

THINK LIKE A

# Software Engineering Manager



Akanksha Gupta

MEAP



MANNING

# **Think Like a Software Engineering Manager**

## **MEAP V07**

1. [1 The engineering manager role](#)
2. [2 Individual contributor to engineering manager](#)
3. [3 Managing people, teams, and yourself](#)
4. [4 Managing performance](#)
5. [5 Delegation - Learn to let go](#)
6. [6 Rewards and recognition](#)
7. [7 Hiring](#)
8. [8 Handling attrition](#)
9. [9 Working with cross-functional partners](#)
10. [10 Project management, execution, and delivery](#)
11. [11 Managing expectations](#)
12. [12 DevOps and operational excellence](#)
13. [13 Organizational change management](#)
14. [14 Time management](#)
15. [15 Beyond this- grow yourself](#)

# 1 The engineering manager role

“Productivity is most important by engineering management rules, but enjoyment is most important for engineers. One stems from the other.”

~ [Rob Pike](#)

## This chapter covers

- The basics of management and the need for it
- Key roles and responsibilities of an engineering manager
- The nuances between leadership versus management
- Traits that differentiate the excellent engineering managers
- Success metrics
- Leadership styles

Let's address the elephant in the room, what exactly is an engineering manager (EM), and why do we need one? Management is a core element that, irrespective of the industry, has some underlying basic concepts that positively impact company growth and employee satisfaction. It involves ensuring the well-being of your staff and aiding the business in achieving its objectives. This entails overseeing your employees, addressing their immediate and long-term needs, offering constructive feedback, facilitating conflict resolution, harnessing their potential to drive business goals, and strategizing for overall company and business expansion. Moreover, you are the crucial link between employees and senior leadership, bridging the gap between these entities. Management is also making hard choices, saying no at times, giving difficult feedback, removing blockers, and much more. The overall objective is to empower employees by providing the necessary support for success in their roles while adhering to company policies and regulations.

## 1.1 Management 101

In a pizza restaurant, there are ten chefs capable of making four pizzas each per hour. In an 8-hour shift (excluding breaks), the restaurant produces 320 pizzas. As the newly hired manager, you observe inefficiencies and identify workflow improvement opportunities. For instance, you streamline the dough preparation process and consolidate similar pizza orders, assigning them to a single chef to expedite production. Additionally, you allocate specific areas of responsibility to each chef during their shift, such as Charlie operating the dough machine and Frank managing the pizzas in the brick oven.

The team's productivity improves under your supervision. Each chef is now able to prepare six pizzas per hour. As a manager, you introduce a monthly award program to recognize exceptional work based on customer feedback. The chefs also receive regular training on the latest techniques. These efforts boost morale and acknowledge the chefs' outstanding contributions.

You also promote effective communication within the team, encouraging everyone to voice their opinions and ensuring they feel heard. Recognizing each employee's strengths, you create opportunities for them to enhance their potential. As a result of these enhancements, the team's overall output increases to 480 pizzas per day, representing a remarkable 50% boost in productivity. This example demonstrates the value of having a manager in enhancing individual performance, fostering a happier work environment, and contributing to increased productivity. This managerial role extends beyond the pizza industry and finds relevance in various sectors such as pharmaceuticals, technology, customer care, academia, and more.

Effective employee management is crucial for company success and employee growth. It involves various responsibilities such as coaching, motivating, hiring, firing, delegating, and project execution. Whether managing employees in a warehouse or as a mid-level manager in a technology company, three key pillars define the manager's role: **people, projects/products, and processes**. This book is structured into three core sections to delve into the nuances of engineering management, focusing on managing software engineers and addressing their specific needs.

For instance, delegation is an essential skill for all managers, but its application differs between managing a warehouse employee and a software engineer. While the core principles remain the same - identifying the right individual, providing necessary support, and measuring success - the execution of delegation varies between the two scenarios. There are several resources already available that talk about strategy making, so I will not repeat what is already out there. Instead, I will focus on the **concept of management concerning software engineers and how engineering management plays an essential part in the success of companies in the tech industry**. Regardless of your role as a manager, you can grasp the foundational principles of management presented in this book and draw inspiration from strategies for effectively overseeing your employees, projects, and time.

## 1.2 Demystifying the engineering manager role

We all have heard the word- Engineering manager or engineering leader. **But what does it bring to the table?** Engineering management is a multifaceted role that combines both art and skill. It is a complex title, as expectations and responsibilities can differ from one company to another. To simplify, an engineering manager can be seen as the fusion of an engineer and a manager, bringing together technical expertise and leadership capabilities. To put this simply: **Engineering manager = Engineer + Manager.**

As an EM, your role involves overseeing the engineering team, which includes software engineers, and occasionally collaborating with cross-functional partners. Companies may use different titles that have equivalent responsibilities. Common titles I've come across include engineering manager, software development manager, senior manager of software engineering, manager of application development, SRE/DevOps manager, and more. While companies may have different titles and not all include "engineering," the role's responsibilities and expectations can remain unchanged.

Establishing a new company, particularly in the tech sector, requires the harmonious functioning of various components. This demands robust and continuous leadership across diverse domains, including determining project goals, crafting execution strategies, financial management, and more. During a startup's initial phases, an individual often juggles multiple responsibilities of engineering management. This setup arises because the company's early focus is typically concentrated on a specific task set manageable by a single person. In such scenarios, the position of an engineering manager might not be formally present, and engineers might report directly to the Chief Technology Officer (CTO). However, this doesn't invalidate the necessity of an engineering manager; instead, the CTO takes on added responsibilities as an EM. As the company expands, this approach becomes unviable. In a medium-sized enterprise, it's customary to have multiple EMs, and the career progression path becomes more distinct, outlining clear expectations for each role. Why not construct a company entirely comprising engineers without EMs? As earlier mentioned, a prosperous and growing business functions akin to a well-coordinated mechanism. Absence of an EM would likely lead to the sudden emergence of friction points within this well-coordinated mechanism, significantly compromising the overall health of the company. However, having an EM becomes crucial when the company needs to function smoothly.

- A set of engineers thinks one project is more important to do than the other.
- Resourcing for specific projects is not appropriate
- Conflicts brewing within or across different teams
- The productivity and morale of the group are down

Not all Individual Contributors (ICs) are suited or interested in transitioning to EMs. Successful EMs possess qualities that enable them to manage at all levels, demonstrate empathy, possess soft skills, build relationships, plan and execute effectively, and, most importantly, have a genuine interest in the role. These skills are typically acquired and refined through experience, particularly in handling ambiguous situations. It's worth noting that a highly accomplished IC may not necessarily excel as an EM and vice versa. Thus,

pursuing a career progression or change doesn't necessarily have to lead to becoming an EM. There are alternative paths to consider, such as remaining as an IC (e.g., as a solution architect) and focusing entirely on technical expertise or exploring parallel roles like product/project managers, business analysts, and DevOps architects. There is no universal model, and it is perfectly acceptable to be a staff or principal engineer or lead architect while still taking on some management responsibilities. In Chapter 2, we will delve into the career trajectories of ICs and EMs and explore commonly used titles associated with these roles. In the next section, I will explain the EM role and understand the nuances that come with it.

### **1.2.1 Roles and responsibilities: the core competencies**

What does an EM do? The answer isn't a simple one. Engineering management encompasses a diverse set of responsibilities that require individuals to juggle multiple roles simultaneously and excel at context-switching. At the same time, one needs to be aware of the perils of unwanted context switching. In this section, we will explore some of the fundamental competencies that define an effective EM.

#### **Build talent**

Building talent is one of the fundamental responsibilities of an engineering manager. This involves:

- First and foremost is to understand the roles and responsibilities of people needed in the team. This will help better plan the team composition
- resume screening, hiring new talent for the needs of your team and/or company, and providing them a runway for success.
- with regular one on ones and career conversations for engineers(or managers if you are a middle manager), understanding their motivations, aspirations, and blockers
- helping coach and mentor by addressing career gaps, bringing the best out of people, helping build their careers, and promoting them when

ready

- managing performance by helping both high and low performers and providing constructive feedback
- growing people from within the organization and knowing when to hire externally
- retaining existing talent by keeping them challenged and motivated
- build high-performing, self-organizing teams
- demonstrating empathy, honesty, and emotional intelligence

The goal is to empower others and unleash their full potential to accomplish the work. It is essential to prioritize your team's experience, as it fosters engagement and job satisfaction. When the company cultivates a strong people-centric culture and nurtures talented individuals, it builds a solid foundation for success.

## **Visionary**

Having a clear vision significantly enhances people's productivity. As an EM, it is your responsibility to shoulder the task of imparting a clear vision and mission statement to your team by translating organization-wide strategy and vision to the team, ensuring they comprehend the purpose of their existence as a cohesive unit. This entails providing your team with explicit principles and a charter they adhere to while understanding how their contributions align with the organization's broader goals. Providing clarity will involve the following:

- Clarity in terms of the team's mission and charter
- How the team goals fit the bigger picture and company objectives and key results(OKRs)
- Staffing decisions
- Priority of tasks so that engineers have clarity on what to work on next
- Think big and share the short and long-term vision with the team
- Help remove roadblocks for the team so they can be effective, productive and make progress on tasks. This involves helping with any blind spots that others might have missed. The EM can use their

50,000-foot view and years of experience to clear paths for the engineers.

As an EM, you need to be authentic, honest and demonstrate integrity. You will have access to more information than ever, hence in the role of a visionary, you need to learn how to communicate the message in a manner that is healthy and promotes transparency.

## **Lead projects**

Engineering managers are no exceptions when it comes to leading projects. You will be involved from the initiation of the project to the end. You are responsible to:

- Inspire your team of engineers and cross-functional partners to lead projects to the terminal line in a smooth and timely manner
- This involves making staffing decisions keeping engineers' career aspirations in mind, creating a project roadmap, running project standups, budgeting, stakeholder alignment
- Helping run smooth agile ceremonies in collaboration with the scrum lead(if the role exists) and product manager
- Communicating with other managers, C-level executives
- Wearing the product hat as you actively participate in shaping the product
- Presenting to stakeholders teams' vision and how they translate into products for end customers
- Foster a culture of innovation and collaboration
- Help with conflict resolution within and outside the team
- Coding will take a backseat. Instead, your focus will be on the delivery and execution of the projects with technical excellence in mind. You still are expected to share your experiences with writing high-quality code and the importance of refactoring
- Thinking about operational stability once the product makes it to production

This journey will be full of bumps and ambiguity where sometimes you must step up, make decisions with limited knowledge and resources, and own the outcomes. As an EM, you are supposed to know when to delegate and avoid ambiguity. It is the EM's responsibility to ensure decisions are made, sometimes it means they are taking them. It is better to make a bad decision than to take none (and fall into the analysis paralysis trap). That said, if that decision goes wrong, it provides valuable learnings and feedback that can be used to improve future decisions. Often decisions can be revisited and adapted to changing conditions. It is important to take ownership when things go well and more when they go wrong. Some managers hide their mistakes and blame their team, for example, for not understanding what the EM said.

**NOTE:**

The “[Extreme Ownership](https://www.amazon.com/Extreme-Ownership-audiobook/dp/B015TM0RM4/)” (<https://www.amazon.com/Extreme-Ownership-audiobook/dp/B015TM0RM4/>) book might be an interesting read to learn more about ownership.

As EM, use your best judgment based on experience and skills. Mistakes can prove more costly than before, and you will be responsible for your work and your team’s work. Hence it is of utmost importance to keep a problem-solving mindset when leading projects and help cut ambiguity.

## **Effective delegation**

Can you manage multiple projects simultaneously, conducting career discussions with engineers, aligning stakeholders, and attending meetings on your own? The honest answer is no. It is essential to be pragmatic about your time and responsibilities and master the skill of delegation to expand your capacity. This entails:

- Get away from the ‘doer’ mentality and adopt a ‘trust and verify’ mindset.
- Having a coaching mindset

- Share knowledge to have a multiplier effect
- Create a psychologically safe environment for everyone on the team
- Being organized to manage time well and prioritize tasks effectively
- Be smart about identifying what can be delegated, when it can be delegated, to whom, and how it should be delegated to be an effective delegation

At the same time, this will help boost morale and build trust with your engineers as you provide them with opportunities to grow, keeping their career aspirations in mind. At the same time, you also free yourself to pick up other work that will make you advance in your own career. Remember to think about scaling to manage multiple teams at once.

## **Focus on engineering excellence**

If you are satisfied with where you are, you will not continue to learn and grow. As an EM, the first step is to understand the existing process and invest time to understand the constraints that led to the current state. Next, you can strive to foster a culture of process improvements, business and operational excellence. Apart from delivering on engineering projects, this often includes improving service performance, infrastructure costs, reducing downtime, faster delivery, adding more value to customers, and more. You cannot do everything alone, so bring:

- Awareness about tech debt and get others in the team inspired to raise their own bar
- this can be in the form of better documentation of on-call runbooks, setting up guard rails on change Management, having standard operating procedures(SOPs) for the processes, and so on
- Help raise the team's operational excellence bar to scale for future
- Help review code reviews, actively participate in design discussions, hunt down bugs, and ensure team services are legally compliant

These measures will guarantee that what your team implements today is designed for long-term optimization and scalability. With greater responsibility comes a greater role. Now that we understand the core

responsibilities of an EM, let's look at the nuances between leadership and management. Understanding the differences helps you build a mindset of always keeping the two facets of the EM role in the forefront rather than thinking about it as an afterthought.

### 1.2.2 Leadership versus Management

Sometimes "management" and "leadership" are used interchangeably, but does that mean they are identical? While the title or role may be referred to as an EM or engineering leader, the actual practice should embody engineering leadership. Holding a managerial position to be a leader is optional, and vice versa. However, if you approach managing your team with an **engineering leadership** mindset, you increase the likelihood of achieving success for your team and yourself. The most effective managers are those who combine managerial skills with leadership qualities. Let's examine the table below to highlight some key distinctions.

**Table 1.1 Differences between leadership and management**

Category	Management	Leadership
Role	It is more of a title or role. More of a tactical role	It is more of a mindset, skill, and approach to things. More of a strategic role
Duties	Management refers to the more operational side of things. You will help with the running of sprint ceremonies, project deliveries, and more	Conversely, leadership is a superset of operational plus the emotional aspect of doing things. This means you will help inspire and coach your team in the process

Outlook	Management is more in terms of the current situation and present in mind	Leadership is providing vision keeping the big picture in mind, keeping the present and future in mind
Value	Management is about increasing the developer velocity and striving for stability, focused on quantifying the value of the team	Leadership is trying new things and challenging the status quo, thinking out of the box, which is creating value rather than 'just' delivering
Culture	Endorse the culture of the company	Helps shape the culture of the company

The differences between the two seem very nuanced, but in the larger scheme of things often differentiate a successful EM from its counterpart. An engineering leader focuses on team management and provides vision and guidance to help create their next success story. They tie all loose ends of different pieces together and look at the problems more holistically rather than focusing on a specific aspect. But how do you identify important features of a strong leader? Let's look at some traits that make a good leader stand out.

### 1.2.3 Traits of a good engineering manager

An EM is like the chief of the military. Your traits can make or break the team and projects around you. So what are the key traits and qualities of a good EM? We need to answer this to understand what it takes to be successful and satisfied in the role. Below, I will share some traits that help a good EM to stand out from the crowd. Let's also follow the journey of two EMs(Alice and Bob) who show opposite traits and see how it affects certain situations.

## Caring for their people

The primary role of an EM is to support and engage with their team members. This involves various aspects such as facilitating one-on-one discussions, conducting career conversations and performance reviews, providing guidance in navigating ambiguity, and offering emotional support during workplace or personal challenges. A caring EM fosters trust and respect while effectively motivating and challenging team members. They prioritize honesty and authenticity in their interactions. Additionally, they contribute to team building activities to boost morale and act as a patient listener and problem solver for their team. A fair and unbiased mindset is important, ensuring equal opportunities for all team members and providing constructive feedback to aid their growth. Like family and friends provide continuous feedback for personal development, an EM plays a similar role in coaching and mentoring their team members. Constructive feedback is crucial for ongoing improvement and success. **While debugging code is relatively straightforward, managing individuals and projects requires more effort, as each person is unique.** The EM's responsibility is to protect and support their team, acting as a leader and ensuring their satisfaction. An EM is akin to a gardener nurturing the growth and development of their team members' careers. Moreover, successful EMs can manage at all levels, which builds trust and loyalty among team members, even in challenging situations.

Consider the following scenario: A recently onboarded engineer named Claire deploys a new code that unintentionally introduces a bug, leading to a temporary issue. Claire had thoroughly tested, but unforeseen circumstances resulted in an unexpected failure. In response, two EMs, Alice and Bob, demonstrate contrasting reactions.

Alice takes a proactive approach. Upon discovering the problem, she swiftly seeks a resolution to address the immediate issue. Subsequently, she delves into identifying the root cause of the problem, aiming to understand how the faulty code was introduced and how preventive measures can be implemented to avoid such incidents in the future. Alice also takes the initiative to talk to Claire, providing reassurance that failures are a natural

part of the rapid software development process. She encourages Claire to view this experience as a valuable learning opportunity rather than excessively worrying about potential consequences.

Bob, on the other hand, follows a different path. Like Alice, he identifies the issue and works towards finding a solution. However, instead of offering direct support to Claire, he reprimands her for the mistake. Furthermore, Bob may deflect blame onto the partner QA team, focusing more on assigning fault than resolving the issue. While this approach may bring short-term changes and potentially reduce future failures, it adversely affects the team's overall morale. It instills fear among team members, making them hesitant and less effective as a cohesive unit.

Alice, in contrast, understands that being in a challenging situation can be tough for a new and motivated engineer like Claire. She recognizes the importance of providing additional support to help Claire emerge stronger from the experience. By adopting this approach, Alice ensures that her team remains highly productive in the long run, fostering a positive and collaborative work environment.

## **Clear vision**

An effective EM can articulate a compelling mission statement and vision for the team. This ensures that team members understand the purpose of their collective efforts and encourages them to think critically. The EM inspires the team to think ambitiously by providing a big-picture perspective. Let's consider Alice, who recently joined a new team. She quickly realizes that the engineers in her team prefer collaborating with cross-functional partners rather than solely working within their own team. In the early stages of her role, Alice proactively seeks out meetings with various cross-functional partners across the organization. This enables her to grasp the broader context and gain a comprehensive understanding. By working closely with different stakeholders, she successfully establishes a clear and concise vision for her team, igniting motivation among its members.

In contrast, Bob also joins a new team. However, he believes the team is self-sufficient and chooses not to involve himself in setting a vision. After a few months, the team begins to encounter challenges, experiencing frequent pivots due to the difficulty faced by individual engineers in establishing connections with all cross-functional partners. Despite the initial sense of empowerment within Bob's team, they find themselves on a path toward failure. With a clear vision set by their leader, it becomes easier to motivate the people around them. On the other hand, Alice has set her team up for success. The team clearly understands their goals for the upcoming years, and they only need to strategize how to accomplish them.

## **Lead with emotional intelligence**

Emotional intelligence encompasses the capacity to recognize, comprehend, and manage one's own emotions, as well as the emotions of those around them. It is regarded as a critical leadership skill because it empowers leaders to communicate, establish relationships, and effectively make informed decisions. A proficient EM can exhibit patience and composure despite conflict, stress, ambiguity, and escalations. They guide their team with empathy and emotional intelligence. The EM should possess the maturity to assess the team dynamics before implementing disruptive changes. They should also display self-awareness and excel as a team player.

Let's imagine a scenario where a crucial project deadline is looming, while simultaneously, the lead engineer assigned to the project experiences a personal tragedy at home. How should you handle this situation? Leading with empathy and emotional intelligence becomes paramount. Grant your team members the time and space to recover, and inquire if they require support from you or the company. Simultaneously, as you are responsible for delivering the project, it is crucial to communicate the risks to stakeholders and devise a contingency plan. Leading with emotional intelligence and fostering a psychological safety net for your team cultivates an environment conducive to growth, where individuals can

freely express their true selves. Moreover, it fosters optimism, resulting in heightened productivity.

## A learner with business and technical acumen

A proficient EM embraces a mindset of continuous learning. They understand that they don't need to be the most knowledgeable person in the room and are comfortable admitting when they don't have the answer to a question. Instead, their strength lies in their ability to find the right person to help unblock and provide solutions for the team. They should be adaptable and quick to adjust to changing circumstances while motivating the team.

EMs should possess a blend of business and technical acumen to guide the business and engineering teams effectively. In terms of business skills, they should have a knack for comprehending business goals, metrics, capital expenditure (capex), revenue streams, and objective and key results (OKRs), among others. Simultaneously, they should be tech-savvy and willing to understand the challenges engineers face to help them be more efficient. This includes actively contributing to code reviews, participating in technical design discussions, and engaging in brainstorming sessions during the design phase with meticulous attention to detail, all while demonstrating humility.

Let's consider a scenario where a junior engineer is grappling with a decision on whether to perform in-memory calculations or utilize a tool like elastic cache based on data size. Sensing the engineer's need for guidance, they approach their EM, Alice. Alice, who values technical knowledge, demonstrates her understanding of the problem and directs the junior engineer to online courses that cover the appropriate use of Elasticache. Alice also connects the engineer with the tech lead from a sister team with experience working on similar projects. This approach instills confidence in the engineer, making them feel heard and more comfortable communicating with their manager.

In contrast, Bob leans more towards being a people manager with minimal emphasis on technical acumen. If the engineer had approached Bob in the scenario above, there would likely be a communication barrier, and Bob would simply redirect the engineer to speak with the tech lead in their own team for assistance. As an EM, you can also demonstrate the value you place on technical knowledge by sharing learning resources such as online courses and books, encouraging team members to attend technical conferences, and giving them the time and space they need to use these resources.

## **Effective communicator and facilitator**

As an EM, it is essential to speak two distinct languages: one tailored for your technical audience, including your team, partner tech teams, and engineering peers, and another for the non-technical audience, such as cross-functional partners and stakeholders. The goal is to effectively convey the same information without delving into the intricacies of technical complexities. Communication is a crucial skill for both engineers and managers. As an EM, it plays a significant role in various forms, ranging from written communication for project roadmaps, promotion documents, and performance reviews to verbal communication addressing technical and non-technical audiences. This includes presenting and speaking to higher-level leadership, potentially engaging with customers, and honing public speaking skills. It also involves providing constructive feedback to team members in a meaningful way that helps guide and coach them. To be an effective EM, it is important to establish a healthy communication cadence across channels and foster an environment that encourages open and diverse dialogue.

Let's consider the experience of a new engineer, Jason, who recently joined a team. He often feels discouraged and hesitant to speak up during team meetings because he needs to perceive more space being created for incorporating various opinions. Recognizing this issue, Alice effectively identifies and communicates the gaps to Jason and takes proactive steps to address them. She implements a practice where the team uses the room to

solicit feedback during technical discussions, ensuring everyone's input is valued. Additionally, she actively works with Jason to increase his involvement in the team's technical decision-making processes.

On the other hand, Bob relies heavily on the senior engineer to present plans without actively seeking input from the entire team. As a result, under Alice's leadership, Jason can overcome his initial reservations and become more engaged in the team's day-to-day operations. Alice helped Jason by being an effective communicator and facilitator. In contrast, the senior engineer on Bob's team may continue to feel demotivated and ultimately disengaged from the team's operations due to the lack of inclusivity in communication and decision-making processes.

## **Effective delegator**

Mastering the art of delegation as an EM involves understanding your team members' career aspirations and how the tasks at hand can help them progress toward their goals. Skilled EMs excel at delegation, creating a multiplier effect by bringing out the best in individuals. The key is to establish trust and verify that trust without engaging in micro-management. Let's explore a real scenario where two senior engineers, Alice and Bob, transitioned into EM roles and initially struggled with delegation. Coming from an engineering background, they both had a natural inclination to maintain control over tasks.

Alice and Bob were assigned to different significant projects requiring collaboration across multiple cross-functional teams. Recognizing the need for delegation, Alice delegated her tech lead to engage with other teams, gather information, and establish a scrum of scrums to track progress and provide consolidated reports to the upper leadership chain. This approach made her tech lead feel valued, and the teams developed mutual trust. On the other hand, Bob wanted to keep full control and attempted to manage all aspects of the project, overextending himself, which resulted in confusion and frustration among team members. This also deprived other team members of potential development opportunities.

This scenario illustrates that as new EMs, there may be instances where you feel overwhelmed and stretched thin while adapting to the new role. The key is to identify this gap early and seek proper mentorship and coaching to embrace delegation. It may not be a straightforward process initially, but as time goes on, you will start to trust your engineers and delegate effectively. This will also provide you with the opportunity to tackle more significant and challenging problems instead of getting lost in the technical details.

## **Lead by example**

Yes, it is important to walk the talk and lead your engineering teams by example. This instills confidence and trust among the team members that you are not asking them to do something you would never do yourself. The team then trusts their leader and is motivated to take things to the finish line. At the same time, it is essential to understand what action will instill confidence in the team that you, as EM, are willing to put in the work too, and that the knowledge you have gained can help your team succeed.

Let me share my personal experience here- during the project planning phase for the upcoming half, my previous employer team was tasked to build a greenfield backend service in AWS using its platform as a service capability from scratch. The engineers in our team and I needed to become more familiar with AWS stack as it's a vast marketplace. So as a first step, I dived into the project's high-level details with my senior engineer and then decided to prepare myself for the AWS developer certification exam. Not only this, I paired up with another senior EM to start group learning sessions on AWS for broader adoption within the teams. I studied for a good two months or so and then cleared my exam. This is when I shared the news with my team members, influenced and inspired them to also deep dive and proactively prepare for what's coming ahead. As a result, six out of my eight engineers went on to get AWS certified and delivered the project involving the AWS suite in a timely manner, what a win right?

## **Foster innovation**

As an EM, you are responsible for optimizing your team's delivery. This means you should be receptive to feedback and new ideas on the table and help build a conducive environment that promotes innovation. People around you should feel heard. This can be through promoting team hackathons or valuing innovation through brainstorming new ideas that help build a future backlog. At the same time, companies need to support employees by giving them the time and resources to participate in such hackathons and team events. Several companies have tried to incorporate the [postmortem blameless](https://sre.google/sre-book/postmortem-culture/) (<https://sre.google/sre-book/postmortem-culture/>) culture as part of the workplace.

Similarly, a strong EM believes in correcting errors instead of a blame game. The idea is to learn from the failure and do a root-cause analysis. Innovation is often a fail-fast endeavor, and failure is sometimes a bad thing. Let's say the company organizes a hackathon for the employees, maybe ten ideas are presented, but only three make it to the top three that get leadership buy-in to be taken to production. Does that mean the other seven ideas failed? Not really- they are good potential candidates for future backlog and might not be the highest priority for the company at this moment. As EMs, your job is to create an environment for healthy mistakes and opportunities for team members to learn.

## **Recognize effort/success**

Recognizing the talent around you is the key to building trust and motivating others. Rewarding and recognizing the talent shows you acknowledge the hard work of others around you and are vested in team members' growth. If someone around you went above and beyond to get a task done, had a quick turnaround and delivered a project before the deadline, helped organize a social event for the team, and many more- these are some basic examples to acknowledge and recognize the people around you. Such gestures from the engineering leadership team send a positive affirmation to the team. This also means that a good engineering leader will represent their team well and find forums to give the team the necessary

visibility. We will learn more in the reward and recognition chapter, so stay tuned.

Some ways which I have used to bring my team's recognition are:

- Shoutouts to the person or team on the company chat in team and leadership channels. Also, sharing the situation and how their role had a huge impact.
- Recognizing people on internal platforms or tools where I could applaud them against the company's leadership principles
- Using monthly meeting forums to acknowledge all the great work
- Finding opportunities for team members to present their excellent work in All-hands or leadership meetings for them to get visibility

## Organized

A skilled engineering leader excels in managing their time effectively. They face the challenge of balancing multiple responsibilities, such as attending meetings, planning for staffing needs, conducting one-on-one sessions, overseeing project execution and delivery, and reporting to senior leadership. They must be able to manage time efficiently and possess strong organizational skills. The organization encompasses various abilities, including time management, strategic thinking, prioritization, discerning urgency, distinguishing between essential and non-essential tasks, and more. Let's consider a situation where, as an Engineering Manager, you consistently need help to fulfill your commitments. It is crucial to differentiate between your personal commitments and the project commitments of your team. Falling behind in your work will make it exceedingly difficult to catch up, clear your backlog, and effectively address the other responsibilities mentioned earlier. Tools like Slack, Trello, Jira, Asana, etc., can greatly assist in enhancing organizational skills.

The aforementioned traits are not an exhaustive list but encompass the essential qualities that set apart a strong engineering leader from others. The goal is to continuously refine the core skills of an Engineering Manager, enabling both yourself and your engineering team to reach their full

potential. It is uncommon for an EM to excel in all the mentioned traits, as there is always room for growth. Focusing on how one is improving over time can help understand the growth opportunities better. As we have observed, some skills can be coached and developed over time. Having a comprehensive understanding of the key traits of a capable EM, let's now proceed to the next section, where we will explore the success metrics to consider in an EM role.

## 1.3 Success metrics as an engineering manager

While the above traits are important to understand and practice to be a successful EM, it is also important to define some ways to measure your success objectively. This is to ensure the adjustments and changes you are making for each of the above-defined areas contribute to your success. As an old saying goes: *You can't improve what you don't measure.*

A question that pops up in every EMs's mind at one point or another is, can we even measure success as an EM? If yes, then how do we measure it? Success is a moving target. What would look like success to you at one point might have a different connotation for someone else. Also, with regards to yourself, success, when you were finishing your studies, might be to get good grades and finish assignments on time, while when you joined a job as an engineer, success was to love your work, give your best and maintain a work-life balance. Success metrics help provide an assessment to understand if we are meeting the performance and success criteria. This also helps you plan strategically and identify areas where you and your team need support. While there are several avenues to optimize and improve, I have generally observed the following key areas, which provide the easiest examples to measure success.

- Optimizing for key business metrics and delivering value
- Increasing the team's development velocity
- Creating and maintaining healthy team morale, increasing team members' career progress
- Optimizing for operational metrics

While these four areas define some crucial avenues to explore, it is critical to remember that all of these areas are intertwined, and success and failure in these areas may go hand in hand. For example, Increasing developer velocity can directly impact moving key business or operational metrics, given proper processes are in place to reap those benefits. At the same time, sustained failure to move business metrics can also directly impact the Team's Morale, which can go south in tough times. Hence, let's dive deeper into each of these areas to gain a better understanding

### **1.3.1 Business metrics**

The success of an EM can be measured by business metrics, which reflect the value delivered by the team to the business or customers. These metrics encompass various scores such as customer satisfaction (CSAT) or Net Promoter Score, customer retention rate, and customer feedback. Additionally, it is essential to ensure the metrics are realistic and that the work aligns with the company's overall Objectives and Key Results (OKRs) and strategic direction established at the beginning.

For instance, let's consider a basic example where your team ships a new product that increases the CSAT score by 20% and reduces customer escalations by 30%. How do you determine if this is a success? As an Engineering Manager, you consistently assess the goals and define how success will be measured against them. During the project's initiation, you collaborate with your cross-functional team and stakeholders to establish success metrics that will be used to evaluate the project's outcome. Therefore, based on the predefined success metrics, meeting or surpassing the set goals indicates a successful product. While the specific threshold metrics for success may vary depending on the situation, the key is to establish success metrics before the product development begins and evaluate the outcomes against them once the feature or product is launched.

Optimizing for business metrics may appear straightforward in many scenarios, especially when technical complexity and resource allocation are properly addressed. However, challenges arise when dealing with scenarios involving multiple stakeholders/vendors and ensuring accurate

identification of the root cause behind metric changes, which requires meticulous analysis. For example, a 30% reduction in customer escalations may be attributed to improved training for call center staff and the troubleshooting team's creation of additional troubleshooting guides, with the new app contributing only a small portion. In such cases, it becomes important to question whether it is fair to solely attribute success to the metric related to the new app. Similarly, if customer escalations increase, the responsibility should be acknowledged rather than shifting blame to other teams ("the call center staff were not properly trained to use our app"). While valuable, optimizing for business metrics can become complex in certain situations. Let's now explore one such scenario, increasing development velocity.

### 1.3.2 Development velocity

Now that we looked at the business metrics, the next is to measure the success using the development velocity of your team. [DORA metrics](https://www.leanix.net/en/wiki/vsm/dora-metrics) (<https://www.leanix.net/en/wiki/vsm/dora-metrics>), which stands for the DevOps Research and Assessment team, play a key role in measuring the performance of the engineering team. This includes measuring four key metrics:

- **Deployment frequency** - How many times in a week are you able to deploy your services to production, are you able to leverage continuous integration, and continuous deployment to ship products faster
- **Lead time for changes** - It refers to the duration from when a code change is committed to the point where it becomes deployable.
- **Mean time to recovery**- In case of an issue, the time it takes to recover your service or product back to normal and get to a resolution
- **Change failure rate** - It measures the frequency at which a team's changes or hotfixes result in failures once the code has been deployed.

Some other metrics beyond DORA that can help are:

- Sprint velocity- the number of story points your team can deliver in a sprint, this includes assessing the burndown charts of the sprint
- Time to complete code reviews- ensuring engineers contribute to helping review peer code to improve as a whole
- Number of bugs reported and resolved- Monitoring the number of bugs reported by customers and QA and bringing the count down

As you can see, there are several sub-avenues you can try to improve here, but ensure the quality is maintained. We discussed a few key metrics that give a fair idea of measuring development velocity, but there are much more. Development velocity metrics are harder to move and require a much longer time to observe and make fair assessments. For example, measuring velocity in a way that allows for successful comparisons between team members, teams, and projects is incredibly hard. So focusing on whether we provide value versus velocity is becoming the norm. Your teams might be using story points to point out the stories in the sprint, but they should serve as a reference point to understand the team capacity, group understanding of the task, and how good the team is with estimations. Sometimes using story points to compare individuals can lead to confusion and unfair assessments and hence the focus on delivering the overall value.

Therefore it is important to pick a few areas in which the team is struggling and focus on those high-priority items. Once you start to see improvements, expand to pick other areas. For example, let's say you have noticed that for some of your essential services, you have seen issues that brought down the fleet for over 15 mins. If this downtime is unacceptable for your services (15 mins is a very long time), then optimizing just those can move the needle for your team.

### **1.3.3 Team morale**

Team morale is delicate and needs the utmost care and attention. Teams with high morale help bring out the best in individuals, building a culture of collaboration and open communication. Such members are willing to help each other and have a higher job happiness quotient and engagement factor. On the other hand, teams with low morale tend to have a negative attitude

toward each other, with more chances of bottlenecks and single points of failure.

As an EM, one of the key success metrics is team morale which has a direct impact on teams' performance and helps decide:

- Participation in meetings and activities like retro, planning
- Job happiness quotient
- High retention rates, that is, lower attrition rates
- Trust and collaboration levels

A few ways to understand team morale are:

- One-on-one discussion with team members
- Casual coffee chats to keep them informal and candid
- Quarterly or annual surveys across the company and at the team level
- Periodic team meetings to provide an open forum to voice concerns and find ways to improve as a team
- Upskilling team members and ensuring they are progressing in their careers
- Keep a truly open line of communication with team members so they feel you are approachable

The team morale dictates if the members are willing to go that extra mile to help each other and help build a conducive learning environment, hence as an EM, you are successful in the role if the team morale is high. This can also be measured through team surveys and acts as an objective metric.

### **1.3.4 Operational excellence metrics**

Operational excellence and process improvements are the backbone of an engineering culture in a company.

They dictate the health of services and processes and signal how much the teams focus on launching new products versus nurturing existing ones. Some metrics that can come in handy here are:

- Tech debt in the team
- Defects resolution rate(incoming vs resolved)
- Test code coverage(Note: While the presence of test coverage does not provide definitive proof, its absence can indicate a problem.)
- Extensibility and readability of code
- Code maintenance and on-call reports
- Reducing costs and downtime
- Meeting defined Service Level Agreements(SLA) contracts
- Focus on quality over quantity
- Availability of services in production

Operational excellence metrics provide valuable insight into the state of team services and the codebase. Over time, every team accumulates technical debt, but successful teams actively manage and reduce it while implementing process improvements to enhance their efficiency as a developer team. However, it's important to be cautious about specific metrics that may lead to confusion, such as the number of lines of code or the raw count of user stories closed by a developer. Although these metrics involve numbers, they can be misleading since there are multiple ways to write the same line of code that produces the same output as an Engineering Manager. A true testament to your success is whether the team can continue operating smoothly even when you are on vacation.

Allow me to share a personal example from my previous team. We noticed a gradual increase in on-call work, the first line of defense for any issues raised during a designated on-call period. Team members typically rotate and take turns assuming these responsibilities. Upon closer examination, we discovered that our core services deteriorated slowly due to various factors. If left unaddressed, this situation could have had severe consequences for the team in the long run, including reduced developer velocity (as the team was occupied with issue resolution rather than delivering new products), decreased team morale, and an increased risk of burnout. To tackle this challenge, our team collaborated closely with the site reliability teams to generate ideas and improvements for enhancing the overall health of our services. We prioritized and dedicated a quarter to working on initiatives related to this endeavor, aiming to achieve a significant impact. This

exercise was crucial for the aforementioned reasons, and ultimately, the team regained its total development capacity.

Having examined straightforward methods for measuring success metrics as an EM, it becomes evident that a high-performing company and team function like a well-coordinated machine. To effectively manage and lead a team, it is crucial to recognize different leadership styles and determine which aligns best with your team's dynamics. In the following section, we will explore common leadership styles and examine their advantages and disadvantages.

## 1.4 Leadership styles

A defined leadership style helps set the right expectations for your team and you. There are several leadership styles to consider. Let's meet and consider six common engineering leaders for our discussions and learn about their leadership styles. Again, this list isn't exhaustive, and there are many other types of leaders with various permutations and combinations of different styles. The leadership styles will give us a window into what works versus what doesn't. Note, it is not important and instead not suggested only to pick one leadership style and run with it, one of the most important skills to master as an EM is to adapt your leadership style given the situation.

### 1.4.1 Autocratic leadership

Meet Alissa!

She has been an EM for quite some time and likes to follow autocratic leadership. She is particular about the inputs and outputs of the problem statement and likes being prescriptive to her team members (micro-manager). She feels she has vast industry experience and prefers her team members to do as she says and follow her. She likes to feel in control of the situation and believes this style helps her move fast by making decisions for her. Recently, Alissa decided that every Thursday (the day she is in the office) will be a day for all team members to be physically present.

Her team members feel this is restrictive to their work and career growth. They don't have many channels to give feedback or voice their opinions, hence creativity and innovation take a backseat. They feel they have no open voice and don't feel heard. There is a lack of morale and trust among the team members. One of her engineers recently had a child, so she cannot make it to the office on Tuesdays and Thursdays due to missing daycare support. Alissa strongly dislikes this situation and expects everyone to be in the office every Thursday.

You have an autocratic leadership style like Alissa if you:

- Like to keep the situation fully under your control
- Don't value the opinions of others
- Tend to micro-manage others
- Feel they are primarily right or, to say, has overconfidence in one's knowledge
- Are happy being a restrictive leader, practicing your full authority
- Like to follow the rules and a playbook
- Take decisions without asking about others' opinions and preferences
- Maybe more self-focused than being team focused

While this leadership style can work effectively in some scenarios, for example, a tight deadline for a project or a team where accountability is missing, and an autocratic leader can help provide guidance, it has not been well received. Let's look at some other leadership styles that may resolve issues surrounding this approach.

### **1.4.2 Democratic leadership**

Meet Brayden!

He is a relatively new EM. He believes in relationship building and wants to hear and learn about his engineers' opinions, even though he plans to take a final call in the decision-making process. He believes that such active participation of his team members helps bring new ideas to the table and helps build trust in the team.

In the recent decision-making process of when to have sprint grooming for the team, he sent out a poll to the team on the Slack channel. He also brought people to discuss their views in the team standup. From the poll, the top two options were Monday and Friday. He felt that many people prefer to take Fridays off when planning long weekends and hence took a call for Monday to be the day of sprint grooming for the team.

His team members feel they actively participate and are part of the decision-making process. They get to share their ideas and feel heard. Sometimes, they proactively bring ideas to the table even before being asked. There is a sense of satisfaction in the team members for their work.

You have a democratic leadership style like Brayden if you:

- Value ideas from your team members
- Are a good listener
- Open to innovation and creativity on the table
- Welcome people with differing viewpoints
- Can budge sometimes to keep the team happy and motivated

While this is an effective way to mitigate the problem, it has some cons. For instance, decision-making can be slow as team members need to answer a poll or the EM needs to chase people to answer the poll. This can be detrimental if the business needs to move fast. You may also find other leadership styles addressing the same issues. Let's look at another.

### **1.4.3 Delegative leadership**

Meet Charlie!

Charlie has been an engineering leader for several years across several industries. He believes in the idea of delegation to help him scale as he manages multiple teams across different geographical areas. He aims to build a culture of trust and follows the 'laissez-faire' or delegating leadership style. He enjoys empowering individuals by giving them the authority they need to make decisions, with minimal supervision from his

end. This is a hands-off approach where he focuses on ‘what’ is to be done and not ‘how’ it is to be done. When his organization had to decide how to load test the backend services for the upcoming peak traffic days, he delegated the decision to his team to decide the tooling, timelines, and a plan to run the load testing across all services. He was mainly interested in the outcomes of the load tests that ran and if he needed to help somewhere.

His team feels motivated as they have a leader who trusts them. The team is fully motivated and feels a positive work culture around them. They enjoy full autonomy in their work but sometimes find it detrimental to their career growth as they make decisions based on the best of their knowledge and feel they need to learn more from their leader. Some junior engineers would prefer hand-holding and more supervision than they currently have.

You have a delegative leadership style like Charlie if you:

- Like to offload decision-making to your team members
- You believe in your team members and that they will make the right decisions
- Prefer a hand-off approach to management
- Foster a culture of empowerment and leadership in the team.

As you can see, delegative leadership can also work effectively for the most part. There are some scenarios where a more prescriptive management style is preferred, especially when dealing with more junior engineers. An inherent drawback of this approach is striking the right balance between hands-off and hands-on EM. In such cases, it may appear that the EM only shows up to collect results, but a blame game ensues when something goes wrong, and it is too late to rectify it. Striking a good balance can significantly alleviate these challenges and yield excellent outcomes for your team. Let's explore another style that may yield favorable short-term results but carries the potential for numerous long-term issues.

#### **1.4.4 Transactional leadership**

Meet Dave!

Dave follows a transactional approach where he incentivizes his team members for their work. He wants to create a gamification of the work and feel people thrive in competitive situations. He enjoys acknowledging the team members' efforts and providing rewards for expected work. He also is quick to apply consequences when necessary if the work is not as expected and defined. He is focused on providing training to engineers to deliver results and enjoy the perks of it. He has a rockstar engineer on his team and loves to reward and recognize them. They share their stories with other engineers to show what is valued in the organization. He also has an under-performer on the team who feels neglected and rejected as his work is overlooked.

His team members feel like they are following a leader in a marching band. They don't feel they are valued for bringing creativity to the table. Everyone feels like a rat race and is trying to work on impressing Dave so they are rewarded accordingly. There is a less open environment, and people tend to feel uncomfortable sharing things openly with each other.

You have a transactional leadership style like Dave if you:

- Like to incentivize people for work(if you do excellent work, you are well rewarded and are top of your game) and punish if not done your way(struggling and could very well drown in the river)
- Building a competitive culture in the organization
- Tend to micromanage
- Value hitting the goals

While transactional leadership may work to improve Business or Operational Metrics, they certainly adversely affect the overall Team Morale. This can cause unintended consequences like unexpected attritions, a clear divide between strong and weak performers in the team, and a general sense of disaffection. Now that we understand different leadership styles, let's look at a few more to identify what can be brought to the table.

#### **1.4.5 Transformational leadership**

## Meet Eric!

Eric likes to be a role model for his team members. He is a visionary leader and keeps the company's goals and mission as the top priority. He has a magnetic personality and can aggressively drive toward companies' goals and objectives. He focuses on the big picture and doesn't hesitate to make bold decisions. He recently inspired his team to get away from a legacy platform they used for the messaging service and instead adopt the latest state-of-the-art technology. This would be a bold move as they had to keep the lights on for the existing service and spend resources on building a new one.

Eric's team likes the idea of thinking about the future and how it will change the world. They see Eric as a strong visionary leader. They sometimes feel that more minor details are missed in the big-picture thinking and might not always have a voice in such bold decisions.

You have a transformational leadership style like Eric if you:

- Likes to lead through inspiration
- Considered to be a visionary leader
- Keeps the company goals and missions as the top priority
- Unites the team to help achieve the goals
- Believes in being strategic and bold

Transformational leaders are an absolute necessity for any growing company to survive. This is often the style associated with enterprise architects, DevOps/DevSecOps teams, as they are generally tasked with being transformational. However, this leadership style must also be carefully balanced with other leadership styles. Focusing only on the future and not the present can also cause unintended consequences. For example, let's say a complete re-architecture of a core piece of infrastructure was proposed by the manager. Still, the effort and resources involved in developing this took away much time from operational excellence. Do you see any issues with this? If you are already stretched thin for operational

reasons, you may stretch your team even more during this migration phase. So use this leadership style with caution.

### **1.4.6 Servant leadership**

Meet Frankie!

In a few words, she calls herself a people manager with someone who has a people-first mindset. She wants to bring out the best in her engineers by providing continuous mentoring and coaching. She meets each of her team members weekly and has an ongoing career document with each engineer. She has set SMART goals with them, understands each engineer's strengths, weaknesses, and aspirations, and tries to find opportunities that align with their career goals. She strives to bring the best out of them and keeps her team engaged through team-building activities and active collaboration. She provides constructive feedback to the team members to help them grow. She demonstrates emotional intelligence and empathy to her team members, who love being coached by their managers. But, she sometimes struggles to push back in times of conflict when her engineers want to work on something which doesn't align with the company's vision. One of her engineers was looking to shift from being an engineer to a product manager. She connected this engineer to the sister team's product manager and spoke to her leadership to find product-related opportunities for the engineer. She worked on a transition plan and helped the engineer move to product management as it aligned with their career aspiration.

Her team is actively involved in the decision-making process and brings new ideas. They feel personally satisfied, confident in voicing their opinions, empowered, and trust their leader. There is a high retention rate, and team members feel their leader is vested in their career and growth.

You have a servant leadership style like Frankie if you:

- Like to keep a people-first management approach
- Enjoy mentoring and coaching others
- See your role as enabling others to be successful

- Like learning the motivations and aspirations of your team members to provide them with the right opportunities
- Provide guidance and feedback to the team members
- Believe in team building and are willing to do whatever it takes for the team to succeed

This is also another healthy way of leading the team. The team feels engaged and motivated, members have a sense of belonging, and most importantly, your team members are happy with their work. It is important to play the devil's advocate here to ensure your team is not coasting around; however, practicing this style with grains of caution will produce good results.

So who do you resonate with the most? Is it Alissa, Brayden, Charlie, Dave, Eric, or Frankie? Let's see how one can go about choosing their leadership style.

#### **1.4.7 Choosing your leadership style**

So which leadership style do you relate to? Now that you learned about various leadership styles, you may ask yourself what type of leadership style I resonate with.

Let me clarify, there is no silver bullet or one size fits all approach here. You can find yourself strongly aligned with one of the leadership styles above, or you may demonstrate a mix of the above styles, as each has its own pros and cons. A hybrid leadership style is Ok and, in fact, the most used. This is because leadership styles depend on multiple factors:

- You - your experience as an engineering leader and your personality
- Team - how big is your team, what experience level of your team members, their preferences and aspirations, and how they perceive you
- Company - Several internal and external factors dictate how the company is run, the company culture, the product strategy, competition, time constraints, and so on.

Hence, choosing a leadership style is very situational. That means depending on the situation you are in at that instance of time, you will look at each situation uniquely and decide what is appropriate, you might go with a particular leadership style or use a mix of styles and form your own **unique hybrid approach** while keeping a positive attitude and growth mindset which involves embracing learning opportunities, in contrast to a fixed mindset, where one stays confined to their own comfort zone.

To illustrate this point, let's consider a scenario where we face a decision-making deadline, and consensus from all parties involved is not a prerequisite for reaching a decision. In such a situation, we might employ autocratic and delegation approaches. We would help resolve conflicts and reach a decision, but due to time constraints, we might delegate the execution to a team member better suited for the task. Therefore, based on the specific situation, we combine two leadership styles.

Similarly, while we aspire to be servant leaders who prioritize the well-being of our team, there are instances where tough choices must be made. I have experienced situations where companies had to make the difficult decision to downsize their workforce due to unfavorable market conditions. However, this does not mean that the leaders were not servant leaders or failed to prioritize people. In certain circumstances and constraints, leaders must adapt their styles and explore uncharted territories. This may require a combination of transformational and transactional leadership styles.

Another important factor is the need to be willing to unlearn/relearn a new approach or leadership style if the initial one proves ineffective. Let's consider another example to illustrate this. Imagine you are leading a large engineering organization. The organization's overall mission includes maintaining core backend services developed a decade ago and have been the cornerstone of the company's business. Recognizing the presence of technical debt within the teams, you adopt a transformational leadership style. You appoint a core team of engineers and DevOps specialists to focus on process improvements and reduce technical debt and operational burden associated with these core services.

Over the next three months, the team diligently works on this initiative, but the gains in operational excellence are minimal. During this time, you discover that a competitor has launched a new customer service support feature, providing 24/7 availability, directly impacting your company's business. The intense focus on transforming existing services has inadvertently led to unintended consequences in the present. It becomes clear that, at this stage, the company needs to prioritize increasing revenue sources to stay competitive, setting aside any immediate operational excellence initiatives.

In response, you swiftly pivot to address the urgent situation. You collaborate with your leadership team and propose an organization-wide hackathon to generate innovative ideas to counter competitors' offerings and increase company revenue. This shift in approach embraces a more democratic leadership style, promoting a vision of revenue growth and fostering innovation. It is a typical situation where the initial plan and leadership style must be discarded, and a new direction is embraced based on the current circumstances and the company's immediate needs. In sum, there is no one cookbook to choose a leadership style. Given the circumstances, you, as a leader, will face new and unconquered situations. Hence the core idea is to understand the fundamentals, adapt as you follow the situation, and strive to be an effective engineering leader.

### **What do other leaders have to say about it:**

“The most important traits of an engineering leader are focusing on recruiting, motivating, enabling, and retaining the strongest team possible. Think about the team and cross-functional stakeholders as a machine with inputs, outputs, and internal processes - figure out where the bottlenecks are, where there isn't enough backpressure, where adding extra capacity will help, and where adding extra capacity won't help.”

**~ Jean Bredeche, Head of Engineering at Patch, (ex) Robinhood, Quantopian, Hubspot**

“I think this trait is incredibly important in people, but especially in leaders: empathy. Mentoring or teaching others is one of the more obvious situations in which this is useful since it can allow a mentor to discuss the relevant subject matter in a way that the mentee understands better. But this extends to all the relationship-building that a leader does as well. An engineering leader often partners with product leaders, creating an information bridge between them and engineers. By understanding the incentives of their coworkers, they can also more effectively assess the structure of an organization (of people) and whether those incentives could be better aligned.”

**~ Richard Frank, Senior Software Engineer(ex-manager) , Two Sigma | (ex)Robinhood**

“The most important trait for an engineering manager is understanding tech and product. As an EM, you work very closely with engineering and product teams, and you can only contribute effectively if you understand both sides of the world. I resonate with democratic leadership. With so much opportunity in the current market for engineering, you need to take your team along with every decision you make.

**~ Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

“Though many traits are important to be a good leader and classifying anyone as most important will not be completely right . While creativity, curiosity, collaboration, and communication might top the charts, I think Empathy is crucial. It is extremely vital to be empathetic towards your team members, your clients, your end users, and everyone you interact with.”

**~ Devika Ahuja, Technology Leader | Strategist**

“For challenges as EM, in all the companies I’ve worked with recently, the prevailing leadership perspective was that engineering wasn’t delivering enough or was going too slow. This is presented as opinions like “engineering is lazy” or “engineering doesn’t consider our needs.” Yet, when you talk with engineers, they consider themselves overworked,

frequently putting in nights and weekends to hit deadlines they don't understand. The biggest challenge I have faced in working with these companies is rehabilitating the reputation and trust of engineering teams. The first step has continually been improving the work visibility of engineering efforts. Once you get leadership to internalize that folks ARE working hard, that they ARE working on their understanding of priority, and that folk DO WANT to do the best work for the company, it becomes easier to have reasonable conversations around constraints, challenges, etc.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader.**

“The most common issue I’ve seen EMs face is being able to make prioritization decisions on the move. Most of them struggle with using the right data to make these decisions. I would recommend using the [Eisenhower Matrix](#) to put things in perspective quickly. Another way of making quick prioritization decisions is asking yourself -Would this be an easily reversible decision or not? If it is, it doesn’t need as much scrutiny as a non-reversible decision.”

**~ Adish Agarwal, Director, Software Development at Audible, Inc.**

“Based on my journey of 15 years of learning as a leader, I’m continuously working to be an Authentic Leader. I define an Authentic leader as a human being whose internal and external personas are aligned. They do not have to act as differently as a leader, they present themselves the same at work as they are at home. In my view, there are 3 traits of an Authentic Leader:

- 1 They are highly self-aware of their strengths, weaknesses, and various emotional states in other words, they have a very high EQ(Emotional Quotient)
- 2 They are good human beings with high integrity who want to sincerely help others and make life better for everyone around them.
- 3 They make time to do new things every day, from learning new things to having fun doing activities they love for physical and mental fitness.”

### **~ Sumit Kumar, System Engineering Manager at Cisco**

“To me, the leader of any engineering team or org has to come from a strong Technology background. In addition, all the best engineering leaders have a great eye for the future of the industries and trends and stay in touch with the latest. Another key trait is how they treat people and how people respect the leader. I see an engineering leader as a mix of technical acumen, people skills with empathy, and a strong personality to stand for what he or she wants for his organization. I align more with a servant leadership style. As I transitioned from an IC to a manager, it was a natural transition for me to be close to the team and understand the pain points. EM's role is the mix of technology meets business without impacting people's focus. You are always flirting with those fine lines impacting one of those areas. A successful EM figures out the balancing act.”

### **~ Rajakumar Sambasivam, Delivery Manager at Microsoft**

“An engineering leader should have a good overall understanding of the technology landscape the team is working on, along with people management and leadership skills. They should know when to get into day-to-day execution mode and when to switch to formulating strategy and vision for the team. They should also be able to tune their communication skills according to the group they manage - managing down, up, or sideways.”

### **~ Sarin Panambayil, Principal Engineer at Yahoo**

“The most important trait of an engineering leader is to deliver the product, and you have to be able to say to the product owners and your bosses that I'm going to deliver these features in this product platform by this date and a new set of features you know could be phase two and I am delivering a stable, scalable way, by another date and here's how many engineers I need to do it and here's the tools I need to do in the budgets I need to do it.”

### **~ Larry Gordan, Managing Director at Emtec Inc.**

## **1.5 Stop & Think: Practice questions**

1. Who has been your favorite manager in your career, what was different about them?
2. Who has been that one manager you have disliked working with the most, what was different about them?
3. In your opinion, what is the most important trait of an engineering leader?
4. If you have been an engineering leader for a while, think about your first role as an EM, what are the best memories from that time?
5. In your opinion, what are your strengths and weaknesses as an engineering leader?
6. Which leadership style do you resonate with the most?

## **1.6 Summary**

- The role of a manager, in its basic terms, is someone who helps supervise the employees, catering to their needs to be effective and productive in the role. This also means helping build a strategy for the business to grow.
- The role of a manager cuts across various industries like technology, customer care, sales, finance, restaurants, and more. The core responsibilities include but are not limited to hiring, managing performance, delegation, recognizing talent, project delivery and execution, keeping employees motivated, and helping improve overall team productivity.
- The engineering manager role is both an art and a skill, with an EM helping with resourcing, allocation of engineering resources, conflict resolution, managing expectations at all levels, and helping boost the morale of the team of software engineers.
- This book is for people looking to make a career move from IC to EM, are new to the EM role, an experienced EM looking to hone their skills, an EM looking to prepare and change jobs, or a middle manager who will soon have new EMs reporting to them.

- The core roles and responsibilities as an EM include building talent, providing clarity to the team, leading projects, practicing effective delegation, and focusing on engineering excellence. It is important to be an engineering leader instead of a manager by helping the team provide a vision and charter.
- Various traits are exhibited by good EMs that help differentiate them from the rest like
  - people first mindset
  - clarity on vision
  - lead with emotional intelligence
  - exhibiting technical acumen
  - delegation
  - communication
  - honesty and integrity
  - leading by example
  - fostering innovation
  - recognizing the talent around
  - keeping themselves authentic and organized
- It is of prime importance to measure the success of an EM and look for metrics such as business metrics, development velocity, team morale, and operational excellence metrics to objectively define the success in the role.
- Various leadership styles exist, each bringing its own unique impact to the team members.
  - The autocratic style works in time-constraint scenarios but is generally poorly received by team morale.
  - Democratic style brings everyone together to make a consensus decision, but it can be time-consuming.
  - The delegative style works in cases where the person being delegated work is senior and might not work for cases where a more prescriptive approach is needed.
  - Transactional style helps boost business and development metrics but hampers team collaboration and morale.
  - The transformational style needs to be balanced with other leadership styles, as focusing on the future only can cause dire consequences in the present.

