

# **Team First: Lessons of a Tech Lead**

**By Kevin Meredith**

For my family

## Table of Contents

- Introduction
- Ch 1 - Zoom Out
- Ch 2 - Take Care of the Team
- Ch 3 - Sleep in the Same Quarters as the Troops
- Ch 4 - Keep the High Ground
- Ch 5 - Sports Team

# Introduction

## Background

My name is Kevin Meredith. I began writing software professionally in 2009. For a few years I treated software development entirely as a job. It wasn't until I took a graduate class on the Scala programming language that I craved to learn more and continuously improve. Once I attended the 2013 NEScala Conference at Drexel University, I fell in love with functional programming. Since 2013, I've worked as a Software Engineer on a full-time and contract basis across Medical Insurance, Logistics, Cruiseline, Online Banking, and Gambling industries. I wrote "A Little Book on http4s," a book on the Scala HTTP library. I've worked as a Software Engineer, and then Technical Lead, at two separate companies.

In late 2020 I learned about Stoicism through Ryan Holiday's podcast, "Daily Stoic". It's a practical life philosophy that aims to make its practitioners live fuller, better lives. From Stoicism, I learned about Robert Greene's books, namely, "The 48 Laws of Power" and "The Laws of Human Nature." Right before learning about Stoicism, I met Jocko Willink's book, "Extreme Ownership." The aforementioned authors and their books have influenced my own life greatly.

## What is a Tech Lead?

The book, "Art of Scalability, The: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise," defines leadership as:

‘influencing the behavior of an organization or a person to accomplish a specific objective.’

A Technical Lead's role is to apply leadership in a technical capacity. The author, Patrick Lencioni, perfectly describes the criticality of teamwork in the beginning of his excellent book, "The Five Dysfunctions of a Team:"

**Not finance. Not strategy. Not technology.**

**It is teamwork that remains the ultimate competitive advantage, both because it is so powerful and so rare.**

**If you could get all the people in an organization rowing in the same direction, you could dominate any industry, in any market, against any competition.**

In this book, I share the principles that I believe are fundamental for a Tech Lead to practice.

## Why I Wrote This Book

I decided to write this book based on my own personal experiences as a Technical Lead at two companies. I aim to apply the principles that I've applied and learned from the aforementioned excellent authors into this book.

I hope to contribute to the leadership development of existing and to-be Technical Leaders. Lastly, I hope to receive feedback, critical and positive, on this book.

“Win or learn,” not “Win or Lose,” as Jocko has taught me!

## Ch 1. Zoom Out

In this scenario, Tech Lead and Bob the Intern work on the same team, who's responsible for developing back-end web services. Your author is the Tech Lead in this example. The following conversation took place in Slack in a channel devoted to an important feature. The Vice President of Engineering regularly posted her opinions to questions and comments.

Tech Lead: OK, Bob the Intern \*, all of the feature's server-side changes have been deployed to User Acceptance Test (UAT)\*\*. Let's coordinate with iOS and Android on testing. I suggest they test the changes on their own, and then let us and the Product Team know their results.

Bob the Intern: Come to think of it, I'm not so sure asynchronous is the right way to go. There's many tests to run and our user journeys are complex. How about we schedule 30 minutes with the Mobile Engineers and Product Owner to make sure Product signs off?

Tech Lead: I agree, Bob. You've laid out strong arguments for why we should choose synchronous over asynchronous. I'll set up that meeting for later this afternoon. Thanks!

Let's examine the above conversation. I, the Tech Lead, had proposed an idea on how to test out a major feature. My teammate, Bob the Intern, disagreed, laying out a strong argument for his idea.

My initial, fleeting, ego-driven thought to Bob's response was:

What does Bob know about Software Development? He's been working in the field for a few months. I've got ten years of experience. Where does Bob get the nerve?

Luckily, I realized this nonsense was driven by my Ego. Taking a breath, I zoomed out and looked at both options objectively: testing synchronously or asynchronously, including pros and cons of each. Looking at them, the choice was clear: asynchronous. Then I responded in Slack as I had above, applauding Bob's insight.

Remember that leadership is defined as "influencing the behavior." Is it in the best interest of the company for decisions to be made based on individuals' Egos or an objective analysis of pros/cons? Which decisions are likely to be better: those driven entirely by Ego or ones made based on objective analyses of facts? Which decisions are likely to produce higher revenues and profits for the company: those made due to Ego or those driven by the truth alone?

