feedback may be useful when these kinds of conversations get tough. Try to realign by considering the following:

- What are their personal goals?
- What are the company needs?
- Are we able to address the above two with the work being done?
- Ask what they think we could do together to make further progress.
- Restate the expected outcomes and clarify that they can reach those outcomes however they think is best, allowing them to iterate on their tactical plans if necessary.

What's outlined above does tend to work. But in cases where it doesn't work over time, you may have to put the employee on a performance improvement plan, which is fair if you have made your expectations clear and there are no surprises.

Try your best to support this person without ego. Sometimes though, even if you try your best, a successful path just isn't there. It could be a combination of your collective chemistry. It could be that they're dealing with something personal you don't know about. It could be that the company culture isn't aligned with their personal values.

If you're in management long enough, you will likely encounter someone who is harder to manage than others. This is natural! Please don't feel like a failure, I know it can be isolating and a bit scary to be in this situation. Understand that supporting a person is about *the person, not ourselves,* and they may just find a better match elsewhere: in a different department, or even a different company. It happens. Let it go.

SEEING THE TEAM AS AN ECOSYSTEM

It can be tempting to zoom in close to each individual and try to optimize for each one. However, leadership means taking care of the wider system. Leadership means creating wider goals and strategies, and connecting the teams to the greater purpose of their work. It means adjusting team dynamics to suit the widest-possible range of people, and also setting boundaries when individuals try to shift culture to unhealthy dynamics. This may even mean letting go of people if they are doing active damage to everyone around them, and don't respond well to clear feedback. It can be painful to do so, but in the few times I've made this call, the team was immediately healthier and more productive as a result.

We need to care for the team both on the individual level and by stepping back and seeing the team as an ecosystem. You may have noticed a common theme I set for my team: the higher your position, the more you scale yourself to help others. This is done in part so that people actually make sure they're multiplying efforts across the team and not duplicating work, but also to intentionally set up a culture of collaboration.

If the very nature of advancement is making sure everyone around you is supported, people start to look out for one another. They can no longer "hero in" to a situation and expect a reward, they have to take everyone along with them.

I've seen this have great outcomes for team behavior; creating an inclusive environment becomes not just a side task to running the team, it becomes part of the core development of the team.

CHAPTER 8

1:1S ARE CRITICAL

Having 1:1s (one-on-ones) are critical. A 1:1 is a private conversation. This can be a meeting between an engineering manager/lead and their employee, but it can also be between skip levels, peers, or folks in other parts of the organization.

I have been a lead, a manager, and also an independent contributor/software engineer, so I've sat on each side of the table. I've had great experiences on each side, and have also made mistakes on each side.

At their root, 1:1s should reduce uncertainty by making both parties feel more connected to the rest of the team while clarifying intent.

One of the biggest mistakes I see in new managers, again and again, is thinking that 1:1s are optional. This isn't done out of malice, but they see them as taking up a big amount of their time when they could be doing other things. Having been in the employee seat and entirely unsupported, I will say: "**Your 1:1s are the most important thing you do**. This is a hill I will die on."

BETWEEN MANAGER AND EMPLOYEE

Because 1:1s are critical, they should not be the kind of meeting anyone takes lightly, either on the management or employee side. The meetings should have a regular cadence, scheduled either once a week or biweekly and only canceled for pressing circumstances. And if they have to be canceled, it's a good practice to let the other person know why rather than simply removing it from the calendar.

It might be tempting to think remote working means fewer 1:1s, but it's quite the opposite. Since each person is in a different space on a day-to-day basis, 1:1s help make up for sporadic contact by meeting regularly.

One-on-ones should be conducted in a space with the smallest amount of distractions possible. If you are in a room with one other person, shut off your computer and use a notepad so you won't get notifications. If doing a 1:1 remotely, make sure you're in a quiet place (ideally, not a noisy cafe) and that it has stable internet bandwidth. And please, avoid taking 1:1s in a car or while running errands. If you have to be outside during the 1:1, wear headphones. Again, this is all for the benefit of limiting distractions so that everyone's focus is on the meeting itself.

Honestly, I would rather someone cancel on me or push the meeting off until they're in a quiet place than take a call swarming with distractions. Nothing says, "I don't value your time," like multitasking during a 1:1 meeting. **The whole purpose of the 1:1 should be to make the other person feel valuable and connected.**

If you're remote, it might be worth asking the other person which modality works best for the meeting. Some people prefer video chats; some people prefer phone calls. Learning preferred modalities for 1:1s is all part of fostering a better connection.

DIRECTION AND GUIDANCE

One thing that a 1:1 can be useful for is guidance. On a few occasions, I've checked in with an employee who has communicated that they're feeling like they're in over their heads—whether they've overcommitted or have such a tall task in front of them, they're not sure how to proceed and feel anxious to the point of paralysis.

As mentioned, 1:1s provide a great opportunity for a manager to reduce uncertainty and provide clarity. Some ways to do that are as follows:

- **Prioritize**. If there's too much work, spend time talking through the most important pieces, and even perhaps offer yourself as a shield from some of the work.
- **Make action items**. Sometimes a task is too large, and the employee needs help breaking it down into organized pieces. This makes it easier to know where to start on a task and how to move forward.
- **Clarify vision**. People might feel overwhelmed because they don't know *why* they're doing something. If you can communicate the necessity of the work at hand, then the employee can align with the goal of the project and the work can start to feel more rewarding and valuable.

For example, on the employee side, you might not be sure whether to invest your time in Task A or Task B. The progress of your commits slows down as a result. Which task is higher priority? On the manager's side, you might not be sure why commits have slowed. The employee could be stuck on a problem. They could be burnt out. It's tough to be sure. It's totally normal for someone to get stuck once in a while, but it's common to not want to announce it in front of others, perhaps out of fear of embarrassment, among other things. A 1:1 is a good, safe, private place to explore concerns before they become tangible problems.

CONNECTION

Belonging is a core tenet of Maslow's hierarchy of needs[4] because humans are designed for connectedness and kinship. Engineers are no less in need of empathy and human connection than any other person in any other profession.

Connecting with others on a personal level is something I really need to work on. I'm awkward. I'm an introvert. I don't always know how to talk to people. But I do know that there have been plenty of 1:1s where I either felt heard or that I was hearing someone else. In other words, I felt *connected* to the other person, be it through shared goals, personal similarities, or even common gripes.

A friend of mine mentioned that, "People leave managers, not jobs." This is, for the most part, so true! Simply taking the time to develop a connection, where a manager and employee both know each other better, creates a higher level of comfort that can go a long way, and can also improve employee retention.

Employees and managers alike should be fully invested in 1:1 meetings. This means using active body language that shows attention. This also means emphasizing listening and speaking in turn without interrupting the other person.

1:1s Are More for Employees Than Managers

Don't let that headline give you pause. Yes, these meetings are for both parties. They really are. **But in the balance of power, the manager can always speak directly to the employee. The inverse isn't always true.** There are also dynamics between teammates. That means the manager's job in a 1:1 is to provide a space for the employee to speak clearly and freely about concerns, particularly ones that might impact their performance.

Ideally, a manager will do more listening than speaking, but a back and forth dialogue can be healthy too. A 1:1 where a manager is doing most of the speaking is probably the least productive. This isn't team time; it's time to give an employee the floor because it otherwise might not happen. At Google, we have a Chrome extension that you can add to your meetings, which shows the percentage of time you're speaking in the meeting. I love this. I have goals around being under a certain percentage in meetings with my employees, so that I'm not dominating the conversation and it really is *their time*.

One risk here, for the manager, is passive listening. For example, there's a fine line between knowing when to let an employee vent and when that venting needs actionable solutions. Or both! I have no hard rules about when one is needed over the other, and I sometimes get this wrong. This is why eye contact and active listening is important. You'll receive subtle cues from the person that help reveal what is needed in the situation.

It can be helpful to address the listening mode transparently. I say to people, *"I am a 'fix it' person and will default to trying to fix your problems. If you want to vent, that's absolutely fine by me. But you may have to let me know that's what you want, because otherwise I will try to problem solve, and that can be frustrating. Just tell me what you need."*

Figure out your default state, and make it okay for your employees to ask for what they need. When these meetings are focused on the employee instead of the manager, they help the employee feel heard and motivated, which can bolster their career and also give the manager the ability to make bigger decisions about how everyone works together to accomplish their individual and collective goals.

Candid exploration of high-level topics, like career goals, or even low-level topics, like code reviews, are best done with one person in a private space rather than with a full audience out in the open. At their best, 1:1s are a good environment to address some of these topics.

YES, AGENDAS ARE REQUIRED

Even though 1:1s have a tendency to be informal, because everyone already knows each other, they are way more successful when there's an agenda. And no, it's not important for the agendas to be very formal. In fact, sometimes keeping them a bit more open allows for the discourse to also be a bit more open. They could be a couple bullet points on a sheet of paper, or even items added to a private chat channel. **What's most important is that both parties come prepared to talk.**

If the manager and the employee each have agendas, my preference is to either defer priority to the employee, or compare lists up front to prioritize items. It might be that the manager has to discuss something pressing and sensitive, like a team reorg that affects the employee's agenda.

I tend to create a shared private doc with folks I work with where we can both add items, and also store notes we write during the session. Regardless of the doc's structure, *communication is key*. In a best-case scenario, you're both in lock step and all agenda items overlap.

Agendas for Employees

Sometimes, weeks are tough, and it's easy to get frustrated. Taking time to write an agenda keeps the meeting from centering around general frustrations ("I hate everything and how could you have done me so wrong!") and more focused on actionable items. Why not just vent? Sure, there's a time and place for venting, but the problem with it is that your manager might not know exactly how to help you on an emotional level. Having specific topics and items listed on an agenda facilitate more actionable feedback for your manager and therefore make them better able to support you.

Agendas for Managers

Let's face it, you're probably juggling a million plates. (That metaphor might be wrong, but you catch my drift.) There's a lot on your mind, and most of it is confidential. An agenda gives you the context you need to prevent wandering into topics you might not be at liberty to discuss. It also keeps things on track. Are there four more things you need to cover and you're already fifteen minutes into a thirty-minute meeting? A concise agenda makes you less likely to pontificate about your early career or foray into irrelevant paths, and encourages you to stay focused on the task and the human right in front of you.

MEETING ADJOURNED . . .

You spend many waking hours at work. It's important that your working relationships, particularly between manager and employee, are healthy and that you're intentionally checking in with purpose, both in the short-term and the long-term.

One-on-ones may appear to be time hogs on the calendar, but over the long haul, you'll find they save valuable time. As a manager, having a team of employees who feel valued, aligned, and connected is about the best thing you can ask for. So value them, because you'll get solid value in return.

4. https://www.simplypsychology.org/maslow.html

PART 2

CHAPTER 9

COMMUNICATING AS A MANAGER

For better or worse, communication is the most important part of our work as managers. Part 2 of the book dives deep into very specific conduits for communication. Before we do this deep dive, let's look at the big picture of communicating as a manager.

LEADING WITHOUT DRIVING

The more you manage, and the bigger the teams you manage, the more you have to focus on outcomes. This can feel unnatural. If you're a good engineer, you want to dive in. You may even know exactly how to dive in and communicating what you want will take longer than teaching your team. But you have to stop yourself from the short path of doing it yourself, and slow down to teach people to fish. The effort and scale will be recovered, and then some, over time. This is much easier said than done, and I've made this mistake more than once.

Your job is now to align people to the *outcomes* instead of tactical details of *how* to get something done. The *how* is up to them. It's okay to make suggestions when they're stuck. It's okay to align on a reasonable delivery date for these outcomes. It's expected that you talk about requirements for the project, be curious and gain context for the work, and offer suggestions.

This is so tough! As an engineer, I have Opinions™. Even if I try not to, I have strong preferences about development. At the same time that I was writing this book, I was managing many different framework teams at Google (including Angular), among other web infrastructure groups. I am a former Vue core team member. I was an active member of the React community before that, giving keynotes at events like React Rally. It is so tempting for me to lead with my opinion when we work on these frameworks, as I have plenty.

Instead I have to **ask questions** of these very smart, very capable teams who know their space and framework well. When we talk about strategy, I ask questions like, "Is this API definition the best we can offer our users? We are getting feedback that it should be revisited." Or, "Is this the authoring format that will carry this framework forward for the next ten years?"

The answer to that question is up to the team.

More than anything else, ***we have to be clear.*** At times we have to provide clarity where none exists. This means we also have to do some preparation before the communication happens so that we know what direction we mean to convey. We have to do our best to think through the problem, the given solution paths that might be available, and the tradeoffs for each. All software has some degree of tradeoffs, which is why "it depends" becomes a more and more prominent statement as people gain experience.

There was a time a while ago when I asked my team to do something, and none of them did it. What happened there? Given that it's a team of highly efficient, strong collaborators, do you think they just all table-flipped and didn't take action? Not a chance. I was the one who wasn't clear. In fact, if a whole group of people don't understand or take action, the chance is that you, the manager, are the common denominator for why something is blocked.

Alignment with the *Why*

In the scenario I just described, part of the problem was that I didn't align anyone with the *why* (the purpose) of the task. It's pretty easy to forget, or not prioritize, doing something if you have no clue why you're doing it. Repeat yourself and align the group with the importance of the task, and you'll likely have a better result.

If you can properly address the mission, the goal, the reason behind something, you will find a lot of passionate engineers will follow you to the end state of the journey.

REPETITION

No one likes to feel like they're repeating themselves. It's annoying to say something more than once, and it's annoying to hear something over and over again. But if you have a big enough group and there's enough going on, things are going to slip through the cracks, so repetition becomes an important tool to make things stick. **The trick is to convey the same message, but in different ways.**

Think of all the methods we have to communicate these days: chats, emails, video meetings, texts, document comments, and so much more. Because some people communicate better in one medium than another, using all of these platforms becomes a strategy for repetition without nagging.

You may have even noticed times when I repeated myself in this very book. *It's all a part of my master plan.*

WRITING AND SPEAKING

The higher you go in management, the more you have to convey ideas to large groups of people at once, whether it be your teams, stakeholders, or peers. This is why written and verbal communication become crucial skills to develop to align groups to a common objective.

When writing, you own the material. It's a thinking exercise as much as a writing exercise. It forces you to work through the problem as you attempt to translate the information in your head onto paper.

The act of reading, on the other hand, is an entirely different experience. A person shifts context from whatever they were doing into parsing the document. They need to spend time understanding the intent of the writer, using formatting and style as cues along the way. They do not “own” the material; they are adjusting their own mental position to try to accommodate the stance of the writer, often skimming quickly as they do so.

It can be tempting to assume that once you’ve written or said something, you have effectively communicated. In an ideal state, that would be true. However, in day-to-day life, this is not always what occurs. There are some steps we can take for better communication.

Think of it from the audience perspective

The single most important thing we can do to write better documents and give better speeches is to write them first, and then reread them from the audience’s perspective. It seems like such an obvious thing to do, but doesn’t happen enough in practice.

As you reread, consider the following:

- What would I think if I watched or read this without context?
- What would I need to know right away?
- What are the key questions I might have and how can I quickly resolve them?

Does the audience need to know who is involved? Do they need a road map? Do they need justifications?

After this reflection, it’s pertinent that you edit diligently. Restructure your draft, let go of anything that is crowding your message. Rewrite anything that is potentially confusing.

Format Things Clearly

Your audience's attention is valuable; the chances are high that they are processing as quickly as possible so that they can move on. This makes formatting extremely important. Take a moment to structure the document or slides so that the headings and outline are expressing the most important points; and if someone needs to jump to a section, they can get the information they need quickly. Don't overwhelm slides or written material with a ton of information; your audience will try to look for patterns in the noise if you do so. You might as well declutter and make it easier for them to find what they need.

It's a good idea to put the author and last edited date on your materials. This allows anyone who comes upon them without you to gain a sense of who you are and when this document was important.

Be Careful With Your Words

Like it or not, when you enter leadership roles, your words have weight. Something that has seemed like harmless venting in the past now has the potential to create large rifts in your organization. Acknowledging issues is important, but even the slightest misstep can demoralize people.

Be careful with your words. Be careful when representing yourself as the only person who can guide your team. Be careful to give credit when it's due. Be careful to celebrate successes and be sober (but strategic) when talking about problems, without blame.

There is no single way to lead a team. Spend some time thinking through the kind of leader you want to be, and take action. Think through what values you want to uphold on a team and lead by example.

You're also going to mess up sometimes (I know I have), and that's okay. Try to learn from your mistakes so that you don't repeat them, but offer yourself some forgiveness too.

Transparency

People often say they "value transparency." That's cool. It doesn't really reflect how complicated this statement is though. I think they mean that they don't want to see others withholding information. But there are many degrees of transparency, all very nuanced, so let's unpack it together.

Transparency is one of the most important ways to connect with your teams. If they don't feel like you're telling the full story, trust erodes. On the other hand, if you're someone who will cut to the chase and be real with your team, that honesty is part of a solid foundation on which to build amazing things together.

When communicating, it's worth asking yourself what context people need to see the bigger picture. Erring on the side of saying more, not less, can help build trust. Try to speak in plain English rather than disguising messages in corporate blather. You hired pretty smart people. More than likely, they can handle and process information along with you, especially if you provide enough background—and if many sides, not just your own, are addressed.

People are looking for different amounts of transparency from you as a leader, and it's okay to ask them how much depth they want as well. Sometimes I'm surprised by the answer.

A note of caution: Some things that may look like transparency are actually toxic. Think gossip, manipulation, and unproductive venting that pits one person or group against another. This may seem obvious, but it can be hard to reflect in the moment. I've certainly made this mistake, especially when stressed. But when transparency can be a way of honestly connecting with someone, where's the line?

I try my best to draw it at: What can I say that would not embarrass me if another person or group heard it? Have I given the party mentioned this feedback directly? Being human and honest is good. Trashing other people is not. My suggestion is to be as honest as you can without negatively impacting others.

WRAPPING UP

As engineering managers, our jobs become less and less about writing code, and more about writing the expression of large strategies or sets of ideas. Clarity is a critical part of our roles, and providing clarity is a sign that we respect our team's time and energy.

Rather than thinking of how you yourself can move things forward, think about aligning people with *why* we're headed down a particular road, and what the desired outcome should be.

CHAPTER 10

CHANGE MANAGEMENT

"It's not the strongest or most intelligent who will survive, but those who can best manage change."

—Charles Darwin

One of the hardest challenges we have as managers is to guide groups of people through change. When managed poorly, change can leave folks feeling like the ground beneath them is shifting and they have no path to success. However, when managed well, change can be the most pivotal force of transformation for good.

Change for the better is not just possible, but absolutely necessary. A good leader cannot sit and watch things carry on as usual without sorting out how transformation can occur. In all of my years in this industry, across companies large and small, I have never seen a team that doesn't need to change and evolve over time. In fact, even though change is hard, I have seen more damage done in situations where people *failed* to change than those who made mistakes during change. In an industry that moves as quickly as technology, you have more to lose by staying still.

You're not just in this role to uphold the status quo. You are there to *improve* the organization: culture, process, systems, and products. As engineering leader Marc Hedlund wrote on Twitter:

"Make things better. That's the job."

THE BIGGEST LYNCHPIN: CULTURE

No change will take root unless the organizational culture supports it. When change fails, we look at the base outcome: The team didn't go in this direction, the product wasn't built, the organizational changes were so hated that nothing got done.

But those are only symptoms of the underlying problem: The real problem is people. People weren't aligned. People weren't supported. People weren't bought in. **In order to make transformation, the culture has to shift to allow for it.** This is why culture eats strategy for breakfast.

A Personal Story

I once joined a team that was used to saying no. They had to because they had to support a ton of customers and infrastructure, and the potential for scope creep was everywhere. In a way, saying, "No," was a tool to keep the team healthy and marching toward reasonable goals.

The problem was, we did have to evolve. Tech doesn't stay in one place and, in order to create the right products that were worth maintaining, we had to start thinking about change over time.

I am not exaggerating when I say that in every other meeting I went to, I heard the phrase "It's impossible." It bothered me. First of all, nothing is impossible. Everything has risks and tradeoffs. Some things are worth pursuing and some are not, but we could never get to that conversation because even evaluation of our current situation was shut down by this statement. Some folks here and there were curious and wanted to see us evolve, but because these pessimistic attitudes were pervasive, any chance of evolution would get swept away by the current.

The first thing I did was think about what values one has to have while learning, and ended up with "curiosity" and "humility." I started slowly noticing those who seemed to imbue these qualities, and then I got to know them as much as I could. We all started talking excitedly about the future. We all started working together, and gathering and promoting projects that showed these qualities. We all started beating the drum of why evolution was important.

We did this in large speeches, in small team meetings, and we did this in 1:1s. We found partners in TPM (technical project management), in PM (product management), and in strategic partners and customers. We wrote documents and got feedback.

I am summarizing here, but it's very important that you know that **this was challenging, and took many months of consistent work.** There were times when my confidence was shattered; I was ready to throw up my hands and just let the group carry on with the status quo, but other members of the team reminded me of the purpose of our work, and I'm very grateful.

Over time though, people started representing evolutionary goals without that core group of naysayers. It started small and was almost imperceptible. But then people began framing future designs around these concepts. And when I eventually proposed an R&D unit of the organization, people were aligned. It seemed like it just fit the overall strategy that we were all acting on, rather than a top-down reorganization.

We made changes to our products, our OKRs, and even the structure of our organization. **But the cultural change was first.**

WORDS OF ADVICE . . .

In order to make real change, we need to connect people to the *why*. The change is made to address something. What is that? In the experience I talked about above, the team wasn't aligned with "We need an R&D function" or "We need to change our process." They were aligned with the shared vision that we needed to evolve our products.

Seek to understand. While we communicated this shared vision, we also talked through risks, we talked through things from different sides, we weighed outcomes. You have to do this too. It's not enough to tell people what to do and never listen to their thoughts and opinions. Remember, the people you hired are smart. They are close to the metal. You need to trust them to tell you the truth, and tell you things you might not have considered. You still need to set direction at the end of the day. But your direction will be clearer and more tangible if you have understood the whole problem space, just like a technical problem.

Be specific about why the changes are made. Cover many parts of the reasoning for change, because this will resonate over a spectrum of beliefs and points of view. Address tradeoffs too, so that people know you have taken them into account. Be transparent (See previous chapter). In this process, it can be tempting to talk down the old way of doing things. I have made this mistake. Don't do this, because you risk people feeling like you don't value their previous accomplishments, and that you don't understand that people tried their best, given that context. Be forward looking and honest about problems you're addressing, but don't brush off past work.

Here's something I don't hear people address very often: When we're talking about large-scale changes that take a lot of time and effort, **you may feel a crisis of confidence at some point.** Your team was doing fine. And here you are, introducing all these changes. *Is that right?* If you have a strong vision, you are the catalyst for the change. You are the spark for the fire. And hopefully at this point, people are bought into the vision, because then it's bigger than you. A good vision takes off on its own, and is bigger than your own ego, bigger than your own fear.