- Servant leadership is when the team feels motivated and engaged
- One does not need to resonate with a particular style and should adapt to a hybrid style depending on the circumstances.

## **2 Individual contributor to engineering manager**

"My job is not to be easy on people. My job is to take these great people we have and to push them and make them even better."

~Steve Jobs

### **This chapter covers**

- Differences between an individual contributor and an engineering manager
- Common misconceptions when transitioning from individual contributor to engineering manager
- Sharing intentions and motivations to be an engineering manager
- A three-phase approach for individual contributors to engineering manager transition
- Helping others transition from individual contributors to engineering manager

In the previous chapter, you gained insights into the responsibilities of an engineering manager (EM) and the significance of managing teams and individuals as you progress in this role. As more software engineers contemplate management as a career path, it becomes crucial to comprehend the fundamental distinctions between the roles of an individual contributor (IC) and an EM. Let's start by examining a concise definition of IC and EM at a high level.

**IC** - An individual who contributes to the team and the organization without managing others. If you are currently a software engineer, you fall into the IC category.

**EM** - A person who supervises and manages other engineers. This could involve directly overseeing software engineers or, in the case of a senior manager (manager of managers) or higher up in the chain as directors.

Hence, if you are responsible for a group of software engineers or managers and engage in people management, you are considered an EM.

This chapter delves into the differences between an EM and an IC, factors to contemplate when deciding which path is right for you, and how to navigate such a transition. Along the way, you will gain insights into the common challenges associated with this transition and discover strategies for supporting others undergoing similar changes. Whether you are an IC exploring the possibility of transitioning to EM, a relatively new EM, or an experienced EM who will eventually oversee ICs transitioning into the role, this chapter offers valuable perspectives covering all angles.

Now first and foremost, any IC-to-EM transition stems mainly from three main reasons (not limited to):

1. You are genuinely passionate about the EM role and have been thinking about it.
2. It is a forced move where your leadership asks you to take the role due to a lack of EMs in the organization.
3. You inherit the role by default. You continue as team lead and find yourself managing others, and then are promoted because “you are doing the job already.” Kind of a variant of #2, but one can call this the “surprise” role change.

I recommend that whether you are considering becoming an EM or have been invited to assume the position, it is essential to pause and reflect on the following inquiry:

"Does this job/role align with your suitability?" But to answer this question, we need to dive into the differences between the two roles, an IC and EM role.

## **2.1 Differences between the individual contributor and engineering manager role**

We explored in the previous section why someone would consider transitioning from IC to EM - maybe by choice or something being forced upon them. The IC and EM roles differ in various aspects just as who do you give more priority to, how time commitments change, importance to people aspect, success metrics, how specialization changes, role mindset and many more. To ensure we make the right call for ourselves, let's look in depth at how the two roles of an individual contributor and an EM role differ in the tabular form.

**Table 2.1 Differences between IC and EM role**

Category	Individual Contributor	Engineering Manager
Priority	<p>You will value yourself and then the team, thinking about how to increase your efficiency as a developer</p> <p>You are task-oriented and typically directed to complete tasks by the team, team lead, product owner, or manager.</p>	<p>As an EM, the team comes before you. You are always thinking about how to build a high-performing team and increase the developer velocity. You are responsive and proactive and typically prioritize needs for others, being flexible and adaptable. Rarely do you follow a set or defined task list that includes personal agenda.</p>
Time	<p>The majority of your time is spent coding and shipping products. You get more focus time to work on tasks.</p>	<p>Coding takes a backseat. Most of your time is spent managing people, building teams, building strategies, and influencing people around you. You get more</p>

		meeting time than focus time.
People Aspect	You can be mentoring other engineers but not worry about the people/career aspect of the mentoring relationship	Here, you are responsible for the career growth of your team members. Along with it comes the other not-so-fun HR issues and legality. You help people get promoted and manage under-performance
Success metrics	You have tangible outcomes at the end of the week- products shipped, services deployed to production, along with some intangible things like learning and development. Hence, it is easy to measure success. At the same time, not all metrics are tangible. Some are subjective: are you learning something new? Expanding your skills in your position?	Success is abstract. At the end of the week, you might start questioning yourself in terms of what your output was. True! You did participate in meetings and spoke to people but might not have something as tangible as an IC might have.
Specialization	You can be a specialist in your area or a subject matter expert in your technical area of expertise	You are more of a generalist with a breadth of knowledge on technology and will be an expert in people management
Role Mindset	'Doer' or 'creator' mindset. As an IC, you might be	'Delegator' or 'enabler' mindset. Also, educator,

	following examples of others.	mentor, and career coach in many ways. You lead others by example.
Strategy	You tend to think about the 'what,' 'when,' and 'how' of technology. Senior ICs can help contribute towards the 'why' aspect.	You must consider the 'why' for a project/technology/business. You must operate at a higher level. Managers fly at the 50,000-foot level, not the 2,000-foot level.
Outcome	Your work can see more immediate results, such as projects shipped, the velocity of delivery, product/project completion, DORA metrics, etc.	Your efforts can take time to show outcomes, such as successfully leading an organization-wide program. Your outcome is far more subjective and will think of the successes of your people.

To summarize, although both roles come with big responsibilities, they differ in the sense of the core competencies needed to be successful in the role you are in. Your thought process, success metrics, how you spend your time, mindset- your role will impact everything. But is knowing the differences between the two roles(although the responsibilities of the role might vary slightly from company to company) enough for us to make an informed decision? No! We need to get a 360-degree view and understand any misconceptions or expectations that we need to keep in mind as we take the call.

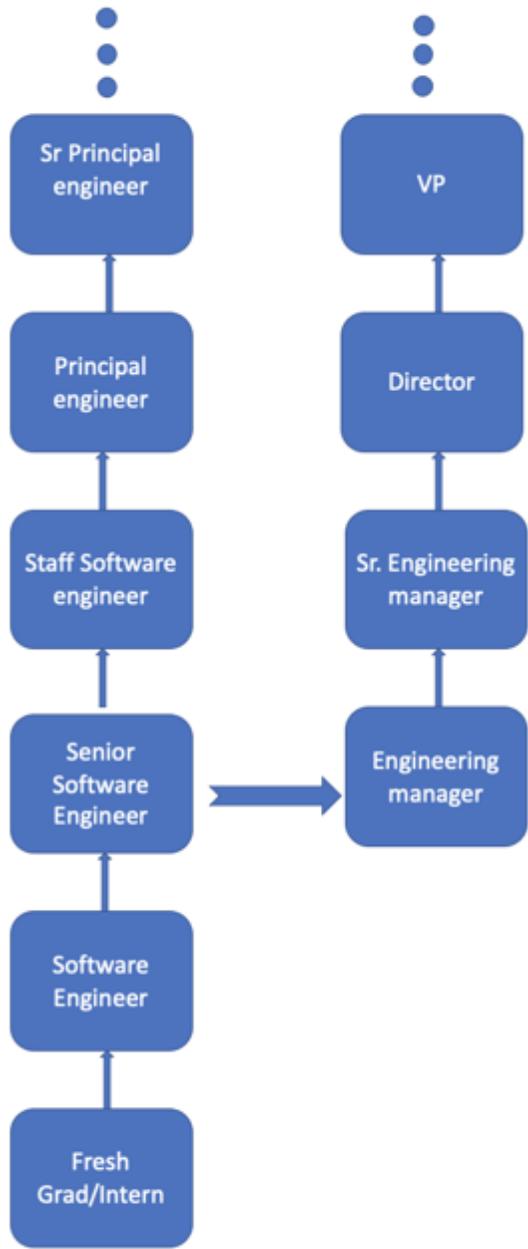
## 2.2 Common misconceptions

There are a few common misconceptions that are widespread in the tech industry? Wait - really? Yes, I will share some common misnomers with you, which will help us align on the expectations of the EM role.

### **2.2.1 Individual contributor to engineering manager is not a promotion**

Traditionally, a move to management is considered a promotion. It is still true in several industries, but not in tech anymore. IC and EM roles are two separate career ladders in tech these days. We don't need to be a manager to grow in our role. Major big companies now have two separate tech tracks, one is focused on growing as an IC, and the other lets you laterally move to EM roles after reaching a particular experience level. As an example, let's look at the following sample career ladder. What is interesting to see is how after reaching a certain level, you can choose and decide whether you would like to continue on the IC track or do a lateral move to a people-focused track- the EM track.

**Figure 2.1 Sample career ladder demonstrates the lateral shift between individual contributor and engineering management roles. As you can see, it is not necessarily a promotion but a lateral career track change.**



The same is true for moving back to IC from an EM role, as long as you continue to hone your coding skills and keep yourself up to date with technology. This is the beauty of the career ladders above (although some small companies might not provide this flexibility), nothing can be entirely a one-way door only, and it provides us the flexibility to choose what we like and are good at.

## 2.2.2 Should you write code in the new role?

One important topic is whether EMs are expected to code. While there is no definitive answer, it's crucial to understand your motivations for pursuing the EM role. Perhaps you are drawn to the people-oriented aspects of the job, the opportunity to shape team mission and vision, or you're seeking a new challenge in people management.

As an EM, coding is not an immediate requirement for success in your role. It's important to be comfortable with the idea of stepping away from coding. However, you can still actively contribute to architectural discussions, allowing you to stay connected to the product and code by designing high-level architecture.

If you are part of a small team where your technical expertise is crucial to set the technical direction and coding is expected of you, you may have limited choice but to continue coding until a suitable replacement is found. Remember, the expectations regarding coding involvement can vary based on individual circumstances. It's essential to align your priorities with the requirements and expectations of your role as an EM.

First-level managers who have recently transitioned from individual contributors (ICs) to EMs may maintain some involvement in coding as long as they effectively balance their roles and responsibilities as EM. However, it's crucial to avoid taking on tasks that could become bottlenecks or single points of failure for the team.

For example, non-mission critical tasks like onboarding a team's service to a tool such as SonarQube (Learn more about SonarQube [here](#) (<https://www.sonarsource.com/products/sonarqube/>)) can be acceptable for an EM to handle if necessary. On the other hand, focusing on implementing and pushing code for a critical project may hinder the team's progress if it compromises timely completion due to increased managerial responsibilities.

As an EM, it's essential to prioritize and find a balance between people management and technical involvement. Attending every code review meeting or engaging in extensive technical discussions may be impossible due to other responsibilities. While you can participate in reading, reviewing, and discussing code, your primary role may not be focused on coding. If needed, you may be willing to assist with minor tasks as a manager, but expectations beyond that level of involvement should generally not be set.

Remember, as a manager, your responsibilities differ from those of an IC. It's essential to fulfill your current role effectively, which includes tasks such as participating in technical discussions and keeping the engineering excellence bar high. However, with the role comes the focus on all three pillars of people, projects, and processes, so you should allocate your time accordingly.

### **Recommendation to reader**

If you as an EM want to keep your skills current and top-notch, do coding on the side. Consider coding for non-profits through UpWork, Develop For Good, Taproot, or find a good SourceForge or FOSS project to contribute to in your spare time. That should satisfy your need to code, and it won't get in the way of your success as a manager or that of your team.

Or, if you want to code, and are considering a matrix management position, look to the technical architect role some companies provide. It allows you to code for prototyping and exemplary projects that other engineers use, but it still has an influential management component(sometimes direct reports or something indirectly by mentoring engineers)

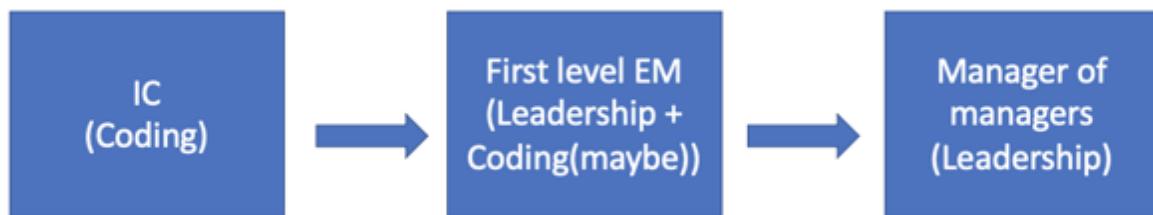
While it is possible to code as a first-line manager, I strongly recommend prioritizing acquiring skills needed for success in your EM role. Trying to fulfill both IC and EM responsibilities simultaneously can lead to burnout. However, this doesn't mean completely distancing yourself from coding. To stay connected to the codebase, actively engage in code reviews, participate in technical system design discussions, and contribute to agile ceremonies

like retrospectives, where you can share your experience and mentor the team. This involvement keeps you closely aligned with the product and emphasizes your importance on the technical aspects.

Participating in company-wide Hackathons or innovation weeks with your team is another effective approach. This allows you to contribute without becoming a bottleneck on critical projects while demonstrating your commitment to innovation. Additionally, writing scripts to automate report generation for the leadership is a valuable task that provides flexibility and avoids creating bottlenecks. It also ensures that you are providing other engineers on the team with opportunities.

Once you have gained comfort in your new role, feel free to allocate some time for coding purely for personal enjoyment. Striking the right balance lets you focus on developing the necessary skills for your EM role while maintaining a connection to coding when appropriate. The next image shows the distribution of skills between coding and leadership varies based on different roles. As an IC, the primary focus lies on coding and technical expertise. On the other hand, if you choose the EM track and become a manager of managers, your main focus will shift entirely to developing and honing leadership skills.

**Figure 2.2 Breakdown of coding and leadership skills based on roles. As an IC, the focus will be on coding, and if you take the EM track and become a manager of managers, you will focus the entirety of your time on leadership skills.**



For middle managers in the following management step, coding, and code reviews are generally not expected. Their focus and expertise are better utilized in formulating strategies for the company and organization.

However, since they possess a technical foundation from their previous roles, they can still actively participate in technical discussions if needed. In the past, some senior managers and VPs in my past experiences engaged in coding for personal growth and staying updated with their coding skills, even building fun projects like a voice assistant skill for their kid.

Whether or not to code in this new role is not a straightforward answer. It depends on how one defines coding and the intention behind it. There are various ways to stay connected to the code and contribute to the team without overshadowing the development opportunities for engineers. As a manager, one of your responsibilities is to ensure the team delivers high-quality software solutions. While coding might not be your primary task, leading the team's technical direction is crucial. This involves more than just writing robust and testable code; it may require developing new skills. Striking the right balance of involvement without undermining the team's trust is vital for an EM.

### **2.2.3 Coding rounds when changing jobs**

Now, if coding differs from what's expected out of our role, why do some tech companies ask EMs to go through coding rounds? Well, as EMs, you are expected to be good at system design and understand the technical tasks to some degree. You were an IC at some point, so you should have a flair for designing large-scale distributed systems with some basic coding expertise. Similarly, companies assess whether you can comprehend and break down a coding problem when an engineer approaches you for help. The goal is not to write the code for them but to guide them toward relevant resources or connect them with other colleagues who can assist in resolving the issue. This approach fosters trust and credibility with your team members and instills confidence in their confidence in being led by a technically proficient leader.

Most companies offer EMs to go through a code review instead of a code writing round as that aligns more with what an EM might be doing as part of their day job. Anyone looking to prepare for such rounds should align with the 'interviewer' to the fact that there is

- No expectation for them to write or review the code flawlessly,
- Pick any language you have used in the past,
- You can make minor mistakes with syntax, and the code might not be executable- that is fine,
- Your concepts can be rusty, as long as your thought process demonstrates you are a problem solver, you are good to go. The examination here is whether you understand the concepts, can speak to architecture and design, and have a skill in communicating. You should be able to talk through the thought process of getting there. Using English concepts instead of a programming language is a fair game.

Several resources provide information on preparing for coding rounds, so that I won't get into those details. Remember the idea of thinking out loud and sharing your thought process with interviewers to ace the interview. Next, I will clarify another widespread misconception about choosing the best engineer in the team as the next EM in line.

## **2.2.4 Every sound software engineer is NOT a good manager**

It is a misnomer to consider the team's rockstar engineer to be the next in line to be an EM. A great engineer only automatically translates into being a great EM if they possess the skills to be a great people manager. Also, some skills are a prerequisite to the role, while others can be gained and honed over time. Remember that if you push yourself too hard to be an EM or someone else, you will lose great IC talent to mediocre EM talent. At the same time, you end up bringing managers that might be micro-managers as they tend to dive into code. This can be counter-productive for the team members and go against showing empathy. Project Oxygen by Google is a great example of the idea at hand.

### **Did you know?**

#### **Project Oxygen**

The research conducted by Google, dubbed Project Oxygen, was intended as a business strategy to train future business leadership to institute best

performance practices and to drive continuous improvement among Google's management team. The outcome of the research was that EMs mattered. Learn [more \(<https://hbr.org/2013/12/how-google-sold-its-engineers-on-management>\)](https://hbr.org/2013/12/how-google-sold-its-engineers-on-management)

As part of the program, the company identified the eight key characteristics that are essential for any manager to succeed in the role. All of them touch upon the people aspect of the role by being a coach to the team, not micro-managing, performance-driven, focusing on results, strategy building, career development, and technical skills to advise the team if needed. If you look at all these characteristics, some can be honed over time, but some should resonate with you as a person, for example, putting people first, and hence only some rockstar engineers are cut out to be successful EM.

### **Myth Buster**

Every sound software engineer differs from a good EM. Both roles overlap from a technical and mentoring standpoint but also diverge in terms of the people aspect, and hence it's essential to look for the right blend of skills to identify the next EM.

### **Successful IC ≠ Successful EM**

When transitioning to an EM role, it's essential to assess each engineer individually before assuming they will excel in the position. A friend of mine experienced this firsthand. Initially, he set a "no micromanaging" norm, expecting everyone to thrive, but later realized that his A player struggled without micromanagement. To adapt, my friend conducted 1-on-1 meetings with each team member, recognizing that not every engineer thrived under the same management style. Some needed clear task assignments, while others preferred autonomy. This experience taught him the importance of tailoring management approaches to individuals.

He found a compromise by providing task management training to the A player, leading to their swift return to high performance. This taught him that managing a team requires flexibility, as no one-size-fits-all solution

exists. Understanding individual needs and preferences is crucial, emphasizing open communication and regular 1-on-1 meetings to address issues proactively. Evaluating engineers before assigning them to the EM role is essential, so let's explore a framework for this evaluation.

## **Evaluating whether an engineer has the potential for an EM**

To evaluate whether the team's technical lead or senior engineer will make a good potential EM, firstly, it's important to understand what is expected in the other role. They should be ready for the new role and challenges. Let's explore a four-step framework I have used before for assessing engineers that have worked.

- 1. Gauge interests** - The first step is to determine if the individual contributor (IC) is genuinely interested and passionate about becoming an EM or if they are merely going through the motions, perhaps due to their tenure with the company. Understanding the underlying reasons for wanting to transition to management is crucial. We will explore this further in the next section, exploring the intentions and motivations behind aspiring to be an EM. One effective approach is shadowing someone currently in the EM role for a few weeks. This allows the IC to gain firsthand experience and better understand the role.
- 2. Assess skill set** - Next step is to assess if the IC (which can be you) possesses the skills needed to be a successful EM. A few of those skills include
  - leadership skills to influence others
  - motivating others and identify their strengths and weaknesses
  - leading by example
  - Authentic, demonstrating integrity, honesty, and kind
  - demonstrating empathy and emotional intelligence toward others
  - can or learning to delegate by staying away from a 'doer' mindset
  - can help resolve conflicts, with an unbiased mindset, within and outside teams
  - Effective communicators so can get their thoughts across to others
  - Help build a strong team culture
  - believes in 'we' over 'I' and can recognize the work of others

- Is a strong team player and gets excited with people around them
- can help with hiring and building high-performing teams
- can understand the ‘why’ behind the actions and believes in correcting errors. The idea is to allow room for ‘fail fast’ and ‘recover fast,’ not ding team members for trying something out, even if it doesn’t work. But at the same time, if they do the same thing three times, that’s a different conversation altogether.
- a patient listener and is open to feedback

It's important to note that sometimes employees have already demonstrated the required skills mentioned above. However, if not, as EMs, we can assign them tasks or responsibilities that require using those skills to assess their capabilities. Not all engineers may have had the chance to showcase these skills, and even asking about relevant situations can shed light on their abilities.

During this phase, a helpful tool is a gap analysis document. It involves listing the competencies expected from an EM based on the company's career leveling guide (or using the skills mentioned earlier for assessment if no specific guide exists). This document should be collaborative, objective, and data-driven to minimize subjectivity. It helps identify skill gaps and allows for a plan to address them.

### **3. Understanding of company mission statement and organization**

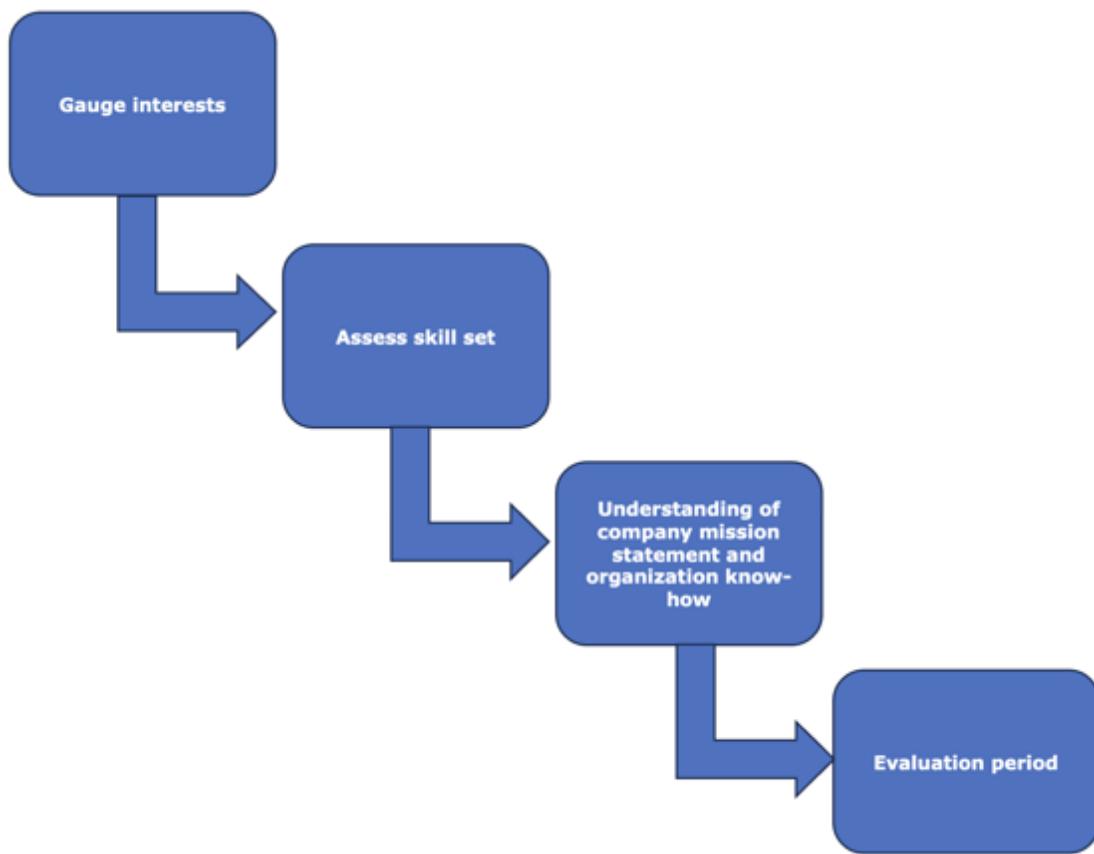
**know-how** - The individual must comprehend the company's mission and establish a clear team charter. They should be capable of defining team-level Objectives and Key Results (OKRs) while promoting an innovative culture to retain talent. It's important for them to take an interest in understanding the overall business and eventually contribute to strategy development. Developing business acumen is crucial as one progresses in management, and providing business acumen training can be helpful if the ICs need it. It is an essential aspect of advancing the management ladder.

### **4. Evaluation period** -In the final step, we will evaluate the individual by offering them a chance to experience the role before committing. For more details on this transition period, please refer to the three-phased approach shared below. By considering the outcomes from this period,

we can make a thoughtful decision that benefits both the individual and the company, creating a win-win situation.

Below is a visual representation of the four-step framework we learned above.

**Figure 2.3 Four-step framework to evaluate whether an engineer will make a good EM. At the end of the last stage, one can evaluate based on the experience and the feedback gathered during the various steps.**



I had a senior engineer, David, on my team who showed interest in the EM role. He had been with the company for 5+ years and was the team tech lead. Through our 1-on-1 discussions, I learned that he was inclined toward becoming an EM.

To assess David's suitability for the role, I gauged his interests and motivations. He enjoyed mentoring and coaching engineers, taking pleasure

in seeing their growth. I then evaluated his skill set and found that he met most of the requirements for an EM, such as mentoring others, displaying leadership in cross-team projects, and driving operational excellence initiatives. However, there was an issue with his communication, as he sometimes used technical jargon with non-technical stakeholders, making it difficult for them to understand the problem. In our 1-on-1, I addressed this concern and discussed improving his communication. We partnered him with a Marketing manager as a mentor to help him convey information succinctly to non-technical audiences. Additionally, we identified relevant LinkedIn courses to enhance his communication skills.

During our meetings and team gatherings, I observed if David understood the company's strategy and goals, focusing not only on the 'how' but also on the 'what' and 'why' aspects of the business. After ensuring alignment and addressing any skill gaps, I offered David a chance to try the EM role on a trial basis. We created a 90-day plan where he acted as an EM for a subset of my team, owning a small charter and executing projects in that domain. He exceeded my expectations and successfully transitioned into the EM role.

In this section, we discussed common misnomers in the tech industry and how to approach them. The following section will cover the fundamentals of a successful IC to EM transition, emphasizing the importance of intentions, motivations, and maintaining a learning mindset.

## 2.3 Intentions and Motivations

Being an EM requires a deep understanding of motivations, both your own and those around you. If you're considering the role, challenge yourself to answer why you want to be an EM. This pause allows you to uncover your true motivations. Ask yourself, "Why do I want to be an EM at this point in my career?"

Suppose your answer revolves around enjoying the people aspect, mentoring, fostering team growth, providing vision, and having the aptitude for leading others. In that case, you're on the right track and more likely to

succeed. However, if your answer involves seeking more control, higher pay, escape from coding, following a natural growth path, or chasing titles, there may be better moves for you. In such cases, reconsider your decision and consider your long-term career goals carefully. Money, titles, power, and advancement are valid considerations, but they alone won't make you a good manager or leader. If you anticipate being in the EM role for the next 15+ years, take a long and thoughtful approach to ensure it aligns with your career aspirations, particularly if you're considering the transition early in your career.

Some of the questions you should answer for yourself to understand the motivations and intentions are:

- Why do I want to be an EM?
- Do I enjoy coaching and mentoring others to help them grow?
- Do I like to be involved in strategic ambiguous situations?
- Do I like to remove roadblocks for the team members?
- Can I trust others for the work and help verify?
- Am I open to feedback from others?
- Will I be honest, authentic, and kind when things go wrong?
- Will I keep my integrity, honesty, and kindness when I know information that might affect my team?
- Do I like to drive alignment between the people in the discussion?
- Do I believe in building a positive work culture?
- Am I a strong advocate of equality of opportunity?
- Do I like to be available to help others when they need me?
- Will I miss coding daily?
- Do you still want to manage people, even if the title on the wall does NOT say manager? If you can say it's no big deal if the title doesn't state that, then you're probably in the right mindset.

Remember David from the previous section, who was the team lead and wanted to move to an EM role, yes, David answered all the questions above, so he has clarity in his head of his motivations for moving to EM. This helped him to ask himself if he would enjoy the EM role as much as he feels he would enjoy it from the outside.

If you answered yes to most of the above questions and are open to exploring the rest, you are on the right track. You have positive intentions required to be an EM. On the other hand, remember that alternative paths exist that are equally good. You can explore being a product manager, technical program manager, or architect or stay as an individual contributor and continue what you are good at coding. Maybe one of these other career paths is a better fit for you- so take your time and settle for what works for you. Now that you understand the driving factor for yourself, in the next section, let's look at a three-phase approach to initiate and do the actual transition.

## **2.4 A three-phase approach for individual contributors to engineering manager transition**

Now that we've explored the responsibilities and considerations of an EM role, are you ready to try it? It's important to make an informed decision about whether this role is the right fit for you and if you're prepared to take it on. You might wonder why I use the term "try" instead of fully committing to the role. Well, many tech companies offer the opportunity to try out the role before making a total commitment. This allows you, as someone transitioning, to have a clear understanding of what you're getting into and whether it aligns with your expectations and goals. Simultaneously, it allows the company to assess if you are the right fit for the position. Think of it as a basketball match. You want to practice and prepare before the final game day to sharpen your skills and ensure you're ready for the challenge. Similarly, by trying out the role, you set yourself up for success and avoid committing to something that you might regret later on.

To try the role, you need to prepare before taking up the role, during the transition period, and have a continuous learning process post-role transition. This is precisely how I have divided the IC to EM transition in a three-phased approach, as shown in the image below.

**Figure 2.4 A three-phased approach of going through an IC to EM transition**



Before you are an EM

- Align with your manager
- Find mentor(s)
- Start small
- Talk to more people who have done the journey or plan to

Transition Period

- Leverage company levelling guide
- Learn about your team members
- Be an active observer
- Active participation
- Fail fast, learn fast
- Be a learner
- Retro at end of transition

Post transition period

- Measuring success
- Identify your management style
- Outcome not as intended
- Change of plans in future
- Grow in the role

Now, if you feel strongly that you have a knack for putting others before yourself and prefer managing people over code, let's see how you can prepare yourself for the IC to EM journey in a three-phase approach and go through them step by step.

### **2.4.1 Phase I- Before you are an engineering manager**

Moving to an EM role can feel intimidating. You want to ensure you only jump into the fire with the right supporting gears. We proactively prepare ourselves for the role in the first phase, keeping a growth mindset. Below are some ideas that can come in handy as you prepare yourself for the challenges ahead as a new EM before even being in the role.

#### **Align with your manager**

It is important to align with your manager to set the foundation for a healthy and productive working relationship, benefiting both you and the organization.

- **State your intentions and set expectations** - First, meet with your manager to set clear expectations through open and honest communication. Collaborate on creating a 30-60-90 day plan with achievable milestones as an EM for your probationary period. Use this

time to learn and assess whether the role suits you while identifying areas needing improvement. Work with your manager to address any skill gaps, ensuring you can effectively communicate and present your thoughts to engineers, stakeholders, cross-functional partners, and leadership, as this is vital for success in the EM role. Remember that being an EM entails accountability, so establishing alignment with your manager is key to success.

- **The time is right** - I aimed to transition into engineering management when I had a strong technical foundation. Starting as a software engineer and progressing to a senior engineer, I shared my aspirations with my manager early on. This allowed us to align and identify opportunities for my growth toward the EM role. We regularly discussed my goal in our 1-on-1 meetings and worked on projects that provided leadership opportunities. This transition only happens after a period of time, requiring patience and trust in the process. Timing is crucial; balancing strengthening technical skills and learning leadership aspects is essential. Establishing a timeline with your manager, often around 90 days, helps evaluate your adaptation and performance in the new role. This ensures clear expectations and facilitates the decision-making process for the final official transition. Additionally, discussing potential outcomes if criteria aren't met ensures you won't be penalized and can continue as an individual contributor if necessary.

## Find mentors

Finding mentors as you transition from an IC to an EM is crucial for your success. Look for multiple mentors, both inside and outside your company, who have experience with the transition and can offer valuable insights. Utilize mentorship platforms like [Plato](https://www.platohq.com/) (<https://www.platohq.com/>), [GrowthMentor](https://www.growthmentor.com/) (<https://www.growthmentor.com/>), and [FastTrack](https://fasttrack.firstround.com/) (<https://fasttrack.firstround.com/>), or attend local meetups and networking sessions to connect with like-minded professionals. Leverage your LinkedIn connections to reach out to potential mentors with a clear purpose and defined expectations for the mentorship.

Gaining various perspectives is beneficial at this stage of your career. Different mentors may have diverse views on their EM roles, helping you understand various management styles and find what resonates with you. Avoid relying solely on your current manager as your mentor to gain a broader understanding of the world.

Work with your manager to explore different scenarios and real-world examples to stretch your thinking and prepare for the challenges of an EM role. For instance, imagine managing an underperforming employee who stops responding to messages or emails. This exercise helps you practice situational leadership and handle unexpected and ambiguous situations.

By seeking guidance from mentors and working through scenarios, you'll build a strong support network and develop the skills necessary to transition to an EM role successfully.

## **Start small**

When moving from an IC to an EM role, it's essential to begin with manageable tasks and set realistic time-bound milestones. I started by mentoring summer interns and being an onboarding buddy for new hires, which allowed me to experience the people aspect and assess my enjoyment in helping others grow.

Collaborating with my manager, I selected projects involving multiple teams and stakeholders to gain exposure to cross-functional collaboration and stakeholder management. My manager provided support and guidance throughout this process. To expand my skills, I identified common pain points and implemented process improvements for operational excellence. Gradually, I increased my involvement in hiring interviews to explore my interest in recruitment compared to coding. Additionally, I shadowed my manager, learning from their observations.

A significant step in my journey was taking on the role of acting manager during my manager's paternity leave. This experience involved 1-on-1s, project execution, and various EM responsibilities. Based on my

performance during this period, my leadership team gained confidence in my abilities, leading to the official transition to the EM role.

Starting small, learning along the way, and gaining experience through practical steps prepare you for a successful transition to an EM role.

### **Talk to more people in the role already or who are thinking about the role**

This is your exploration period about the role and whether you would enjoy being an EM. So my recommendation would be to talk to as many people as possible- people who are in the role, people who are thinking about the move, and if you can get people who did the IC to EM transition and then moved back to being an IC to get a 360-degree perspective. Especially people you would soon be working with - your potential new peers.

Some common questions you can ask the mentor or people in the role could be:

- What are the key responsibilities and challenges of being an EM?
- How can I effectively balance technical expertise with managerial responsibilities?
- What skills and qualities should I develop to become a successful EM?
- What strategies can I use to build and maintain a strong engineering team?
- How can I effectively communicate and collaborate with stakeholders, both within and outside the engineering department?
- What common pitfalls or challenges do EMs face, and how can I navigate them?
- Can you guide me in setting goals and expectations for my team members?
- How can I foster a positive and productive work culture within my team?
- How can I support the professional growth and development of my team members?
- Why do they enjoy leading and managing the teams?

- Are there any recommended resources, books, or courses that can further enhance my skills as an EM?

Remember, these questions serve as a starting point, and you can tailor them to your needs and circumstances. The goal is to gather valuable insights from your mentor's experiences and expertise to aid in your transition into the role of an EM.

Most of the management challenges boil down to dealing with those skeletons: the hard HR rules you must follow or the things you didn't expect to know or deal with. Your new peers can often shed light on these kinds of things and expose them. It will make you think twice. Now this might not be as straightforward as it sounds. Maybe we have someone in our network who might be a good fit but is super busy with their schedules, or perhaps we don't have someone in our immediate network, so it requires us to cold message someone externally. Whatever the case, here are a few underlying things to keep in mind:

- Be very clear and specific about the reason for reaching out and what's the outcome you expect
- Be transparent about the time commitment expected, although starting from a quick 15 min chat is always better to understand if there is a mutual fit. Also, give them enough time to respond to your request.
- Also, offer something to this relationship so that it is a two-way learning
- If they agree to guide you- you are all set. If you do not get a response or end up with a no- don't feel disheartened; treat this as a learning experience and head to the next person in your network

You can use some of the ideas shared above to conduct these informational interviews, which will help you prepare mentally for what you are getting into and get wisdom from people who have walked the talk.

## **2.4.2 Phase II - Transition period**

Now that we prepared ourselves proactively before the transition began, we have a fair idea of what to expect with the information we gathered. Also, you should have clarity on the timeline of the transition period (usually ranging between 90 days to 6 months) and what success looks like in this new role, in alignment with your manager and leadership. Below are some tools that can help you in your transition period.

## **Leverage company leveling guide**

Big companies typically have leveling guides that define roles and responsibilities at different levels. For startups or early-stage companies without such guides, you can create a similar framework to define roles objectively. During your transition, it's crucial to utilize this framework to maintain a document outlining your work and how it aligns with the expected responsibilities for your level. This way, you'll clearly understand what's expected of you and the criteria that will be considered after the probationary period, usually around 90 days. During my own transition, I tracked my work and how it related to the role expectations. For example, managing summer interns fell under the people management aspect of the role. If your company lacks a clear leveling guide or it's confusing and inconsistent, communicate with your manager to ensure a clear understanding of expectations and document your work accordingly. Creating a living document and validating it with your manager will greatly benefit you and ensure that everything is well-documented and aligned.

## **Learn about your team members**

As an EM, it's important to establish rapport with your team members through one-on-one meetings. Use this time to understand their motivations and career aspirations. Sometimes, team members may hesitate to discuss their goals with you as a new EM. To foster open communication, promote psychological safety, ask relevant questions, and create a comfortable environment. During the early stages of your role, focus on listening more and providing advice selectively. Building relationships with cross-functional partners is also crucial. Recognize that you and your team

members are transitioning, and strive to establish a partnership as you navigate this new phase together. Avoid adopting an IC mindset when addressing team issues or mistakes. Instead, focus on helping the team learn from errors and foster an environment of empathy and trust. It can be challenging for former ICs to handle this aspect. You can cultivate an atmosphere of openness and collaboration by being transparent about your own fallibility and demonstrating a commitment to earning their trust.

### **Be an active observer**

As soon as you enter the EM transition phase, there can be an urge to change things or fix things you feel do not work for the team. Stay away from immediate actionism, be patient, and observe from behind without making assumptions. The initial phase is sensitive, so you don't want to come across as disruptive and not open to understanding other perspectives. Ensure to tell the team that you are not here to change everything up; instead, you value input from others and are looking for ways to improve by learning and not just fixing things. Document your learnings, and when the right time comes, instead of saying about the problem directly, pose questions to the team during team meetings about what they feel about something that is not working well and gather feedback. Make people part of the process, and you will be surprised by your team's talent. Also, it is one of the best ways to get buy-in from a new team.

### **Participation**

EM roles can be challenging as you sail from one meeting to another. As you try to adjust to the new role, ensure to continue your participation in team sprint meetings, design discussions, or other technical forums. The last thing you want is for people to think you choose an EM role to avoid technology. At the same time, finding the fine line between how hands-on (stay away from micro-management) versus hands-off (stay away from neglecting your team members) you should be will come with time as you iterate and fine-tune your style. Being cognizant of giving the autonomy your team engineers need to learn and grow. State your expectations to them

about your level of participation in their technical roadway so everyone is aligned.

## **Fail fast, recover fast**

This role is new for you- it is ok to fail sometimes. As you go through this journey, look for 360-degree constructive feedback from your engineers, manager, and cross-functional partners. A regular cadence of feedback will help avoid any surprises and also help you iterate faster if something is not working out. This gives you time to change your action and understand what works best for you and your team.

## **Be a learner**

This is your time to learn and grasp as much as possible, so crave learning. Block focus times on your calendar to learn new skills. Whether this means going through online courses (Linkedin learning, Pluralsight, Youtube courses), reading books(Harvard Business Review series and leadership books), or materials(blogs, in-room classroom training) provided to you by your support network- take it all in. Hone your communication skills, learn to delegate, and use your time wisely to build relationships with partners and stakeholders. If you feel like you have a gap, or something significant needs to be added, ask your manager or mentor for support and training. Most companies support this, especially for new leaders, and have it as part of an onboarding program for new EMs. Even if it's unavailable as part of onboarding, still ask for it, and you shall be rewarded (I hope).

## **Managing your own team where you were an IC till yesterday versus a different team**

Management comes with its own challenges. Now within the same company, there can be three common scenarios we will discuss below. In all three cases, effective communication is the key to success and is common for all cases.

- **You are transitioning into an EM role for a different team** - If you're tasked with managing an already established team different from your current one, consider the following approach. Recognize that the team has its own style, processes, and dynamics. Put yourself in their shoes and use the 'Observer' tool to understand their current setup without being seen as outsiders. To bring in changes, do so gradually and in small chunks to avoid disruption. Involve the team in the process, asking relevant questions and making them feel part of any changes you introduce. Avoid pointing fingers or badmouthing previous leaders. Take time to do your homework on the team by reviewing their wiki and documentation and speaking with the tech lead to understand their pain points and desired improvements.  
Recognize that there might be a learning curve in understanding both the people and the technical aspects of the team's charter. Engage in team-building activities to foster rapport, especially as a new leader. Consider offsites or casual lunch/coffee chats to connect with your team members. Additionally, set clear expectations with your team, especially if your management style differs from the previous leader's. Solicit feedback from the team on what they expect from you as their new manager.

Suppose you're also transitioning to a new company while taking on the new EM role. In that case, the challenge might be greater as you must familiarize yourself with the company's business model, culture, and more. Utilize your manager and support network to plan your first 90 days with clear expectations and alignment.

- **You have been tasked to build a team from scratch** - As a new EM, you've been tasked with building a team from scratch, which has its ups and downs. On the positive side, you can shape the team culture, but on the downside, you need to gain experience in team-building. Start by answering why a new team must set a clear scope and mission for the hiring process. Engineers may become frustrated if they need to see a clear roadmap and how their work adds value to the company. Leverage your network and work with a recruiter to find the best engineers for the team.

Seek support from your manager, mentors, and allies to learn from their experiences and avoid reinventing the wheel. It's essential not to

hire all at once; instead, take gradual steps and regularly assess the team's needs to work lean. Building teams can be exciting, fun, challenging, and risky, but it may be more suitable later in your management journey after gaining more experience in other EM roles. Starting with other roles and gradually honing your skills before building a team from scratch might benefit your growth as an EM.

- **You are transitioning into an EM in your current team where you were an IC-** Transitioning into an engineering management role can be challenging, with potential outcomes ranging from success to difficulties. Being patient throughout the process is important, as team members may take time to accept you as their manager. Build trust before officially assuming the role, and communicate your willingness to learn alongside the team. Remember that acquiring a manager title doesn't automatically result in immediate influence and change. Developing trust with the team requires effort and time. Be mindful of any biases you may have carried over from your previous IC role, and treat all team members fairly. While it's tempting to maintain close friendships within the team, be cautious, as your new role entails providing constructive feedback to everyone. Strive to avoid favoritism and approach interactions with an unbiased mindset. Some ways to help you keep an unbiased mindset could be:
  - **Self-awareness:** Stay aware of your own biases and prejudices. Recognize that everyone has biases, and addressing and challenging them is essential.
  - **Equal treatment and opportunity:** Treat all team members consistently and equally, regardless of their background, gender, race, or other characteristics. Provide equal opportunities to all. Base your decisions and actions on merit, performance, and objective criteria.
  - **Active listening and communication:** Allow every team member to express their thoughts, ideas, and concerns. Listen attentively and without preconceived judgments. Encourage open and honest communication. Communicate expectations, goals, and performance evaluations to all team members.
  - **Objective evaluation:** Evaluate team members objectively. Use standardized assessment methods and avoid subjective judgments

or personal biases.

- **Continuous learning:** Stay informed about unconscious biases and attend training sessions or workshops on diversity, inclusion, and unconscious bias. Educate yourself and strive to improve your own understanding and behavior.

Remember, creating and maintaining an unbiased approach requires ongoing effort and vigilance. Regularly reflect on your actions and decisions to ensure fairness and equality within your team.

Keep in mind that this transition may take a lot of work. Some team members might question your suitability for the role or feel that they deserve it instead. Be open and honest with everyone, but remain aware of the potential for sabotage from within the team. Ideally, your growth will be met with excitement and support, but it's important to be prepared for possible challenges.

In all cases, see the brighter side, which is a great learning opportunity as a new manager to hone your leadership skills. Set up a candid communication channel with your team members and look for early candid feedback, as time is of the essence. Share learnings with your mentor to help balance and clarify some opinions and results.

## **Retrospective at the end of the transition period**

As you come close to the end of your transition period, do a quick retro with yourself to see where your head's at.

Some questions to ask yourself at this point are:

- What areas were frustrating to you as part of this EM transition journey?
- What areas were satisfying for you?
- Do you still feel connected to code more than people?
- Did you enjoy seeing success through others' eyes?
- What will your peers/people report to you say about you?

- Did you do justice to the team members in your transition phase?
- How was the team's morale before and after you took over?
- Did you recognize the talent around you?
- If you were to go back in time, what would you change about your transition period?
- Where are your gaps? What do you need to address those gaps (perhaps training)?
- Was I honest and authentic with the team when provided feedback?
- Did I keep my integrity and honesty despite my biases?
- Did I give my best, my 100%, at all times?

Answering these questions will help you gain clarity and self-awareness during your transition. Take this opportunity to reflect on your aspirations and whether the role change will bring you happiness and satisfaction. It's important to remain grounded and open to the possibility that you may not have all the answers. Seek support from peers and mentors, engaging in discussions about the challenges you face. Celebrate your achievements, stay humble, and acknowledge areas where improvement is possible.

### **2.4.3 Phase III - Post-transition period**

Ok, so now is the moment of truth; we are done with going through a transition phase of an EM. We have asked ourselves some questions to see how we evaluate the role and ourselves against them. Now is the time to set up a meeting with your manager and discuss feedback, where you share how you felt during the transition period and gather 360 feedback. The goal at the end of this meeting is to ensure you both have clarity if the official transition is being made or not. If yes, decide on the timeline it will be in effect and plan for the remaining time. If not, identify the gaps(if you are willing to move in the future) or align on returning to your previous role, which you are great at, being an IC.

### **Measuring Success**

The way I think about measuring success at this stage is to see how in the pre and transition phase you felt by answering a few questions that are a

combination of subjective and objective criteria like:

- Were you comfortable all the time?
- Were there instances where you felt challenged?
- Were there instances that gave you a sense of satisfaction?
- Were you able to identify the mission and vision of the teams?
- Do you feel content with bringing a multiplier effect to the team?
- What percentage of projects were you able to deliver successfully in a timely manner?
- What signals does your peer feedback from engineers and cross-functional partners suggest?
- Were you able to meet or exceed the team's development velocity goals?
- What was the attrition rate for the engineers during your time as EM?

The goal should be to understand if, pursuing this role for the next few years, you will just survive or thrive in the role. Also, remember to take a pause and celebrate your journey so far. Of course, you have challenged yourself by making a career change, whether temporary or permanent.

## **Identify your management style**

This is an opportunity to discover your management style. Learn about different management and leadership styles, such as autocratic, democratic, delegative, transactional, transformational, servant leadership, and hybrid styles. **You don't have to align with just one style; instead, find your own hybrid approach.** During the transition phase, observe and shadow your manager and mentors, gathering practical knowledge to complement the theoretical understanding. Reflect on the management styles that resonate with you based on your experiences. It's normal to evolve and adapt your style over time. As you enter the post-transition phase, you should have a rough idea of your preferred management style, but remember that learning and growth in this role are continuous. Managers should always strive to learn more and stay open to new ideas.

## **What if your transition did not go as intended?**

No decision is perfect. If the EM transition doesn't work out for you during the transition phase, don't view it as a failure. Instead, see it as a valuable learning experience that helps you understand yourself better and identify your preferences. It's okay if management isn't the right fit for you now; it might be a possibility in the future. Embrace the opportunity to stay in your current role as an awesome engineer and reinforce your capabilities. Use this time to learn and grow, and remember the value of this experience in gaining self-awareness and knowing what you truly enjoy.

Reach out to your mentors for support and motivation. Instead of complaining about the outcome, acknowledge the opportunity and remain positive. Remember that staying as an IC is not a demotion; it's a chance to shine and determine what you want to do next. Consider exploring other paths, such as becoming a product manager, technical program manager, or architect, or simply continuing to excel as an individual contributor and honing your coding skills. Take time to find the right fit for your career as you control your journey.

### **What if I like the role now, but my plans change in the future?**

Change is inevitable, and it's important to recognize that transitioning from an EM to an IC role (or vice versa) is possible. It's okay if, after some time, you find that staying hands-off from coding bothers you. The key is to continue honing your coding fundamentals, regardless of your chosen role. Moving to an EM role should be seen as something other than a one-way door, but it's worth noting that the longer you stay in a management role, the harder it may be to transition back to being an IC, especially after a significant amount of time. The main concern when switching roles or transitioning back is how it may be perceived by others, particularly in terms of job titles and resumes. Be prepared to explain your motivations and reasons for making the change.

I have witnessed engineers who successfully transitioned to EM roles and thrived for a period of time, while others eventually decided to return to being an engineer or architect. It's important to have a strong answer that aligns with your motivations and desires. Whether you prefer to provide

technical capacity support while leading engineers or explore different roles, the perspective gained from trying both positions can be valuable, and you can leverage the learnings from one role to enhance the other.

### **Continue to grow in the role**

Even as an official EM, the learning journey should always continue. Take advantage of resources such as your company's learning platforms, LinkedIn courses, Harvard Business Review blogs, and mentorship circles. Engage with local engineering leadership groups to discuss challenges and learn from others' experiences. If such a group doesn't exist, create one and invite others to join. Keep a growth mindset and seek out learning opportunities that challenge you.

In my experience six months into my EM role, I faced the challenge of managing an underperforming team member. It was a new and challenging situation, but I sought guidance from mentors and my manager to learn the necessary processes and mindset to address it. Maintaining a learning mindset allowed me to grow and handle the situation more effectively.

Many companies invest in their employees' growth by offering tuition reimbursements and opportunities to attend leadership conferences. I have been fortunate to benefit from such perks, including a workshop called the "Confident Club" organized by my previous employer. The workshop focused on building confidence and improving public speaking skills. Following the "see one, do one, teach one" approach, leverage your mentoring network and consider mentoring others once you feel confident. Don't limit yourself to office work and networking alone; explore leadership opportunities in various contexts, such as community service or organizing public events, to continuously learn and develop a well-rounded skill-set.

## **2.5 Challenges of a new engineering manager**

As humans, we tend to resist change as it can cause discomfort and bring its own set of challenges. As a new EM, you will be responsible for the success of the team and the team members as a whole. As I said at the start, this

journey will be full of bumps, peaks, and valleys. Let's look at common challenges that someone new to EM can face are shared below, along with my thoughts on what you can try to overcome them.

### **2.5.1 Coding over people**

New EMs can be urged to do what they have been good at so far and feel attracted to coding and their job as an IC. Instead, you are now responsible for team velocity, hiring, and feedback loops, which can all be new and overwhelming initially. Suppose you continue to focus on your previous role responsibilities. In that case, you are taking away a golden time that should be used to learn more about people around you, mentor them from a career perspective (not everyone wants to grow at the same speed), and enable your team to bring out the best in them. For example, what makes a junior engineer happy versus a senior engineer can be very different.

#### **What can you try to do?**

- Initially, let coding take a backseat, spend time in 1-on-1s to know people's aspirations and motivations, and invest learning in leadership skills.
- Do ad hoc coffee chats with the team
- Stop being a tech lead for the team
- If you like to code, participate in hackathons and innovation weeks where your work is not a bottleneck and also sends a positive signal to the team
- Pick the career leveling guide for the EM of the company and start putting your actions against each of them to ensure you are channelizing your energy in the right direction
- If you have a fear of missing out (FOMO) on technology, take learning courses and read blogs on the latest technology, you don't necessarily have to code for work to stay up to date on technology

### **2.5.2 Setting clear goals and expectations**

You are now tasked to set clear goals and tenets for the team. The team will look up to you as their leader, so it's of utmost importance to work with leadership and bring transparency and clarity to the goals and OKRs (Objective and Key Results) at the team level. This will give a vision to the team members and ensure they feel together as a cohesive team and are valued. You will have the deal with the ambiguity around competing priorities. Using the SMART approach for goals and expectations will help for career conversations, which we will cover in the managing people chapter.

### **What can you try to do?**

- Build relations with cross-functional partners and stakeholders
- Work with leadership and cross-functional partners to help set up team strategy and document it
- Shadow your manager/mentors in org-level meetings as they setup org level OKRs
- Ask questions and share what you can with your team in the spirit of transparency

### **2.5.3 Struggle with delegation**

This is one of the classic challenges reported by new EMs. You have a ‘doer’ mindset and like to keep things under your control or are not open to trusting others. The reality is you are one person, and you can't be omnipresent in all meetings. Also, too much into the weeds can signal micro-management to your team members, which goes against building trust. As an EM, you must continuously strive to scale and make self-organizing teams by using the potential of the team members around you. This will also develop a strong culture of trust in the team.

### **What can you try to do?**

- Talk about delegation with managers, mentors, and others around you to learn about their journey.

- Start with delegation by creating a win-win situation for you and the engineer. Trust and verify
- Learn more about your engineers for tech discussions, but actively participate by asking the right questions instead of the urge to do it yourself- such as ‘What is the risk with approach A versus B? How will the system handle fault-tolerance? etc.’
- In Chapter 5, you will learn in-depth about why you need delegation and a step-by-step process that can come in handy

#### **2.5.4 Success Metrics change**

As an EM, the definition of success changes, your individual success is no more a metric, instead your success is through the success of your team. Even the definition of productivity will change for you. There can be times that you feel not as productive by the end of the day because you attended meetings but feel you did not do ‘actual’ work.

##### **What can you try to do?**

- Bring awareness and understanding of the success metrics in your new role and how they are different from your previous role as an IC
- Company leveling guide and your manager will be your support system in this
- Measure your success against the EM and team success metrics

#### **2.5.5 Conflict resolution**

We as humans are conflict-averse. People are much more challenging than coding because the same code will always give you the same output. At the same time, everyone will be different and have different motivations. Failure to acknowledge a conflict and address it in a timely manner can have a catastrophic impact.

##### **What can you try to do?**

- Identify the conflict and acknowledge it as soon as possible

- Approach the situation with an unbiased and data-driven mindset
- Be a patient listener
- Upskill yourself to handle difficult conversations
- Facilitate the resolution, but don't force your decision on others
- Don't defer, act in a timely manner
- I will discuss conflict management in detail in the chapter on Managing people.

## **2.5.6 Speed versus quality dilemma**

You want to ship things faster as the ship's captain, but at the same time, quality is of the essence, such as tech debt introduced, on-call burden, operational excellence, etc. Also, there will be times when what an engineer wants to work on versus what is needed for the business to grow may differ.

### **What can you try to do?**

- Keep a data-driven mindset when evaluating trade-offs
- Stress the engineering culture you would like to build in the team
- Aim for alignment by bringing everything together from the product, eng, and cross-functional aspect
- Lead by example, wherever possible
- Lay emphasis on work-life balance, and hence learn to defend the team's WLB

## **2.5.7 Acknowledge and recognize others**

As the EM, you are not required to be the smartest in the room or have answers to all the questions. On the contrary, you need the smartest engineers and your team to help solve engineering problems. It is the mindset of 'we' versus 'I.' Recognize the talent around you.

### **What can you try to do?**

- Reward and recognize the individuals around you. This encompasses not only your engineers but also the cross-functional partners to

- promote a culture of recognition
- Represent the engineering team in good light and find forums to bring visibility to the team- perhaps demos in org-wide sessions, shoutouts on Slack, or company recognition tools

## **2.5.8 Keeping promises from previous leaders**

This has been a common challenge for new EMs to tackle any promises from previous leaders. This can include promises of promotion, working on critical projects, etc. Additionally, make sure not to promise what you aren't sure is possible or you don't have control over. For instance, you can't say, "This company will never do layoffs." You don't know this, but no one in the company knows, so that no one can guarantee this.

### **What can you try to do?**

- Instead of simply imitating the previous leader, being authentic and honest with your team members is important. Embrace your unique style and approach, bringing your own perspectives and ideas to the table.
- During handover, do a three-people meeting for each team member transitioning under you, with the invite list being - you, the previous leader, and the team member. That way, you can avoid surprises later on.

## **2.5.9 Managing people and performance**

As an EM, one of your key responsibilities is managing people and performance. This involves overseeing and directing employees' work to meet expectations and contribute to the organization's goals. Regular career conversations and helping engineers build career roadmaps are important aspects of this role. Managing both high performers and under-performers is crucial for team success.

As an EM, you may be involved in recruitment, training, and employee development. Managing people also means giving feedback, and handling

HR and immigration issues, which can be challenging, especially for first-time EMs. Therefore, it's vital to develop strong people skills. Honing your people skills is an ongoing process, and you should continuously strive to improve and adapt to changing situations and employees' needs. Below are some ways you can work on improving your people management skills.

### **What can you try to do?**

- Managing with empathy and compassion involves understanding and supporting employees' needs, actively listening to their perspectives, and creating a trusting and open work environment. By demonstrating genuine care, managers can foster motivation and a sense of value among employees. It's important to lead by example and set clear expectations while prioritizing the well-being and development of the team.
- Providing timely and constructive feedback aligned with individuals' goals and aspirations fosters clear goal-setting, candid communication, and personal/professional growth opportunities, ultimately enhancing productivity and trust within the team.
- There will be instances where you will have more information in the role as an EM relative to being an IC - both in terms of people and projects, so learn to filter out what's needed and be mindful of sensitive information. It is your job to maintain confidentiality wherever needed and consider what can and cannot be shared with the individuals. Maintaining trust and strong relationships with your seniors, peers, and team members is essential.

