HOW TO BE A TECH LEAD

A practical guide for new and experienced **tech leads** to increase their **confidence**, **effectiveness**, and **impact**

Michael Rice

founder of the
**Tech Lead
Coaching
Network**

How To Be A Tech Lead

A practical guide for new and experienced tech leads to increase their confidence, effectiveness, and impact

Michael Rice

This book is for sale at
http://leanpub.com/how-to-be-a-tech-lead

This version was published on 2019-04-06

Contents

About the Author

Michael Rice started writing software when he was fourteen years old on the first generation IBM PC using BASIC. He later got the opportunity to keep playing with computers with the first generation of Apple Macintosh and many more over the years.

He started writing code for a living in the mid-1990s with small startups and consulting firms and continued his career at companies like Intel, Accenture, and Red Hat. He has been involved in hundreds of projects at dozens of Fortune 500 and governmental clients.

He discovered the critical importance of the tech lead role as a field engineering manager at Red Hat where he was responsible for deploying and managing many field engineering teams on critical projects. There, he noticed the clear correlation that the most successful teams had effective tech leads leading them.

He started thinking about how to consistently mentor and coach new tech leads and, eventually, created the Tech Lead Coaching Network as a community driven, volunteer-based organization for tech leads to support each other.

He lives in the Los Angeles area and still writes code for a living and steps into as many leadership moments as he can, which you will learn about in these pages.

He can be reached at coaching@michaelrice.com.

Introduction

This is a short book with two simple objectives.

First, I want to inspire you to make the most of role. The tech lead role, which has a lot of different definitions and demands, is a huge step up in your career. As an industry, we do not talk enough about what a big transition it is for new tech leads.

Second, I want to share some practical skills, capabilities, and approaches you should consider developing so you will be effective in the role and confident that you are doing the right things the right way.

It is written primarily for software engineers because that is my background and experience, but the role and what I teach about it could be applied to other technical fields as well, such as systems engineering, hardware engineering, data science, QA, and so on.

Why I Wrote This Book

I was a mediocre tech lead.

I wish I could look back on my twenty or so years of experience and be able to tell you stories of challenges I overcame, teams I helped build, and the success and growth we shared. But, up until the past five years or so, I really can’t.

Looking back, the things I'm probably most proud of are the technical decisions and the products I've been part of over the years. I think the choices and directions I set were based on solid analysis and they were clever, even creative choices at the time. Unfortunately, I bet almost all that code I built, either individually or as a tech lead, is long gone.

In one of my earliest memories of being a tech lead, my managers tapped me to lead the replacement a complicated Excel-driven process with an architecture that I thought was pretty cool. It would be a multi-tiered, distributed architecture splitting the user interface from the business and data layers. N-tiered architectures were pretty hot back in the late 1990s.

I had a small team of developers: two offshore and one developer in the office with me. It took us about three months. Things got done. We launched ontime. The system did what it was supposed to do. But if I were my own manager back then, I probably would have given my performance a grade of B. Maybe C for just average.

The technical parts were easy. Humans were the hard part. Questions and challenges swirled like:

- How do I assign work?
- How do I tell people what to do in a way that doesn't make them mad?
- What do I do if they do get mad?
- How should I handle it when someone builds something in a way I didn't agree with?
- What should I do when a programmer's work was taking too long or didn't do what it was supposed to?
- What was I supposed to say to my management when they started asking for more than we could do?

- How was I supposed to keep the project managers (this was pre-agile) from driving my team and me crazy?
- How was I supposed to deal with the discomfort of all these people looking to me for answers.

I remember not sleeping very well, and I mostly got through it with a sense of relief. Having not been able to point to my leadership accomplishments or personal growth, for a long time my resume highlighted technical accomplishments, and I took credit from them as “the lead.” I have seen a lot of resumes like that since then. I wonder if they merely survived the experience of being a tech lead like I did.

I’m sure they retired that code we wrote long time ago by now, so now the most I can say about the experience is that it’s useful for remembering how mediocre the experience was.

Lots of time went by after this, and I got better at thing–mostly from trial and error, not the focus on progress I want you to have. Even more years passed, I got the opportunity to observe other, far more successful, tech leads at work. My job as a manager of consultants for client projects at Red Hat basically depended on having solid tech leads on the team–raw technical skill alone, and I saw a lot of that, was clearly insufficient.

And I began to learn how to cultivate and spot the talent. It was definitely a haphazard journey where I had more failures than success in the early days, so I’m very grateful to my former employer for basically teaching me how this works.

In this short book, I’ll give back my insights to you about what I know works. The tech lead role is a pretty varied one, and the personalities that end up in the role are even more varied, so some of the skills and comments will be more useful to you than others.

I’m excited to be on this journey with you. I really do hope you will reach out to me and tell me how it’s going.

Michael Rice

Hermosa Beach, California

We’re on a Journey, Tech Leads!

Welcome to a new journey, tech lead.

Being a tech lead, especially for the first time, can be hard. There are new skills and capabilities that you need and don’t yet have. But this is a huge opportunity to grow professionally and personally. In this book, I’m going to share with you what I know works to make it easier and get the most out of the experience, and it isn’t hard for most to do. Even you.

A New Chapter in Your Career and Life

Of course I can’t know how it is you came to be a tech lead or how it is you decided to open up this book. Maybe your managers saw something in you and tapped you to lead a project. Maybe they want you to lead a small initiative for your team or your organization. Maybe you are not in the role yet quite yet. Maybe you are still considering making the jump. Maybe you are trying to figure what tech leads actually do or are supposed to do.

No matter how you got here, in this chapter I want to tell you how excited I am that you are at this point in your career and how grateful I am to be here the journey with you.

Toward the end of this chapter, I will introduce you to the rest of the book, which is focused on the practical things you need to do in the role and the skills you should consider developing.

In many ways, stepping into your first tech lead role is the biggest step you will take in your career. This is the first time in your career, maybe in your life, that you are rising up in the organization, from one whose contributions come solely from your individual work to one who get multiplied results through others.

Getting Results Through Others

Making the step up to tech lead may seem like a minor step. After all, most tech leads are still deeply technical and remain primarily individual contributors. Especially as a new tech lead, it is likely that you are still going to be writing code as much as ninety percent of your time, depending on the day and specifics of the role. Most tech leads are still, principally, individual contributors and the role is still a deeply technical role.

You may only be leading ten percent of the time, so why should you bother to read this book, and why am I so excited for you? Because, possibly for the first time in your career or life, as a tech lead, you are now responsible for getting results through other people, and this is a very big change in who you are as an individual.

Odds are in the role because you were a successful individual contributor. So up until now, success in your career has been been largely tied to the results you were able to individually produce. Your individual success is why your management considered putting you in a lead role or why you are considering stepping into it on your own.

As they say, and you will come to appreciate when you are in the tech lead role for a while, "What got you here won't keep you here."

As a tech lead you are going to learn how to get results **through** the work of others. You are still going to be involved in all the technical work that you probably still enjoy, but now the experiences and the results will not be solely your own. The results, the learning, and the experiences are going to come from what you have done through and with your team.

If you have been around effective managers or leaders in your organization or watched technical leaders who inspire you, maybe like Steve Jobs or even more prosaic examples like David Heinemeier Hansson, what they do day to day may seem a little confusing or hazy to you. You see their results, however. Through their leadership, they are able to produce far more impact than they ever could have on their own.

Here, you and I stand together, possibly at the beginning of your journey. You have a long way to go to be the next Steve Jobs (and so do I). But here we are at the beginning, and I am so excited for you.

Should You Be a Tech Lead?

Even Steve Jobs had to take this critical first step a long time ago. He made a choice. He was human just like you are today, and years ago he faced choices like you do now. Sure he made the choice to be an entrepreneur, but doing so meant he also implicitly *had* to make the choice to also step into tech leadership. You are facing the same choice.

I was a small child, and some of you may not have born, when Steve Jobs was making his choices, but I am willing to bet his first days leading the first teams at Apple Computer, trying to get results through them were frustrating and confusing. You are going to be frustrated and confused at times too.

You are going to learn it is sometimes incredibly hard to get people to understand your vision and execute on it the way you want them to. On bad days you may feel like your team is lazy, frustrating, fail to understand the simplest things, everything is their fault, your people aren't good enough, or, on a very dark day, you're going to think they're actively trying to sabotage you. The reverse is true too, some days you are going to be blown away by how wonderful your team is.

So I want you to think about whether you want to put yourself on this human roller coaster. Maybe that is a hard question to answer right now. You may not be sure yet.