Clearly these questions' answers are self-evident. As a Tech Lead, your duty is to influence the company in making technical decisions that are the best. When evaluating multiple options, clearly and calmly evaluate their pros and cons, and encourage technical debate with your teammates. Failing to uphold this fair

assessment will result in lesser choices for the company. That, ultimately, will result in fewer profits and lower return on investment to shareholders.

In short, given the choice between winning and losing, which will you pick?

\* Certainly I didn't address this person as "Bob the Intern," but the person's title helps towards demonstrating the spirit of this chapter.

\*\* UAT is a deployment environment that precedes Production.

## Ch 2. Take Care of the Team

During one part of my career, I worked on a team with a particularly active on-call rotation. By “active,” I mean an on-call rotation where the FireFigher could get paged twenty times on a bad weekend. This situation was compounded by a short rotation consisting of six engineers. As a result, every engineer “held the pager” every six weeks for an entire week. The root cause of the active on-call was a flaky third-party integration that was critical to the company. Multiple engineers on the team had raised this problem to the manager, letting them know the company ran the risk of burning out the entire team.

A new teammate, an intern turned Full-Time, Sammy, had joined the team a few months ago. Given Sammy’s experience and the need for more people to join the on-call, Sammy was enlisted in the rotation. Although Sammy was new, she was dedicated to executing on her tasks and producing results.

During one weekend, Sammy received thirty pages on a Saturday. As the Tech Lead, I had checked in our #on-call Slack Channel every few hours given that Sammy was new to the rotation - to see how they were doing. To my dismay, I observed the high number of pages that must’ve been ruining Sammy’s day. I sent a Direct Message to Sammy around 10 PM on Saturday:

Me: Hi Sammy - I've seen the number of pages fired today. How are you doing?

Sammy: I'm OK. Yes, it's been a rough day.

Me: You must be exhausted. Please let me take over the page now and you can pick it back up on Monday morning. Please get some rest on Sunday.

Sammy: OK - thank you!

After I concluded the message with Sammy, I sent an email to my manager:

Dear Bill -

Sammy received thirty pages today. All of the incidents appear to be related to our flaky integration. I've been in her shoes and know how nerve-wracking and unpleasant a thirty-page day is. I think it's fair that Sammy be granted an additional day of Paid Time Off (PTO) given that her entire Saturday was consumed by answering PagerDuty. Can you please approve of that?

I know the team has kicked around a few ideas on how to apply a band-aid fix to this problem -in the name of improving the on-call experience of an Engineer. I personally think our team will burn out given the frequency of pages, as well as the demoralizing effect of not being able to address the root cause.

I set up a meeting on Monday to discuss a few solutions to improve this dire situation, please.



Thanks,  
Kevin

### Ch 3. Sleep in the Same Quarters as the Troops

I don't recall where I had read or heard this story. It goes something like:

During a war, soldiers were sleeping in tents outside in the elements. Their leader slept indoors in a building with air conditioning and a comfortable bed. When trying to rally the troops to boost their morale, the leader said, "I know we've had to endure the hardship of this camp's tough conditions." At he went on, the troops glanced at one another in disbelief. How could their leader relate to their conditions as he slept in a comfortable bed indoors while they, each night, faced Nature in tents alone?

From Ch 2, I mentioned that I had led a team of engineers on an exhausting on-call rotation. That four-man, 5-week rotation looked like:

Primary: Me / Sally / Me / Sam / Bill  
Secondary: Sally / Me / Sam / Bill / Me

Read it as, for week 1, "Me" was the primary, and "Sally" was the secondary. Then, at the start of week 2, "Sally" took on the primary, and "Me" was secondary.

Over the five-week rotation, I took two shifts, whereas the remaining three team members took one. I took it upon myself to bear more of the burden as the leader. It was my duty to my fellow troops.

Steven Pressfield's book, "The Warrior Ethos," perfectly captures the epitome of a leader in this story of Alexander the Great:

Once, Alexander was leading his army through a waterless desert. The column was strung out for miles, with men and horses suffering terribly from thirst.

Suddenly, a detachment of scouts came galloping back to the king. They had found a small spring and had managed to fill up a helmet with water. They rushed to Alexander and presented this to him. The army held in place, watching. Every man's eye was fixed upon his commander. Alexander thanked his scouts for bringing him this gift, then, without touching a drop, he lifted the helmet and poured the precious liquid onto the sand.