### **2.5.10 Time management**

Time is of the essence, and managing it well is an art. We want to feel valued beyond our meeting time. You will have to learn the art of context switching as you will be pulled in many directions. Feelings of not having done 'actual work' might creep in.

### **What can you try to do?**

- Maintain a to-do list of tasks with a priority to get control of the calendar. If it's impossible to maintain a to-do list, use some system that allows/helps you to manage priorities and organize ideas.
- Block focus time slots on your calendar
- Promote a no-meetings day, one of the days in the week for the engineers. Try following that for yourself as well.
- Use time management tools to be organized. Example- Clockwise. You protect your calendar, so have lunch breaks and focus time blocks to protect and manage it well.

We will dive deep into more tools to use for time management in a later chapter.

### **2.5.11 Documentation**

As a coder, you write, you write code and technical design documents. As an EM, you will be writing strategy documents, promotion packets, and process documents, to name a few. Document writing can be challenging, especially for folks if the language you are writing the document in is different from your first language.

#### **What can you try to do?**

- When writing the document, keep the audience at the front and center. If it's a document for a non-technical audience, avoid technical jargon.
- Ensure the document clearly states the purpose as to why it should be read. TL; DR(too long; didn't read) is a common way to put at the start to align why the document exists
- Remember to value the importance of good document writing. Something I have used personally is the concept of '[Deep Work](#)' (<https://www.amazon.com/Deep-Work-Focused-Success-Distracted/dp/1455586692>) from the book by Cal Newport to plan my document writing slots. This means blocking long continuous stretches of writing to do productive deep work away from distractions.
- Try not to beat about the bush. I have seen documents that use flowery language and can be subjective to interpret. Keep the objective and

data of your document-driven

- Proofread, proofread, I cannot emphasize enough the importance of reading the document at least twice to ensure it makes sense, is grammatically correct, and will cater to the readers. In my document writing journey, I took online courses through various platforms to hone my written communication skills

## 2.5.12 Communication

As an EM, you will spend most of your time in meetings. Communication is a critical aspect of getting your thoughts to others. New EMs need help to convey the gist to others as they sometimes get too technical or in the weeds of the discussion. Both written and verbal communication are the skills to focus on as you plan to thrive in the role and eventually move up the ladder.

### What can you try to do?

- Hone the art of effective communication.
- Learn to know the audience.
- Indulge in talks, whether presenting in internal meetings or external conferences (perhaps a Toastmasters group). You can start small by finding a Meetup group near your area and then taking it from there.

Being an EM is a challenging role with multiple hats to wear at different times. Now that we have learned about the IC to EM transition and what to expect as we step into the role, let's learn how to do our part and help others around us going through such a transition. The next section will be helpful if you are an experienced EM helping your ICs go through such a transition or a new EM which will eventually help their ICs grow.

## 2.6 Assisting the transition in roles

How about helping others around you going through an IC to EM transition? This can be core to your role to help and develop other EMs around you. This IC may be in your reporting structure or your mentee.

Here is a framework you can use to support this individual so they are put up for success:

- **Understand their motivations, strengths, and weaknesses** - This will provide a reference point and ensure your energy is channelized to help them keep their aspirations in mind. Use one-on-one conversations and the gap analysis document to gain insights into them.
- **Help set achievable and time-boxed goals** - If you are working directly with this person on the transition or indirectly (maybe as a mentor), work with them to set up milestones in the journey. This can involve helping build a 30-60–90 day plan and sharing ideas of how to nail down the goals by sharing your own personal examples and experiences. Keep them time-boxed so they can be evaluated at the end of the timelines discussed.
- **Help them build a network** - As an individual helping this person to go through the IC to EM transition, help them understand the importance of teamwork and collaboration. One thing that has worked for me is I always ask my ICs going through the transition or someone who is a new EM hired in the role to do the following:
  - Write the name of all engineers you will manage on a piece of paper
  - Set up a 1 on 1 with each of the engineers on this list (15 mins should be good)
  - Ask them about the pain points in the team and what is working well. Next, ask them to give you at least 3 names of people they recommend you should talk to. If they currently do not exist on your list of people to talk to, add them and continue.
  - By the end of this exercise, you will have formed a graph of people you would have met and learned in detail about what works well and what's not in the team. This is a good stepping stone to help build a network in the company.
- **Open communication channel with feedback and recognition** - Ensure to build a conducive communication channel with this individual. This should be a bi-directional flow of communication where you get together, share progress on milestones, recognize the progress, and share constructive feedback.

I once assisted a senior engineer in transitioning to an EM role. We discussed their aspirations and motivations and created a transition roadmap with achievable milestones. During the process, I noticed they sometimes used technical jargon when communicating with non-technical stakeholders, causing ineffective communication. To address this, they were paired with a marketing mentor to gain a non-technical perspective. I also provided coaching on communication techniques, like gauging the audience's understanding and creating an interactive forum. Online communication courses were recommended too. The engineer successfully bridged the communication gap with dedicated efforts and support and transitioned to the EM role.

### **What do other leaders have to say about it:**

“I first moved into an EM role (managing a couple of teams) at a small startup where I was in the first EM role the CTO had created. Before that, he had directly managed all the engineers. It was a learning experience for me, and very much learning while on the job since we didn’t have much formal training. I think the empathy and TLM (technical lead/manager) pieces came naturally to me, and the new aspects (end-of-year reviews, compensation, attrition, etc.) easily filled up many of my conversations with my manager, essentially just in time when the need arose. I often felt a bit nervous and somewhat out of my depth during that time, but I also felt like my manager (and cross-functional partners) had my back, which typically eased my concerns. “

**~ Richard Frank, Senior Software Engineer (ex-manager), Two Sigma | (ex) Robinhood**

“I have always been a people person. I believe in the system of Model, Coach, and Care, which I feel is a core part of management. If my senior engineer shows interest in being an EM, I would first understand the motivation behind the same. If they are in for the right reasons, I would support their choices with the right coaching.”

**~ Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

“Coding is not a primary skill, but it is something that EM should be able to do when required. I do code reviews, troubleshoot issues, and sometimes do code. A solid understanding of system architecture, data structures, and programming language has always helped me communicate with my developers, understand their challenges, and take an appropriate mitigation plan. EM should focus on developing a wide range of technical and interpersonal competencies. They should know the latest tools and technology, development methodologies, and industry best practices. Other important skills are Risk management, Effective communication, Dealing with conflicting and ambiguous requirements, and attention to detail.”

**~ Devika Ahuja, Technology Leader | Strategist**

“I am an avid reader, so I have a daily routine to read books, blogs, articles, etc, to keep myself updated. My favorites are Medium, Reddit, GeeksforGeeks, List Apart, Google Developers, and Dzone. Other than that, I have registered on platforms like ClassCentral, Coursera, Udemy, and LinkedIn, learning to keep myself updated with the latest tools and technology.”

**~ Devika Ahuja, Technology Leader | Strategist**

“I had great EM role models during my career, and I looked up to them for all sorts of guidance, not just work-related. This initial fascination with the role led me to research more about it and understand how critical the role is towards the success of an organization. It is common knowledge that engineers leave managers, not companies, and I wanted to be the reason talent stayed with my organization rather than going elsewhere. Moving to a different role always feels like stepping outside one’s comfort zone. Deciding to move from an individual contributor to an EM role felt the same way to me. I was uncertain if I could leverage skills acquired as an IC in this new role. My manager helped me overcome these inhibitions by creating a plan for the transition. This allowed me to check things off a list, which helped me feel more confident that I was heading in the right

direction. I was also mentored by experienced EMs who provided invaluable input. Apart from all the people who helped me make the switch, my company's policy regarding trying out career opportunities without facing any backlash minimized the uncertainty involved in this process, which helped propel me to success as a transitioning EM.”

**~ Adish Agarwal, Director, Software Development at Audible, Inc.**

“If my senior engineer comes to me to be an EM, I will celebrate that they've identified a goal for themselves that we can work towards. I would also discuss why they want to be an EM. My goal would be to help them assess how their interests match the EM role or whether they see it as the only path beyond senior engineering. I would acknowledge that the role needs to be a match between the company and the individual filling that role, so we should: assess what behaviors or skills lie in the gap between that tech lead's current demonstrated abilities and the EM role, identify goals/tasks to close that gap; keep our eyes open for an available EM opportunity (if it's not already present)”

**~ Richard Frank, Senior Software Engineer(ex-manager), Two Sigma | (ex)Robinhood**

“As a senior leader, there are a few ways I stay up to date with coding and technical skills, such as doing code reviews for my team and asking nonblocking questions when I don't understand something. This gives a side effect of allowing engineers to educate their managers and is great for team building. I tinker at home on personal coding projects. None of these have launched, but scoping them adjacent to my team's technology helps me keep up to date. I can't enumerate the number of times that I have solved a problem in a personal project that has become relevant a month later in my “real job.” My entire web experience was built this way, and now I'm coding full-stack features for my team between meetings.

What I could improve at is keeping up with the newest news about all the frameworks, languages, etc. While I casually read tech news and frequently have moments of “Oh, that's cool,” I've never been a “technology for technology's sake” kind of engineer. The best I do is file those snippets

away so that when I have a problem that could use that solution, hopefully, it is remembered.”

**~ Nathan Bourgoin, CTO- Alakazam Inc, Technical Advisor, Engineering Leader**

“When one shifts from IC to being a manager, the challenges are countless, especially if you’re promoted within the team. To succeed, you need to not only maintain the relationships within the team (developers/technical folks within your own team) but also maintain positive relationships vertically(managers, direct reports) as well as horizontally(with your colleagues at the same level, most likely the managers from the business or other teams. Team building activities like fun meetings and outings can enhance team cohesion. Virtual or in-person networking with colleagues is a great way to build relationships and make an impact.”

**~ Vindhya Avvari, Cloud Integration Manager, University of Southern California**

“EMs coding depends on the team and Organization. But I feel they should be close to the code and continue coding at least on their own as garage projects. I keep myself updated by taking online learning and training. I try participating in Hackathons, brown bag sessions, and design workshops. I also monitor the support and escalation threads to identify quality issues and health monitoring.

The choice to make the transition was organic. But the transition itself was challenging. Getting out of the habit of jumping into solutions and design versus delegating was the hardest. Delegating is not a natural one for me, so I had to force myself to stop advising while letting the team figure it out. ”

**~ Rajakumar Sambasivam, Delivery Manager at Microsoft**

“Many people transitioning to managers are seen to do lots of IC work by themselves, causing less time to spend on people and roadmap-related activities. This is mainly because they have yet to learn to influence and

lead the people under them properly. The easiest path with the least stress is to do things independently, and many of them default to this route.

It's ok to code during the transition from an IC and the first few months. This way, the change will feel more gradual. But once you start picking up more EM-oriented responsibilities, you should be spending less and less time coding and instead spend time on coaching, motivating, and supporting the team under you. This is going to be super helpful for the team since you are caring for them as well as good for your personal growth also. Since you have mostly decided to go forward with the management route and the higher you go, less coding-related responsibilities you should have. You can code on side or personal projects to keep the coding skills active.”

**~ Sarin Panambayil, Principal Engineer at Yahoo**

“To succeed as an EM, you essentially are working for your team. Your people are always your #1 priority, you must ensure you provide them opportunities and resources to be their best in the role. Every team member has a unique definition of success. If you can create an environment where you bring all the help and remove the roadblocks to success, that will make you a great EM.

I lead the Pre Sales Engineering Team in the Networking industry, so I am often asked if the EM would have to be hands-on, and the answer is No. Your role as EM is to help the business needs into technology if you can do that without being hands-on, you are good. I’m not completely denying the need for it, but in my view, it is not a mandatory need. It is more of a good practice that increases the chances of success if used rightly. There are two pitfalls you have to consider:

- 1 It will take time that you can spend on key activities for your role as a leader.
- 2 Would the quality of your technical work create comparison and disturb the emotional dynamics of the team?

I would rather spend time bringing in new ideas and awareness of new technologies in the team instead than demonstrating my hands-on technical skills.”

~ **Sumit Kumar, System Engineering Manager at Cisco**

“Read, read, read! If you're not coding daily, you should stay abreast of emerging trends and technologies, look at best practices, explore new architectures and ideas, and learn from your peers. In addition, the best way to keep your skills up is to teach what you know: speak at conferences, panels, seminars, and meet-ups. Even within your own department, if nothing else.”

~ **Bruce Bergman, Manager at Lytx**

## 2.7 Stop & Think: Practice questions

1. What brought you to management?
2. Why did you make/or want to make a change from IC to EM, and how does it align with your long-term goals?
3. Share your experience of doing the IC to EM transition.
4. What skills do you look for when growing ICs into EM roles?
5. What are your success criteria to be a successful EM?
6. Are there constraints in your current role stopping you from transitioning? If yes, do you plan to combat them?
7. What are the new problems or opportunities you are looking for from your new role as EM?

## 2.8 Summary

- Individual contributors and engineering managers are two separate career tracks with different skill sets as a requirement.
  - An IC contributes to the team and the organization but needs to manage other people. If you are a software engineer, then yes, you are an IC.

- An EM is someone who manages other engineers. This can include direct supervision of software engineers, or if you are a middle manager (Director), you will be a manager of managers.
- In the tech industry, there often needs to be more clarity about the role of an engineering manager. One common misconception is that transitioning from an individual contributor (IC) role to an EM role is solely a promotion rather than a lateral career move. However, this assumption may only sometimes hold true.
- Remember, career changes are not one-way doors. There are always opportunities for learning and skill development, regardless of your role. By prioritizing personal and professional growth, you can succeed and advance in your career, even when changing or transitioning to a new role.
- Another misconception is whether an EM can code in the role. EM can always stay in touch with the code and product by actively participating in technical design discussions or code reviews without necessarily writing the code themselves.
- Every sound senior software engineer only sometimes translates to being a good EM. It requires skills such as empathy, technical acumen, and others that can be acquired, such as organization during the transition process from IC to EM.
- One needs to evaluate the skill sets and ensure that it's the right fit for you and that you will enjoy the role for years.
- Engineering managers driven by title, power, or a desire to distance themselves from coding may struggle to succeed. Conversely, EMs who genuinely enjoy people management, are passionate about helping others grow, and prioritize building trust and credibility with their team are likelier to excel.
- Using a three-phased pre-transition, during, and post-transition approach for IC to EM move gives one more time and space to understand the role. At each phase, we evaluate our experience and validate it against what we were looking for and what was expected of us to make an informed decision.
  - Phase I - Before you are an EM, take your time to understand the role and responsibilities to align with your manager, find mentors

and build a support network and start small with activities like managing interns to get a flavor of the role

- Phase II - This is the transition period where you leverage the company leveling guide (if it exists) or set up alignment with your manager regarding expectations in the role. Also, you use this time to set up a rapport with team members.
- Phase III - post-transition period, where you try measuring success from the previous phase, identify what your management style is, and introspect if this transition is what you would like to pursue(or explore alternative career paths or go back to the IC engineer role)
- There are several challenges faced as a new EM.
  - Putting code over people
  - Setting clear goals and expectations
  - Struggle with delegation
  - Conflict resolution
  - Change in success metrics
  - Speed versus quality dilemma
  - Acknowledge and recognize others
  - Keeping promises from previous leaders
  - Managing expectations etc.
- Helping others around us going through such an IC to EM transition will help bring a multiplier effect, help build networks and open a candid channel for communication and feedback with them.

# 3 Managing people, teams, and yourself

“It’s not about money. It’s about the people you have and how you’re led.”

~ Steve Jobs

## This chapter covers

- The importance of managing people, teams, and yourself
- Manage people with a focus on career conversations, support, and conflict resolution
- Manage teams by focusing on goals, morale, and trust
- Manage yourself as an engineering manager to identify opportunities and acknowledge mistakes

The art of management thrives on a comprehensive understanding of what is being managed. Success in any role requires a clear awareness of the elements under your purview. Just as an army can secure victory by comprehending its own strengths and personnel, understanding both its own team dynamics and those of the opposing forces to formulate strategic moves, a chief of the army must possess the ability to manage themselves effectively in order to lead the troops. Similarly, as an engineering manager (EM), it is paramount to acquire the skills to manage the individuals within your sphere, the teams, and yourself. Effectively managing people involves understanding their motivations, providing support, and fostering their growth. It builds trust, promotes employee engagement, and enhances team productivity. By recognizing individual strengths and aligning them with suitable responsibilities, managers can maximize the potential of their team members, leading to higher job satisfaction and retention.

Managing a team involves several key aspects, including defining goals and vision, addressing morale issues to create a positive work culture, setting clear expectations, building trust, managing remote teams, and evaluating success. A well-managed team encourages collaboration, synergy, and innovation, enhancing team performance and goal achievement.

Furthermore, a strong team dynamic promotes knowledge sharing, continuous learning, and mutual support.

For an EM, self-management is crucial for effective leadership. It encompasses self-awareness, self-regulation, and personal development. By prioritizing self-care, efficiently managing time, and continually upgrading skills, managers can maintain their well-being and perform at their best. Effective self-management is a positive example, inspires others, and empowers managers to make informed decisions, adapt to challenges, and drive personal and professional growth.

In summary, successful management of people, the team, and oneself results in higher employee satisfaction, increased productivity, and improved organizational outcomes. It fosters a positive work environment, nurtures talent, and contributes to long-term success and growth. In the following section, we will delve deeper into the understanding of managing people, teams, and personal growth and adaptation as a leader.

## **3.1 Managing People**

Mastering the art of managing people is essential for any manager, including EMs. It encompasses being present for your team, comprehending their motivations, maintaining regular communication, fostering their growth within their roles, and resolving conflicts to facilitate mutually agreeable solutions, among other aspects. As each drop contributes to forming an ocean, each team member with unique strengths plays a vital role in building a strong team and contributing to its success. Let us now explore the key considerations to bear in mind when it comes to managing people.

### **3.1.1 Career Conversations**

Career conversations, also known as one-on-ones, constitute regular meetings and discussions in a private setting between two or more people to discuss an agenda. In the context of this discussion, the agenda is a career conversation between the manager and the employee to help with

information sharing, relationship building, career growth conversations, and feedback. This is also a great opportunity to discuss the highlights and lowlights around projects or career development opportunities with each other and have a candid bi-directional flow of information. One-on-one is a great forum to connect with your team members to discuss career opportunities and help coach them to come up with actionable feedback for areas of growth opportunities. A good framework for EM to discuss career goals is using the SMART framework.

### **Did you know?**

The SMART framework is a common structure to set clear, achievable goals for a project or product launch. This helps keep assumptions aside and sets clear guidelines for delivering the goals. It stands for

**Specific** - The goal needs to be specific regarding what needs to be performed, who will perform it, what steps will be taken, and what support will be provided for achieving them.

**Measurable** - The goal needs to be quantifiable and measurable. For instance, if we would like to reduce the number of trouble tickets or customer escalations, then the goal would be to reduce trouble tickets by 20% so it is easily measurable.

**Achievable** - You want to ensure the goal is realistic and achievable in the desired time frame and with the resources provided. For a project to be delivered, it is important if the deadline is in two months or six months and if you are investing 10k dollars or 1 million dollars. Hence the goal should be set up to be achieved in the given time frame. This is like a true reality check.

**Relevant** - This means the goal should be relevant regarding why we are doing it. We must understand its intent and why the team and company spend those resources on the goal or project.

**Time-bound** - Regarding project success, we need to clarify the time boundaries of the project. This helps answer when the team starts working on the project and when the anticipated end date. This helps to track the timelines and ensure the team works towards a decided end date of completion.

The SMART framework can help set short-term and long-term goals, helping better understand the motivations and context of the goals.

As an EM, you are responsible for career conversations and ensuring your high performers are challenged and moved to the next level when they are ready. At the same time, the low performers in the team are given the support they need to help them improve their performance. A gap analysis document(also referred to as a career growth document) is a good way to help team members analyze their performance and understand the gaps that exist. The document can be a basic row-column format sheet where the rows have the information on the competencies required in the role, and the columns have the situations that demonstrate those competencies using the Situation-Behavior-Impact (SBI) format.

### Did you know?

SBI (<https://www.ccl.org/articles/leading-effectively-articles/closing-the-gap-between-intent-vs-impact-sbii/>) is a format to express an overall scenario to provide detailed context. Situation refers to the overall outline of the conditions that existed, behavior refers to the set of actions taken by team members, and impact refers to the outcome of the overall behavior in the situation. As an example, let's look at a sample SBI scenario.

**Situation** - As part of a new product launch, multiple promotion capabilities were requested by the product to offer the best discounts to customers. There was no existing promotion capability or tool that existed.

**Behavior** - Xavier stepped up and deep-dived into the scenario. He then analyzed the various industry tools that could be leveraged, analyzing their pros and cons. Next, he ensured each engineer understood the full scope of

the project and worked on the technical design. Time and again, he collaborated well with the cross-functional partners to ensure the timely delivery of the project.

**Impact** - His contributions led to a 10% increase in company revenue and a timely delivery of the new product. Further, it helped reduce customer escalations by over 50%.

Above is a simple example demonstrating a complete project scenario broken down into the SBI format for clear understanding.

Such documents can help identify the gaps that exist between the intent and the impact. Also, a note, sometimes the competency list can be out of sync with the realities of day-to-day work. As the EM, ensure the list is consistent for all employees at each level and there are clear distinctions between role capabilities and expectations. A competency matrix which is a list of competencies, how an individual demonstrates them and the progress made against each can help analyze the gaps for a person. This helps the EM and individual identify opportunities to address the gaps. Let's look at a sample competency matrix that represents a gap analysis document template.

**Table 3.1 A competency matrix representing a sample gap analysis document**

Competency	How do you demonstrate it?	Status(Todo, In Progress, Done)
Technical skills- Employee should be adept at coding and system design skills. Experience	Situation #1: Dave was tasked to deliver a project to build a continuous data streaming service due to the missing mechanism.	Done

with large-scale system designs	<p><b>Behavior:</b> Dave stepped up to understand the problem statement and did a deep dive into the architecture design. He worked with senior engineers in the team and developed a technical design document detailing the pros and cons of each approach. Next, he implemented the project under a tight deadline.</p> <p><b>Impact:</b> His efforts led to a timely launch of the streaming service and helped reduce message latency by over 80%.</p>	
Participating in technical events	Situation #1: <>	In Progress

	<p>Behavior:&lt;&gt;</p> <p>Impact: &lt;&gt;</p>	
Sharing learnings and expertise with cross-functional partners/influence cross teams	<p>Situation #1:&lt;&gt;</p> <p>Behavior:&lt;&gt;</p> <p>Impact: &lt;&gt;</p>	In Progress
Help improve operational excellence metrics for the team	<p>Situation #1:&lt;&gt;</p> <p>Behavior:&lt;&gt;</p>	TODO

## Impact: <>

Engaging in career conversations provides an excellent opportunity to discuss individuals' future goals and potential internal mobility positions, allowing them to grow and expand their skill sets. It is also a suitable time to explore the training and resources that can assist them in further refining their abilities. Facilitating connections between mentors and mentees is another valuable aspect that promotes learning and personal development. Lastly, career conversations offer a platform to ensure the well-being of team members and address any concerns they may have. This can include how they are doing in general, how life is going, and anything they want to share with you. As for checking project status, it is best suited for project status meetings, stand-ups, or progress meetings. Now, let's delve into how as an EM, you need to strike a balance in the support you provide to both junior and senior engineers within your team.

### **3.1.2 Supporting junior versus senior engineers**

As EM, you will manage people with various skills and tenures spent in the role of software engineering. The career growth of team members is of utmost importance to EM. There is no one size fits all approach for managing a junior engineer and a senior engineer. Of course, the overall support EM you provide will be similar in the sense you will spend time with the members one-on-one to understand their short-term and long-term goals, motivations, strengths, and opportunities. You will then work with them to set up SMART goals as discussed in detail previously in section 3.1.1. Let's dive deep into how the support might differ for a junior versus senior engineer.

Junior engineers are early in their careers and might be more energetic, passionate, and moldable at the same time. Also, the type of problems and opportunities to keep them challenged and engaged might be relatively easier to find. Now being new to the industry, they might require more

attention, guidance, and coaching around growing their skill sets. This might mean you, as EM, must be prescriptive sometimes. For example, you have a junior engineer in your team where the team is tasked with writing a new service from scratch using the AWS technology stack. Now the engineer might need help with resources to get started and ramp up on the technology, plus you might have to encourage them to think about doing an AWS certification in parallel to attest to the knowledge they learn in the process. Further, you might suggest they share the knowledge through technical talks or sessions with the team. At the same time, junior engineers will need your support to be paired up with mentors to help them get technical guidance. They can be like a blank sheet where you can help them succeed by giving them an appropriate model for the rest of their career.

A senior engineer in the team has been in the technology sector for a while and might have dealt with larger-scale problems. They can be rigid sometimes and might have stronger opinions on things. They would love more autonomy, and the opportunities to keep them challenged and motivated will also be a little difficult or rather more complex in nature. Also, for senior engineers, you would like them to have a mentor, of course, for their growth but also encourage them to guide mentees. For example, you have a senior engineer in the team, and you, as EM, have challenged them to think about an organization-wide problem regarding configuration management for the backend services. Your teams basically need a kill switch feature in configuration to control features, especially if a deployment went wrong or a project needs to be coordinated for launch across multiple marketplaces. Also, there is no central visualization tool that gives visibility into all the configuration's current state or ability to control them from a single location. This has been a major hurdle during testing and always needed a coordinated roll-out and, in case of failure, a coordinated rollback. This is a big ambiguous problem for your organization and is impacting all the teams. You, as EM, challenge them to take this problem statement and provide the desired resources they need to excel at it. Your senior engineer is excited about the problem and steps up to work on it and find a solution.

As we saw previously, there is a stark difference between the type of support and the type of problems that can excite a junior engineer versus a senior engineer. Juniors are at 1000 feet, mid-level engineers are at 5000, seniors are at 10000, and staff are at 20000 feet. As we grow on the ladder, we need to expand our thinking from the micro to the macro and tend to work less on smaller problem statements and instead pick up larger, widely dispersed complex problems. Of course, everything will be situational and depend on the team member, but the above should give a fair idea that different people based on the tenure, will need different support from you as EM.

### **3.1.3 Conflict between team members**

Team conflict is inevitable and sometimes can also bring up healthy competition among team members, helping them bring the best of their work to the table. Different people bring different perspectives and experiences to the table, and hence conflict is but natural sometimes. As EM, it will be on you to identify a conflict in the team and help your team members navigate through the scenario. Let me share a personal experience to share some learnings.

One of my team was working on launching a revamped version of a specific promotion capability to allow customers to redeem a membership type. We were in the process of deprecating the legacy tool, and our next step was to find a home to set up such promotions. During the technical design discussions, a major conflict arose between two engineers in the team. Jack and Jamie are the two engineers with differing ideas on setting promotions. Jack wanted to build a new user interface and a tool to set up such promotions. His idea was that the newer feature should have a separate setup tool for itself. According to him, this would give us more control in terms of setup and controlling the backend behavior of such promotions. Jamie wanted to reuse another existing internal tool owned by our sister team to be the new place where such promotions are configured. His idea was this is similar to the type of promotions setup from the tool owned by the sister team, plus will have a faster onboarding of markets as our salespeople are already aware of this existing tool. Various discussions

happened between the two engineers but ended up in vain. As their EM, I stepped in to help with the conflict, starting from ground zero and acknowledging the conflict first of all.

The approach first was to be hands-off and observe from behind. I let engineers put forward their thoughts and discuss the pros and cons of each approach to promote open communication. At the same time, I ensured the discussion was data-driven and no one person was bullying or dominating the situation, giving everyone a fair chance to speak. Of course, it did not seem as simple as it sounds, and the discussions did not land anywhere, so next, I set up one-on-one with each of them to listen to both sides and create a safe space for them. This was also a good coaching opportunity for me to help them understand the bigger picture of the situation. I coached through asking questions such as how much the effort will be with your preferred approach, what the timelines would look like, will the solution be reusable, what resources are needed, and pros and cons. With such questions, the engineers were empowered to think holistically, and the engineer who wanted to design a new tool from scratch acknowledged that the proposed idea will be a stretch, keeping in mind all the factors, and would add an infrastructure cost as well. This brought the employees together and converged on one solution, reaching a consensus. As an EM, learning from this conflict situation, I also learned that I can help bring more transparency to my team regarding cost, effort, time, and other key areas of prioritization to keep in mind as they continue on technical discussions. This led me to set up some lunch and learn sessions along with my product manager to share the insights and what factors go into the planning and prioritization phase with my team members.

### **3.1.4 Challenging team members**

Working with diverse individuals can be challenging at times. Managing difficult team members provides valuable learning experiences that cannot be obtained from educational materials alone. To illustrate how to handle such situations, let's consider the case of Dave, a senior engineer in your team. Although technically skilled and high-performing, Dave lacks teamwork and dismisses ideas from junior team members. For example,

when discussing library options for backend service throttling, Dave was absent from meetings and disagreed with the choice made by James, a junior engineer who conducted research and presented his findings. As an EM, it becomes your responsibility to address this issue. You have a one-on-one conversation with Dave to understand his absence and discuss how his behavior affects the morale of junior team members. By providing concrete examples, you show how his actions can impact his own career growth. To address the situation, you pair Dave with James as mentor and mentee, enabling them to learn from each other's strengths and successes. Although navigating this situation was challenging, your coaching efforts led to positive changes in Dave's behavior, such as active participation in meetings and mentoring other engineers.

EMs can also encounter non-technical challenges within their teams. For instance, if an engineer makes an insensitive comment in a group setting after an inclusivity and equality session, your intervention becomes necessary. You need to raise awareness and coach the individual on the impact of their remark while also organizing inclusion training for the team to prevent future incidents. In more serious cases, involving the human resources department may be required to address sensitive matters effectively.

### **3.1.5 Lead by example**

Leading by example is a way of setting expectations with the team members and not just talking about it but showing how it's done to set an example or 'walk the talk.' It is a way to shine light and show the path to the goalpost. Employees trust their leader and follow their steps if a good example is set. We learned in detail about this and how it is an important trait for a good EM(some other skills for a good EM include- caring for the team members, having a clear vision, demonstrating emotional intelligence, having business acumen, and being an effective communicator and delegator, stress on innovation, recognition, and organization). Refer to chapter 1, section 1.2.3, traits of a good EM to refresh the concept.

In this section, we learned about managing people, which is a core competency for an EM. In the next section, let's look at handling a collection of team members that actually make a team.

## **3.2 Managing teams**

Managing teams is one of the core responsibilities and expectations of an EM. The team members make the team, but since each individual is different, once they come together as a team, it is important for you as the captain of the ship to ensure that it sails well. You will be responsible for the success and growth of the team where you need to focus on providing a clear vision to the team, build trust with them, provide the desired support they need to grow and hone their skills, and help manage conflicts. Your teams might not be all in one place and so it is important to be adept at managing remote employees as well. Effective management of teams can bring out the best in the team members and retain them in the company as they feel proud to be part of the team and the company.

### **3.2.1 Defining team goals, vision, and strategy**

One of the most important steps as an EM is to help set the tenets, vision, goals, and strategy for the team. This helps bring different team members together on this shared common mission. Each company follows a way of planning; some use objective and key results (OKRs) to plan, while some can have half-yearly and yearly goals. Whatever the case for your organization, it is important to set the goals keeping in mind “why” we are doing this and “how” we are going to do it. At the same time, several factors will come into play, such as cost, development effort, timelines, feasibility analysis, dependencies, and much more. When setting goals, you, as EM, must collaborate with your cross-functional team to bring everyone on the same page. Next, you will be tasked with putting a plan in place from the project's ideation to the launch. You will work with stakeholders and work on dependency management, plan the sprints, and get buy-in from senior leadership. It will be discussed in more detail in chapter 9- Working with cross-functional partners and chapter 10- Project Management,

execution, and Delivery. In the process, you gather feedback and iterate as needed, helping unblock your team with any blockers. You promote effective communication and transparency by sharing sprint end demos and regular team meetings to keep everyone on the team on the same page. Everything, in the end, needs to tie back to the original goals and vision of the team.

### **3.2.2 Handling morale issues**

When managing teams, the morale and general sentiments of the team members are crucial for the success of the team and also for keeping the individuals happy and for retaining them in the long run. Low morale can be a silent killer in the workplace and can dramatically impact the productivity and performance of individuals. Some common things low morale can lead to are - poor quality of work, missed deadlines of projects, and lack of communication and engagement within the team. Some of the ways you, as EM, can go about navigating this situation is to first of all identify the morale issue and acknowledge the issue that exists. Putting a blindfold on it will do more harm, and hence it is important to act on time. There are several indicators that you can observe to identify potential issues within your team. These signs may include team members consistently missing meetings, showing disinterest or disengagement, exhibiting emotional outbursts, high turnover rates, declining productivity, a lack of accountability among team members, increased burnout, reduced communication and innovation, and noticeable signs of stress among team members, among other things. Once identified, some steps that can be taken to help with morale issues are:

- Organizing team building events
- Promoting shout-outs and recognition to team members
- Using one-on-one forums to identify motivation and coaching team members. These can be frank discussions for folks to speak up. Just be careful it does not become a dumping ground.
- If people are stressed, you can work with them to identify the root cause and help with load balancing of work or support for flexible work schedules

- Work with your HR business partner to understand the reasons behind attrition and take steps to mitigate them
- If an employee feels less challenged, find interesting and challenging opportunities for them to keep them engaged, and give more autonomy if they desire so
- Ensure team members have a good work-life balance, use forums like team meetings to gather feedback and brainstorm ideas with the team
- Create open communication channels with the team members to promote transparency and addressing the pain-points of the employees

Above are some of the ways to navigate morale issues and keep your employees motivated. Next, let's look at building trust and transparency in our teams.

### **3.2.3 Trust and Transparency**

Teams long for a positive work environment that provides them with psychological safety. This helps build trust and a safety net where people can bring their original selves to the workplace. A team is one where employees can trust each other and bring their full potential out. Some of the actions that can promote trust and transparency in the workplace are:

- Candid conversations with employees through one on ones and team meetings
- Providing clarity in upcoming teamwork and team charter so they have a sense of belonging
- Continue to lead your team by example and do not ask them to do something that you yourself won't be comfortable doing
- Communicate in an honest and meaningful way, ensuring conversations are unbiased
- Knowledge sharing through lunch and learn or technical talks shared to help promote knowledge sharing and avoid any bottlenecks in the team
- If mistakes happen, encourage members to treat it as a learning experience and learn from the mistake in a constructive manner. Think

about steps that can be taken to avoid the problem from happening again.

- Be a patient listener; let your team members feel heard with the open communication flow
- Engage employees with regular surveys to gather feedback, sometimes you can also have anonymous surveys so people feel comfortable saying what's in their mind.

With this, let's look at another important factor that comes into play when managing teams, collaborating with other teams to modify their roadmap to fit in something critical for your team.

### **3.2.4 Convince teams on dependencies in the team roadmap**

As EM, you have control over your team's technical roadmap and the charter, but is that sufficient?

Not really, and there will be many times when you need to convince other teams to modify their roadmap to account for some dependencies the teams might have on each other. These are situations that really require good preparation at your end to share what you need and why it is important to help convince the other party why the work is equally important for them as well.

Let's take a scenario, as the EM managing the platform team for a car rental company, you are responsible for revamping the company's legacy promotion capabilities. Currently, the company runs online accessory promotions where customers enter a code to receive a free car accessory, which they pay a royalty to the accessory providers. However, during the revamp process, it became clear that the royalty functionality should be separated from the promotion system and integrated into a dedicated royalty system. Since your team lacks expertise in royalties, you reached out to the royalty team and data warehouse team for assistance.

Initially, there was resistance from the royalty team, but you organized a kick-off meeting to discuss potential synergies and explore how the teams

could align the roadmaps. During the meeting, the EM for the royalty team and you discovered that the top priority for the year was to revamp the royalty calculation system and transition to a third-party tool. Recognizing an opportunity, you drafted a one-page business value document outlining the urgency of integrating your use case into their roadmap. You emphasized the financial burden of maintaining the current infrastructure and proposed incorporating our royalty calculation requirements into their ongoing revamp efforts.

Furthermore, you presented a couple of options on how both teams could proceed, highlighting the benefits and potential cost savings of collaborating on this shared project. The royalty team agreed to include your use case in their current technical roadmap, seeing the value in addressing both initiatives simultaneously. To demonstrate your commitment to the collaboration, you offered an engineer from my team to assist with the development work. By leveraging the expertise of both teams and aligning the efforts, you and the other EM aimed to streamline the promotion and royalty systems, reduce maintenance costs, and improve overall efficiency for the company.

You negotiate a trimmed-down version for phase one to unblock your team and create a fully-fledged feature for phase two. This helps unblock the team and removes royalty from the promotion system. It's a common scenario you may encounter at work, and navigating through it requires certain key essentials:

- Earn the trust and build rapport with cross-functional partners and teams by being a patient listener and understanding their motivations.
- Lead by example in all situations. Don't ask others to do something you wouldn't do yourself. If you want other teams to follow certain production deployment guidelines, pilot them within your own team first and have metrics ready to demonstrate their benefits and impact.
- Reward and recognize members of cross-functional teams to encourage positive behavior. Give credit and share recognition to foster collaboration and a culture of appreciation.

Now that we learned a few simple ways to navigate convincing other teams to modify the roadmap, let's steer our discussion to understand how to manage remote teams.

### 3.2.5 Managing remote teams

Remote teams, including hybrid teams where some employees work from the office part of the week and the rest work remotely, have become increasingly common in the post-pandemic world. Remote teams consist of employees in different geographical locations, time zones or work from home instead of a central office. Establishing clear team norms is essential for effective collaboration and avoiding subjective interpretations, particularly in remote and hybrid settings. One of the significant advantages of remote teams is the ability to attract talent from around the globe without being limited by physical proximity. However, managing remote teams also presents certain challenges, such as budget constraints, skill gaps, and difficulties related to time zones.

Overall, it is crucial to establish clear guidelines and expectations for remote and hybrid teams to foster smooth collaboration and ensure that all team members can work together effectively, regardless of their physical location. Some ways you can ensure to manage a remote or a hybrid team well are:

- **Clear purpose and vision** - Clear goals and vision ensure the team is aware of what is the common mission and what brings them together as a team. It also instills a sense of togetherness and cohesion where people can build a culture of trust and transparency, with, of course, your support as EM of the team.
- **Infrastructure support** - Setting up a functional work-from-home environment can significantly enhance the productivity of remote employees. A well-equipped workspace, including a suitable desk, chair, and other necessary equipment based on the job requirements and company budget, plays a vital role in supporting remote work. Some companies have adopted a "hotelling model," where desks are not assigned but can be booked by employees when they plan to be in

the office. As an EM, you can facilitate opportunities for team members to sit together or have lunch meetings, fostering a sense of togetherness and collaboration. In some cases, companies may cover the cost of internet connectivity, which is often appreciated by employees as a valuable perk. This gesture can boost employee confidence and instill a sense of pride in the company they work for. By ensuring that employees have the necessary resources and support for remote work, companies can promote a productive and positive remote work experience.

- **Regular check-ins** - We discussed the importance of regular one-on-ones with individual team members to understand their motivations better. With virtual employees, this is more important as it is one of the few opportunities to catch up in a one-on-one setting with the individuals in the team and show empathy. Similarly, do regular check-ins with team members in a team setting to promote social connectivity. Also, don't keep all check-ins super formal, rather give room for others to connect with you on an informal level and be candid. Some EMs do not understand the purpose of these check-ins and start making the team members feel they are being micromanaged. It is important to treat them the way they want to treat each other.
- **Team events** - This is the fun part team members usually look forward to. Conducting team events help promote employee morale and connectedness. Some team event ideas that you can try are doing virtual escape rooms, food-making/noodle-making classes, sip and paint events, remote lunches, or even a regular happy hour can help boost team morale. Virtual hackathons are another great way to unite team members to promote innovation and thinking outside the box. Use these meetings to promote the culture of video calls so people can put a face to things and feel connected, especially people who live alone. At the same time, once in a while, maybe every quarter/or half year, try to get the team together physically in one location so people can actually connect in person. This can give a good balance between the virtual and physical worlds.
- **Clear work boundaries** - One significant challenge of remote work is establishing clear boundaries. In a physical office, leaving your desk or office space indicates the end of the workday. However, in a virtual

setting, it becomes more complex as communication primarily relies on chat messages and calls. Notifications become the primary means of reaching out, leaving individuals at the mercy of their team's chosen communication tools. To address this challenge, it is important to communicate and set clear expectations regarding your working hours. If you collaborate with teams across different time zones, consider adding your official working hours as a message in the communication tool. This allows others to be aware of your availability and work hours. However, it is equally important not to feel obligated to respond to every message outside your working hours. Constantly attending to messages during personal time can lead to meeting fatigue and burnout. Most communication tools offer features to set working hours and mute notifications beyond those hours, providing a way to create a healthy work-life balance. By being transparent about your working hours and utilizing the available features of communication tools, you can establish clear boundaries, prevent burnout, and maintain a sustainable remote work routine.

- **Emphasis on documentation** - A good set of robust documentation can really help the team to grasp concepts, have clear ownership, and have solid troubleshooting runbooks in times of high-impacting customer issues. In virtual environments, documentation is the key to learning and ensuring less dependency on other team members if the documentation quality is good.
- **Training** - Each team member needs your support in terms of being a sounding board, providing training and resources to help grow their technical or social skills. With remote new hire onboarding, stress on more virtual training has been laid. Ensure to give new team members the desired time, support, and space to learn and come up to speed as they join your team. Several resources can come in handy such as company internal resources, [Coursera](https://www.coursera.org/) (<https://www.coursera.org/>), [Pluralsight](https://www.pluralsight.com/) (<https://www.pluralsight.com/>), [Linkedin Learning](https://www.linkedin.com/learning-login/) (<https://www.linkedin.com/learning-login/>), [Udemy](https://www.udemy.com/) (<https://www.udemy.com/>) to name a few.
- **Flexibility of work** - Your team needs your support as EM to empathize with them and provide the work support they need. Let's say someone in your team has to pick up their kids from daycare every day

at 5 pm. Now that time might be non-negotiable for them, and once they have conveyed to you about the timing, try to accommodate the request and avoid keeping meetings in that slot. This can be especially beneficial for individuals with family responsibilities, as they can work around their children's schedules or other obligations. Similarly, if there are time zone differences or any culture differences, it's important to be cognizant of them and help accommodate them to your best to set the right culture for the team. If there are offshore teams, strive for a good crossover of hours each day to maintain a balance.

- **Mental health days** - Mental health is a critical aspect of overall health and mental well-being. Mental health issues can have significant negative impacts on a person's life and their ability to function on a personal and professional front. Nowadays, companies are proactively thinking about mental health and providing mental support resources and, at the same time, the idea of mental health days. This can be practiced as one of the days, let's say in a month or quarter, where the entire team will take a day off to focus on health and well-being, or each team member can decide a day in the month/quarter that works better for them. Whatever you and your team choose, it is important to acknowledge the importance of mental health and how it can lead to an improved lifestyle.
- **Recognition**- Recognizing the work of people around you is important and one of the traits that helps identify a good engineering leader from a not-so-good one. Appreciating and acknowledging the work of people around you affirms the message that you recognize the good work and are vested in individuals' growth. We will learn more about reward and recognition in chapter 6, so stay tuned.

Now that we looked at some ways to manage remote/hybrid employees, let's understand that as EMs, we have to live with the decisions we make and adapt to them.

### **3.2.6 Accept responsibility for your decisions**

Take ownership of your decisions, be transparent, and explain your reasoning. Commit to and implement your decisions, and learn to navigate

scenarios where the consequences are not as intended. Imagine this scenario: As an EM, you have a solid roadmap for your team in the current quarter. However, there is a sudden change in priorities at the executive level, and your team is asked to support a project that cannot be deferred. Although this project was not initially planned, its impact requires you to assign one engineer full-time to it for the next three months. To minimize disruption to the team roadmap, you identify a senior engineer who is currently working on a task related to operational excellence, which is not a high-priority item for the quarter. While this engineer is relatively new to the company and has limited experience with the codebase of the service being updated, you communicate the opportunity and urgency of the project to them, ensuring their comfort in taking it on. You view this as a chance for the engineer to work on a high-visibility project and gain in-depth knowledge of the codebase within the project's timeline. You proactively seek feedback from cross-functional partners to address any difficulties early on. Despite your calculated risk, negative feedback emerges regarding the engineer's reliance on guidance and lack of ownership in completing tasks. Additionally, the engineer was defensive when called out for missing a technical detail in a discussion. In summary, you faced a situation where an unplanned project required reallocating a senior engineer. Despite proactive measures, the engineer received negative feedback for seeking significant guidance and lacking ownership, as well as displaying defensiveness during a technical discussion.

As the EM, you approach this situation as a coaching opportunity. During one-on-one sessions, you share the gathered peer feedback with the engineer, creating a safe space for candid discussion. The engineer reveals their struggle with the service codebase, which led to missing the milestone one deadline for the project. To address this, you provide the engineer with tips such as reviewing training materials and runbooks, watching recorded sessions from subject matter experts, encouraging thorough consideration of questions before responding defensively, and emphasizing ownership and completion of tasks. To course correct, you assign another senior engineer, a subject matter expert on the service, to assist with the implementation and help bring the struggling engineer up to speed. Although missing milestone one had an overall impact, your team recovered by temporarily allocating

another engineer to the project for two weeks. As a follow-up, you collaborate with the struggling engineer to create a gap analysis document, identifying areas of improvement and pairing them with a mentor to address these gaps. After two months, you gather another round of peer feedback to evaluate progress, which shows significantly more positive feedback. In terms of learning from this experience, you delve into the onboarding process for this engineer and ensure that technical documentation for all services your team owns is comprehensive and up to date. You also review the onboarding plan to identify potential opportunities, such as providing access to recorded technical talks that could have aided the engineer in this scenario. In sum, by taking ownership of your decisions and learning from your mistakes, you can build trust with your team and set yourself and your team up for success in the future.

### 3.2.7 Evaluation metrics of the team

Evaluating metrics to measure the success of your teams is important as an EM to identify any gaps that exist so that they can be addressed in a timely manner and, at the same time, help celebrate all the good work and victory moments. Some ways you can evaluate the success of the teams you manage are:

- **Establish a clear baseline** - Establish clear metrics, charter, vision, and boundaries for the team so that the success can be measured in a data-driven manner. This can also include understanding the delivery quality and meeting the milestones. Some concrete data-driven metrics that can be used here are - meeting project timelines and service level agreements such as threshold promised, hitting the objective and key results (OKR) established at the start of the project undertaken by the team, project delivery success rate for the team and amount of technical debt introduced as part of the project. If there are blockers, people can resolve them and move forward quickly.
- **Employee satisfaction** - Employee morale in the team through surveys and feedback collected throughout the year. This also signals how empowered team members feel in terms of trust, ownership, and

accountability in the team. This helps understand the overall success and health of the team.

- **360 feedback** - Throughout the book, we will talk about the importance of constructive feedback. 360 feedback is the feedback collected from your peers, people below your job level, above your job level, and any other cross-functional partners you work with. This helps gauge the sentiments of how people feel and also helps share growth opportunities with each other. This also signals how effective the communication is within the team.
- **Team development rate** - This is also another metric to define the success of your team. This signals that the team members are focusing on career opportunities, upskilling themselves, and focusing on personal well-being, and the by-product will be that deserving candidates are moving to their next level(getting promoted). This also involves knowledge sharing within the team to avoid single points of failure.
- **Other non-engineering metrics** - Some success metrics which are not directly engineering related are the value proposition to the customer, what is the adoption rate of the product by the end customer, usage patterns, and so on. These metrics help evaluate the success and failure of the team and help with learnings for the future.

Your success is basically your team's success. As EM, ensure to support your team for them to be successful.

### 3.2.8 Avoid burnout

Burnout is a major problem, especially when it comes to working virtually with no concrete working boundaries. Burnout occurs when a member of the team lacks the motivation to get the work done due to a few or more reasons that impact an individual's productivity- the person is overburdened with work, does not like the work assigned, work is mundane in nature, less innovative work, lack of support from the team, personal problems or lack of self-care. It is important to address burnout issues and avoid it for yourself and your team, ensuring a healthy conducive work environment. Some common ways to avoid burnout in the team are:

- **Load balancing work** - Ensure the work is load balanced well in the team, and everyone feels valued. Don't give too much work to the rockstar in the team only to burn them out and keep your low-performing team members idle to cause further deterioration in their work quality. We need to balance the workload between all the team members in a team. Keep an eye on capacity planning as part of agile ceremonies to ensure a fair balance of work between team members.
- **Open communication** - Promote a culture of open communication so people can speak up about their concerns and not sit on the sidelines. This way, you can proactively address issues or challenges team members face by involving them in the solution-finding process.
- **Work-life balance** - Work-life balance can be a real game changer for someone who feels burned out at the workplace. Work-life balance simply means a balance between your professional work and personal life. This means as an individual, you take out the proper time to focus on your work during work hours and similarly can focus on your family, friends, and personal self during off-work hours. This refreshes you and helps maintain a healthy lifestyle free from stress. Having clear work boundaries, not reaching out to individuals who are on vacation, and giving people space during sick or bereavement times are some of the ways you can give time and space to others. Also, team activities can help bring a pleasant break from strenuous work and rejuvenate the energy your team needs.
- **Stay away from cognitive bias and celebrate success** - We have emphasized creating an environment that values diverse perspectives, promotes critical thinking, and minimizes cognitive biases. Similarly, rewards and celebrating success together help feel valued and avoid burnout, where people see the outcome of the hard work they put in. Ensure to give shout-outs to team members and promote a culture of recognition in the workplace. We will learn about it in more detail in the rewards and recognition chapter.

Creating a culture of a healthy work environment is paramount for the growth and success of the team. A positive work environment fosters productivity, innovation, and excellence. I'd like to share my experience with the team I'm most proud of. Initially, the team I was assigned to

manage was in complete chaos. They were responsible for critical legacy services that frequently caused significant customer issues. However, the team members rallied together, supporting one another during root cause analysis and striving for a smoother on-call experience. They actively assisted each other on Slack and organized knowledge-sharing sessions like lunch and learn. I witnessed a remarkable display of teamwork when one team member fell ill with a tight project deadline approaching. Two engineers shared the workload to ensure we met the deadline. I also noticed that the team embraced failures as opportunities for learning rather than assigning blame, which is crucial for personal and professional growth. During conflicts, I offered support from the sidelines, encouraging open technical discussions to reach a consensus, and we celebrated successes together. As an EM, I prioritized keeping the team motivated, fostering a shared vision, celebrating achievements, identifying career development prospects, and providing coaching when necessary.

My most significant takeaway from working with this team was the ability to bring out the best in each member and collaborate towards a common goal.

### **3.3 Managing yourself**

In the preceding sections, we discussed the importance of managing team members and the team itself. However, it is equally crucial to learn how to manage oneself in order to lead and manage others effectively. This aspect of self-management is sometimes overlooked by EMs, but it plays a vital role in personal and professional development. As an EM, you are responsible for setting the tone for the team, as team members look up to you for guidance and mentorship. The team culture you establish significantly impacts each individual within the team.

By exemplifying traits such as being organized, motivated, valuing innovation, and leading by example, you can instill trust and confidence in your team members, allowing them to bring their authentic selves to the table. Simultaneously, managing yourself effectively enhances your productivity, enabling you to focus on more strategic goals for the team.

These practices not only contribute to your own growth and development but also build credibility with your team, peers, and leadership.

### **3.3.1 Identify opportunities**

Opportunities don't always come knocking. To achieve your goals and motivations as an EM, it's important to have your own goals and a career roadmap while tracking your progress. Self-awareness is a critical tool that can help you in this journey by understanding your strengths and weaknesses and finding opportunities to develop your skills.

During EM interviews, a common question is what your reports say about your strengths and weaknesses. This question aims to gauge your level of self-awareness. EMs often highlight strengths such as organization and time management skills, as well as strategic thinking abilities. However, many struggle to identify growth opportunities, which are just as important. Recognizing areas for improvement allows you to focus on honing your managerial skills.

To identify growth opportunities, you can:

- Conduct one-on-one meetings with team members, peers, and leaders to seek feedback.
- Send anonymous surveys to the team, encouraging candid feedback.
- Consider the team's morale score, reflecting their confidence in you as their EM and the work culture.

Personally, I discovered a growth opportunity when implementing core working hours for the team. These were specific hours when everyone, including myself, would be available in the office and schedule meetings. While the intention was to streamline communication, I realized that it limited flexibility for some team members who may have been unavailable during those hours. This feedback came from my engineers during one-on-one meetings, and I viewed it as a learning opportunity to address and improve upon.

### **3.3.2 Success as a manager**

Success in managerial roles differs significantly from success as an individual contributor. As an EM, success revolves around prioritizing the needs of others, whether it's the individuals on your team or the projects you oversee. In terms of people, your success as a manager is often measured by the growth and promotion of your team members. I can share a personal experience from my first managerial position, where I had the opportunity to guide a college intern. As her manager, I played a crucial role in recognizing her potential, understanding her strengths and weaknesses, and providing guidance for her career development. Witnessing her journey from an intern to a software engineer and eventually to a senior software engineer filled me with immense joy and a sense of accomplishment.

From a project perspective, success as an EM involves adapting to changing requirements, proactively identifying dependencies and risks during development, avoiding single points of failure within the team, and cultivating a positive work environment. It's important to acknowledge that success metrics now revolve more around the achievements of others rather than solely focusing on personal accomplishments or code-related achievements.

### **3.3.3 Everyone makes mistakes**

It's important to acknowledge that as human beings, we are all prone to making mistakes, even as EMs. Rather than being the smartest person in the room, we must surround ourselves with intelligent individuals who can help us make sound decisions. Let me share a personal example of a significant mistake I made as a people manager and how I handled it.

One of my biggest mistakes was becoming overly friendly with some of the team members I supervised. This may have been influenced by my previous role as a software engineer on the same team before transitioning to an EM. However, I failed to establish clear boundaries from the start. As a result, two team members did not view me as their direct manager and did not take

my constructive feedback seriously. It was important for me to prioritize the growth and learning of my engineers, but being excessively friendly turned out to be a major misstep. For instance, when a project deadline approached, I asked a close engineer friend to step up and assist. Unfortunately, the engineer didn't take my request seriously, leading to a delay in the project's QA process. I took responsibility for the situation, scheduled a one-on-one meeting with the engineer, and clarified expectations. I emphasized that friendship should not interfere with day-to-day responsibilities. This experience taught me the importance of setting clear expectations upfront to avoid future consequences.

Similarly, on the project front, serious mistakes can occur by assuming something will work without conducting a proof of concept. This can result in inaccurate estimations of effort and timelines. Another mistake is failing to allocate the right balance of resources to a project. For complex projects with significant architectural design, it's crucial to have senior engineers overseeing the work of juniors. EMs are not exempt from making mistakes either. As an EM, you are responsible for the entire team, not just yourself. The key lies in assembling the right individuals and making informed decisions based on available data.

### 3.3.4 Manage emotions and well-being

As EMs, we bear the responsibility of caring for both our team and ourselves. However, this can often lead to stress and taking on too many commitments. Effectively managing stress becomes a critical aspect of nurturing our emotions and well-being. It is essential to prioritize self-care to support and care for those around us effectively. Here are a few points to consider for managing your well-being:

- **Make your well-being a priority:** Taking care of yourself is crucial to lead and support your team effectively. Take the time off you need, and encourage your team members to do the same, ensuring they have the necessary rest to bring their best to work.
- **Establish boundaries:** Sending work-related emails late at night might inadvertently set an example that working such hours is

expected or acceptable. While it may be possible for you to find time at night, when others in your household have already slept, it is courteous to indicate in the email subject line that the message should be read the following day, such as using "<Read next day>" or a similar indication.

- **Dedicate time to activities you enjoy:** Set aside time for hobbies or activities that bring you joy and help you relax. During one-on-one meetings with your team members, ask them about their weekend activities as an icebreaker, and share your own methods for de-stressing. This can help prevent burnout and foster a positive work-life balance.
- **Manage your emotions:** As an EM, you will inevitably encounter challenging situations, such as managing underperformance or conflicts within the team. While empathy is important, it is crucial not to let emotions cloud your judgment. Make a conscious effort to base your decisions on objective data and focus on effectively managing your emotions. Allocating time to prioritize your well-being will aid in maintaining emotional balance.

By prioritizing your own well-being, setting appropriate boundaries, engaging in activities you enjoy, and effectively managing your emotions, you can ensure that you are taking care of yourself while fulfilling your responsibilities as an EM.

The role of an EM entails overseeing the entire 360-degree world surrounding you, which includes managing individual team members, multiple teams, and, of course, yourself. As an EM, you navigate the complex dynamics of leading individuals toward a common goal, fostering collaboration among teams, and ensuring your own growth and effectiveness as a leader. It is a multifaceted role that requires a holistic approach to effectively manage the various components that contribute to the success of the team and the organization as a whole.