Support people to grow the skills they need in order to succeed. In the example above, I set up pairing across the organization for specific skill sets people needed help with in order to evolve our products. Whatever team you're working with has developed a skill set to be successful in the current organization. If you're asking them to evolve, you need to plan for that growth and make room for them to develop new skills to succeed.

Be accountable. Your new concepts and vision might come under fire. Some things may fail. No innovation comes without some failures. When this happens, you need to be accountable. You can't blame the team. You can't blame the process. You have to own the situation, and you have to acknowledge that as the leader you are accountable. The quickest way to lose trust is to send a team out on a journey and not back them up when the going gets tough. If they don't succeed, that's on you. **Have their backs.**

WORDS OF CAUTION . . .

The problems I've seen in change management again and again have taken similar shape. Let's address those, so that we can avoid them.

Leaders take too long to act, or act too brashly. Yes, I know, that's hypocritical. But different types of change management require different speeds. Organizational changes that involve reporting structures tend to work best when the planning is up front as much as possible, and then the change happens fast. Being told you'll have a new manager and then not getting one for months is confusing and stressful, aligned or not.

But if an organization has a hard time with change and you change all their processes overnight, that can be harsh for people to consume. Process changes shouldn't be drawn out, either, but you may have to hear many perspectives and introduce things slowly to not completely derail people's work and understanding of the system they're working within.

Focusing on the short term versus the long term. As mentioned earlier, focusing on the vision is the most important thing. A trap we can easily fall into is talking about the end state and the solutions without talking about the problems we're trying to solve. By the way, I have also seen people talk too much about the vision while others sit trying to figure out if they are going to be reorganized under different management. You need to be clear about the vision *and* the tactics. Respect your audience, and think about what information you might need, if you were in their shoes.

Perception of unfair decision-making. This typically happens when you haven't done the "seek to understand" step above, or when you don't talk through the risks when communicating the new vision. If the decision-making process is completely opaque, sometimes people feel like there's something you're hiding or that there is underlying malice in the decisions being made. Be as open and honest as possible, while being firm. Offer to talk it out 1:1 with people afterward, if there are concerns. **Respect the back channel**. If it's on fire, something wasn't communicated well.

WRAPPING UP

Change is challenging, but necessary. It can be hard to navigate, but if you solidify your larger vision and goal, and enable the team, you can do it.

In engineering work, we make changes and get to see the results of that change immediately. "Yes! It compiled!" In engineering management, the cycles can be much longer, have more parameters, have more risk. But they also have the potential for a much grander scale of impact. You are there to help everyone be better: build better products, have better organizations.

You've got this.

CHAPTER 11

GIVING FEEDBACK

You hold a team meeting. You are about to propose shifting directions on a project that's many months in. It's a serious overhaul, but one that's appropriate given the longevity of the project and the technical direction. You're nervous, and a little scared. This represents more work for the team. This means they'll have to realign with stakeholders. This means we all have to question our base assumptions, which is a tough exercise. But it's also the most responsible thing to do for the health of the team and project. You open the meeting link, and talk into the camera. You look at the faces that appear on your screen. They look sympathetic, understanding, and thoughtful.

This is feedback. Feedback can exist as an important discussion between two people, as comments on a document, and as almost imperceptible emotions shown on a person's face as you convey information.

This chapter suggests a framework for being more intentional when giving feedback. I've seen feedback turn an unhappy, low performer into a happy, high performer, and I've also seen it ruin relationships when done poorly. You probably have too. We focus more on critical feedback than positive feedback, because critical feedback is tougher and the stakes are higher when it goes wrong. Thus it requires more exploration and thought.

Feedback is difficult in part because it's not objective. It's part of a process that uses our own and other people's judgment, which means that it can be messy and non-deterministic. There's a lot we have to untangle to do it well. How can you tell if you're helping? How can you tell if you have blind spots?

Unfortunately, feedback isn't easy, which is why there are two full chapters devoted to it. Truly, there are many books on feedback as well as myriad studies. I've cited them here and incorporated my own experiences, but there's so much to discover on the subject. I feel that I'm always learning in this area.

WHY IS FEEDBACK IMPORTANT?

Many of us want to find purpose in our work. We're there for most of our waking hours, right? We can steer our own ship to some extent, but at some point we can't always see our own work clearly, and we need input from others to improve.

We need to be clear with our team what they should be focusing on so that they can put their efforts where it's most useful. Ideally, we are clear about priorities from the start, but it sometimes takes a feedback process to gently provide guidance over time.

We should also give *and receive* feedback from those around us to keep the team working together toward the same goals. No one is working in a silo. We all have to adjust a little to one another's communication style in order to make our working environment happy, respectful, and productive.

In terms of feedback, there is a bit of natural tension within our own egos. We want to learn and grow, but we also want to feel accepted. We need feedback so that we can hone our focus and challenge ourselves to get better. But in order to truly hear this feedback, we have to work around internal systems at play that are designed to protect ourselves, which can be tricky.

My suggestion is to treat all feedback exchanges, as much as possible, as though they are a **partnership**. You are not coming down from a mountaintop of perfection to give advice; rather you're two people working together toward shared goals, and feedback is part of making that partnership healthy and productive. Approach this more like, "Hey, you have stuff in your teeth." A trusted heads-up, something you know the person may need to hear, but can't see for themselves. Be sure to convey that you're not judging them in this process.

CHALLENGES

Feedback that is perceived as negative can be tough to give because you're in essence trying to navigate against a person's natural neurological response. Elena Shackelford, director of global talent and development practices at Nike (who also holds a PhD in industrial-organizational psychology), wrote[5]:

> *"The fundamental organizing principle of the brain is a 'minimize danger, maximize reward' function that is designed to keep us safe from anything perceived as a threat. When we feel threatened, we shut down, and that's what happens when we're given feedback in an ineffective way. . .*
>
> *The **amygdala** is the key part of our brain that receives information through our senses and translates these into emotions. When our brain perceives a threat, our amygdala is hijacked in order to avoid that threat, leading to **reduced analytical thinking, creative insight, and problem-solving**. Giving feedback the wrong way can send the receiver into a threat response, where they can't properly internalize or respond to the feedback you're giving them."*

In essence, to give good feedback that actually helps an individual or team grow, it's pertinent that they **don't feel threatened** and feel that it's coming from a place of care.

Have I gotten this wrong? Oh, you betcha. And it damn near destroyed me. If you make this mistake too, it's one that you'll try hard not to repeat again.

Unfortunately, humans are not pure functions. What you put in one day will not necessarily resolve to the same output on another day. We're dealing with so many other issues concurrently: family, emotions, hormones, outside circumstances, our whole life's work before this; and when it comes to receiving feedback, all of these other factors might come into play.

All of these challenges and risks might have you feeling a little gun-shy about giving and receiving feedback. That's actually okay, because it's pertinent that feedback be done carefully and thoughtfully.

TEAM DYNAMICS

I have found feedback less challenging to give and receive when there is mutual trust. We talked about how to build trust in earlier chapters. Some other things you can do in your day-to-day work can cultivate a learning culture:

- Ask your team members how they prefer to get feedback, and listen to the answer. The more they are in tune with their own state of mind, the better they think critically and communicate.
- Point out positive growth on the team and give accolades for it in public. By doing this, folks will know you really mean it when you say you want them to learn and improve.
- On the other side of the spectrum, don't ever expose someone's growth in a way that belittles them. Don't tolerate this or any kind of bullying from your team members, either. This will stifle growth and trust.

Remember, as my coach Jessi says, people are often driven by either love or fear. As a leader, we'd like our teams' decision-making to be driven by love rather than fear—not just for hippie reasons. As mentioned before, fear-based decisions tend not to be the most logical. I've seen fear-based decisions result in very poor business outcomes indeed.

A PROCESS

A typical process for giving feedback could look like this:

1. Do the pre-work of hearing how your team member likes to get feedback.
2. Make sure the feedback is necessary.
3. Check in with your own motivations and biases.
4. Give them specific feedback *as soon as you can,* framing it in a way that they may best be able to hear it.
5. Keep the conversation open, and allow them to ask questions.
6. Talk through expectations and what could be better in the future. If your feedback is related to an HR violation, be *very* clear about next steps and follow-up.

All situations are slightly different, so you may need to go down slightly different paths—make sure you alter things based on what the situation calls for. Always be aware that you're dealing with a human, not a machine, so there will not always be one right solution.

PRE-WORK

In thinking through your own and your team's motivations, it can be helpful to reflect on the earlier values exercise. Some of our triggers and responses to feedback can be related to our values or share context with values.

I started doing an exercise with my team where I ask the group as a whole how they prefer to receive feedback, as the way one person gives feedback may not be the way they prefer to receive it. Not only does this practice open up ideas, it also encourages each individual to think for themselves about how they prefer to receive feedback. Normalizing this type of vulnerability and self-reflection can help us all feel like partners, instead of dealing with some top-down edict.

We can try an exercise that connects one's values to the way in which they prefer to receive feedback. (Remember to go first, unless someone on the team wants to start.) The statement might sound something like this:

> *"I find I need x because of my values of y."*

Here's an example of this in practice:

> *"I find that I need feedback to be a little more gentle, and it helps for it to be related to something else I did well. This is because one of my values is respect, because I've had some bad experiences with a lack of respect at other jobs."*

Or:

> *"I prefer my feedback to be very direct. One of my values is honesty, because being direct was how my parents showed affection when I was a child. Please don't feel you have to add any caveats to even negative feedback."*

Neither of these examples are "wrong." Everyone's a little different. You can leave it up to the team if they are comfortable tying the value to the feedback preference. For some, that's too intimate. But I encourage having at least the feedback statement be something everyone participates in.

The truth is that even my advice here is imperfect. Feedback is tough. Being honest and improving together as a team is awkward. It's incredibly worth it though. That's where the real growth is. That said, no two people are alike, no two groups are alike, and you may have to use your best judgment given the situation at hand.

MENTAL STATE

Before giving feedback, it can be a good idea to check in with your own and the other person's mental state as much as you can. In their book *Thanks for the Feedback*, Douglas Stone and Sheila Heen discuss a few factors to consider when giving feedback:

- **Your baseline**: This is your default emotional state. Some people are more naturally optimistic, others feel more general anxiety.
- **Your swing**: This is how much you move off your baseline when you receive feedback. Some people react more strongly than others to either positive or negative feedback.
- **Your recovery**: This is the duration of your reaction. Some people bounce back from setbacks faster than others. Some people maintain a positive boost for longer.

Try your best to give feedback when people are in as good a mental state as possible, but also as soon as possible. Ideally, you're not looking for the sun, moon, and stars to align, but to deliver feedback within the next day or so of an event, when you and the person are in the most-focused mindset you can be. Then you can get down to work and have a healthy discussion.

CHECKING IN

We already discussed the pre-work before giving feedback, but there's one more thing we have to do: we have to check in with ourselves.

Is the feedback entirely necessary? Or as my friend Sarah Ratchye would say, "What's the upside?"

Are you giving feedback to truly see this team member grow? What's the best possible scenario that can come out of this? If the answer is, "They have better tools for communicating with their coworkers," then you are probably coming from a place of care.

If on the other hand, the answer is "Because I don't like X," you may have some more digging to do within yourself even before talking to the other person. Why does X upset you? Is it a *you* thing or a *them* thing? If it's a *you* thing, is it something you should try to work out on your own? Feedback is a tool and can be weaponized for our own insecurities and blind spots if we're not thinking things through.

In terms of blind spots, unfortunately bias may also come into play when delivering feedback. Race, gender, and myriad other pieces of our outer identity may cause us to be inconsistent in our delivery of feedback. We tend to like people more if they are like us[6]. This is not great news for a working environment where you as a manager have to be a human while also navigating around your own biases.

That said, in order to check oneself, we can try to first think through a few things. Would you give the feedback at all or in the same way if this person looked like you? (Really try to be honest with yourself here.) What about if this person were a member of an in-group, such as a tall, white, male? Would your feedback be the same? Would it be expressed the same way?

Really check in here. It costs you nothing to do so. Sometimes the answer is that you have been inconsistent. We all have biases.[7] In fact, those who think they have none can risk driving biases deeper because they are not self-evaluating.

Think it through and adjust. Try to treat people equally across the board and hold yourself accountable to being even across your teams.

GIVING THE FEEDBACK

There's no one right way to give feedback, but here are a few guidelines:

- **Go back to the mental state:** Is this person in a good mental state to hear this, and are you in a good mental state to handle the conversation on your end? If not, and you still need to deliver the feedback, how can you adjust the tone and setting to be accommodating?

- **Is the feedback immediate?** All of the pre-work aside, you should ideally give feedback as close to the event that warrants it as possible. It can be disheartening and even embarrassing to find out in a performance review six months after an event occurred that you should have been doing something differently.
- **Are you remembering to give them feedback in their preferred method?** If they asked for gentle feedback, remember to touch on positive things they've been doing as well.
- **Keep the conversation open.** Allow them to ask questions while they process the information. They may need to understand some nuances to fully engage with the information. This dialogue can also help calm down the flight-or-fight part of our brain.
- **Be specific, and speak to your own experience**. "You are X" is an ad hominem attack on the other person's character. Consider instead, "I felt X this happened." They can't argue with the way something made you feel, and by providing a specific example, you are talking through a specific action rather than condemning their character.
- **Feedback is more helpful if you can really nail down what exactly could have been improved and why**. Align them with a vision for why this feedback is important. Not to be harsh, but if you don't know the answer to this, the feedback is probably unnecessary.
- **You can even connect the feedback to their values and goals**. For instance, if they value decisiveness and innovation, but they took so long to complete a project that the audience lost interest, the

feedback you are giving them can be in service of their own values over time.

I think it's okay to admit to this person that it's hard for you to give them critical feedback. By showing vulnerability, you make space for them to be honest and imperfect too. I think this is the best base state in which to create kindness and sympathy on both sides.

Next Steps

After you've delivered the feedback, give guidance for the future. Without guidance, people may feel like they're left on their own to navigate the problem. If you're worried about being too prescriptive, you could also stop short of making a recommendation and ask, "Is it okay to make a suggestion?"

For example:

> *"Your presentation today was excellent, but the intro was too long. It's a bit more respectful of your audience to remember who they are too. Twenty-plus autobiographical slides for an audience of people who you work with is time that could have been spent on the content itself, because they already know who you are. Next time, if it's an internal presentation, perhaps just a slide or two for setup would be a better balance. You're so good at giving talks, I want to make sure people can hear your message clearly."*

Is better than:

> *"You talk about yourself too much."*

Note that giving details also means expanding on the *why*: Why is this information helpful to them? But details also help you steer away from an ad hominem attack, or from hijacking their amygdala and hindering their ability to think logically.

In her book *Radical Candor*, Kim Scott mentions that "caring personally doesn't mean personalizing." The second example above is so broad, it feels like you're making a judgment on their character rather than pinpointing a particular thing that can be improved.

Ideally the feedback we give is in service to the person: It's necessary, it's thoughtful, and it helps them improve at work. I like to think through whether the feedback would serve them anywhere: **Is this something they will be able to carry in their careers? Will this help them level up?** Is this something that will help them even if they were in a different job? If the answer to these is "Yes," chances are it's feedback we're giving in service to them as a person.

Whew, moving on . . .

PRAISE IN PUBLIC, FEEDBACK IN PRIVATE

We've covered trust and vulnerability, and how feedback can help or hurt the cultivation of both. You've probably heard the saying "Praise in public, feedback in private." I still believe in this statement, but it is actually more complicated than one might think. So let's spend a moment on it.

The crux of the idea is this: When you're celebrating someone, be as loud about their accomplishments as you can so they feel valued. But harder feedback can feel embarrassing if done in public. Why is this idea so complicated?

First, you've seen how many parameters we are dealing with in simply assessing how and when and whether to give the feedback at all. This compounds with more people.

Second, it's much easier to hijack a person's amygdala and put them into a fight or flight state if they feel judged. This can happen when giving feedback. Now imagine this scenario in a public setting, versus delivering the feedback in a 1:1. Not only can that feel bad for the individual, but it can have negative impacts on other team dynamics, creating a culture where people don't feel safe around one another. Feeling safe in a team environment is important because psychological safety in general tends to help people feel happier and more productive.

COMPANY-WIDE FEEDBACK PROCESSES

Some companies have a review process during the year, which is great. Remember not to wait for those time periods to give feedback, though. If something happens, deal with it immediately while it's fresh in everyone's minds. It's best that the review process reflects and captures major themes that your team members already know.

If your team has a season of feedback, be sure to build in breaks and self-care times ahead of that time, so that everyone has mental stamina for what's ahead. Model this behavior, as well. There's nothing worse than trying to have a hard, heartfelt conversation on low sleep or low energy.

What if multiple people on a team need feedback?

Did a whole project go off the rails with an entire group of people? What's the common denominator there?

Hint: It's you.

There have been times where I said, "How can they all have done this?!" A whole team of people doesn't misunderstand a direction—you as a leader weren't clear, as we touched on earlier.

In this situation, you don't start by giving feedback, you start by *asking for feedback*. Typically your team will be able to give you some context for why things went awry and what you could have done better. Your willingness to hear their thoughts on the matter generally opens up a larger conversation for what could be done better next time. I also find that people are willing to own up to things themselves in this framing, because you've shown it's safe to do so. Which leads to . . .

While feedback is an important part of a healthy, growing team, it has to be executed thoughtfully to bring out real benefits, both for the individual and for the group. There's no better way to grow than learning to internalize what others can see clearly which you sometimes cannot. We can also learn to have a healthy relationship with our own egos—one that is resilient but also allows us to set aside our egos.

ESCALATED FEEDBACK

Let's talk about the really hard stuff, because we have to.

There are, of course, times when you don't have a choice but to deliver hard feedback. If someone on your team has an HR violation for instance, you need to be more firm with expectations. Work with your "people" team in advance to talk through the issue and understand how to handle it. Sometimes if a person is doing harm to one or many people, it may have legal implications that you are not aware of.

Or there may be situations where many people on the team are reporting having issues with the same person. In those situations, feedback is less optional. You have to ensure a good working environment for the greater group, and that involves working with an individual in an actionable way to understand the impact of their behavior on the team.

Even in those moments, you should still be approaching the situation with care for the individual. They need to know that there are boundaries in a working environment and what is expected of them in the future, with clarity. Anything else is a disservice to them.

Ask Questions

During the call, it is helpful to be as specific as possible because you can unpack what happened, and why there was a negative impact. Here's an example of such questions to ask a team member when giving hard feedback:

"What were you trying to say here?"
"How do you think this sounded to the person that heard it?"

"Is this the impact you were aiming for?"
"How could this go better next time?"

Note that you're allowing them to talk through their perspective of the events, while also soliciting them to think through what the other person might be thinking and feeling, which they may not be processing. If they don't come to the right conclusions, you'll have to guide them there.

When peers in management talk about the heartbreak of their jobs, they usually talk about this part of it. People don't usually mean to hurt others. And managers typically take no personal happiness in this part of their jobs.

But if issues keep happening with team members, even after workshopping and providing feedback, you need to work on a growth document together with specific expectations on what they need to improve on. A next phase could be a PIP (performance improvement plan) with the people team, and then eventually separating ways. You're in charge of a system of people, you need to make sure you are taking the wider group's health and safety seriously. It sucks, I know.

In terms of feedback, there's a lot to be mindful of, and so many benefits too. So far we've talked about giving feedback, but we haven't talked about receiving it. Let's talk about that next.

Record the facts

If you have a conversation that goes off the rails, I'd suggest spending a bit of time after you're off the call to write down what you think happened.

I tend to give myself a section to just write through the **facts** of what happened, and then another to write through my **feelings** of what went poorly and what could have been better. It helps to check in with the facts separately because our human brains can sometimes try to protect us by amplifying a particular version of an event. This can be hard to do, but checking in with just the facts helps ground that a bit.

5. https://www.bluebeyondconsulting.com/2018/11/this-is-your-brain-on-feedback-how-understanding-a-little-brain-science-can-make-a-big-difference-in-your-next-feedback-conversation/

6. https://www.psychologytoday.com/us/blog/close-encounters/201812/why-do-we-people-who-are-similar-us

7. https://ed.stanford.edu/sites/default/files/uhlmann_et_2005.pdf

CHAPTER 12

RECEIVING FEEDBACK

As a manager, you can also ask for feedback, so that you can grow, iterate, and evolve your ability to lead and help your team. A nice side effect of this practice is that it cultivates a culture in which growth and learning are valued and seen as a natural parts of work. It can move feedback away from a place of fear and into that place of care we were talking about earlier.

Receiving feedback can be difficult, but it's also the best way to grow, and the best way to demonstrate the kind of vulnerability you will need to cultivate to have honest communication on your team. Because receiving feedback is so vital, you have to be careful to ask for it when you are mentally equipped to receive it. In this chapter, we cover some tools for how to ask for and process feedback. Please note that receiving feedback is intrinsically tied to how well you take care of yourself and build your resilience as a leader, which we cover further in Part 4, "Your Work."

The more you listen to what areas you can grow and address them well, the easier it gets to do it again. It will likely feel uncomfortable for you the first time, not unlike the first time a programmer writes a for loop. Over time it becomes more natural and can be seen as an opportunity, and a kindness the other person is offering you.

SOLICITING FEEDBACK TO GROW AS A LEADER

There are many ways we can solicit feedback, here are a few examples:

- **Send out an anonymous feedback form**. This can allow people to be honest without fear of retribution, because there's always going to be power imbalance.
- **Ask for feedback in a group about a particular project or event.** This is useful because it keeps the feedback specific and focused, with less chance of an awkward ad hominem attack situation. If you do this, be okay with an uncomfortable silence so that people can process and think (I have a hard time with this one).
- **Ask for feedback in a 1:1.** You can do this a few different ways, but it is a good idea to let people know ahead of time that you will be asking for feedback so that you don't throw them off guard. Power dynamics can make it tough for people to assess what they're comfortable sharing on the spot.

You will have much less control of a situation when you get feedback than when you give it. However, your job as a leader is to try to keep yourself open to feedback, criticism, and still evoke calm. Breathing slowly and with purpose, and sitting properly in your chair can push back on your flight-or-fight response. This is crucial because acting quickly in the moment can be detrimental in this situation.

This may seem hard, and that's because it is. But over time, and the more you do it, the more natural it becomes. I keep returning to the importance of going first as a leader, because I believe you have to repeat what's important. When it comes to trust and vulnerability, you have to model the behavior. People are looking to you for clues on what behavior is acceptable and what's celebrated. Show them.

DEFENSIVENESS

It's Monday morning. You had a rough weekend. Your kid wouldn't sleep, wouldn't put their clothes on, wouldn't eat their food. You didn't have a chance to take a shower, you're wearing jammies under your desk. You had a nightmare, but you can't remember what it was, just the feeling it left you with.