If you are not sure how to answer the question, it may help if you take a moment to consider your motivations and intentions for the role. Why do you think you might want to be a tech lead? If you are already in the role, why did you get into it?

I want to propose that if you are considering being a tech lead or accepted it because you want show career progress on your LinkedIn profile, are only looking for more money, or, worse, want people to think you are significant because you now have the word "lead" in your title, then I want you to think deeper–a lot deeper.

On that last point, I want to share an insight I developed by being around senior leaders. Most successful people I know in leadership do not think of themselves as particularly important. If you have ever heard a manager walk into a room and say, "I am the least important person here," it is because they believe it. They know their results depend on the work that their team accomplishes.

Read on for some better ideas why you should consider the tech lead role, because for all I just warned you about, it is a deeply rewarding step in your career.

Focus on the Process Not the Results

I know in this this chapter I have talked about results. I talked about your own results as an individual contributor and, now that you are you are moving into a tech lead role, I focused on how you are responsible for getting results through other people. Getting results matters–getting results is what most of us get paid to do.

However, if I can shift your focus away from results for a moment, then we have more freedom to explore one empowering, I think, answer to the question of whether you should pursue being a tech lead or not. The answer I am going to propose is the one that I think will put your best foot forward as you begin taking this journey.

I think the best answer to the question is that you want to be a tech lead because **you want to discover something new about yourself**. I think this is a much better answer than some reasonable alternatives like, "I want to do things better," or, "I want to change the world," or, "I want to have a bigger impact." Those are all perfectly popular, valid, and solid reasons, but they are focused on results and outcomes.

In your early days as a tech lead, unfortunately, you are unlikely to get the results you really want–like doing better things for your team or organization, having a bigger impact, or changing the world–because you still need to learn the basic management and leadership skills necessary to get those results.

If you focus on your own growth as a tech lead first, eventually you are going to start getting the results you want!

Focusing on your growth will mean different things to different people, but it helps if you can start with an **open mind**. By this I mean it is helpful if you not cling too tightly to your expectations for yourself, your goals, or the results you want. I do not mean to say you should let go of them. I only mean that I want you to not focus on them so much that you fail to notice all the other things you are going to learn along this journey.

With an open mind, you will be able to discover more as you grow than merely the results you seek. Some of the things you learn might be that the results you were clinging to may turn out to not be important to you, the organization, or the team. You may discover that there are aspects of the role you are better at than you expected. You may discover you are weaker in some areas than you thought you were. Have an open mind.

Next, **expect to fail** frequently in the beginning. As I discussed, getting results through people requires you to develop new skills. And learning new skills, just like when you were learning to program or even to walk when you were a baby, takes practice and mistakes. Accept that and be patient with yourself.

Finally, commit to **finishing strong and evaluating**–no matter how the project or task you have as a tech lead goes, get to the finish line one way or another. Then you can evaluate how it went.

You may discover that you do not really like being a tech lead, after all, but finish as strong as you can and learn as much as you can about the process. The experience will make you a better individual contributor later. You may love the role. But at the end of whatever project you have, make sure you get to the end and then evaluate what you learned. It is very hard to grow if you do not evaluate and learn from your experiences.

If you can keep that mindset and work through the skills I am going to propose in the balance of this book then, eventually, you are probably going to get **quite good** at the tech lead role. And some of you will become **great** at it. And all of you will change yourselves, a little, and maybe the world, just a little, along your tech lead journey.

From Here On, Let’s Get Practical

In the rest of this book, we are going to shift from my feeble attempts to inspire and motivate you to make the most of the role to, maybe gratefully, more practical topics.

First, in the next few chapters, we will explore what the tech lead role is, and what tech leads do. The tech lead role is quite varied, and the term gets applied loosely in most organizations, so we will spend a few chapters taking a deep dive to explore some of the ways it comes up and what some of the expectations are for it.

Next, we will explore what I call the five Pareto capabilities that tech leads need. By this, I mean the twenty percent, or so, of the set skills you may need as a tech lead that will get most tech leads eighty percent of the impact they need. Your mileage will vary, but if you can focus on developing:

- strong listening skills,
- a bias for action,
- an ability to craft clear and compelling technical visions,
- your tracking and adjusting skills, and
- having a positive growth mindset for you and your team,

then you are going to be well on your way to being a good, maybe even great, tech lead.

I am so excited to be on this journey with you!

What is the Tech Lead Role and Why Do We Need It?

“Tech lead” is a hazy term, to be sure. There are thousands of articles seeking to clarify the subject and a few books that touch on the subject. For reasons that will become clear in this chapter, I do not have the illusion that I will settle any debates on what a tech lead is or should be, but I do believe I can help you understand the role a little better.

In this chapter, we will explore what the role is at a high level, why it exists, and why you do not have to wait for anyone to give you the title to step into a tech lead role. In the next chapter, we will go through some of the common tasks and activities that tech leads often do.

Popular Thinking on What a Tech Lead Is

Patrick Kua was one of the first writers on the tech lead role. In *Talking with Tech Leads*, Kua defined a tech lead as “a software engineer, responsible for leading a development team, and responsible for the quality of its technical deliverables.” For Kua, a tech lead should still be hands on and writing code at least thirty percent of the time.

A similar, common perspective is that the tech lead is one of three key roles on an engineering team. In this perspective, there is an engineering manager, a project manager (or equivalent in an agile environment), and a tech lead. The engineering manager is responsible, generally, for people management and budgeting. The project manager is responsible for managing the project and schedule. And the tech lead is responsible for the technical deliverables and/or direction of the team.

In this view, the manager leads the people, the project manager manages the project, and the tech lead provides the technical leadership. This is a fairly clean division of labor in theory, but in practice the roles overlap a bit.

One important thing to note is that it is uncommon for a tech lead to have people management responsibilities, which is to say, tech leads typically do not have direct reports. If you do, you will still get value from this book, but people management is a big topic on its own that we will not explore in this book.

Some Organizations Have Formal Tech Lead Roles

In some organizations, the tech lead role is a formal job title where you might appear in the company directory with the title Tech Lead attached to your name. For example, Patrick Kua, who I just mentioned, was working at ThoughtWorks, a leading enterprise technology consulting firm, where they seem to have a formalized tech lead role. Other companies, like Google, also have a formalized tech lead role. Google even has a hybrid tech lead manager role that they use in some instances. Other software companies I have worked with also have a formalized tech lead role. The formal title may be less common outside the software industry.

In some cases, job descriptions for titles like Architect, Staff Software Engineer, or Senior Software Engineer often include many activities that are consistent with the tech lead role, whether expressly or by implication. Sometimes, managers will refer to engineers with these job titles as a tech lead.

Becoming a tech lead at these companies may require the company to formally promote you. The promotion may or may not come with an increase in pay. In some companies, the management may want you to take on the duties of the role a period of time before formally promoting you. Six months is a fairly common period of time to be in an acting role before getting promoted, but, depending on the company's budget, organizational design, and your progress, it could take more or less time.

In summary, some companies have formal tech lead titles or senior roles that come with tech lead duties expressly or impliedly. The role may or may not require a formal promotion or come with an increase in pay. In all cases, the scope of your duties and day to day work will surely expand.

Informal Tech Leads

I suspect, without proof, that the more common variety of tech leads are those who are informally in the role. Exploring the nature of informal tech leads sets us up to understand why we need tech leads.

The informal tech lead role could emerge in two ways. We will explore the first in this subsection.

Your manager may informally “tap” you to be a tech lead. Maybe he or she taps you to be tech lead for the team for a long period of time. In this way, you become almost like an assistant or surrogate engineering manager when the manager is away in meetings, which they always seem to be.

Alternatively, maybe he or she taps you to be a tech lead for a specific project or initiative. This is probably the most common varietal of tech leads. For example, there may be a project that comes up and your manager wants you to drive it, for whatever reason. Or perhaps the task is a little smaller, like upgrading the team’s continuous integration server, and the manager needs you to figure out how to do it and lead the project to complete the upgrade. There are millions of possible examples in this category.

Promoting Yourself to Tech Lead

The second informal way a tech lead role can emerge is organically. This could already be happening naturally.

Possibly it is because you are the strongest technical person on the team. Or, possibly, you have the most experience in the team. These are the two common reasons engineers organically become a tech lead.

There are other path too, however, for why your peers or managers may look to you to provide tech leadership, even if you are not the most experienced or strongest technician on the team. Some possibilities include:

- Perhaps you take more initiative than others on the team.
- Maybe you are able able to more clearly communicate technical issues.
- Maybe you have a poise to the way you hold yourself that inspires confidence.