### **What do other leaders have to say about it:**

“The biggest challenge when managing people is that every person is unique in terms of their motivations and learnings. As a leader, you are always on a mission to help them to become the best version of themselves so a good manager has to spend time and effort on effective methods of recognizing and motivating each of the team members and also find the most effective ways of providing feedback. My one-on-one effort is to ask the right questions to empower them to find solutions to grow personally and professionally. These also allow us to know team members outside the work enabling us to build trusted relationships.”

**~ Sumit Kumar, System Engineering manager at Cisco**

“One-on-ones with your team should never be about their work but about them and your understanding of them. Always go in with the right set of questions to ask them. I have seen mostly engineers come un-prepared in their 1 on 1s and you need to guide them towards the right direction to identify what is that works or doesn’t work for them”

**~ Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

“The one-on-one is mostly for the employee, rather than for the manager. My goal is to hear how they are doing and what’s on their mind (both what they want to share and maybe to tease out other important things that they didn’t mention) and to hear what they need. If there are course corrections or advice I need to give, then I will deliver that information.”

**~ Jean Bredeche, Head of Engineering at Patch, (ex) Robinhood, Quantopian, Hubspot**

“Listening with an open mind results in one of the most productive 1 on 1s. EMS should provide an open platform to listen to newer ideas, concerns, and continuous ways to improve the team's productivity.”

**~ Devika Ahuja, Technology Leader | Strategist**

“For the biggest challenge to manage people, it is very easy to conflate team productivity with individual performance. Don’t. A team doesn’t have “performance”; a team has productivity. Team productivity is not a sum of the individual performance of members of that team. Individual performance is how well an individual operates in the context of the team and company dynamics. One of your jobs as a manager is to own team dynamics and externalities for your team and to work toward them being more favorable.

The best way I have found to break this connection is not to assign work to individuals and instead have individuals opt into taking the highest priority work from the backlog. Having a team organized around “how do we deliver this important work” takes the bullseye off individuals and shifts it to the team. Individual performance becomes a management retrospective, where I (their manager) can inspect the work someone did after the fact and how they operated within that team-owned plan.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“As an EM, resolving conflicts and getting alignment across a broad spectrum of people becomes part of your job description. No two people have the same perspective, and finding common ground is crucial to progressing in this plethora of views. The most common conflicts I see are software ownership when working on projects across multiple software teams. I recommend not keeping ownership as a primary factor for solutions. It plays a part in long-term maintenance but should not make technical designs more complex/convoluted. I have found that setting design tenets helps teams focus on what’s important. For some, performance might be higher than availability or security. For others, cost or maintainability might be a bigger concern. Look for what matters to your team and use that to inform your decisions.”

**~ Adish Agarwal, Director, Software Development at Audible, Inc.**

“Managing a team in itself is a challenge, and it becomes more complex when you have virtual teams and furthermore complicated when the teams are in different timezones. The biggest challenge in my career was managing people in different time zones. First and foremost, identifying all your challenges early is the key. If there are many, prioritize them and tackle the most critical ones first. Once your final list is ready, you can spend the rest of the time developing a strategy. The best part is you don’t have to find the resolution alone. In my situation, we had to figure out how to collaborate, review the code, and conduct meetings among the many. To combat it, I’ve invited the whole team to brainstorming sessions and engaged the team to pitch in their ideas, share their opinions, hear them out & weigh in on each option. Created virtual polls to gather everyone’s inputs, and eventually a consensus was reached among the team with little guidance & facilitation.”

~ **Vindhya Avvari, Cloud Integration manager, University of Southern California**

“Regarding managing people, since they come from various backgrounds, the biggest challenge is to understand their roots so you can assess the situations and ask with proper context. One way I always try to approach this is first getting a level of connection with the people and trying to put myself in their shoes. I have tried various approaches for 1:1s. The recent one is to have the teammate set the agenda for the meeting and not you as a manager driving the meeting. Of Course, you can add your topics, but one-on-one is for and about them.”

~ **Rajakumar Sambasivam, Delivery Manager at Microsoft**

“For managing people, the biggest challenge is to assign the right person to the right job then most of the other challenges will be taken care of by the individual employee. Being a mentor and coach to the employee will speed up productivity and reduce conflicts. For successful one-on-one’s- keep track of action items, provide status to the employee on the action item, even if you did not make progress, let the employee talk, ask a more open-ended question, and balance it with all the aspects like job/career/team

health/ job satisfaction / new ideas. Basically, it is not just career coaching. And don't get into teaching/preaching mode.”

**~ Saravanan Subramanian, Senior manager at Amazon AWS| (ex) Capital One, Citi, Cisco**

“When moving from an IC which works mostly with machines to EM, one striking difference is, unlike systems which are mostly molded in similar dies, people are way different from each other. Each person's motivation, drive, fear, etc., are very different, particularly in a well-balanced team with diverse backgrounds. Navigating through this may be the biggest challenge.”

**~ Sarin Panambayil, Principal Engineer at Yahoo**

### **3.4 Stop & Think:Practice questions**

1. As a manager, what is one of the most difficult management situations you have been in? Why was it difficult?
2. What is your expectation from your manager when it comes to career conversations? Do you practice what you expect?
3. What is the biggest challenge you've faced when it comes to working in a remote environment?
4. What are your success criteria? How do you measure it?
5. What is the biggest mistake you have made at the workplace? If you could change one thing about it, what would it be?

### **3.5 Summary**

- Management requires a comprehensive understanding of what is being managed. Effective people management involves understanding their motivations, providing support, and fostering growth.
- Managing a team involves various tasks such as goal setting, addressing morale issues, setting expectations, building trust, managing remote teams, and evaluating success.

- Effective leadership also requires self-management, self-awareness, self-regulation, and personal development. Prioritizing self-care, efficient time management, and skill enhancement helps managers maintain their well-being and perform at their best.
- Managing people - Mastering the art of managing people is essential for any manager, including engineering managers. It majorly involves:
  - Career conversations - A career conversation helps with information sharing, relationship building, discussing career growth, providing feedback, and more.
  - Supporting junior versus senior engineers - There is no one size fits all approach for managing a junior engineer and a senior engineer. A junior engineer will need more support than a senior engineer in the team.
  - Conflict between team members - Team conflict is inevitable and sometimes can also bring up healthy competition among team members and bring the best of their work to the table.
  - Challenging team members - There will be times when you manage people who can be challenging to work with. Such situations are a learning experience and teach us skills that maybe no book or learning material can directly teach us.
  - Lead by example- Leading by example is a way of setting expectations with the team members and not just talking about it, but showing how it's done, in better terms, 'walk the talk.'
- Managing teams- Managing teams is one of the core responsibilities and expectations as an EM. The team members make the team, but since each individual is different, once they come together as a team, it is important for you as the captain of the ship to ensure that it sails well.
  - Defining team goals, vision, and strategy- This helps bring different team members together on this shared mission. It helps us understand what we are doing, why, and how behind it.
  - Handling morale issues- Maintaining high morale and positive sentiments among team members is vital for team success and long-term retention of individuals. Low morale can have a detrimental effect on workplace productivity and performance.

- Trust and transparency - It is important to build trust and a safety net where people feel they can bring their original selves to the workplace. A team is one where employees can trust each other and bring their full potential out.
  - Convince other teams to account for dependencies in the team roadmap - Sometimes, it becomes necessary to persuade other teams to adjust their roadmap to accommodate interdependencies between teams. By presenting a compelling case, you can help the other party understand the mutual benefits and significance of the work for both teams involved.
  - Managing remote teams - Managing remote teams can get tricky as people work from different geographical locations and time zones. As EM, it is important to keep your team together and find a sweet balance between in-person/ virtual setups.
  - Accept responsibility for the decisions made - Managers must take ownership of their decisions and accept the consequences that follow. Being accountable and responsible means being transparent about decisions and willing to explain their reasoning. Once a decision is made, it is important to commit to it and ensure successful completion.
  - Evaluation metrics of the team - Evaluating the success of your teams is important as an EM to identify any gaps that exist so that they can be addressed in a timely manner and, at the same time, help celebrate all the good work and victory moments.
  - Avoid burnout - Burnout happens when team members lose motivation to complete their work for various reasons. It is crucial to address and prevent burnout within yourself and your team by fostering a healthy and supportive work environment.
- Managing yourself - This aspect of self-management is sometimes overlooked by engineering managers, but it plays a vital role in personal and professional development.
    - Identify opportunities - Opportunities will not always knock on the door. It is important to have your own goals and a career roadmap to get to your goals and motivations and track progress as EM. The critical tool that can help here is self-awareness,

being cognizant of your strengths and weaknesses as a manager, and finding growth opportunities to hone your skills.

- Success as a manager - The definition of success significantly differs between being a manager and an individual contributor. Success as a manager is more centered on prioritizing the needs of others, whether it be the individuals within your team or the projects at hand.
- Everyone makes mistakes - It's important to acknowledge that everyone, including engineering managers, is capable of making mistakes. The role of an engineering manager entails taking responsibility for the entire team, not just oneself. The crucial aspect is to assemble a competent team and make informed decisions based on available data.
- Manage emotions and well-being - As engineering managers, we must balance the care for our team and ourselves. By prioritizing self-care, setting boundaries, engaging in enjoyable activities, and effectively managing emotions, we can fulfill our responsibilities as engineering managers while taking care of ourselves.
- As an EM, you navigate the complex dynamics of leading individuals towards a common goal, fostering team collaboration, and ensuring your own growth and effectiveness as a leader.

# 4 Managing performance

“Performance management involves embracing employees’ strengths and being open to innovative ideas – even ones that change the status quo.”

~ **Steve Jobs, Co-founder, Apple Inc.**

## This chapter covers

- The importance of managing performance
- Best practices around performance management and expectations
- Dive into performance reviews
- How to manage high performance
- How to handle low/underperformance

Suppose you are a relatively new engineering manager (EM) at a company, working on some critical product features to be launched by next quarter. David on your team is a strong engineer who ensures to push deliverables to production on time. This half of the year, his productivity was no different, where he completed all his deliverables and, on top of it, picked some stretch work here and there. On the other hand, Jason is another strong engineer on your team who identified specific gaps in the existing technical delivery process and proactively spent time streamlining them as part of his stretch goal work. Because of his time investment in this work, he missed some key deliverables that were important for the next round of product feature iterations. The contributions made by Jason may have a larger impact as his work cuts across different cross-functional teams and solves a key bottleneck issue in project delivery. Still, it led to a significant impact on key project deliverables.

Given the two situations above, who do you think performed at or better than their initial expectations of the project deliverables? Managing performance is an art that helps the manager understand their team members' motivations and aspirations and brainstorm with them a roadmap to get to the goal state. In the process, managers provide coaching and

frequent constructive feedback to help the individuals grow. This also serves as a point of reflection regarding what has been accomplished and an evaluation of the work done. As part of goal setting, together you set up SMART (specific, measurable, achievable, relevant, and time-bound) goals, and continuous feedback helps to provide early feedback to avoid late surprises. The process helps build trust and boosts the morale of individuals, helping coach them to treat this as a learning opportunity instead of merely an evaluation exercise.

Returning to our initial example, let's examine the two cases. David completed all his tasks, including some additional objectives. This undoubtedly meets his expectations. Jason's situation is more complicated. Although his work this time has the potential for significant long-term impact, he needed help to balance his time between the main project tasks and the additional objective. Moreover, he failed to raise any concerns early on, which could have allowed the team to provide support or adjust priorities if necessary. This missed opportunity to allocate work effectively between the main project and the additional objective ultimately affects the team's project timeline. Therefore, as the EM, it is your responsibility to intervene, coach, and mentor Jason regarding his role expectations and performance management. This entails recognizing and rewarding engineers who exceed expectations while coaching those who may have deviated or failed to meet the expected standards. With proper guidance, Jason could have focused on the main project tasks and resolved the bottleneck issue more quickly. Additionally, he could have assisted the team by promptly alerting them to any delays in the main project, allowing for proactive planning. Now that we understand what managing performance is like at a high level, let's look at why it is essential.

## 4.1 Importance of managing performance

When used correctly, a performance management system helps bring the full potential out of employees and helps them grow. It provides a structured way to keep track of employee performance by providing clear goals and objectives. On the contrary, if you, as EM, do not collaborate with

your engineers to plan ahead for their careers, help them grow, and groom them for success, you can end up doing more damage to the team member and the team.

Now there are several benefits one can reap from using an effective performance management system, such as

- employees showcasing sustained productivity and more engagement with the work
- helping employees with career planning and professional development
- sharing constructive feedback with each other proactively to prevent surprises, in the end,
- ensuring alignment of each team member on the goals and expectations and how their contributions fit in the bigger picture of company strategic goals
- higher chances of employees staying in the company and feeling valued
- helping our leadership and recruitment plan for future hiring initiatives

Overall, managing performance is critical for the success of team members, teams, and the business. It helps promote continuous improvement, employee engagement and retention, and a culture of accountability, all essential for achieving goals and maintaining a competitive edge in this fast-paced world. Let's examine why an effective performance management system helps the team members, teams, and the organization.

#### **4.1.1 Sustained productivity and engagement**

Performance management tracks expectations and employee performance, fostering trust and demonstrating the company's investment in its growth. This creates a sense of safety and happiness, increasing productivity and engagement. It also enhances team trust as individuals progress and the company promotes from within. Performance management serves as a reminder that staying on track brings rewards. Additionally, it enables leadership to reward and recognize high-performing individuals while addressing underperformance early on.

From our example above, David showed sustained productivity and performance, finishing his deliverables and the stretch goals aligned with the manager. On the other hand, Jason had a tough time balancing the sustained productivity and went over expectations on the stretch goal and thinking about long-term strategic goals but missed critical project deliverables that were like a must-have for the team.

#### **4.1.2 Career planning**

A company-wide performance management system (which involves the performance review process and the tools used to ease performance planning) helps with career planning and identifying individual growth opportunities. With continuous one-on-ones and frequent performance discussions, the managers get an opportunity to understand the strengths and motivations of the employees and open a forum for candid conversations. This helps to carve out short-term, mid-term, and long-term career plans for the employees and sends a positive affirmation to the individuals. This, in turn, helps to measure performance and introduce any improvements as needed in a timely manner. From the employee perspective, they get a sense of work satisfaction to see the company is vested in their career growth.

David and Jason, both members of our team, underwent the process of career planning. While David adhered to the plan and progressed toward the milestones he and his manager had established, Jason initially had well-defined objectives. Still, he got sidetracked by his interest and skill in project management. Although he was able to resolve a bottleneck in project delivery, he should have raised concerns promptly about missed project deliverables, which caused him to fall short of the career goals he had set with his manager. As an EM for Jason, if you fail to bring his focus back on key matters, you are also doing him a disservice since you could be rewarding the wrong behavior.

#### **4.1.3 Proactive feedback rather than reactive**

Continuous performance management helps engineering leaders to provide proactive feedback rather than reactive to team members. Of course, there is a reactive component to it where feedback is provided in response to an incident or a pattern noticed, but the idea is to not wait for long annual cycles to share feedback. Sharing feedback as part of regular 1 on 1s and frequent career discussions help steer the employee in the right direction. This helps when an employee has diverged or is underperforming and needs some course correction actions or adjustments early in the process. Also, this opens a two-way dialogue between the individual and their managers to ensure each side is well heard and feels part of the process.

If, in the aforementioned scenario, Jason had alerted his manager in a timely manner about the missed project deadlines, or if the manager had proactively noticed the issue, the negative consequences could have been avoided altogether. In such a case, the manager could have provided proactive feedback to help Jason get back on track and ensure that the crucial project deliverables were met, thereby preventing harm to the team.

#### **4.1.4 Alignment of strategic goals**

Performance management helps create unified goals for the team members and the teams, keeping them aligned with the business goals. Regular performance check-ins help ensure an individual has set clear goals and ways to measure success and are meeting their goals which in turn contribute to the team goals and overall success of company-wide goals. Also, if there are any risks or blockers, a continuous performance review ensures to keep a tap on that and identify risks earlier in the process. This ensures all employees are pulled in the same direction as one team, aligned with strategic goals.

In our example, the company or the team needed to follow regular performance check-ins as Jason pursued his stretch goal and missed the project deliverable without it being seen as a concern by the team until it was too late in the game. Had frequent performance check-ins been in place, a goal check on progress made could have helped Jason earlier in the process.

#### **4.1.5 Better retention chances**

Who does not like a job that helps with career growth & development (learning opportunities to work with the latest technology stacks, work on different problem domains, and more) and keeps us away from surprises in the end regarding performance? Most of us would like to work for an employer vested in the employees' career development and helps them with overall career aspirations. This means such companies will have a better chance of hiring and retaining new talent to avoid widespread attrition. This also helps with overall savings in terms of cost and time spent on hiring new people, onboarding them, and bringing them up to speed.

The incident involving David and Jason highlights that David is likelier to remain with the organization and feel appreciated. In contrast, Jason may resent the unexpected outcome and lack the drive to continue in his position. A robust performance management system can play a critical role in determining the outcome of such situations.

#### **4.1.6 Planning for future hiring and training programs**

One key benefit of a robust performance management system is to help the leadership and human resources department plan for what's next in terms of hiring goals and look for patterns in the performance calibration. This also helps the company in terms of financial planning (i.e., a constant view into the fiscal impact of promotions). Also, it compels the human resources team to do the calibration through market reviews and surveys or other means to ensure everyone is compensated near industry expectations.

In terms of training, many of the employees had opportunities to grow in the system design space; this means as a company, leadership and HR can plan for some courses and resources to help hone system design skills for the individuals. Similarly, as EM, you can use the opportunity in the pre-planning phase for projects and one-on-one with employees to understand what training resources might be useful for them. You can then point them to technical resources or tech conferences that help them hone their engineering skills. Further, planning the hiring and training initiatives

provide an opportunity to plan for any specific diversity, equity, and inclusion initiatives the company should plan for to build a safe and conducive work environment. To conclude, there are many benefits of having a robust performance management system to help at all levels, starting from individuals to teams to the overall organization. It helps tap into the real-time progress of individuals and intervene in cases where necessary to avoid repercussions later in the process. In the next section, we will discuss some best practices for managing performance.

## **4.2 Best practices for managing performance**

Performance management is a complex process that requires careful consideration of several key factors. It is not a one-time task but an ongoing process that supports the development and improvement of every employee in the organization. As performance management can significantly affect an individual's perception of their role and performance in the company, it is crucial to adhere to industry-standard best practices that facilitate the growth of employees. Let's explore some of the essential best practices to consider when managing performance, such as being transparent with the employees, bringing fair to all employees and keeping biases aside, keeping away from the halo/horns effect, appreciating the excellent work around us, laying emphasis on effective communication, provide the right support to our team members so they can bring their best performance to the table and ensure to treat performance management as this continuous iterative process to improve and learn.

### **4.2.1 Transparency**

Promoting a transparent environment is a key objective for organizations seeking growth. Instilling a culture of psychological safety is the first step toward promoting transparency. Psychological safety in the workplace refers to an environment where employees feel safe, comfortable, and supported in taking interpersonal risks, such as speaking up, sharing ideas, asking questions, and challenging the status quo, without fear of negative

consequences or punishment. Some of the key characteristics of psychological safety are:

- Trust
- Open Communication
- Non-judgemental environment
- Inclusivity
- Supportive leadership
- Interpersonal respect.

At the same time, you may be required to withhold some information at certain times as an EM. Therefore, transparency does not imply the need to share every detail with all employees. It's about finding the right balance between sharing and withholding information. For instance, managers should share feedback from various sources during the performance calibration process without revealing the specific providers. This approach maintains clear feedback channels for individual development while avoiding personal animosity. As an engineering leader, being transparent about the team's mission, goals, and expectations is essential. Setting clear SMART goals for each team member ensures that feedback, whether appreciative or constructive, does not come as a surprise. Utilize effective team one-on-one meetings to provide transparent feedback and coach individuals in the right direction. This approach fosters a culture of thinking big and aligning with organizational goals and performance, reflecting the value of holistic thinking.

#### **4.2.2 Ensure Fairness and Avoid Personal Bias/Favoritism**

The workplace environment significantly influences individual productivity. Fairness is essential in ensuring consistent behavior and equitable distribution of work and recognition while preventing recency bias, where recent events are given undue importance in decision-making. Most companies utilize a leveling guide to define performance standards and calibrate employees against objective criteria instead of comparing them to each other. As an EM, you can advocate for creating a leveling guide if your company doesn't have one. This approach fosters a fair assessment of

individuals based on company expectations, reducing subjective bias. Fairness promotes satisfaction and a sense of safety and drives engagement. Initiatives such as diversity, equity, and inclusion programs reinforce the importance of treating each individual professionally, fairly, and with respect. Achieving this requires raising awareness, providing respectful training to cultivate mutual respect, and ensuring discussions are data-driven to avoid favoritism and unconscious bias. Moreover, establishing a mechanism for employees to challenge perceived unfair treatment and pursue a fair appeal process in collaboration with the HR department further enhances workplace fairness.

Revisiting our example of David and Jason in our team, although Jason took up something highly impacting and cut across multiple departments, a balanced outlook is needed to understand the misses on critical project deliverables. Now both team players should be compared against the company leveling guide to ensure their work is consistent with the expectations in the job role.

With that being said, there are instances where given the employee situation, there will be some exceptions. For example, if one of the employees needs time to take care of an elderly at home or you have an expecting mother on the team, it is important to be empathetic and give them the support they need to take care of family/personal things. Also, in such situations, bring your team together to help each other and use the divide-and-conquer strategy to fill in for the missing person.

#### **4.2.3 Avoiding Halo and Horns Effect**

The Halo/Horns effect is a common term used to refer to biases around how the employees are treated based on how they work in the team and demonstrate some personal traits.

##### **The Halo Effect**

In layman's terms, the halo effect means when someone is idolized and appreciated based on a subset of good characteristics noticed. The human

mind is naturally inclined to focus on distinctive individual attributes. Researchers have found that if such an attribute is positive, it will affect ratings of the other attributes.

### **The Horns Effect**

While the halo effect makes your colleagues look like angels, the horns effect has the opposite effect. Horns effect is the tendency of one negative attribute to negatively influence other aspects of an employee's work.

This means that using a subset of habits and characteristics, the whole character is either considered positively or negatively, overshadowing what other things the employee has to offer at the table. A simple example is if someone uses academic background(degree/no degree/code champ) to judge whether someone will make a good or bad hire. Another example of a halo situation is someone who is a key contributor with unique knowledge and is therefore over-recognized simply because they can't afford to lose them. And since one reason leads to another, we either keep pushing to like them more or hate them more depending on the halo or the horns case. This builds up conscious and unconscious biases in the minds that harm employees' development, and as engineering leaders, we should stay miles away from this behavior. Such behavior can be avoided by keeping conversations and reviews data driven backed by facts and peer feedback to ensure a holistic view is considered.

### **4.2.4 Recognition**

Recognition instills confidence, keeps employees motivated, and makes them feel valued. Valued employees have a higher chance of retention than employees who do not feel valued. Recognition is important to recognize people who consistently go above and beyond and deliver that extra mile. Such behavior also sends positive affirmations and results in higher productivity. As you manage performance, think about behaviors where the employees used their strengths and went that extra mile to appreciate them for the work. At the same time, reward and recognition should be driven by

facts and kept professional, keeping biases aside. The chapter on rewards and recognition dives into the topic to share some ways to use this superpower to drive employee engagement.

#### **4.2.5 Effective communication**

Effective communication is another key tool to keep handy in your toolbox when it comes to managing performance and can lead to lost opportunities if not used correctly. The idea is for the information to flow both ways, the team member speaks openly and candidly, and you are a patient listener in understanding their perspectives and vice versa. This allows both sides to speak up and ensure they are heard, making them feel fully engaged. Also, since managing performance acts as a retrospective to understand what actions were taken in the past and what can be improved, it is important to keep conversations frank and effective. This helps communicate the expectations/goals for the employee, rewarding them for their work and sharing the feedback received from their peers. Since no one size fits all, you, as the engineering leader, must cater to the communication style so the person on the receiving end is receptive. For example, it can be a very different conversation from appreciating your high performer to discussing instances of underperformance and the consequences it will have for your low-performing employee.

#### **4.2.6 Training**

Not everyone will know everything on their own. As the manager and engineering leader, it's our responsibility to help our team members hone their skills. This means we should work with them to identify the gaps in their performance or skills and provide the training resources and support they need to succeed in the role. Let's say a senior employee in the team is struggling with delegation. You should be able to help identify the issue, understand the individual's perspective and help them with a roadmap to overcome the struggle. This may involve you guiding them through some LinkedIn learning training, sharing leadership training opportunities, seminars, and conferences, pairing them with an experienced engineer as a support mentor or buddy, and even sharing your own delegation struggles

and how you overcame them. Providing the desired support helps build trust with the individuals and helps set them up for success.

#### **4.2.7 Continuous performance management - ongoing process**

Performance management is an ongoing (agile) process that should not be treated as a ‘fire and forget’ activity. Instead, consistency is the key when planning, coaching, and helping the employees grow, using learnings from the past. This involves monitoring progress through frequent check-ins at a regular cadence and crafting a roadmap to execute the plan. This ensures the employees feel part of the process and are equally involved. Of course, since it's done continuously, it needs to be lightweight, with standardized processes followed across the company to ensure the process is fair.

Another advantage to this approach is that problems/gaps are addressed earlier in the process instead of waiting for bigger damage. There have been instances where continuous performance management has helped to turn around underperformers. For example, suppose an employee moves tangentially to the expected job role and is underperforming. In that case, continuous performance management helps to provide early feedback, provide the desired support and bring them back on track. Managing performance is a crucial task that engineering leaders are required to perform and help the team members grow. With the above best practices, one can ensure the system is inviting to all, fair and keeps all sides involved. In the next section, we will look at performance reviews and the steps involved in the process.

### **4.3 Performance reviews**

The performance review or the talent cycle is a process done at a regular cadence to reflect on what was achieved by each individual over time, understand what motivates them, how one performed against the company evaluation guide for the level of performance, and any specific trends on one’s talent. This is essentially an important input to the **compensation changes** that take place, depending on specific ratings of the individual employee. Although, I would call out that the regular cadence is something

your organization, you, and your team might have to identify. Performance reviews done every quarter(too often) might take too much toll on HR and managers as they go through the cycle of calibration, while if done once every two years(too seldom) can impact the company's overall performance. Hence, finding the right balance and seeing what works best for your organization and the team is important.

With performance reviews come compensation changes, as we touched on above. Let's say an employee who is rated as a top performer will get a higher pay bump than someone rated as a solid performer. On the face of it, it can look like a time-consuming and bulky chore, but it is one of the essential pieces for career conversations and developing your employees as an engineering leader. Usually, this exercise is overseen by the organization's human resources (HR) team to ensure the process is fair and employees' concerns could be addressed. This is also treated as a way to understand if there are gaps in terms of what employees need to be successful versus the resources provided by the organization. Let's look at key features to prepare for performance reviews.

### **4.3.1 Evaluation**

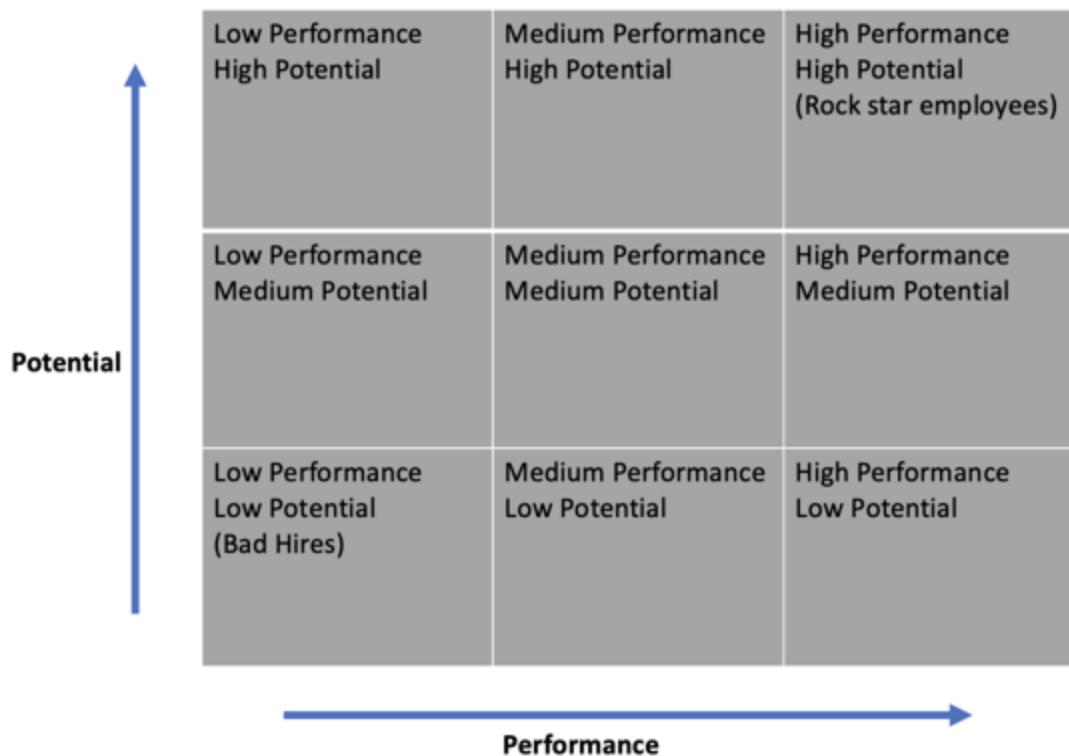
Typically, the first stage involves assessing an individual's performance against the company's expectations and evaluating their potential for the upcoming year. The most important part of this process requires collecting data, including self and peer feedback, revisiting established goals, looking at project deliverables, and much more. It is important to be impartial and considerate, examining the employee's motivations and aspirations. During this evaluation, the manager assigns a potential rating to the employee, which is finalized during the calibration phase. Some companies use a bell curve, while others use a rating scale, 1 to 5, to assess performance. To ensure an individual's personal and professional goals are aligned, and they are on a path for growth, any gaps must be identified and corrected in a timely manner. The aim is to identify goals and create a roadmap to help the employee achieve them. For instance, the employee may benefit from being involved in hiring initiatives, where the manager could provide the necessary tools and support. This might include mandatory hiring training,

shadowing interviews with a recruiting coordinator, and mentoring new hires or summer interns. These are just a few examples of how a manager can support an employee's professional development.

### Did you know?

A common evaluation/performance management technique used across industries is the 9-box-rank approach or the 9-box ranking. In this, the employees fall into one of the nine buckets considering their performance (in the present) and potential(for the future). This helps evaluate the talent consistently and helps identify high-performing rock stars and underperformers that need attention.

**Figure 4.1 The 9-box ranking approach that takes into consideration performance and potential**



The approach helps provide leadership a view of the current talent and helps with talent planning for the future. Keeping a data-driven approach when categorizing employees in one of the nine buckets ensures intentional and unintentional biases are kept out. One can help identify new opportunities for high performers and provide training and support to under-performers.

## Collecting feedback

Feedback includes both- self-feedback or evaluation and peer feedback. Self-feedback allows employees to reflect on their work and write down their thoughts and understanding of how they performed against the goals and expectations set for them. This helps identify their success moments and opportunities for growth where they could have done things differently. Also, this is the time for them to ask for support for the future to help them grow and hone their skills. This also helps refresh the memory of the EM, who might have a large team or manage multiple teams at a given time and might forget some key moments or contributions of the employee.

**Peer feedback** is a crucial part of the evaluation process. Team members collaborate with various individuals in cross-functional teams, such as UX developers, product managers, QA engineers, and technical program managers. This feedback, also known as 360-degree feedback, provides valuable insights into an individual's work and impact. It helps identify growth opportunities by reflecting on how others perceive their work. It is essential to approach this feedback positively at all levels, including as an engineering leader. Seeking feedback from engineers, peers, and leadership ensures a comprehensive perspective. To gather holistic feedback for team members, reach out to peers identified for feedback, allowing them sufficient time to provide both strengths and areas for improvement. This feedback informs the employee's performance review and contributes to data-driven evaluations. However, it is crucial to establish a culture and environment where peer feedback is not misused or used to undermine team members. Some resources for how good feedback should be are this [HBR article](https://hbr.org/2019/05/what-good-feedback-really-looks-like) (<https://hbr.org/2019/05/what-good-feedback-really-looks-like>) and

this [blog](https://blog.businessleadershiptoday.com/what-does-good-feedback-look-like/) (<https://blog.businessleadershiptoday.com/what-does-good-feedback-look-like/>). As an EM, it is your responsibility to foster a culture where peer feedback is a constructive tool for improvement and learning from past experiences.

## Adjusted expectations

Adjusted expectations conceptually mean to account for any unique circumstances the individual employee had to go through throughout the course of the year that could impact their performance and hence the evaluation. These can include a multitude of reasons, such as any role changes the employee went through, any external reasons beyond their control, or taking a leave of absence due to a medical emergency, family reasons, etc. This doesn't call for the evaluation process to change or lower the performance bar, but rather just to account for the time a person was available in the role so that the evaluation done is fair and just.

Let us take an example to understand this. Let's say someone in your team was pregnant and had a medical complication. This made the person go on leave of absence a few months before delivering the baby and then the maternity leave once the baby was born. Such a person might not have been available for half the year. So when doing the performance review, ensure we look into the superpowers exhibited during the time they were available and how they exhibit potential for the next year. Stay away from comparing this individual with someone who was available for the entire year and must have worked on more technical projects and initiatives in comparison to this individual who was available for half the time. Never compare employees, instead calibrate them against the career growth plan using the gap analysis document referred to in chapter 3, section 3.1.1. Also, if you see instances of slack, you, as their EM, could provide them options such as flexible hours or working at reduced capacity programs, keeping in kind the HR rules and regulations to help support the individual. Some companies offer employee assistance programs and mental health tools to help needy employees. Let us look at another example where, let's say, employee David went through a role change transition from an individual

contributor(software engineer) to being a manager sometime mid-year. Now for this individual, the performance review should be done keeping in mind the job role change amidst the performance cycle. We should use the latest expectations in the role of a manager as the baseline for the review cycle.

#### **4.3.2 Calibrate across teams/organizations**

After conducting individual evaluations, the next step is calibrating employees across teams and organizations, led by directors and VPs. During calibration sessions, managers present strengths, weaknesses, and growth opportunities, sharing potential ratings. Open discussions ensure alignment with the company leveling guide, promoting fairness and uncovering any favoritism or harassment. Human resources business partners participate to ensure consistency and fairness throughout the organization. These sessions offer valuable insights into overall organizational performance and facilitate discussions on attrition, retention, and promotion opportunities. Underperforming employees can be identified for improvement. Final ratings are determined based on these discussions. Successful calibration sessions require effective time management and thorough preparation from managers during the evaluation phase.

Additional resources on best practices for calibration meetings can be found in the [Lucid chart blog](https://www.lucidchart.com/blog/how-to-conduct-a-performance-calibration) (<https://www.lucidchart.com/blog/how-to-conduct-a-performance-calibration>) and [lead dev blog](https://leaddev.com/mentoring-coaching-feedback/managers-guide-performance-calibration) (<https://leaddev.com/mentoring-coaching-feedback/managers-guide-performance-calibration>).

#### **4.3.3 Preparation and conducting performance reviews**

Ok, so now the evaluation is done, you have identified the superpowers and opportunities for each of your employees and have ensured the ratings are well calibrated across the organization- so what's next? Yes, next is the actual step of preparing and conducting the review in a one-on-one setting with you as manager and the individual employee being reviewed. This phase will involve the manager setting up the calendar invite, ensuring there is buffer time in case the employee has additional questions, and going well prepared with the writeup that is data-driven and the facts speak for

themselves. The write-up should be well-balanced and unbiased and should reflect the impact of the individual and what other team members have to say about them. This helps provide holistic feedback for people to learn and reflect as they plan for the next year. Also, documentation helps provide a clear black-and-white picture of the support manager has provided to help the individual grow. What we should do is give the employee some time to read through the feedback the EM has written, let them process it, and then start discussing the strengths to recognize all the wonderful work of the individual. With that, let's segue into the opportunities or growth areas for next year and how you, as manager, would support them. Our main aim is to ensure the employee feels the discussion is a safe space and it's a two-way open communication. A few other tips that can come in handy are:

- Be a patient listener during the conversation, you also want to hear what the employee has to say
- Keep the conversation future focussed and try to understand what motivates the employee. A resource to refer to is the book, '[Why motivating people doesn't Work and what does'](#) (<https://www.amazon.com/Motivating-People-Doesnt-Work-Second/dp/1523004126/>)
- Keep a noise-free and distraction-free area to conduct the conversation, be it in person or remote. If it's a virtual conversation, make sure both sides are on the video to understand the facial expressions as well
- Foster transparency and drive consensus. Of course, you have to be prepared for curveballs as not all these conversations are easy
- Make sure to use the right phrases, there can be two ways to say the same thing. As an example, you can appreciate Alice by acknowledging their hard work over a weekend whereas you can share the incident and appreciate how critical the issue was, given the impact on the customers and how Alice went above and beyond to tackle the incident. Remember, everybody loves praise. They should also understand what are the skills they need to improve.
- This conversation may include discussing financial matters like promotions and compensation increases based on performance. Be prepared for such discussions. Your company might offer training to

help you grasp the compensation structure and the underlying philosophy behind changes. It is crucial to effectively communicate these aspects to employees when explaining adjustments in compensation.

- Prepare FAQs for the team members ahead of time or point them to HR resources to tackle similar questions again and again

As an engineering leader, our job is to keep our team members happy and inspired for career growth conversations. One can usually spend the first part of such conversations looking back at what the direct report has achieved since the last sync-up. It is a time to celebrate if they have improved and done a good job!

Next, you can discuss the areas that need improvement or what the best steps would be for them in their career growth. Also, you need to discuss action items and tackle those in the future one on one.

In the end, summarize what was achieved, congratulate them once more, and encourage them to celebrate the outcomes. Such meetings should not be treated as a one-time chore, but rather a continuous opportunity to work with individuals and help them plan a career roadmap.

#### **4.3.4 Handling pushback on the feedback**

Is it realistic to anticipate that every conversation with our team members will be without any challenges? If you believe so, you may be in for a surprise. Engaging in positive conversations, particularly those involving top performers and highlighting strengths, may generally be smoother (though not always) compared to more difficult and critical discussions addressing diminishing performance or underperformance. Also, sometimes employees might push back due to the fear of looking incompetent or if they feel the feedback jeopardizes job security. As EMs, we need to be smart about handling such scenarios so as not to hurt any team members and instead build trust with them. Some ideas are shared below. An expected negative/hard conversation with an employee can also be very stressful on the EM. Hence even for EM, it requires more effort and

preparation if they were to deliver bad news. It's hard to do such crucial conversations with both the employee and the EM.

Team members may not fully agree with everything mentioned in their performance review. They might have their own impressions about their performance, which may differ from reality or how their peers perceive it. As a manager, be prepared for pushback or defensive responses in such situations. Here are some key points to consider in handling these scenarios:

- Provide clear feedback on what is working well and what needs improvement, supported by specific data and instances. For instance, acknowledge and appreciate instances of going above and beyond, like handling critical production issues effectively. Conversely, address incidents such as code being pushed to production without proper testing and its negative impact on customers. While immediate feedback is ideal, reiterating important points during the performance review helps ensure clarity and discussion.
- Performance reviews should primarily consist of topics previously discussed in one-on-one meetings with team members, avoiding surprises. Maintain notes from these meetings to reference and reflect on progress over the last six months to one year. Encourage active participation from team members during the review meetings.
- Avoid false praises, deferring feedback, watering down critiques, and comparing employees. Evaluate individuals based on the company leveling guide, not against each other. Supplement these conversations with pulse surveys and peer feedback to ensure holistic and multi-perspective evaluations.
- Practice humility and patience, avoiding speculations. Utilize the career growth framework and emotional intelligence skills for guidance.
- Share the roadmap and your support as a manager. For instance, if your engineer faces communication challenges with non-technical partners, pair them with a mentor from the business organization, provide coaching on adapting language for the audience, and suggest relevant LinkedIn learning resources. This approach clarifies

expectations and empowers employees to take necessary steps for improvement.

- Follow-ups are essential to complete the cycle. Regular check-ins with team members ensure they are progressing appropriately. If there's stagnation or deviation, step in to offer assistance.

Keep the mentioned tools handy to handle team members who push back or are not aligned with feedback. If needed, be assertive and have tough conversations. Utilize the toolkit, maintain composure, and rely on data for these crucial discussions.

#### **4.3.5 Performance review template**

Each company follows a different format or template for conducting performance reviews. Some smaller startups or mid-size companies might not follow a consistent process due to a lack of standardization. Below is a sample performance review template that can be used in case your company is missing one. The idea is to cover the basics we discussed above, evaluate the individual's work, reflect on what was done and what are the growth opportunities for the future, and align them as the next performance cycle is kicked in. Imagine you have an employee named Alice on the team, where we need to recognize her good work throughout the year but also remind her of upcoming opportunities and growth areas. Her performance review might look something like this.

**Table 4.1 A sample template for performance review cycle**

<b>Employee Name</b>	Alice
<b>Employee ID</b>	1242221
<b>Department</b>	Software Development

<b>Review Period</b>	Jan 2021- Dec 2021
<b>Manager Name</b>	David M.
<b>Manager Evaluation Rating</b>	Top Tier
<b>Rating Scale</b>	1(Assuming a rating scale from 1-3 with 1 being top tier and 3, the lowest)

What are some of the way's this individual created the most impact in the review period?

(Make use of company leadership principles and career leveling guide to refer)

<b>Employee Evaluation</b>	<b>Manager Evaluation</b>
In the first half of the year, I worked on the delivery of project ABC which involved creating a complex categorizer engine. I collaborated with cross-functional partners and helped increase the test coverage from 30% to 70%.	I agree with what all Alice has shared. On top of it, I will add that Alice has shown customer obsession by helping resolve a mission-critical customer bug(within 2 hours of the incident) that had a blast radius of 20k customers. Further, she showed a bias for action by documenting her learnings and creating an operational runbook for troubleshooting production deployment issues for on-calls.

What are some of the ways this individual can increase impact/opportunities going forward?/ Areas of improvement/Growth opportunities

<b>Employee Evaluation</b>  Focus on depth of knowledge in services and infrastructure	<b>Manager Evaluation</b>  Alice has an opportunity to help with hiring initiatives for the company. She can help with interviewing and helping onboard new hires to the team. For this, her manager has paired her with a mentor to coach them on hiring and mentoring other junior engineers.
--	---

### **Any additional comments/notes?**

N/A

<b>Employee</b>  Aligned, accepted	<b>Manager</b>  Aligned, accepted
--	---

As you can see, using the template above, we recognized the hard work of Alice, and at the same time, using this as an opportunity to identify some growth areas to help her improve for the next year and provide some focus points. As EM, during performance reviews, ensure the data entered by individual employees and the manager is data-driven and is not subjective. This helps ease conversations and for the performance conversations to stay on track. Once the review is delivered, the process is completed by both

parties aligning and accepting the performance review, moving it to a terminal state.

In the next section, we will look at managing high performance and how to identify the potential to support them.

## 4.4 Managing high performance

is a skill that is often neglected by EMs as they are busy putting out fires with project timelines or deadlines with underperformers which we will focus on in our next section. The general bias of high performers being self-organizing and can easily run in auto-pilot mode brings in this complacency of neglecting them when time is a constraint. We need to remember that high performers are folks who are like a wish come true and are not only a gem for the company to retain but also have a very high cost of backfilling them if we lose them. Let's look at some of the characteristics that help identify someone in the team who is a high performer.

### 4.4.1 Identifying high performance

Understanding how to identify high performers helps distinguish team members who consistently exceed their expected responsibilities with a passion for going the extra mile. They boost the team's development velocity and inspire and motivate other members. These individuals are highly sought after in the job market and are the go-to persons within the team for questions and technical doubts. Recognizing the traits of high performers also aids in establishing a baseline for hiring top talent in the industry. Let's examine some of these traits.

- **Proactive rather than Reactive** - High performers are proactive self-starters who can anticipate and address issues before they arise. They rely on their intuition to plan for potential risks and bottlenecks ahead of schedule. One of my engineers exemplified this trait during a period of rapid company growth and expansion into multiple marketplaces. The challenge was to add a new configuration file for each marketplace, requiring manual configuration changes and production

deployments. This process resulted in significant development efforts and deployment time for each new marketplace. However, the team member took the initiative and created a tool to streamline the process. The tool offered a user-friendly interface to automatically and safely add the necessary configuration files, eliminating the need for manual code changes by engineers. It also performed formatting checks to detect and fix bugs. Once confirmed, the tool would automatically merge the code and submit a deployment request. As a result of this innovation, the cycle time from code change to deployment was significantly reduced from several days to just a few hours. This team member's bias for action and proactive problem-solving demonstrated their high performance and commitment to improving efficiency within the team.

- **Growth Mindset** - High performers exhibit a growth mindset where they are interested in not only the immediate task at hand but also are open to big challenges, and see the big picture and how it propels growth for the company. This way, they can think out of the box and bring innovative ideas to the table. As an example, imagine a movie provider app, one might be interested in adding a new carousel panel in the mobile screen today to show the latest movie releases, but what if, in the future, we need to add another carousel that can help categorize the new movies based on their categories like Drama, comedy, etc. This will then require some rework. What if the app is developed keeping a growth mindset and server-driven architecture is used for it? What that means is any new carousel can be added or updated by a mere change in server-side code that is controlled by backend engineers and can help avoid the mobile deployment cycles that span over weeks.
- **Go-to person** - High performers are the reliable individuals that team members turn to for assistance with debugging and technical matters. They possess a coaching mindset, willingly sharing their expertise and knowledge with others. I once had a high performer in my team who stood out because they were not only the go-to person for backend-related technical issues within the team but also for engineers outside the team. This team member went the extra mile by conducting weekly office hours, offering 15-minute slots to engineers seeking guidance on

technical matters. Their genuine interest in the success of others is what truly sets them apart and makes them invaluable to the team.

- **Authority** - High performers thrive in authority and hence like to be accountable for their work and work with minimal intervention from others. They demonstrate a high work ethic and are reliable. They like to take control and move things forward to the finish line.
- **Self-motivated** - High performers are members who are self-motivated, self-aware of their strengths and weaknesses, and self-organizing to some extent. This helps them to bring the best out of themselves and be an inspiration to others. I had an engineer on the team who was self-motivated towards operational excellence and helping reduce tech debt in the team. She went on to bring more awareness to this in team meetings and started doing small proof of concepts to share with other team members. Slowly and gradually, she helped bring awareness, and from being the only one picking such operational tasks in the sprint, ended up bringing a culture in the team where every engineer started picking 1-2 story points worth of story on operational excellence.

The above are some of the traits that help distinguish a high performer from the rest. They are approachable and willing to help others in a team by acting as a role model. It is essential to show and let high performers know the organization values them. Let's look at some ways how, as EM, we can support the high performers.

#### **4.4.2 Supporting high performers**

High performers have a higher impact on productivity and company culture. They are the ones who are often neglected, but given the right opportunities and support from leadership can really help take the organization further. Let's look at some common ways we, as engineering leaders, can support our high performers for sustained performance.

##### **Autonomy and authority**

High performers thrive in autonomy. Such folks should be supported by staying away from micro-management and giving them the space to learn and grow. Whenever the situation permits, EMs should aim to delegate work to them to help them grow and create opportunities for their learning. Remember, these folks might sometimes be stretched thin as they work on helping other low performers around them. At the same time, as EM, make sure they do not end up as single-point failures of your team. What if they win a lottery tomorrow and decide to retire? Would your team be able to manage without them? It is crucial to keep the “Bus Factor” in mind at all times. The bus factor, representing the number of team members who could be hit by a metaphorical “bus” (i.e., leave or become unavailable), emphasizes the importance of distributing knowledge and responsibilities among team members to prevent critical knowledge loss and maintain productivity. As EM, you are responsible for supporting your high performers but also with the expectation that they help coach and mentor other junior engineers to avoid being a bottleneck for the team.

## **Growth opportunities**

To support high performers and foster their success, we should create and share growth opportunities with them. This involves providing career support, such as assigning challenging projects (stretch goals), connecting them with mentors for career development and networking, and opening doors to opportunities beyond their current team or organization. The goal is to nurture their skills and bring out their best potential. I have a close friend who was a senior software engineer and a top performer in his team. He expressed a passion for transitioning to product management, which aligned with his innate abilities. Recognizing his aspiration and strengths, his manager promoted him for his exceptional engineering talent and facilitated his move to the product management role within the same team. Today, he thrives as a successful product manager. To further enhance his product management skills, I suggested that he join the mentorship program at FirstRound Fast Track. This example highlights the importance of engineering leaders prioritizing growth opportunities for team members,

even if it means supporting their transition to different teams or roles outside the immediate organization.

## **Formal recognition**

Every high performer will have a different craving to acknowledge their good work. Formal recognition is something that is one of the most sought-after ways high performers feel valued. This involves shout-outs in team and leadership channels, formally recognizing them across the company's leadership principles, and sharing this in all hands to bring them visibility. At the same time, you can have an introverted high performer in the team who might not be very comfortable with public recognition. In the chapter on rewards and recognition, we dive deep into understanding the various forms of recognition and how to go about them.

## **Compensation**

Compensation is one of the ways we can show that high performers are valued. The majority of organizations have performance reviews that help review the performance of team members and, in turn, show an increase in their annual compensation as part of total compensation or release some form of spot bonus for going above and beyond the roles and responsibilities. As high performers are always sought after in the technical industry, it is important to compensate our high performers well and compete with industry standards.

## **Constructive feedback**

Constructive feedback involves providing data-driven information with positive intentions to help individuals identify strengths and weaknesses, enabling them to work on areas for improvement. Supporting high performers through constructive feedback allows them to learn and grow in their roles by honing specific skills and accessing appropriate resources.

For example, one of my high performers struggled with delegation, impacting their performance. In a one-on-one meeting, I shared instances where delegation could have been beneficial and how it affected their overall performance. Utilizing a data-driven approach, I recommended LinkedIn learning training and a delegation framework. Additionally, I arranged a mentorship with a senior engineer from another team to help them develop their delegation skills.

By providing timely feedback and support, this team member successfully leveraged delegation and maintained their high performance.

## **Keep them challenged**

High performers thrive when kept challenged. This means the projects and problems they work on excite them, help them understand how their efforts fit the bigger picture, and hence help them grow. As EMs, we need to provide such opportunities to our high performers so we don't limit them, at the same time being aware of not causing any burnout. The idea is to set challenging yet realistic goals as these individuals respond well to increased responsibility. I worked with a high performer in my team by identifying an organization-wide problem- a missing mentorship program for backend engineers. The company had several mentorship programs to help the IOS engineers, and web developers but somehow had no major mentorship support for backend engineers. Together we brainstormed a few ideas, put together a one-pager, and eventually worked to make this idea a reality. Together we set up some challenging yet realistic goals and ensured we understand that it is ok to say no to avoid burnout with so many other things on our plate.

As EMs, honing the skill of managing and supporting high performers is crucial. Let me share two examples of high performers I've encountered with different experiences. Firstly, Alice, an expert backend engineer, excelled technically but lacked a team-oriented attitude. After persistent discussions on fostering empathy and supporting colleagues, Alice eventually received a promotion, albeit later than expected. On the other hand, Bob, a skilled engineer, demonstrated teamwork and approachability,

inspiring the team and earning recognition from cross-functional partners. Due to his leadership and effective collaboration, Bob received a well-deserved promotion, meeting the high standards outlined in the company's leveling guide. Now, let's move on to managing low/underperformers, a relatively difficult conversation

## **4.5 Managing low/under-performance**

Let's start with answering the simple question- What exactly is underperformance? Underperformance, in layman's terms, is performing 'under' the expected bar at the job role. This means the employee is not performing the job duties as expected and is falling short, either by not meeting project timelines, bad quality of work, or bad behavior at work, all of these, in one way or another, impact the overall team performance. This leads to an overall decline in morale for the team.

It is equally important to manage underperformance as it is to motivate and manage our high performers. The key to successful performance here is to identify any instances of underperformance early and in a timely manner. This gives a fair opportunity to the employee underperforming to course correct actions, given the right support, and also to the organization as a whole to plan better in case there are no signs of improvement. Now let's look in detail at a framework to manage underperformance around you where I share my own personal experiences of having dealt with such difficult situations as a manager.

### **4.5.1 Framework to manage underperformance**

Underperformance requires due diligence and root cause analysis to help the individual who is underperforming, as well as the business unit. As said earlier, it is important to advocate for the importance of managing underperformance as EMs. This information should not come as a surprise to the individual, as forums like one on one sessions and performance reviews are meant to provide continuous actionable feedback. Let's dive

into an easy-to-use framework for underperformance that I have used in tough situations.

## **Identify the underperformance**

The first and most important step is to identify the cause of underperformance around you or in your team. Now you may wonder, all my engineers are awesome and deliver results on time, with a few situations here and there. Well, this is our personal bias to believe that everything around us is just fine. It is the skill of reading between the lines and keeping our eyes and ears open to identify such underperformance cases.

I had a junior software engineer, John, on my team. After transitioning to this team as their new EM, I noticed consistent disengagement and missed deadlines, which was unexpected given the positive feedback from the previous EM. During biweekly one-on-one meetings, John provided various excuses for the delays and showed a lack of attention to detail. His productivity was significantly lower than other team members, taking three times as long to deliver features with bugs and requiring twice as many code review revisions. Additionally, John hesitated to demo or review progress with the team. I immediately investigated the root cause behind this behavior.

## **Root cause analysis**

When dealing with underperformance, it's crucial to understand the reasons behind it objectively, setting personal biases aside. Several factors could contribute to underperformance:

- Lack of technical skill sets
- Personal reasons, like health issues or family concerns affecting emotional and physical well-being
- Poor role fit, where an employee's skills don't align well with their current position
- Insufficient tools and resources for success

- Lack of clarity about role expectations and scope
- Shift in interests, where employees become more interested in other roles or responsibilities

In John's case, I wanted to uncover the root cause without biases. During a candid one-on-one, I asked if there were any personal issues affecting his work. However, he had no compelling reasons and was simply avoiding tasks. My next step was to share consolidated feedback with him to discuss the impact of his work.

### **Share consolidated feedback**

This step in managing underperformance is critical in sharing the news and keeping a data-driven mindset. In this step, we gather 360 feedback for the team member from other peer engineers, cross-functional partners, or other EMs that the engineer worked with. The idea is to get a full perspective on the work and behavior of the employee to appreciate what went well and what are the opportunities for improvement.

So it was time to share the consolidated feedback with John. I shared the exact misses of him by keeping a data-driven approach with a paper trail to refer to. The 360 peer feedback came in handy as it gave an idea of how others also feel around him. I shared what is working well- his work as the scrum lead for the team, and what is not working well- his missing project deadlines and producing poor-quality code. Next, I referenced with him the company leveling guide to set up a gap analysis document to identify the gaps in his performance at the current level in terms of expectations versus reality. This would help us set up a plan in place to help put John up for success.

### **Performance improvement plan (PIP)**

The next step in the action plan is establishing a structured performance improvement plan (some companies refer to it as a performance empowerment plan), also known as a coaching plan. This plan translates the

consolidated feedback into actionable items, providing a concrete course of action with estimated timelines to ensure time constraints are respected. In the spirit of transparency, clear communication was made with my employee, John, regarding what will be measured and how it will be measured. Additionally, the consequences of not meeting expectations were conveyed. Initiating a performance improvement plan is a delicate and challenging process that should only be considered if the employee cannot incorporate your feedback and is not making significant efforts to improve. Starting a plan of this nature typically involves involving an HR representative from the company, and if the plan's outcomes are negative, it can result in the employee's termination.

So John and I brainstormed together a coaching plan for him with specific competencies in mind with clear timelines in place. We timeboxed it to 90 days(you can work with your HR to see if they would like a 30, 45, 60, or 90-day plan), working together with HR, to ensure we give enough time for John to succeed. The idea was to ensure John gets full support from me, his EM, and the company as a whole as we worked on setting some SMART goals (covered in Chapter 3) for John. Such goals should be very specific so that there is no ambiguity as to whether the metrics were met or not. So for this situation, I kept myself approachable at all times and listed a structured coaching plan. As an example, a few things that we had there were (this is simply an example to illustrate how a coaching plan expectations should look like):

- Code Reviews- The comments must be addressed in two business days, and code reviews must be moved to a terminal state, with emphasis on increasing code quality to avoid more than four code revisions.
- Timelines/Delivery and execution- Deliver ‘x’ story points by the end of the sprint, unless any technical blockers are identified, and the agile velocity of the team is well defined
- Documentation/Knowledge sharing - Contribute to improving the team’s documentation with clear correspondence added on trouble tickets

Above is a list of things of what was expected out of John, let me share a list of things that we aligned on that will be provided as support to him so he succeeds in the coaching plan.