Your first 1:1 with a direct report is rocky. They are upset. They aren't aligned to the wider goals you have for the year, and some of this has to do with some false assumptions on their part, and some of it has to do with you not providing context you didn't even know they needed until just now. Their team has been really excited about the direction, so now you have to decide if the misalignment is just with you two, and this person isn't talking to their own team (one problem), or if the team is scared to tell you the truth (another problem).

"You have no clue what we're dealing with at our level and these high-level goals are out of touch," they say to you at 9:35 a.m.

Are they right? Maybe. Probably. But it still probably feels unfair that they come at you like this instead of trying to find common ground and collaborate. Fair or not, you have to try your best to keep your cool and unpack the situation with them. Getting mad will do nothing productive. You need to try your best to divorce your ego, get curious, and dive in. And you have to acknowledge their feelings and feedback as valid along the way.

Does that seem hard? *That's because it is.*

Defensiveness is natural when someone says something about you that you may perceive as negative. But there's a part of you that must be able to confront both the good and the bad in order to grow as a person and improve as a manager.

Depending on the circumstances, this can be incredibly difficult. I find myself working on this quite a bit. Shout out to Lauren Sell, who was vice president of Marketing while I was at Netlify, who was a consummate professional, and someone I looked up to a great deal. Whenever someone gave her hard feedback, her first response was always, "That's a great point," and then she would explore why, along with discussing the tradeoffs, challenges, and proposals for the future. *Bravo.*

So let's break down the triggers for defensiveness, so we aren't just blithely responding and we have some tools.

Blind Spots

Feeling defensive can be a great indicator that we have a blind spot. When something feels particularly threatening, try to consider it as an indicator that you may have some internal exploring to do.

See if you can ask more clarifying questions so that you can give yourself a moment to ponder the feedback and what made this defensiveness rise. Did the other person accuse you of something you don't feel you did? Why did they feel this way? Are they calling out an action they think was less beneficial for the group? Do you know why you did what you did? Sometimes we have good reasons for the things we do, but the way in which that thing was done could be debated.

Asking questions not only gives you more information about what they're conveying, but it allows you to approach the problem with a bit more of an academic handling, which can also disengage flight or fight. It also gives you an opportunity to chew on the information without figuring out next steps right away.

Go back to values. What values does this person have that you may have not honored? What values do you have that pertain to this situation? Sometimes feedback is given because you've unintentionally stepped over someone's boundaries or violated their values. If you can step away from defensiveness and think critically about these values, you can try to regain control over that pesky amygdala and force it into a more logical state of mind.

Ego Resilience

My coach Jessi once said to me, "The ego gets a bad rap. An unchecked ego is a nightmare, but ego can be important for understanding where you are, where you sit, and what you need." So true.

I had a person on my team who scored high on an evaluation for ego resilience at a previous job. This trait made him one of my favorite people to work with, and indeed I trusted him to be my replacement and take care of that team when I left. His ego resilience meant that people could give him productive feedback, and his feelings were not hurt. He took feedback seriously, took notes, and read it back to the person to take ownership over what he had learned. This also made him a great person to "pair program" with because it was never a big deal when something went awry. He just kept going, and had no issue with taking ownership of the mistake.

Ego resilience affords you the ability to separate yourself from others a bit and to take responsibility. Will your whole career collapse if you understand that you need to improve how you react to a certain situation? Probably not. Will it be the end of the world if you acknowledge a mistake in front of people? Probably not.

But a healthy ego can *also* help you separate what's yours from what belongs to other people. Sometimes in feedback, there's a piece of information that is vital for you to understand. And sometimes, an event is colored by where the other person is at in their life. Are they going through something stressful and end up screaming at you? Your ego can help separate "Okay, they didn't deal with that well, and that's theirs to hold, not mine," and "What's mine to hold is that I need to adjust X."

This can also help with blind spots and switchtracking, which we cover next. Having a handle on your own ego can help you check in and do the work to ensure you're not protecting yourself too much. It can also help you let go of what's not meant for you. A good manager has some resilience.

Switchtracking

The book *Thanks for the Feedback* by Douglas Stone and Sheila Heen warns against *switchtracking*, which is when someone gives you feedback (possibly critical) and you put something else back on them.

"You did X. That was incredibly unproductive."

"Oh yeah? Well, you did Y!"

This is a fairly common example of switchtracking, and it's entirely unhelpful. Even if you are correct, and the other person did Y, these are two separate conversations and should not be handled at the same time. **A person having a fault or needing feedback of their own does not negate your need for feedback as well.**

If you catch yourself doing this or desiring to do this, try your best to apologize and stay on track. There's a time and place for your alternate feedback, and this isn't it.

IF IT GOES POORLY

Feedback is a bit treacherous. So much of a person and their sense of self-worth and being can be challenged in a feedback process. Unfortunately, even if you try to do everything you can to help things go well, people are not I/O functions. You may make a misstep or fail to factor in something. And as we noted previously, you are also human and no feedback is truly 100 percent objective. Shit happens.

Or you may be on the receiving end of feedback, and you may not take it well . You're going to have a ton of tough conversations as a manager. Sometimes you do well and sometimes, well sometimes you don't.

So what if it goes poorly?

Write everything down. Try your best to write down explicitly *what happened*, not your *feelings* about what happened. Why? Because we're going to do that thing we spoke about earlier: check in with the facts.

There are a lot of reasons you may think of things from a different angle, be it because of good intentions ("I was just trying to help!") or not-so-great intentions ("They have gotten away with this for long enough."). Whatever the reason, you need to try to sort out what actually occurred. This can help us double-check our own biases, intentions, and objectives.

Are we going to escalate and share these points with HR? Typically not, but it may depend on the circumstances. It's usually an exercise for ourselves, to keep ourselves from going down defensive rabbit holes. It can be useful to do when at a point of serious escalation, though.

AD HOMINEM ATTACKS

What happens if someone attacks your character, rather than giving you real feedback? (We discussed this earlier.) This is what's called an *ad hominem attack*. If it happens, try your best to step it back. For example:

> *"I think we have the same goal here, but it's tough for me to process and move forward with a statement about me personally. Can you please give me more specific details about X?"*

However, if someone attacks you personally, **stepping it back is incredibly hard.** It's completely okay to say that you're not looking for feedback on your character. Express your boundaries around what feedback you *are* willing to listen to and what you are not. **You are also within bounds to leave the conversation.**

In my years of experience, ad hominem attacks tend to be where bias breeds. I have time and time again seen women and people of color attacked for their person, instead of the kind, actionable feedback that tends to be given to some of their peers. It's not okay, and it is cause to go to HR, either for yourself, or on behalf of someone else. Alternatively, it's also right to create boundaries, reconfigure a team, or even pursue alternative employment options when the cases are repetitive and severe.

If you are a hiring manager, this is something to keep in consideration: You may see job-hopping happen more with URM (underrepresented minorities) than with other folks. My suggestion is to consider that these job-hoppers may have had more bias to contend with and more situations where they have had to express their boundaries by leaving.

All of this is to say: put the health and safety of people above all else. People do their best work when they don't feel they are under personal attack. Try your best to cultivate an environment where this is not tolerated. And do not tolerate it for yourself either. Not all feedback is valuable.

CHAPTER 13

GOOD MEETINGS

Like it or not, meetings are essential for communication and nurturing a good working environment. Therefore it's crucial that we managers make them as productive as possible. In this chapter, we explore myriad ways to keep meetings coordinated and well documented. We also talk about how to recognize and steer away from anti-patterns.

I'm timid to write this chapter because I have not always hosted good meetings. I have however hosted thousands of them, so I've learned from my mistakes and my successes. In all likelihood, if you do any kind of management or lead work for a while, you'll also see your own *spectrum* of meetings: meetings with different types of agendas and purposes, meetings with varying levels of awkwardness, meetings that didn't have a formal outcome. We dive into all of these in this chapter, as well as some tips for each.

A meeting by nature can almost never be perfect because it is by definition: a group of people. That group of people will consist of *different people* with different tastes, different opinions, different priorities, and different values. There's little chance that everyone will agree on what a perfect (or even great) meeting is. So half the journey is aligning on *that.*

THE GOOD, THE BAD

One thing's for sure: We can agree on what a *bad* meeting is. So let's start by using that as a ballast. Some features of a bad meeting are as follows:

- There's no clear purpose or direction.
- It feels chaotic.
- The wrong people are there.
- People are generally disrespectful of one another.
- Everyone feels it's a waste of time.

From those assertions, we can then derive what a *good* meeting is:

- The purpose of the meeting is clear.
- There's an agenda (we dive into the complexity of this in a moment).
- The right people (and the right *number* of people) are in the room. **Not too many people where communication is overly complicated, and not too few people where those you need to move forward aren't there.**
- There's some order. People aren't dropping in and out, talking over each other, or being generally inconsiderate.
- There's a clear decision, outcome, and next steps at the end.

PURPOSE OF THE MEETING AND DIRECTION

The first and the last points listed above are connected: To have a good meeting, there is order among the core attendees. You've all come together for a unique purpose, and the end of the meeting should encapsulate what you've learned about that purpose and what the next steps are. Thus the beginning and end of the meeting might sound a little similar.

Ideally, the following points encapsulate information everyone needs:

- What is the shared purpose?
- What are we doing to get to that outcome?
- Who is owning what, and how?
- What are the timelines?

For example, at the beginning of the meeting:

> *"We're all here today to discuss how we're going to support the next version of framework X. I have some new data to show you that frames the direction; Hassan and Jenna are here to talk about some of the details of the implementation; and Angela, we'd love to coordinate with you on a rollout process because it affects your team."*

And at the end of the meeting:

> *"Okay, so we decided we're going in direction Y. Angela, your team seems comfortable doing Z, is that correct? And the rollout timeline we've agreed on is five weeks. The next steps are to explore the impacts of A, B, and C and reconvene in a week on our findings and process."*

This is just an example; it's not important to model this precisely. But you should be aligning at the beginning and end of the meeting to make sure that nothing major is missing, and everyone is on the same page. If you haven't come to a decision by the end of the meeting, then your next steps may be to figure out who will make the decision and inform everyone, or to roll the conversation over to a new meeting.

THE AGENDA

Beyond deciding that a meeting should have an agenda, there are so many ways and means an agenda can be used. Strangely enough, an agenda can also be a way to *not* have a good meeting, so let's explore that too.

Word of caution: I've seen strange cultures arise from making strict rules around agendas. The point of the agenda, and the meeting, is to collaborate on something. That point is nullified if folks are putting process ahead of that impetus. The best cultures I've worked with use both meetings and agendas as tools for working together effectively—tools that everyone feels *equally* responsible for making useful.

An agenda should ideally always state the purpose of the meeting. I personally love to then include some bullets as talking points, as well as space to take notes right in the document during the meeting.

Sometimes people use an agenda to write down thoughts before the meeting, and I strongly suggest steering clear of this. There's nothing wrong with a person who keeps notes for the meeting, but if you come to a meeting where an agenda is locked top to bottom with material, it can sometimes shut down the collaborative aspect of the meeting, which means it shouldn't be a meeting at all; it should just be a shared doc, to be consumed async. Part of the purpose of the meeting is the discussion itself.

Again, louder: *Part of the purpose of the meeting is the discussion itself.*

NOT ALL MEETINGS ARE THE SAME

There are also *different kinds* of meetings. Let's go over what type of agenda you might use for each.

Cross-Functional Meetings

This is where a more-formal and prepared agenda can be really helpful, so that all parties have enough information about the purpose and what's being discussed. If you have a lot of information though, I suggest creating a one-sheeter and sharing that ahead of time instead of adding everything to an agenda. If I know everyone is too busy to read everything ahead of time, I will sometimes give the first five minutes to the group to read through the one-sheeter so we're all on the same page. People usually appreciate this. Your mileage may vary.

Weekly Discussion or Daily Standup

For informal weekly discussions or daily standups, I typically have folks add whatever they like to our agenda, asking that they preface their contribution with their name and a small category. For instance, **A** for announcements, **RD** for rapid decision, **D** for discussion, and **Q** for questions. Here's an example:

> *[Sarah, RD] should we block off four hours to triage our iceboxed (items stored for later) issues?*

Many teams I manage use a kanban board during the standup, and people take turns talking about what they're doing for that time period. This is nice because it helps solidify the tasks and priorities for the week and allows for some course correction if there's accidental misalignment before the work is done.

In these meetings, we also talk about what was done or shipped in the previous week, so that we can celebrate a little, especially the tasks we know took the person a lot of time or effort. A few teams found, through trial and error, that twice-a-week check-ins were suitable: once on Monday to kick off the week, and again on Wednesday to keep us aligned and our momentum going.

I manage a lot of teams, and they all do things slightly differently based on their styles and what works for them. This is okay! It's not really that important whether all the teams you manage follow the same process. We should use what makes us feel most productive.

Brainstorming Session

Perhaps you don't want a full agenda, just the purpose and a notes section, or even an online whiteboard tool (Miro is one such example) to use for capturing people's thoughts, with small areas stubbed out. For these brainstorming sessions, you don't want a full agenda because nothing breaks the flow of active collaboration like pressure to move on and follow a structure.

It's okay to capture major themes on an agenda to provide some structure, but the more they are bullet points and less fleshed out, the more productive the session will be. It also takes some trust in a group to throw around ideas, as well as a group size that's not too large. If teams don't have trust at the outset, you may have to start with some ice-breakers.

Even though I'm a huge fan of remote work, and you can hold these meetings remotely, I find that unlike other meetings, brainstorming sessions really do seem to be easier in person. It can be nice to bake brainstorming sessions into team offsites to generate some energetic discussions.

ALL KINDS OF AWKWARD

Okay, say you led a meeting. You gave people purpose, you set direction and timelines. *But why was it so awkward?!*

Not all forms of awkward are bad, really. Some kinds of awkward are quite natural. Some are more harmful. Let's analyze this, starting from most innocuous to something more insidious.

You All Didn't Know Each Other Well

The teams I worked with at Netlify were some of the silliest, most-collaborative, and trustworthy groups I've ever had the pleasure of working with. We actively cultivated this culture, and it was great fun. Every meeting started with goofing off and chatter. Then we got to business.

The meeting flowed effortlessly because we were all comfortable together. One time, a friend in the People Department asked, "What do you do to break the ice with your team?"

I jokingly responded, "*I*ce? Our team? No . . . We don't need that . . . Maybe we should be frozen!"

Not all conversations are going to be like this. In this situation, we knew each other fairly well and actively worked to have vulnerability together.

If your meetings with other groups you don't know well have awkward moments, that's actually pretty natural, and nothing to be too concerned with. In fact, if you're doing a good job bringing in diverse people and perspectives, teams will feel a little less comfortable to a degree,[8] and that will actually help them perform better in the long run, as they will be able to innovate and see risks from multiple perspectives.

You can try to make conversation, and that can help; but trying to force it too much can also feel a bit stilted, so just **ease up on the guilt for this one**. There's nothing wrong with you, I promise.

There Were Too Many People Or Not The Right People

Remember the statements for the purpose of the meeting and the direction? If some people at the meeting do not need to know this information, they probably shouldn't be at the meeting in the first place.

During the pandemic, my husband and I sometimes tried to replace in-person dinner parties with Zoom versions of the same. We learned they didn't quite work at scale. When you have an in-person party with twelve or more people, everyone doesn't really stay in one huddle together, they break off to smaller conversations. When we started hosting the Zoom parties with smaller groups, the calls became more fluid, relaxed, and comfortable. This is a personal example, but work environments have similar dynamics.

There's a certain scale at which conversation begins to feel performative, because there are so many eyes on a person when they're speaking. Meetings are very much the same. **Try not to invite too many people to a meeting.** If you are worried folks might not feel included unless you invite them, you can either mark them as optional or let them know you'll be sure to tell them the outcome.

If you're inviting too many people because the company culture is that everyone should be involved in every decision, that might be a sign of a wider issue that needs solving. Companies at a certain scale start to have issues functioning if there is no clear understanding of ownership. **If you're inviting everyone to meetings out of fear of hurt feelings, it's likely not a problem with your meetings, and more a sign that roles and responsibilities aren't clear for everyone.**

There's Something People Aren't Saying

This kind of awkward is probably the most harmful. If the meeting is awkward because people don't feel comfortable telling the truth, or there's an elephant in the room, or there's a smell that needs to be dealt with. Elephant smells? Okay, *moving on.*

As managers, we should watch out for this and try to do something about it. Personally, I'm a "walk toward the fire to put it out" kind of person, and will actually just *acknowledge that it's awkward* because it doesn't feel like we're being transparent with one another. I'll state what I know from my perspective and then ask if other folks are feeling the same.

If you do this, you'll usually have to wait a beat or two. People will likely be a bit shocked that you came right out and said it. It may take them a couple of seconds to adjust and consider what will happen if they tell the truth too. It's crucial that you not speak to fill the silence in these moments. It may feel uncomfortable, but I promise, you have to let the silence hang for a bit before someone speaks up. From there, people will typically all start speaking, and you can actually dig into the problems.

CONFLICTS

An entire chapter is devoted to conflicts because the topic is big and nuanced enough to warrant its own time and space, but let's apply some of the principles here because there is an intersection of good meetings and dealing with conflicts directly.

The most important piece here is that conflicts are not something to be avoided. It's not bad that people feel passionate about their work; it's great. Not all conflicts are negative; the point of the meeting may be to bring to light where folks aren't aligned. Perhaps there is some base premise or problem they are all trying to solve, but they see the solution differently. It helps to find the alignment there, so ideas can be fleshed out without being attached to a particular person's identity.

The identity thing can be a pitfall because, if you have two people discussing *their* idea instead of *an* idea, it can feel to them like someone is rejecting *them* rather than *a concept*. We want to try to guide our peers toward a place where it doesn't feel like anyone is attacking anyone else, and where people are not being disrespectful to one another.

It's the job of a manager to disambiguate healthy conflict from attack, so that respectful discourse is encouraged. If folks are putting out ad hominem attacks, it's on you to reel that in, and move the conversation toward the work instead. Otherwise it really is hard for the conversation to stay productive.

Typically, it's good to hear people out and then rein things back in by discussing what you think you're hearing and tying it back to a shared purpose. Then we can find where we have common ground.

Here's an example:

"What I'm hearing is that Rashida feels that team X is migrating a system that affects her team while they are trying to release a big feature. Is that correct?"

"And I'm hearing that Jerome feels that it's crucial that team X be able to migrate the system soon for stability purposes. Is that correct?"

"Okay, it sounds like we have a shared goal of making sure the company can ship features with some stability. Perhaps we can talk through which timelines are immovable and which are not so that we can stay coordinated?"

"I'm sure we all want to be able to ship said feature without any hiccups and also get the new system up and going."

Here we stated what we thought we were hearing, which allows for the person to either feel heard or to correct us if we're mistaken and there's been a miscommunication. (Sometimes there is!)

Then we stated the *shared goal from both parties*, as well as risks and constraints that may play a part in some of the conflicts that need to be ironed out.

You'll note in the last sentence above, we try to tie a knot for a vision of stability that addresses both parties' understandable needs.

> *NOTE*
>
> *I'm giving an example here. You absolutely don't have to model this exactly . Most important is that folks feel heard and that you all agree on* ***what the conflict is****. Also important is that you remain open to that discussion while finding the real reason you're even talking about it.*
>
> *There can be times where a strong conflict arises during a meeting, and you're at an impasse. You need to give folks time to regroup. I'd suggest calling another meeting in a week as a follow-up and try to hear people out individually in the meantime. Sometimes people need a little distance from a matter, or they're having a hard day, and that's totally okay.*

THE DRI

A good meeting must have a DRI **(directly responsible individual)**, and it is not necessarily the person who called the meeting. It might not be you. But you must designate who owns the project and ultimately makes decisions when there's one to be made.

Why do you need a DRI? Well, as much as you want to hear input from everyone, eventually you have to make a decision, and there are plenty of things in software development (and in life) that don't necessarily have one true answer.

Note that the phrasing is not PWMD (person who makes decisions), though that acronym looks pretty hardcore. Instead, we use directly **responsible** individual because that's also core to deciding who this person is. *That is the person who is going to own the outcome.*

This demonstrates why everyone who attends a meeting can't have equal say. If it's your project, and you are on the line for the outcome of whatever decisions are made, you can see how you will also need to own the decision-making. Likewise, if people who have no skin in the game make the decisions, they might not understand all the moving parts or invest as much in the gravity of the matter.

There are several project management framework systems of ownership you can explore, such as DACI (driver, improver, contributor, and informed), which separates out each stakeholder so that everyone knows their role. There are several others, such as RACI (responsible, accountable, consulted, and informed) and RAPID (recommend, agree, perform, input, decide). Use whichever system makes the most sense for your organization.

The appointment of the DRI not only unlocks the groups to make final decisions and move forward, but also places the responsibility on the party that will carry the weight. I find it best to identify this person early on in a project and make sure it is restated at the start of a meeting (it can be included on the agenda as well), as it helps greatly if you find yourself at a crossroads.

WRAPPING UP

At times, it may feel like meetings are a drag on any engineering process, but it doesn't always have to feel this way. There's something special about collaborating with a group of people who are respectful and working toward a common purpose. Good meetings can provide clarity and save people hours and days of work when they're headed in the wrong direction.

Having clear ownership, documentation, and only the right people in the room can keep many teams in lockstep, even when problems are complex.

8. https://hbr.org/2016/09/diverse-teams-feel-less-comfortable-and-thats-why-they-perform-better

CHAPTER 14

MANAGING CONFLICTS

When you pivot to engineering management from a career in software engineering, managing conflicts is one of the most shocking "This is now actually your job" moments. It's also one of the most critical parts of our role as leaders. We absolutely need to learn how to manage conflict well and hopefully make things better, not worse, in the process.

Managing conflict is not something you learn how to handle once and are completely done learning about. I am constantly learning new things about managing conflict even after a decade of managing.

Similar to egos, conflict has a bad rap. The word itself evokes images of clashed swords, people dividing, and furrowed brows. Not so.

Conflict is actually a very necessary and healthy part of a productive working environment. Why? Because we work best together when we're all learning from one another, and that can't happen without some disagreement along the way.

In his book *Hell Yeah or No*,[9] Derek Sivers writes:

"You have to know your preferences well, because no matter what you do, someone will tell you you're wrong."

Most truly innovative concepts challenge the status quo. Part of this process means that you need conflict to challenge and flesh out an idea, to consider the risks, and allow everyone involved to see the size and the shape of the thing at hand.

Conflict can sometimes represent opportunity for growth and learning.

If you review the times in your life where you truly leveled up or learned a lot, some conflict was probably involved. Whether it was internal conflict, *Why won't this compile!?* or external conflict, "I don't think we should build it this way. This way will be better!" dealing with what needs improving and really sorting it out yields better results than telling ourselves everything is fine when it's not.

Conflict can mean we're outside our own comfort zones, and conflict can also give us an opportunity to see what we're made of. It won't always feel like that, and not all conflict is good. But viewing productive conflict as an opportunity for growth can save your sanity for the times when it's not so productive.