There could be many qualities you demonstrate that your manager or other business stakeholders (like product managers) seem to gravitate to you to provide updates or insights about the team.

This category is a de fact lead role and it is entirely possible for you step into it without anyone asking. In fact, this is often a fast track to earning your way into a formal tech lead role or an even bigger promotion.

As a manager, I was endlessly grateful when this happens. It could start in small ways, such as noticing something that needs to be done on the team, maybe some tech debt that you want to tackle, maybe showing the team how test driven development works, maybe you found a new framework that will help the team. You can step up and fill the void or drive a vision.

Why We Need Tech Leads

Organizations need tech leads, whether they work under formal or informal titles. On any reasonably sized team, there are usually going to be too many technical choices and activities for an engineering manager to be able to lead and track every choice, so leadership has to be distributed among the team.

As a manager, I was spread far too thinly to be able to understand every facet of my teams' work. So I liberally handed out informal tech lead roles to those on the team who seemed to have some of the qualities I will describe later in the book. I did not care their formal job title was when I moved them into informal tech lead roles. Of course, if they were successful in the role, we would want to have discussions about formally promoting them as soon as we could.

In the short term, probably much like your own managers, I needed people who could lead others through much of the technical minutiae that always needs to be solved and the many projects and initiatives that need first level leadership.

Thus, there will always be a strong demand for impactful tech leads.

What Do Tech Leads Do?

In this chapter, we will explore some of the activities that tech leads do. This might be a relief to you. We could have started with this chapter, but I felt we needed to first set some context for why the tech lead role exists and a sense of the endless variety of tech lead configurations. Almost everything I describe about these common tasks may apply to your actual role as a tech lead. Or maybe none of it will.

We will start with a non-exhaustive list of the typical tasks that many tech leads perform. Then we will pick out a few common themes to explore later in the chapter. Later in this book, we will dive into five key, crosscutting, capabilities and skills that will make you more effective in all these tasks. For now, we will just explore some of the work and the themes to consider in your role as a tech lead.

Tech leads may, on any given day, be doing some or all of the following:

- teaming up with engineering management to be the interface between the technical team and other related functions, such as product management, quality assurance, project management, and etc.;
- architecting, designing, or driving buy in for key technical decisions or designs;
- leading design or code review sessions;
- mentoring junior developers and helping them to solve problems in the process;

- understanding, clarifying, and triaging bugs and tech debt;
- tracking what the team is doing at a deeper technical level than typical project management;
- reporting the technical status of the work to the next level of management or other constituencies;
- setting technical standards and project goals for individual contributors and making sure their contributions align to the standards;
- continuing to make their own technical contributions; and
- much, much more.

Now let us take that list and teach out some common themes for how your work will change as a tech lead. The first, most common theme, is that you will need to be spending more time communicating. Second, you need to change your mindset to have a broader view of the project, team, and organization. Third, you will need to become much more proactive than you previously were.

We round out this chapter with some discussion of the separation between technical, hands on skills versus leadership skills as well as repeat that the role is a fluid one in which you can expect to shift what you do frequently.

You Will Be Communicating More–A Lot More

One activity that may have stood out in the list of activities is that the amount of communicating you will be doing as a tech lead and the need for you to be effective as a communicator changes quite a bit. Of course, individual contributors need to communicate as well. But the quantity of communication is significantly lower than a tech leads.

For example, one common activity for tech leads is to act as the interface between the technical team and product management, which is going to require more meetings to attend. After the meeting, you will need to then communicate with your team to explain what the product manager wants. Then there will be follow up conversations, Slack messages, emails, or one on one conversations with the product manager.

Communication can consume many hours of a tech lead's week. Plan to spend even more time and energy on this activity if you are not yet effective at communicating.

Other tasks on the list such as driving buy in for technical decisions, participating in design or code reviews, mentoring junior developers, tracking team activity, reporting status, and so on all require a heavy emphasis on communication. Being in a lead role means that you need to get results through other people, and this implies that communication will be something you will be doing a lot more of as a tech lead.

You Will Be Taking a Broader View

Your role as a tech lead also changes what you focus on in your daily work. Typically as an individual contributor, it is sufficient for you to focus on your individual task. It helps if you can take a broader view, but it typically is not required.

This broader view applies to both technology and the team. For example, tech leads often find themselves in tasks where they need to architect or design solutions so you have to design not only on the immediate task but also the long term roadmap of the product, company, and industry. By way of another example, tech leads will be in code reviews where they need to think about the long term maintainability of the code or the consistency with existing patterns. You also need to know how to make good trade offs between shipping code in the short term and any long term tech debt the team is accruing.

You also need to start taking a broader view of the team. Although tech leads do not typically have people management responsibilities, you still need to help your manager develop the team and its overall capabilities. Many of the activities require you to work very closely with the team, sharing information, making adjustments, providing feedback, and mentoring. While the leadership you are providing is fundamentally technical, there is a large human component to it.

If you want to get the best work from your team, you need to have a broader view beyond the immediate tasks. You need to be mindful of what motivates people, connect those motivations to the tasks at hand, and understand how sustainable the pace of development is on your team. In other words, I believe tech leads will be failing in their duties if they focus solely on the technical leadership aspects of the work and overlook the health of the team.

You Will Be Active, Not Passive

As an individual contributor, you should ideally be proactively looking for useful contributions and work to do. Usually, individual contributors can do their jobs successfully by letting others give them direction, however. As a tech lead, your activities and behaviors will need to shift.

We could pull any task off the list of activities above, and I could explain why your behavior will need to shift from passive to active. Take mentoring junior developers, for example. To be a mentor, you will need to initiate those kinds of conversations or interactions. You cannot wait passively for them to approach you for mentoring. There will be moments that come up for mentoring, such as a discussion about a technical task, or a code review, or a one on one meeting and you will need to proactively notice that this is an opportunity for mentoring and do it.

The same is true if you are working to design or architect a new system. To do this successfully, you will need to proactively seek out information about the future design of the products or the needs of the company. You will need to proactively validate the design against your team to make sure they can execute on it and that it is valid. It will be insufficient to simply create a Google Doc and ask people to comment on it. Waiting around for feedback is not driving a technical decision the way impactful tech leads do.

Finally, when you combine the broader vision we just talked about with the need to be active, not passive you will find future tasks are far more successful because you will take actions today that help in the future–whether it's preventing a problem or taking advantage of an opportunity.

Still a Deeply Technical Role

The tech lead role has two words in it: tech and lead. Interestingly, both words have the same number of characters, so the concepts seem balanced in characters, but the actual weighting of your tech skills and lead skills will shift considerably from day to day in practice.

So far, we have talked a lot about soft skills, and there is a lore more talk about soft skills to come in the book. Before we continue, then, I should make clear that tech leads will need to stay current with their technical skillset and continue to invest in that core skillset. Staying current not only keeps you grounded in your technical field but also keeps you engaged in the engineering team through the day to day technical work and the many small choices that need to be made as your team does its work.

Some say keeping your hands on the code increases your respect with the team, but I tent to think “respect” is the wrong word. I think people will respect you when you respect them. Instead, I think at this level of leadership, you are going to be talking with you team about the codebase frequently, so writing some of the code with the team keeps you fluent in their language. If you are going to be talking about the code, it helps if you know what you are talking about.

As you grow in your technical leadership, the emphasis of your learning may evolve from a deep learning on, say, a specific aspect your technology stack to a broader fluency in software engineering practices (or whatever specific technical field you are in), such as more sophisticated source control methods, continuous integration techniques, software craftsmanship generally, design patterns and anti-patterns, and more.

If you move up into full blown engineering management, you may largely leave the code behind. If you stay on an architect track then you may be staying close to the code for the rest of your career. Those decisions will come later in your career. For now, as a tech lead, you need to stay close to the code. Being a tech lead is a deeply technical role.

This guide will not say much more about technology since only you know about your technical field and I could not possibly cover them all here, obviously. Besides, you can likely find plenty of guidance on what you need to know somewhere else.

The Tech Lead Role is Fluid

By now, hopefully it is clear that the tech lead role is a fluid one.

The role itself may shift and change over time. Your time in the role may come and go if you are in it informally. If it is a formal role, the projects or teams you are on may have more or less need for you to step up and lead at any given moment.

Moreover, the day to day tasks you may find yourself performing as a tech lead are not going to be consistent nor always predictable; instead, they will change depending on the organization, the team, and state of the project.