At once, a great cheer ascended, rolling like thunder from one end of the column to the other. A man was heard to say, "With a king like this to lead us, no force on earth can stand against us."

Who could actually compete with an organization with such a leader?

## Ch 4. Keep the High Ground

In one of Jocko Willink's podcast, he discusses the concept of "high ground" from "About Face: The Odyssey of an American Warrior" by David Hackworth.

It's a critical concept for leadership and human development. To me it means taking the path of fairness and truth. The "high ground" is unassailable. "High ground" is what professionals take. Anything else is for the "amateur," as Steven Pressfield (\*) might say. The "high ground" is true leadership.

Consider the following exchange I had with one of my teammates. In this example, Phil and I work on Team Alpha. Phil had opened a pull request against one of Team Foobar's repositories. He encountered resistance from Team Foobar that made Phil bristle.

Me: Hey Phil. I know you were having some problems with Team Foobar on your pull request. How's it going?

Phil: It's rough. The quality of their code-base is low. They made completely unreasonable comments on my pull request. It's making me upset and I don't even want to bother to respond.

Me: I'm sorry to hear that. Can you give me an example?

Phil: They told me to use the existing library that's used throughout the whole code-base. But their open-source library hasn't been updated in 2 years. I introduced a new library that's been downloaded a million times a week and it's used by big, mature tech companies.

Me: I can see how that would be frustrating. How did you respond to that comment?

Phil: Oh - I haven't responded yet and don't intend to. It's a waste of my time to respond to those people.

Me: What do you think about responding professionally to their comments?

Phil: Really?

Me: Let's look at a few ways to respond to the PR comment and the various outcomes.

- (1) Answer professionally on why you made that choice. Include your rationale in a straightforward, respectful way. Mention that you care about the quality of the code-base. Choosing a battle-tested library supports the business by standing on the shoulders of a highly tested library.
- (2) Dismiss or ignore their comments

How do you think the comment's author would respond in each case? In the first, the best case is that you become engaged in a thought-provoking, data-driven debate. In the second response, it's most likely going to ultimately extend the amount of time required before merging it. Secondly, and perhaps most importantly, your relationship with this person, as well as Team Foobar, will suffer.

Since our team's road-map requires building more changes to this code-base,

it's our team's (and the company's shareholders') interest that our team has a good relationship with Foobar. Otherwise, code reviews will drag on, mean and bitter comments will dominate, and the team will ultimately miss its deadlines. Our competitors will delight in this outcome.

Phil: If I respond professionally to their comments, I doubt that Team Foobar will engage me seriously in a fair "data-driven" debate. On one of my code changes, their teammate, Bob, had commented, "Why did you do this? (thumbs down emoji)"

How am I supposed to respond to someone so rude?

Me: I agree with you that such a comment is unprofessional and unhelpful. But, if we respond professionally to try to extract a meaningful question from Bob's comment, we keep the high ground.

Phil: Bob's just going to respond with another rude comment to my professional response.

Me: If one of Team Foobar's teammates respond disrespectfully to your honest question, the high ground protects us.

Phil: High ground? I don't understand.

Me: Let's say Bob continues to respond rudely to your honest, professional responses - and thus blocking your progress. At a certain point, we'll need to escalate this blocker up the chain of command. Whom will Management find at fault for your stalled Pull Request: Team Alpha, whose code author, you, responds calmly and professionally, or Team Foobar, whose engineer is acting unprofessionally and rudely?

Phil: Yeah - I see your point. Let me keep the high ground.

Me: Thanks, Phil. Let me know if I can help at all. To say more, the purpose of the High Ground isn't to win arguments or escape blame. The value of the High Ground is the inherent value in practicing it, as well as its influence on others. At the end of the day, our company is going to be stronger and better if Teams Alpha and Foobar work together. We're on the same team. Our competitors, the enemy, relish our in-fighting.

When I think of the High Ground, "The 48 Laws of Power"'s (\*\*) story on Talleyrand immediately comes to mind. It adds a further dimension to the High Ground - keeping cool and thinking clearly under pressure.

The story goes something like this:

Talleyrand was a close advisor to Napoleon. Napoleon learned that Talleyrand had effectively betrayed him. Napoleon, red with rage, screamed and shouted at Talleyrand in front of his other advisors. Throughout the entire outburst, Talleyrand maintained his calmness and composure - almost aloof to the situation.