FALSE HARMONY AND PRODUCTIVE CONFLICT

It can be tempting to think that good working environments are devoid of conflict or disagreement, but it's not so. The key is to create a healthy environment where people can say they disagree without the world collapsing or sinking to personal attacks.

For instance, let's say you bring up a concept to a group of people. You want to make a major update to a particular feature in the roadmap. Consider these two responses:

> *Silence. Then, "Sure, sounds good."*

Alternatively a peer says to you,

> *"I don't think this is the best possible path because of X and Y."*

In the first scenario, you have what is called *false harmony*. It seems like people agree, but everyone is quietly holding their tongue. They're not sure what the cost is of speaking their mind.

In the second scenario, this colleague either told you about a risk of this idea you weren't thinking about, or brought up a concern so that you can address it. That kind of information is powerful. Someone bringing up a risk you didn't see is an enormous gift. You need this information to build a great plan, or to not build it if the risk is great enough. In the case that it's something you've already considered and have an answer to, this is also great! It means you can address it head on, and negative thoughts and feelings don't simmer and boil over time.

As you move up the leadership chain, the further you will be from the actual metal of the problems, and thus, the more you will need to rely on your staff and others to tell you the truth of the problem space.

Don't fear conflict; embrace it. At the same time, we need some tools to keep this type of conversation productive. We need to sort out how to share ideas without attacking one another or feeling attacked. We need to understand how a conflict might find its source in our own values, which might differ from someone else's. And we need to understand that there's a time and place to disagree, and also a time to disagree and commit, with an appointed decision-maker.

ACTING IMMEDIATELY

When there is a conflict, be it on an individual or organizational level, it's crucial to deal with it immediately. This doesn't mean dropping everything every time someone has an issue, but it does mean being fairly clear at what constitutes a conflict to be dealt with and what might be personal preference or opinion.

Letting things go on without resolution can also introduce a degree of uncertainty among staff, and that lack of clarity can be distracting and demoralizing. My suggestion is to deal with things as quickly as possible and with the attention that the problem deserves, so that things don't fester. This can be tough for those of us who are conflict-avoidant, and hope things just pass. There certainly are times when leaving a conflict alone will help. Unfortunately, as a manager, these are less frequent, and your job does require you to go into the fire to put it out.

SEPARATING IDEAS FROM IDENTITY

As covered in the chapters on feedback, it's incredibly difficult for humans to approach conflict from a thoughtful, objective place when they feel attacked. Thus, the best way to have a productive disagreement is to disengage from any kind of personal attachment to an idea from the get-go.

It's very common to bring an idea to a table as "my idea," the operative term there being *my*. It's natural to desire credit for your work and thoughts. However, the point of discourse is not to prove your worth and identity. Ideally, if you collaborate with your team and colleagues well, this falls into place naturally.

Can you approach an idea as "an idea?" We live in a world where so many possibilities exist. The idea that you put forward is one concept. Maybe it has a good basis, but needs polish. Maybe it was a good idea based on the information you had previously, but you just got new information. Whatever it is, many concepts exist. It's not important that it's *yours*. It's important that it's right for the company and the people within it.

I have to admit, I really struggle with this. It's so much easier said than done, and if you've ever worked in environments with toxic egos, over time it's natural to become more protective of yourself. However, it's never too late to try to build a better environment, starting with our own behavior.

When I struggle to separate my ideas from my identity, I will check in with myself. If I feel defensiveness over an idea, I try my best to step back from the conversation a bit and ask someone who has been a bit quiet what they think. This does two things: it allows me to hear from someone whose voice might be drowned out because other folks are louder, and it allows me to take a breather from that feeling and try to give myself a bit of time to think things over so I don't trample someone else's idea.

You're leading a team—they're looking to you to model good behavior. You probably want a team that's as open and collaborative as possible. As a manager, it's important you try to imbue these qualities.

UNDERSTANDING CONFLICT AND VALUES

In Part 1, we talked about the value of values: why understanding individual values has so much to do with how a person operates and perceives the world. Values play a huge part in conflict as well. When there is conflict, it often helps to align on the values that are shared, or give context to why some people may have one value over the other. Let's look at both.

Let's say you have one team member who values "debate," another that values "stability," and yet another that values "learning." A problem arises, and they may all view the same discourse completely differently.

The person who loves debate and the person who loves learning get along at first because they both want to try to find a new solution. But the person who values stability might see this change as a frightening thing—you are upending the foundation of their work, and they don't feel prepared. The learning person might see this start to play out and back away because they no longer feel like the group is learning most productively, but the debate person is riled up! This is it! We're finally getting to the heart of the matter and sorting things out!

Without a bit of values work, this type of interaction can feel a little chaotic. How do we allow everyone to feel included?

I try to think through my team members' values and cater to each of them a bit, sometimes even addressing that difference out loud. (Tread lightly here though, because this may also come off as a personal attack.)

Let's see how this values exercise might give us some tools:

"Alright, this is great. We're all coming together to sort out a problem with one

another. We may want to ask ourselves during this process, 'Do things need to change? What are the risks if we alter things or don't alter things?'"

And perhaps to show some personal vulnerability so that people know it's okay to do the same:

*"What might I be missing here? What are we **not** talking about?"*

This isn't a be-all and end-all example, but you might see that we're addressing the fear of change and talking through the potential benefits of stability. We're also learning, and we're allowing for a bit of debate.

Thinking through how something might sound to the individuals on the team can help everyone get back to that common ground and think a little more logically about a situation and the outcome.

CREATING PSYCHOLOGICAL SAFETY

In building teams, it's important that people have an area where they can do work, collaborate, and feel safe before showing a wider audience of stakeholders.

This can be helpful because not everyone will feel comfortable speaking up in a large space with a lot of eyes. The risk here is that we can accidentally make an environment where the loudest and most opinionated people get their way.

In a team with some shy members who don't know each other well, it's easy for someone with a ton of confidence from another group to march in and drop an opinion bomb on them, to which they may all begrudgingly and quietly agree. I've managed the type of team that's quieter, and it was upsetting to see them cave to a random opinion that they didn't feel was right just because it was uncomfortable to speak up.

A group like this might also stay quiet because they're scared of being wrong. The trouble is, you do need a place where people can throw out ideas and not be so precious with the final outcome. These places are more common in a smaller, trusted group than out in the open, at least at first.

I can at times be that opinion-bomb person, so I have to do a lot of reflecting about how to create environments for everyone. I have to be comfortable with situations when I shouldn't be included. If you are this type of person as well, understand that you may need to give your team some space.

It may at first seem like having everything out in the open and having everyone get a say is the healthier mode of working, but if you're not providing some safe places for shyer people to speak up, you may be losing out on their input and accidentally self-selecting to those who are more boisterous. Remember, shy folks have great input too.

FINALIZING A DECISION

When a conflict arises, it's typically due to clashing values or misaligned outcomes and goals. Aligning everyone's values is not possible, but we can create clarity around outcomes where none exists. Focusing on the mission or outcome and aligning to that can be very productive. It's amazing how quickly it dissolves turf wars and unneeded bureaucracy.

Some conflicts can be settled in a data-driven manner: look at past data, time-box a study, involve data analysts where you can. But there are also conflicts for which there really is no right answer, and it comes down to a matter of opinion. The trouble with opinion is that if you're not directly talking about a human rights issue (we get to this in a minute), there are multiple paths forward. In this circumstance, let everyone know that you value their opinion and want to hear their thoughts on the matter, but also acknowledge that you have to make a decision that aligns with the wider goals and strategy for the company, not popular vote. At this point, a DRI (directly responsible individual) or decision-making framework works best and needs to be clear to everyone. Bonus points if this DRI is the kind of person who changes their mind based on new information.

> *NOTE*
>
> ***Why doesn't everyone get to make the final decision together?*** *We've talked about this already, but it's worth mentioning again: The folks who have the most responsibility for the outcome of the decision should have more say than those who do not. Do you work in another organization and have an opinion on how that organization is run? You may be missing context that's invisible to you: How much tech debt there is, historical data, social nuances, etc. The people who are working on the project directly or have to be responsible when something goes awry should, in all fairness, have a bit more say in the decision.*
>
> *Regardless, it's important to listen and gather differing perspectives. It helps to say things like "What I'm hearing from you is . . . Is that correct?" It not only allows the person to feel heard without judgment, but also allows them to correct you if you're misunderstanding them. You might be surprised (or maybe not!) by how often I think I understand what someone is saying and doing this exercise allows them to drive clarity for me.*

WRAPPING UP

The very word "conflict" evokes people at battle, opposing forces. But not all conflict is negative. Conflict can illuminate risks, allow us to collaborate and improve processes, and affect positive change. The key is to continue to provide clarity, and usher folks toward productive conflict that is devoid of personal attacks, and focused on the larger goals.

If you can keep the team focused on the big picture, allow voices to be heard, and provide psychological safety for the team while also time-boxing the discussion and appointing a person who makes the decision, you can avoid situations where conflicts are either undiscovered or drawn out forever. Here again, clarity is key.

9. https://sive.rs/n

CHAPTER 15

CROSS-TEAM AND OPEN SOURCE COLLABORATION

The chance that your engineering organization will be working in a silo is low. If you're a small group, you may find that you do consulting with other companies; and if you've grown in size, you may find that you have to collaborate with other parts of the company.

At the same time, open source software is thriving. Large corporations are building on software that rests on open collaboration, enjoying the many benefits of significant community adoption. Free and open source software is amazing in its ability to bring together many people from all over the world, and join their efforts and skills.

The etiquette for collaboration with any other group both within and outside the company is not so different, so we talk about cross-team collaboration and open source collaboration at the same time. It's worth taking a moment to reflect on how we can work best together.

The manner in which you conduct yourself while working with others can impact whether your work is merged, whether someone works on your issue, or in some cases, why you might be blocked from participating in the repository in the future. This chapter is a guide to keeping these communications running smoothly. We look at this both from the point of view of being the maintainer of the project that has other stakeholders coming in to collaborate, as well as a user of that project, coming in from another team.

An overarching theme for both perspectives is to be respectful of one another's time, provide clarity, and express gratitude. People are often donating their time on either end, and being thoughtful and thankful when working together can go a long way.

COLLABORATION FROM THE MAINTAINER'S PERSPECTIVE

You and your team are likely very aware of every nuance of your project, both good and bad. You know how you prioritize, you know what's simple to fix, and you know the areas where "*Thar be dragons*." When you're so close to a problem, we can start to get what's called "curse of knowledge," where we feel that everything is as obvious from the outside as it is to us because we know it so well. Unfortunately that's not always the case, and you can save yourself a bit of time and frustration by providing some clarity for users and potential contributors of the project. Here are some ways you can provide this clarity:

- Use labels like "help wanted" or "beginner friendly" to guide people to issues they can work on if they are new to the project.
- Create a makefile, CONTRIBUTING.md, and/or do a writeup in the README of how the project is constructed, what kinds of styles are used, and the types of PRs that you're looking for (or not looking for, as the case may be). Here's a small example from my Array Explorer project:
- https://github.com/sdras/array-explorer/blob/main/CONTRIBUTING.md
- Make good use of issue templates. In some cases, like if you have a very popular project, it might make sense to create your own form and do some validation (we do this with Vue: https://new-issue.vuejs.org/?repo=vuejs/vue). This can help ensure that you're getting the information that you need and not wasting cycles

closing issues that don't give you enough information to work on the task.

- When you ask someone for help or label an issue "help wanted" and someone PRs, write a comment explaining why you are closing it if you decide not to merge. It's disrespectful of their time otherwise, as they were following your call to action. I even suggest commenting on any PR that you close OR merge, to explain why or to say "thank you," respectively.

Hygiene

Having so many people vying for your time and energy on a project can feel a bit draining. Here are some tips for how to keep things tidy and running smoothly on the human side:

- Provide a code of conduct that makes clear what type of behavior is acceptable and what is not. Make its presence known. You might consider the contributor covenant code of conduct.[10] GitHub also now offers easy code of conduct integration with some base templates.

- If someone is confused, try to assume good intentions. If you're feeling impatient with that person, give yourself some space from the issue, and try to feel gratitude that they're interested in the project.

- Don't close a PR from an active contributor and reimplement the same thing yourself. Just . . . don't do this.

- If a fight breaks out on an issue that gets personal, shut it down to core maintainers as soon as possible. Lock the issue and ensure to enforce the code of conduct if necessary.

Collaboration from the User's Perspective

Now let's consider an outside team member's perspective. From the get-go, make sure you're doing everything on your side to be respectful of the maintainer's time and the project. This involves a little bit of homework before you can even write any code:

- Before doing anything on a project, familiarize yourself with the contributor guidelines, often found in a CONTRIBUTING.md file at the root of the repository. If one does not exist, file an issue to ask if you could help create one.
- Saying "thank you" for the project before making an inquiry about a new feature or filing a bug is usually appreciated.
- Avoid entitlement. The maintainers of the project don't owe you anything. When you start using the project, it becomes your responsibility to help maintain it. If you don't like the way the project is being maintained, be respectful when you provide suggestions and offer help to improve the situation. You can always fork the project to work on your own if you feel very strongly it's not the direction you might take it.
- When opening an issue, create a small, isolated, simple reproduction of the issue, using an online code editor (like codepen or codesandbox) if possible and a GitHub repository if not. The process may help you discover the underlying issue (or realize that it's not an issue). It will also make it easier for maintainers to help you resolve the problem.
- When opening an issue, suggest a solution to the problem. Take a few minutes to do a little digging. This blog post[11] has a few

suggestions for how to dive into the source code a little. If you're not sure, explain that you're unsure how to proceed.

- When opening an issue, if you're unable to resolve it yourself, explain that. The expectation is that you resolve the issues you bring up. If someone else does it, that's a gift they're giving to you (And you should express the appropriate gratitude).

- Don't file issues that say things like, "Is this even maintained anymore?" A comment like this is insulting to the time the maintainers have put in. It reads as though the project is not valid anymore just because someone perhaps needed a break, or were working on something else, or their dad died, or they had a kid, or any other myriad human reasons for not being at the beck and call of code. It's totally okay to ask if there's a road map for the future, or to decide based on past commits that it's not maintained enough for your liking. It's not okay to be passive aggressive to someone who created something for you for free.

- When you want to submit a really large PR to a project that you're not a core contributor on, it's a good idea to ask via an issue if the direction you'd like to go makes sense. This also means you're more likely to get the PR merged because you have given them a heads up and communicated the plan. Better yet, break it into smaller PRs so that it's not too much to grok at one time.

Once you're involved a little deeper in the project, there's some general hygiene to be aware of, as well:

- If someone respectfully declines a PR because, though valid code, it's not the direction they'd like to take the project, don't keep commenting on the PR. At that point, it might be a better idea to fork the project if you feel strongly about the need for a feature.

- When running benchmarks, show the authors of the framework/library/etc. the code you're going to run to benchmark on before running it. Allow them to PR (it's okay to give a deadline). That way, when your benchmark is run you know they have your approval and it's as fair as possible. This also fixes issues like benchmarking developer errors instead of production or user errors.
- If you work at the same company and a PR or issue starts to drag on for a while, suggest jumping on a call to resolve it rather than continuing the thread. Sometimes you can see eye-to-eye more quickly in a call. Some things are hard to resolve over text that mirror neurons can help with when you see one another's faces.

WRAPPING UP

The overriding theme of these collaboration tips is to be polite, respectful, and kind. The value of collaboration between teams and open source is immeasurable. We can make it a better place for everyone by following some simple rules of etiquette.

Remember that maintaining a project can be tough, and often thankless work. On the other side, many users of projects are truly looking to help. We can keep this in mind when communicating and working together. By doing so, we can be more collaborative and productive.

10. https://sive.rs/n
11. https://kentcdodds.com/blog/what-open-source-project-should-i-contribute-to

PART 3

CHAPTER 16

PRIORITIZING YOUR TEAM'S WORK

What if you were required to go to the grocery store, get a haircut, and go exercise all within one hour? How might you feel? Perhaps a little panicked, knowing not all of these tasks could be realistically done. Having no sense of which task is most important, leads to further confusion. If you can't get all of them done, which task should you prioritize?

A manager's job has a lot to do with prioritization. Any role (tech lead, engineering manager, director, or VP) needs to understand the importance of being able to prioritize effectively. Prioritization is crucial to giving teams the clarity they need to succeed in their roles. Otherwise chaos ensues.

Teams need their work to be prioritized because:

- With too much to do and not enough time, they need to know what's most important to do first.
- They need to have focus in order to do those tasks well.

It's also crucial to prioritize *your own time* so that you can help your team be effective, though we address that piece later in the book, in Part 4, "Your Work." This chapter is specifically dedicated to our teams. More details related to team prioritization are in the following five chapters as well.

THE BASIS OF PRIORITIZATION

The very nature of how you approach prioritizing your team's time has a lot to do with the size and stage of the company. Do you have a product function? Have you taken funding, and do you have a board? What type of market are you building in? Is your group serving the needs of folks within the company, or a sales function, or a group of developers, or a group of users? All these factors impact your prioritization.

In this chapter, we'll go through some tools for prioritization in case none exist already and you're making something from scratch. We'll also talk through how to understand systems that typically exist at companies, like OKRs, so that you have some background on how to work with this part of business needs.

WORKING WITH OKRS

An overarching factor in a company may be OKRs (objectives and key results)—a heavily adopted pattern across industries for establishing a data-driven system where results are measurable, and you can hold yourself and your teams accountable. We don't dive into details about OKRs here, but if you're interested in more information, I highly recommend John Doerr's book *Measure What Matters*, which discusses the history and purpose of OKRs.

What you should know about OKRs is that they are separated by "objectives" and "key results."

Objectives should be high-level goals that inspire a picture of the future and imagination. For instance, at a point in Chrome's history, their objective was to "make the browser an operating system," and a point in Microsoft's history when their objective was to put "a computer in every household." You can see how these high-level objectives might inspire all sorts of work to be done. And everyone can check their work against such a goal.

Key Results are specific measurable pieces of data over time. For instance, “I would like to grow users” would not work as a key result; but “We need to triple signups by next year” would work. For Chrome in 2009, their original strategy was “a healthy disregard for the impossible,” but the actual key result was “111 million weekly active users by the year’s end.”

> *NOTE*
>
> *It’s important to consider that Chrome was not on track for this success. The previous bar was set at fifty-million weekly active users, and they had only reached thirty-eight-million weekly active users. It should be clear to everyone that there is some safety in moonshots and setting aggressive goals. “Measure What Matters” suggests the 70 percent rule: be comfortable with only succeeding 70 percent of the time.*

In order to create a good, actionable OKR, work on both objectives and key results. And crucially, there can’t be millions of them. Limit the number of overall OKRs so that your team can focus.

Other important characteristics about OKRs:

- They are defined by a top-level business goal or strategy.
- Once defined, they should be easy to find and clear to everyone. If you have a handbook, put them in there. Repeat them often, until you’re sick of saying them.
- While being defined, they need to be reactive to feedback, especially from your peers. If every group at the company has wildly different OKRs, then everyone will be trying to work on different things and unable to effectively collaborate to reach a common goal. Your OKRs don’t have to be the same as your peers across functions, but your peers should agree that what you’ve set out to do is important, be willing to help, and vice versa.

- As suggested above, such OKRs will likely make you mildly (not wildly) uncomfortable. The goal should be kind of exciting because you're trying to do something that's a challenge.

AN EXAMPLE SYSTEM

Let's first dive into how you might go all the way from a top-level OKR to day-to-day tasks. You don't have to use this system; frankly, in some companies you may be at a scale where these steps are untenable for various reasons. But outlining a mid-level example in concrete terms allows us to see what gaps we might need to fill in, or conversely what is out of our hands.

Step One: Defining the OKRs

Depending on where you are in the hierarchy at a company, as well as the relative size of the organization, you may or may not be involved in defining the highest-level OKRs. Any system for metrics is going to be flawed for a few reasons: we can't see into the future, we can only evaluate the past, and the industry shifts rapidly. So depending on the space we're in, **we may have to adjust at times**. But OKRs are still valuable to understand and evaluate progress.

At a top level, I like to define future OKRs by doing two things: looking forward and looking backward. In looking forward, take myself out of the day-to-day and consider the big picture: What do I think our company could do that will make waves in the next few years? What could I do to contribute to this?

While you form some of these concepts, take a look backward: What does the data say about this direction? What pace is the growth of this idea currently? What data do I have access to? What are the unknowns? What are the gaps I could close?

If you have a data team at the company, this is an excellent time to involve them. Typically, they are trained to ask the right questions about your questions as well. Be open to that type of curiosity while working through the OKR definition.

Clearly communicate this process to everyone while you're doing it. Let your teams, peers, and stakeholders know what you're thinking about, and allow them to give feedback during the process.

Step Two: Breaking OKRs Down Into Something More Tangible

A top-level OKR of "increase signups by 300 percent" may be reasonable, but it's difficult to see how that might translate into actual engineering work. Typically, you'd have a product function that works on the ideation and road map for the features, and you as engineering manager would focus on the *how*. But at a small company, or at a big company for particular teams, this may not be broken down for you, so let's talk about it.

In order to increase signups, you can look at what has historically impacted your company. Any spikes in the data (if you don't have data, you'll have to deal with that first) should give you some clues. For this example, what I've seen lift signups have been:

- Big feature releases
- A company conference
- Good SEO/information hierarchy on your app and marketing sites
- Repair of broken or confusing signup flows
- Docs improvements and better onboarding
- Case studies
- Funding announcements
- Developer advocacy

You can't do everything. Given the size of your staff, the relative impact of each of the above events, and how much time you need to dedicate to making these happen, you can start to come up with some plans.

Let's dive into an example. Perhaps you work at a place where the things that made the greatest impact were the feature releases, case studies, and funding announcements. You aren't the one in charge of funding, so that's off the table (though you can contribute to it by jumping on calls with investors). Case studies are likely not your department either, but you can promise to collaborate with Marketing and give them any information they need or have your engineers be interviewed by them.

This leaves the feature releases. Historically, every bump from a feature release has increased signups by +=50 percent, so we can begin to walk backward from our target into how many releases we need to get over the line, in conjunction with other initiatives.

Let's say that number is three, given all the other things the company will do to create impact. From there we can work together with our staff to find a pace that seems reasonable within the given time period, always with some padding; things always take longer than you think they will.

Of course we need to break down those releases even further. I like to use milestones in GitHub to add a slew of issues to, because you can easily track your progress and add labels for more refinements. The next chapter goes into more detail about this process.

Step Three: Creating A Quarterly Planning Spreadsheet

OKRs are typically defined over a one-year span and, though you may be very motivated by them at the start, it can be difficult to keep up that momentum. A quarterly planning spreadsheet is helpful for a few reasons:

- As a helpful reminder, put it in a place where everyone can see it (I put mine at the top of my organization's docs).
- You can guard against "scope creep," where people pile tons of other things on your team. For instance:

Stakeholder: "Hey, can your team do X?"
Me: "Maybe, but how does it fit in with the priorities we already set out to do?" (I show them the spreadsheet.)
Me: "I'm happy for us to work on this if we deprioritize something else on this list."