For example, you may be an individual contributor and unexpectedly find yourself in a moment where you need to, say, lead a team on how fix a bug, be the point person for developing a complicated new feature with the rest of the team, or define a new architectural approach where leadership and coordination skills are crucial to the project's success. When the bug is fixed or the decision is made, the need for your leadership declines.

Put differently, you do not need to think about leadership all the time as a tech lead, since most of your day will likely continue to be spent deep in whatever technology stack your team uses. Being a tech lead, as a I said, is a deeply technical role.

That said, the remainder of this book is focused on making you effective when it time to step in and lead. So I hope you will read on.

Your Tech Lead Toolbox: The Five Pareto Capabilities

If you have not heard of the Pareto principle before, Vilfredo Pareto was a Nineteenth century Italian economist who noticed that twenty percent of Italians owned eighty percent of the land. He went on to study this phenomenon.

The Pareto principle was popularized as the “80/20 rule” because it seems to hold in so many circumstances such as economics, sports, safety, crime, gambling, health care, and many more. I believe the Pareto principle roughly can hold for your role as a tech lead as well.

The following five skills or capabilities, if you focus on them and grow in maturity within each of them, will make you an impactful tech lead. These are not all the skills you need all of the time, but these carefully curated five will take you eighty percent of the way down the road to being a great tech lead.

We will go into more detail in the chapters that follow, but, in summary, the five Pareto capabilities are:

1. an ability to **listen** effectively to your team, your organizational network, your management, and etc.;
2. **taking action and initiative** for you and your team to be accountable;
3. crafting **clear and compelling visions**;

4. **tracking and adjusting** the team's work, and
5. paying attention to your own and your team's **growth mindset**.

Foundation: Navigating Hard Conversations

Before we get into the five core Pareto skills, however, I want explore a key aspect of your work as a tech lead that few discuss publicly. It cuts across almost all the activities you will have to perform and the capabilities we are going to go into detail in next.

If you recall from the list of common tech lead activities, most of them, to be effective, require you to speak up in certain moments and therefore, possibly, trigger conflict with others. For many of us, anxieties about the possibility that they will need to engage in a hard, contentious conversation, undermines their impact as tech leads and slows their growth in the role.

For example, let us imagine in your tech lead role you are, say, the final decision maker for a technical decision between two options, say, using Ruby or Elixir as a programming language. Team members disagree with each other on the relative merits and tradeoffs of each choice. No matter which choice you make, some members of the team will disagree with you (especially when it comes to programming languages).

If this kind of disagreement on the team concerns you, you may end up making the choice based in part on your desire to avoid conflict instead of staying true to your vision, to name just one possible negative side effect. For example, you might go with Elixir simply because one team member is particularly vocal, even if you think staying with Ruby will make the code more consistent with the rest of the company. (Just an example.)

If you have spent most of your career thus far as an individual contributor, conversations like the example I just provided could be quite uncomfortable. They are hard for everyone in any context, but they can be particularly hard as a tech lead where you likely do not have much experience nor true authority in hard conversations.

I highly recommend you read books like *Crucial Conversations* by Kerry Patterson, Joseph Grenny for more detailed guidance on how to engage in these challenging conversations. In a nutshell, the authors recommend entering into the conversations with the objective of getting all of the information and viewpoints out in the open, including your own.

They, generally, recommend you:

1. Start with a positive, open intention.
2. Stay in the dialog as long as you need to in order to get the points out.
3. Create a safe environment for everyone to communicate.
4. Avoid getting caught up in the emotional aspects of the issue.
5. Agree on a mutual purpose for what you are trying to resolve (remember this point when I get to vision crafting).
6. Focus on the facts, not the narrative.
7. And, agree on a clear set of next steps.

Put differently, you **can** learn simple, effective skills for navigating conversations that, today, may seem impossible. It will take practice and maybe give you a little stress in the beginning, but you can get there.

I believe the Pareto capabilities that we will explore next (especially listening) will give you valuable tools, specific to the tech lead role, to help you work through hard conversations or, even better, prevent them from coming up in the first place.

Sooner or later, however, conflict and hard conversations will come up. As you review the capabilities that follow, keep what I said in this section in the back of your mind as we explore them. Ask yourself whether I trigger some anxiety because you are concerned about hard conversations.

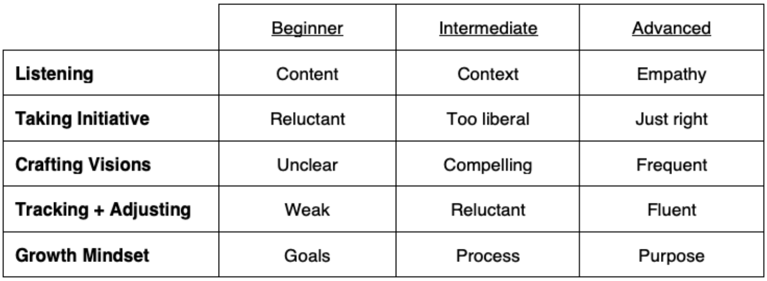
Tech Leadership With the Five Pareto Capabilities

Leadership is a big topic. In fact, it is far too big a topic to cover in this short book. But I do want to focus on the one aspect of it that matters to making you impactful in your role as tech lead: **influence**. John C. Maxwell says, “Leadership is influence; nothing more, nothing less.”

As we discussed, the tech lead role exists because managers need to distribute leadership among engineering teams to build momentum and to get things done. This means the organization needs you, as a tech lead, to influence the actions and activities of your team.

That might sound daunting at first, but I think you will find the five Pareto capabilities are practical, easy to act on, and aimed squarely at helping you build influence with your team.

The following table summarizes the capabilities. On the left, vertical axis, I list the five capabilities: listening, taking action, crafting visions, tracking and adjusting, and the growth mindset. Then on the horizontal axis, I propose levels of growth or maturity within the capability ranging from beginner level to advanced levels.



	Beginner	Intermediate	Advanced
Listening	Content	Context	Empathy
Taking Initiative	Reluctant	Too liberal	Just right
Crafting Visions	Unclear	Compelling	Frequent
Tracking + Adjusting	Weak	Reluctant	Fluent
Growth Mindset	Goals	Process	Purpose

Summary chart of the five Pareto capabilities

In the chapters that follow, we will take a deep dive into each of the capabilities, including more detail about the maturity scale in each.

Capability No. 1: Listening

In the earlier chapters I emphasized that you would be communicating a lot more of as a tech lead. Indeed, communication is the hallmark of leadership.

Communication is a big topic, however, and has many aspects to it–too many for this short book. It is such a big topic, in fact, that one can get a four year degree in communication and advanced levels too. Of all the facets of effective communication, we are going to focus on being a skillful listener.

Listening is the first skill to focus on because, through listening, you get the raw data you need to take the right and to make the most of all the capabilities that follow. You cannot lead if have not heard what the organization, the team, the management, or the individuals have said. Being a great tech lead starts with great listening skills.

In this chapter we will describe what aspects of listening I want you to emphasize to increase your maturity level in this capability. Then, because I can’t say it enough, I will drill into even more detail on why listening is so important to your role as a tech lead. Finally, we will close the chapter with three possible blockers to your ability to listen skillfully.

Maturing Your Skill as a Listener

Before we start, remember the maturity model is here to help you improve. You are not simply a good listener or a bad listener–despite what your significant other says–it is not a binary capability. It is more correct to say that you are either more or less skillful at it. Moreover, your listening skill probably varies a bit with the context and/pr the speaker, so you can think of your skill level on a continuum within a context, where each level builds on the next.

Beginner listeners focus on content

At the **beginner** level, tech leads tend to focus principally on the content of the message transmitted to them. Beginner listeners focus on facts, the factual assertions, and the raw data coming from the sender. Please don't misunderstand me. We are in a technical field, so rich, technical content is is obviously critically important to receive. But there are more powerful ways to listen that we will explore.

Unfortunately, some tech leads fail to achieve even this level. They simply do not listen. There could be many reasons for it, which you should think about and reflect upon. Possibly there is someone in your life that can help you diagnose this. Or you could talk to a mentor or one of the coaches from our community network that I will tell you about at the end of this book.

If you are listening, one way to be more effective, even at this level, is to make sure you are actively soliciting content. That is, do not passively listen for information. Instead, probe, ask questions, clarify what you heard. Ask people to repeat themselves so you are sure you heard what you thought you heard.

One way to self assess where you are on the listening scale is to reflect on a recent work conversation. Ask yourself questions such as:

1. Do you actually remember the content that the sender transmitted to you?
2. Were you active or passive in the conversation?
3. Did you miss details that you wish you had not?
4. Did you fail to solicit those additional details?
5. Did you disregard or ignore some of the content?