When we encounter pressure or petty attacks, look to Talleyrand. Talleyrand saw an imminent death sentence, but kept the High Ground. We can do it too. It will make us better, as well as our team via its magnetic influence.

`\*` - Read "The War of Art" and "Turning Pro" from Mr. Pressfield

on these excellent works.

`\*\*` - "The 48 Laws of Power" by Robert Greene - excellent book  
on human nature and power.

## Ch 5. Sports Team

Performance Management consists of observing an employee's contributions and behaviors, as well as taking actions to affirm, better understand, improve or remove the employee. This function is critical to the success of a "Sports Team" company.

In late 2022, Peloton chose a new CEO, Barry McCarthy. He provides a clear explanation of a "Sports Team" company in the New York Times (\*):

We're a sports team, and we're trying to win the Super Bowl. And so we're going to put the best players on the field we can. And if you go down the field, and we throw you the ball, and you drop it a bunch, we're going to cut you. Because everybody else who's trying hard to win the game deserves to have the best players on the field. And if you're a good player, you're going to love being on this team.

Typically a Technical Lead does not have the direct authority to grant raises or fire people. However, it is their primary purpose to help the team to win. As a result, the Technical Lead possesses indirect authority towards Performance Management.

In an organization consisting of a Manager with six to ten direct reports, spanning two separate development teams, the Manager usually knows if someone is performing well or failing miserably. However, reaching the "failing miserably" point takes considerable, precious time. It's the role of the Technical Lead to identify and contribute to the resolution of performance problems promptly.

Napoleon Bonaparte remarked:

I may lose a battle, but never a minute.

Depending on how close the Manager is to their team, the Technical Lead must have a closer gauge of each of their teammate's progress given they're more involved with the technical details, such as code and architectures. As a result, they must be quick to applaud teammates doing a solid job, as well as step in early to help teammates who are struggling. Ultimately, the Technical Lead must speak up to their Manager and/or decision-making powers if the struggling teammate fails to improve.

Mr. McCarthy's earlier quote clearly explains the need for a high-performing team: to win. Another important aspect of keeping standards high is morale. In my experience, I observed a low-performing teammate sink the morale of one of my team's top performers. Essentially a new Engineer, Jason, joined my team with the title, "Staff Software Engineer." This person's job was to, primarily, coach other Engineers and host weekly training sessions. Management decided that these "Staff" roles should directly contribute technically, so this person got assigned to join my team. Jason's first task was to pair with a high performing "Senior Software Engineer," Bob, on my team on a larger project. After one-and-a-half weeks, I had my regularly scheduled 1-1 with Bob.

Me: Hey Bob - how's everything going with Jason?  
Bob: Terrible. He's not opened a Pull Request or made any visible progress.  
Me: That's not good. Has he sought any advice from you or asked any questions?  
Bob: None. He's slowing me down since I'm dependent on his task for closing out this project. If he doesn't show any progress by end of this week, I'm going to do the work myself.  
Me: I understand, Bob. Let me talk to Jason to see if I can get to the bottom of his zero progress. Let's discuss this further at our next 1-1.

At my next weekly 1-1 with my Manager, I raised the concern of Jason's poor performance. Unfortunately, Jason continued to struggle and not deliver results. At my next few 1-1's with my Manager, I let my Manager know of Jason's lack of progress.

Personally, I'll confide that I was frustrated that my Manager did not take faster action to eliminate Jason from the team. From my perspective, as a Technical Lead, Jason was getting paid more than I was, yet was not contributing to the team in any way. As a result, the team's morale was sinking, especially amongst a top performer. However, I kept my thoughts to myself and never shared my frustration with my Manager. Remember that the Manager has the power to hire and fire. The Technical Lead has influence, but not the direct authority. Look to Robert Greene's first law of power in his excellent book, "The 48 Laws of Power," as to why I did not share my thoughts:

Never outshine the master

Sharing my true thoughts would've put me at odds with my own Manager. Revealing those thoughts would deliver an implicit criticism: "I know better than you." As a Tech Lead, help the team, but know and respect boundaries of your Manager.

\* - <https://www.nytimes.com/2022/02/19/business/dealbook/barry-mccarthy-interview-peloton.html>