- A quarter is a reasonable amount of time because typically things shift, but not dramatically in that time; and if they do, it's not the end of the world. And in three months, you can usually build something, but not everything, so you can get some data on your project at length.

An example of the fields I include in a quarterly planning spreadsheet:

- The name of the project
- The docs link to more information on the project
- The point of contact/project owner/lead
- The priority level (I usually make a tag for "high priority" and limit three.)
- A tag for whether the project is "ongoing" or "contained" ("Ongoing" means something the team will always do, e.g., maintain X template; and "contained" means a project will start and be completed, e.g., plan a talk about project X for a conference.)
- A link to any data/dashboards available for the project
- Release date, if there is one

You don't need to use these exact fields; use fields that make sense for your teams. It's most important that this spreadsheet is useful.

Make sure you communicate this document widely, in all forms of communication: slack, email, in meetings, in handbooks, you name it.

Ideally, you want to spread awareness so **there are no surprises** when three months pass and you communicate findings. Avoid the horrible feeling you get when working toward a goal and getting to the end and hearing from a stakeholder, "I didn't even know your team was working on this."

Once the three months are done, I suggest sharing the results and what you learned. This can be in the form of company/org presentation, newsletter-like email, or whatever method of communication is common and effective at your company.

THE LIMITATIONS OF METRICS

You can also measure engineering throughput during the three-month span. Some great tools, such as LinearB, can be used to assess status of things like issues closed, PRs merged, and the like. This is useful for setting per-team targets for cadence and to highlight if something is wildly off. (A full week and three PRs merged for a team of ten? What's happening there?)

> *NOTE:*
> *A word of caution here. Engineering throughput metrics are easy to be gamed: close a lot of small issues, only work on the easiest PRs, and so on. Some of the "gaming" is actually healthy: You don't want people to be submitting huge, lumbering PRs that are impossible to review; but on the other hand, there are a lot of things these metrics don't tell you.*
>
> *Was one PR a really nasty bug that was causing all sorts of site reliability issues and untangling that took the whole team days? That's actually incredibly productive work, and I'd rather my team focus on getting it over the line than doing a lot of busy work items to make their numbers higher.*
>
> *What about reviewing PRs? You could potentially have someone quickly put in a sloppy PR, and a diligent, careful reviewer notes there are side effects, builds the branch, and carefully tracks down what needs to be improved.*
>
> *These are tough things to measure. That's why you should never use*

engineering metrics alone as a guide. It must be supplemented with qualitative analysis that can only come from communication with your team.

Engineering throughput can also be a sign that you're moving quickly, but it doesn't necessarily track what people are working on. It's vital that you communicate with your team so that you can double-check that everyone is working on the right things for the business and not just busy work.

Metrics are incredibly valuable for being able to talk through anomalies, to get an idea of cadence, and to get more accurate estimates for work over time.

Trust comes into play here too: If you set OKRs, you need to follow through with your team and measure them, monitor progress through the year, and celebrate successes when they get projects over the line. Goal setting can give your staff a sense of clarity, if done well. Your team should see that what they are working on is moving the needle. Otherwise you will all need to think through alternative approaches.

IF EVERYTHING IS IMPORTANT, NOTHING IS

You can't generate a million OKRs and metrics and expect a positive outcome. Part of looking at the big picture and prioritizing means letting go of some projects and being aggressive about what is most important.

You have to be clear what your highest goals are and be willing to let go of the others. There are a few ways of going about this:

- Take stock of all projects and prioritizing, cutting out what's on the bottom.
- Decide at the outset that particular projects aren't key to your strategy and cutting them.
- Keep something, but be clear that it's not a big priority, and give it a very long timeline. This one is not as recommended as the other

> strategies, but there are times when you have no other choice (for reasons like maintenance and dependencies). It can actually take more energy to entirely remove/kill a project than just keeping the lights on.

As hard as it is to say no to things, you have to in order to give your team bandwidth and space for what is critical. If you're in middle management, this may take a lot of work as you gather context from your reports and communicate the need for removing projects to above. I usually communicate with stakeholders by aligning them to what they want most, and make it clear that the risk is sacrificing those things if we don't properly prioritize. These are hard conversations, but the more data you bring to bear, and the more you treat it like a partnership, the easier they become.

CHAPTER 17

HOW TO SCOPE DOWN PRS

One of the most-valuable skills one can possess as a manager is to help your team scope down large, ambitious, abstract work into more manageable pieces, and often the smaller the units the better. Why the emphasis on something as seemingly trivial as size? For many reasons, but here are some of the most pertinent:

- It's easier on reviewers.
- It's easier to test.
- It's easier to iterate.

Helping a team scope down their work is one of those tactical skills no one taught me, but I learned on the job over time. If you've been a manager or led for a long time, you may already know how to scope down pull requests, but for those who are new to engineering management, it's worth covering it because it's an important part of the job. Let's revisit the three reasons why scaling down PRs (Pull Requests, also called other things, like CLs at Google) is so important.

SMALL PRS ARE EASIER ON REVIEWERS

If someone has to read through and make sense of a lot of code in order to evaluate it, it's easy to miss something. Think about it from the reviewer's standpoint: It's their job to make sure whatever code they're reading makes it through this process in the best state possible. The aim should be to lower the cognitive load that is imposed on any reviewer so that they can envision all the moving parts and any potential risks.

The higher the number of changes, the more conversation you'll have to have, until it is too big and then you'll get none. Large PRs also require reviewers to do much more work as a communicator, which takes two skills: the actual code review, and the ability to communicate how/why the changes need to be made.

As a reviewer, sometimes I'm handed huge chunks of code, some of which I would like to take, some of which I can't. This means I now have to break these concepts apart, explain the differences between them, and talk through an iteration plan. This is okay, but it requires a lot more mental overhead for everyone involved, particularly because it can potentially be damaging to the team if the communication goes poorly. Smaller PRs keep the conversation focused, and as such reduce the need for any communication gymnastics.

SMALL PRS ARE EASIER TO TEST AND ITERATE

Large PRs have the potential for more hidden side effects and moving parts you can't easily see. It becomes more opaque to see what's been altered. When there are fewer moving parts, it's not just more clear what the testing strategy is, but also how much coverage the tests offer.

For instance, if I have an enormous PR and there are a few tests, I still have to spend a good amount of time matching the contents of the PR with the types of tests involved to see if the coverage is adequate.

Having fewer moving parts also makes it clear what else in the codebase is affected. The bigger the PR, the more potential surprises you might have. It's easier to iterate if work needs to be adjusted, revisited, or not accepted. There's much less investment, both personally and in terms of developer time. The chance that you went in the wrong direction for a long time without any course correction increases with the size of the PR.

Some of this investment is personal: a person may be more attached to their work when they've sunk a lot of time into it and less willing to take in feedback. Some of this investment is a business concern: developer time is expensive. If you went heads down for a week and didn't communicate, and some of the work can't be used, that's not helpful to the company's bottom line or its ability to ship products.

When you create smaller pieces of work, you can adjust and communicate more quickly, even as things outside the project change or if priorities shift. Your teammates will also trust that they'll see subsequent work from you soon. You might be amazed at how much more quickly they're willing to review your work and get it in.

BREAKING DOWN A PROJECT

There are a number of ways to scope down work. Let's go over them in this section.

Clear Expectation Of Output

This one has less to do with tools and more to do with people. If your company has a project manager role (this is different from a product manager), they are typically very effective in helping break down tickets into more manageable units and estimating the size and time of the work. On some teams, this duty falls more to the engineering manager or product manager.

It's incredibly difficult to estimate engineering work; some things that you think will take two minutes turn into hours or days, or vice versa. So start with the end. What output do you expect to see from that ticket? Now we can walk backward. What parts of the codebase will be affected by this change? If you're newer to the project, ask someone with more seniority on the team to help you track which areas will need to be updated.

If possible, break those into disparate tasks. Even if they are dependent on one another, you can usually find a way to break it into step 1, step 2, and so forth. If you're using the branching model outlined below and communicate to your reviewers clearly, they know to expect many pieces of work from you and not one giant finished product.

Branching

In a branching model, you communicate what you're working on through the iteration of work on branches. Typically these branches are named semantically with the project in question: FeatureName or RefactorName, sometimes with versioning as well. The nice thing about using branches like this is that you can do frequent work in one place, and no one is expecting everything to be done in one go.

But branching does require coordination: People on your team should know that this work is correlated to a certain piece and not to merge it until the work is complete. If there is a ton of work that needs to be done on a specific branch, you likely need some end-to-end tests to validate that the merge won't break anything.

Issue and PR Labels

I typically have my teams use branching with labels to scope down work. They're very effective when used together. For instance, maybe you're working on one part of a particular feature that's part of a 1.0 road map. Your coworker is working on another piece. If you both use the same label, but different branches, you can clearly see who is working on what and make sure the changes are tracked together. This means everyone can work incrementally because there's insight into what work is done globally on a project as well as the small parts.

Feature Flags

It's possible to use feature flags to gate check and merge code slowly that you can test in production. When I worked at Trulia/Zillow, we used this model frequently with cookies that one could set in their browser, so that stakeholders could not just check against it, but we could also run split-testing and work out conflicts long before it ever hit primetime. We rolled things out to production slowly, monitoring the work in partnership with other teams as we went. It allowed us to be very coordinated, which was especially important for big releases with a lot of moving parts.

We made good use of feature flags when I worked at Netlify for our most recent release of faster deploys.[12] Specifically, we used a Ruby gem called Flipper.[13] We liked testing in production, and this process allowed us to roll out to production from day one, and make use of telemetry to monitor how well the changes were functioning.

Number Of Concerns

The most effective PRs I see and ones I love as a reviewer, are small, focused, and scope the work down to the smallest number of concerns. Think through your work in terms of how many things are changing, and try to limit this number as much as you can. There's a great post about the Single Responsibility Principle,[14] which outlines how every module or class should have responsibility over a single part of the functionality, and how this can be applied to pull requests as well.

At first, it might seem critical that all changes be made at once, but with the branching model you'll find over time that this isn't necessary. You can communicate to people within the PR that there's more work to do as well.

Furthermore, each PR should, when possible, only have one goal so the next iteration can build on it. In other words, if you and a teammate are both refactoring old code, and introducing new tests, that's much harder on a reviewer than if it is split. If you have two iterative PRs, it's easier to ensure the functionality didn't change prior to modifying the implementation.

TRACKING WHAT'S LEFT

If you're going to leave unfinished work, it's helpful to keep track of what's left. This is not just for you, but for any stakeholders like your teammates or product managers and project managers. A lot of teams will make use of project boards for this, which also helps the team to review what's going on in the project. See the example GitHub board in Figure 4.

Some tools, like GitHub and Jira, will help you with some automation as well. See Figure 5 for an example of an automatic project board that is a feature of every repo.

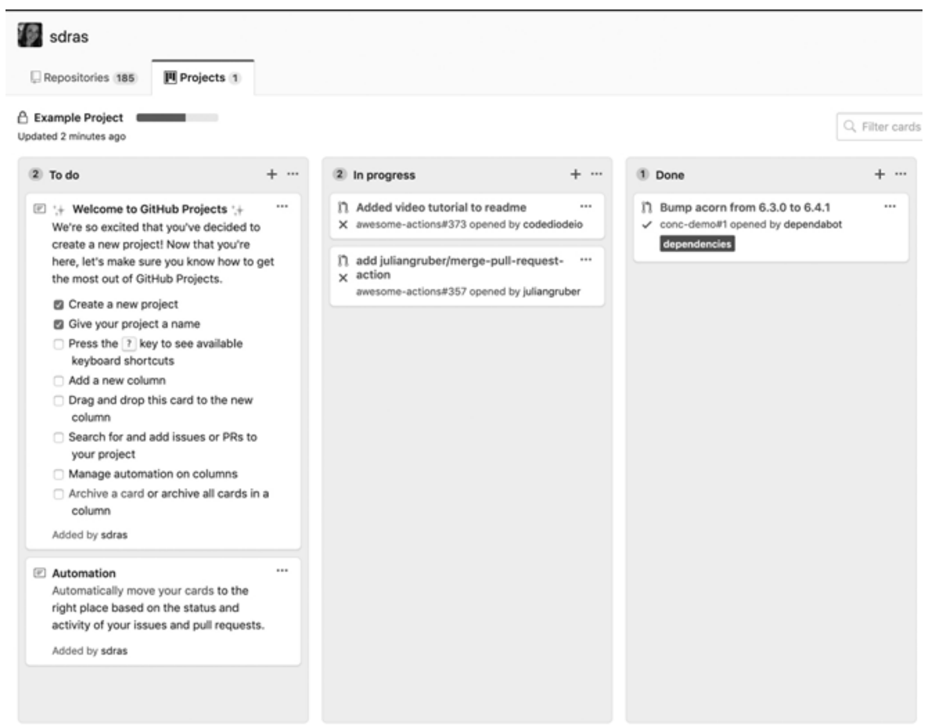

Figure 4: Example GitHub board. You can add issues as well as notes.

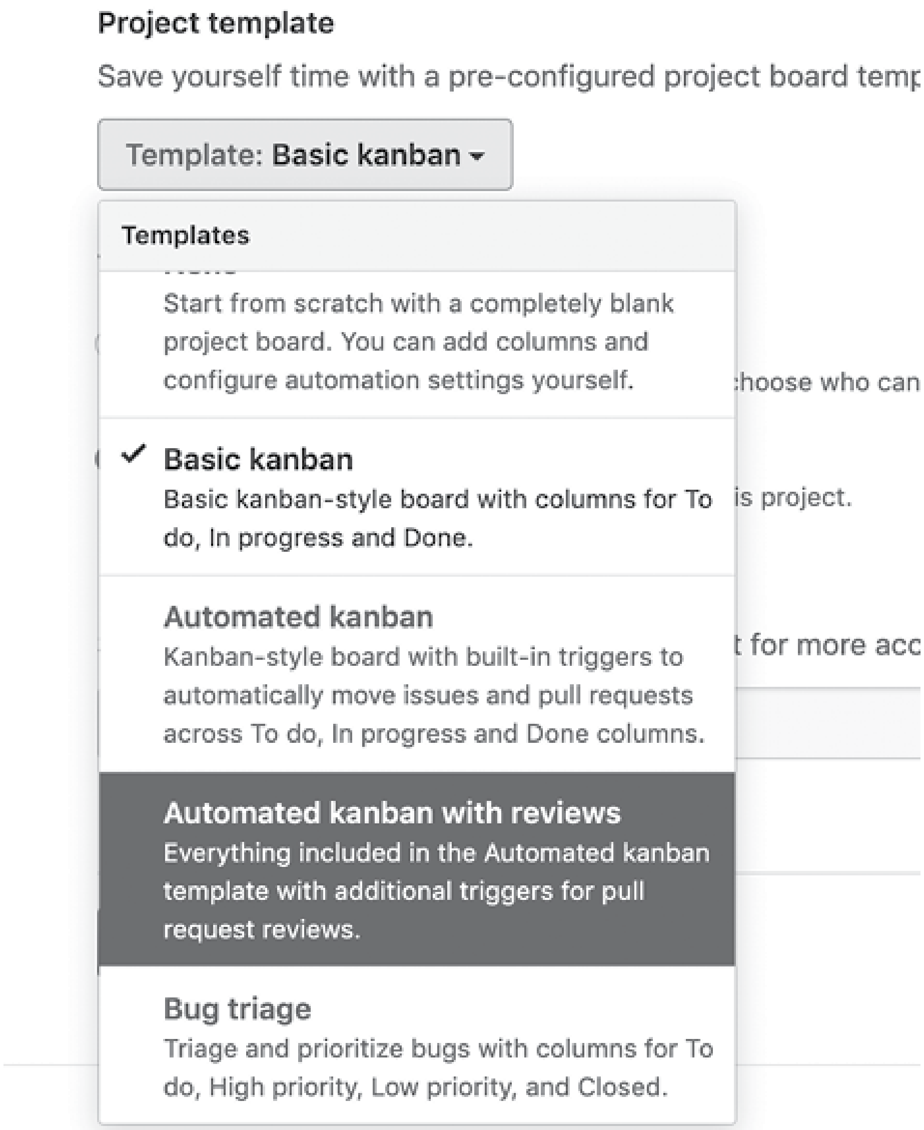

Figure 5: A dropdown of kanban views available in GitHub.

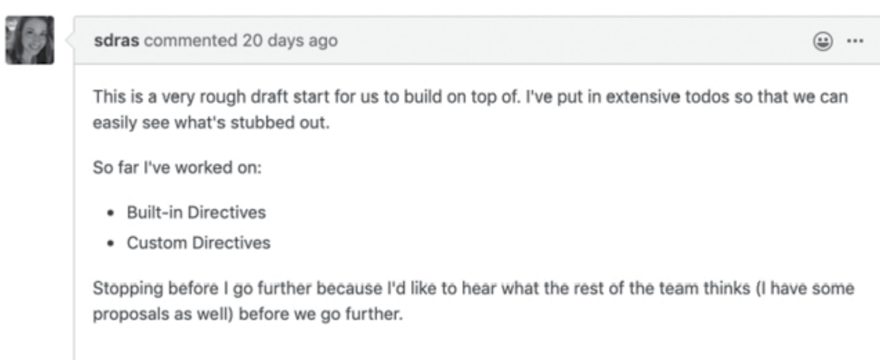

Figure 6: A description of what the PR contains, and what it doesn't.

Figure 7: An example of a TODO extension highlighting TODOs and FIXMEs.

Even if you're tracking things within these boards, it can also be useful to your coworkers/reviewers to generate a to-do list or list of changes in the PR itself. This helps them to immediately see the scope of the PR without having to check any other references, which can quicken the speed of execution. See Figure 6 for an example of a description that can be added as you submit a PR to give other collaborators context.

As mentioned above, you can also make use of to-dos. I often do this, and will track what's left with [this VS Code extension][15] that goes through and highlights any TODO comments for you. You can also override and customize it, should your team have a different convention.

These are just a few ways to track work at the developer level. Truly there is a whole project management discipline that employs many different tools and techniques! Covering everything is out of the scope of this book, but hopefully some of these tools help your team during coding.

Working On A Few Pieces At Once

This one is tricky, but a skill you might develop as you level up. As you get better at scoping down work, you'll find that there are times that you're waiting for review on these smaller pieces. So what do you do with that time? In fact, I think this is where some will assume that one large PR is more effective because they can spend the time working before submitting the PR. Unfortunately, this will often come back to haunt them when it gets locked up in review.

As you get better at breaking large projects down, you also start to learn how to cut work up into many pieces, the result being two concurrent pieces of work that are not reliant on each other. Perhaps in a personalization project, changes need to be made within the navigation as well as updates to the data coming into the sidebar. They may be similar pieces, but figuring out ahead of time how to break them apart does a few things: it can cause fewer side effects as the navigation and the sidebar are now not potentially going to break one another. While you're waiting for the navigation review, you can get cracking on the sidebar piece. This is a simplistic example, but one that illustrates that you can plan for this situation in order to make the most productive use of your working hours as possible.

WRAPPING UP

We talked about OKRs in the last chapter, which can give us a big picture and also help us align with other stakeholders. But OKRs will never give us a sense of how to actually break down tasks in the day-to-day with our teams.

Bridging the gap between the immense amount of work and the actual PRs that update our codebase can help us work through tactical work with our teams: in execution and in collaboration.

12. https://www.netlify.com/blog/2020/03/25/announcing-faster-deploys-for-large-sites/
13. https://github.com/jnunemaker/flipper
14. https://hugooodias.medium.com/the-anatomy-of-a-perfect-pull-request-567382bb6067
15. https://marketplace.visualstudio.com/items?itemName=wayou.vscode-todo-highlight

CHAPTER 18

SPEED OF EXECUTION

I like to move quickly. The teams I work with tend to deliver fast as well. We do not, however, deliver the full project in any kind of resting place. And this chapter will explore how we can do so.

First I use caution, and with so much to learn, I see value in iteration. When we talk about speed of execution, we are not talking about sloppiness. We're talking about a speed in which you can move forward and feel a bias toward action, without incurring massive tech debt. But this is much easier said than done. Quick iteration cycles require the ability to scope down work and prioritize, as covered in the last chapter. However, a few other considerations are at play.

SET DIRECTION FIRST

My friend Jose Gonzales was the head of Product at Netlify while I was working there. He used to say: "It's not enough for us to go fast. You can run fast in the wrong direction."

As you might assume, he was a great product manager. It's true, speed of execution alone will not make your product or company great. Arguably, at any company of size, engineering as a function is assigned the duty of the execution piece. So you either need to partner with a product function for the *what* while you address the *how* and *how fast*. Or if your company doesn't have a product function, you must do both. But you cannot merely move quickly without forethought and assessment about the *what*. That is crucial.

We need to step back and look at the big picture.

If we constantly work on tasks without taking the time to think bigger, we risk being tactical rather than strategic. We can end up working in a silo, which can lead to burnout and anxiety. We risk opportunities to spot errors early and to reduce technical debt[16].

WHAT IS AN MVP?

The idea of quick iteration cycles and the concept of an MVP (minimum viable product) have become fairly industry-standard in recent years. An MVP is a way to ask the team, product, and yourself, "What's the smallest amount we can ship?"

Here's where a lot of projects fail. I have on many occasions defined the MVP in too limited a manner; it's so stripped and bare bones that none of the exciting reasons exist for working on it in the first place. Or they exist, but are unusable because the happy path is too narrow, and only a few people can use it.

The truth is, a customer's time is precious. If they have a horrible time onboarding or using your product, they're not coming back for when you "improve it next time." A small counterpoint to this is a developer-focusing product that you label "beta." Some developers will definitely give you the benefit of the doubt to try it again. Fair warning: others may not.

This does not mean you need to make the MVP perfect. A narrowly scoped MVP is a wonderful thing to do user research and testing on, and then iterate on internally. But please consider this a balancing act. Wait until perfection to put it out and you stand to burn out a team, lose profit, and be slow to market. Put it out while it's half-baked, and your audience will lose trust and not come back when you feel like it's finally complete.

The other hidden peril of an MVP is internal. Sometimes when stakeholders see an MVP out in the world, they will have you move on before you are able to get to final iteration cycles. Why? Here's the hard truth: sometimes it's not what you initially agreed on. Sometimes you are overruled, not out of malice, but because of competing priorities. And sometimes this scenario incurs some tech debt on the engineering side for things you said the engineers would have a chance to clean up later.

There are tons of different ways to work through this and solve for these conflicts. Similar to gambling, where you never bet more than you're willing or able to spend, never put out an MVP you're not willing to let lie.

HOW DO WE MOVE QUICKLY?

The answer is not just telling people to move faster. It's more complicated than that. So how do we move quickly?