Intermediate listeners receive richer data

At the **intermediate** level, you are able to receive not only clear content, but now you are synthesizing content with context, such as the body language of the sender. This is a big step up in your listening capability because, even in our highly complex field, there is almost always more to the message than the raw content. Thus, being able to read non-verbal signals correctly within the context is the mark of someone who is making significant growth in their ability to listen.

A large percentage of people are fail to achieve this level very often, so if you are able to do it, and do it consistently, you will be making significant progress in your tech lead journey.

One way to self assess is to repeat the exercise I just mentioned. Ask yourself questions such as the following:

1. Did you have a solid understanding of the speaker's context?
2. Were they under pressure or relaxed?
3. Were they in a physical place where they could speak openly?
4. Or were they surrounded be managers or other people who might influence the message?
5. What about the sender's body language or tone of voice?
6. Did they seem confident? Nervous? Relaxed? Body? Happy? Frustrated? Bored?
7. Did certain subjects seem to change their body language?
8. Now, importantly, how did those additional signals influenced your understanding of the raw content?

Advanced listeners suspend their own needs when they listen

Intermediate listeners have rich data to work with. However, even at the intermediate level, your ability to hear the full message can impeded by your own needs, whether it is pressure from the project, your own insecurities, or your feelings about the sender.

This internal processing and thinking can block important signals in the message or filter aspects of the message. Thus, **advanced** listeners are able to put their own needs aside and absorb the full message empathically. We will talk more about blockers to communication later in this chapter, which shoud increase your ability to listen more empathically.

We are increasingly talking about empathy as an industry. There is a lot to say about the topic, but here I want to suggest that I only mean something practical. By empathic listening, I mean you are able to suspend your own needs and internal dialog while you are listening.

The power of this level is twofold. First, people feel truly listened to, and in some cases they can form strong bonds with you as a result of it. Second, you are actually going to hear the full, true message.

This is a very hard level almost for everyone to attain and I am not sure anyone on earth could consistently so relax and do your best to remove the blockers we will talk about below as often as you can. Since this is so hard to do and actually does take significant energy, you may want to limit your times when listening is critical, such as an important meeting.

Again, to self assess whether you are at this level, think back to a recent work conversation and ask yourself if you remember more of your own thoughts from the conversation than the speaker's content or context, such as body language. Also, think back, were you interrupting the speaker? Were you offering your own advice before hearing everything the speaker had to say? Whose needs seemed more important to you in the conversation? Yours? Or theirs?

Listening Is Critical to Developing Your Tech Lead Influence

Some ignore the need to listen. Maybe you have worked for a tech lead or manager who seemed to operate like this. They simply bossed people around, apparently confident in their role as tech lead and/or their superior access to management or information. It probably was not a great experience.

If I didn't make this clear earlier in the chapter, I strongly believe we badly overlook the importance of listening in our roles as tech leads. Listening is bedrock, ground zero foundational to leadership because it is a the key to unlock influence. Listening is critical to building your influence for at least two reasons. First, it gives you the data you need to understand the team's motivations. Second, it creates the connection necessary to build your relationships, especially with empathic listening.

A large part of your job as a tech lead is to connect the company's mission, or the project, or the tasks to the individuals who will work on those tasks. The most effective way to do this is to connect the tasks to the individual's motivations. And the only way you will know what their motivations are is to have listened closely to them. As a lead, you want to connect work to motivations, not just assign tasks.

You could try to ask them, "Hey, what motivates you?" But I would be surprised if you got an answer that you could really use, especially from a junior developer. Instead, the way to get the true information is from the day to day work. It could take weeks or months of close listening to really get enough data to be able to understand the team's motivations.

Listening is also critical critical to building relationships on the team. From *The Lost Art of Listening* (Second Edition) by Michael Nichols:

> Few motives in the human experience are as powerful as the yearning to be understood. Being listened to means that we are taken seriously, that our ideas and feelings are recognized, and, ultimately, that what we have to say matters. The yearning to be heard is a yearning to escape our isolation and bridge the space that separates us.

Think about when you were an individual contributor. How did it feel when you voiced your concerns, and your tech leads or managers did not really listen or seem to care about your concerns, and yet they told you to proceed with tasks regardless of your concerns. Not being listened to or heard is a painful experience that all of us go through pretty regularly.

As a tech lead, you can build a strong connection simply by listening. Listening and truly hearing the people on your team validates their own self worth because they believe they are being taken seriously and they feel like individuals connected to the organization–not merely cogs in an organizational machine.

Blockers to Skillful Listening

It is true that some people are hard to listen to. Sometimes they are inarticulate. Sometimes they mumble. Sometimes they simply do not say much.

It is hard to change other people, but you can improve yourself and removing your own internal blocks to hearing. And the main blockers to listening come from your own thoughts and distractions.

According to *The Lost Art of Listening* by Michael Nichols, you need to suspend three categories of thoughts while listening: “memory, desire, and judgment.” Nichols was speaking to all listeners, not just tech leads. So I want to translate what he said to your role as a tech lead.

First, think of “memory” as your experience and past knowledge of your technical domain or environment. When you are trying to understand what someone on your team is saying, listening to key technical details about the task or a getting a good read on an engineer’s confidence in their ability to complete a task on time, you need to shut off your own knowledge and experience for a moment. Your memory of how the system works or how code should be written or whatever else you are remembering is irrelevant when you are trying to hear what someone else is trying to say to you.

This is hard because, when an engineer is relaying something important or some aspect of his or her work, it is likely to trigger your own memories of your own experiences and the things you know about the project or technology. Suspend those thoughts in your mind while you are listening. Later, when you start making adjustments, which we will discuss later, or crafting visions, then your knowledge and experience is going to be very useful. By right now, you are trying to understand what someone else is telling you.

The next category of thoughts to suspend your desires. By desires, I mean that, as a tech lead, you need to get things done. You have pressure to get things done on time. You may have an urgent need for an engineer you are listening to to complete their task. You may be frustrated that the work is not consistent with the vision you have or the standards you hold.

The problem is that those needs are going to filter or color what you think you are hearing. Worse, if you articulate those needs while the person is talking, you run the risk of shutting down the speaker or triggering a conflict when what you really need is information first. (We will talk about making adjustments later in the book.)

Of course these needs are important. Deadlines from the company do not change so that you can practice being a better listener. I know that. I am only asking you to suspend your needs long enough so you can actually hear what is being said.

The next category of thoughts to suspend come from your judgment. Judgmentalism is a serious problem in our business. As a tech lead, you might actually be in the role because you are the strongest technician on the team or maybe you are the one who voices opinions most often. You may have strong and sound judgments about how things should be. Your judgments may be valid. They are useful for knowing what to do next with the information you receive.

While you are trying to hear what someone is saying, try hard to suspend your own judgments. For example, an engineer may be explaining a solution to a problem that includes an open source library you really do not want to use. While the person continues to talk, you are starting to feel frustrated that this engineer wants to use the library. You start wondering how you are going to tell him or her to not use it. And while you are having these judgmental thoughts, you are missing a lot of what is being said.

In summary, keep in mind the three blocks to listening: memory, desires, and judgment, and try to suspend them while listening in your next conversation. You may be surprised by how much more you hear as a tech lead.

Is your team block you?

There is a special case I want to mention before we move on from this topic. You may not be hearing enough because the people on your team are not saying enough. If this is the case, *The Lost Art of Listening* offers this possibility: "When people don't say much, it's less likely that they have nothing on their minds than that they don't trust the other person to be willing to hear it."

I mention this situation because I have seen it on many teams, but I am not accusing you of it. But I do want to propose, if you have this situation, it is possible that your team has experience trying to communicate with you while you have been mired in your own listening blocks, and therefore it is possible that some members on the team do not feel safe communicating with you.

If you suspect this is the case, think about whether your memories, desires, or judgment is getting the way of hearing what your team has to say.

Capability No. 2: A Bias for Action

Having a bias for action and taking initiative are normally thought of as skills managers emphasize for individual contributors. Given that you are reading a book about being a tech lead, you probably are already someone who takes a lot of action and initiative, but it can be an impactful skill to bring with you to your early leadership roles. It is the skill and personal drive you need to step into as many of the right leadership moments as you can.

Stepping Into Leadership Moments

I often say leadership is not really a role. Leadership is an **action**. It is not a noun, it's a verb.

Think back to when you were an individual contributor. Imagine you are a senior engineer in a conference room, along with three other engineers. The tech lead, director of product, and your engineering manager are in the room with you. There are seven people in the meeting and three of them have manager or leadership roles.