- EM to empathize with the situation and be approachable to John at all times, except when in meetings or off business hours
- John was paired with a buddy in the team to help with technical guidance support. This made things like pair programming or an extra pair of eyes before the code review was sent out to the rest of the team members possible
- Instead of biweekly 1-on-1s, shifted to weekly 1-on-1s for this individual to provide faster support and feedback. Each 1 on 1, we had clearly defined goals for next week and what was completed last week.
- Made a Slack channel with John, the buddy, and myself to cater to questions faster
- Shared learning resources like internal learning videos and technical documents to help brush up the technical skills

**Note:**

If it's a case of misconduct, keep your HR involved and you might not even have to go through a coaching plan for the same. Usually, cases of misdemeanor or misconduct are handled directly by HR.

The key is to ensure the timelines expected are clearly communicated so there is no subjectivity to what is expected out of the individual. So as part of the coaching plan, not only do we share what is expected of the employee, but rather how and what support will be provided to them to achieve the same. A copy of this is usually maintained in the employee's file maintained by HR for future reference. PIP is like a precursor to drastic steps such as firing an employee if they show no signs of improvement.

## **Monitor and review**

Once the feedback is shared, the next is to monitor and review the progress and share early feedback so the employee can course correct if digressed. Weekly 1 on 1s is a good forum to share the progress and give constructive feedback.

The feedback was shared with John in the weekly 1 on 1s and was clearly documented over emails. Every time we met, I would share written minutes of the meeting in terms of what we discussed, the progress made, and any feedback shared. There were a few instances where the employee would simply come over the call and cry, as an EM, I could empathize with their situation but had to be fair and go with the leveling guide of the company. The employee showed little to no signs of improvement and was unwilling to cooperate.

## **Final call**

The last phase in the process is to measure the individual's success and take a final call. Now this can go two ways.

- Firstly, the person did a good job and could meet the defined expectations put in the coaching plan in a timely manner. This means they performed well against the coaching plan and successfully got out of the coaching/focus/development plan. Great news!
- Not always do things go as expected, so it can happen that the person is unable to cope and continues to not perform the expected duties in the role. This calls for some tough discussions. You will likely need to meet HR to discuss the outcome of the PIP and seek their guidance on different options that the employee can have. Most companies will offer the option of taking severance and letting the candidate go for the good.

Suppose the employee chooses to take the severance package as they are fired from the role due to performance reasons. In that case, the company may pay them the salary for the next 'x' months, depending on the company's rules, and the employee will no longer work for the company.

They can go ahead and find a job that fits better with their work style and expectations.

In the case of John, since he was unable to meet expectations and cooperate with the plan, I had to get HR, him, and I in the same room and do the harder conversations of offering PIP and the other option of taking severance and leaving. In this case, the candidate was persistent and chose a formal PIP plan.

#### **4.5.2 You win some, you lose some**

Regarding John, if an employee cannot improve their performance and meet company expectations, tough decisions like letting the employee go may be necessary. However, not all cases of underperformance result in termination. Let me share some of my experiences with underperforming employees and how they were resolved.

In a previous team, I had a software engineer II who had been a strong performer for several years but showed signs of disengagement in the last six months, with repeated missed deliverables. After investigating the root cause using the framework mentioned earlier, I discovered that the employee had developed a keen interest in project delivery and execution instead of coding. During regular one on ones and feedback sessions, I discussed this with the employee and conducted a gap analysis based on the job role description. Recognizing their newfound interest, I suggested exploring the role of a technical program manager (TPM) within our organization. I paired them with an existing TPM for shadowing and understanding the role better. We agreed on a three-month evaluation period to ensure the role was a good fit. With regular check-ins and early feedback, the employee successfully transitioned to the TPM role after three months. They excelled in leading critical projects for the company, going from an underperforming employee to a rockstar TPM. In the above case, the employee showed a shift in interests, and I helped them transition to a role that aligned better with their motivations. In another instance, a different engineer was identified as underperforming due to technical skill gaps. We collaborated on a plan to address these gaps, providing mentorship, training,

and support. Eventually, the employee improved and became a solid software engineer within a time-boxed plan of sixty days. As engineering leaders, it's essential to share relevant information with the broader team. In the next section, we'll discuss dos and don'ts for effectively sharing such information.

#### **4.5.3 Communicating the outcome to the rest of the team**

Matters involving employee performance are part of the confidentiality agreement between the employee, HR, and the manager. If an employee is managed out, communicating such news can be painful and, at the same time, difficult, keeping in mind confidentiality. In such matters, it is paramount to coordinate with the company's HR and Legal departments and work with them to get guidance on such matters. I will share a few things to remember as you communicate such news to the rest of the team. Keep in mind that this is based on my knowledge gained over working at various companies and in no way is a silver bullet rule. Please work with your company's HR and Legal team on such matters.

- Calling for a team meeting to update that an employee no longer works at the organization. Check with your HR about what can and cannot be shared. This proactive message from the manager's end avoids any speculations from team members, keeping in mind the company's rules and regulations
- Expect people to ask questions and reasons, but since there are legal complications, either bring in your HR and legal team to handle those questions or just mention the employee has moved on to pursue other opportunities and let's respect their confidentiality
- Reassuring others that they are secure and not to expect any immediate staffing changes (unless this change was part of a layoff, in those cases, we want to be careful not to promise something that might not be even in our control)
- Remind your team members that it is their responsibility to avoid any defamation of the terminated employee and treat the matter professionally

- As an EM, keep an open door and ask them to reach out to you for any questions or concerns
- Convey that with this change, there will be some impact on the work re-distribution and load balancing among the rest of the team

These are some of the most crucial conversations you will handle as an engineering leader. Key things to keep in mind are you are not alone in this, seek help from the HR and legal department to ensure you are abiding by the formal law and regulations, and demonstrate empathy, in the end, remember, there is another human being on the other side of this who is actually going through the situation.

#### **4.5.4 Managing an underperforming team**

Managing an underperforming team is an onerous task, especially when you are new to the team and are yet to learn the team dynamics. While there is no silver bullet solution to managing an underperforming team, one can follow a similar process as shared for managing underperformers, where you start with identifying the underperformance. As a team, there can be several reasons for the team to underperform- ineffective leadership, lack of skill sets, lack of motivation, etc. One common scenario to identify if the team might be underperforming is if the team is going through tasks spillover every sprint. This means the team committed to, let's say, finish 14 tasks in the sprint and ended up just doing 8, leading to a spillover of 6 tasks. If the trend continues sprint over sprint, it is definitely a case of an underperforming team, and one needs to find the root cause to understand what the gaps are. Let me share an example where one of my team was underperforming. So my team of six engineers was repetitively spilling over story points from one sprint to the other. In various instances during standup, it was identified that two engineers had picked up the same story task(a sign of ineffective communication), leading to duplicate efforts. In one of the other instances, one of the engineers went on a week-long vacation without clear communication with me as manager and the teammates. The engineer had a time-sensitive task assigned to them in the sprint. Had proper communication been done, as manager, I could have carved out a transition plan for someone else on the team to take over the

time-critical task. Also, in my 1 on 1s, three developers mentioned code reviews remaining open for a long time and waiting on peer feedback and comments. All these instances pointed to a lack of communication and collaboration within the team, leading the team to underperform.

Now let's look at some key things to keep in mind as you try to handle and turn around an underperforming team and then see what steps I took to help with managing my underperforming team.

- **Observer and learn** - The idea is to start by observing the team and being a patient observer by keeping the urge to 'doer' away. This is the time to listen to your team members, acknowledge the problem and understand what they feel is working and what is not working.
- **Set clear expectations** - Once you identify the gaps from observation, use the time to effectively communicate them with the team and set clear expectations regarding what is expected. This can be the agenda for team meetings and ensuring to create a safe environment for people to speak up their minds. If you feel the team is not candid in a public setting, send anonymous surveys to collect feedback. Also, communicate the team mission and timelines being targeted to provide clarity to team members.
- **Support** - You, as the EM, have the crucial role of supporting your team by providing the right set of tools and resources for them to succeed. Find ways to pair people with mentors if they are struggling or point them to learning materials to help them hone their skills.
- **Monitor and provide feedback** - Trust your team but verify to avoid later surprises. Keep track of the team's performance and ensure the progress is in the right direction. Also, provide constructive feedback to the team individually, identifying their strengths and areas for growth.
- **Crucial conversations** - You will do everything you can from your end. But if there still are some members that continue to underperform or are not receptive to team-level changes and feedback, you, as EM, will have to step up to take hard decisions. Since underperforming members can cause a domino effect, fixing this in a timely manner is the need of the hour.

So keeping the above principles in mind, for my underperforming team, I acknowledged the problem and started from the ground up. I called a team meeting to share specific instances where the team had dropped the ball to keep a data-driven and factual discussion. I then asked my engineers for their input and suggestions on how we can improve as a team, making them part of the solution-finding process. I patiently listened to them and was open to feedback from team members and also the cross-functional partners. Together with the team, we came up with a game plan and decided to take some measures as below:

- Actively make use of Slack group to communicate
- When someone decides to pick up a story, they are responsible for real-time updating the Jira board(a tool we used to keep track of all stories in a sprint) to ensure no one else picks up the same work
- If someone is heading for a vacation, send an out of office invite to the entire team and bring up in standup if they need to switch on-call schedule or feel any portion of the project might slip over due to this so actions can be taken in advance
- Also, setup additional 15 mins team catchup time with open agenda, so people can get some facetime, and open the floor for open conversations

Using the above measures, paired with coaching individuals during the 1 on 1s, helped to turn around the team and increase productivity. Handling underperformance, be it at an individual level or the entire team is one of the most difficult tasks to tackle as engineering leaders. With experience and practice, one can learn to navigate the situation by considering the fundamentals shared above. Next, let's look at what experts say about managing performance based on their experiences.

While much of our focus tends to be on high and low performers, it's important to give attention to average performers within the team. These individuals play a crucial role in keeping the work flowing. Although they may not stand out as high achievers, they consistently complete their tasks to the best of their abilities and within deadlines. They can be relied upon to get the job done and often act as a unifying force for the team. Additionally,

it's important to recognize that everyone experiences different phases in their personal and professional lives. Sometimes we excel, dedicating time and energy to our career growth, going above and beyond, and striving for promotions or raises. However, life circumstances affect us all, leading to periods where we can only perform at expected levels due to limited time, energy, or mental capacity. These periods can arise from various factors such as marriage, homeownership, relocation, parenthood, supporting sick friends or relatives, and more. During such phases, it is perfectly acceptable for individuals to maintain their current performance levels.

### **What do other leaders have to say about it:**

“Managing people is an art where you have to balance high and low performers in the team, along with keeping the average people happy. Always try to understand where the other person is coming from before jumping to conclusions.”

~ **Sanjay Gupta, General Manager at HCL Technologies**

“Performance reviews are an important part of being an EM and should be approached with utmost responsibility. I always look for outcomes achieved by an individual rather than the activities which he is doing. Being outcomes driven to rate someone is the right model as it serves good for both employees and the business.

A typical performance review should talk about outcomes in the tenure, areas of improvement, and peer perspective.”

~ **Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

“Managing people is very different from managing projects and deliverables. People have different needs and expectations that change with time. Keeping high performers engaged requires understanding their needs and expectations at different stages of their careers and being able to help them fulfill them. Some things high performers on my team generally

appreciate are visibility and acknowledgment for their work, the opportunity to solve complex problems, and being able to learn and teach new trends and technologies. Some other expectations I have entertained include accommodating different working hours for a limited period, remote/in-office working requests, and nomination to extra-curricular activities and conferences.”

**~ Adish Agarwal, Director, Software Development at Audible, Inc.**

“To grow managers, in many organizations, new managers are high-performing ICs who either are pressed into it organizationally (that’s the only way to continue growing) or are looking to see if it is a fit for them. I consider “team leaders” and “squad leads” in this grouping. Frequently, IC skills do not translate directly to management skills. As a result, new managers need patience.

For more senior managers, I use manager growth as a mechanism to also demonstrate my career growth. By delegating tasks to managers above their role and helping them manage through those tasks, you increasingly demonstrate that they are growing into the next role and that you can grow folks into that role. In the past, this has included demonstrating that a manager working for me was ready for my role, and since I was able to grow into that, so was I.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Engineering Leader**

“Many first-time managers (myself included!) don’t realize how different it is to manage vs being an IC. Your job is no longer to produce work but to build the team that produces the work. I help my managers frame their efforts and measurement in the three axes of people, process, and product.”

**~ Jean Bredeche, Head of Engineering at Patch, (ex) Robinhood, Quantopian, Hubspot**

“For managing performance, providing feedback early and often if important. No sugarcoating low performance and providing constructive

feedback is the key. Data collection and preparation of the message are key. Getting feedback from teammates and other external stakeholders helps to get a full picture.”

**~ Rajakumar Sambasivam, Delivery Manager at Microsoft**

“For managing performance, we must continuously observe our team member’s performance and provide in-time coaching and feedback. My top three practices are

1 Set clear KPIs for the role.

2 Provide in-time coaching and feedback aligned with their goals

3 Recognize small wins.”

**~ Sumit Kumar, System EM at Cisco**

“To grow managers, let people run with things, get them training and use one on one to learn about them.”

**~ Larry Gordan, Managing Director at Emtec Inc.**

“Performance review discussions should never come as a surprise to either party. As time passes, email yourself notes and collect tidbits of information about personal successes and efforts, challenges, acknowledgment from others, and areas where growth was noticed for each of your direct reports. These will help you when it comes time for crafting an accurate and SMART review that will help your employee improve and grow.”

**~ Bruce Bergman, Manager at Lytx**

## **4.6 Stop & Think: Practice questions**

1. Does my company value the importance of performance management?
2. Are my engineers aware of the roles and responsibilities expected of them at their job level?

3. What are some of the best practices I have seen other managers exhibit when it comes to managing performance?
4. Does my organization identify leaders from within the organization and help develop them?
5. Does my organization help provide the right support and training resources for the individuals to grow?
6. What is one of the most difficult conversations I have had with an employee?
7. Why was it difficult for me? If I were to go back in time, what would I change about the conversation?
8. How can I recognize my team members for their good work?
9. Are there any gaps in the current performance management system in my organization?

## 4.7 Summary

- Managing performance is an art that helps the manager understand their team members' motivations and aspirations and brainstorm with them a roadmap to get to the goal state. In the process, managers provide coaching and frequent constructive feedback to help the individuals grow.
- Performance management is a structured approach to monitoring employee performance, fostering sustained productivity and engagement. It offers career planning opportunities, proactive feedback, alignment with strategic company goals, improved retention rates, and facilitates future hiring planning.
- Managing performance is a complex task that requires careful consideration. It is not a one-time chore but an ongoing cycle that involves reflection and identifying developmental opportunities.
- To ensure effective performance management, it is important to practice transparency, provide fair and unbiased reviews and feedback, avoid the halo-horns effect, recognize individuals, promote open communication, offer necessary training and resources, and maintain a continuous review process.

- Performance reviews may initially appear time-consuming and cumbersome, but they play a crucial role in career conversations and the development of employees under your leadership as an engineering leader. The performance review consists of an:
  - Evaluation phase - where based on the work done, you provide a potential rating to the individual. This is the time to gather 360 feedback and adjusted expectations if needed.
  - Calibration phase - In this, we calibrate the individuals across various teams and organizations, usually up to directors and VPs. HR is present in such conversations and ensures there is fairness and consistency across the company.
  - Preparing and conducting performance review - During this phase, the manager presents the performance review, schedules the meeting, allows buffer time for any employee questions, and comes prepared with a data-driven write-up. Difficult conversations may arise, and employees may push back. As an engineering manager, it is important to navigate such scenarios tactfully.
- Managing high performers is often overlooked, but it is crucial to remember their value to the company. High performers are a valuable asset, and the cost of replacing them can be significant. To effectively manage high performers, it is important to provide them with autonomy and authority, show appreciation through compensation, keep them challenged, and much more.
- On the other hand, managing under-performers can lead to some crucial difficult conversations. A framework to manage underperformers is to start with identifying the scenario of underperformance, which can sometimes go unnoticed, then doing a
  - root cause analysis to get to the bottom of it(personal reasons, technical skill set gaps, etc.)
  - sharing 360 consolidated feedback with the individual to share the overall feedback
  - If needed, put the person on a performance improvement plan (PIP).
- The PIP will be time-boxed and at the end of it, you will work with HR to take a final call- if the person was able to show improvements-

great news. If not, they will be asked to leave with mostly a severance package paid by the company.

- Situations like managing out are critical and can be stressful at times. It is important how and what you communicate to other team members to keep a balance of confidentiality for the parting employee and, at the same time, avoid speculations in the team. This also calls for planning how work will be load balanced after the employee has left.
- Managing an underperforming team can be challenging. While there is no one-size-fits-all solution for addressing underperformance, you can follow a similar process as described for managing underperformers. This involves identifying areas of underperformance, setting clear expectations, and providing the necessary support through one-on-one interactions.

## 5 Delegation - Learn to let go

“Don’t be a bottleneck. If a matter is not a decision for the President or you, delegate it. Force responsibility down and out. Find problem areas, add structure, and delegate. The pressure is to do the reverse. Resist it.”

~ Donald Rumsfeld, former Secretary of Defense

### This chapter covers

- Concept of delegation
- Differences between delegation versus allocation versus substitution
- Understand the When, Why, Which, Whom, and how of delegation
- Framework for delegation
- Teaching delegation to others around us
- Delegation can go wrong

It's a Monday, 9 am, and you switch on your laptop to start your work week. I was excited to get a lot done this week. Boom, the first email in your inbox says, ' [Urgent]Need Operational Dashboard that shows your team's incoming trouble tickets by severity in the last 6 months.' Ufffff- an urgent task (just like several other tasks coming to EMs) and needs creating a new dashboard for this specific information. What should you do? You are already stretched thin as it's that time of the year when you work on performance reviews for the engineers with a deadline.

Well, the delegation comes to your rescue. Fortunately, there is an operational excellence lead for the team passionate about operational excellence and reducing tech debt. You set up a meeting with them and took it from there. Delegation is the assignment of authority or tasks to another person. This means one needs to trust the other person's abilities to perform the duties and patiently observe from the sideline as they perform the delegated tasks. A manager's role is to remove bottlenecks from any situation and help the team succeed by eliminating roadblocks, collaborating and bringing folks together to solve issues, opening up communication, and looking at the bigger picture ("systems thinking"). Delegation does NOT mean passing off work to another person without apparent authority and ownership. One might mistake delegation to be only for subordinates which is a common misconception - one can delegate to anyone at any level of authority. In this chapter, we will learn what effective delegation is and answer the various W's of delegation to understand when, why, which, whom, and how to delegate. We will then learn a framework for delegation that can be used in our day-to-day work to help us scale. Furthermore, not only do we hone our own skills of delegation, we will learn how to have a multiplier effect and teach delegation to others around us.

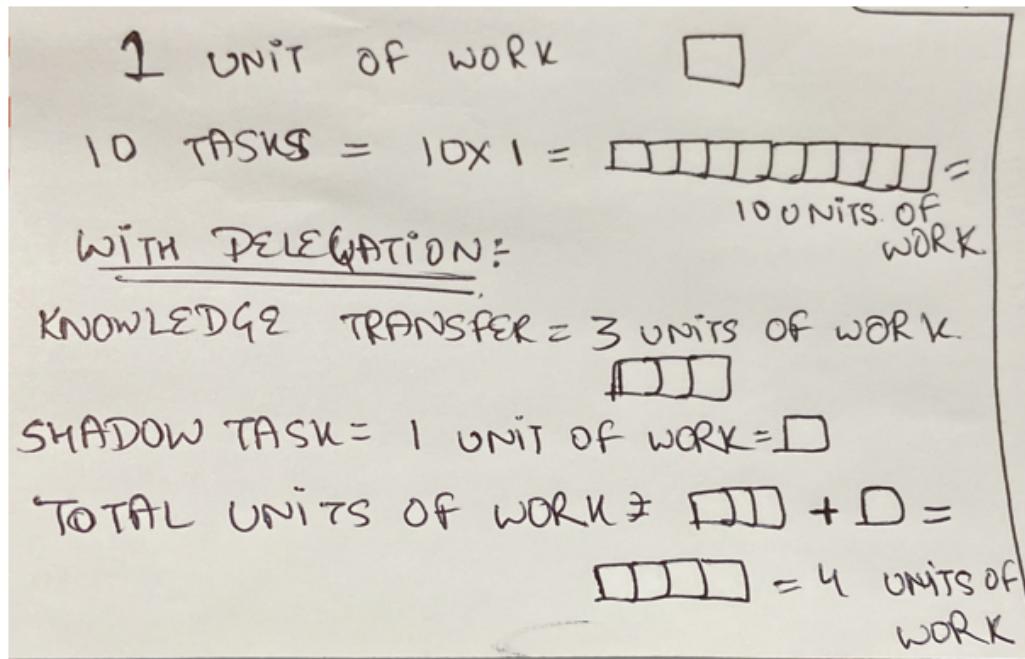
## 5.1 Delegation is an art

Delegation is an art where one needs to move away from the '**doer mentality**' and step into the '**coaching mentality**.' It helps you scale as you grow in your career and go up the leadership chain. Strong leaders have mastered the art of delegation to empower them. Someone has rightly said- delegation is like extending your presence through others' actions. You are getting more done by helping the hands of other team members by empowering them to finish the tasks successfully. At the same time, you ensure that each task gets the desired time and attention for it to be completed successfully. Yes, it's about the outcome and the delegation journey. But, one can shy away from the delegation for several reasons we will discuss next.

### 5.1.1 Idea of moving fast

We as humans tend to sway towards doing the job ourselves, especially if we have prior experience handling a similar task. We might feel the initial burden of teaching another person the task, so time spent in knowledge transfer can be intimidating. But let's be honest, once it is taught, you can reap its benefits for a long time. We always have this urge to use the principles from what worked last time and move fast to ship the task. We want to roll out our sleeves and jump into the weeds straight. This is only sometimes beneficial as with such a mindset, we cannot bring more ideas or perspectives to the table, hence lacking the spirit of innovation. Let's look at a very basic example using some basic math. Let's say we have 10 tasks that are similar in nature. When you do it yourself: Each task takes 1 day to complete. So for 10 tasks, it will be **10 days**. When you delegate and teach another person: A knowledge transfer session takes 3 days. You show them how to do 1 of the tasks(where they shadow you), another 1 day. This overall will take **4 days** as shown in the figure.

**Figure 5.1 Units of work with and without delegation**



But, after this, you have the task delegated, and you saved **6 days'** worth of your work.

Also, if such a task comes up again in the future, you have saved time for yourself by teaching it to another person and, simultaneously, removing yourself as a single point of failure. At the same time, an opportunity and skill are created for another team member. How cool is that?

### 5.1.2 Be the sole knowledge bearer

Similarly, we should be the keeper of knowledge, which results in us being a bottleneck to the team. Some EMs may not be open to loose control. This is detrimental to the team members as we, as EMs, are taking away the opportunity from others to grow and learn. We fear giving away knowledge and authority to others in the team and sending a wrong signal to our engineers.

Let's imagine a scenario where the senior engineer is an expert in the pricing engine of the company. Now this engineer has grown in the role and has been chosen to be the new EM for the team. The engineer (now EM) does not believe much in the documentation and hence becomes an EM with little knowledge transfer done to other engineers reporting to him. What is the problem here?

The engineers will struggle to fill in the gaps of the new EM as most knowledge will be tribal knowledge that remains undocumented. Further, each engineer will depend on the new EM, who will have difficulty scaling if they are a bottleneck for each upcoming project. Hence the need to spread the knowledge and avoid being the sole knowledge

bearer. We can also agree that this situation has been poorly managed by their previous manager.

**Lacking trust** - EMs may struggle with trust and exhibit controlling behavior, impacting team performance and morale. For instance, consider a scenario where a senior engineer with expertise in backend services had issues with punctuality and knowledge sharing. This engineer arrived late and unprepared in two project meetings, indicating time mismanagement. Consequently, the team's EM began distrusting not only the senior engineer but all team members, leading to micromanagement and excessive reminders before each meeting. In a few instances, this lack of trust and generalization caused confusion and weakened team morale. Additionally, the EM's time was consumed by managing calendars and sending reminders, detracting from their own responsibilities. This behavior reflects a desire for perfectionism and a lack of trust in others, preferring to rely on oneself rather than the team.

**The guilt of off-loading work to others** - EMs may feel guilty about delegating work originally planned for themselves. However, it's essential to understand that delegation doesn't absolve them of accountability. It means empowering another person to handle the task, which can contribute to their growth. Instead of feeling guilty, EMs should feel happy about delegating. It's crucial to realize that not all tasks assigned to the EM must be done exclusively by them and can be delegated to others, except for people management tasks that require the EM's involvement specifically. It's important to note a nuance regarding delegation when a new task arises within the team. If the task is delegated to an engineer, it can impact team velocity and focus. In such cases, the EM may take up the task temporarily to prevent immediate bottlenecks. The key is to foster shared responsibility and provide growth opportunities while being mindful of time management.

Consider a scenario where the EM is already overloaded with five crucial tasks. If they simply offload two tasks to a senior engineer without proper consideration, feelings of guilt may be valid. However, if the EM follows a systematic process of identifying suitable tasks for delegation, selecting the right person, and showcasing the value and career growth potential in taking up those tasks, it becomes an opportunity for the senior engineer to learn and grow. This approach creates a win-win situation, enabling the EM to scale their workload while offering meaningful development opportunities to team members.

All in all, delegation is an art that promotes trust and morale-building among the people involved in the process. Without delegation, EMs and leaders can feel overwhelmed with work and not channel the energy to the right set of tasks that needs their attention. Remember, you will succeed if your employees succeed, so it's your job to empower them to be successful. Although when the tasks are delegated, there will be minimal intervention from you to give others space. But, in the end, the one delegating still has to own the process and be accountable for it. Hence, the need to be smart about when you

delegate, why you delegate, what you delegate, whom you delegate, and how you delegate are essential in the overall equation of effective delegation. In the next section, we will learn about the nuances between delegation and allocation and how both are different from substitution.

## 5.2 Delegation versus allocation versus substitution

Now that we have been introduced to delegation, we need to learn about the core concepts of allocation and substitution and how they differ from delegation. Again, each of the three are ways to share the work responsibility to get things to the finish line. The critical difference is how they are executed and the kind of support the delegator provides to the delegatee during the process. Let's dive into how it differs from allocation and substitution.

### 5.2.1 Delegation versus Allocation

Differentiating delegation from task allocation is crucial in selecting appropriate tasks and individuals for delegation. Many people mistakenly consider delegation and allocation synonymous, leading to potential pitfalls. However, they are distinct approaches with slightly different outcomes.

For instance, let's consider the task of project metrics tracking, which is essential for monitoring project progress. In the case of task allocation, you assign the software engineer the responsibility of preparing and sharing the project report within the deadline. They attend check-in meetings and are solely responsible for timely submission. This approach focuses on task completion without much involvement beyond that.

Now, imagine an alternative scenario where you identify an engineer with interest or expertise in operational excellence, possibly the team's operational excellence lead. You sit down with them to discuss the task's expectations and develop a strategy, such as creating a report template for easy regular updates. You provide the necessary authority and support for the engineer to complete the task on an ongoing basis successfully. You also assure them of your approachability and readiness to assist when needed. This approach establishes a culture of trust, learning, and support, exemplifying effective delegation. In this situation, the emphasis is on facilitating the engineer's growth and learning rather than simply instructing them on task completion. By understanding the distinction between delegation and task allocation, we can foster a culture of continuous learning, empower team members, and enhance their long-term skill sets through effective delegation.

As you can see, it is all about the execution of how the task is asked for or taught that makes the difference. Let's look in the details in the table shown the differences between

the two:

**Table 5.1 Differences between allocation and delegation**

Aspect	Allocation	Delegation
Skills	Allocation is when we simply want the work to be done. That's it.	Delegation is a process where we want the work to be done, but along with it we are looking for some learning skills that will continue beyond the immediate assigned task.
Process	This is a more instruction-based method where the person follows instructions to complete the work.	This is a coaching/mentoring-based approach where the person is told what is to be done but not how it is to be done. The individual is given the responsibility and ownership to execute the tasks.
Approach	This is like a 'assign and forget' approach. In allocation, the ownership is with the new person working on the task, as it's a fire-and-forget approach.	Delegation is assigned, and then remember, you are part of the process all along the way and do check-ins, feedback, and the retro. This is more like a continuous process. With delegation, the person still has complete ownership and accountability on the task, even though it's being performed by a delegatee under their supervision.
Authority	Here, the authority is handed over to the new person, and they are accountable for completing the task.	Here, the person delegating the task will provide a set of authority to the delegatee that is essential for the completion of the task and will keep the rest with them as ultimately they are solely responsible for the task.
Long term opportunities	As part of allocation, we focus on getting the work done without necessarily	One of the outcomes of delegation is to help the person grow in the job and boost their morale and trust.

considering long-term growth opportunities.

Although delegation and allocation may seem similar, they have stark nuances. As we saw above in the table, how the task is expressed and executed to the other person changes the outcome and whether it will be short-term or long-standing. Now, in the next section, let's look at the differences between delegation and substitution.

### **5.2.2 Delegation versus Substitution**

With delegation, the person delegating still has full accountability for the task. They have a representative doing the work on their behalf, which will eventually help this individual grow in their role. You have provided them with the authority and autonomy to take the task to the finish line with a result-oriented mindset.

On the other hand, substitution is when someone has been entirely replaced and is no more accountable for the tasks at hand. Substitution should be considered in cases where in the future, you will not be picking up the task again.

Let's take a simple example- Imagine you, as manager, were out of the office (due to prior travel plans) for a good six weeks. To fill in your absence, you identified a senior engineer in the team to cover you for the day-to-day tasks of an EM. This individual had shown interest in being an EM in the long term, and so it perfectly fitted in the scenario. This is how you have delegated your duties, by providing them the support they would need in your absence, following the delegation framework you will learn below. Now let's contrast this with another scenario where you had spun up another team from scratch at another employer. The idea was to get the team up and running and then hire a manager against an open requisition you had to run the team long term (let's say you are already managing more than one team). This is where you were responsible for helping set up the team and hit the ground running, and once the team felt pretty stable, you can hire another manager to be the new manager for this team. This will allow you to slowly transition the team to them and fade out of the picture- substituting yourself with another person.

Another classic example of substitution is imagine a scenario where you have this high-performing engineer who is almost performing at the next level. You, as EM, are grooming them up for promotion. You start bringing this engineer with you in some cross-functional partner meetings, where they start by shadowing you, then speak on behalf of the team, and you reverse shadow them. Eventually, you fade out from the occurrence of this meeting and have substituted this person in place of you. Now that we understand the core differences between delegation versus allocation versus substitution, let's try answering the various questions about delegation in the next section.

## **5.3 The when, why, which, whom, and how of delegation**

Delegation is a vital skill for any engineering leader. But how does one go about delegation? Is asking someone to attend the meeting on my behalf considered a delegation? Probably not- unless you show them how it will help them in their career development and match their aspirations. Let's try to unriddle the various W's surrounding delegation and understand the right questions to answer at each stage to strive for effective delegation.

### **5.3.1 When and why to delegate?**

Now that we have learned at a high level what delegation stands for, let's dive deep into when we would need to practice delegation and why. Understanding that we **cannot do everything independently** requires a senior role or a leadership position. We must relinquish the tasks to make time for tasks that need our attention more than others. There can be various reasons or situations where one needs to delegate. Let's look into a few of them:

- 1. Available capacity and time-** We are always racing against time in this fast-paced world. In scenarios where there is a time crunch, approaching deadlines, and too many competing priorities at the same time, it is a good idea to consider delegation. By delegating tasks, we are freeing up our time to focus on other critical tasks that need immediate attention. This helps us achieve more and focus on strategic goals (team charter, tenets, Northstar strategy) to help the teams and the business. At the same time, it helps us maintain a work-life balance and not burnout by avoiding spreading ourselves thin. It helps with balancing out work with the mindset of divide and conquer. Remember that when we say work-life balance, we need to be very smart about delegation and ensure our team members enjoy work-life balance and have no signs of burnout. Also, as EMs, you are front and center of critical projects and people-related tasks. Remember to think about delegation if you feel you have too much on your plate and are stretched thin.
- 2. Grooming team members-** EMs prioritize their role as people-first managers, focused on the growth and upskilling of their team members. This involves identifying tasks that, when delegated, can contribute to their team's career advancement and create new opportunities. Even when we have the bandwidth to handle additional tasks ourselves, it can be beneficial to delegate them to team members who are willing and available. Additionally, setting stretch goals for team members can be valuable. By assigning them tasks that may initially be unfamiliar, but providing them with the necessary authority and support, we allow them to surprise us and balance the team's workload. Delegation is a teaching process that involves continuous support for the person receiving the delegated task. Identifying tasks that align with their career and growth aspirations is essential, allowing for a

personal touch and fostering commitment. This approach encourages individuals to become more invested and passionate about the work at hand.

3. **Have a subject matter expert take decisions** - There is an initiative to be led in the team/organization to decide on the throttling strategy for the backend services. You may have someone on the team who is an in-house expert on throttling strategies or an engineer passionate about operational excellence. This will allow them to create a team-wide or org-wide impact. They might have experience implementing throttling in their previous experience and might be subject matter experts with more knowledge than you. Reminder, you don't have to be the most intelligent person in the room.
4. **Delegation on behalf of others** - Delegation can occur on behalf of others too. For example, one of your engineers is out, so that you might delegate something assigned to them, to another engineer in the team, with some additional information and details. This also helps you calibrate if the team has any bottlenecks or single points of failure.

Some questions that can come in handy in this step of delegation are:

1. Am I working on tasks that are repeatable and easily transferable?
2. Do I have subject matter expertise in the team to make critical technical decisions?
3. Will off-loading a few of my tasks generate more opportunities for my engineers?
4. Am I focussing on some tasks that consume the majority of my time but are less critical and can be delegated?
5. Do I see signs of burnout for myself?
6. Does my team and I have a healthy work-life balance?

This is not an exhaustive list of situations when delegation can come in handy, but it is a good starting point. Delegation is a multi-step process that helps show the team members that we are vested in their growth. It provides a conducive environment for people to be themselves, not fear failure, and boosts trust and confidence in the team. Now let's learn how to choose the task(s) for delegation.

### 5.3.2 Which task to delegate?

It's crucial to accept that not every task can be delegated in delegation. For instance, high-criticality reports shared with C-suite and performance reviews should be handled by you, the EMs, and the leaders due to your close involvement and confidentiality. Sensitive or confidential HR matters should also remain within a closed group. Tasks approaching critical deadlines should be carefully considered before delegation to avoid burnout if the team is understaffed. However, tasks like progress reports on operational excellence can be delegated to on-call team members or an operational excellence ambassador.

Remember to choose the right tasks for delegation to set up the delegatee for success. You

can also break down tasks into smaller subtasks for better tracking and delegation and assign smaller tasks to new team members. In contrast, experienced members can handle more extensive tasks independently.

One time or the other, we end up with some mundane or repetitive tasks on our plates. Take an example of providing a quarter-end summary of the progress the team has made on the projects to the leadership. Now if such a report is templated, you can coach the senior engineer in the team to create the quarterly report for leadership. This is a repetitive coachable task that can help free up some time for you as EM and provide a career opportunity to your senior engineer. Evaluate tasks that are coachable to other team members which require little expertise. These tasks are transferable and can be taught to members using knowledge transfer sessions and demos to get some time back for you. Remember, just because the tasks are repeatable does not mean you fire and forget. You will maintain a level of supervision depending on the criticality of the task to ensure the doer feels supported and is moving in the right direction.

In my recent team, I delegated the task of setting up and running the operational excellence review meetings with the team to a senior engineer in the team. This helped free up some time for me to focus on other strategic initiatives for the team. At the same time, it provided the senior engineer more opportunities to run and facilitate meetings and learn at a high level about all critical tasks the team was working on from a tech debt perspective, which was one of their aspirations based on our 1 on 1 discussion.

Some questions that can come in handy in this step of delegation are:

1. Have I identified the right task to delegate?
2. Is this task repetitive in nature?
3. Are there other competing tasks on my to-do list that need more attention than the current task?
4. Is this task even possible to delegate?
5. Can you spare time to teach another employee the work?
6. What is the severity of the consequences in case the delegation goes wrong?
7. Is there a sensitive issue or requirement that won't allow you to delegate it to someone else?

We have now chosen the task to be delegated. What next? We need a person to execute the task. But how do we know the right person for this specific task? Let's look at some core factors to help us decide and find the right person for this work.

### **5.3.3 Whom to delegate?**

Once we have identified the task to delegate, the next step is to find the right individual (delegatee) to whom the task should be assigned. It's like passing the baton to the

appropriate team member who can complete the task. It is acceptable to identify multiple individuals to delegate tasks to, enabling a divide-and-conquer approach and ensuring tasks align with each person's abilities. The goal is to match the right people with the right jobs. It's essential to have solid reasoning behind selecting someone for a delegated task. Understanding their motivations, strengths, weaknesses, and sources of inspiration can help in this regard. Effective 1-on-1 meetings can be utilized to learn these aspects about team members. By delegating tasks that align with their learning and career aspirations, we demonstrate the long-term benefits, not merely passing on the task but creating opportunities for personal growth.

The distinction between "planned delegation" and "unplanned delegation." Planned delegation involves strategizing and collaborating with the individual ahead of time, leveraging tasks/projects as opportunities for growth. On the other hand, unplanned delegation occurs when tasks arise unexpectedly, which may only sometimes lead to effective delegation if rushed through. When deciding whom to delegate tasks to, it is essential to be fair and not always assign tasks to top-performing employees. The goal is to create equal opportunities for all team members.

In my early career as a senior software engineer, my manager knew about my aspiration to transition to a management role from an individual contributor position. He worked with me on a time-limited plan for this transition. During the summers, our company brought interns to work on projects under supervision for about three months. My manager delegated me the task of managing an intern for the summer. This involved setting up their project, serving as their coaching guide throughout the summer, and conducting weekly 1-on-1 meetings. Proper documentation of this process was crucial for the intern's end-of-internship debrief. This delegation aligned perfectly with my career goal of moving into management, providing me with an early opportunity to dip my toes into mentoring and managing someone. It also helped me gauge my interest in such responsibilities before officially transitioning.

Delegation can extend beyond your direct reporting chain or vertical. As an engineering manager, it's important to consider delegating tasks to individuals outside of your team's engineers. Business users, product managers, or members of cross-functional teams can be suitable candidates for delegation if the task aligns better with their roles. For instance, our team used to assign the responsibility of providing a biweekly progress report on various projects and programs to the project tech leads (engineers) within our team. However, I realized that this took away valuable time from engineers who were already providing updates during team standups and project meetings. This task would be better suited for someone managing at a program level without requiring deep engineering expertise.

In our organization, we had a technical program manager who attended our standups but primarily focused on consolidating reports at the organizational level. To address this, I

scheduled a 1-on-1 meeting with the program manager and proposed the idea of them taking ownership of the granular team-level and consolidated reports. This approach would allow them to stay closely connected to the team's products and facilitate easier creation of the consolidated report. Additionally, it would allow them to go above and beyond their regular duties, build relationships with engineers and EMs from various sub-teams, and strengthen their work.

We discussed and agreed upon a plan of execution for this delegation, clarifying how the process would work. This example demonstrates how delegation can be successful outside the immediate vertical of an engineering manager, allowing for effective task distribution and collaboration.

Now let's look at some questions that can come in handy in this stage of delegation:

1. Is this person equipped to handle the task at hand?
2. How will this person benefit from doing this task?
3. Is there another person on the team that is better suited for this task?
4. Who would like to take such a responsibility and enjoy learning?
5. Which person's career aspirations fit best with the tasks at hand?

With all this, we should be done and return to doing our own tasks. The person delegated the task should take it from here - NO. As the delegator, it is our job to create a win-win situation and provide all the desired support and resources to the individual to finish the task properly. Remember, this was one of the core differences between allocation and delegation, and we are learning to delegate effectively by providing the desired support.

#### **5.3.4 How to delegate and support?**

Now that we have identified the task ready for delegation and who to delegate to, let's shift gears as it's time for the actual delegation. The involvement of other team members plays a crucial role in determining the delegation's success. One key thing to remember here is that sharing the vision, reasoning, and context regarding why the task fits better if done by the chosen individual is paramount. It is here where we share how the task fits in the bigger goals of the organization, and team and helps this individual with their personal career aspirations. This will help bring a personal touch and motivate the individual to give their 100%. Also, this means we are not just delegating work for the sake of it, but rather delegating also to help other people learn and grow, causing a multiplier effect. This is your chance to show the person that you trust them with their abilities to be the rightful owner of executing the task. This helps boost morale and builds a more robust long-term relationship.

During delegation, ensure it is the right time and conditions. For example, let's say you have the right tasks to be delegated and the right person who is a subject matter expert in

the area, but the task takes 2 months to complete, and you have only one month from the deadline. In this case, even though what needs to be done and who it is being delegated to is figured out, some external factors and conditions can cause a failure in this delegation. This means that realistic expectations need to be set by either cutting down on some scope and communicating it to leadership or involving more individuals if there is an opportunity for parallelism. We should communicate the timelines and from when the delegation begins to avoid confusion and ambiguity. Remember, for any delegation to be successful, external factors should be accounted for at the start of the process, and with that, having a plan in place for delegation is the key to its success.

Also, remember, this step is like a sell/sale call(a call usually made by the manager to the potential employee to share the job prospects and team details and gauge the potential employee's interest, really a 'sale call') for the work to the employee so they are interested and committed to picking the work, so do your best to communicate the tasks effectively so they feel part of this decision-making process. This can be done in a 1 on 1 setting with the person, where you can also answer any questions the employee has for you.

One small exercise that can come in handy sometimes is to create a list of tasks to be delegated, then survey the people in the team in terms of who is interested in working on something particular, of course with the caveat that you will try your best to match people to their interests. This ensures that you keep an open communication channel with the team members, keeping their aspirations and choices into account and simultaneously increasing team engagement. Understanding the overall when, why, which, whom, and how of delegation helps engineering leaders make better-informed decisions when identifying the task(s) and individual(s) for delegation.

## 5.4 Framework for delegation

Now that we learned about answering the various W's of delegation, let's summarize what we learned above into a delegation framework that we can start using now.

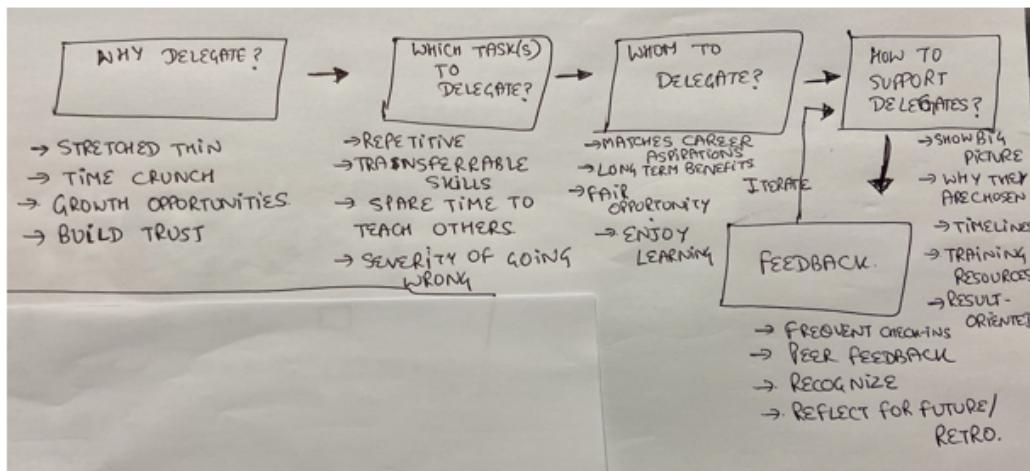
1. We first understand **why we should** delegate by understanding the situation and the opportunities it brings to the table for our engineers.
2. Next, we **identify the task(s) to delegate** and the timelines. This needs due diligence at our level to see if the tasks and skills are transferable if we have the time and energy to delegate, and what is the severity or risks if the delegation were to go wrong.
3. Now we are tasked to **identify the appropriate person(s)** who will be tasked to take the tasks to the finish line. This is where we ensure it matches the strengths and abilities of the person(s) and will provide long-term benefits to them, creating a win-

win situation for all parties involved. This is where you show them the bigger picture and how it will help them with their personal development and career goals.

#### 4. To achieve effective delegation:

- a. provide necessary support and communicate what needs to be accomplished rather than how.
- b. Clearly outline tasks, goals, and performance criteria to ensure clarity.
- c. Make yourself available for support and feedback, or assign a buddy if you're unavailable.
- d. Encourage confidence and provide tools and resources for success.
- e. Promote ownership and accountability for increased job satisfaction.
- f. Offer constructive feedback to acknowledge hard work and identify future opportunities.

**Figure 5.2 Framework for delegation of tasks. We start with why we should delegate, identify the task(s) to delegate, identify the person(s) to whom to delegate, and how to delegate and support them.**



This simple yet effective framework can help streamline the delegation process and keeps the team members engaged and motivated.

### 5.4.1 Supporting delegation

We need to share with the individual we are delegating tasks to how they will be supported in this journey. We want to create a win-win situation both ways.

Delegation is incomplete without providing the essential information and support to the team member to ensure we put them up for success in completing the task. Understanding the skill set of the team member, understanding the skill set needed to do the task, and

then addressing the gaps in the two leads to effective delegation. Key things to consider are shared below, but again given your specific situation and individual, they can vary.

1. **Communication** - Clear communication is vital for successful delegation. Convey the task, explain why the person is chosen, highlight benefits, and clarify your support as their manager. Maintain an open-door policy and foster a growth-oriented environment. Seek feedback by having them repeat and explain the task. Early feedback helps identify misunderstandings. I once faced a communication challenge when assigning an operational review document. I provided links and instructions but overlooked details on presenting metrics. It resulted in incomplete information and rework, reminding me of the importance of effective delegation.
2. **Mutual expectations and agreement** - When delegating tasks, clarifying the desired outcome and establishing agreed-upon timelines and processes is essential. This includes aligning on the task's start time, frequency of check-ins, coordination methods, key delivery milestones, and the final outcome. Communicate the task's priority, any issues, blockers, or risks, preferred communication channels and response times, and whether documentation is required for future reference and knowledge transfer. Transparency is crucial, so discuss these details upfront. Additionally, agree on contingency plans to avoid surprises if the task exceeds the established timelines. Emphasize the importance of documentation to ensure transparency and clarity for all parties involved.
3. **Training and resources** - Once you identify the gaps in skills in terms of what the individual needs to perform the task and what skills the individual chosen possesses, do a gap analysis and identify the resources to help. This can include pair programming sessions, knowledge transfer, demos, and learning resources across platforms like Youtube, LinkedIn, etc. But as a word of caution, depending on the task you delegate, if the knowledge/skill gap is too big for the other person to perform the task, you may be stretching them too much, and you are not setting them up for success. In this case, consider if they are the right fit for the task.
4. **Focus on outcome and results** - Avoid assuming prior knowledge and communicate the desired outcomes and results of the task. Focus on the "what" and "who" rather than the "how," avoiding micromanagement and a rigid approach. Strike a balance between hands-on and hands-off, granting ownership, accountability, and autonomy to foster innovation. For instance, when assigning an engineer to create an engineering metrics report, provide inputs and expectations without prescribing specific tools. Granting autonomy encourages fresh ideas and builds trust.
5. **Feedback/follow-ups mechanism** - Delegation is distinct from simple task allocation because it involves continuous follow-up and a commitment to ongoing learning. Feedback plays a crucial role in this process, allowing us to identify areas for improvement. As engineering managers, our focus during feedback should be on positive reinforcement and acknowledging what is going well. It is also an opportunity to share any positive feedback from peers or cross-functional partners,

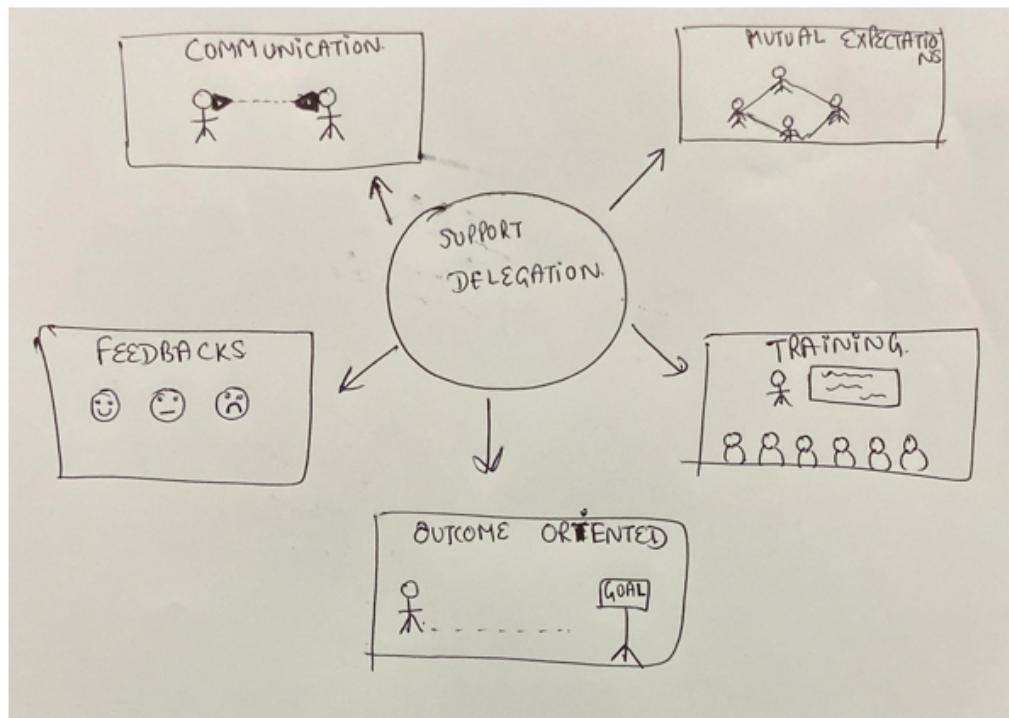
which can boost morale. Recognizing and rewarding the individual's hard work strengthens the relationship between the manager and the person(s) involved.

Additionally, it is a chance to reflect on the delegation process, learn from it, and identify future opportunities. Constructive feedback facilitates improvement and strengthens the delegation process within the team.

Mutually agree with the individual regarding the cadence and how progress will be shared to avoid one big-bang check-in and follow-ups to ensure checkpoint accountability. This way, we ensure that we are not slipping on timelines of task delivery and, at the same time, ensure the individual doing the delegated task feels supported throughout the process.

I would like to do a retro with the person to see if we have some common learnings for the future and how we can make the process smoother next time. As shared in the picture next, the core pieces of supporting delegation are- communication, setting mutual expectations and agreement, providing training and resources, focusing on outcomes and results, and providing a feedback mechanism.

**Figure 5.3 Supporting delegation is a critical part of the process. You want to provide the right resources to put the individual(s) up for success.**



Some questions that can come in handy in this step of delegation are:

1. Have you communicated the desired outcome of the delegation?
2. Have you delegated the right set of authority that is needed to get the job done?
3. Have you communicated the success criteria for the job?
4. Does my team member feel supported in doing the delegated task?
5. Do we follow up regularly on the mutually agreed-upon milestones?
6. Are we providing each other feedback to improve and get better?
7. Do I recognize the hard work of my team members?
8. What should we continue doing, what should we stop, and what should we start doing?

By answering the questions above, we get a clear picture of the execution of delegation and if we provide all the desired resources to put the person delegated up for success. Such execution and support can make or break the effectiveness of the delegation. In the next section, let's explore a framework to use for delegation purposes.

#### **5.4.2 Delegation tracker**

One quick tool I would like to share with you all is the delegation tracker. We spoke about delegation, but eventually, we want to scale. Today we manage a single team, 6 months from now, we will manage three teams.

How do we manage and track all our delegations and scale over time? Below is a very basic delegation tracker template that you can use to streamline delegation. You can use scalable tools like Jira or Azure DevOps for tracking delegation and the tasks to be done. Tracking will help keep you on top of delegation, especially when you have delegated multiple tasks to different sets of individuals. Also, such a tracker can ensure less confusion and help you be on top of all delegated tasks.

**Table 5.2 A sample delegation tracker**

Task description	Delegated to	Why delegated	Timelines	Latest Progress	Status	Notes/Comments/Feedback(s)
Preparation of an operational dashboard to measure incoming	Alice	Alice is the team's operational ambassador. She is passionate about OE,	In 2 weeks, ETA: Sep 11	Requirements have been gathered. Working on setting up the Looker dashboard	In Progress	Alice is working with Charlie for overall coordinating the template for the OE dashboard

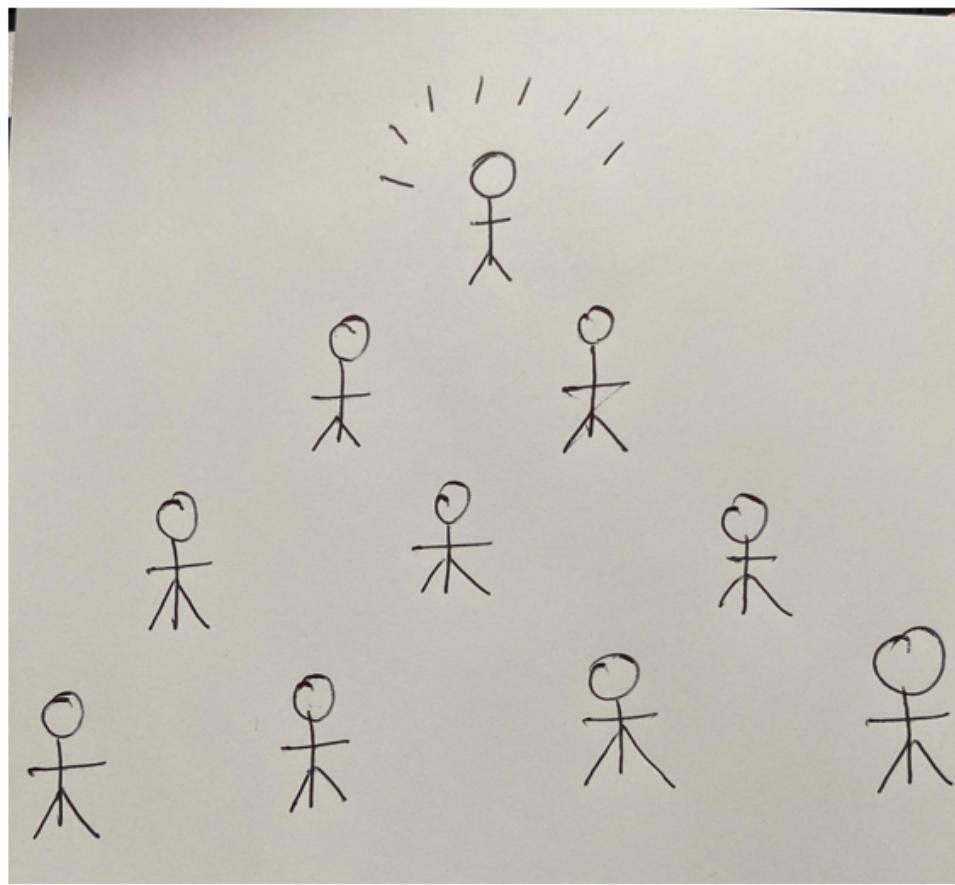
trouble tickets		one of her key focus areas.				
Onboarding new hires to the team	Bob	Bob's long-term goal is to transition to an EM role. He enjoys the people aspect of the role and would like to gain experience on this front.	2 months, ETA: Oct 30	Bob has filled out the onboarding template for the new hire. Next, he will be identifying an onboarding task for them.	In Progress	The joining date of the new hire is Sept 5.
...	...	...	...	...	...	...

This will help you plan your delegation and track them in an easy, seamless manner. Furthermore, sharing this tracker with team members will instill transparency and accountability. Now we have learned the core fundamentals of delegation to practice daily. In the next section, we will learn about creating a multiplier effect by sharing our experiences and teaching delegation to others.

## 5.5 Teaching delegation to others

Now that we have learned about the art of delegation, we know that as we grow in our role, we will have numerous instances where not only do we demonstrate delegation but instead have to train our team members to be better delegators themselves. Think about a staff engineer on your team who is stretched thin on all critical projects. They will need time to plan and hone their delegation skills to other senior engineers to stay motivated. This is where you come in as a leader who identifies such opportunities and now wears the hat of teaching delegation to others in your team. Also, since you have been in those shoes, sharing your experiences and stories will help comfort the person you want to train. This way, you are passing down your learnings and creating a multiplier effect by empowering others around you as shown in the figure.

**Figure 5.4 Teaching delegation to others creates a multiplier effect.**



Few things you can do as you teach others:

1. In a one-on-one setting, share why they would need delegation and how it will benefit them
2. Share your own personal experiences and learnings from your early days of learning to delegate for that organic relationship to build
3. Help identify and provide them the opportunity to delegate one of their tasks, to begin with
4. Share the resources that can come in handy and communicate your support in the process, especially as a sounding board in case they feel stuck or have blockers.
5. Keep a coaching mindset where instead of giving away how they should be doing it, challenge them to brainstorm ideas with you to navigate tricky situations.
6. Recognize and acknowledge the hard work and share constructive feedback to help each other.