I'll answer this question with a question. Who was your favorite teacher in school? Chances are, they weren't even the teacher of your favorite subject. Perhaps your interest in that subject grew because of that teacher. Chances are, they were extremely excited about the material they taught. They understood it to great depth and were also enthusiastic about it. Engineering management can work like this too.

A team that's motivated, has psychological safety with you and among themselves, has good developer experience within their tech stack, and feels aligned with the company's goals will always perform better than one that isn't. These ideas are connected with speed of execution. Let's review that sentence, and break the characteristics into actions:

- **Assess motivation:**
  - Are there external circumstances that the team is dealing with? In this case, fixing and ignoring might not be appropriate, but acknowledgment and care might be. For example, in a pandemic, understand that everyone might be scared for reasons outside of your control. They may have family they're having to deal with during working hours that

they didn't before. Talk to stakeholders and adjust timing. Figure out what's reasonable and remember that they're people, not machines.

- **Assess psychological safety:**
  - Are folks looking forward to getting their work reviewed so they can move forward or is it a painful process? Do they trust each other to give good reviews? How quick or slow is this process for them?
  - If there are issues or concerns with the direction, how are they addressing this? Are they comfortable bringing them up? Conversely, are people dying on every small hill?
  - Are they talking about things to further their own career and ego, or for the betterment of the project at hand? If people are performative in their execution and reviews rather than supportive of the flow of the project, it could be a sign of some cultural problems that need addressing. Happy and productive teams seek out the best technical perspective over their own personal glory.
- **Assess whether the tech stack is serving their needs:**
  - Are they able to make changes without a ton of side effects and other pieces breaking?
  - Is the stack new enough that they can use modern languages and libraries if they see fit?
  - Do you have an appropriate language or style guide in place, as well as linting systems so that PR reviews aren't getting stalled on stylistic details?

    - Is there a part of the system that everyone complains about? If your tech stack had a map, does it have an area marked “thar be dragons?”
    - Are there appropriate CI/CD systems or other tooling in place so that they’re not constantly yak-shaving and context-shifting to get their work done?
    - Most importantly, can any single individual easily and accidentally bring down production or a shared local environment? (Surprise! This one impacts psychological safety too!)

- **Align the team with the company’s goals (the why):**
    - Why are they doing this?
    - Why is it important to the business?
    - What risks will there be if it’s not done?

Doing these types of assessments and doing diligence into what processes and human work you may need to investigate is a bit of a loss leader: It may take time on your end to build trust and get a system working well, but it pays off in dividends.

In fact, the more your team shows that it can deliver, the more you build trust with other organizations as well! The product team, the sales team, the marketing team—building trust with other teams and being a reliable leader that others can depend on can be a hugely impactful piece of working well together as a company. What’s more, when you inevitably do hit a snag (as we all do), these teams will have more patience and understanding in these moments.

16. This article is one of the clearest breakdowns of technical debt I've seen and comes with some great visualizations to boot: https://blog.crisp.se/2013/10/11/henrikkniberg/good-and-bad-technical-debt

CHAPTER 19

SPLITTING TIME BETWEEN PRODUCT AND ENGINEERING WORK

At each company I've worked at, we have had a split between time spent on product initiatives and time spent on engineering work. The percentages always changed, sometimes 70 percent product, 30 percent engineering, sometimes as much as a 50/50 split. The impetus for these splits is to ensure that Engineering spends a portion of their time building new features, but also to ensure they can do their "own" work such as address technical debt, upgrade systems, and document code.

The trouble is, it's one thing to declare a split at the outset, and another to make it a reality. I've seen this model fail, not because people don't understand that it's valuable in theory, but because of some common pitfalls in practice.

- **Product may have conflicts, either with the work itself or the time involved**. This can strain the relationship between product and engineering. If they are caught by surprise, you can potentially find the boundaries of your work getting more restrictive.
- **Your engineers might not understand what's expected of them**. Parallelization of efforts can be hard to do, so building a good process can provide clarity.

- **Maintenance path should be clear:** Are you planning on making a giant system upgrade? This may affect other teams over time and if you're not clear about eventual ownership, it could come back to haunt you. 👻

As nice as it is to have some freedom in your engineering work, communication, and solid processes with clear expectations (product work) can help make sure that you avoid the issues outlined above. In this chapter, we cover some potential scenarios from the perspective of an engineering manager so that you can have a productive path forward.

COMMUNICATION, GENERALLY

Once you figure out what problem you want to tackle, it's critical to **write up a small one-sheeter** that you can share with stakeholders. This will cover the nature of the work, the amount of time it's going to take, and why it's important.

With a large project, you can also scope those pieces down into GitHub/GitLab/Jira issues, and add a label for the type of work it is. This is great because you can use a familiar project management system to elevate the amount of work and expectations weekly. It's good to keep the dialogue open with your product partners on scope and the nature of the work so there are no surprises. What this dialogue looks like will largely vary by the culture of the team and organization.

This kind of regular communication can help provide clarity for your engineers too. If they understand the nature of the work and what's expected of them, it's easier for them to tackle the small issues that make up a whole.

You may find that it makes less sense from a focus perspective to have every engineer split time across product and engineering projects. They may instead prefer to split the work up between themselves: three people on product work for a few weeks, one person on engineering work. There are also times where everyone does need to be involved so that they have equal institutional knowledge (migrations can be like this, depending on what it is). Your mileage may vary based on the size of the team, the amount of product work, and the type of project.

Communication helps here too. If you're not sure what the right path is, it can help to have a small group brainstorm on how you want to get this done. Just be sure you also align everyone with *why* the project is important as you do so.

TECHNICAL DEBT

Technical debt is one of the most common ways to unlock your team. For every feature you write, if engineering effort is slowed, you're not only losing time in terms of product development, you're also losing money in terms of engineering time in salary.

A bit of technical debt is natural, particularly at smaller companies where it makes more fiscal sense to move quickly; but there are some points where technical debt becomes crippling for development and releases, and makes a codebase unstable. Sometimes technical debt needs to be incurred immediately to make sure all your engineers can work efficiently, and sometimes it's incurred gradually.

In a lot of cases, the technical debt pieces are learned by a bottom-up approach: typically, developers who are closest to working with the system will know better about what day-to-day technical debt exists than engineering managers. The challenge as an engineering manager is to notice larger patterns, like when many folks complain of the same thing, rather than one developer who may have a strong opinion. Asking around before you start this type of project can help—poll people on how much time they think they're wasting in a given week, versus the prospect of an alternative.

Sometimes technical debt is a matter of a large amount of refactor. **I've seen this go best when people are up front on what kind of PRs are necessary**. Do you need to update the CSS in a million spots? Or convert old class components to hooks? You probably don't want one huge PR for all of it, but it doesn't make sense to break this work per-component either. Work together as a team on how much each PR will hold and *what is expected of the review* so you don't create a "review hole" while the work is being done.

INNOVATIVE PROJECTS

A lot of companies will do hack week, or innovation week, projects where developers can work untethered on some feature related to the company's product. It's a great time for exploration, and I've seen some powerful features added to well-known applications this way. It's also incredibly energizing for the team to see an idea of their own come to fruition.

> *NOTE*
> *The trouble with doing these kinds of projects in the split engineering time is they can at times make the product team feel a little slighted. Why? Well, think of things from their perspective. Their job is to put forth these features, plan carefully with stakeholders, put together road maps (often based on company metrics and research), and get on the engineering schedule, usually working with a project manager. If you spend half your time working on unplanned features, you can potentially fork an existing plan for a project, go against some of the known research they have, or simply slow down the process to get a core make-it-or-break it feature they need.*
>
> ***The way I've seen this play out well is when the engineering manager communicates up front with product.*** *Consider this a partnership: If product says that a particular feature doesn't make sense, they likely have a good reason for thinking so. If you can both hear each other out, there is likely a path forward where you both agree.*

It's good to address employee's fears too. Are they concerned that there won't be enough time for innovative work? Ask your team directly how many weeks, at halftime, they think the work might take (with the expectation that things might shift once they dig in). Make it clear to everyone that you don't expect it to be done at a breakneck pace.

Ultimately, communication is key. Ideally these are small projects that won't derail anything and that can be done in parallel to the regular work. My suggestion is to try it with something very small first to see what bumps may be in the road, and also to build trust with product. They need to be assured that you'll still get your work done and not "go rogue."

The final piece of this is to figure out who is responsible for metrics and outcomes. Part of the reason product gets to decide direction is because they're on the hook when a project fails. Make sure you're clear that, as an engineering manager, you're taking responsibility for outcomes, both the good *and* the bad to maintain a good relationship with product.

SLOW, ONGOING WORK

Slow, ongoing work is probably the most clear-cut of projects and will likely get the least amount of pushback from anyone. Examples of this type of work are internal documentation, tooling (if you don't have a dedicated tools team), or small bits of maintenance.

The communication needed here is a little different from other projects, as you're not necessarily going to be shipping one constrained project; rather, this is an iterative process. Take documentation as an example: I suggest building time for internal documentation into any feature process.

For instance, let's say you created a new feature that allows teams to collaborate. Not everyone across the company may know that you created a microservice for this feature that any team can use, and what parameters are expected, or how to add functionality down the road. Internal documentation can be the difference between the service being used or not, as well as your team being asked to pair with someone every time it needs to be used. Or worse, your peers trying to hack around and figure it out on their own, creating a mess of something that could have been worked on quicker and more efficiently.

Unlike the innovation projects, slow, ongoing work is typically not something folks really crave doing, so setting a process and expectations straightaway works best. Internal documentation is a sometimes hidden, but very important part of a well-functioning team. It helps with onboarding, getting everyone on the same page about system architecture, and can even help developers solidify what they are building and think through what problem it's solving.

MIGRATIONS

Migrations are handled a little differently than some other types of projects because they likely affect everyone. There is no single right way to do this, and the process will also largely depend on what type of migration it is — framework to framework, breaking down a monolith, and migrating to a different build process or server all may have different approaches. Because detailing each of these types of migrations would likely require a book of its own, let's go through some high-level tips that apply to all of them:

- **First do as much research as possible up front on whatever type of migration you're doing.** There's no way to know everything, but you don't want to get partway through a process to find out something critical. This is also helpful information to share with stakeholders.
- Are there internal debates about which direction your company should head? If so, **timebox a unit of time to work through this problem and make sure you have a clear decision-maker at the end.** A lot of tech problems don't have one "true" solution, so

having one owner make the decision (and everyone else commit) can help. But you also want to give folks a moment to have their voices heard, even if they are in disagreement. They might be thinking of something you're not.

- **Document a migration plan, both at a high level and a low level, and then work through the impact on each team.** This is a great time to explain to product why this migration is important: Is your codebase becoming old and no longer playing well with other libraries and tools? Did a new build process come out that could save your engineers time in a release process? Help product understand why the work is critical.
- **Be clear about maintenance and ownership**. If one team migrates a build process that then causes issues for another, who's fixing things to unblock that team? You should decide this before it happens.
- Some migration paths allow you to do things slowly over time, or do a lot of the work up front. However, there is usually a time when it's going to be critical and all hands on deck are needed. Unlike some of the other work that can be parallelized, **you may have to work something out with product where all other feature work is stalled for a little bit while you get the new system in place**. If you work closely with product, you may find that there are times in the season where you naturally have more of a customer lull, and it could give you the breathing room you need to get this done during that time. If they're willing to let you take engineering time to 100 percent for a little while, you return the favor; once the platform is stable, dedicate 100 percent of the team's time to product work.

CELEBRATE!

This final step might seem optional, but it's a big deal in my opinion. Your team just pulled off something incredible: They parallelized efforts, they were good partners with product, they got something done for the engineering organization at large. It's crucial to celebrate the work of completing a migration just as you would a launch.

The team needs to know you value this work because it's often thankless, but very impactful. It can build trust and help their career path as well. Celebrating what you've accomplished as a team costs very little and has great cultural benefits.

PART 4

CHAPTER 20

PRIORITIZING, HIGH LEVEL

You're faced with a lot of decisions in your everyday work. There are multiple tasks calling for your focus, and you can burn daylight or even burn out trying to decide what comes first.

Related to this is a phenomenon called "decision fatigue."[17] Many studies show that you can make poor choices when you're not able to make a clear decision. The inability to see what is most important can lead to impaired judgment and even poorer purchase decisions.[18] You are more likely to buy something on impulse after you have already made many other decisions. This is why the aisle next to the cash register is reserved for "impulse buys." You have used all of your decision-making processes as you went through the store, and you're more likely to buy a dumb magazine or something with sugar when your brain is low on energy and reserves.

This happens with our work priorities too. If we never take a step back and reflect, and create a high-level system, then we need to constantly make those decisions again and again each week, which sops up some of our very necessary cognition. And in leadership, you need to protect as much of that cognition as possible.

Being overwhelmed with options, decisions, and meetings doesn't happen all at once either. Typically this type of bombardment is cumulative. This means it's pertinent to work prioritization into your everyday life.

So how can you figure out what's most important to work on first, or

even what tasks to work on at all? In this chapter, we explore how to sift through the inevitable weight of our to-do lists so that we can be efficient and clear with our direction.

If you feel like your to-do list is ruling you instead of you ruling it, read on.

VALUES AND PRIORITIZING WORKSHOP

If you're going to think smarter (and not harder) about how you prioritize tasks, you have to invest a little time away from your to-do list.

It's important that we be honest with ourselves; what we work on is an expression of our values. It may be no big thing to work overtime for a while, but something will be sacrificed with this decision. If you're a parent, you might spend less time with your kids. Maybe it's less time hanging out with friends. Maybe you give up sleeping or eating well.

Similarly, if you choose not to work much, you're deciding that your values lie outside your career, which is okay too. But then you could be compromising support for your team.

Where you invest your time is not just about what you value, but also what you don't value. At the very least, we want to make our time investments align with our values as much as possible, and to be intentional. This starts with looking at the big picture and then refining until we reach what we do on the day-to-day.

The book *Hell Yeah or No* captures this concept well. Author Derek Sivers breaks this down further, writing:

> *"Your actions show you what you actually want. There are two smart reactions to this:*

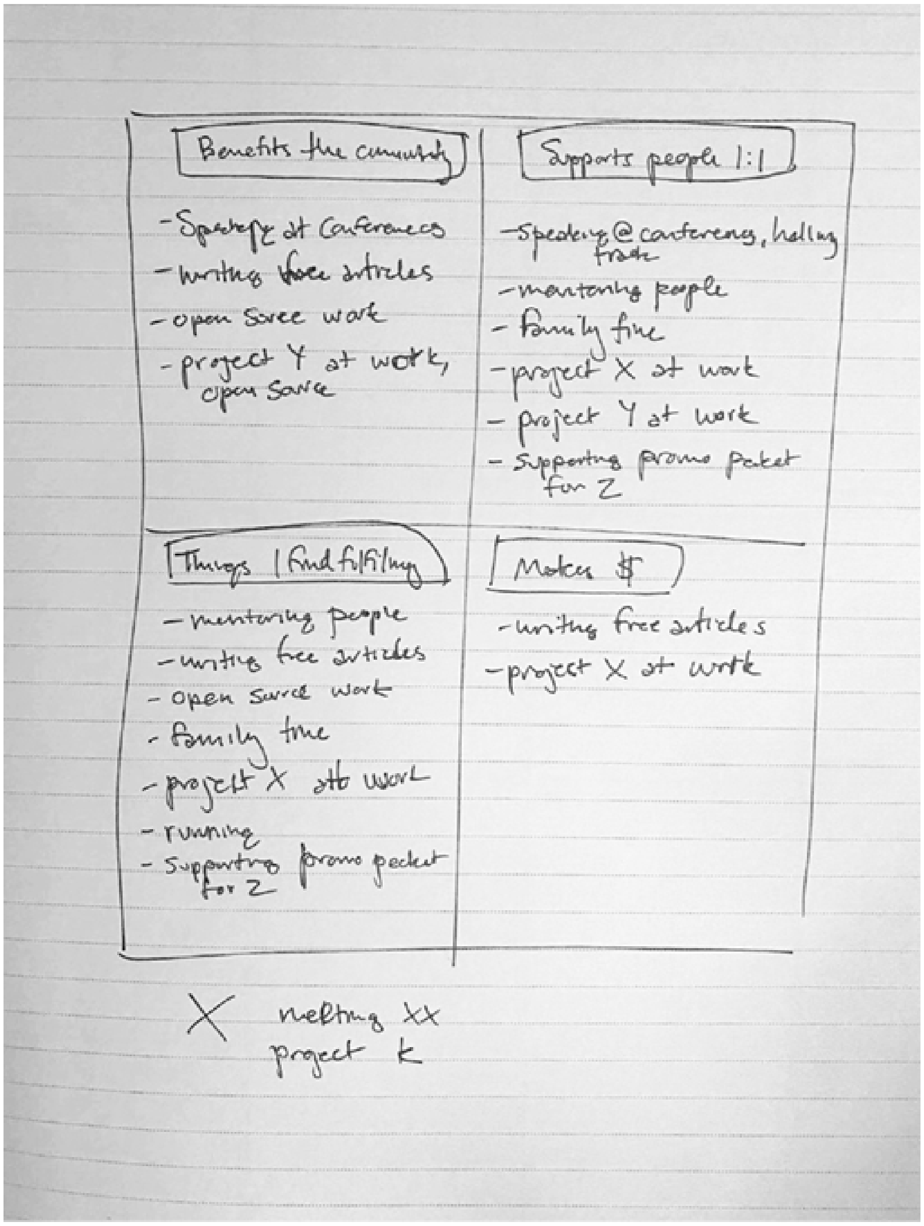

Figure 8: Example of a quarterly grid exercise (I've generalized and simplified mine for demonstration).

Stop lying to yourself, and admit your real priorities.
Start doing what you say you want to do, and see if it's really true."

As a manager, it is crucial that you're prioritizing your own time well to best drive big impacts, and also to be aligned with what you feel is important. This is much easier said than done.

Every quarter or so, I break down all the tasks I've committed to doing. I write them all down, and I create four quadrants containing all of the things I care about. For me this is: helping the community, helping 1:1 relationships (which can include coworkers, friends, and family), making money, and whatever else I find personally fulfilling. Your quadrants can and should be different, as you have your own set of values.

Then I take all the tasks I have going and place them in the quadrants. I see how many of the boxes each one takes up. Some endeavors are counted in a quadrant twice to add weight to it. See Figure 8.

Here are some of my key findings:

- Writing articles is one of the rarest instances where it is featured in all quadrants.
- Mentoring people is in two quadrants: building 1:1 relationships, and whatever I find more personally fulfilling.
- Open source work is also in two quadrants: helps the community, and things I find personally fulfilling.

Anything that fills just one quadrant has to be reconsidered. Anything that does not fit in any quadrant is on the chopping block.

This activity allows me to see where I want to invest my energy, and where I shouldn't. It's illuminating because I sometimes get bogged down in the noise and it's hard to see signals. It can be tough to reflect on what I value and how to prioritize tasks around those things until I do this exercise.

From here, the rest of the prioritization gets a little easier. I now have a better idea of where my efforts are really paying off. I also know when I'm wasting my own time. This distinction helps me prevent decision fatigue quite a bit because I have a guiding light to help make choices.

This is just one of many methods of high-level prioritization based on your values. Find a process that works for you, and do it at a regular cadence. As mentioned in the Values chapter, your values change over time as well. Make sure you are open to adaptation and feel empowered to question the base premise.

Now that you have your high-level priorities in order, let's talk about how we can best schedule our time. Battling calendars can become one of your highest priorities, even if you have an assistant helping you. Let's break this down a bit more.

VALUES AND BATTLING SCHEDULES

About a year ago, I realized my calendar was so booked, I could no longer eat lunch or catch my breath. I knew something had to give.

I couldn't figure out why my calendar was always filled to the brim. Sure, we were trying to accomplish a lot, and that required a ton of coordination. But surely there was some sort of light at the end of the tunnel, right? Every week seemed temporary. If we could just *get through that week*, the next one will be better.

Then I worked with Whitney Stafford, an executive coach at Netlify. She had me fill out a form that asked me to detail my values, priorities, and projects. I also wrote down all of my standing 1:1s, skip levels, recurring meetings, and periodic meetings. Then as one last step, I also wrote the priority order of all of those meetings.

I explained that 1:1s and team meetings (on teams that reported to me and also the leadership team) were the most important part of my work; if I wasn't supporting those teams I was on, not only was I not doing my job, I also didn't have enough information from my teams to coordinate with anyone else on their initiatives.

Your values have to be reflected in what you schedule. If you say you care most about your family, and make no time for them, then you're not honoring your own value. If you say you care most about creating a supportive work environment and never take time to meet with your team, your values are misaligned.

This all means there's a bit of a balance to be had. And it's not something that's done once; it has to be done continuously. You can, however, try to optimize your recurring schedule as best you can so that at least in a default state, things aren't hectic and overwhelming.

If you don't have a coach, here's how you can go about doing what Whitney did for me:

- Think about that values workshop we did and the projects that filled out the most quadrants.
- With that fresh in your mind, pull all the recurring meetings out of your calendar and put them onto a list.
- Map from organizational grouping to meetings into priority order (and don't put meetings with your own team last).
- Figure out which of the meetings could be dropped to a less-frequent cadence, or canceled, and contact the people involved in those meetings.
- For ad hoc meetings, do the same, and note major themes. Are you having multiple meetings to address one thing? Could those meetings be combined? Are you frequently in ad hoc meetings and 1:1s with the same person? Could the ad hoc discussion happen in the 1:1?

Now address focus time:

- Decide how much time you need each week to focus.
- Note when you do your best "solitary thinking" versus "meeting thinking" work. I have the best social energy in the mornings, and focus happens best in the afternoons. Your mileage may vary.

- Also note how frequently you need breaks. I used to schedule meetings back to back all day long, which never allowed me to come up for air. Now I try to schedule a thirty-minute break for every 1.5 hours of meetings, with some flexibility to go to two hours.

Prioritizing Based On Energy Levels

I've definitely had times in my career where I've suffered because of sporadic meetings with ill-defined amounts of time in between that were too small to be heads-down coding, and I had to make up for it at night. That's not a recipe for success; it's a recipe for fatigue.

When I started at Google, some asked me why people stayed at the company so long. My honest hypothesis is that at Google people do schedule breaks and focus time, and don't work on the weekends unless things are truly on fire. This was very different from the startups, or even some large companies I worked at. Longer term, respecting breaks and balance can affect things like attrition.

In a great blog post by Alex Sexton titled, "The Productivity Cycle,"[19] he suggests planning based on personal energy levels, noting where your own cycles occur. *This is brilliant!* Below we're not using this system exactly (though I highly suggest reading through this thoughtful article), but rather thinking through that base premise to evaluate our own calendars and work with our natural cycles instead of fitting our energy around our calendar.

The first step is simply awareness: For two weeks, note your energy levels throughout the day. Next to every task, put a ✔ for good energy, put a ━ for so-so energy, and put a ⊘ for no energy (you can use other symbols if they suit you). Note some patterns. When are you losing steam? When are you most focused? Notice when you're more productive for social activities, and when you're better off working in isolation. The more you study yourself, the easier future planning becomes.