In this meeting, they are talking about upcoming features to add to the codebase that you primarily work on. The tech lead, the engineering manager, and product manager are negotiating what features are next for development.

Nobody is standing at the whiteboard, and there are whiteboard pens (that actually work) sitting on the desk. Nobody knows this codebase as well as you do, including the tech lead. Some of the things the tech lead and the engineering manager are saying about the code you work on are incorrect. You know how to correct what they are saying. But you are are a little nervous about speaking up. This is a leadership moment you could step into. Moments like this come up many times throughout the day.

A leadership moment is one in which you can make an impact and build your influence using the tools we are discussing in this book–listening, taking initiative, crafting visions, and so on–even if you do not have a formal leadership role. In my example, you can step into the moment by grabbing one of the Dry Erase markers, standing up at the whiteboard, and helping to resolve the disagreement.

I want to propose to you that one way to think about being a leader is to adopt the self identity that you are going to step into as many leadership moments as possible. The moments come up frequently, but they're often fleeting–if you miss it, the opportunity is gone forever. Your tech lead influence will grow by stepping into as many leadership moments as you can, which requires you to take the initiative to step into them.

Let's Talk About Your Energy Level

In the introduction I explained I had the opportunity to observe many tech leads who were more effective and impactful than I was at work. One of the most prominent differences between them and me is that they seemed to have a much higher energy level than I did when I was a new tech lead. (I have since improved!)

If you have a high energy level, that leads directly to creating a high bias for action and taking a lot of initiative.

By energy level, I do not mean the kind you might see from cheerleaders or stage dancers.

The effective tech leads were hardly cheerleaders, but it did seem like you could feel their presence everywhere on the team. These high energy tech leads seemed to be actively engaged in almost every check in, Jira ticket, Trello card, Github issue, or whatever. They actively expressed their opinions in the comments. It sometimes seemed like they were everywhere on the team all the time.

They didn't stop with the team, however. They seemed voracious for information about the company or the product, where it was going, how their team could contribute (or, what to stay away from). And of course, they always seemed current on technology trends.

You don't need to be outgoing to do this. You can be fairly quiet and introverted and still be a high energy tech lead. By high energy, I am not talking about your personality, but I am talking about how much energy you project into your team and organization.

When you were an individual contributor, your energy level and your bias for action was not critical to success. You can passively wait for work to come your way. As a tech lead, the situation reverses, and you will find it much easier to increase your bias for action and initiative taking if you bring up your energy level.

If you are not currently a high energy tech lead, don't worry. There is good news on this point. I certainly was not a high energy person either, but what I found is, if you dig deep, and start putting out more energy, you will likely start getting results and feedback you like, and then you are going to find even more energy to give.

Maturing Your Bias for Action Capability

Applied to your tech lead role, maturing in your bias for action capability focuses on how often and effectively you seize those leadership moments. In contrast to the stacking skillset of listening, think about maturity in this capability as a sliding scale.

Beginners are reluctant to step into leadership moments

Beginners are reluctant to step into leadership moments. There could be any number of valid reasons for reluctance to step into these leadership moments, from a basic lack of skill to a more fundamental fear of failure. Indeed, there are probably as many reasons for reluctance as there are tech leads, if not moreso.

One way to start engaging more often is to take a little pressure off whatever is holding you back is to start thinking of leadership moments as simply an opportunity to explore your progress the way I asked you do back in the first chapter. Once you spot a leadership moment, take action. Even if you do not feel fully ready or like you have every skill you need, or you fear failing, but take step into the moment anyway.

If you approach leadership moments with that mindset, you may feel more confident to step into more of them. You could try the following simple success driven cycle:

1. Take action or commit your team with an open mind.
2. Anticipate that you or the team may fail completely or, more likely, you won't achieve every result you seek perfectly. Importantly, align your management on your concerns and the risks so you have some "air cover." Don't be negative–just be realistic.
3. Finish strong and study the outcome: what worked and what did not?
4. Learn from the experience. Then share the learning with the team and your management.
5. Step into another leadership moment and repeat the cycle, bringing your learning with you. Improve.

To self assess yourself on this bias for action capability, think back to a recent leadership moment. Did you let it pass by? Did someone else take the opportunity? What stopped you? Did you feel as if you lacked the skills you needed? Were you afraid of failing?

Those at the intermediate level step into leadership moments too often or don't follow through

An **intermediate** level of maturity with this skill is exactly the reverse of the beginner level. At this level, you may find yourself jumping into many leadership moments and volunteering your team to take initiative with ease. This is a big step up in maturity and is an easy path to promotion because you are likely making yourself visible as an emerging, high energy leader at your organization.

However, there is, of course, a downside risk at this level. If you dive into too many leadership moments or put your team on too many hooks, you could be stretching yourself and your team too thin, and your outcomes may be uneven or worse.

To self assess whether you are at this level, think back to the moments you have stepped into. Did you feel like you were pressured into the moments? Do you feel like you are being rushed or unfocused? Do you feel like you have too many things in process? Do you feel like your team is producing too many failures and not enough successes. It could be you are stepping into too many leadership moments. Or, you could just be having a bad run of projects–only you will know.

At the advanced level, you step into every right moment with force and impact

The **advanced** level, then, is the middle way stepping into too many leadership moments and not enough. But the middle path is not about simply about scaling back.

At this level you have the maturity and understanding of both your own capabilities as well as your team's capabilities and you match them up to only (but as many as possible) those leadership moments to have the most impact.

At this level, you also may be finding yourself delegating more tasks and even distributing some of your own leadership, which is a strong indicia that you are growing into leadership.

Capability No. 3: Vision Crafting

In this capability, we are going to explore how effectively you can craft and communicate clear and compelling visions. This capability requires you to synthesize a lot of information into a clear and compelling vision of what the team needs to do to be successful in any given moment.

Doing this well means you need to draw upon multiple sources of information. Just a few possible sources include:

- your understanding of your team's capabilities (through your listening) and their motivations,
- your situational awareness of the demands of the leadership moment you are in, and
- the broader needs of the project, team, and organization.

How you create the vision is not important. You might create the vision on your own while you are taking a shower on a random Tuesday morning. Or, you may collaborate with your team to craft it. Some teams respond differently to different sources of a vision and have different needs. Sometimes have a need to participate; some do not.

Creating Clear and Compelling Visions as a Tech Lead?

The word "vision" seems like a big word. Using it may evoke almost impossibly aspirational images in your mind of visionary leaders like John F. Kennedy, Martin Luther King, Nelson Mandela, to name just a few global leaders. You may also think of speeches you've heard by tech leaders like Steve Jobs, Elon Musk, and etc.

When I speak of vision for your role as a tech lead, I am talking about something far more mundane, practical, and achievable.

At your first-line leadership level, your vision for a project, feature, or initiative may feel so modest that you may overlook that you have a "vision" at all. It may simply seem like an intuitive sense of what needs to happen in a given moment.

It takes clarity of vision to make things happen, and so you should focus on it as a distinct capability. You would be surprised how many tech leads overlook this step.

For example, you may be starting to lead a database migration project requiring some code changes from your team, a data migration strategy, and a new release. That may feel like "just work," but perhaps you have an idea for how it will go well.

If you are good at this, you may tell the team something like, "Here's what I want to happen: the company is implementing a new database, which is going to be key for the next generation of our app, so we want to get a migration done soon. We want it to happen with the least downtime possible and, ideally, zero data loss for users."

If that does not sound like a "vision" to you, then consider an alternative from another tech lead: "So, uh, our team has been tasked to, uh, migrate the database." Maybe you have heard something like this in the past.

In the modest vision statement I proposed, the team knows why they are going to be doing the migration, what the importance is to the broader organization, and what aspects of the migration need to be emphasized once they start working–namely, low downtime and no data loss.

In the alternative example, the vision (so to speak) could be accomplished in any number of ways, some of which may not match up to what you or the company wants or needs.

Whether you find the vision I offered compelling and clear is not important. It does not even matter whether I find it clear and compelling or not myself. Only the team that needs to execute on your vision can evaluate whether your vision is clear or compelling. So you need to understand the motivations of the team to have a clue whether it will be clear or compelling or not.

Similarly, what information you put in your vision will also depend on the needs of the team. If you have fairly junior team members you might need more content, for example.

At a minimum, however, I think a good vision statement will at least need to include **why** something needs to happen and some notion of **how** you want it to happen.

For some teams, you are going to have to go into a lot of detail about the why, especially if you have a new team or a new project or if you have very junior developers.