I helped teach delegation to a staff software engineer on my team. Initially a senior software engineer, she was promoted to staff, increasing her project responsibilities. In

our recent 1-on-1 check-in, she expressed feeling overwhelmed and struggling to keep up with her to-do list. Upon further discussion, I realized she was attempting to handle everything she did as a senior engineer and her new staff responsibilities. This indicated an apparent struggle due to needing to utilize delegation. To assist her, I shared the benefits of delegation and recounted my own challenges when transitioning from an individual contributor to an engineering manager. We then identified specific scenarios where delegation could have been proper. One example was the weekly project meeting for a critical team project. With her calendar filled with new meetings as a staff engineer, she could have delegated the responsibility of running the project meeting to the next senior engineer. This would have allowed them to enhance their communication skills and gain visibility with cross-functional partners.

I also recommended helpful LinkedIn courses based on my experience and described how I envisioned the delegation process. I suggested she use a delegation tracker as she scaled her responsibilities. Two months later, she successfully embraced delegation, strategically choosing her battles and providing coaching to other engineers to gain experience and create opportunities for them. The above is a successful case of teaching delegation to others around us. But not everything will always go as expected. Along those lines, the delegation has chances to go wrong and fail due to ineffective delegation or external factors. In the next section, we will see some scenarios where delegation can go wrong and how to combat it.

## 5.6 Delegation can go wrong

Delegation is a super powerful management tool. Since it involves a certain level of risk, if not handled properly, it can go wrong and cause far more inappropriate consequences. Majorly it can directly impact the company and the business at stake. Also, if an incorrect person was identified for the delegated task, it is an improper use of their time. Also, the task will not lead to their growth or development, defeating the whole purpose of delegation. This means delegation can be rescinded if the situation warrants. This doesn't mean failure, but other approaches are sometimes needed. Some of the common reasons how delegation can go wrong are:

- Miscommunication: expectations weren't communicated, or the instructions were confused
- Lack of skills: despite taking on a task, they just didn't have the knowledge (probably technical) to do it themselves
- Not asking for help: when stuck, they didn't reach out for your help (or anyone's help) because they didn't want to let you down.
- External forces: someone changed the end goal on them, or the business reprioritized a task
- Etc.

Now let's look at a classic case where delegation has the potential to go wrong, let's say when you have a new junior engineer in the team they have been chosen to take up the delegated task. In this case, it's not the skills gap but rather the newness in the role that can need more help and hand-holding as the engineer tries to accomplish the task. In this situation, it is of prime importance to identify the fine line between how much guidance you, as EM, would like to give for the successful completion of the task. As EMs, it is on us to embrace mistakes and learn from the process instead of shying away from delegation. Don't fall into the trap of 'upward' delegation where you take the task back in control. Instead, this is a learning opportunity for the individual and hence work together with them for a stronger partnership. This is the time to not fixate only on the negatives but instead brainstorm ideas together to fix what has gone wrong and develop a path to the green. One can compare the art of delegation with playing chess. If you train and set the right pieces(team members), you are putting your team and yourself up for success.

In summary, delegation is an essential skill for any leader looking to expand their role and lead broader teams. Mastering this skill will help prevent burnout and make time for things that are higher priority and need your immediate attention. It helps build time for EMs to think big and take a 360 view of the team and projects. At the same time, to the delegated person, you show affirmation that you are vested in their growth and trust them. This instills confidence and accountability and motivates them. If you are new to delegation, start small - identify small tasks that are a good candidate for delegation. Slowly from them, you can move to more significant critical tasks for delegation. With such a shared responsibility mindset, you can tighten the culture, motivate, retain employees and be an effective delegator.

### **What do other leaders have to say about it:**

"Not only is delegating work crucial, but equally vital is providing the necessary support for individuals to accomplish their tasks. This approach should foster a learning experience, enabling them to grasp the fundamentals thoroughly. Consequently, if they need to repeat the task, they will have acquired the groundwork and be able to handle it independently next time."

**~ Sanjay Gupta, General Manager at HCL Technologies**

"Be lazy after you have delegated. That means trusting them to take their own calls and correct them only if needed. Let them own their choices and outcomes."

**~ Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

"For me, the biggest challenge for delegation is becoming comfortable with solutions different from how I would accomplish something. I practice this daily by asking my team to ask me questions like "Here is how I am thinking about doing XYZ" rather than

“How should I do this thing?” I then try to hold myself accountable by asking myself, “Will this have the intended outcome” instead of “is this how I would do it.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“The most important aspect of delegation is trust. You have to trust people to do the work that you don’t have time to do. At the minimum, you must trust that they will execute it well. In the best case, you trust they will do it better than you could because of their experience or perspective in the area.”

**~ Jean Bredeche, Head of Engineering at Patch, (ex) Robinhood, Quantopian, Hubspot**

“Delegation goes wrong when the person does not understand the request or is not the right person for the task. I delegated a task to one of my architects to manage a customer request. As the architect is technology focused and did not entirely understand the business impact, the output fell short of the business needs.”

**~ Rajakumar Sambasivam, Delivery Manager at Microsoft**

“There was a time when delegation went wrong. I had an Engineer back me up in an escalation. However, I didn’t spend enough time with this person to prepare for customers’ expectations and politics. The lack of preparation annoyed the customer, and we almost lost their trust by the end of the call, which took several more weeks of effort to restore.”

**~ Sumit Kumar, System Engineering Manager at Cisco**

“Delegating is the art of letting go. I consider it an art because it's part of trust, empowerment, and growth (for both parties). You're describing a "painting" for your delegate on what needs to be done, giving them permission to achieve it as they see fit, and trusting them to accomplish the task on your behalf. Sometimes the picture can surprise you and need touch-ups and maybe even repainting. Other times it can be a masterpiece showing that everyone worked together towards the same end vision. Over time, delegation becomes easier and easier, and the results more reliable and repeatable.”

**~ Bruce Bergman, Manager at Lytx**

## **5.7 Stop & Think: Practice questions**

1. Give an example of a time you had to delegate a task.
2. How did you go about choosing the right person for the task?

3. Give an example of a time when the delegation went wrong.
4. What would you do differently if we had a time machine and could reverse it?
5. Is there someone better in my team who is better suited to do a particular task than me?
6. Will this work occur again in the future, and maybe a good candidate for delegation?

## 5.8 Summary

- Effective delegation is an essential leadership skill that involves entrusting tasks and responsibilities to others to promote learning, growth, and development. It requires stepping into a coaching mindset rather than a doer mentality.
- When done well, delegation can create a win-win situation for both the leader and the team members, as it allows the leader to focus on high-level tasks and strategic goals while also providing team members opportunities to learn, grow, and take on new challenges.
- Delegation can be an effective way to build a strong, collaborative, and self-sufficient team and is an important part of effective leadership.
- There are several reasons why people might hesitate to delegate tasks and responsibilities.
  - Some might feel that they need to maintain control of every aspect of a project or task or that they are the only ones with the necessary knowledge or skills to complete it.
  - Others may simply not trust their team members to handle specific tasks or may be worried about relinquishing control.
  - However, these attitudes can ultimately hold a leader back and prevent them from scaling their work and achieving their full potential.
- Understanding the differences between delegation, allocation, and substitution can help you make informed decisions about handling tasks and responsibilities within your team best.
  - Delegation - Delegation is a process where we want the work to be done, but along with it we are looking for some learning skills that will continue beyond the immediate assigned task.
  - Allocation - Allocation is when we simply want the work to be done. That's it. It is more of an instruction-based model of telling others what to do.
  - Substitution is when someone has been entirely replaced and is no longer accountable for the tasks. Substitution should be considered in cases where in the future, the current person will not be picking up the task again and will be handed over to the next person.
- A delegation framework can provide a structure and guidelines for making informed decisions about task management and delegation. A delegation framework can also help you identify the tasks and responsibilities that are best suited for delegation and

can guide how to select the right delegatee(s) and communicate and support the delegation process effectively.

- A delegation tracker can be a valuable tool to help managers keep track of tasks that have been delegated and ensure that they are being completed effectively. It can also help managers monitor the progress of their team members and provide support and guidance as needed, which can help managers effectively scale their own work.
- Effective delegation involves finding tasks that align with individual team members' career growth and development and provides long-term benefits.
- The feedback mechanism is critical in the overall delegation process for us to introspect and iterate on feedback faster.
- Effective delegation is a crucial leadership skill with a multiplier effect on the team. Teaching others how to delegate fosters a culture of collaboration and trust, resulting in enhanced performance, productivity, and satisfaction.
- Delegation is a powerful tool for leaders to build self-sufficient teams and achieve goals efficiently. However, mishandling delegation can hinder its intended benefits. For instance, assigning a task to the wrong person can impact the company and impede the individual's growth. Similarly, ineffective delegation can cause confusion and frustration and hinder progress.
- As a manager or leader, embracing mistakes and learning from them is crucial instead of avoiding delegation. By continuously improving your approach to delegation through past experiences, you can empower your team and achieve success. Remember, even experienced leaders make mistakes. The key is to learn from them for growth and improvement.

# 6 Rewards and recognition

"Brains, like hearts, go where they are appreciated."

~ Robert McNamara, Fmr. American Secretary of Defense

## This chapter covers

- Importance of recognition for your team's development
- Various aspects of effective recognition
- Ways you can recognize and reward others
- Nuances between how recognition should be handled at different levels

Recognizing and rewarding employees is a critical part of an engineering leader's work profile. For a tech company, some of the most valuable resources a company has include the engineering talent pool that it possesses. Strong, engaged, and motivated engineers not only help you achieve faster development but also directly shape how effective future iterations of a product would take. Various studies have shown the need and importance of recognizing employees. For example, in [this](https://www.quantumworkplace.com/future-of-work/importance-of-employee-recognition) (<https://www.quantumworkplace.com/future-of-work/importance-of-employee-recognition>) study done by quantum workplace, recognition was identified as the top driver for increasing employee engagement.

Recognition helps employees see that the company values their work and helps them appreciate their contributions to overall success. It increases trust, improves communication as well as keeps the general morale of the team high. As the company grows and changes, effective recognition also helps the employees build a sense of security in their value to the company and keeps them motivated to do effective work.

As an engineering leader, creating a rich recognition-based culture is of great essence. There are several internal and external factors that could lead to employees not feeling appreciated. Effective recognition is a key tool to master and help reduce such scenarios. Effective recognition can help you

steer the team's ship in a positive direction. It helps increase retention, supports employee personal and professional development, and keeps an environment that is conducive to learning. This positive mood reflects in the company's positive culture and creates a conducive environment for growth. Recognition is often a fuel for motivation. A well-appreciated employee often has a higher sense of belonging towards the company and understands that you are vested in the growth. But before we delve too deep into how you can effectively recognize and reward others, it is important to understand distinct types of recognition, so let's get started.

## **6.1 Recognition levels**

Recognizing your team members, peers, and leadership can happen at all levels. The overall idea is to appreciate the efforts of others and send a positive affirmation message to them. Let's look at the three levels of recognition we commonly encounter in a professional setting.

### **6.1.1 Top-down recognition**

This is the recognition received from the leadership chain. Top-down recognition can be given by the immediate manager or someone above the manager. As is the case with all levels, this type of recognition can take various forms and can be delivered via a variety of mediums. Let's look at an example to understand top-down recognition further.

Ryan, a software engineer on Bob's team, recently worked tirelessly to solve a deep technical issue. This greatly simplified an important aspect of the team's pain point and helped reduce a lot of tech debt for the team. Bob, his manager, recognizes the effort Ryan has put into solving this issue by clearly articulating his work with the larger audience, including the senior leadership team. Bob sends an email highlighting what the team's pain point was, how Ryan went about solving it, and explained the engineering complexity surrounding this problem statement. At the end, Bob thanks Ryan for the stellar work he did and highlights the impact that this work has

and will generate. Bob here is demonstrating effective top-down recognition.

As highlighted earlier, top-down recognition can take various forms, which could include public or private recognition, and the mechanism you choose will be governed by a variety of factors which we discuss later in this chapter. What's important here is Bob recognizes the extra effort that Ryan made to solve a long-standing issue in the team. In the above instance, Ryan greatly appreciates Bob sharing his hard work with everyone. This motivates him to work on hard problems in the future as well.

Consider an alternative where Bob, as part of his general recognition process, aggregated achievements made by Ryan for a year and created a public post recognizing all of Ryan's great work on his work anniversary. This greatly motivates Ryan to engage in more challenging technical problems in the near future. At the same time, by only recognizing Ryan's work annually, Bob may have missed various opportunities of providing timely recognition. The annual post may provide some temporary motivation to Ryan however, it is hard to argue that the motivation would last for another year. Also, what if Ryan leaves thinking that his work is not getting recognized instead of waiting for Bob's end-of-the-year public post? Hence it might be important to recognize individuals promptly. These are two different examples of top-down recognition with completely different impacts.

### **6.1.2 Peer recognition**

This is recognition received from your peers as you work with them on different projects. Over the past many years, peer feedback has been an essential factor in evaluating the performance of an employee. Hearing from your boss and the management chain is, of course, necessary and essential, at the same time, having your work recognized by the people you work with is quite another thing.

Consider Ryan working on the same project as mentioned above; however, he also worked with another engineer in solving some critical parts of the

problem statement. After the project was launched, Ryan's peer, David, writes peer feedback to Bob appreciating the work Ryan has done. He details the effort Ryan has put in and explains why solving this problem was challenging but essential. On learning about the feedback received from David, Ryan is further motivated to work closely with different team members and collaborate better. This is an example of peer recognition.

Similar to top-down recognition, peer recognition can also take various shapes and forms. Peer recognition doesn't always have to come from immediate peers in the team. They can also be provided by different cross-functional engineering or non-engineering partners. In this case, the peer engineer recognized Ryan's work, if the recognition came from, let's say the product partner or business stakeholder, it will be considered peer recognition from non-engineering/cross-functional partners. At the same time, peer recognition doesn't always need to go through the manager. There are many mediums through which the recognition is directly delivered to the employee, while notifying others of the same can be an additional option. For example, some companies use tools where you can thank anyone for any piece of work that they have done. Whenever someone thanks anyone on the tool, both the employee and their manager would get the recognition notification. This is an example of where peers don't have to go through the management chain to provide feedback or deliver recognition, which comes directly from the peer. Peer recognition has an extremely powerful impact on how the team functions. A high-functioning team often has a culture rich in recognizing outstanding work, and this is where an engineering leader thrives.

### **6.1.3 360-degree recognition**

360-degree recognition combines top-down and peer recognition, similar to 360-degree feedback. It involves the employee's direct manager highlighting their contributions, showcasing appreciation from the leadership chain, and peer acknowledgment. This process gives employees an overall picture of how their work is appreciated. 360 feedback reinforces different aspects of feedback from various individuals, such as technical recognition from a peer engineer and acknowledgment of extra effort from

a product manager. It also allows employees to see how their work affects individuals in different roles within the company. Collecting feedback from peers, business partners, product partners, and clients offers diverse perspectives. However, for 360 feedback to be effective, it requires an open and honest environment without repercussions. In a toxic environment where underperformers are managed based on this feedback, it can have negative consequences. On the other hand, in a supportive environment where feedback is advisory, anonymous, and aimed at improvement and positive recognition, 360 feedback can be highly beneficial. Now that we understand the various recognition levels that can come into play, let's look at the fundamental recognition principles that form the foundation of effective reward and recognition.

## **6.2 Effective recognition principles**

Understanding effective recognition is critical to creating a recognition-rich culture. There are several things to consider when evaluating the process of recognition. Here are a few things to keep in mind when recognizing someone. These building blocks will help ace the idea of recognition and help promote a healthy work culture.

### **6.2.1 Timely**

Timing plays a significant role in the effectiveness of recognition. Generally, a small gesture of recognition delivered in a timely manner is more impactful than a more significant recognition that comes much later. Consider Ryan's example. Imagine a scenario where Ryan worked on his challenging project during the first quarter of the year and only got praise and feedback for it in the end-of-year evaluations. While Ryan would appreciate this during his yearly evaluation conversation with Bob, Bob missed a great chance to motivate Ryan throughout the year by effective recognition close to when Ryan worked on the project. Here, Ryan had to wait almost a year to feel appreciated, which is a significant missed opportunity. Well-timed recognition is one of the most critical things to

keep in mind and remind yourself to do when working on improving general team morale.

Let's look at another aspect of effective recognition which deals with time.

### **6.2.2 Frequent**

Another important and easily missed aspect of effective recognition is the frequency of the recognition. Frequent recognition keeps the team engaged. It is easy and natural to celebrate significant achievements. However, we should pay special care to celebrate small victories as well. Not all recognition needs to come from you. Creating a process for the team to recognize each other's achievements encourages employees to celebrate accomplishments together. Rewarding the recognition giver as positive reinforcement can also do wonders. As an EM, you need to build this culture of recognizing each other's achievement in your team if it doesn't exist already. At the very least, sit down with your team members and make sure you endorse frequent recognition, and everyone should feel free and comfortable to recognize each other.

Cultural shifts in teams take time. Small and simple processes can help jumpstart this process as well. For example, Jason, an EM, sets up a process to create a monthly internal post on an org-level group recognizing engineers who went that extra mile and have finished the highest number of test tasks for the month. While this is a small gesture of recognition, it helps motivate some new junior engineers to pick up more test tasks as a way of onboarding the team. This is a simple way to recognize the contributions of engineers going out of their way to fix the technical debt of the team. We are all guilty of focusing on big things and not appreciating small wins with equal enthusiasm at some point. Many small wins often lead to significant achievements. Being mindful and careful about this fact and celebrating any and all accomplishments is always essential. If you haven't recognized someone for a decent amount of time, think of various events when you can acknowledge their work. Here are some starter questions you can ask yourself.

- Any employee,
  - Recently finished onboarding?
  - Started a project?
  - Achieved a significant milestone in their project?
  - Delivered an important project?
- Is It Employee's
  - Feedback Cycle?
  - Work Anniversary?

While frequent recognitions are a great tool to keep handy, can you overdo it? Overdoing recognition can sometimes feel like handing out candy which does more damage than good by losing its importance. This leads us to the following principle of effective recognition- being specific.

### **6.2.3 Specific**

Don't just recognize someone only to complete a checkbox. Once you start adapting effective recognition, it is easy to get carried away and recognize only to recognize. This dilutes the value of recognizing an employee and loses its effectiveness over time. Employees can see right through this and lose any associated motivation. Specific recognition ensures the employee feels a sense of belonging to the team. Still, more importantly, it helps the employee understand which specific actions and contributions were made to the team and the company's goal. Being specific means understanding and acknowledging the details of the work you recognize, clearly articulating the impact that this work has generated, and why this work was necessary for the team. A simple "Thank you for the work you did" is often not sufficient. Unspecific feedback risks coming off as ambiguous and hollow. Instead of being specific, like 'Thank you for all your hard work on Project X. You moved fast to implement the categorizer engine even amidst strict deadlines. This categorizer engine will power the next generation of product categorizers and is anticipated to increase customer engagement by 1.1%. Also, your emphasis on load testing ensures we deliver a robust and resilient system to our customers, changing the tone and gravity of the recognition.'

It is also essential to focus on how you deliver the feedback and how you recognize contributions. Let's look at another critical aspect of effective recognition- visibility.

#### **6.2.4 Visible**

Thanking someone privately and appreciating their work publicly have two completely distinct purposes. Private recognition helps the person feel appreciated in a more personal setting. At the same time, public recognition has several advantages. The most obvious is magnifying the impact achieved by a team or an individual. Sharing the praise publicly also provides visibility into the great work the person has achieved.

Moreover, it also opens different opportunities for collaboration recognizing new solutions that might have been overlooked without broadcasting the information. This also sets the expectations and frameworks that other individuals can learn from. This keeps the team engaged and motivates everyone to promote valuable discussion. A simple real example through which I learned the effectiveness of visibility came from when I joined my first tech company as an entry-level engineer. During the second month, my manager wrote an excellent note highlighting the recent work of a senior engineer. He clearly articulated the problem statement, ambiguity, further unanticipated complexities, and the toil the engineer had to go through to solve that problem. My manager had called out the impact this work had generated and highlighted the great work this engineer had done. This helped me understand what great work looks like in the first place. Using this as an example of a benchmark, I was able to make significant improvements in my work in a relatively short period of time. Public recognitions are a great tool to keep in your runbook of recognitions, but do they always make sense? There are various scenarios where keeping the recognitions private is preferred. We will discuss the details of this in a later section.

#### **6.2.5 Fair**

Every leader should strive to give employees a sense of belonging to the team's culture and values. This should be one of the highest priorities for the leader. It is crucial to create an environment that encourages a culture of prosperity and integrity. To achieve this, it is essential to recognize the conscious and unconscious biases in all of us. This bias often leads us to make judgment calls that may not be entirely correct. Acknowledging the existence of this bias is the first step in reducing it. After that, several strategies can be used to reduce that bias.

- **Use Data:** Wherever possible, always use data to help guide you in judgment. A clear, well-defined criteria for recognition is often an easy place to start. This avoids the risk of favoritism and promotes visibility and transparency. Another essential thing that data helps us identify is seeing past the flash. Not all people are alike. Many great employees may be great at one aspect of their work but struggle with others. You may have one or many introvert engineers on your team. Alternatively, someone in your team may be working on projects that reduce technical debt. This may not be the most flashy work, but it is one of an engineering organization's most critical works. It is also essential to see past the flash and ensure their work has been recognized. Using data can give you critical insights into what is valuable or not. For example, reducing tech debt by delivering project X reduced the number of bugs by 8%, helping save 3 engineering months' worth of time.
- **Use everyone's help to recognize contributions:** There are several ways to do this. It could be done via employee surveys or peer awards via nominations, or any other way that includes everyone in the decision-making process. This process also helps bring inclusivity to the team. Having a peer reaffirm your great work can also be highly motivating.
- **Retrospectively look for unconscious bias:** Unconscious bias is hard to identify proactively. Using the above two steps can help reduce this bias but doesn't eliminate it completely. What has worked for me personally is to look for unconscious bias retrospectively as well. Usually half yearly, one can try to look at all the recognition that they have delivered in that half and try to find the existence of some

unconscious bias that may cloud their judgment. This usually helps adapt for the coming half and reduce those biases.

While the above presents a few ways to reduce bias and increase fairness in how you recognize and reward others, no one way works for every scenario. In fact, it is more important to be cognizant of your biases than to focus on debating which way you should choose to reduce that bias. If you are aware of your shortcomings, it's half the job done. In my case, for example, I was almost oblivious to the unconscious bias that I had when I first started my work in the tech industry. Once I was aware of this, I applied various ways in which I could reduce that bias, but that big jump in my thought process was solely driven by my realization of the existence of such biases. With that in mind, are you curious to learn about ways how you can start recognizing the talent around you? In the next section, let's look at various ways we can appreciate the work around us.

## 6.3 Ways to Recognize

Recognition can take various shapes and forms. The first thing to identify here is whether you recognize an individual versus recognizing a group. Either way, there are several things to keep in mind. Let's start looking into some ways you can recognize the achievements of a group of people first.

### 6.3.1 Group Recognition

Group recognitions are a way to celebrate various milestones as a team. Incorporating the entire group and involving as many people as possible to celebrate an occasion is an immense morale booster for your team. Not only does it motivate the groups of employees whose achievement is being celebrated, but it also helps bring the team closer, ensures that all small and significant achievements are celebrated as a team, and, most importantly, it's a great way to blow off steam and reduce stress on a team that is working hard towards the end goal. Let's look at a few scenarios you can leverage as good starting points for this effort.

## **Celebrating project deliverables/launch parties**

Celebrating various project deliverables, be it significant milestone completion or launch of the minimal viable product(MVP) for the project the team is working on, is a great place to start. Usually, most projects require a lot of effort from various individuals, internal and external to the team. This is even more true for larger initiatives that require collaboration with several cross-teams. The team often works really hard to make sure that all the scenarios for the project are well covered and ready to launch well before the delivery date. There may be many opportunities for you to thank each individual separately for the contributions they have made to the project, however celebrating the project delivery together ensures that employees reaffirm the fact that the company values their contributions greatly.

Launch parties don't need extravagant affairs where you must spend a lot of money. In fact, more often than you would like, you would be limited by the budget that's allocated for spending on these kinds of events. If you cannot organize a team dinner, focus on getting the working team together in a room and use that opportunity to celebrate this achievement. If you have a small budget, you can organize snacks, food, or drinks in a small meeting room instead of planning a full-blown party. It is the gesture of acknowledgment of the work that your team has done recently, which is far more important than how you actually organize the event. Regardless of how you organize the launch party based on the budget at your disposal, it is always a good idea to include all the project collaborators in the event, including cross-functional engineering and business partners.

## **Team offsites**

Team offsites are another great way of celebrating project wins. More often than not, you usually have some budget every quarter/half/year to spend on team offsite events. These offsite events allow employees to take a break from their everyday work and spend time together in a non-professional setting. Offsite events work great in keeping the team engaged and happy

and, most importantly, prevent the team from burnout. However, you can use these events for much more. Before every offsite reflecting and introspecting about the work your team has done since your previous offsite is another great way to think of various recognition misses you may have had recently. Creating and managing a forum where every project achievement is highlighted and acknowledging the great work your team has done can also be a great motivator for your team. Leveraging this time gives you great flexibility in how you want to organize these offsites. One simple way to use this time is to start the offsite with some thank-you notes for the different sub-teams that have worked on different projects in recent times. This is probably too simple an example of leveraging this time, but you should get the idea. Let your imagination dictate how you want to utilize this time for recognizing and rewarding others, but most importantly, keep this time at the back of your pocket to evaluate how you have been recognizing others recently.

Another thing to remember when organizing these offsite events is to be as inclusive as possible. This means accommodating such events with employees who may not be able to attend after-hour offsites because of personal commitments or employees who may be remote and might not have the bandwidth to travel for the offsite. We will discuss the details of this in a later section.

## **Project/Team Swags**

Team swag, or Stuff We All Get (SWAG), is a common term for gifts like team-branded t-shirts, water bottles, and other items that instill a sense of pride and connection among team members. Utilizing team or project swag is an excellent way to celebrate recent accomplishments and recognize the team's hard work. While budget constraints may apply, it's important to seize opportunities to leverage such events for recognition and rewards.

For instance, I recently worked on a year-long initiative involving seven engineering teams collaborating on a new framework. Due to remote and hybrid work arrangements caused by the pandemic, we formed a 50-member virtual team to deliver the project, consisting of engineers from

various teams. Additionally, we collaborated with cross-functional partners, including product managers, technical program managers, UX designers, and data scientists. As this virtual team faced challenges in connecting outside of a professional setting (such as different locations and lack of funding for offsite activities), we decided to create project-specific swag items and distribute them to everyone involved. This gesture was a meaningful way to acknowledge their contributions and highlight the team's significant achievements.

Another effective example of project-related swag is the concept of challenge coins. While unsuitable for every situation, challenge coins work particularly well for large projects and significant endeavors. These inexpensive personalized tokens can be presented to team members as different phases of the project are completed, similar to how the military rewards members for successfully accomplishing missions together. The challenge coin serves as a small but memorable token of appreciation. Employees can proudly display these coins at their desks, sparking conversations when others inquire about their significance. This provides an opportunity for the recipient to share their achievements and contributions. Alternatively, other forms of memorabilia, such as customized thank-you cards, can also serve a similar purpose. While these specific approaches worked great for us, there are several challenges with recognition when you have a hybrid/remote team. Let's discuss adapting your strategies to ensure you don't leave your remote workers hanging.

## **Hybrid/Remote Work**

With the recent changes in the tech industry, companies have embraced hybrid/remote work with open arms. Depending on which company you are a part of, you are more likely to manage employees who have hybrid/remote work roles. While these roles give great flexibility to many employees, they also present various challenges for you as an EM. Recently many studies have pointed out that for some employees, it may be harder to connect with the team if they are in a remote role. That doesn't mean that remote roles are going to go away. In my opinion, the future holds more

flexibility for employees where you might have to accommodate different work profiles based on employees' personal situations. Some ideas translate well to managing employees in all scenarios. For example, in our case, since the team was split between 4 different locations and we ended up sending project swags to everyone. However, conducting team offsites, or in our case, a launch party, was not so straightforward.

To ensure remote employees are not left out when celebrating project launches or conducting team offsites, it's important to plan remote-friendly events. With the rise of virtual options due to the COVID-19 pandemic, there are numerous opportunities to engage remote employees. Virtual dinners, offsites, and events like escape rooms, terrarium-building workshops, cocktail-mixing classes, and treasure hunts can be organized through companies specializing in virtual event management. Even a simple remote lunch with fun games can be effective within budget constraints. These activities provide various options to connect and create a sense of belonging for remote workers. For example, my team recently organized a virtual Halloween party. We hired an external company to host the event, and employees participated by dressing up in their homes. It was a delightful way to celebrate the festival while embracing the remote setting. It's important to remember that these events should also acknowledge the achievements of remote employees. Highlighting their contributions and ensuring their inclusion is crucial to fostering a sense of belonging.

In one of my recent experiences, we organized a virtual launch party for our team. The celebration began with everyone receiving DoorDash coupons to order food and drinks. This was followed by a presentation highlighting the team's remarkable work on the project. To promote engagement, we divided the team into small groups and organized a short treasure hunt game with the help of an external provider. This approach worked well for us as it prioritized inclusivity and ensured everyone felt involved. Promoting inclusivity and recognizing the accomplishments of the entire team, including remote employees, should be a fundamental goal for every manager. Above are all examples of how we can recognize people in a group setting; now let's look at how we keep up with appreciating talent at an individual level.

### **6.3.2 Individual recognition**

As we saw above, group recognitions are a great way to motivate the team. However, solely relying on them often will have negative consequences for your team. While group recognitions do a great job of recognizing the team or a group of individuals as a whole, it doesn't provide the personal level of recognition that can fuel an employee's fire to another level. Recognizing the exact contributions of an individual, even if it is part of a bigger project, provides direct feedback to the employee. This process leaves no room for ambiguity and also helps manage anyone struggling with imposter syndrome at that moment. Each project may be worked upon by different types of individuals. Some may be confident in understanding their role and how their contributions shaped the project, others may need someone to validate the importance of their work. As an EM, it is important to be mindful of these differences and focus on providing both direct feedback and direct recognition to employees whenever such an opportunity presents itself. Recognition earned by an employee should also be kept and tracked to be used later for performance review time. This serves as a great reminder of their work and how it was appreciated by different individuals around the employee, including you.

There are two distinct platforms to recognize individuals. Before diving deeper into this one common theme throughout this chapter is to make sure you understand that there is no one-size fits all solution that works for everyone. Some employees may prefer to have their recognition private, whereas others may prefer a combination of public and private recognition. There may be several reasons employees want to keep things private, being introverted is just one of them. One good practice to keep handy is always to ask your employees how they would like their work to be recognized in the early stages of their joining your team. This will almost always help you cater recognition for different individuals in different ways, making the recognition more meaningful and more relatable. Having said that, let's dive deeper into each one of them.

#### **Public recognition**

Public recognition is a great way to recognize an employee's achievements and highlight their contributions. Spreading the word about the hard work someone has been doing is a great way to motivate them to succeed even more in the future. There are multi-faceted benefits of public recognition. Obviously, it has a positive impact on the person you are recognizing. At the same time, this process brings awareness and knowledge to others around you both about the qualities of the individual as well as the substance of the work they have delivered.

Public recognitions are a great way for new employees to be known at a broader level. This provides them with opportunities to connect and network with individuals they haven't had a chance to work with yet. At the same time, more avenues of collaborative work open up because of this process. As an example, an employee in my team recently built a testing infrastructure that worked very well to suit our business needs. However, when we publicized that work, we received many comments about how this work can be greatly beneficial to other teams as well. This opened up a lot of avenues for collaboration and expansion of the project. Not only did the engineer create a testing framework to solve our team's needs, many parts of that system were adapted by multiple teams with a lot of guidance from the engineer that worked on it. This situation was a win-win scenario for every party involved, starting from the engineer to our team to the org and the company overall.

Using internal communication channels is a great way to recognize others. One of the advantages of using such a medium is you can control the amount of transparency and visibility you would like to provide for different sets of audiences you may have. A simple Slack shoutout on the team group may help bring awareness of certain projects to everyone on the team. Several companies use tools like [Lattice](#), [Applauz](#), [Nectar](#), [Kudos](#), etc., or in-house made tools to help have a formal way of recognizing contributions. However, you may want to reword this and highlight the technical complexities that were required to solve the problem statement on a different Slack group which is more catered toward engineers. Your leaders may prefer a different kind of update that talks about how solving this project enables others in the company to succeed, and at the same time,

your product partners may want to know how solving this project helped the company achieve certain business goals. This kind of flexibility gives you great control over how you are messaging the achievement for the different types of audiences. There are many other forums you can leverage to recognize someone's work- team meetings, retrospective sessions, offsite and non-professional events, to name a few. If your company uses a portal or a dashboard like Mango, Sharepoint, Teams, Workplace, etc., that's also a great place to highlight public recognition.

But if public recognition is so good, why not always do it? Are there any scenarios when public recognition can be detrimental or cause more harm than good? Remember, when you publicly acknowledge someone's contributions, one of the first questions other engineers will ask themselves is, "Why that person and not me?". This is a fair and valid question which sometimes can indicate your misses on recognizing others and other times can bubble up different insecurities people may have with respect to their job. Another reason why public recognition may not be super helpful is if the employee themselves don't feel comfortable with public recognition. As you work with various individuals, you may find some folks anxious in public recognition settings. It is important to identify such scenarios and deal with them appropriately. Remember, the goal of the recognition process is to keep the employees motivated ultimately. There is almost never a one size fits all approach that works well in real life.

Therefore, even though public recognitions are a great tool to use, overusing them or abusing them is bound to get you in some trouble sooner than later. Public recognition should be used judiciously, and the content of the recognition needs to be well thought out and clearly articulate someone's achievement leaving no room for ambiguity or misinterpretation.

Another way to alleviate the above concern is to circle back to one of the guiding principles of effective recognition, which is being "Fair." You can leverage many team members to decide on whose contributions should be recognized at a certain moment. For example, instead of you recognizing someone's work yourself, use a survey or a nomination process to figure out who has done outstanding work in the recent past. As an EM, it is

incredibly hard to have a well-rounded perspective of someone without using peer feedback. Recognitions can follow the same train of thought. If the person that nominated another is comfortable, let them recognize the work their nominee has done publicly. For example, keep 15 minutes in your quarterly team meeting for nominators to present about their nominee's work. That way, the nominee also knows who values their work the most, and more importantly, they understand that other team members also acknowledge the hard work they have put in recently.

## **Private recognition**

Private recognitions are a great way to boost your employee's morale and to keep them engaged. Unlike public recognition, private recognitions are delivered in a private setting where you recognize the work someone has been doing personally. With private recognition, you don't have to worry about managing how others feel, but at the same time, you should be extra careful in making sure your judgments and evaluations are fair and accurate. Private recognitions are a powerful tool in your arsenal of things to do when bringing the team's morale up. Firstly, hearing the feedback from you personally reaffirms the employee's faith that their work is not in vain. Secondly, it also ensures that employees know that you are vested in their growth by acknowledging the fact that the employee has made significant strides in their pursuit of growth. This process of continuous feedback and continuous recognition often results in continuous growth regardless of whether the employee is already high performing or not. As we mentioned earlier, tracking the recognitions received by an employee over the course of the year and re-recognizing those efforts during a performance review is of great utility when delivering performance feedback and directly contributes to the professional growth of the employee. Listing these recognitions during promotions review helps highlight why the employee may be ready for promotion and how their work has been perceived by others.

One key thing to note here is public and private recognition shouldn't be used exclusively in different scenarios. In my experience, most scenarios

can have either or both public and private recognition together, and this works really well for employees who prefer a combination of both these approaches. While there are many ways in which you can reward others, let's look at a few of them to understand how you can go about doing this.

## **Monetary rewards**

Monetary rewards can take various shapes and sizes. Regardless, this is one of the best ways to reward an employee. An increase in compensation, spot bonuses, and different award programs are a few ways to utilize this approach. Of course, you are limited by the budget and processes that surround monetary rewards. However, making sure you are aware of how you can create more budget for some high-performing individuals in your team gives you a great advantage in succeeding in the process of recognizing and rewarding others.

Some companies use external tools to give monetary rewards. For example, there are many external providers which can gamify the entire system of rewards. Partnering with some of them can also give you great flexibility. Usually, these companies have a “point” system where points are allocated based on the impact of the contribution that is made by the employee. These points can then be used to redeem cool company swags, get gift cards, or general gift items. Suppose you want to conduct a test-a-thon event geared towards your team writing lots of tests to increase line and branch coverage for legacy code that had missing test coverage, that occurs for 6 months. Every time someone writes a new test, they get certain points which can eventually be redeemed for swag, food, or something cooler. Gamifying this entire process often brings more engagement and better results for you. It also incentivizes many employees to take part in a process that they may find uninteresting otherwise.

Alternatively, you can also explore if the company has any partnerships with learning providers like Pluralsight, Coursera, etc. Subscriptions to these as part of the reward are another great way to both reward and employee and set them up for success in the future.

## **Non-monetary rewards**

Non-monetary will most certainly be your go-to in most scenarios. Non-monetary rewards remove several bottlenecks which monetary rewards have. You don't need to worry about budgeting, approvals, or any other process to get things going. You can also crowdsource recognition easily when dealing with non-monetary rewards.

One simple way to recognize and reward someone privately is to use forums like 1-1s to provide clear and succinct feedback. Acknowledging someone as an expert or the subject matter expert(SME) for a particular area, giving them more responsibility and bigger work if they are ready to, and opportunities to represent the manager or the team in different meetings are other ways to reward the employees further. The reward doesn't always have to be materially defined. A person looking to get promoted in the next year or the next few quarters will often find more joy in getting affirmation that they are growing in the role they are in instead of getting a \$25 Amazon gift card for some work they just did.

Leveraging your leadership chain to recognize the work someone has done recently is another approach that can be taken. Send an email recognizing someone's work and cc your manager or your org leader and the employee's manager and org leader. This will not only help the employee feel their work is clearly visible to all the leaders but will also make them more confident the next time they talk to these leaders. Bridging this gap between your employees and your leadership chain is an important skill to develop for an EM, and this can be a great approach to solving part of this problem. Regardless of which approach you choose, ensuring that you are generous with the praise that you give to others is an absolute baseline. Being clear and concise doesn't mean underselling someone's achievement. Ensuring you have covered all aspects of the work someone did will surely set you up on the right track for an EM.

## **What do other leaders have to say about it:**

"Take the time to comprehend what drives the individual, as this will reveal the most effective ways to recognize their efforts and facilitate their career growth and overall happiness."

**~ Sanjay Gupta, General Manager at HCL Technologies**

"Few ways I ensure people feel valued are 1) publicly celebrating wins, 2) making sure everyone is empowered to speak up since good ideas come from everywhere, 3) making sure that if someone needs something, they get it - e.g., if in a 1:1 someone asks me for something, I make sure I do it, to build trust."

**~ Jean Bredeche, Head of engineering at Patch, (ex) Robinhood, Quantopian, Hubspot**

"Feeling valued is not just determined by their rewards but by their day to experience with the company and team. As a manager, I try to make sure that they have meaningful work which aligns with their skill growth."

**~ Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

"Value and recognition pairs well with work visibility. Once a team's work is visible across the organization and we have cross-company alignment to deliver the most important work, connecting that to individual recognition becomes much easier."

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

"Getting rewarded or recognized for your efforts is one of the most common needs for all individuals. I believe in regularly identifying, recognizing, and rewarding high performance/contribution in my team. Whether it is a simple GOOD job done in morning meetings to relying on "employee of the month" programs to ensure the rewards like merit increases or promotions to the most deserving team members. I also strongly believe that building an environment within the team where each

team member feels comfortable in bringing their best and actively contributing to each other's success is the key to overall success because in our world, teamwork would always triumph over individual brilliance and hence I constantly reward and recognize the overall team along with the individual performance.”

**~ Devika Ahuja, Technology Leader | Strategist**

“Recognition and rewards help to keep people motivated to do their jobs. I am all for recognizing people who push the boundaries, are dedicated, and show a level of commitment to stay above the curve. It’s pertinent for the manager to award people who are doing their best and help achieve outcomes. One important aspect of the teams I have seen succeed is those with the right culture. Teams must build trust to complement their skills and rely on each other. Ensuring the right environment for open collaboration, inclusive workspace, and celebrating the success of everyone have gone a long way to have a great team.”

**~ Rajakumar Sambasivam, Delivery Manager at Microsoft**

“We do not provide enough recognition for people that they deserve. So I intentionally create environments where people get recognized by their colleagues and leaders. A key component of good recognition is to add the evidence of good behavior otherwise, it looks phony. I always highlight the behavior as evidence of why a person deserves the recognition.”

**~ Sumit Kumar, System Engineering manager at Cisco**

“Along with pay, give your team meaningful work so you know that their ideas are implemented, their skills are appreciated, and they have a say in the product they're building. This ensures people are aligned in what they're trying to build.”

**~ Larry Gordan, Managing Director at Emtec Inc.**

## **6.4 Stop & Think: Practice questions**

1. Do I value recognition in my team?
2. Does my team/company have a culture of recognizing and rewarding employees? If not, can I build it within my own team?
3. When was the last time I recognized someone on my team? How did they take it?
4. How often are your team members recognized for their work?
5. Do I feel valued when I am recognized for my efforts?
6. What type of recognition do you value the most?
7. How do my team members, individually, like to be recognized and rewarded?
8. Do you feel the employees are treated in a fair way when rewarded? Why or why not?
9. Do you have recommendations for leadership to introduce ways to recognize the good work around you?

## 6.5 Summary

- Recognizing the work of others is one of the most crucial parts of the work you would do as an Engineering Manager
- Practicing the tenets of effective recognition will make you better at realizing whether you are doing a good job in recognizing others.  
Effective recognition should be
  - Timely: Recognized at the appropriate time. Very early recognition can create unnecessary pressure on your employee. Late recognition devalues the work that someone did and loses its punch.
  - Frequent: Celebrating small wins in addition to big wins is the key here. Ensuring your recognition is frequent will ensure that you don't miss out on some opportunity and by the time you realize it's already late
  - Specific: Avoiding generic recognition is key to effective recognition. Being specific, clear, and concise ensures employees feel the recognition is genuine and not done just to check off a box.

- Visible: Recognitions should be visible as much as possible. Leveraging effective public recognition is the key here. Public recognitions help you spread the great work someone has done and can also help you create more opportunities for the individual later on.
  - Fair: Managing personal bias and rejecting favoritism is the key here. Involve as many people as you can in part of the decision-making process. This way you are never tunnel-visioned, and you always get a diverse set of perspectives before recognizing others.
- There are several ways to recognize employees. Before starting this process, though, understand the difference between how you would recognize a group of individuals vs one person.
  - For group recognitions, if you have the budget, utilize different milestones in the project to celebrate with your team and highlight the work they have done up until now. Post-project delivery and launch parties are a great way to recognize others.
  - Leverage team offsites also to create another platform for recognizing teamwork.
  - Keep your recognitions inclusive and account for hybrid or remote employees you may have in your team.
  - For individual recognition, use public recognition as a key resource to spread the work they have done.
  - Utilize public recognition judiciously and avoid making others in the team uncomfortable or unhappy.
  - For more private recognition, provide clear and succinct feedback to the employee.
  - If you have the budget consider rewarding your high performers with monetary rewards
  - Acknowledging individuals as “SMEs” or “go-to” persons for certain areas of ownership or representing you or the team in different meetings is another way to recognize the work that employee has been doing
  - Regardless of your situation, scenario, and approach- always be generous with your praise

# 7 Hiring

“It doesn’t make sense to hire smart people and then tell them what to do; we hire smart people so they can tell us what to do.”

~ **Steve Jobs (Co-founder, Apple)**

## This chapter covers

- The hiring mindset
- Hiring framework to use while planning for hiring
- Differences between staffing from scratch versus hiring for an existing team
- Differences between hiring externally versus growing internal people
- Building a healthy hiring pipeline
- Introduce various hiring programs to send positive reinforcement messages to employees

You have a product, and you want to spearhead its development. You need a strong team consisting of engineers, managers, and cross-functional partners at the very least. Effective and strategic hiring will play a critical role in how you shape your team and how effective other teams at the company will be. There are many scenarios when you would need to hire people. One key reason is to support the growth of the company. As the company grows

- It plans to start one or many new initiatives which require additional headcount
- Teams require additional support to maintain existing initiatives
- Backfill existing headcount because of natural attrition

Hiring is more challenging than it sounds. It involves perseverance and can often lead to dead ends in a competitive market. But staying motivated and keeping the fundamentals of your hiring strategy in mind is the key to successful hiring. Now, what exactly does hiring strategy mean? It is a set

of principles you follow to understand the general growth direction of the company, the allocated hiring budget, the estimated effort to launch and support planned initiatives, and the actual hiring process. The details of key things to remember are covered in the sections below.

One crucial aspect to focus on is that hiring is significantly costlier than retaining talent. [EBN](https://www.benefitnews.com/news/avoidable-turnover-costing-employers-big) (<https://www.benefitnews.com/news/avoidable-turnover-costing-employers-big>) estimates hiring a replacement takes roughly 33% of an employee's annual salary. Therefore, in most scenarios, hiring an employee only for short-term needs or someone who is questionable or may jump ship later for a better offer is counter-productive. Retention is well worth the effort and investment, and part of retention is hiring for the long haul. Therefore, it is essential for everyone involved in the hiring pipeline to contribute to making it more efficient over time.

A key mindset change that can help today is not to treat hiring as episodic but rather more of a continuous iterative process. This means that don't just focus on a company's near-term needs and open headcounts, instead think about hiring as a continuous planning process at all levels (leadership, recruiting, and managers) where you are always talking and interviewing. This can mean sharing about your team and company in tech forums, meetups, conferences, etc. The idea is to be a forward thinker and recognize what a company might need six months from now or one year from now. Hiring is an opportunity where not only you interview the candidates, but you also give the interviewee a chance to interview you to learn more about the company, its culture, management, the expectations of the role, and much more. This plays a significant part in the interviewees deciding whether the position would be a good mutual fit. Hence it is important to bring your authentic self and treat each candidate as a potential customer of your company's product. In the next section, let's look at a hiring framework to use as a handy tool as you plan for your next hires and the team.

## 7.1 Hiring Framework

Hiring is an integral part of the success of any business and organization. This process needs to be fair and unbiased to avoid any bad hiring decisions. Let's explore below a step-by-step hiring framework that begins all the way from identifying and sourcing the right candidate to bringing them on board with an onboarding plan in place. So let's get started!

### **7.1.1 Identifying the need for hiring**

The most obvious first step in the hiring process is to identify the need for hiring. Taking a look at the current and future needs of the company and the organization to determine who, what, when, and why you are hiring is of critical importance. This process requires a lot of research, budgeting, and planning to get a holistic view of what your hiring goals should be. For any growing company, it is important to understand which avenues it would be investing in and how it will affect the staffing requirements for those areas. Once there is clarity on the need for hiring then the actual hiring pipeline naturally kicks in.

### **7.1.2 Sourcing potential candidates**

Sourcing the right candidates is the next step in the hiring process. In this phase, we are trying to understand what type of candidates will be a mutual fit for the company and the individuals themselves, what skills and experience the individual should possess, and what kind of roles or companies this person might have worked at. It is common for individuals to transition between companies to achieve improved work-life balance and benefits, often resulting in candidate poaching from one company to another. In this phase, I suggest you consider the below tenets as you go about sourcing candidates.

#### **Tap talent from diverse groups**

A diverse workforce is a boon and asset to the company, where we value the differences and help keep an open mind. This allows for greater creativity and innovation and helps create a better work culture where

people thrive. As you begin the process of sourcing candidates, ensure you are trying to tap into the talent pool and reach out to diverse groups to target an overall audience. Some good opportunities can be promoting your company and roles in conferences like Grace Hopper Conference, using Meetups in the area, impact groups, graduate internship programs at colleges and universities, and of course, promoting the roles through online portals like Glassdoor and LinkedIn. This will ensure you interview a superset of individuals who can bring more ideas to the table and show what they have to offer you if they were to be hired.

## **Use gender-neutral language**

A small nuance but something that can go a long way, is making use of gender-neutral language as you put together the job description. This simply means to use language which is unbiased towards any particular gender or social group and is unassuming of which gender the potential hire would belong to. One way is to use preferred names given by the individual to use. This helps mitigate the unconscious gender bias that you may have and, at the same time, sends an affirmative signal to potential candidates about your team's commitment to diversity and inclusive culture in your company. Keep in mind that this is not only important to attract new talent to your company but will also help retain the amazing talent you hire. Another thing to keep in mind is not only should the job descriptions should be gender neutral, but also the feedback entered on the hiring tool once you interview a candidate should follow similar guidelines. Small steps like referencing the initials of the candidate's name versus guessing for him or her, using pronouns like "they" or "them," or using third-party language with generic titles like "the candidate," "person," or "individual" can help change the language to be gender neutral and more welcoming. One cool hackathon idea that an engineer implemented in our company recently was to create a script that ran over filled-in interview feedback on the hiring tool and suggested words that should be updated to be gender-neutral. This hackathon project really helped the company's initiative to create a more inclusive and diverse hiring pipeline.

## **Working closely with recruiting partners and coordinators**

Recruiting partners and coordinators are crucial in the hiring process. They form the backbone of an effective hiring pipeline, working in partnership with Engineering Managers. These partners can be internal recruiting functions or external agencies and placement organizations, or a combination of both. It's important to assess any agreements or contracts in place for using external agencies and the associated rules or policies. Once you have the job description, you collaborate closely with the recruiting coordinator to proceed with the hiring process. Working together, you handle tasks such as gathering resumes, initial filtering and screening, and ensuring potential candidates are a good fit. Recruiting coordinators are key in filtering out most applicants and identifying candidates who align with the job requirements. Establishing a strong working relationship with your recruiting coordinator helps them understand the specifics behind your desired skill sets and job qualifications. This enables them to identify candidates who not only meet the basic requirements but also align well with the job opening. While it's important to hire for long-term goals, there are instances where short-term team goals align with the company's long-term objectives. In such cases, working closely with the recruiting coordinator becomes even more valuable.

Once candidates are identified, the recruiter collaborates with you, the hiring team, and the candidate to schedule interview loops and provide relevant information to the interviewing team. If you decide to move forward with a candidate and extend an offer, recruiting partners typically communicate offer-related details. During offer negotiations, you collaborate with your budgeting team and recruiting partner to determine the best approach for you and the candidate. As you can see, there are multiple touchpoints where you engage with your recruiting partners. Thus, establishing a strong working relationship with partners and coordinators is critical for a successful hiring process for all parties involved.

### **Keep in mind long-term needs**

It is important to remember the company's long-term goals and needs as you hire for a position. We should not compromise on long-term needs to gain some short-term benefits. Think that you are hiring for the company and not just this team. Hence it is important to keep an eye on mutual fit with the next hire. Most engineering companies today offer a seamless transfer of teams for willing employees in certain situations. Therefore, one question to ask yourself is whether the candidate would be a good addition to other teams in the org and not just your team. If they came on board and wanted to switch teams 6 months or a year down the line, would they be just as useful there? Aside from roles that require highly specialized skill sets, the answer should always be to hire someone that could work across different organizations and teams and be a good fit anywhere. Occasionally, organizations establish hiring objectives that target underrepresented groups, aiming to hire a certain percentage of individuals from these groups. When strategizing your hiring process, it's important to consider these company-wide goals to improve planning and prevent inefficiency. For instance, a company can have an underrepresented group (URG) goal to reach for hiring by year-end. Having once been involved in an initiative on URG for a company, I worked with a team to analyze the last six months of loops conducted and came back with a few suggestions like- increasing onsite loops tapping URG talent, tapping more into impact groups like women in tech, Latinos group, etc. and introduction of inclusion training as part of hiring training.

Similarly, as you hire, you want to consider the pros and cons of smaller versus larger teams to make more considerate hiring decisions. Some common perks of a large team are increased shared resources, increased specialization of tasks as you now have more people, fewer chances of a single point of failure, increased division of labor, and strong group identification as it's a fairly larger team. On the other side, some major perks of smaller teams are- effective communication and interactions, higher individual ownership of work as one might own the entire component, ease of information sharing, and recognition of individual contributions in the group. As we can see, there is no silver bullet solution, and the hiring strategy will differ based on the use case and the charter.

## **Understanding the sourcing workflow**

An important part of sourcing is understanding the sourcing and hiring workflow and framework for the hiring pipeline. The EM must know how each role will be interviewed. Several steps will happen before a candidate comes up for an interview.

- Recruiter may contact certain candidates and filter some out before sending them to the EM
- EM can see resumes of all the candidates who applied for the position, and EM takes the decision on resume screens
- Once the candidate makes it into consideration,
- Do they have to give a test or an online assessment?
- If so, does the online assessment cover technical and personality assessments?
- Are you using some external companies for these online assessments?
- Are there candidates that can get an exemption from these assessments? What is the company policy regarding these exemptions?

The structure of workflow and answer to these questions will depend on your company, existing hiring processes, and the situation you are in. It is important to realize that you can't just assume candidates to attend an interview without going through certain workflows. So now that we have sourced potential candidates through various platforms and with the help of our recruiting partners, our next step is to help set up hiring interview rounds for these candidates.

### **7.1.3 Setup hiring rounds and the loop**

The next step in the hiring process is setting up the interview loop for potential candidates. This typically involves multiple rounds of interviews conducted by different individuals within the company, with each interviewer assessing specific skill sets relevant to the job role. The composition of the interview loop varies depending on the position being hired. For example, a junior engineer role might involve 3-5 rounds of interviews, focusing on technical skills (e.g., problem-solving, algorithm

understanding, system scalability, technical design) and soft skills (e.g., collaboration, openness to feedback, adaptability).

Designing an effective interview loop can be challenging. The goal is to create a fair and unbiased process with a diverse panel of interviewers while gathering the necessary data points for an informed hiring decision. Having a diverse set of interviewers serves two purposes. Firstly, it showcases the company's commitment to inclusion and diversity, reflecting a diverse workforce. Secondly, it ensures a wide range of perspectives in the interview and hiring decisions for internal employees. Additionally, it's important to use the company's leveling guide in the hiring process to ensure the selection of well-rounded candidates. In cases where small companies lack a robust hiring framework, it's advisable to start with simple leveling guidelines that can serve as a blueprint for future growth and hiring. The refinement of hiring rounds and the setup of interview loops can be an iterative process to determine what works best for the company.

Some of the key skills to look for in potential candidates can be:

- Self-motivated
- Collaborative nature
- Team player
- Easy going but driven attitude
- Technical and/or business acumen and skills relevant to the job role
- Eager to learn and accept challenges
- Can thrive in ambiguity
- Have a passion for the product and the company culture as they will be the ambassadors for the company
- Strong culture and team fit
- Ability to work across or in multiple areas

At the same time, keep an eye out for some red flags such as

- Show signs during the interview that they are not a team player
- Bad-mouth previous team members and managers
- Seem less open to ambiguity

- Skim answers to interview questions from the surface, but when asked to dive deep, are unable to provide concrete examples
- Signs of distrust or dishonesty
- Sounds defensive
- Goes into the weeds but easily forgets the larger picture
- Unable to communicate concepts or their thought process clearly
- Unwilling to state when they don't know an answer to something, and instead fights to prove a point

While the above list presents a generic handbook of things to look for in a candidate irrespective of the level, other resources that may be specific to the company can also be super helpful in making sure you follow a consistent, scalable and maintainable hiring process. In my teams at Microsoft, Amazon, and Robinhood, some best practices I have seen the companies follow are:

- **Leveling guide** - Companies follow a leveling guide where the levels in a role such as software engineer I, software engineer II, and so forth are marked against the expected competencies at that level. This ensures consistent hiring practices across the company and also ensures we are calibrating candidates against the competencies we need.
- **Interview Question Bank** - Setting up a standardized interview question bank can be an onerous task but is vital in making the hiring process consistent and unbiased. With a set number of questions that are well calibrated with several engineers and managers within the company, this provides an opportunity to maximize value and be efficient in making great hiring decisions. Another concept to keep in mind is ‘adaptive testing.’ What this means is as an interviewer, you might sometimes have to adjust basic things like coding language based on the candidate's preference. For example, let's say your company primarily uses Java, but you are interviewing a candidate who only has experience with the Go language. In such a scenario, you should be able to ask the right set of questions to the candidate and adapt your interview to understand their thought process and, in return, learn something new. Another importance is in the calibration process

to decide the ratings between, let's say, strong hire, hire, neutral, weak hire, and no hire, as the calibrated question bank sometimes also contains what a strong hire response should look like as opposed to hiring or neutral or other categories. Some things to remember are:

- Interview is all about evaluating the candidate in terms of what they claim as they know.
- Interviewers should not be trying to ask what they recently learned or based on their interests
- Interviewers should have a mindset that they are instruments in the process and the final decision is made by the organization or the team of folks working on this hiring.
- **Interviewing and Inclusion training** - We all have opportunities to learn and grow throughout our careers. Companies that are vested in the growth of their employees spend enormous amounts of money to train them for better interviewing and inclusion training. This helps individuals reduce individual biases (or unconscious bias) and helps bring awareness of inclusion and diversity among the employees. Also, such training can help the interviewer learn what they can share versus what they should not share or say during the interviews to ensure we keep the confidentiality intact for the job. Another thing to learn here is how to end the interviews well. Always thank the candidate for their time (irrespective of how the interview went) and let them know about the next steps or who to contact (mostly the recruiter) if they have any further questions on the interview process. This also helps set a positive example for our potential employees by creating a better candidate experience for them.
- **Handling split loops** - Split loops are interview loops where certain interviewers deem the candidate to be fit for hire, whereas others evaluate the candidate to be unfit for hire. During interviews, different interviewers can have different experiences with the same candidate; maybe the candidate performed well in the coding round but did drastically badly in the architecture design round. It is obvious to end up with split loops during the debrief session as you come together to consolidate the feedback collected throughout the multiple rounds conducted. In such scenarios, it's important to follow the leveling guide strictly and put competencies in buckets of raises bar versus

lower bar. Next, it is important to identify if the candidate seems coachable on things that are in the lower bar category and if no other major red flags were observed. At the same time, you would also want to evaluate whether the candidate seems to be a strong candidate and may have had one bad interview round. If the candidate meets these criteria and there are still differences in opinions, then the usual process is to ask for more clarification by doing additional interview rounds. This helps take an objective approach to the hiring process and helps eliminate bad hires. You may also encounter some individuals who may be extremely proficient technically and maybe better than most people in your team, but if they are hard to work with, then it is almost impossible to count this person as anything but a bad hire. A person who is one of the best coders in the world but one with a disparaging attitude towards others will create more problems than they will solve. As a suggestion, do all the mandatory rounds of interviews before rejecting a candidate. They might have some skills and a holistic view of performance across rounds to be considered. Also, between the rounds, do not let interviewers pass on or share feedback with folks doing subsequent rounds. This prevents bias from creeping in.