Second, we can now apply what we noticed to our calendar: For the ⊘ tasks, try to build in breaks. Put a block on your calendar for a half-hour where people can't schedule you. For the ✔ times, try to build some of your more-important alignment meetings in these spots.

Context-Shifting and Batching

When possible, I try to schedule similar tasks around the same time. The higher you go in the management chain, the more disparate projects you will be working on. If there are many meetings needed for a specific project, it can be nice to batch them so that you're not context-shifting and losing information between meetings. It can also be hard if you're jumping from problem to problem to problem without cohesion. Batching similar project-based meetings together in a row can help reduce cognitive leaks and keep you focused.

I like to group 1:1s as well. When I had a small number of reports, it was fairly easy to group all 1:1s together. I could batch them in a time when I had a lot of energy and approach them with some vigor.

Over time, my reporting structure grew, and that option was no longer available to me. What I tried to do then was batch the 1:1s that had similar themes. Are you working on a particular feature? Try to batch the 1:1s for all the folks working on that same project. Do you have a meeting that takes a lot of coordination later in the week? Batch the 1:1s that give you pertinent information at least one day before that meeting so you have the information you need, and you come prepared. And so on.

BUILDING A HIGH-LEVEL SCHEDULE

Now we have the data we need. From here, we can decide on the bigger picture for our weekly calendar. We can set up our project-based meetings, team syncs, and 1:1s in a cadence that our calendar allows, that reflects our priorities, and that builds in focus time, breaks, and cushion for one-off meetings that come up. We can build up blocks of meetings for similar projects so that we're not context-shifting. We can spend more time where the stakes are higher and have less frequent touch-bases where things aren't so pertinent.

This high-level schedule is something to reevaluate quarterly as projects shift and organizations change, and you may iterate on when you focus the best.

EMBRACING OUR LIVES, AND THEIR IMPERFECTIONS

Hopefully if you're prioritizing well, the amount of time that you have to work is structured in a way that actually *relieves* stress and allows you to accomplish the most you can, when you can. The goal of prioritizing is to bring into focus the work you find fulfilling, and remove tasks that don't. That's the sweet spot.

At the same time, we should give ourselves a little grace. No one can do everything, and there's really no such thing as perfect balance.

We're trying to create some prioritization and order in our lives; and it's not something to beat ourselves up about either. So turn down the volume on that self-hating podcast in your brain, okay?

This chapter has offered some tools that hopefully help bring order to the necessary mess in our lives. But there are times when life is more peaceful and times when life is more chaotic. Forgive yourself when life doesn't have perfect structure, and just do your best with whatever tools you have.

17. https://en.wikipedia.org/wiki/Decision_fatigue
18. This article explains why after a long day of work, it's tougher for me to resist those snacks in the mini kitchen despite knowing for a fact they give me a stomachache! Thanks, science: https://academic.oup.com/jcr/article-abstract/39/3/585/1822634?redirectedFrom=fulltext
19. This article by Alex Sexton isn't just great because the concept is great, but the type of introspection he does is truly remarkable: https://alexsexton.com/blog/2014/1/the-productivity-cycle/

CHAPTER 21

PRIORITIZING, DAY TO DAY

We've worked through some ways to gain clarity about our high-level goals and schedule. Now let's talk about what we can do to prioritize our day-to-day work.

In order to figure out a plan, the first thing to do is gather all of your tasks, large and small. Letting those things fester in our minds can make us feel burdened by what we have to do, so get it all out on proverbial paper. You can use notebooks, a homemade to-do list, applications like Notion, Evernote, or Clear. It's up to you.

Now you're ready to start day-to-day prioritizing.

PRIORITIZING THE TASKS

"It's amazing what you can get done if you just sit down and do it."

—Chris Coyier, creator of CSS-Tricks and cofounder of CodePen

Some tasks are pretty straightforward. You need to do the thing. It's indisputable. Other people are counting on you, or your future self will depend on it somehow. These are actually easier to prioritize. Then there's the other tasks. How do we go about prioritizing all of our tasks?

First, I break down large tasks into smaller pieces. That helps me put things in the following order:

1. Things that are actively on fire or are time-sensitive
2. Things that can be done quickly
3. Things that need a scheduled block of time
4. Things that I may get to further along

Part of the reason we do the small things early on is because **morale is important**. When I accomplish a task, I feel more inclined to get my other work done because being productive feels good—a small dopamine rush[20] is associated with every check.

This is also why I put things I've already accomplished on my list. It may sound silly, but acknowledging accomplishments makes me more likely to keep going and pushes me through the more complicated tasks.

SCHEDULING THE TASKS

I like to use the weekly planner in Notion for scheduling tasks (See Figure 9). I'll take all the things I think I can get done in a week and break them down day by day, based on when it makes the most sense to do each task. When I finish a task, it's dragged to the top and all "done" tasks are grouped together.

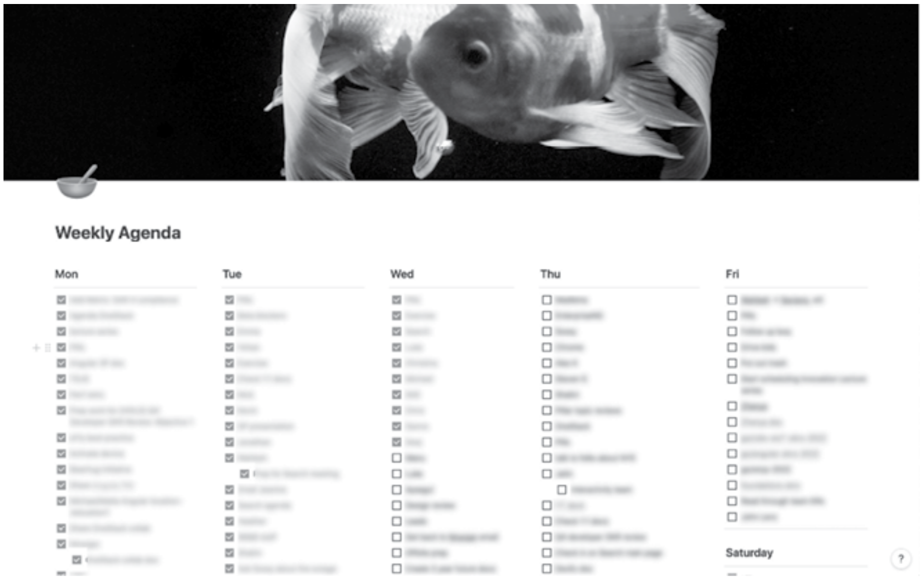

Figure 9: A view of a half-completed weekly agenda.

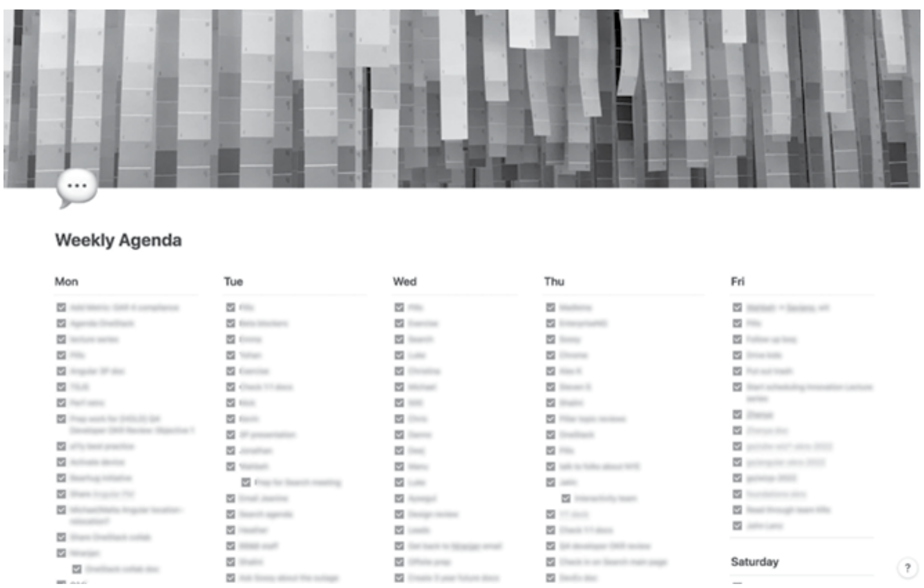

Figure 10: A view of a fully completed week's agenda. Anything not done will carry over into the next week's board, spread across days.

By the end of the day, anything that's unchecked at the bottom moves over to the next day. At a glance, I see if things are piling up on a particular day, which signals when I need to shift priorities around. Anything left over by the end of the week is dragged to next week's board.

I also have a category for "General To-Do" that is for longer-term tasks or those that aren't high priorities. I check in on that list every once in a while and pull tasks into my "local" weekly calendar when it makes sense.

I like this approach much more than a kanban board or a homegrown to-do list for the week because they feel unsatisfying and never-ending. In this model, the big picture is broken down day by day, and at the end of the week, it's clear I've gotten things accomplished, which helps me feel motivated.

I do keep a kanban board for larger projects, but day to day I try to maximize my dopamine: I give myself credit for anything I do, even small tasks and meetings, never leave something hanging, always marching on.

Find a system that motivates you and allows you to be proud of the progress you're making, despite the fact that new things are appearing all the time.

In The Case Of Emergency . . .

In this model, you have space to reconfigure and prioritize if something pressing comes up. You should never consider your schedule locked up at the week's start. If something pressing comes up, it goes on your calendar for that day, and you can move items off from that day to future days, knowing you will get to them when the time comes.

Create Systems

> *"If you want better results, then forget about setting goals. Focus on your system instead."*
>
> —James Clear, *Atomic Habits*

Figure 11 is a template I created in Notion that I duplicate each week, because I have meetings that stay consistent, or errands like "feed the dogs" that will always be there. Having a template allows me to move a little quicker to build out the schedule because all of my weekly recurring meetings are already in here, and I'm only adding the new to-dos and the ad hoc meetings.

This goes back to decision fatigue. The more we rely on grander systems to automate our work, the more mental energy we have left over to think through things that really need our attention.

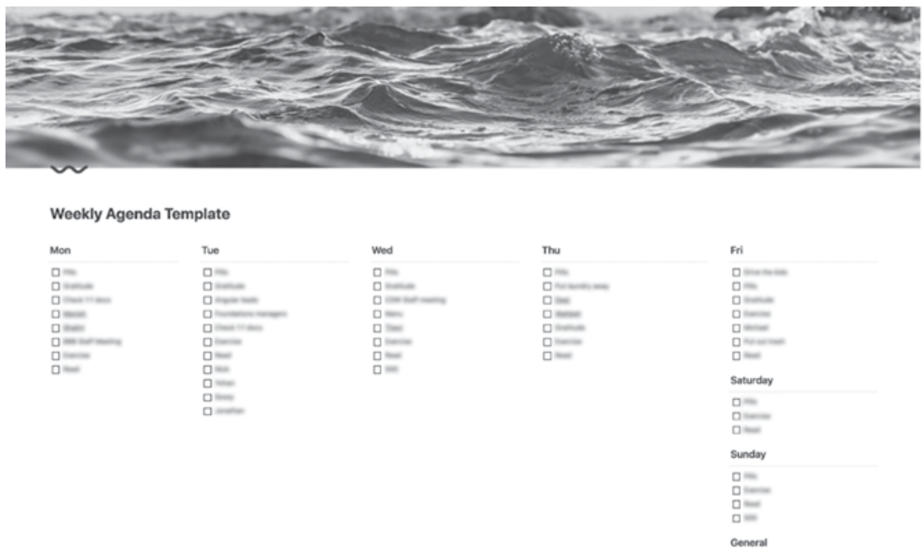

Figure 11: Custom weekly agenda template built in Notion.

GIVE YOURSELF CREDIT FOR TIME

My planner doesn't just include tasks. It also includes the names of people or meetings. I put these on my planner and check them off as I have the meeting.

It can be frustrating as a manager that your time is no longer spent making things in a hands-on way any longer, and your work exists less in tasks than in these meetings where you're coordinating tasks and driving alignment across groups or in 1:1s.

I write all the meetings into my planner each day. That way, when I finish those meetings, I also recognize that the time I spent aligning with others was well spent. This can help me from feeling like "All I am getting done is meetings! I'm not getting my tasks done!" The meetings ARE my job as a manager. I give myself credit for that work so that I don't prioritize task completion over the hard, human work.

KNOW YOURSELF

Most of these systems work for me because I know my strengths and weaknesses, and I play to them: I work more efficiently when things are broken down into steps. I work best when my work has the most meaning, and I remove the cruft. I work best when I work along with, not against, my energy levels, where possible.

I know what motivates me too. I customize the banner and icon for the weekly calendar so that I can tell the difference week to week, and it doesn't become a slog. It can help me feel just a small jolt of joy to see something I like as I traverse the list. I joke that I'm a bit like the character Leslie Knope from the TV series *Parks and Recreation*, when she says one of her hobbies is *"Jammin on my planner."* This type of output is so different from coding! But if you set it up in a systematic way, you can use similar thoughtfulness as we do in coding processes.

Some people, like programmer Alice Goldfuss, have this down to a science:

> *"I have a kanban board for my personal life to-do list and a cron job that empties my Done column and texts it to me every week, so I feel accomplished."*
>
> —@alicegoldfuss, May 25, 2019

Find a system that works for you.

20. https://en.wikipedia.org/wiki/Dopamine

CHAPTER 22

SETTING BOUNDARIES

The nature of our work means that we are constantly supporting others, and it can be easy to overextend and sacrifice ourselves for other people. If you're the kind of person who typically puts others before themselves, it can be hard to create healthy boundaries, as it can feel selfish. But setting boundaries for ourselves helps us support others.

If we don't set these boundaries, we don't live our values, and ultimately can't properly support our teams. Saying no feels strange and uncomfortable at first, but is an act of courage on behalf of your team, in an indirect way. This is a hard job. Take care of yourself, it's the best way to take care of your team.

SAYING NO

Overcommitting yourself and then feeling the pain of having too many responsibilities at once. It's tough because sometimes everything creeps up on you. You say yes to one thing at a time and, before you know it, you're completely overwhelmed. Or perhaps you said yes to one thing in particular that's a ton of work, but you're getting absolutely nothing out of it.

In setting boundaries, it can be tough to come to the decision to say no. Remember that you don't have to answer someone right away. You can ask them for some time to reflect and decide. I typically time-box an answer so that I don't forget or procrastinate too long on it, as that can produce its own kind of stress as well. Saying no when someone initially asks, not down the line, is important. It may feel bad or strange at the time to say no, but consider this: If you say no at the get-go, you're incurring less damage down the line. The person can make other plans, you don't feel haggard and overworked on something you're not aligned with. Your family and friends will thank you because you'll have more spoons to be there for them as well.

If you find yourself in a situation where you did say yes, and you probably should have said no (unless it's a family or other type of emergency), use the unpleasant feeling that you feel when you're overcommitted to help guide you in the future. "Well, we don't want to feel this way again, now do we?" Then suck it up and get that work done. And make better choices for yourself in the future.

TOO MANY MEETINGS

What do you do about meetings you don't find valuable? You will have to cancel meetings sometimes, and reschedule others at an infrequent cadence.

You can avoid being in too many meetings by delegating better and defining better boundaries, especially in cases when people want a lot of you. You have to be able to set expectations and needs. For example, "Let's optimize our time by meeting once every two weeks. We can catch up via email or chat if things come up in the interim."

If this feels very uncomfortable, especially as your responsibility grows, check in. Are you trusting the managers who work for you to run with their work? Are you meeting with them and their teams a lot because of issues of trust or issues of alignment? Try to work on the base premise here instead of using meetings as a Band-Aid.

Wherever necessary, remember to align people with outcomes. Let them know what you're expecting of their teams, and decide together on when a meeting is possible. If they know what outcomes you're looking for, and you both align on what's possible, then the team can be trusted to drive those outcomes as they see fit, and you don't need to be close to the metal on how they execute every step. This will become a more and more crucial skill to learn the higher you go in management.

Boundaries of Boundaries

It's worth mentioning there that you can go the other way as well. For a new manager, it can be tough to understand that your job has changed and, though you may need some focus time, focus is not your primary function anymore. Meetings are now your job. Coordination is your job. Your job is to be interruption-driven so your team can stay focus-driven. This means you shouldn't be protecting yourself *too much* from meetings.

I can't tell you how many times I've heard ICs (independent contributors) say that they feel their manager is absent and doesn't care about the team, and concurrently hear the manager brag that they make sure to protect their calendar so that they can focus. If your team feels out to sea because you protect your time, you're likely not doing your job well.

Make sure you don't use boundaries as a way to protect yourself from everything that feels uncomfortable to you. Sometimes growth is uncomfortable, and you may be using boundaries as an excuse to not push yourself. Some self-reflection may be required here.

SELF BOUNDARIES: MENTAL

To be a resilient manager, you have to **forgive yourself.** You'll be more productive on some days (or hell, even years!) than you will be on others. That's okay. Life is full of ebbs and flows.

It's helpful to check in with yourself often: Is your waning productivity the result of simply moving a little slower? Might it be a sign of depression?

Does it represent misalignment with personal goals (or the company where you work), or are you reacting to a toxic environment? If you're just moving a little slower because of personal, natural cycles, patience with yourself is key. But if the latter might be true, please seek out support, be it therapy, coaching, friendship, or making changes that positively affect your health and well-being.

In all cases, forgiveness is in order. In some cases, you may need more than a prioritization plan. But truly, you'll never be able to do *everything*, because no one is capable of that.

People you see on social media might only be showing their successes—the higher their follower count, the more harassment they get—and it might not be safe for their own mental health to share everything. **Please don't compare yourself to what you see from other people on the outside.**

The fact is that we all get distraught, and sometimes mental illness isn't something we want to announce over the megaphone of social media or even to the people within our own companies and social groups. I have a personal rule that I try to extend to others where helpful: No compare and despair.

You are on your own journey. **Make decisions that are right for you; trust yourself to know what you need.**

SELF BOUNDARIES: PHYSICAL

Skipping vacations only causes harm for you *and* your team. You may try to see how long you can go without a vacation if you're in a place that does payouts, or you may have a culture that idolizes those who soldier through work. Whatever the case, I waiting too long to have a vacation, or skipping vacations entirely, doesn't actually allow you to perform at your best. You need rest.

It also sends signals to your team, like it or not, that it's improper for them to take the time they need. It's hard for them to know for sure that they're okay taking time off unless they see you do it first.

BOUNDARIES IN PRACTICE

One strange thing about boundaries is that when someone states a boundary, many times the first thing another party wants to do is test it. I have no idea why this is, but it happens. So in practice, setting boundaries may just be the first step.

My coach Jessi says "It's only a boundary if the other person hears it." Once the boundary is set, you have to work to maintain it.

Try not to test boundaries as a manager. Be particularly thoughtful about this and reflect on it whenever you can.

If you state a boundary and someone else still crosses that boundary, it can feel like a real violation of trust. People don't always realize when they're crossing boundaries, even though it may feel obvious to you. You can reinforce what you said before. I have noticed that stating boundaries in a firmer way the second time usually works. It seems to be a bit of a cultural phenomenon in the United States that drawing boundaries feels like too strong an act for some people. When I lived in Greece, it was more "normal" for people to express boundaries, and people were generally accepting of other folks being clear.

To unpack about culture and values is a bit out of the scope of this book, but I highly recommend *The Culture Map* by Erin Meyer, which dives into this subject more thoroughly.

WRAPPING UP

Boundary setting is a nuanced subject; it can relate to day-to-day priorities, the way we communicate with others, what we are willing to tolerate at an individual level, and even what we're willing to tolerate at the company level. Don’t forget your values work here. Living your values takes constant adjustment—in what you let in, and also what you step away from.

It’s worth noting that you never need to be rude when you define a boundary. There is nothing wrong with setting boundaries. Beyond your working hours, you don’t owe pieces of yourself to your job or your community. You will ultimately be able to do your best work in either circumstance if you know where your own line is.

CHAPTER 23

FILLING YOUR CUP

I was once in a session with my coach Jessi, and I was lamenting some internal politics. “I can’t believe he said that!” I complained. “It was so unhelpful to the entire leadership team.”

“That sounds awful,” she said. “I’m sorry you’re dealing with that. So . . . ,” she probed, “what are you doing to become resilient to this?”

The question took me off guard. A person was being disrespectful. Why was the action on me?

For a very good reason, it turns out.

Leadership is fraught with hardships, miscommunication, and tough work around alignment, priorities, and incentives. Try as I might, I was never going to be able to iron out every wrinkle. I had to learn how to live with imperfection, and I had to learn how to take care of myself while I did it.

In a role where so much of the job revolves around supporting others and tying things together, it’s pertinent that we don’t come undone. This does not mean denying adversity or a bad situation. Note that in this example, Jessi didn’t downplay the negative interaction or pretend it was actually a positive one. We acknowledged it, and then figured out how to move on. It’s important not to deny yourself feelings either. There’s room for the full gamut of human emotion. But we also have to figure out what we’ll do after we’re done processing.

Jessi was giving me tools for longevity—with some understanding that for me to do the best job for others, I had to take care of myself. It's actually an easy thing to lose sight of, and eventually with a stressful enough job, it takes work in and of itself.

SELF-CARE

Self-care has become something of a buzzword in the last few years, so much that it may have lost some of its meaning. Yet it's still an important principle. When Jessi asks me about resilience, she often asks how I'm taking care of myself.

It may seem like all self-care acts have to be "good for you," and certainly being good to your body can absolutely help your mind and spirit. But too much "have to" and not enough "want to" in your life is a bit dangerous. I suggest both in moderation. And perhaps even relieving yourself of the judgment that comes from defining things as "good" or "bad."

Self-care doesn't have to be about accomplishing big goals either. I have a New Year's resolution to sit in my favorite comfortable chair whenever I'm doing focus work between meetings this year. I get exhausted just sitting at the same desk all day. This small bit of scenery change has made me look forward to heads-down time each day.

Finding Joy to Become More Resilient

It's important to find joy in small moments. They needn't be an extravagant vacation or a decadent cake (though those can be amazing too!). They can be as simple as stretching in between meetings, or spending a few moments gazing out the window and looking at the light, and even as silly as making up dumb jokes to share. Every moment is an opportunity for rest and joy.

In Being Peace, author Thich Nhat Hanh writes:

> *"I sit here, and I know where I am. This is very important. We tend to be alive in the future, not now. We say, 'Wait until I finish school and get my PhD degree, and*

then I will be really alive.' And then after that, a job, then a car. After the car, then a house. We tend to postpone being alive in the present moment. . . . Therefore, the technique is to be in the present moment, to be aware that we are here and now, and the only moment to be alive is the present moment. This is a wonderful moment."

The more you cultivate moments of peace, the more resilient you become to inevitable moments of strife or misalignment. If we feel chaotic internally, it's hard for us to do our jobs and also take care of our inner well-being.