For the how, you have a similar set of considerations. For example, if you have senior folks that know their way around the organization and its products, you can probably just putting some guard rails in place in your how. For junior developers, you may need much clearer guidance and structure.

Maturing Your Vision Crafting Capability

You do not need imitate Nelson Mandela. Creating clear and compelling visions is well within your current abilities, even as a first time tech lead. The only questions are:

1. Are you crafting clear visions?
2. Are you making them compelling?
3. Are you repeating them often?

Growing in your capability for vision crafting follows those questions in a stacking progression.

Beginners fail to create clear visions

At the **beginner** level, you may not be aware that you should define a vision. Or, if you sense you should, you may fail to clearly articulate what you want the team to do.

Since you were likely a competent and capable individual contributor, you may have an intuitive understanding of what needs to be done. Thus, you may mistakenly believe that your team shares the same intuitive understanding and "just knows" what needs to be done.

An easy way to assess where you are in this capability is to ask yourself what you told the team about the project.

1. As you were describing the project, can you clearly remember a moment where you gave them a statement about what the mission is?
2. Did you explain why and how?
3. Based on your understanding of the team make up (from listening), should it be clear to them?
4. Was it compelling?

Most of the tech leads, and even more senior leaders, are at this level so don't be too concerned if you find yourself at this level.

Intermediate level vision crafters just don't articulate often enough

At the **intermediate** level, you are able to craft clear and compelling visions based on your knowledge of the team and the situational needs of the project, team, and company.

The vision you articulate to the team is attainable and clear (like getting through the database migration described above), and it is messaged in a way that the team believes they can execute on it (even if they need to stretch a bit).

To self assess whether you are at this level, write down what you said. Does it make sense? Is there any reason in the statement that an engineer would want to come to work on the project (aside from earning a paycheck)? Maybe even better, ask a team member to these questions to get some feedback on whether you are doing well or not.

Advanced vision crafters repeat their clear, compelling visions often

The **advanced** level of this capability is a powerful, impactful one, even with fairly mundane initiatives. This is where you are able to articulate not only a clear and compelling vision, but you articulate it often in many situations.

If do not do this consistently today, don't worry. Again, many senior managers fail to reach this maturity level. But if you can do it, you can be highly impactful. As a manager or leader, it is easy to be clear on the vision in your own head and forget that your team is not as focused on the vision as you are, and they are likely to drift as they get bogged down in the day to day work.

When you are able to reach this level, it means that you can remember that your team needs constant reminders (we will get to the course corrections in the next chapter).

You could create the clearest, most compelling vision and articulate it from a giant stage in front of your whole team (with fireworks too), and people will forget. They will get entrenched in the details, lose track of how each task ties to the vision, or how to apply it when new issues come up. So as a lead, you need to repeat it. Often.

To self assess if you are at this level, simply ask yourself how often you repeated the vision in the past week or past few days.

Capability No. 4: Tracking + Adjusting

Like crafting visions for relatively small technical initiatives, this capability can seem subtle and easy to overlook. You may barely be conscious that it is a capability you need to pay attention to and do well. It is critically important to track and adjust often if you want to drive results through others.

As important as the other capabilities and activities are as a tech lead, this capability is a strong indicator of how you are growing in the role because many of other capabilities come into play and compound as you practice this.

Tracking the Team

This capability is closer to a management capability and function that most of the other skills we have discussed before. Tracking work and making adjustments to what your team is doing are two different motions. Tracking draws on your initiative and listening skills to get up from your desk, walk the halls, and check in on the team. Even if you do this virtually or over something like Slack, it takes a combination of proactivity to reach out to people and strong listening capabilities to understand what each individual contributor is doing.

Tracking simply means that you are clearly and frequently monitoring each team member's progress against the vision on both technical and human dimensions. You may already have project managers that do this at the project level, but as a tech lead you need to know more detail about **how** the work is getting done than **when** it will get done.

One way to do this is to consciously ask your team questions designed to resolve these constant issues:

1. How closely is the technical work aligning to the vision you crafted?
2. What tech debt is the team accruing as they try to hit schedule targets?
3. Is the work staying aligned to the standards for the task, project, and organization? If not, what trade offs are you making and are you comfortable with them?
4. Is the work still aligned to the individuals' motivations? If not, how sustainable is the situation and what could you do to correct long term?

If you are a strong listener and good with taking initiative, the tracking motion should fall into place pretty easily. If you are not doing very well tracking, go back and reassess yourself on your initiative and listening capabilities, knowing that tracking is something you need to do effectively.

One thing to note is that you probably have a stronger or easier relationship with some members of the team than others, so you may be more effective tracking those you are close to than others. While this is natural, the goal is to be to be able to track effectively across the entire team, even those who you do not naturally have rapport with.

Making Adjustments

Making adjustments is usually the harder motion for new tech leads. By making an adjustment, I mean you ask or direct someone to do something differently than what they are doing. You might need to make adjustments for any number of reasons, such as needing to have the work more closely to the vision.

For example imagine on a simple project, as you were tracking, you noticed a software engineer on your team is copying and pasting the function a few times across the codebase. You might want them to not do that because it is inconsistent with how you want the codebase to be. Asking them to do something differently, like putting the function in a common utilities package, is what I call an adjustment.

How you have that conversation will vary dramatically based on the context and the individual software engineer. Maybe the engineer is under a lot of pressure to hit a target date and was planning to raise it to you tech debt and hasn't had the opportunity yet. Maybe the engineer is unaware this is usually a bad practice. How that conversation goes will depend on circumstances like these and many more.

In some cases, adjustments are very easy. In some cases, they can be quite hard. Recall we started this section with the caution that being able to have hard conversations is a touchstone skill for tech leads. Shying away from these hard adjustment conversations will make it hard for you to do well in this fundamental capability and can let problems fester.

Bringing Tracking and Adjusting Together

Tracking and adjusting naturally come together as one activity in practice, of course. Merely by asking someone what they are working on or asking questions designed to verify how well the work is aligned to a vision can feel like an adjustment to the team.

In the example I gave you above, simply asking a question like, “Why are you copying these functions all over the codebase?” will carry the strong implication (almost accusatory in the way I phrased it) that you disapprove of the way the software engineer is doing his or her work. Thus, tracking and adjusting motions happen so closely together that they come together in one capability.

Very importantly, making adjustments is substantially easier if you have developed your listening and vision crafting capabilities. That is, if you understand the individual you are working with well and they feel that you actually understand them and their motivations, then they are naturally going to be more receptive to the adjustments you want to make.

Moreover, and this also important, if the adjustment is based on a clear and compelling vision, then people are generally going to be more receptive to making changes. As you know, in the software industry, there are many opinions of how to get things done, and some opinions are held strongly. Having a unifying, clear, and compelling vision can, in some cases, reduce conflict over opinions.

Maturing in Your Tracking + Adjusting Capability

The maturity scale for tracking and adjusting is a stacking one. Each maturity level builds on the next.

Beginners track and adjust weakly

Beginners at this level are either not proactive about the capability or do it very tentatively. One way to self assess how well you are doing with tracking and adjusting is to simply ask yourself how well you understand the answers the questions I posed above *right now*? If you are not very clear or only know where some team members are, you are probably at a beginner level.

Another telltale sign that you might be at a beginner level with this capability is to consider the extent to which you rely on technology to do the tracking and adjusting. For example, do you rely excessively on technology-based tools, such as Github pull requests, Trello comments, Slack, and so on to keep track of your team?

Those are great tools, but if you feel like you may be hiding behind them (only you will know) because you are concerned about, for example having hard adjustment conversations, then you may want to consider growing in this capability.

At the intermediate skill level, tracking is stronger, but adjustments are still hard

At the **intermediate** level you may be more willing to get up from your desk and engage in face to face conversations fairly often, but you may be still reluctant to engage in some of those difficult conversations that we talked about before. Or, alternatively, you may be making progress with *some* members of your team, but you are still reluctant or wary to check in on other members.

This level is a step up from the beginner level, however, so you are making progress. At least you are flirting with hard conversations or putting yourself in positions where they could come up.

One way to self assess if you are at this level is to think about the tracking conversations. Were there moments where you *wanted* to make an adjustment, but you didn't? Maybe you told yourself you would do it later. Or maybe, you went back to your computer and sent the adjustment message via Slack instead.

At the advanced level, you are fluent with the tracking and adjusting motions

At the **advanced** level you are on your game. You know exactly what is going on and you are frequently adjusting. You are engaging one on one with your team on difficult issues and able to do it in such a way that each team member feels valued and buys into the adjustment you are making.