Now, we have mostly taken the decision whether the outcome of the interview rounds was in favor or against the potential candidates we interviewed. Candidates who do not meet the company hiring bar should be informed politely about the outcome to avoid any confusion and anxiety at their end and promote a spirit of transparency. This can be done through either a response email or a phone call, which friendly recruiting partners can take care of as they are well-trained in this area. Note that many companies have a policy of not giving reasons or having follow-on conversations about “why,” while other companies allow it in limited situations. Depending on your situation and the interviewee, you can work with the recruiting coordinator to let them know of any shareable piece of feedback. For candidates who do meet the hiring bar and if we as a company are willing to extend an offer to them, let's do a sell call for them.

#### **7.1.4 Sell call**

A sell call is an integral part of the hiring process to help close a successful candidate. This involves the hiring manager setting up some discussion time with the candidate and walking them through details about the job role, which may not be easy to pass via the recruiting coordinator. It is always beneficial to discuss what their day-to-day work would look like and how their 30/60/90-day onboarding plan would work. This is all an effort to make sure they don't feel surprised about anything when they join. In one of the recent candidates that we wanted to hire, the candidate had a lot of questions about the team's work culture, technical roadmap, what opportunities they would be a part of, and whether those opportunities would ultimately help them get to where they want to in their professional career in the short term. Sell call seems to be a term where you are "selling" the company's job role to the potential recruit, but another way to think about it is that it also gives a space for the candidate to make sure that they are signing up for something that they are really comfortable joining. The worst outcome that can occur from the hiring process is that you end up hiring someone who hates their job once they join because they didn't have enough transparency into certain aspects of their job that would matter to them. Therefore, "Sell Calls" are the perfect place to ensure you answer all questions from the potential hire and also provide them with information that they might not really ask for but you think is important.

One example of this is when I was recently trying to hire a senior software engineer who had multiple offers. The engineer here was fairly popular in the job market, and at the same time, their resume mentioned that they had done a couple of AWS certifications. I used the sell call as an opportunity to gauge their interest in working with AWS-related technology stack and laid out some projects from the roadmap that would both have opportunities to provide guidance (provided the AWS certification they had) as well as opportunities to learn in more depth about using AWS technologies at scale. This was a key factor in the engineer picking our company over others when making their next career decision. Again, you often try to attract top-performing individuals who might be passive in the interview process. Hence it is important to have an elevator pitch ready for the job role, assure them that they will be valued and the job change will be worth it for them.

Sell calls are also a great opportunity to discuss what other company benefits would be available to the employee if they decided to join. At the same time, some companies have rules and policies about leaving the benefit-related information to specific organizations in the company(usually HR), while others don't mind if the EM shares it as well. The reader should do their due diligence before discussing benefits-related details in the sell call. If you find yourself in a position where you can discuss company benefits, then this call is a great way to understand what kind of benefits would be good for this person. For example, for the engineer that we were trying to hire above, one of the key things that they were looking for was continuous learning and continuous technical growth. Luckily my company did offer some education budget every year, which helped them cover many courses they wanted to take in the coming months. This also was one more reason which helped the candidate make a clear decision to join our company.

It's important to acknowledge that not all job positions will be equally attractive as those found in large technology companies. As an EM, your responsibility is to provide an accurate representation of the job role and its expectations rather than obscuring them. Being honest and transparent helps establish realistic expectations for everyone involved and prevents unwelcome surprises later on. One approach to managing such situations involves highlighting the opportunities and potential impact associated with the job role, which can contribute to raising the overall standards within the company. Simultaneously, it is crucial to demonstrate your commitment to their personal growth and emphasize the value they can gain from embracing the role, thereby assisting them in making a positive decision.

One more thing that I always do with all of my potential hires is to encourage them to talk to other members of the team, some members/EM of sister or partner teams as well as non-technical cross-functional partners that our team closely works with. This provides the candidate with a 360 lens of what it would feel like to work in our team before joining it. Remember, by the time you are making these calls, you already feel that the candidate is a good fit for your team. The goal of the sell call here is to make sure you provide all the transparency that is required for the candidate

to decide whether the team is a good fit for them. Having a 360 lens and talking to different people is a great way of achieving that.

In the end, regardless of whether the candidate decides to join the company, ensure you always thank the candidate for their time, as they are not only your potential employees but also your potential customers. At the same time, making a decision to join a certain company is a nerve-wracking decision for most of us. So, if things don't work out this time, you always want to keep the door open for any future collaborations. This is actually not only true for candidates you want to extend an offer to but to all candidates in general. A disgruntled or disrespected candidate will have nothing good to say to their friends and peers about your company. They might take it to online forums to share their experiences, good or bad. Glassdoor and Fishbowl are some platforms that folks use. If, for any reason, the candidate doesn't end up joining the company, you still hope that they enjoy the interview process and spread this good experience to others in the industry.

### **7.1.5 Onboarding plan**

The cycle doesn't end with a sell call and the candidate deciding to join the company. It is the onboarding plan that makes a huge difference in terms of how successful one will be in the initial phase at the new company. This involves the EM setting a detailed onboarding plan for the new hire that sets a 30-60-90 day plan. To aid the smooth transition, an onboarding buddy/mentor/co-pilot might be assigned, more frequent one-on-one, and ensuring the right training material is provided for onboarding is essential. Time spent preparing the onboarding plan is time well spent. Below I will share a sample onboarding plan template that can be used if you don't have one.

**Figure 7.1 Sample onboarding plan for new hires**

## **<Sample> Onboarding Plan for Alice**

Hello Alice!

We are excited to have you in the team.

**Team Overview and Details:**

<Other description on company, mission, team and tenets>

**Onboarding Buddy:** Bob

**Onboarding Buddy Contact information:** <Slack Handle>, <Email> (Whatever communication channel your team prefers)

**Key people to meet:**

<Here you suggest names of key people the new employee should meet. This can be set of peer software engineers they will work with, cross-functional partners like product, technical program managers, UX, QA and other leaders in the chain>

**Tasks/Activities:**

<This section explains the desired courses, trainings or logistic paperwork that needs to be completed in the coming weeks>

Task #	Task description	Target Finish week	Status (TODO, in progress, completed)
1	Complete onboarding training bootcamp	Week 1	In Progress
2	New hire paperwork	Week 1	To-do
3	...	...	...

**Key projects/deliverables:**

<This section goes in further details of plan for next 30, 60 and 90 days. This can either be combined with the above section or keep separate to describe the projects and key deliverables to own in the upcoming three months>

**First 30 days:**

- E.g., Understand the landscape of services and infrastructure
- Meet critical stakeholders

**First 60 days:**

- E.g., Take ownership of core services and prepare for launch of project 'XYZ'

**First 90 days:**

- E.g., Prepare to go on-call for team's weekly operational on-call rotation
- Help launch the core pricing service
- Help prepare a robust runbook for ABC feature

Now that we have a hiring framework in mind where we follow a step-by-step process to hire the best and maintain our hiring bar, let's look at what factors go into making hiring decisions that involve staffing from scratch or hiring for an existing team with an established charter.

## **7.2 Staffing from scratch versus hiring for existing team**

Staffing decisions can vary depending on the team you are hiring for. Once hired, you, as EM, cannot easily go back on the decision, and hence it is important to be thoughtful and attentive to details when making such critical hiring decisions. Staffing from scratch is generally applicable for scenarios where you are setting up a brand new team. This can stem from

either a new initiative the company would like to invest in or with the growing years, the charter has increased, and there is a need for a separate team to own a set of features and components. This type of hiring will involve you as EM to address the primary question of the purpose and vision of this new team and how many job roles are needed to get the work done. This will be important to share with potential candidates to help attract them to this team. On the other hand, hiring for an existing team is when the team already exists and has a clear purpose and direction defined. You are hiring more members due to increasing workload in the team, charter increase, or if the team went through some recent attrition and the roles need to be backfilled. In such a situation, it is more clear what the potential candidates will be working on delivering, but at the same time, you need to ensure the new hires can gel up well with the existing team members. Let's look below at the detailed differences between the two to help us make a considerate staffing decision for our team.

### **7.2.1 Purpose**

When staffing team members from scratch, be very clear with the purpose and tenets of this new team. A new team should have an organization chart penciled in and at least a rough idea of the purpose and mission of that team. This will help have a clear vision from the beginning and ensure you are making responsible decisions as you work on hiring and staffing this team. This will also help you to share the vision with potential candidates who are also interviewing you to find their next mutually fit job. When hiring for an existing team, purpose, direction, and tenets usually will be clearly established for an existing team, unless the team is going through a reorg or charter change. In this scenario, it is easier to sell the team to potential candidates as we know its layout and what they will be coming in to deliver.

### **7.2.2 Sizing**

If the team is being staffed from scratch, it is important to estimate the sizing and headcounts needed in at least a 1/3/5 year timeframe. The charter above will help take the best guess in terms of how many engineers would

be a good fit to start with. You can always hire more engineers as needed, but the key is to avoid over-hiring initially, which might lead to insufficient work for team members and, eventually, attrition. Ideally, I agree with the common opinion that any scrum team should not exceed more than two pizza teams, a team size that can be fed by two pizzas. That entails max 6-8 engineers on the single scrum team.

When adding to an existing team, there can be several reasons why we need to hire. It could be that you are expanding the team's charter, or the existing charter requires additional headcount to support existing initiatives because of customer growth or backfill for an existing role caused by attrition. In all three scenarios, the requirement to fill the headcount is fairly evident based on the immediate requirements, and hence team sizing doesn't really become a big issue. By the time you want to hire, the importance of the role is well established, and hence you pretty much know the exact requirements for the candidate that would fill that role. The only thing to remember is not to grow the scrum team larger than the two pizza constructs or, alternatively, something that you feel may not be manageable in the long run.

### **7.2.3 Specialization of skills and experience**

When building a team from scratch, you must remember that you have a mix of team members with various technical skills and years of experience. Let's break it into parts to understand it fully. The variety of technical skills will help to make a balanced full-stack team if you aim for one or people who can learn new skills and bring fresh ideas to the table from their previous work experiences. Similarly, having a mix of junior and senior engineers in the team helps to balance off the years of experience. Junior engineers can be amazing because they are curious and willing to learn. One can mold them based on the strengths and needs of the company. Senior engineers bring a variety of experiences to the table, having worked with large-scale distributed systems and an opportunity to coach other junior engineers in the team. Remember, there is no silver bullet ratio for junior, mid, and senior engineers to have in the team as it depends on various factors such as project, and work environment, to name a few.

Depending on the project and the type of work required, a blend of engineers with various years of experience will help bring the best of all worlds together.

For existing teams, filling roles based on skills and experience can be relatively easier as you are already aware of the build of the current team. If you have 4 junior engineers and one open role, it makes sense to hire a senior engineer to help coach the junior engineers in the team. Similarly, if your team is a pure frontend team and you have existing iOS and web developers, it is obvious to hire an Android engineer if you have an open headcount. Hence, the aim is to maintain a balanced ratio of software engineers early in their careers to senior engineers.

#### **7.2.4 Diversity**

Hiring for diversity reflects the values and needs of a diverse customer base. By having a team that mirrors the diversity of the target audience, organizations can better understand and meet the needs of a broader range of customers. This can increase customer satisfaction, loyalty, and overall business success.

To ensure diversity considerations are integrated into the hiring process, organizations can implement strategies such as using diverse candidate sourcing channels, conducting blind resume reviews, promoting inclusive language in job descriptions, and implementing diversity training for interviewers and decision-makers. It is also important to establish inclusive policies and practices that support and retain a diverse workforce beyond the hiring stage.

One can plan better for diversity to help bring the outsider lens as you are building the team from scratch. This allows you to decide on the diversity of skill sets and backgrounds you are looking for and then hire based on the criteria you need. For existing teams, since we have a frame of references of who are our current team members, using that information, we can tap into different diversity programs to maintain a good balance of diversity in the

team. Also, it is helpful to make the team members undergo inclusion training to promote allies and a healthy work environment.

### **7.2.5 Team dynamics**

Team dynamics is a fundamental aspect when building a team, as you want to ensure the team members collaborate well and work together as a team. When building a new team, this has less of a role to play as there is no established team culture yet or no functioning team dynamic that exists and will be built over time as you hire and grow the team. At the same time, it is important to be clear on the expectations you have from each member of the team so as to avoid any confusion later on. For example, you don't want to have two senior engineers on the team competing to be the team's tech lead. This can cause unhealthy competition and thrash, which will be detrimental to the team's progress. Instead, clearly articulating which engineer should work as the tech lead and what the expectation from the other is the way to avoid further miscommunication.

With an existing team in place, team dynamics play a crucial role as you want to ensure the new hire in the team can gel up well with the existing team mission and culture. Each one should be accepting of one another and be open to working together as a team. For example, if the current team culture is to do the majority of planning together and collaborate, bringing in someone new who is not a team player can be detrimental to the team culture and morale.

### **7.2.6 Tale from the trenches**

One of the times I had to spin up a sister team from scratch that would focus on payments portfolios in our global markets. I started to understand the vision, mission, and tenets of the team before I got into hiring for it. For example, one of the tenets of the team was to provide a seamless payment transactions experience to our customers, which is also extensible for future use cases. This also meant defining the success criteria for the team. I had 6 open headcounts and so started interviewing folks with a variety of skill sets we would need to be successful in the team. I ensured a sustainable

ratio of software engineers to senior software engineers to lead software engineers in the team. I also saw interest from existing team members from the sister team and was also able to successfully move two people internally. One of the times, I was tasked to hire for an existing team that initially had 6 members. I had 4 open headcounts, so I planned well based on what skill sets exist in the team diversity and then targeted potential candidates using that information. I had to ensure we are still closer to the two pizza rules in Agile to avoid disruption and continue with effective communication and, at the same time, ensure new hires maintain the team dynamics and don't provide disruption to the current team.

As we saw in the above sections, there are a variety of factors that help decide how one should be staffing the team and building a hiring strategy. Since going back to who you have on-boarded and hired in the team is not a good option in the decision-making process, it is paramount to take it slow and make iterative check-ins instead of a big-bang hiring approach. Next, let's look at the core differences between hiring someone from the outside versus growing people internally from the team to help shape the team culture.

## **7.3 Hiring externally versus growing internal people**

Hiring externally new team members versus growing internal people from the organization is a common dilemma that most EMs face frequently. Well, there are advantages and disadvantages to both approaches, with no right or wrong answer. Aiming for a mix of both, based on the situation at hand, is the key to making such a decision. Let's look at some scenarios where one approach might fit better.

### **7.3.1 External hiring**

External hiring brings fresh perspectives and enriching experiences to teams. Candidates hired from outside bring knowledge and insights from their previous organizations, whether from large companies or startups.

This diversity of experience enhances the team's thought processes, challenges existing assumptions, and fosters an innovative culture. For instance, if the team lacks certain skills, such as machine learning expertise, hiring someone with that background can significantly benefit the team. Similarly, external hiring can be beneficial when seeking a balanced seniority level within the team. For example, if there's a junior Android engineer who would benefit from guidance and mentorship, hiring a senior Android engineer can help bridge that gap and ensure a balanced Android experience in the team. It's important to note that external hires may face a learning curve to understand the technology stack and company culture. Consequently, their short-term productivity may be relatively low. Allowing new hires ample time to ramp up and build a strong foundation will ultimately contribute to the long-term growth of the team.

Another big problem with a team that has not seen a new face in a while is the tunnel vision syndrome that can occur. Often the team doesn't even realize what perspective they are missing until someone new comes in and provides an outside perspective of how things can be improved. Utilizing the new hires' observation skills and collecting feedback on how they think this team operates differently from their previous teams often points us to inefficiencies that are not easy to find on your own. One example is that a new engineer in my recent team pointed out how technical design documents were written in their previous company and why they think that our team could adapt certain aspects of the design documents to make things better and the technical review process smoother. This actually led to an entire overhaul of the design review process. The team created a new template for design review documents and brought consistency to what is expected from each design document. This not only made things consistent and the reviews faster but also forced the writer of these documents to think about certain aspects of design which some would have missed otherwise.

### **7.3.2 Growing people internally**

As EMs, it is our job to help our team members plan for career development so they thrive in the role and when the time comes, are ready for their next level. Helping internal people grow and fill in the available open roles is a

great way to set positive reinforcement and send the message that you, as EM and the company are vested in their growth. Having a tendency to always hire from outside with no effort to grow people internally comes at a massive cost to the company. As you and your employees start to realize the fact that growing internally is not an option, the easiest thing for people to do here is to leave and join another company that would care about their professional growth. One of the biggest issues caused by this is the sudden drop in the contextual knowledge that engineers who have been on the team for a while would have. Contextual knowledge is hard to build and easy to lose, and yet without it, it's nearly impossible to have a high product delivery rate without causing regressions. Having an EM vested in their employees' growth helps motivate people to build a sense of engagement and trust. Of Course, this also helps the retention rates as it sends a positive signal to the employees. Further, there is less ramp-up time for such folks as they know the technical stack and processes in place, saving the cost factor to onboard and ramp up a new hire.

While growing people internally seems like a great option, it is, in fact, a great option just doing that and not having any external hires can risk your team having a narrow vision. Hence without having external hires, you risk losing adding new skills in the team that you wouldn't get by just growing people internally, nor do you get access to the plethora of experience that new folks can bring in having worked in different companies. Therefore, it is important to have a balanced approach of hiring externally vs growing people internally. If opportunities are created that you think existing team members can step into and grow, then that's a win-win situation for all parties. At the same time, realizing what requires an external hire is equally important. For example, Bob, an employee in the company, has expressed interest in becoming a product manager. With the recent departure of the team's product manager, this presents a great opportunity to support Bob's career aspirations while ensuring a smooth transition, considering the company's pricing product. Bob's familiarity with the company will expedite his onboarding process and allow him to hit the ground running in his new role.

On the other hand, the company's data scientist recently left for personal reasons. Alice, currently working as a test engineer, has expressed her desire to become a data scientist in the future. However, Alice currently lacks the necessary skills and experience to function effectively in that role without guidance from a senior data scientist. In this case, it would be beneficial to hire an external data scientist who can not only fulfill the immediate needs of the team but also serve as a mentor to help Alice grow into a proficient data scientist over time. By strategically considering the career aspirations of employees like Bob and Alice, as well as the specific needs and opportunities within the company, we can make informed decisions about internal promotions and external hiring to foster professional growth and drive the team's success.

## 7.4 Developing the hiring pipeline

Hiring, especially in competitive markets, can be challenging and onerous. The latest trend is to consider hiring as a proactive measure rather than reactive based on hiring needs. This helps in multiple ways - first to ensure we are moving fast to make hiring real and efficient, helps ensure we are sourcing the highest quality candidates using our resources, and helps us stay ahead in the game compared to competitors; called 'forward thinking.' One important thing that we need to be aware of when building a talent pipeline is the strategic direction of the company. That will help plan and build a roadmap to reach the right potential of candidates out there. Let's look at a few steps organizations can take to distinguish themselves.

1. **Branding** - Setting up a company's brand is the first impression any potential candidate has when they are introduced to the company. It is the company's time to shine and tell a story to the external world. With a clear company vision and mission, it is easy to reach candidates and share what skill sets you are looking for. This includes the career page that shows the open roles as well as advertisements about the company culture and also the social media page for the company. Other ways to help build your company brand include advertising on social media, LinkedIn, networking events, and meetups. Having these resources

helps candidates learn more about the company's value system and nudges the candidate who shares a common vision and values to hope to work at the company.

2. **Streamlined hiring process** - This step involves having a well-thought-out hiring process to funnel in candidates through the right channels(as an example, reach out to an active and well-known speaker on iOS topics when you need to build out competency on your mobile app team), advertising jobs with gender-neutral language to have maximum outreach and be inclusive, setup interview loops with the desired training to interviewers to avoid unconscious bias, and fair debrief and hiring outcomes. Using the right tooling can take one long way in the process, so there is a central hiring portal that avoids confusion. Certain roles can be hard to fill that require particular niche skills; in such scenarios, it is important to tap the right forums where such talent will be available. For example, if you are looking for React developers, it is a good idea to attend a React conference to tap the talent that will have most of the audience interested in React development.
3. **Personalized reach**- As people go through the interview process, it is on us as interviewers and potential employers to make the candidates feel valued, which is a big driving factor to sway someone to join a company. This personalization can start from the first step, when recruiters or you as hiring manager reach out to potential candidates through channels such as Linkedin, all the way to the end of the hiring process. This can also help engage passive candidates not actively looking for a role change. It will also help if we target the pool of already vetted candidates who have either previously applied to open roles, are coming through referrals(often have better chances of success as compared to cold candidates, so worth valuing referrals more), or have cleared some rounds in the past but had paused the process due to unwanted circumstances.
4. **Completing the circle** - Yes, it is extremely important to complete the circle and nurture your potential candidates. This means always circling back to your candidates whether they made the cut and have been extended a job offer or were unable to clear the interview bar but can still be great employees for the future, instead of ghosting the

candidates. This helps to close the loop, and help candidates have a boolean answer to help them plan ahead. It also provides a good candidate experience so they keep the company in mind for future searches. A good tip is to always thank the candidates for their time and effort. In addition, don't drag things out. There is no reason for interviews to go on for weeks (or months in some scenarios) to find the right candidates. Keep things short and sweet, and respond quickly. Candidates will always appreciate a quick turnaround, even if the answer is a no, rather than a long, dragging sequence of visits.

As a note, with remote interviews rising, it has become important to clarify the remote interview setup, including the platforms used for conducting interviews. Remote interviews may limit your ability to observe non-verbal cues compared to in-person interviews. Pay close attention to verbal communication, tone, and candidate engagement to gather as much information as possible. Developing a hiring pipeline and a steady stream of talent can be difficult at the start, but once created can really help ace the hiring process and plan ahead for attritions, if any. Also, using metrics, one can keep a data-driven approach to see what works well and if any opportunities can be introduced in the hiring process. Successful companies master the art of developing a hiring pipeline to keep the funnel going and plan ahead for the future.

## **7.5 Hiring programs for positive reinforcement**

Hiring is not as straightforward as it sounds. One needs to have the right checks in place and a robust hiring program to tap talent and promote the employer brand. But can we do more? Here comes the importance of opening up channels for showing that the company is vested in individuals and instituting positive reinforcement programs, as shared below, to spread awareness of the brand and attract talent. If done right, the strategies below can help build a healthy and diverse talent pipeline.

### **7.5.1 Internship/Campus hiring programs**

Are you looking for someone with fresh perspectives who can bring academic learning to practice? Is some of the work at your workplace coachable for a new hire fresh out of college to learn and perform? Well, yes- internships, co-ops, and campus hiring programs are a great way to engage and attract young talent with fresh perspectives and recent graduates for starting at the company. One big advantage of campus hiring is that such hires are mostly early in their career and can be molded as per the job requirements. They have a fire to learn and grow and have fewer perceptions or biases due to their limited experience. We rarely hire them based on what programming language they know, but more regarding hiring for the team and the company fit first. We can train them in the skills needed to do the job and create a win-win situation for us and them. This also plays a crucial role in branding the company as the college hires are inquisitive to know about the company values, culture, and of course, the perks the company has to offer. Also, this is like helping set up an early customer base for your products.

Several companies use campus events and career fairs as an opportunity to build their brand and bring awareness among the university crowd, and to get to the alumni network to recruit experienced individuals. As an example, if you have existing employees who are alumni from a particular university, taking them to conduct a few on-campus awareness sessions in the same university and sharing their success stories will show the current students the benefits and growth opportunities at your company-hence, increasing the chances of them considering full-time opportunities in your company. This is also a good opportunity to collect resumes, answer relevant questions, and share a point of contact if students have questions beyond the career fair or the event.

With the pandemic, virtual career fairs have been on the rise. The pro is that we now have opportunities to invite talent across borders for internships and full-time opportunities. The con is that we miss Facetime and personal touch because of being virtual. Personally, I was able to recruit interns and full-time employees from countries like Canada and Europe to work with my US-based team. As an EM, it is also personally satisfying to see college hires joining as full-time employees and reaching great heights.

## 7.5.2 Returnship programs

Several big companies have acknowledged the fact that things happen and with life incidents, there can be several scenarios where people take a break from the career for various reasons:

- Care for elderly
- Care for kids
- Take a part-time role as they support someone in family
- Relocating because of family reasons
- Are in search of remote roles only
- Many more...

Companies like Amazon, Meta, Google, [Thoughtworks](#) (<https://www.thoughtworks.com/en-in/careers/program/talk-tech/vapasi>) and many more have established returnship programs to help individuals restart their careers after a break of 1 year or more. The idea is to help these individuals by ensuring they have the basic qualifications needed for the job, a structured environment, training them on the job with the latest technology, and having a probation test period, where they can work more like interns. Based on the work and the company hiring bar, the employer can decide to extend a full-time offer for hire. As returnship programs become more and more popular, let's look at the benefits of instilling such a program:

- As you hire people by providing a platform for them to refresh skills, they bring in their existing experience and skills from past and can help bring new ideas and fresh perspectives to the table
- This helps attract diversity back to the workforce as women have been more prone to leaving jobs than men., Such programs provide an equal opportunity for them to upskill themselves and get back to business
- Such programs help create a company brand. As you aim to attract talent, having such programs to welcome the workforce back to work is a way to positively affirm you are vested in the people you employ

In my time at Audible, I saw several successful cases of returnship candidates who, with some guided coaching and ramp up programs, were able to pick up work quickly and join the company as full-time employees. Clearly, this is one good way to attract diverse talent and help build the company brand.

### **7.5.3 Diversity, equity, and inclusion programs**

Diversity, equity, and inclusion (DEI) programs are essential for the success of any company. DEI encompasses having a workforce with diverse backgrounds, ideas, faiths, experiences, age groups, genders, sexual orientations, and walks of life. This variety brings a unique flavor to the company's culture. Equity, conversely, ensures that individuals with equal skill sets and experience are compensated equally, regardless of other factors.

As an EM and leader, it is crucial to promote awareness of diversity and inclusion, attract diverse talent, and retain existing talent. Companies utilize various initiatives such as working moms groups, groups for Latinos, women in tech, pride tech groups, support for veterans, individuals with disabilities, and black employee networks to attract and retain diverse talent. These programs often include fireside chats, awareness weeks, and reminders about the importance of diversity to foster a positive workplace environment. Additionally, providing employees with flexibility in their work environment and being accommodating and empathetic contribute to creating a psychologically safe workplace.

I can provide an example of an Allyship program I was involved in. The program aimed to gather like-minded individuals to become better allies who promote awareness of diversity and inclusion. This was achieved through working groups, open discussions on relevant topics, watching inclusion videos together, and mandatory inclusion training. These initiatives not only fostered strong bonds within the organization but also enhanced understanding, openness, and empathy toward others' situations and perspectives. It's important to note that DEI is not just about hiring or introducing programs; it's a continuous improvement process and a mindset.

It requires finding the right balance and ensuring that DEI principles permeate every aspect of the company's operations rather than being treated as mere policies or procedures. While companies can use various ways to affirm positive reinforcement towards hiring and growing, as engineering leaders, it is on us to take that message and help spread the word among our team and practice what we preach.

### **What do other leaders have to say about it:**

“As a leader, I’m always scouting for talent, even when I do not have a headcount. Talented people can help us achieve success, and they can be found everywhere. So I make it a point to stay connected with talented people I meet.”

**~ Sumit Kumar, System EM at Cisco**

“My Hiring Philosophy is to hire for the attitude, and you can always train for skills. To elaborate skills in the technology world go through many changes, and every professional must regularly update and upgrade themselves. However, the inherent character of an individual is what makes them successful. The attributes I look for are the ability to learn, work in a team, problem-solving skills, and last but not the least “can-do attitude”. In today's world of various tools like GitHub, StackOverflow, or the latest ChatGPT, we all are expected to do well on tech competencies however none of these tools will ever teach a person to be a great team player, which in a tech world is probably the single biggest need from every individual. ”

**~ Devika Ahuja, Technology Leader | Strategist**

“When deciding to hire externally versus internally, I prefer developing my own team members for the new positions or even hiring from the larger organization internally; however, there are many circumstances one has to hire externally too. First is the growth scenario when I want to grow my team for various business needs. Second, if there are immediate short-term requirements that cannot be fulfilled within the team, and lastly, if there are special skills required which are difficult to train or if you want to build

new competencies in the team. Sometimes it helps to hire new team members externally as they bring a fresh perspective and newer ideas which helps the overall team.”

**~ Devika Ahuja, Technology Leader | Strategist**

“When deciding to hire externally versus internally, some of my considerations tend to be:

- 1 Has an existing staff member expressed career interest in this need?
- 2 Is this need incremental to an existing staff member, or is it net new skills for the team/organization?
- 3 What is the impact of moving an individual into this role/need on their existing team? Does it shift the skill gap to a new place, or does it fill the skill gap?
- 4 Is there an existing staff member looking to demonstrate growth to help solidify a promotion or improvement in performance?
- 5 Is it easy/cost-effective to hire someone with this skill?”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“When hiring people, I look for the growth rate. I don’t mind as much where someone is right now, but I want to get enough data points to understand their trajectory. A high growth rate usually means they can learn fast, adapt, work hard, etc. (Obviously, there are ways to fake this, and this is not the only relevant data point). I also look for grit - how will this candidate handle adversity?”

**~ Jean Bredeche, Head of Engineering at Patch, (ex) Robinhood, Quantopian, Hubspot**

“For closing senior engineers, creating an environment of Trust and transparency is key! Earning trust is a two-way street; a manager has an equally important responsibility to earn the trust of engineers to be in lock-step in all team matters. Transparency by providing well articulated and actionable feedback, sharing opportunities for growth and improving along with acceptable behaviors.”

~ **Saurabh Gandhi | Sr. Director Software Development Audible | Amazon | Ex-Amex**

“Hire for team fit first, and technical skills second. You can teach (and learn) new technologies, new languages, and new skills. If you find someone that is a good fit for your team and your company and matches your corporate culture, you'll have a much more successful time working toward the right technical fit. On the contrary, if you hire for technical skills first, and you don't have a good team fit, you're negatively impacting not only your new hire and yourself but your existing team as well.”

~ **Bruce Bergman, Manager at Lytx**

## 7.6 Stop & Think: Practice questions

1. How many open roles do I have in my organization?
2. What kind of open roles do I have in my organization, and is there a need to backfill them?
3. Is my org consistently meeting the annual hiring goals?
4. Do we encourage team members to use referrals to attract talent?
5. How are referrals treated or considered at my company?
6. What is the ratio of offers extended to accepted?
7. What is the retention rate of employees?

## 7.7 Summary

- Hiring new employees is a complex and time-consuming process that demands careful planning and attention to detail. It involves

identifying the required skills and qualities, sourcing candidates, reviewing resumes, and conducting interviews to find the ideal match.

- The process can become more challenging when there is a need to evaluate potential candidates, even when not continually hiring. Ultimately, successful hiring necessitates a proactive and well-organized approach.
- Hiring framework involves four key steps:
  - Sourcing of high-potential candidates that might make a good fit for the role and the company. This also involves tapping into diverse talent to bring different perspectives to the table
  - Set up hiring with clear competencies defined per interviewer, focus on a fair interview loop with caution against unconscious bias
  - A sales call or sell call to share about the team and the product. This involves sharing with the potential candidate what the team has to offer and how it would align with company goals and candidates' career development goals
  - The last step is to ensure a smooth onboarding experience for the new employee. This involves creating a robust onboarding plan plus assigning them an onboarding buddy to help navigate the first few months at the new company
- Hiring for a new team from scratch and hiring for an existing team can involve different considerations and approaches.
  - When building a team from scratch, you have more flexibility to plan and strategize regarding the skills and diversity you want to include. You can also establish the team's culture and values from the beginning.
  - When hiring for an existing team, you must consider how the new hire will fit in with the existing team dynamic and culture. It's important to ensure that the new employee will be able to work well with the rest of the team and contribute to the overall success of the group.
- When considering whether to fill an open job role by hiring externally or promoting from within, it is important to weigh the pros and cons of both options.

- Hiring externally can bring in new perspectives and expand your professional network. P
- Promoting from within can demonstrate your commitment to your employees' growth and potentially increase retention rates.
- Ultimately, the right decision will depend on the specific needs and goals of your company.
- Maintaining a healthy hiring pipeline is an important aspect of successful talent acquisition. It helps ensure that you have a pool of qualified candidates to draw from when you have open positions, which can help you move quickly to fill those roles.
- To build and maintain a strong pipeline, it is important to focus on your company's brand and reputation, streamline the hiring process, and make personal connections with potential candidates.
- In addition to internships and campus hiring programs, which allow companies to recruit young talent from universities, some organizations have implemented other initiatives to attract and retain a diverse and talented workforce. Returnship programs, for example, offer support and encouragement to individuals who have taken career breaks and are looking to return to the workforce.
- Diversity and inclusion programs also play a crucial role in building a positive and inclusive work culture and can help companies attract and retain a diverse workforce. These types of initiatives can help companies to foster a positive and supportive work environment, which can ultimately lead to increased employee satisfaction and retention.

# **8 Handling attrition**

When you ask someone to leave where they are or ask them to join you, you are asking them to make a commitment. I take the commitment seriously.

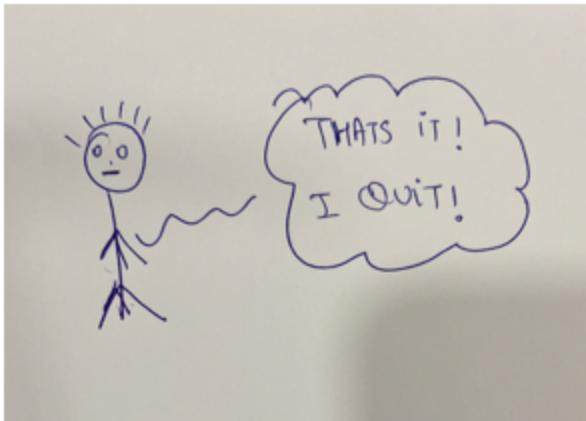
- **Myra Norton `President of Company Arena**

## **This chapter covers**

- Concept of attrition
- Attrition is inevitable
- Reasons why people decide to leave a company and move on
- Impact of attrition
- How to get ahead of attrition to reduce its impact

You get a ping notification on your cell phone, it's the [Blind](https://www.teamblind.com/) (<https://www.teamblind.com/>) app- sharing information about someone in the technology sector who doubled their compensation by moving from one company to another. Reading this, you are tempted for sure and curious to see the latest job trends in the market. Similarly, you get hit up by recruiters weekly through LinkedIn and other sources, telling you all about the job roles, the fancy perks like free meals, and the higher compensation the companies offer. You might have ignored some of the previous emails from the recruiters, but this makes you wonder if you are in the proper role and at the right place.

**Figure 8.1 An employee feeling frustrated and ready to quit**



Yes, these are classic examples of what leads to attrition with the current employing company. In simple terms, attrition is defined as the reduction in the size of the workforce by being unable to retain talent or letting people go, which leads to a knowledge drain and context loss for the current employer.

## 8.1 Attrition is inevitable

It can be a Northstar goal for a company to achieve a close to zero attrition rate, but the reality will mostly be very different from it. Attrition is a phenomenon that is inevitable in the changing world, and hence one needs to be competent to plan ahead for it and ultimately try to get ahead of it. Imagine a scenario where your team member Dennis quits the job. Is the next step as simple as finding a Jack who can replace Dennis? The answer is no.

To make up for the loss of departing employee Dennis,

- The engineering team needs to share and balance the workload in the interim until a new hire/backfill role is figured out
- The recruiting team needs to be informed, and plans need to be made for listing an opening for a role
- The accounting team needs to do final settlements and pay out the employee

- Information security (InfoSec) needs to lock down accounts and remove access for departing employee
- The hiring team needs to tap potential candidates for the role and set up interview loops
- The engineering team needs to spare time to interview a pool of candidates before they are ready to extend an offer to matching candidates. Unless you hit the jackpot with the perfect candidate up front, you are looking at a team commitment of several weeks just to interview the pool of candidates
- Next, the candidate has to accept the offer to join the company (imagine a hot job market where people have multiple offers in hand, so some of the companies will lose good candidates because of the candidate accepting a different job offer)
- Once accepted, the team needs to set up an onboarding plan for the employee to help them get up to speed on technology and projects. Also, one cannot deny that gaining expertise in an area will take time to build contextual knowledge.

Also, in a case where someone quits (and does not quit immediately), one can expect the contribution from the employee to be much lower than expected. When someone has committed to leave, they are starting to check out mentally. So looking at the above steps, clearly filling up the position of a lost employee takes time, effort, and resources and hence has a considerable cost associated with it. To summarize:

***Cost of backfill = Cost of hiring + cost of onboarding and training + cost of learning and skill development + cost of unfilled time and much more***

Some [studies](https://www.huffpost.com/entry/how-much-does-employee-turnover-really-cost_b_587fbaf9e4b0474ad4874fb7) ([https://www.huffpost.com/entry/how-much-does-employee-turnover-really-cost\\_b\\_587fbaf9e4b0474ad4874fb7](https://www.huffpost.com/entry/how-much-does-employee-turnover-really-cost_b_587fbaf9e4b0474ad4874fb7)) suggest the cost to backfill an employee is close to tens of thousands or 1.5-2x the employee's annual salary. People and products are any company's strength, so keeping employees happy and retaining them is essential. Also, I would like to touch upon the fact that attrition can sometimes be suitable for a company-

Well, yes, attrition can be good for a company where low performers were managed and let go or departed on their own, as the job role was not a mutual fit for the employee and the company. Although it is tough to see people go, this lets the company bring in fresh talent and perspectives that are a better culture fit match for the company. ‘Quite quitting’ is another deadly form of not necessarily quitting the job but not working hard enough to go above and beyond. This leads to reduced output at work where the individual only performs the bare minimum needed. Sometimes in such cases, it is better for the employee to leave and the company to take this attrition versus retaining them. In the next section, we will learn about various reasons influencing an employee's decision to leave and move on to the next employer.

## **8.2 Reasons people leave**

Traditionally, we have seen people staying with one employer for a long time. We have seen this commonly in government and defense company roles where folks would work for 20-30 years with the motivation of pensions (rather than 401ks), job security, security clearances, and many more. Nowadays, people tend to move more from one company to another. Especially in the technology sector, people tend to jump between companies in search of better compensation, perks, and work. Gone are those days when people would stick for a lifetime with the same employer they started their career with. Oftentimes, people leave an employer for various reasons; some can be voluntary, while some on the other hand are involuntary. In most cases, the attrition might be known to the manager (at least eventually), while in some cases, the leadership can get a sense and might suspect an upcoming attrition. We will see in one of the sections below how to get ahead of attrition to handle such scenarios. For now, let's deep dive into each of the two broad categories that constitute reasons for someone moving on.

### **8.2.1 Voluntary reasons**

In the majority of cases, the reasons for attrition are voluntary. It is the employee's choice to change employers for many reasons, as below. Let's explore some of the voluntary reasons that cause attrition.

### **Alignment with company vision and principles**

Alice works for a company called ABC that is on a mission to create a first-in-class social network site for people to connect and network. Alice, being a mother of two, feels the social network has proved a distraction for her kids, who would spend hours on the ABC website when instead they could have spent that time working on their college projects. She is at a point where she doesn't feel aligned with the company's vision and principles, not that they are wrong in any way, but just that for Alice and her family, they don't seem to be working well in her eyes. She decides to eventually move to a different product company, XYZ because she is not aligned with the company's vision and mission statement.

In the scenario above, it is a classic example of someone's personal belief and choice that, in turn, does not align with the company's values, and hence the employee is less keen on continuing the job in the role. There can be times when people are not aligned with senior leadership and management practices, hence another reason for not staying with the current employer. Also, it is worth noting that company vision and principles can change over time; some reasons can be changes in company direction and change of leadership (like CEO, CTO), to name a few. So it can happen that someone started at a company that matched their own personal ethics and mission but, with the change in direction, has pivoted to a different mission statement that does not align with the individual anymore. Personally, I have seen people who are not gamers themselves and have struggled to work for a company that makes games while a whole set of people who love the gaming industry are choosing product companies that deal with making games.

### **Dissatisfied with the role**

Now let's look at Bob, who works for a retail company ABC. Innovation has always been something Bob is passionate about. He joined ABC to build unique products that help innovate on behalf of the customers. During two years in the role, Bob had fewer opportunities to innovate, plus was unable to get dedicated time carved out to focus on innovation or building products they are passionate about. Bob meets his friend over a coffee who sells about the company they work for, where the company gives a week every quarter to each employee as innovation week to work on projects they are passionate about and bring them to leadership. Bob is convinced and decides to explore opportunities in the company his friend works for. As a result, Bob decides to leave the ABC company.

This is one of the many reasons employees can be dissatisfied in their current role and hence look for opportunities outside the current employer. Some examples would be:

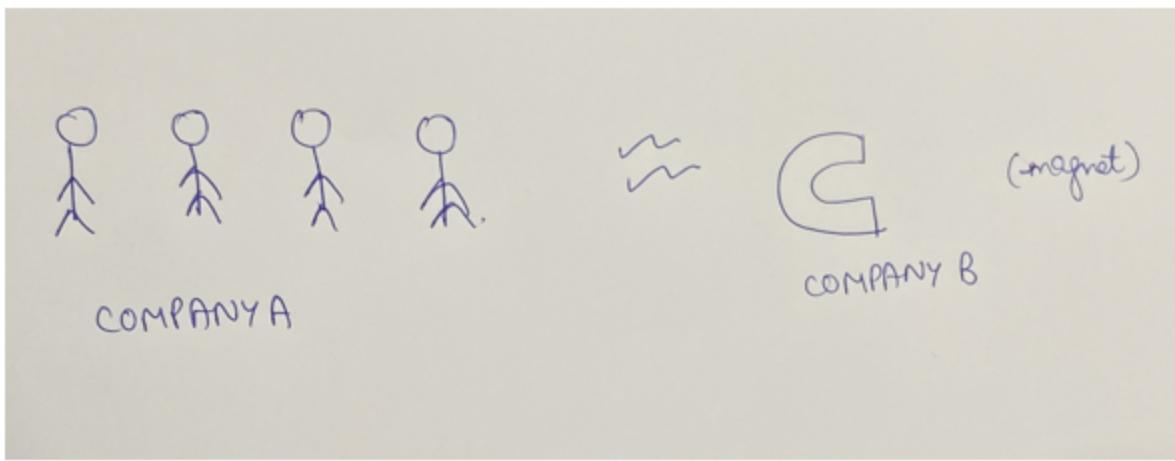
- An insufficient work-life balance (if it's in a person's office- it can be long working hours in the office or long commute hours, for the virtual remote model- it can be virtual meeting fatigue, expectations to be online for a long, etc.), especially if the team is working from different time zones and people are not inclusive of time zones, a shorter commute to the office, etc.
- Someone working on maintenance activities for a long that overshadows feature development (let's say just working on bug fixes instead of any new product development)
- Huge tech debt and operational burden, and hence it's painful to work on the codebase
- Internal mobility is a big no-no where moving within teams or working on other products is not welcomed.
- Longer promotion queue to advance in a career, there may not be enough openings (tiny teams or startups), or the company is heavily hiring from outside instead of internally growing talent.
- Missing learning and growth opportunities- unable to learn new skills due to insufficient resources or time constraints due to project deliverables

So there can be several reasons for individuals to feel unhappy in their current role and explore outside.

## **Compensation and benefits**

Chris works for a retail giant XYZ and has been with the company for five-plus years. During this time, Chris has gone through one round of promotion. He meets his friend at a party, who works for a competitor of XYZ. They catch up over a drink, and Chris realizes he is making much less money in comparison to the tech market's current pay levels. He goes back home and researches using websites like [glassdoor](https://www.glassdoor.com/index.htm)(<https://www.glassdoor.com/index.htm>) and [levels.fyi](https://www.levels.fyi/) (<https://www.levels.fyi/>? [compare=JPMorgan%20Chase,Capital%20One,Google&track=Software%20Engineer](#)) and finds out what people in his role and designation are making based on different locations. Chris decides to prepare for interviews and move to another company. In this case, the individual actively seeks information by visiting the websites above. Still, another emerging trend is where recruiters are reaching out to employees left and right, poaching and telling them they can make more money and get more perks, as shown in the image. Now it boils down to who doesn't want a combination of good pay and perks at the workplace? Your answer is definitely- I need that high-paying job and perks.

**Figure 8.2 An example illustrating how people are poached from one company to another**



Compensation and benefits also constitute a big reason why people leave their current employers in pursuit of new ones. It also gets tricky if the companies poach each other's employees by offering extra compensation in a hot job market.

## Company culture and policies

Company ABC has been a well-known brand in social networking. Existing employees love their work and the culture. With a recent change in leadership, the new management has introduced new expectations which have been deemed unrealistic by a majority of current employees. These include:

- Mandatory work-from-office policy all five days of the week
- Little to no flexibility to work from home
- Removed the spot bonuses award, which was awarded to employees monthly who went above and beyond in their work
- Lack of career growth and development opportunities
- Lack of collaboration and transparency between team members
- Missing 360 peer feedback to learn and improve
- Incompetent performance management system, especially to handle underperformers
- Ineffective communication with leadership- In scenarios where there is a lack of communication with leadership or the flow of information is

just one directional from top to bottom with a strict hierarchy in place

Will you expect the same set of people to be happy? Probably not! This is a case of introducing unrealistic expectations in a workplace which can be a wide range of expectations. Also, for the same set of expectations, one can be ok with them and ready to go by the flow, while for another employee, it might turn into a deal breaker. For example, if the company introduces a mandated work-from-office policy all five days of the week (especially after how our work lives have changed during pandemic times), someone close to the office is ok with the new policy. Still, someone else who shifted to a further place during the pandemic might not be ok with this back-to-work policy. On similar lines, multiple other factors can be part of company culture or policies that satisfy some but will be problematic for the rest of the folks. Another example is work flexibility, especially for expecting moms or someone who recently gave birth. Is the policy around parental leave fair for both partners? Does the company invest in giving back to the community and promote volunteerism? In such scenarios, you need a company that supports your family and personal needs given the situation you are in, and at the same time, you feel aligned with its mission statement. Hence the company culture plays a huge role in employee satisfaction criteria.

## **Personal reasons**

Personal matters are essential in one's life and directly impact their professional life. Think about one of your team members who is an expectant mother. It is obvious to provide her with the desired support and sometimes accommodate their needs. For example, in one of my previous teams, an expecting mother was supported by a few gestures such as ensuring not to keep meetings late in the evening, accommodating doctors' visits, and sharing the on-call burden with the rest of the team near her delivery date. Now let's say the mother did not feel welcomed and supported by her leadership and team; she might not be happy to work at such a place, leading to attrition. Similarly, several personal reasons can

lead one to take a call to move on from the current employer. Some of these reasons can be:

- Family reasons - Let's say the spouse needs to move from the east coast to the west coast for work, the person also has to change the base to the west coast and hence might have to leave the job in case there is no subsequent office in the new location or the leadership does not support the move. Also, this encompasses scenarios where someone takes a career break to look after children or the elderly at home.
- Higher studies- One wants to pursue higher studies like doing a Master of Prost doctorate and hence would like to go back to academic work for a bit.
- Burnout/Sabbatical - I have seen instances where people might go on a sabbatical or break either because they have an idea and want to try something of their own(entrepreneur) or feel burnt out and need a career break.
- Work stability - In times of a downturn in the economy, especially with the fear of layoffs at the workplace, people might be looking for a more stable job. For example, someone on a work visa in a different country might value immigration and visa stability over a high perk job. This is one way to think about someone looking for stability in their role and changing a job, looking for more job security.
- Retiring - Someone is at a point in life where they would like to retire and spend time with their loved ones.
- Change the sector completely - Let's say someone currently works in the retail sector but has always been fascinated by the gaming industry. They get introduced to a new role in a gaming company and are intrigued to join it, hence willing to leave their current role and ultimately would like to change the sector they work in. Similarly, someone can be tired of working in giant companies and wants to experience the startup world.

Above are some common personal reasons that can trigger someone to seek new opportunities and change the job. As an engineering leader, it is on us to understand the reasoning and support our team members in ways under our control.

## **8.2.2 Involuntary reasons**

Now let's look at some of the reasons that might not directly be in one's current situation and can lead to involuntary attrition in the company.

### **Layoffs**

In times of downturn and recession, layoffs are a common phenomenon for companies to balance the operational cost and the revenue they make. Especially with the pandemic, several companies went the layoffs route to adjust the employee base to meet the current needs of their customers and reduce the hypergrowth and hiring they did. Layoffs are almost always financial decisions. Someone looks at the books for the year and decides they aren't going to make EBITDA (Earnings before interest, taxes, depreciation, and amortization, which is used to measure core corporate profitability). Then the first (easiest) reaction is to cut staff. Stock market panic and anxiety are good examples of recent layoffs too. Other times, layoffs are done for more reasonable purposes: a division is shutting down because the customer base is gone, the company is restructuring, or the product is no longer viable. In these cases, it makes much more sense, but all are related to the financial stability of the company.

When layoffs happen, various reasons can come into play to decide who to let go- performance, entire organization shutdowns, role terminations, etc. Layoffs are usually not in the employee's control but can lead to involuntary attrition.

Another aspect of layoffs that is not 'true' but leads to unregretted attrition is when someone is managed out due to underperformance. This is common across companies and requires engineering leaders to have crucial conversations. We have covered this in-depth in Chapter 4 of managing performance. In pre-pandemic times, companies like Airbnb, eBay, Yahoo, etc., had to undergo layoffs to aspire for financial stability and manage increased competition to stay in the game. One of the recent examples of layoffs during the pandemic as seen in recent times is when several high-value technology companies- Microsoft, Amazon, Meta, Robinhood, and

many more were prey to over-hiring. Eventually, everything normalized, and then the companies had to take the tough call to downsize themselves to get the finances to a more stable state. This caused several unemployed people to be available for work simultaneously, causing competition.

## **Return to office**

With the pandemic, we all learned a new way of working, yes, working from home and aiming to be productive at a similar level. With things returning to normal, some companies might demand the workforce be back in the office and make it a mandate. This forced return to the office might work for some employees but can be counterproductive for others who have a long commute or have some reasons that don't allow them to be in the office for the expected number of days. Also, as more and more companies offer the hybrid model and remote roles than ever before, more people expect this as a new norm. Such a forced return to work policy can force people to change employers.

Further, the pandemic completely rearranged priorities for many people. With the realization that they could work from anywhere and the long run of COVID, many employees moved away to another state or so far away that it didn't make any sense to come in. When things started to settle down, some companies were surprised that their employees could no longer come back into the office, nor that they wanted to do so. Within my team, I had an employee who moved from the New Jersey area to Philadelphia and increased the commute time from 30 minutes to 2 hours. Unless we wanted to let them go, there was no way to force them to come back to the office, let's say, five days a week. More than just distance, there is commute time (which now takes away from family/home time), high cost of gas, potential exposure to others in the office, and even just religious or political concerns. Forcing someone to return to the office is usually a "you're in, or you're out" type of situation. For those that do hybrid, it is still a challenge to manage it well. For example, if folks are more than an hour away from the office, they may tolerate one to two days a week in the office, but if we push it to three to five days a week, they will likely quit.

## **Acquisition and merger**

Acquisitions or mergers can indeed lead to employees leaving the company. During such organizational changes, employees may experience uncertainty, leadership changes, shifts in job responsibilities, cultural differences, or concerns about job security. These factors can contribute to employees seeking opportunities elsewhere.

## **Competitive market**

A competitive or a 'hot job market,' as we call it, can also lead to significant attrition. Let's say Bob is passively looking for opportunities outside the current employer. Now if the job market is hot, they can interview with several companies and get offers. This also puts the candidate in a solid position to negotiate if they have multiple offers in hand as the companies try to poach each other's employees. With such factors in place, with a mix of voluntary and involuntary reasons, the person is bound to leave the current company.

## **8.3 Impact of Attrition**

Attrition brings a wave of impact to people at all levels of the organization. For bottom-line engineers and managers, attrition brings in more work and stress as they continue to support the business with limited resources. This causes demotivation and burnout in the individuals and more chances of them being unhappy in the role. Not only does the extra work get added up or have to pick up the pieces, but the team can be emotionally affected too. A very senior "key" player can be a substantial negative hit for a team that depends on them. A favorite and loved employee can bring down the emotional space of a group. Demotivation has to be viewed as addressing the objective and the subjective impact. For middle management, this adds pressure as they might be expected to deliver the same amount of work but with restricted resources which can be impractical most of the time. As for the top level, they might be stressed due to losing business as they need help to cope with increasing customer demands, missed opportunities, and

low team morale. Also, the company's reputation gets impacted, making hiring new people and attracting talent challenging.

In competitive markets, people might have more options than usual, and attrition is evident as people move between companies in search of better perks. Well, there is a considerable cost to replace an employee as it's not only about the knowledge loss that happens if someone leaves, relatively the cost of finding replacement talent, hiring, onboarding, and so forth. Also, it can add a temporary dependency where someone in the team is a single point of failure for a while and hence can be counterproductive for the team. Let's look at some of the everyday things in depth that are a direct consequence of attrition at the workplace.

### **8.3.1 Low team morale**

Attrition in the team directly impacts the morale of existing employees. Because an employee has left, the work quality as a team might be impacted. Further, the existing employees will feel less motivated to work, can have a negative outlook towards work in general, impact on overall productivity and performance, and hence have pessimism surrounding them. A team member departs from the team in favor of a better-paying job. Existing employees are bound to get attracted by the idea of a higher-paying job. One might see frequent missing meetings, deadlines, and low engagement among other employees. Also, this can cause a further ripple effect in the team, ultimately leading to low developer velocity.

In one of the teams I had worked with, the company was going through some difficult times and had to let go of several employees. Unfortunately, it included one member of my team. There were two noticeable problems I saw with the existing team members. First was the survivors' guilt- we are safe, but how did they choose this individual and why. This phased off eventually, but most importantly, everyone on the team started to feel insecure about their job security. Also, remember, several employees in the technical sector are employed on a work visa, and for them, job security is even more closely tied to their life stability. So the team morale was

impacted negatively, and struggles were real as most team members were on work visas.

### **8.3.2 Loss of Knowledge**

Every departure in the team causes a loss of knowledge and context to some extent. Imagine a team of five engineers where each one is a subject matter expert in their respective areas. Now, if one leaves, the other needs to build that domain knowledge and ensure everything is well documented for posterity reasons. Hence, touted the idea of avoiding single points of failure in the teams. Loss of knowledge can be detrimental to the team as not only it adds a work burden to the existing workforce but also causes the same work to take longer than expected until the expertise is built. This can often lead to a significant impact on business and not addressing customer needs in a timely manner.

Suppose a company fires nearly half its workforce due to imperfect market conditions. The employees who stay might not know how half their systems work. The only reason they continue to work is that they haven't broken, but the moment they do, it'll be chaos. Loss of knowledge, and the transfer of knowledge, are significant concerns. It's a multiplier too. One person with specific individual knowledge has to transfer it to the team (ideally), so if it requires a week of their time to document and explain to a team of 5, then you have a month's worth of lost time for the team. And how do you determine the most critical information to transfer? It's always a tricky question, left to the departing individual, who is already checking out mentally.

### **8.3.3 High Hiring and backfill cost**

Hiring is a complex and costly process that involves several stages. First, there is the effort and resources required to identify potential candidates. This is followed by setting up interview loops, which demand a significant time commitment from existing employees. After the interviews, there is the debriefing process, sharing the job offer, and conducting promotional calls to highlight the role and the team. Finally, there is the expense of

onboarding the new employee. These steps require considerable time and money and cannot be accomplished by a single person alone.