GRATITUDE JOURNALING

My coach Jessi tried for years to get me to keep a gratitude journal. I thought it sounded kind of New Age-y and not my style. I always thanked her for the idea, but never went through with it. Then my husband's cancer relapsed. I felt like I was swimming and slowly being sucked under the water's surface. Nothing made sense, and there was nothing to hold on to. I was fairly desperate to try anything that might help. I tried her journaling idea.

I was shocked at how much better I felt. As I mentioned earlier, your brain has a negative bias, and it takes some work to retrain ourselves to see all the good around us. Neuroscientists have found that a practice of gratitude can actually alter your brain chemistry.[21]

In *The Happiness Advantage*, author Shawn Achor discusses the Tetris Effect. That is, if you start thinking about something a lot, you begin to notice it everywhere. And applying that to a gratitude practice can allow you to see what you might be grateful for that you could otherwise miss in the day-to-day. Achor writes:

"When our brains get stuck in a pattern that focuses on stress, negativity, and failure, we set ourselves up to fail. The Tetris Effect teaches us how to retrain our

brains to spot patterns of possibility, so we can see—and seize—opportunity wherever we look."

My journal is not sophisticated. It just includes the date and a quick list of what I feel grateful for. When I first began, it was a little difficult. The relapse was heavy on my mind, the pandemic had just begun, and work was especially tough. I didn't feel "in the mood" to engage in some false positivity. What I found was that what I had to be grateful for wasn't false at all. We had a lovely house we could afford, we have great kids and cute dogs, the flowers out back had started to bloom. I have been broke in my life, and that wasn't the case anymore. I could buy a new toothbrush if I needed it, shoes if they got a hole in them. And then I was grateful for small things: I liked how the cement seemed to glimmer after the rain. The little giggles of the kids when they thought they tricked us into thinking they're asleep. Moments when my team supported each other without performance, just because they cared for one another. Coding on a side project late into the night on a weekend.

I started noticing small, wonderful things around me preemptively, without forcing myself to. My outlook on life and work improved. So did (as Jessi predicted) my resilience.

Was I a better manager because of this practice? *Of course I was.*

When my husband relapsed, I was at my worst. I had very little control over my emotions, moment to moment. I was overly sensitive. I cared too much and didn't care enough all at once. Jessi's trick didn't save me. Humans aren't Booleans, and the situation didn't magically improve with this practice. But it did give me some tools to get through this trying time. Tools I desperately needed.

You don't have to be perfect about writing, I personally add it to my to-do list every other day. If I miss one, it's fine, I'll just pick it up again next time. I do find it helps my overall outlook if I keep up a steady pace. It might do the same for you too.

Your Own Favorites

Your gratitude journal might also expose some of the things that you need in order to recharge. You could also figure these things out without a journal—it's not a necessity. But it is important to figure out the things that help you get through.

For me, I can't function well if I'm not eating well and drinking enough water. Exercise is, regrettably, also important. I just have way more energy and feel happier if I exercise, even though I don't always love it. I used to hate exercise, but I found that by switching from listening to music to listening to audiobooks and making sure to be outside when I could, I found a whole new joy in it.

I have noticed that, for days at work filled with tough conversations, I have more tools and better mental clarity if I exercise for at least forty minutes. This is tough! Those days are typically the days when I would love to have some time back. But time and again, I've seen I am less anxious if I carve out this time for myself, so I do.

I also love wine, and wine on the porch can rejuvenate me after a long day, along with a long chat with my husband or a friend. Baths soothe me, and coding fires me up. I use either in tandem to give myself what I need to get through.

My tools won't be yours, and that's okay! Find your joy, and cultivate it.

21. It's incredible that something so small can alter your perspective so deeply: https://greatergood.berkeley.edu/article/item/how_gratitude_changes_you_and_your_brain

CHAPTER 24

BELIEVING IN YOURSELF

In grad school, I worked on a project for Harvard and the Cartoon Art Museum where we taught undereducated children how to read and write through making autobiographical comic books. I taught the same children year after year (fifth, sixth, and seventh graders), and I noticed something of a trend. The fifth and sixth graders could all draw. They were wonderful, with creative stories and imaginative characters. But the seventh graders insisted they couldn't draw anymore. They always pointed to one individual in the group and said, "Ask Alex. He's the artist."

Middle school is a rough time. Something happened between sixth and seventh grade that made the children feel more defensive and shy, and also bind their own identities to one trait. "She's the sports girl." "He's the book kid." Somehow, they start to decide they are either good or bad at something like it's a Boolean. Ask any five-year-old if they can draw, and they'll look at you like you have three eyes. Of course they can draw. Ask any adult the same question and you're likely to get a very different result.

But science doesn't support this "You're good or You're bad" concept.

You may have heard of *growth* and *fixed* mindset: people with a fixed mindset tend to believe that their skills are immutable, and they can only be so good at something. On the other hand, someone with a growth mindset understands that, with effort, we can change.

A 2007 study by Lisa Blackwell, et al.,[22] tracked 373 seventh graders, studying whether they had a growth or fixed mindset. Their progress was tracked over the next few years. In two years, the students with a growth mindset steadily raised their GPAs while those students with a fixed mindset stagnated.

You can cultivate a growth mindset by identifying what goals you want to accomplish and slowly work toward them. You can also remind yourself from time to time of the value of your work.

In fact, research shows that belief in your own ability can be a better predictor of future success than previous job performance. In his book *Self-Efficacy*, Albert Bandura discusses this. Some of the great takeaways from his research are as follows:

- Belief in your own competence and abilities is related to self-confidence, but is not the same. He calls this self-efficacy. Self-efficacy perceptions are "a key factor in a generative system of human competence."
- When we feel we have some agency in our lives, when we believe in our own competence to act on what's within our locus of control, we thrive.
- Self-efficacy not only helps us be successful in endeavors, it also helps us become resilient to obstacles and failures.

Believing in yourself isn't egocentric. Being a manager is very challenging, and you can't support everyone else without being supported as well.

SURROUNDING YOURSELF WITH PEOPLE WHO BELIEVE IN YOU

Your mindset also affects the performance of others. Believing in someone else's potential can impact their performance.

The *Invisibilia* podcast aired an episode titled "How to Become Batman."[23]

In this episode, scientists took similarly performing rats, but told human handlers, “This rat is slow” and then ran it through a race. The rat in question then underperformed. The rats that humans were told were stellar, ran faster. It turns out that the small ways that the humans were handling these rats changed based on these preconceived notions of them. And it actually affected the rats’ self-confidence!

Now let’s apply that to humans. How much does it matter if you’re surrounded by people who support you and value your work? That one friend of yours who is “negging” or doubting you, or making you feel uncomfortable, what does that cost you? Maybe a lot!

In the field of psychology, the Pygmalion effect[24] is the name for the concept of how high expectations from those around us lead to better performance, and low expectations lead to the opposite, becoming a self-fulfilling prophecy.

Surrounding yourself with people who support you is very important. There are many methods of support. Some people support by loving protectively and fiercely. Some people support by reminding you of your worth. Some people support by being available to play Fortnite and drink whiskey. Seek out folks who act without jealousy, and **make sure you support them too.**

In the wise words of Leslie Knope from *Parks and Recreation*, “We need to remember what’s important in life: friends, waffles, and work. Or waffles, friends, and work? Doesn’t matter. But work is third.”

COACHING AND MENTORSHIP

Having supportive peers and/or groups you can count on to be your cheerleaders during rough times is so important. As a manager of others, you need support. You need to have a place you can go to sort things out for yourself that’s not with your direct reports. I have a coach (as I have mentioned many times in this book), as well as a cabal of other managers, some who are peers within my company and some who work elsewhere. All are very important to me.

Being a manager can be a little lonely in the default state; you don't get to process things the way others on your teams might. You have to evoke calm in critical conditions. But you are still human, and you can't pretend you don't have emotions, or leanings, or worry, or fear. Find a group of people you can lean on in the hard times, ones you can laugh with, and ones who will help you feel included and valued.

My coach is one of the smartest people I know. She gives me tools for handling situations that feel out of my control. She guides me when I'm headed down an ill-advised path. She listens when I just need to vent and helps me workshop and navigate difficult conversations. By the way, she's taking on new clients, and I can't recommend her highly enough: jessikovalik.com[25]

If you are looking for other resources, services like BetterHelp[26] allow for remote therapy and coaching. Though it's not important that you use that service, I suggest finding someone who provides a safe space and tools for you. You can potentially lean on a partner or best friend for this, but I recommend you don't, because a coach is formally trained to guide you through tough waters.

The manager cabal was created by a friend, Simona Cotin. Author and trainer Lara Hogan suggested she create it during one of her workshops. Like the coach, we all listen to one another and help give advice, but this is a different kind of safe space. Where coaching gives me tools, this cabal allows us all to have a place of inclusion. We can laugh and express remorse, but more than anything, it's a place to be ourselves and not have everything be so serious. We can give actionable advice and compare notes because it's not an academic study of management; we're all going through it. This peer group is meaningful because we can't really tell how everyone else is managing; it can appear from the outside like everyone has it under control, except you. We take turns processing events, listening, and helping. And yes, we vent too.

I meet with both the coach and the manager cabal once every two weeks. This doesn't need to be your cadence; figure out what's right for you. I suggest keeping the manager cabal to eight people or less, generally with a sweet spot around six. More than that and it becomes a bit harder for everyone to share, and possibly more performative than truly helpful. You want a group that's small enough that no one is waiting forever to speak, so the conversation flows nicely, and the shyer voices aren't left out.

Being a manager is tough. Your mistakes impact people, and that feeling of pressure can be a little isolating. I've made all of the mistakes discussed in this book, and more. I feel it's critical to share so that when we encounter pitfalls, we don't feel so alone. We have a potential path forward.

GIVING BACK

Sorry, Ayn Rand. Research confirms[27] that people who give are more likely to feel happiness over a long period of time.

> *"The effects of altruistic spending are probably akin to those of exercise, which can have immediate and long-term effects."*
>
> —Dr. Elizabeth Dunn, professor at University of British Columbia

This research is compelling, but I found the continued journey of this research even more fascinating. Dr. Elizabeth Dunn, author and professor at University of British Columbia, gave a TED Talk, "Helping others makes us happier,"[28] about how giving in a connected state is more important than just throwing money at a problem. In this talk, she covers how seeing the impacts of giving and feeling connected to the outcome give the greatest sense of joy. I heartily recommend watching Dunn's fifteen-minute TED Talk. She's a great speaker.

Giving back can take many forms. Sometimes in our positions, we're given a lot of opportunity or access. A great way to support someone else in their career, even if they aren't on our teams directly, is to open these doors. This could mean passing along a speaking opportunity, adding someone to a meeting that might give them a seat at a table in an important discussion, and/or introducing them to someone who might help them get a better job. So many options are possible.

Another way to give back is to provide mentorship for others. A note on mentorship outside of leadership: Never force mentorship on someone. If someone asks you for mentorship, you can decide to do so, but being guided by someone is a very personal thing. In a mentorship or a sponsorship, always ask permission first before you do something on behalf of someone else.

Mentorship and sponsorship are different: mentorship is giving someone advice or consultation, while sponsorship is offering someone an opportunity or connecting them with people who may offer an opportunity. I do some mentorship, but sponsorship can better in a lot of ways because it's not really about you; it's about providing people access for themselves.

Finally, if you're in a position of engineering leadership, you may have a bit of capital to burn, so donating to some great causes can also be a good way to give back. Here are some of my favorites:

- **Dev Careers** (devcareer.io): Helps African individuals who are trying to get a foothold in tech, both with education and materials.
- **She Code Africa** (shecodeafrica.org): Helps sponsor events and initiatives to empower African women in technology via a donation or more.
- **Black Girls Code** (blackgirlscode.com): Increases the number of women of color in the digital space by empowering girls of color

ages seven to seventeen to become innovators in STEM fields, leaders in their communities, and builders of their own futures.

There are many ways of giving back. Consider this a powerful force—for you and for the world around you.

WRAPPING UP

Management can be selfless, and much is asked from us—our time, our care, and a ton of very difficult emotional regulation. Taking care of ourselves is intrinsically connected to whether or not we can do our best work.

If you're the kind of person who always puts others first, don't forget to put the oxygen mask on before helping others. Take advantage of the many ways to take care of yourself; it's very important that we all do so.

CONCLUSION

You made it! You read a whole book even though you also have to lead one or many engineering teams. That in and of itself is a huge accomplishment.

As I've stressed throughout the book, management can be a tough job. Your words matter. You set direction. You can help organizations feel supported and healthy. Ongoing learning is a critical part of supporting our teams. That's a lot to carry on our shoulders. But we have the possibility **to make positive change in the groups that we lead**. This is a responsibility that's sobering and compelling.

Humans are nondeterministic; what works in one situation might not work in another. By sharing what we know, we gather tools that may serve us when paths seem unclear. Using these tools, we can make better technical decisions, create better products, and create processes that benefit our staff.

The more we open ourselves to learning and developing, the more we explore our potential to make our industry better. By learning and developing as managers, we can impact projects on a technical level and also support the crucial backbone of any industry: the people.

22. https://news.stanford.edu/news/2007/february7/dweck-020707.html
23. This episode is particularly great, but I highly recommend all of the Invisibilia podcasts: https://www.npr.org/programs/invisibilia/378577902/how-to-become-batman
24. https://en.wikipedia.org/wiki/Pygmalion_effect
25. https://jessikovalik.com/
26. https://www.betterhelp.com/
27. https://www.science.org/content/article/secret-happiness-giving
28. https://www.youtube.com/watch?v=lUKhMUZnLuw

ACKNOWLEDGMENTS

My true thanks to Christian Nwamba, Val Head, Marvin Carlisle, Angie Jones, Lauren Sell, Emily Freeman, Ashley Willis, Dalia Havens, Jen Klimas, Stu Steene-Connolly, Simona Cotin, Shweta Saraf, Phil Hawksworth, Brian Stoler, Chris Bach, Claire Jackel, Kristi Mill, Shalini Verma, Wahbeh Qardaji, Matt Biilmann, and Jason Lengstorf for their help, guidance, and support over the years. Every one of these people has at one time or another listened to a very strange idea I've had and said, "Okay, let's do it."

I thank the kiddos: Megan, Matthew, Emily Smith, our stone-cold pack of weirdos. You are all so easy to love. You make me so proud every day.

I thank my coach, Jessi Kovalik, who figures prominently in this book because she's truly one of the smartest folks I've ever met, and I would be nowhere without her.

Thank you so much to my editors, Christina Frey and Nicole Taché, who took this book from a pile of incomprehensible disparate concepts into, well, a book! Thank you to Joel Hooks and the entire Egghead crew. Your tireless work to support this effort, especially when I had hard life things going on, is a debt I can never repay. Thank you.

Thank you to Srdjan Vidakovic of Widakk Design for the gorgeous cover art. It inspired me to do the illustrations in the book because I loved it so much.

Thank you to dinosandcomics for letting me use one of your wonderful cartoons in this book. Your comics are so touching and poignant. Thank you.

I also thank Kenny Eze, Tara Manicsic, Ben Hong, Cassidy Williams, Rachael Stravansky, Kristen Lavavej, Jacklyn Carroll, Nick Behrens, Sossy

Mansourian, Madleina Scheidegger, Yohan LeNerriec, Lacey Horton, and Whitney Stafford for everything. Thank you to Manu Murthy and Deej Ajisebutu, who always have deep insights, great feedback, and gave me support when I needed it most. Thanks to Kent C. Dodds, who worked on a long-time original iteration of the open source chapter with me and gave permission to refactor it for this book, and also for being a sweet guy in general. You have all been fantastic to work with, and I really valued the time we spent working together.

I thank my family for all the support over the years, for teaching me to process things with humor, for teaching me hard work can pay off at times, and for showing me that discipline gets things across the line. I also thank Dizzy's family for being so welcoming to me.

But mainly I thank Dizzy whose tireless support means more to me than I can express in words. You're the best human, and I'm not even biased at all.

FURTHER RESOURCES

Engineering Management for the Rest of Us is just one resource for anyone looking to grow as an engineering manager. I highly suggest learning as much as possible in this area. Not all advice will apply to every situation, but it's valuable to have different perspectives.

Achor, Shawn. *The Happiness Advantage: The Seven Principles of Positive Psychology That Fuel Success and Performance at Work*. Currency, 2010.

Banaji, Mahzarin R., and Anthony G. Greenwald. *Blind Spot: Hidden Biases of Good People.* Delacorte Press, 2013.

Bandura, Albert. *Self-Efficacy: The Exercise of Control*. Freeman, 1997.

Blackwell, Lisa S, Dali H. Trzesniewski and Carol Sorich Dweck. "Implicit Theories of Intelligence Predict Achievement Across an Adolescent Transition: A Longitudinal Study and Interventions." Society for Research and Child Development, February 28, 2007.

Blue Beyond Consulting. "This is your Brain on Feedback: How Understanding a Little Brain Science Can Make a Big Difference in Your Next Feedback Conversation." 2022. https://www.bluebeyondconsulting.com/2018/11/this-is-your-brain-on-feedback-how-understanding-a-little-brain-science-can-make-a-big-difference-in-your-next-feedback-conversation/

Brown, Jennifer. *How to be an Inclusive Leader: Your Role in Creating Cultures of Belonging Where Everyone Can Thrive*. Berrett-Koehler Publishers, 2019.

Clear, James. *Atomic Habits: An Easy & Proven Way to Build Good Habits & Break Bad Ones.* Avery, 2018.

Conti, Gregory. "A Brief Guide to Better 1:1s—For Makers and Managers Alike." *Help Scout*, July 4, 2016. https://www.helpscout.com/blog/one-on-

ones/
Csíkszentmihályi, Mihály. *Flow: The Psychology of Optimal Experience*. Harper Perennial, 2008.
Doerr, John. *Measure What Matters: OKRs: The Simple Idea that Drives 10x Growth*. Portfolio, 2018.
Dunn, PhD, Elizabeth. TED Talk, "Helping others makes us happier:but it matters how we do it." 2019. https://www.ted.com/talks/elizabeth_dunn_helping_others_makes_us_happier_but_it_matters_how_we_do_it?language=en
Epstein, David. *Range: Why Generalists Triumph in a Specialized World*. Riverhead Books, 2019.
Eyal, Nir. *Hooked: How to Build Habit-forming Products*. Portfolio, 2014.
Festiger, Schachter, and Back. *Social Pressures in Informal Groups: A Study of Human Factors in Housing.* Stanford Press, 1963.
Forsgren, PhD, Nicole, Jez Humble, and Gene Kim. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution Press, 2018.
Fournier, Camille. *The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change*. O'Reilly Media, 2017.
Hanh, Thich Nhat. *Being Peace.* Parallax Press, 2005.
Hogan, Lara. *Resilient Management.* A Book Apart, 2019.
Hogan, Lara. "The Complete Demystifying Management Program." Wherewithall. See: https://courses.wherewithall.com
Invisibilia. "How to Become Batman." NPR podcast. January 23, 2014. https://www.npr.org/programs/invisibilia/378577902/how-to-become-batman
Jana, Tiffany, and Michael Baran. *Subtle Acts of Exclusion: How to Understand, Identify, and Stop Microaggressions.* Berrett-Koehler Publishers, 2020.
Kaplan, Mark, and Mason Donovan. *The Inclusion Dividend: Why Investing in Diversity & Inclusion Pays Off.* Bibliomotion, 2013.
Kovalik, MA, LPC, Jessi. Counseling, Coaching, and Consultation: https://jessikovalik.com/

Lencioni, Patrick, M. *The Advantage: Why Organizational Health Trumps Everything Else in Business*. Jossey-Bass, 2012.

Levav, Jonathan, Nicholas Reinholtz, and Claire Lin. “The Effect of Ordering Decisions by Choice-Set Size on Consumer Search” *Journal of Consumer Research*, Vol 39:3, October 1, 2012. https://academic.oup.com/jcr/article-abstract/39/3/585/1822634?redirectedFrom=fulltext

Meyer, Erin. *The Culture Map: Breaking Through the Invisible Barriers of Global Business*. PublicAffairs, 2014.

Pink, Daniel. *Drive: The Surprising Truth About What Motivates Us*. Riverhead Books, 2011.

Rock, David, Heidi Grant, and Jacqui Grey. “Diverse Teams Feel Less Comfortable—and That’s Why They Perform Better.” *Harvard Review*, September 22, 2016. https://hbr.org/2016/09/diverse-teams-feel-less-comfortable-and-thats-why-they-perform-better

Rosenberg, PhD, Marshall B. *Nonviolent Communication*: *A Language of Life: Life-Changing Tools for Healthy Relationships.* PuddleDancer Press, 2015.

Scott, Kim. *Radical Candor: How to Get What You Want By Saying What You Mean*. St. Martin’s Press, 2017.

Seldman, PhD, Gwendolyn. “Why Do We Like People Who Are Similar to Us?” *Psychology Today.* December 18, 2018. https://www.psychologytoday.com/us/blog/close-encounters/201812/why-do-we-people-who-are-similar-us

Sexton, Alex. “The Productivity Cycle.” Blog. January 15, 2014. https://alexsexton.com/blog/2014/1/the-productivity-cycle/

Sivers, Derek. *Hell Yeah or No: What’s Worth Doing.* Hit Media, 2020.

Steele, Claude M. *Whistling Vivaldi: How Stereotypes Affect Us and What We Can Do*. W.W. Norton & Company, 2011.

Stone, Douglas, and Sheila Heen. *Thanks for the Feedback: The Science and Art of Receiving Feedback Well*. Viking, 2014.

Uhlmann, Eric Luis, and Geoffrey L. Cohen. “Constructed Criteria: Redefining Merit to Justify Discrimination.” *Psychological Science*, Yale

University. Revised 6.22.04. https://ed.stanford.edu/sites/default/files/uhlmann_et_2005.pdf

Williams, Bärí A. *Diversity in the Workplace: Eye-Opening Interviews to Jumpstart Conversations about Identity, Privilege, and Bias*. Rockridge Press, 2020.

Zhuo, Julie. *The Making of a Manager: What to Do When Everyone Looks to You*. Portfolio, 2019.

ABOUT THE AUTHOR

Sarah Drasner is Director of Engineering, Core Developer Web at Google, where she runs the web infrastructure teams that power Google's web apps. These teams include the JavaScript and TypeScript languages at Google, multiple frameworks including Angular, ACX, and Wiz, Build/Serve, Web Testing, including Karma, and CSS/Sass, and the Platform Boq Web.

Sarah an award-winning speaker, and former VP of Developer Experience at Netlify and Principal Lead for Microsoft Azure. Sarah is a Vue core team member emeritus and an O'Reilly author, and has given Frontend Masters workshops. Sarah is also co-organizer of ConcatenateConf, a free conference for Nigerian and Kenyan developers.

Sarah likes finding all the ways to die in Fortnite, discovering new cheeses, and hanging out with her talented, adorable, and mischievous family.