Wherever you feel you are with this capability, relax a little and believe me when I say this is not an easy level to consistently perform at a high level, so don't worry if you struggle with it. We all do–even experienced leaders.

Capability No. 5: Growth Mindset

This is a short section because I feel I have already given you plenty to work on already. I include it because I want to pique your interest in this area.

Your growth mindset capability is really just the mindset that you bring to the job every day. A strong capability in growth mindset means that you believe, correctly, that all people have the ability to grow in *every area* of their lives, including you.

I emphasize it in the five Pareto capabilities because if you are positively biased on this, it gives you a high angle of trajectory to growing not only your own skills but your team's as well. Conversely, a negative bias on this will change the shape and trajectory of your growth as a tech lead.

I will not go into much detail on this since the self help aisles of book stores are full of titles that can cover the subject far better than I can.

But I do want to really assess your thinking on this point because if you believe, as I think you *should*, that you, your team, your management, your company, even your family members are not static in their capabilities.

Beginners are goal driven

As a tech lead, you are at the **beginner** level if you focus primarily on day to day goals. The goals could come from your management or they could be goals you have articulated for yourself or your team.

There is nothing particularly wrong with this level. Goals are a good thing. They help to make things happen, so relax a little knowing that this is still an impactful level.

The reason that I placed you at the beginner level is because there are more advanced ways to approach progress as a tech lead or a team.

Those at the intermediate level focus on process

The higher level, the **intermediate** level, takes your impact to another level. By focusing on goals, you run the risk of putting the goal ahead of the needs of the individuals on the team and their capabilities to contribute to future goals.

By shifting your focus to the process you and and your team follow to consistently reach goals, you will start operating at a much higher level.

To self assess, ask yourself if you have a process for yourself? Do you have one for the team? Is everyone part of the process?

At the advanced level, you are purpose driven

You are probably a new tech lead, so I will not dive into this too much. I believe that, when you have a strong capability in this area, that you are beginning to see you role as a tech lead as driven more by purpose than your team's tasks or your own growth. If you start exploring this space, you will find that those at the highest levels of leadership are often driven by purpose less than any one team, product, feature, process, or goal.

Wrapping Up

We have covered a lot of ground in a short space tech leads.

We started our time together discussing why this is such an exciting time in your career and life. This is maybe the first time that you are going to start getting results through other people, which requires you to develop a new set of skills and capabilities. You are going to learn a lot about yourself on this journey, so I am excited for you.

I asked you some questions about your intentions for taking the step, but if there is one thing I could tell you before you start as a tech lead is to relax a little and focus on the process of personal growth, not the results–especially in the beginning.

Then we spent some time thinking about the tech lead role. It is an interesting role because it can come in so many shapes and sizes. In some cases, it is a formal one. In probably more cases, it is an informal role. We also spent some time thinking about the common activities that tech leads do, although there is substantial variation among tech leads.

Next, we dove into the five Pareto capabilities: listening, a bias for action, crafting visions, tracking and adjusting, and your growth mindset. Along the way we mused about some interesting topics like the nature of leadership and leadership moments. We explored some practical issues like having hard conversations. I made a big deal about why your capability as a tech lead starts with solid listening skill.

I cannot say enough how excited I am for you to be taking this journey. And I hope this short book will give you some ideas about how to be successful in the role.

Good luck to you tech lead! Let me know how it goes!

For More

Please do not take this advice as the final word on what it takes to be a successful tech lead. Keep growing and keep searching for guidance and advice that works for you. There are many more sources, including my own evolving thoughts on the role

More From the Author

First, there is more work that I produce on the topic with more thinking than I wanted to include in this book.

For example, I podcast daily on the tech lead role on behalf of the Tech Lead Coaching Network. Search for Tech Lead Coaching Network on all the major podcasting platforms including Apple Podcasts.

In some cases I also do personal coaching with tech leads or CTOs of startups, which have very similar needs to tech leads. Sometimes just reading a book is not as effective as we wish it were. If you want to explore this with me, reach out to me at coaching@michaelrice.com. I would love to know how your journey is going and if there is anything I can do to help.

Other Recommendations

Talking With Tech Leads by Patrick Kua, which I cited in this book, contains both useful interviews with tech leads as well as some useful information similar to what I have written here. Different, but similar.

CTOCraft is also great source of information, and a good network for CTOs to consider joining.

Tech Managers Weekly is an email list containing some information that may be useful to you as well. Most of it is aimed at managers.

Methods and Tools is a relatively “off the radar” mailing list that has some useful information.

Getting Help

I hope this book proves to be a useful guide for getting started. But it is only a start. As you go through your days, you are going to have some ups and some downs. You won't always remember the book or the guidance. Or if you do, you might not be clear on how to apply it to your situation.

So, if you can, I hope you can find a mentor or coach. Ask around your company, or you own personal network. You may be able to find a supportive manager or experienced tech lead to help guide you through the process. That person could be inside your organization or outside. Some people have had some success reaching out to others directly on LinkedIn.

If you cannot locate anyone there, or maybe in addition to that, I created the volunteer-based Tech Lead Coaching Network as one resource you can turn to for help.

As I mentioned earlier, I do some private coaching as well in some cases–if you cannot find any help from your own network or the Tech Lead Coaching Network.

Why Should You Consider Coaching?

To explain the value of coaching for your tech leadership, let us start with a process that may be familiar: programming. Programming is a skill. To be a programmer, you have to learn the nuances of a programming language. Maybe you took classes on the subject. Maybe you are self taught. But learning to write code and get it to compile and perform correctly is a well travelled journey you can go on.

The journey never really ends, however, because good programmers continue to invest in their craft. They try new ways of writing new code. They read other people's code. They learn new skills and techniques from books, other software engineers, and sites like Stackoverflow. And they improve over time. Between the compiler's feedback, the unit tests' results, and the internet, the feedback loop for programmers is pretty fast these days.

Now contrast those fast feedback loops with leading people. When working with people, things take time. Sometimes you get negative feedback. Sometimes you get feedback when you didn't want it. Sometimes you get no feedback at all. It is hard to know what worked and what did not.

Moreover, there are relatively few ways to Google your way to optimize or debug your tech leadership. While there are plenty of websites and forums where you may be able to go for some ideas, individuals are as unique and complicated as their fingerprints. They get even more complex when they are part of a team.

Coaching is a way to get more and faster feedback as you grow your tech leadership skills. Just as programmers learn that there is nothing magical about code, there is similarly nothing magical about leadership. It is simply a set of behaviors and skills that need to be developed and practiced.

What is the Tech Lead Coaching Network?

Our view of tech lead coaching, at least in our community, is that it should be a peer based relationship and series of conversations where you work with another tech lead to develop your leadership skills.

Before exploring it further, let us begin by explaining what tech lead coaching from the Network is not. First, it is not training. Tech leads who coach other tech leads do not need to be experts at the role nor do they need to be good at training.

It is also not some form of counseling or therapy. Coaching conversations should not explore the distant past in too much depth nor should they explore the faraway future. Put differently, the relationship is not intended to be a series of extended conversations about where your life is going or how it got here.

Finally, and most importantly, coaching conversations are not intended to be sessions where a coach diagnoses and fixes problems for you. Only you can do that for yourself. It is natural for coaches to want to dive into problems and for tech leads to ask for that kind of advice, but really, that is not the goal of tech lead coaching sessions.

Now let us explore what tech lead coaching should be. Tech lead coaching should start with a one to one relationship between two tech leads. The coach should have more experience in the role than the individual being coached, but it doesn't need to be a substantially lot more experience. Some coaches are already good at it, and can draw on existing coaching skill as opposed to a large body of experience in the role.

The relationship should lead to a series of periodic conversations. The conversations can be in person, on the phone, or via a video conference service. The conversations can be as frequent as is convenient for you and your coach.

There should be some structure to each conversation and a process to the series of meetings. The structure and process is intended to make sure that you walk away from each conversation with something actionable and that the overall series of conversations should help you tech lead skills and confidence grow.

Of course, there should also be considerable flexibility in the structure and process so that the coach and the tech lead can form a relationship, adapt to the moment, and not feel artificial or stilted.

Finally, the conversations should be focused on what's happening in the immediate future. As mentioned above, the conversations are not meant to be a long rumination about the past or explorations of what the future could hold. Using the conversation structure above, it should focus on what's happening in you immediate environment and project and specific steps that can be taken.

Please consider joining us at techleadcoaching.com. It is, and always will be, completely free, volunteer-based, and open sourced.