In competitive markets, hiring can become even more challenging. For instance, if you interview ten candidates, extend offers to four, and only one person accepts, the cost of interviewing the remaining nine candidates and going through the entire hiring process becomes apparent. It can be frustrating to invest so much effort only to have a single successful hire. To make matters worse, imagine if the top candidate declines the offer in favor of another opportunity. This puts you back to square one. At this point, you may face a dilemma: should you extend the offer to the next best candidate, or does that mean you are compromising on your standards and settling for someone less qualified? This situation clearly illustrates the high costs associated with filling vacancies and hiring new employees, especially compared to the benefits of retaining and satisfying existing employees.

We saw above some of the common ways the existing employees, team, and the business gets impacted by attrition in the company. Also, we looked at the massive cost of backfilling an open role (that sometimes can go up to 1.5-2x times the annual salary of an employee), the cost of taking a hit to pay someone better, or investing in current employees makes more sense. Let's do some quick basic math to understand the concept.

Let's say we want to calculate the total cost to backfill an open role for a software engineer, paid a 100k annual salary. (this is just an example to demonstrate the point)

- **Hiring cost** = recruiter cost(let's say the annual salary is 60k and they spent 40 hours (1 week) worth of effort. This makes monthly salary to be  $60k/12 = 5k$ , and for simplicity, let's say each month has 4 weeks, so for 1 week of effort =  $1.25k$ ) + engineering team cost that involves interviews loops, calls with potential candidates to do a sale call(let's say each engineer makes roughly 100k, 4 engineer loops done for 10 potential candidates before finding the prime candidate. Roughly, each engineer spent 2 hours per candidate, so  $4 \text{ engineers} * 10 \text{ candidates} * 2 \text{ hours}$ )

hours = 80 hours = 2 weeks. This makes it  $8.3k/2 = 4.15k$ . This makes a total of  $1.25k + 4.15k = 5.4k$

- **Onboarding cost** = training cost(let's say we used an internal/external trainer for 2 days) [5k] + ramp-up period cost(the engineer was given 60 days to ramp up to the new technology stack and the company culture)[ $100k * (2/12) = 16.6k$ ]. This makes a total of 21.6k.
- **Miscellaneous cost** = cost of putting up job ads, using tools to filter through resumes, etc. = 3k

All the above make it to 30k worth of backfill cost.

Imagine the cost to retain this employee was 10k (maybe to match the new job offer they got or invest in their training). This is one-third the cost of hiring a new employee, let alone the loss of institutional knowledge when an existing employee leaves. Looking at these numbers, investing in the existing employees might make sense. Now that we understand the importance of retaining our existing pool of employees, let's look into some ways one can get ahead of attrition- remember, we said at the start that attrition is inevitable - so you have to plan and get ahead of it!

## 8.4 Getting ahead of attrition

As we have seen from various examples and scenarios above, attrition is not a phenomenon we can eliminate. Instead, the need of the hour is to plan ahead for it. This means we as engineering leaders can take a series of proactive measures to prevent the situation in the first place, reactive measures if we know someone plans to leave and try to retain them, or the last option where the person has made up their mind to leave and you as a leader are now tasked to make that transition as seamless as possible for your team. At the same time, this is an excellent opportunity to share the insights with the HR/benefits team to see what can be done at a company-wide level to avoid attrition. Again, you will face a variety of scenarios where sometimes attrition will be known or suspected, while sometimes it will come to you as a surprise. Let's deep dive into some steps one can take depending on the situation they are tasked with.

### **8.4.1 Proactive measures**

If employees feel safe and valued, they are bound to stay at a company for extended periods of time. Proactive measures help leaders get ahead of attrition and plan for it in advance. This means you, as the leader, continuously try to iterate and look for ways to improve the workplace so people love where they work. Some standard proactive measures include:

#### **Optimize hiring**

Hiring strategies are crucial to tapping the right talent in the industry where the potential candidate has a combination of technical skills plus is a culture fit. By optimizing for hiring, I suggest not treating hiring as just a want to acquire new employees but equally focusing on the retention side of things. Retention is to counter attrition and build a culture where people are happy to work and want to stay and work. One way to promote hiring efforts is to promote referral hiring, which helps spread positive word of mouth.

#### **Promote internal mobility and development programs.**

Promote the development of employees to keep them happy in the short and long term. This encompasses having regular 1 on 1s and career discussions, ensuring the employees are learning and growing in the role by setting some stretch goals for them, keeping in mind their aspirations. This also means having robust onboarding programs for new hires to focus on foundational knowledge and striving to upskill employees continuously. In one of my past companies, we ran an organization-wide initiative to train everyone in introductory machine learning skills. This helped our team members to proactively prepare for what's coming in the future roadmap and also learn a skill to hone their technical skills. On a similar note, we should promote any scenarios where employees would like to change roles, for example, move from a software engineer role to a product manager role or vice versa, or look for opportunities internally within the company. One of my team members was a senior software engineer adept with project delivery and execution skills but had faded interest in just writing code. As

their manager, I proactively identified the situation's root cause and helped them to move to a role that better fits their aspirations (from a senior software engineer to a senior technical program manager). Remember Bob from above, who wanted to innovate and, due to a lack of support on the innovation front, decided to leave the company to pursue another one. Now imagine if the company could identify this gap early and introduce some form of internal mobility for folks looking to work on advanced innovative projects, Bob could have been retained.

## **Recognizing employees**

We looked in detail in the Rewards and Recognition chapter at their importance and how it helps to keep the workforce motivated. Recognizing employees frequently and objectively is essential to appreciate their excellent work, assert positive sentiments, and motivate them.

## **Conducive work culture**

A healthy working environment brings positivity and optimism to the employees. The idea is to ensure the workplace is conducive to the learning and growth of individual employees, drives them to work, and keeps them motivated. Some of the ways to ensure an excellent working atmosphere are:

- Keeping communication channels open and transparent where the employees feel heard and have a bi-directional flow of communication
- Team vision and goals are communicated so people understand what they contribute to and how they are tied to the company mission and vision. This helps promote support and inclusiveness.
- Innovation is emphasized where people can also put forward their ideas which can then be prioritized by product leaders. This can be in the form of hackathons or innovation challenges that lead to a new product offering with buy-in from leadership.
- Build a diverse and inclusive workforce where people feel safe and an outsider lens is appreciated so that people can freely share their ideas

and opinions. People from all walks of life should feel welcomed and valued at the company.

- The flexibility of work- can be work-from-home support, ramp-back programs with limited capacity in the office (let's say 50% capacity for 2 months), or initiatives to support people who are specially-abled, working moms, etc. In the new paradigm for hybrid working mode, the team may follow a no-meeting day used by team members to work on their independent tasks and protect their time spent in meetings.
- Provide training programs for leaders to help them manage performance and train them for career growth discussions. This can include attending, participating, and presenting at tech conferences and conventions.
- All work and no play make Jack a dull boy. We all have heard about this adage, and it stands true for employees at a company. You want to create a culture of bonding where they get opportunities to have fun holiday events and team-building activities to build relationships.

## **Periodic feedback mechanism**

Periodic surveys or some form of feedback mechanism help companies to capture continuous real-time feedback. This helps reduce the time it takes from identifying a problem to proactively working towards addressing the concerns and avoiding sudden surprises. This can encompass issues around developer tools, processes, work-from-home setup and support, career growth opportunities, leadership support, and more. This helps directly engage the employees to understand the top concerns and strive to work towards them.

If you can build a candid relationship with your employees and they find you approachable when they are unhappy and/or thinking of leaving, it can help them do something about the problem in advance. At the same time, you must promise not to misuse this information by ushering them out the door, sharing it with HR, or using it in performance reviews. The idea is to get ahead of their exit.

The insights through the discussions and surveys also help the company prioritize what is most important for a majority of employees and hence direct resources accordingly. In some of the companies I have worked at, such survey results are looked at across all levels and trigger actions by leadership to address the concerns to strive for improvements continuously. Also to call out, such surveys should provide an option to be anonymous so people can freely share their thoughts without fear of management, with the expectation that employees keep professional decorum and ethical boundaries.

## **8.4.2 Reactive measures**

We can end up with scenarios where we sense an upcoming resignation from the employee or sometimes can be surprised. In any of the cases, there can be saveable actions (such as a change of job role to find a better fit, looking into compensation adjustments, etc.), and the rest are unsavable actions (someone has to move as their partner moved across the country). Well, in such scenarios, the idea is to take appropriate measures now that the damage is done and use that as a learning experience for the future to avoid similar situations.

### **Understanding the root cause**

Now, let's say David in the team has decided to move on and resigned. What do you do as their manager?

It is of utmost importance to first understand the root cause of this decision. Is it the type of work they are unhappy with, the flexibility, personal/family reasons, compensation etc? It is also essential to understand if the manager (you) are the reason. As an EM, it is a bitter pill to swallow, but EMs should introspect, retrospect, and think if they are doing something that might cause attrition. Some of the reasons above might be under your control as a manager, and you can try to do something about it. For example, if David is moving on due to the work/project they are in, we as managers could have identified this gap earlier and could have done something about it by

providing them with the proper support or swapping the project for them. But, now that David has decided to move, we can share a few options they have on the table- an opportunity to lead a big upcoming project. But on the other hand, if they are moving due to family reasons- that is not under our control, and we might have to live with it. I once had a team member who left the company for family reasons. His partner had relocated to the West Coast, and the company had no office location on the west coast but was east coast based. This was a typical situation where I, as a manager, had little in my hand and hence wished the employees good luck for future endeavors with the message that if they ever changed their minds and relocated, we are always open doors for them.

### **Last resort to keep the employee**

In such cases, where an employee is considering leaving the company, it becomes crucial for the employer and the employee's manager to make every effort to retain them. This is known as a "dive and save" situation, where you are willing to go the extra mile to keep the employee on board, considering their value to the company. To do this, it is essential to assess whether the employee's decision is final or if there is an opportunity to address their concerns. I had a team member with a Ph.D. in machine learning who had initially joined my team for software development work without any involvement in machine learning. After a year, the employee felt dissatisfied and sought employment elsewhere in a role aligned with their interests. When the employee shared this decision with me, I was determined to support them fully, as they were a high-performing individual, and losing them would be a significant loss for the company. I took the initiative to connect the employee with the Data Science team and facilitated discussions with engineering managers from relevant teams within the company. Our goal was to find a suitable position to leverage their machine learning expertise, especially as the company was building a next-generation machine learning platform. Leveraging my networking skills, we identified a better role aligned with their goals and aspirations. By successfully finding a new home for the employee within the same company, we were able to prevent their departure and retain their skills.

This shows the promotion of internal mobility and highlights the willingness to go above and beyond to retain talented employees in critical situations.

### **8.4.3 Point of no going back**

In the above two sections, we looked at some proactive and reactive measures one can take to help prevent attrition, retain the employee, and find a better match for them within the same company. Now there are situations where the decision to leave the employee is at the point of not going back. This means there is little under your control to help with the attrition. This happened to me when one of my employees got multiple offers from the competitive market, and the pay offered was not at par with what our current company offered. I tried to work with the compensation department and the HR business partner to see what best we could do for the employee, but the efforts were unsuccessful due to the huge difference in the pay scale. This was a situation where we could not offer what the employee was looking for, and hence it was a known fact that the employee was leaving and moving on. So the next instinct was to plan the departure. Let's dive into three steps one can take when faced with situations where an employee is bound to leave.

#### **Action plan for a seamless transition**

As the engineering leader, your primary goal is to make this departure and transition as seamless as possible for your team members and the employee leaving so as not to be disruptive for them. This is done to reduce the overall tech burden in the team as the team suddenly falls short of one less team member, and our goal is to keep the lights on. Your engineer, a subject matter expert on pricing services decides to leave. The pricing service is a tier one service, which is critical for the business. In these situations, you want to ensure the knowledge is somewhat distributed in the team through lunch and learn sessions, technical talks, and robust documentation. This allows them to avoid any single point of failure in the team and ensure a balanced workload. At the same time, you want to ensure the employee

leaves on a good note, as people always remember what it felt like working with someone even years later and may not remember the specifics of a project. Hence, plan for a seamless transition and a good employee experience for someone leaving the company. Remember to write them a nice thank you card signed by the team to appreciate their excellent work and cherish those memories for the future. Treating them with respect and thanks is essential and will go a long way. If nothing else, we can lead by example, and our team will notice how we treat employees, even those who are departing, sending a positive message to the team about the team's culture. In the figure, I share a sample offboarding list that can come handy for a seamless transition.

**Figure 8.3 A sample offboarding checklist from the technical end to use for a seamless transition when an employee departs from the team**

	Action plan	Status
<input type="checkbox"/>	Provide robust documentation on the projects being worked on and their state/outstanding tasks.	
<input type="checkbox"/>	Document the important system designs and the processes. If the documentation already exists, make sure its useful, relevant and up to date	
<input type="checkbox"/>	Hold technical talks or lunch and learn sessions that are recorded and cover the essential details of the systems and the projects worked on in the time at the company	
<input type="checkbox"/>	Do any pair programming sessions with team members for critical components as needed, especially to share troubleshooting mechanisms for common production issues, ensuring they are well documented	
<input type="checkbox"/>	Review any access permissions and revoke all access to critical systems, infrastructure, etc. from the departing employee	
<input type="checkbox"/>	Important contacts for cross-functional partners per project/point of contacts	
<input type="checkbox"/>	Do a manager level exit interview	
<input type="checkbox"/>	Thank you note for the employee from the team	

Every attrition experience teaches us what works well and what might not work well. So use the experience to learn for the future. One can motivate

and engage existing employees through team building activities and being better at communicating the clear vision and mission statement of the team. Ensure to make good use of the one-on-one conversations to better understand the goals and aspirations of the employees so nothing comes as a surprise later.

## **Feedback mechanism**

With every departure, there is an opportunity to learn and collect feedback to iterate on processes for the future. So post departure, ensure to collate the notes, debrief and collect feedback received. This can be through a formal conversation with the employee to understand the root cause of the change and then a follow-up with HR as an exit interview to gather insights. This allows the company to identify any specific trends and patterns that leadership can follow at a broader HR and company level. One such example is a trend of people leaving for compensation reasons. This can be rooted in the fact that the company's current compensation might lack industry standards and hence is losing employees to competitors. Having such a candid feedback mechanism helps the company learn and take action steps to help retain employees. This is also an opportunity to brainstorm ideas with staying employees on retention, hear any concerns, and discuss how the team can work together to pick up the slack from the departing employee. With better working conditions and lucrative perks across the industry, job hopping is on the trend. Attrition is painful for any organization that goes through it. Small measures can be taken to scale an organization and keep up with attrition. Focussing on the right things and using such experiences to learn and iterate can create a positive work environment and attract talent to the organization.

## **What do other leaders have to say about it:**

“In some kind of dystopian ideal world, everyone is fungible, and attrition leads us simply to rearrange our “resources” (team members) according to project priorities and meanwhile to hire back a new resource. But in reality, everyone is unique, i.e., not fungible; they have different interests and skills

gained from different experiences over different careers. This means that we can't quite so easily replace someone.

We expect some constant (hopefully low) attrition rate and have enough of a pipeline of team members moving up in tenure and expertise such that when folks leave, others can fill in (temporarily or permanently). I think it's rare that we have that pipeline fully fleshed out, but we can try to mitigate it through documentation and other knowledge sharing. All the while, we should be training our team up”

**~ Richard Frank, Senior Software Engineer(ex-manager), Two Sigma | (ex)Robinhood**

Some reasons for attrition - Growth mismatch: Companies need to grow at some pace for their employees to feel like they are also growing (e.g., there are professional opportunities to grow, like promotions, more responsibilities, etc). If there is a mismatch between the company growth rate and the employee growth rate, then attrition is expected. Systematic compensation problems: If lots of people are leaving due to compensation problems, then we need to fix that. I've fortunately never been in this situation. One-off cases: It's natural for people to pursue other opportunities - hopefully because they are running to something new, not running away from their current situation (if so, the company needs to examine what happened). In all cases, it's important to maintain trust with the remaining team. Share what you can share about the attrition, explain the backfill plans, have 1:1s with anyone you're worried about, etc. In terms of getting ahead of attrition, assuming compensation is not the issue; it's all about matching what an employee wants (their goals, what fulfills them, etc.) with the responsibilities they are being given. An EM should know what their teammates' goals are and ensure (as much as possible) that they are being met - or else being upfront with them and sharing how they might be met later on.”

**~ Jean Bredeche, Head of Engineering at Patch, (ex) Robinhood, Quantopian, Hubspot**

“Very recently, I had an employee who decided to put down papers primarily because of pay differences. I started by showing him the career path we had planned for him and the opportunities he would have if he continued with the team. I also shared his path toward becoming a senior and how that would take him closer to the offer he had from outside. Even though he still took the outside opportunity, that showed how much we cared for our employees, keeping an option of returnship.”

**~ Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

“The first time someone on my team told me they were leaving, they were visibly emotional about it. I imagine feeling guilty and potentially uncertain about my reaction - it was their first time leaving a job. Given my own short tenure in my role, I felt somewhat emotional myself. I felt like I’d let them down, that their leaving was my failure, even though they were leaving for a role more aligned with their interests. They had originally joined our company to pursue a particular focus (NLP), but our team did not move in that direction. My next step was to discuss with my manager how to proceed. We talked about immediate tactics (which I don’t recall much anymore), and he reassured me that I wasn’t the cause and reminded me of the inevitability of employees changing companies. I remember walking to calm down and letting the information sink in.”

**~ Richard Frank, Senior Software Engineer(ex-manager), Two Sigma | (ex)Robinhood**

“My philosophy to handle attrition is that the personal relationship impact of attrition is unavoidable. People will leave what others like, and you can do nothing about it. The best way I have found to manage this is to make sure that when someone does leave, they aren’t also leaving the team with a skill gap. Cross-train and remove skill silos to reduce the team risk when someone does leave proactively.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“To handle attrition: Ensure the team is motivated and has enough challenges and consistent opportunities to learn & grow, Work on building the bench strength and encourage knowledge sharing, Employees often want to try new things, and as EM, it’s important to find those opportunities as short-term/long-term assignments even if they are outside of your org.”

**~ Saurabh Gandhi | Sr. Director Software Development Audible | Amazon | Ex-Amex**

“Always maintain an open and honest line of communication with your employees about how they're doing, whether they are happy in their role, and whether they feel respected and appreciated. Encourage them to come to you FIRST when they are unhappy so that you can work with them to either improve their condition, transition to a new role elsewhere in the company or work towards helping them find their happy place somewhere else. An EM's first response to such a conversation should be, "How can I help?"

**~ Bruce Bergman, Manager at Lytx**

## **8.5 Stop & Think: Practice questions**

1. Do I feel motivated to go to work? Does my company promote innovation?
2. What top three things are essential for me at the workplace?
3. Does my company value giving recognition for good work?
4. Is anyone other than myself involved in the career development process for me?
5. Am I aligned with the mission and vision of the company I work for?
6. Do I feel supported by my leadership and the company culture?

## **8.6 Summary**

- Attrition is the reduction in the workforce, which can occur due to reasons like the employer's inability to retain talent, personal

circumstances, or involuntary factors like layoffs or performance issues.

- When an employee departs a company, the cost in terms of time, resources, and efforts it takes to backfill the role is a rather painstaking process involving people at all levels, such as recruiting, engineering, leadership, and the customers and project deliverables.
- People move on due to two major categories of reasons:
  - Voluntary reasons - This can encompass reasons such as the employee is not aligned with the mission statement of the company, is currently dissatisfied in their role(e.g., looking for an opportunity to innovate but doesn't get that), compensation reasons, company culture is not conducive for growth or family reasons.
  - Involuntary Reasons - This happens when the reason to move on is beyond the immediate control of the employee and includes cases like layoffs, underperforming employees let go, forced return to the office that does not suit the employee, merger or acquisition, or a competitive job market.
- Attrition negatively affects employee morale. It also results in a loss of contextual knowledge and increases the burden of technical operations on the existing employees temporarily. Additionally, the cost and effort involved in hiring and filling the vacant role can disrupt the team's immediate plans.
- It is the need of the hour for engineering leaders to get ahead of attrition by taking measures such as:
  - Proactive measures include optimizing the hiring process, creating career development opportunities, promoting internal mobility, fostering a reward and recognition culture, establishing a supportive work environment where employees feel safe and valued, and implementing regular feedback mechanisms to prevent unexpected issues.
  - Reactive measures: If informed about an employee's decision to leave, it is important to undertake specific measures. This includes identifying the root cause and, if necessary, discussing compensation concerns with HR. Additionally, exploring

alternative roles within the company that align with the employee's skills and interests can help retain them.

- Point of no going back- When an employee is resolute in their decision to leave, it is important for the leader to prioritize a smooth transition for both the departing employee and the remaining team members. It is also valuable to view this situation as a learning opportunity, identifying any gaps that can be addressed within the team.
- Small measures shared above can help an organization to scale and keep up with attrition.

# 9 Working with cross-functional partners

"Talent wins games, but teamwork and intelligence win championships."

– Michael Jordan

## This chapter covers

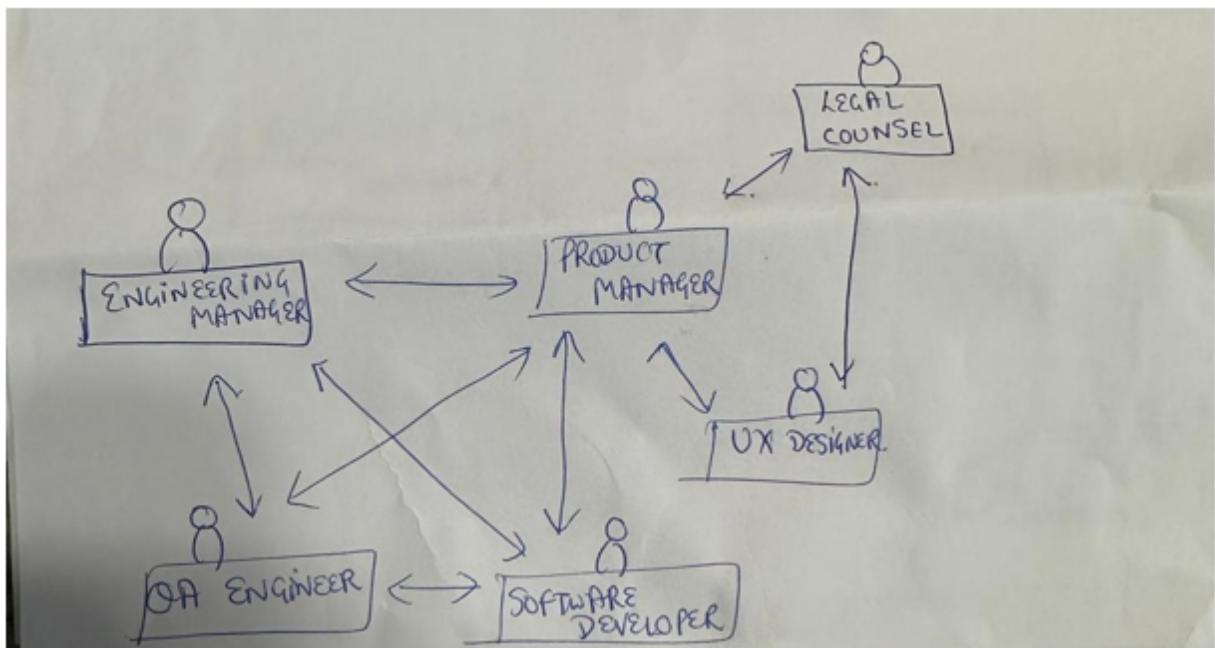
- Importance of effective cross-functional collaboration
- Common challenges faced while working in a cross-functional setting
- Effectively collaborate with cross-functional teams and partners
- Effective communication in an excellent collaboration experience

David, an engineering manager (EM) at an imaginary company Musica, helps support music recommendation platforms at an audio streaming company. In the past year, there has been a more significant push to embed music recommendations in different parts of the company's products. Hence, David's team is involved in a lot of different projects which require cross-functional collaboration. Of late, his team's productivity has been trending down. Most of the projects are at risk of missing their initial set deadline. Based on feedback from the team, almost all members are struggling with ineffective cross-functional collaboration. How do you approach this problem?

Cross-functional collaboration is the process of multiple teams and functions within a company working together towards a common goal. As an EM, mastering effective cross-functional collaboration is crucial for success. Different teams specialize in various aspects of work, such as product management, engineering, design, quality assurance, and legal. When these teams come together with a shared mission, it leads to successful collaboration. Collaboration across teams and functions is illustrated in the image below, where different functions in the company (such as PM, EM, UX, QA, developers, and legal) closely collaborate to execute projects. Product teams understand the product needs and create roadmaps, while engineering builds the product based on these

requirements. Designers contribute their expertise in specific areas, and quality assurance engineers focus on ensuring product quality through testing. When these diverse players unite as a team, it results in effective cross-functional collaboration. Projects requiring cross-team and cross-functional collaboration offer opportunities to leverage the expertise of different teams, leading to the creation of outstanding product experiences. They also facilitate networking, allowing you and your team to understand the company strategy, influence future strategies, and identify and address relevant problem statements. Cross-functional teams are typically formed for significant company projects or high-priority initiatives. Embracing such opportunities enables you to learn and grow as an engineering leader.

**Figure 9.1 Sample interdependency of different cross-functional partners**



Ineffective cross-team collaboration is one of the most common issues you would encounter as an EM. In a [study](https://hbr.org/2015/06/75-of-cross-functional-teams-are-dysfunctional) (<https://hbr.org/2015/06/75-of-cross-functional-teams-are-dysfunctional>) published in Harvard Business Review, it was found that nearly 75% of cross-functional teams are dysfunctional. Inefficiencies in such a collaboration can be stemmed from many different factors, one of the most important, however, is the leadership's inability to

follow a structured approach to solving problems. Such teams often lack guidance and adequate and precise governance around operations, and having different stakeholders muddies the ownership of the entire system, ultimately leading to less accountability. At the same time, having strong leadership alone is nowhere close enough to trying to solve this problem. There are a multitude of reasons why cross-team collaboration could be hard. For different teams to work together and collaborate as efficiently as they would within their own team requires meticulous planning and careful oversight. Let's look into some common issues which can hamper effective cross-collaboration.

## 9.1 Common challenges

Imagine you're working on a greenfield development for your product that involves completely redesigning the application's profile page. You need to collaborate with various individuals to bring this project to fruition. In this case, you collaborate with the marketing team to grasp their requirements for developing the new profile page. The UX team handles the design aspect, while your engineers are responsible for implementing it. However, as the feature nears completion, the marketing team conducts a thorough analysis and uncovers crucial unanswered questions that must be addressed before launching the new page.

Consequently, you postpone the launch and attempt to contact the product manager (PM) to align on the next steps. Unfortunately, the PM is extremely busy with meetings and planning for another project. The PM and the engineering team have conflicting immediate priorities, resulting in dysfunctional cross-functional collaboration. Finally, after a week, you get in touch with the PM. However, no definitive next steps are established due to their limited availability. As a result, the project feature remains uncertain while the cross-functional collaboration endeavors to resolve the next steps.

Instances like these are all too familiar in the technical industry and are examples of challenges faced when working with cross-functional partners. Setting aside office politics, most people approach these projects genuinely intending to help and contribute. Nonetheless, the required time

commitments from everyone for the successful delivery of the project are lacking due to the competing priorities of different individuals. Individuals are either overloaded with work or allocated to multiple tasks, creating challenging situations. The example scenario presented here is one of many difficulties hindering efficient collaboration with business partners. As an EM responsible for managing cross-functional collaboration, you will encounter this problem in a recurring pattern. Let's look at some reasons which make cross-functional collaboration inefficient at different levels.

### **9.1.1 Competing priorities across different teams**

Competing priorities across different teams can significantly hinder effective collaboration in cross-functional projects. An example is the delay in delivering certain product features caused by competing priorities across various organizational teams. Each team typically has its own set of goals that they strive to achieve. However, when working on projects that involve multiple cross-functional partners, these individual team goals often need to align better with the overall project goal. To illustrate this further, let's consider a case study. In a previous project, my team focused on improving customer engagement metrics for a specific surface on our mobile app. Enhancing customer engagement was a shared goal among several cross-functional partners, including the product manager, other relevant teams, and the UX team collaborating with us. However, the core infrastructure team involved in the project did not prioritize customer engagement as much. Although their infrastructure work aimed to enhance the overall customer experience, their team had competing priorities as they were responsible for various other features requested by multiple product teams.

Consequently, it was challenging to directly align the customer engagement goal with the core infrastructure team's priorities. We had to constantly work with the core infrastructure leadership team to emphasize the support required for the project's delivery. Unfortunately, the investment from the core infrastructure team fell short. Our team made the best effort with limited support. Still, midway through the project implementation, the core infrastructure team was assigned another high-priority task, leading to further divestment from our project and causing delays. We had to move

resources around, cut short on requirements and negotiate the end product with customers. Ultimately, the project was successful, but it was completed four months behind schedule. Engineers working on the project faced additional frustrations and obstacles due to dependencies and inadequate support.

This example highlights how misaligned goals can disrupt cross-functional collaboration. Another typical cross-functional case includes platform teams with other teams depending on them. When abstractions regarding what goes into the platform are not done correctly, it causes trouble as teams start pulling the platform in conflicting directions. Therefore, it is crucial to identify such risks early in the project and work with leadership to clarify critical strategic goals, as well as the priorities and processes required to achieve them. This ensures that each relevant team is adequately involved. In some cases, it may be necessary to collaborate with the leadership chain of dependent teams to create awareness and alignment and secure the necessary investments, especially when project goals do not directly align with those teams' goals. Establishing a consistent company-wide prioritization of projects can help align all teams and foster collaboration. Competing goals are a challenge for them to handle. But what happens if the goals of the project itself are ill-defined? Let's look at the next section to explore this further.

### **9.1.2 Ill-Defined Goals**

Another common reason why cross-functional collaboration becomes ineffective is ill-defined goals. Sometimes the teams are tasked with fuzzy projects and target a general problem statement that doesn't translate easily into concrete problem statements. The team may not be given concrete metrics to improve or may fail to define those metrics quickly when the overall problem statement is presented. Furthermore, there may not be clarity on how much investment to make to solve this problem and how quickly they want to solve it. This often leads to teams running in circles, trying to chase a problem statement they don't fully understand. Consider a scenario where David, an EM of a small social company, manages a team of 8 engineers. On one fine day, his manager points out that the customer

sentiment on the website seems to be deteriorating and the team should focus on fixing that. He also gets a product manager and two other team's EM in the same room as David and presents this problem statement. David is assigned as the primary point of contact for this problem. He looks at the problem statement later that day. He starts to realize that the problem requires far more significant support and investigation than anyone had anticipated, provided there are several avenues of the app where customers engage, and determining the customer sentiment holistically will require a massive undertaking. This is an example of a reasonably ill-defined goal.

Nothing concrete backs the problem statement, and the project seems to have been prioritized in an ad-hoc manner. There is no investment from any data science or data engineering teams, which can first determine how to measure customer sentiments. There is no clarity of where the negative sentiments are coming from. Is it how people converse over the app, or is it because of the customer reviews for products sold through the app or some other mechanism? Anticipating quick results in a project like this without adequate investment is an easy shortcut to failure. It is hard for David to progress here without high data quality to analyze the problem. Product managers can also only do so much if they don't have any adequate data to back these product decisions. Therefore, it is essential to clarify what we want to achieve with a project in as great depth as possible and strategically determine which teams would be required at the base minimum to get the ball rolling. Clarity on the project is also sustained through effective coordination and communication. But when working with cross-functional partners, this communication and coordination can also become a soft under-belly for the project.

### **9.1.3 Inefficient coordination and communication**

Coordination and communication breakdowns are everyday happenings in the delivery of any project. At the same time, they are far more likely to happen for a project that requires cross-functional collaboration. Many factors can cause communication breakdowns, but each individual or team involved in the project may have different communication protocols. Some teams may be used to a weekly standup cadence, whereas other teams might

have daily standups and would require far more frequent check-ins. At the same time, it is possible that people working on specific aspects of the projects just don't know who they need to communicate with when posting project updates. Many times in different cross-functional teams, engineers don't provide regular updates or visibility to different stakeholders just because they are not used to the process. This is a common problem that you would face, especially after a series of processes have been in place for some time. This can lead to various communication breakdowns, ultimately leading to inefficient collaborations.

Communication breakdowns can occur due to personal reasons, like employees' resistance to adapting to new working methods. For instance, a team used to weekly stand-ups may face challenges when transitioning to daily standups for a project that demands cross-functional collaboration. While pushback and concerns are valid, ineffective collaboration may arise. Adjusting the standup frequency based on the project's urgency, such as shifting to daily standups during critical phases, can address this issue. Such situations illustrate how even minor changes can impact communication in projects requiring robust cross-partner collaborations. Communication is also crucial to clarify what role someone plays in the project and what responsibilities come with it. Let's say the various parties involved have agreed on the communication protocol and cadence but are still unable to deliver projects in a timely manner. What if there is a lack of clarity in the roles and responsibilities? Let's look at this further.

#### **9.1.4 Unclear roles and responsibilities**

Having unclear roles and responsibilities is another reason which can lead to a breakdown in effective cross-partner collaboration. Let's first look at an example scenario to increase our understanding of this further. Rajesh, an EM for the growth team, is actively involved in a project focusing on increasing customer growth. More specifically, their team is responsible for building the machine learning infrastructure required to provide better recommendations to the users when customers are browsing the application. The hypothesis backing the project was that if it is easier for customers to find new content, they are more likely to sign up for the membership. One

of the critical collaborators for this project is Ian's team which supports building the model required for different ranking and recommendation use cases. At the start of the project, Rajesh's team and Ian's team decide to use an existing framework to build their machine learning stack. The research scientists will be responsible for the machine learning part, and Rajesh's team will be building the ranking APIs that will be used by the app.

Towards the second half of the execution of the project, Rajesh's team realized that certain aspects of setting up a machine learning model to work in production were done neither by his team nor by Ian's team. Both teams think it's the other team's responsibility to work on that aspect of the project. This caused a lot of churn for the project as neither team carved out bandwidth for that aspect. This directly resulted in the project's delay and ultimately was a missed opportunity to improve customer experience sooner than later. So what was the problem here? The roles and responsibilities of different project teams were vague and unclear. Had the teams thought of this at the planning stage, this could have been quickly sorted out, and adequate funding for the project would have been secured easily. Realizing this quite late in the project would mean that one of the teams would need to return on some of their prior commitments to support this project instead. Therefore, having clearly defined roles and responsibilities is extremely important for the execution of projects where different parties are involved. Another essential factor that's important for the successful delivery of the project is managing conflicts. Let's explore that further.

### **9.1.5 Conflicts**

Conflicts occur when the parties involved are not aligned and disagree to come to a common ground. It can be caused by a variety of factors. Regardless of the root cause, conflicts are sure to put the project at risk no matter the situation. Let's look at a few different scenarios of conflicts to get more clarity. Healthy conflict can lead to a lot of progress for all parties involved. But at the same time, too much conflict can lead to adverse working conditions. An overly competitive nature can be seen at various times and various different levels. Some technical leaders may get more competitive on who gets credit for what, or in another scenario, senior

engineers working on the project may want to be perceived as the tech lead. Different teams may want to work on some of the most critical components of the project which give them the most recognition. At the same time, if a project involves various folks of different seniority, power dynamics can also get tricky. Personal conflicts can arise when you work with individuals who don't provide adequate space and time to incorporate feedback and thoughts from everyone involved in the project. Different individuals would react differently to such scenarios. Some may be more aggressive in raising this as a red flag.

In contrast, others may just not feel respected in the workplace and build up animosity towards the individual and the company, which can have a lot of after-effects. Personal conflicts can also occur between team members because of non-professional reasons. Differences in personality, way of working, or general bad blood because of past history contribute to this conflict. In such scenarios resolving conflict is almost always beyond your scope. However, in any of these scenarios, managing conflicts is critical to bringing everyone back to focus on the common goal. Note the characteristics described above are destructive behaviors that any good EM would not display or have the urge to avoid. If you haven't already, Refer to Chapter 1, where we talk about the traits of a good EM.

Conflicts can arise not only on the people's front but also on the technical front. In one of our projects, my team collaborated closely with another team to solve a business problem, requiring a technical design that integrated components from both teams. However, the two senior engineers, one from each team, had conflicting ideas on how to design the system. Their disagreement led to a prolonged back-and-forth with no real progress. While it's essential to strive for consensus, disagreements can occur due to individual solid preferences. In this situation, needing a designated authority figure to make the final call and unblock the issue exacerbated the conflict. When a project involves multiple engineering teams with many senior engineers, starting without a designated authority can lead to challenges in resolving conflicts. Identifying a technical leader midway through the project can cause disruption. To avoid such issues, it's beneficial to designate one engineer as the final authority from the outset of the project,

with the consensus of all team members. This approach simplifies decision-making during conflicts and streamlines project execution. Now, let's look at other reasons which make effective cross-partner collaboration a hard nut to crack.

### **9.1.6 Geographical constraints**

The recent pandemic has redefined how companies and employees think of remote work. More and more employees are opting in to work remotely. Often this leads to employees working in different time zones far from the base location of the team. Many teams may be remote only, so you could have scenarios where individuals from the same team work in different time zones. Even pre-pandemic many companies have offices in different parts of the world. Therefore different projects often require collaboration across teams that are used to working in time zones where the overlap is hard to find. This issue can severely damage how cross-functional collaboration takes place in your team. One of the projects that my team worked on required collaboration from teams in London, India, and Australia. It was relatively easy to find common working time with folks from the London team, but working with teams from India and Australia was a challenge. This often means early or late meetings for our team and the teams from India and Australia. Cross-partner collaboration can get tricky if good working relationships and protocols are not established beforehand. Consider a scenario where all the communication relies only on emails and chats. If an engineer wants to do a technical design session with another team, not having an established process may discourage them from setting up meetings with the other team members at slightly late business hours. If the other team knows what to expect from this team and things are clarified at the beginning, none of this comes as a surprise to anyone.

### **9.1.7 Missing support/underperforming partners**

In many instances, missing support or underperformance by a cross-functional partner becomes a crucial bottleneck for some projects. This gets harder to debug and resolve as you are not the direct manager for those employees. There are many reasons why a person may be underperforming.

They may be going through some personal issues, pursuing other job opportunities, or just being mentally checked out because of various factors. Social loafing, a phenomenon where a person exerts less effort to achieve a goal when they work in a group as compared to when they are working alone, is a genuine problem. It is important to identify under-performance scenarios early and make a plan to address them. Importantly, identifying an underperforming cross-functional partner requires additional validation. You should work with other team members to collect soft feedback to ensure your analysis is correct. Once you have identified underperformance as an issue, providing guidance and strong support is an easy way to bring things back into action. Several immediate actions can be taken to address short-term performance issues. Pointing to internal resources, pointing to some mentors, and working with their manager to set concrete goals are a few ways to solve this problem. Refer back to the chapter on Managing Performance to refresh your memory and dive deeper into this topic. Another reason for underperformance could also be related to having so many moving pieces of the project. Let's discuss this in some detail.

### **9.1.8 Being overwhelmed with work**

Cross-functional (xfn)l collaboration has a lot of fluid parts. For an engineer who prefers to sit and code, managing connections with many different individuals is overwhelming. In one of my teams previously, we had a new engineer join our team. They came from a core infrastructure team background where most of their previous team's work was self-contained. This limited the engineer's opportunity to learn about cross-functional collaboration altogether. The first significant project this engineer picked had a lot of difficulty coming off the ground. PMs weren't getting proper transparency, other engineers were having a hard time understanding the requirements for the project, and UX was nowhere in the picture even though it very much should have been. As we started diving deep into why no progress was being made, the root cause boiled down to the engineer simply being overwhelmed. In such a situation, one could have worked with another senior engineer in the team to provide adequate support to the project to get the ball rolling. This could have been effective in moving the

needle for the project. This project also shows that people may need additional support on projects which require xfn collaboration.

## **9.2 Collaborating with cross-functional teams**

To make cross-functional collaboration work requires different aspects of the system to work harmoniously. Now that we have a clearer idea of some common pitfalls, let's look at some common strategies to mitigate those issues. To begin with, let's discuss the most critical aspect of this process, which is precise goal setting.

### **9.2.1 Clarify and align on goals**

Having too many competing priorities and unclear, misaligned goals are significant factors leading to dysfunction in cross-functional teams. To ensure success, teams must have clarity and specific objectives, as communication might not be as fluent as within immediate teams. When the project scope is broad or goals are ambiguous, concrete results may not be effectively delivered. Therefore, engineering leaders should be responsible for defining well-aligned goals and mission statements to guide the team's efforts.

One common scenario of misaligned goals occurs between user interface (UX) and engineering teams. UX teams often work ahead of engineering in goal setting and planning, which can create challenges in timely delivery. To mitigate this issue, assigning an engineering point of contact to collaborate closely with UX designers can help break down the effort into milestones, allowing the team to work on a minimal viable product (MVP) more promptly. Although the initial UX experience may not be perfect, having a roadmap with concrete milestones and well-defined goals enables better user experience in the long run while also meeting timely business needs and incorporating customer feedback. In summary, clear and well-aligned goals are essential for effective cross-functional team performance. Adapting on the fly can help mitigate misaligned goal issues, enabling

teams to deliver results efficiently and iterate on improvements for a better user experience.

### **Did you know?**

Quoting from [Wikipedia](https://en.wikipedia.org/wiki/Nemawashi) (<https://en.wikipedia.org/wiki/Nemawashi>)- “**Nemawashi** is a [Japanese business](#) informal process of quietly laying the foundation for some proposed change or project by talking to the people concerned and gathering support and feedback before a formal announcement.” This is also another way to ensure the concerned people are in alignment and can help set the foundation before the central formal meeting. This also gives people space to discuss personally if they have more questions and concerns before being part of the broader audience.

Having well-defined milestones and a clear roadmap helps with the immediate delivery of the project. It reassures the people involved in the project that the company is invested in the project and has its priorities well-defined. Having a clear vision, OKRs, KPIs, robust documentation and alignment on communication channels, frequency of communication, usage of project management tools like [JIRA](#) (<https://www.atlassian.com/software/jira>) or other alternatives like [Asana](#) (<https://asana.com/>) / [Trello](#) (<https://trello.com/>), etc. brings more clarity to everyone involved in the project which is necessary to have fluid execution of the project that requires cross-functional collaboration.

### **9.2.2 Trust and Transparency**

Building trust and maintaining transparency is crucial for successful cross-functional project execution, as it involves various processes and working with new people, which can present unique challenges. In such situations, understanding the strengths and opportunities of team members takes time to develop trust.

Trust breakdown in cross-functional projects can occur due to several reasons. Some common factors include:

- Lack of transparency - The biggest reason trust is eroded over time is the lack of transparency. If people involved in the project don't have enough transparency and are just handed over decisions that they have to live with without any background of why those decisions were taken, it can severely impact trust in your team.
- Lack of Communication - Ineffective communication can lead to misunderstandings between different individuals. These misunderstandings can lead to missed deadlines, more stress and ultimately can bring down levels of trust between different individuals.
- Poor work quality - Trust often takes time to build. It is created and maintained by understanding different problem statements, respecting fellow workers, and adhering to expectations and commitments. When it's perceived that you cannot meet your commitments regularly or people start to doubt a person's ability to come through, we start to lose trust in each other.
- Lack of accountability - Another reason people can start to lose trust in each other is a lack of accountability. If there is no one to fall back on, especially when issues arise and a lot of time is spent playing hot potato with the problem statement, leadership's trust in you to deliver on your commitments can start to deteriorate
- Personal and Professional Conflicts - Conflicts make it hard to work with each other. Be it personal animosity, office politics, or playing the blame game when things go bad, all of such scenarios can impact the trust people have in each other as working with each other becomes increasingly complex.

Trust breakdown can occur between different teams due to various reasons. For instance, in a scenario where a new initiative is undertaken to revamp promotions in the application's marketplace, a lack of trust may arise if the point of contact for the project makes decisions independently without incorporating feedback from relevant teams. This can be especially problematic if there is pressure from leadership to support the project without having sufficient input. In such cases, it's essential to assess the level of trust established between the teams working on the project. Are team members on board with all the design decisions that have been taken? Do they feel they have space to raise their concerns? These critical

questions can impact the collaboration's effectiveness in successfully delivering the project.

Transparency plays a vital role in building trust among team members. A culture of transparency needs intentional effort to develop, where openness and honesty are rewarded even in challenging situations. For example, a software engineer like Rishi, who takes ownership of a bug and communicates it transparently to the team, seeking feedback and working towards solutions, exemplifies a healthy engineering team that promotes transparency. In conclusion, trust is a fundamental aspect of successful cross-functional projects. Transparency and open communication are key components in establishing and maintaining trust between teams, enabling effective collaboration and problem-solving. A culture of engagement and collaboration fosters ideas that otherwise might have never seen the day of light. More diverse ideas help create a vision that everyone buys into. At the same time, maintaining a high level of transparency also helps all the team members be informed and, more importantly, understand why certain decisions were taken instead of others in different scenarios.

### **9.2.3 Managing conflicts**

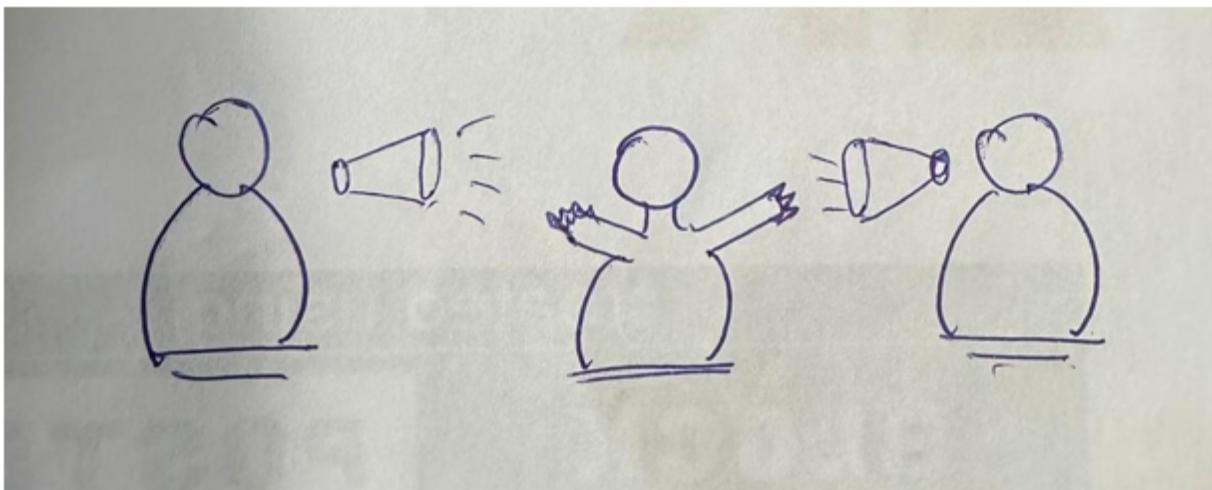
Conflicts are common in the workplace and can occur between individuals or teams. Handling conflicts requires a flexible approach, as one size does not fit all. To successfully collaborate, it's essential to focus on the project's long-term goals and remind team members why they were chosen for the project. A kick-off meeting, led by the EM or PM, can establish the collaboration's direction and shared goals. When conflicts arise, referring back to this initial roadmap can help keep things on track and prevent them from escalating beyond control. Personal conflicts between team members should not hinder productivity. Understanding that being the best of friends is not necessary for efficient work and achieving desired outcomes is important. Instead, focusing on the project's objectives can help overcome personal conflicts and maintain a productive working environment.

“There’s a myth that you have to be best friends to win championships. We only have to have one thing, and that’s respect”

**- Shaquille O'Neal**

Not only would you need to manage conflicts between different individuals, but there are many scenarios where you would have conflicts with other EMs or teams. Having an open conversation to clear the air, understanding where the other person is coming from, and ensuring they have enough space and time to hear their opinions are reasonable first steps. This is true for your own conflicts and helps resolve conflicts between different team members. Consider that you are managing a team of engineers and closely working with 3 other teams to build a new framework. Many senior engineers, including 4 tech leads (one from each team), are involved in the initial design of the system. Two tech leads are highly opinionated about how backend services should interact and have been going back and forth for a week trying to persuade the other to accept their approach. Nobody is ready to move, and both engineers feel their way is the only “right” way. This discussion is starting to get more detrimental than good as it impacts the delivery of the entire framework. How do you go about solving this problem? Talking to both parties separately to understand their viewpoints is a natural starting point. Understanding all viewpoints is critical to future decision-making and helps give a clearer picture of the underlying problem. Bringing both engineers together for another discussion and using a data-driven approach to make decisions are also good follow-ups. Sometimes it also helps if a neutral party provides feedback in such discussions, as shown in the figure.

**Figure 9.2 Helping a neutral party to help bring alignment in times of conflict**



For example, having another senior engineer review different design choices and provide feedback is another strategy that can be used to resolve conflict by keeping the discussion data focused. Irrespective of whichever route the team ends up taking, you should always make sure that all opinions are heard, and everyone feels part of the process and has enough space to provide their opinion. Working towards a consensus and not necessarily having everyone agree on the solution is another great way of tackling this problem. If you have exhausted other avenues to make progress and resistance and conflict are still a blocker, having everyone in a meeting and explaining how we will go with consensus works excellent. Everyone does not have to agree on the path forward, and folks can still retain their opinions and ideas. Still, if everyone agrees that the consensus should be to move toward a specific direction, then at least the deadlock is unblocked. If there are issues later, this can all be revisited.

Conflicts are a tricky beast. They can stem out of nowhere, and often the solution requires a lot of effort. Managing conflicts requires cautious disentanglement of the twisted situation and often requires working on personal issues with individuals. Having an open mind, a healthy discussion environment, objective decision-making, and, most importantly, respecting everyone's opinions can make your life much easier.

#### **9.2.4 Providing extra support when needed**

Projects involving cross-functional partners can present unique challenges, and identifying situations that require extra support demands close observation and effective oversight. For instance, managing a team of 8 engineers, where one remote engineer operates in a different timezone, coordinating recurring meetings for progress updates can be complex. In this scenario, the engineer's availability needs to be considered, and support is required to ensure all team members are aware of the situation and find a suitable meeting time. Providing such support is crucial as remote work can bring challenges, and engineers may hesitate to voice concerns without intervention. Being mindful of individual circumstances when making decisions is essential. While this example illustrates one form of support, additional support can manifest in various ways.

- Bringing in a pinch hitter engineer when there is a rough spot and the team is lagging behind.
- Providing training on a topic they would like to learn about
- Taking the team out for a break when they're stressed and need to step back for a moment
- Just bringing in coffee and donuts to cheer the team and to show appreciation for the hard work they are doing

As discussed earlier, cross-functional collaboration can simply be overwhelming for specific individuals. Identifying individuals suddenly struggling is a key step in mitigating the process. Coaching and mentoring them yourself or by another mentor are good next steps. Extra support will be required by many individuals that you end up working with. Having an eye on the project execution will highlight many, if not all, such scenarios.

### **9.2.5 Open and continuous feedback mechanism**

One of the many challenges that occur in a project that involves cross-functional partners is the expectations that different cross-functional partners may have from the project as well as in their collaboration with you and your team. Different things may be necessary for individuals within the project based on their own situations. For example, the primary focus of a UX team would be to ensure that a robust and well-defined UX is designed

and developed for the feature. In contrast, the engineering teams may be more focused on ensuring the functionality and reliability of the systems they are developing. Within engineering teams, different teams would be focussing on different components. Each team may have specific criteria for evaluating their experience with this project. Therefore it is essential to be open and transparent about the work you and your team have been doing and collect constructive feedback on the overall working experience for your collaborating teams. This process is essential as it highlights inefficiencies in the system early in the stage and allows you to take corrective measures in time and without hampering the delivery of the project. Listing some of the processes you can try in collecting feedback in such scenarios here:

- 1-1 meeting with different stakeholders on a regular cadence to collect their feedback; this helps to identify strengths and opportunities from the team members and in the overall execution process
- Retrospective sessions with the entire cross-functional team at a regular cadence help to identify what went well, what could be improved, and the action items that can be taken to address the gaps.
- Working with other EMs to gather offline feedback on collaboration from their team members
- An anonymous survey questionnaire from immediate team members, which includes specific sections on cross-functional collaboration, allows people to express their views freely.

No one might be perfect. Having an open continuous feedback loop helps ensure individuals and teams are improving on a regular cadence. This allows for open channels of communication where people can share feedback to help each other grow and build a conducive work environment. Additionally, this also feeds and supports transparency. From time to time, if you probe your team members about how things are going and if the general response is conflicting or you get emotional responses, then it's clear that your team is not as open and honest as they should be. Bringing everyone in a room and having a heart-to-heart conversation with the expectation that we all get on the same page about where things are at the

moment can help provide transparency to everyone and foster a culture of building trust.

### **9.2.6 Reasoning about technical work with non-tech partners**

One challenge in a cross-functional project is ensuring all the relevant stakeholders are on board with the project's goals. While many goals are defined on product metrics instead of engineering metrics, ensuring good operation excellence practices are being adhered to is usually in the interest of everybody in the long term. However, as you are laying out the roadmap for a project, there will be many situations in which you would need to load balance time allocation on the project between new product enablement and operational excellence work. Operational excellence encompasses ways to continuously improve the work by focusing on reducing technical debt in the code, increasing code coverage, improving the metrics, monitoring alarms when a service goes through a heavy load, etc. Heavily skewing one way or the other is ultimately detrimental to the long-term growth of the company. Non-technical partners often won't understand why a particular project should be done by themselves. As an EM, it is crucial to clarify why prioritizing operational excellence work is essential and help paint a picture of the work to non-technical folks involved.

Let's take an example to understand this better, suppose you are managing a team that owns a graph database platform. In the last year, your team built a new framework to support stream processing in the platform. This platform has seen massive adoption in a short time and requires keeping up with many newly requested features. With every new customer onboarding, the software setup required one developer two days to create configuration files and hard-code specific values to make the system end-to-end for their use-case. This inefficient way of onboarding was built as a stop-gap solution to launch the MVP. This is a clear tech debt that has just not been prioritized. During quarterly prioritization, you don't see enough time allocated to work on reducing this tech debt which can be a massive win in the long run, especially given the rate of adoption of this event streaming system. How do you work with different stakeholders, especially your product manager, to prioritize this work? One of the ways that can be followed is simplifying the

problem into concrete issues, the increased tech debt has caused, and communicating it.

In such scenarios, you can start by understanding the impact on various fronts.

- Has the platform reliability gone down? If yes, what is the impact that has been caused?
- Has the platform violated any Service Level Agreements (agreements between your team and the team that uses what you build)?
- Direct product impact to customers measured in different product metrics that use this platform
- The increased cost of maintaining the service, both in terms of people and performance
- Short-term cost versus long-term cost analysis

Once you have the data, using it in layman's language (keeping technical jargon aside) to explain to your non-technical partners is a powerful way to resolve gaps. A good EM will be adept at communicating technical concepts to a non-technical audience. This is important as, without this, non-technical folks are constantly anxious about progress and feel left out of the loop. An essential skill to learn is the ability to influence people on all sides by sharing knowledge and making sure everyone is at least comfortable with the concepts and approaches. If they aren't, then bring them in and discuss it. In our example above, suppose you onboarded 40 new clients to the stream processing system; resolving this technical debt would have saved your team  $40*2 = 80$  engineering days which is roughly 4 engineering months (considering 20 working days each month). If you assume an average engineering salary of \$100,000, then this translates to \$400,000 of cost reduction. A simple projection of anticipated new clients can help project the number of engineering hours and ultimately cost your team, and the company can save by prioritizing this work. This will help product managers to weigh the priority of the technical debt in concrete terms with other project features that are being requested. This also helps level the playing field for operational excellence items. While the list of

data points will vary from situation to situation, what's important is to measure impact appropriately and communicate it.

### **Did you know?**

A common strategy for group team development is the [Forming, Storming, Norming, and Performing model](#) first proposed by Bruce Tuckman. This helps build high-performing teams by bringing in the best potential for the team members.

**Figure 9.3 The four stages of the form-storm-norm-perform model for team development**



**Forming** - This is the first stage where the teams are built. This will involve meetings and networking sessions to get to know each other. As the EM, you will collaborate with other cross-functional partners to ensure the team gels up and sets a clear direction and mission for the team. This is the time to move away from small talk and set the foundations for a longer working relationship ahead.

**Storming** - As shared earlier, only some things go as expected, and conflicts are bound to happen. This is what the storming stage is about, where disagreements happen, and the idea is to be empathetic towards each other. It's essential to think from other people's shoes and find common ground.

**Norming** - Once the team has figured out difficult situations of storming, it is reasonable for them to find a working relationship and make that a norm. This does not mean everyone is always thinking the same, what it means is that open discussions and negotiations help them bring to a joint resolution. Getting to normal might not be an easy walk and will take time and energy,

sometimes iterating over a few ways of working to find the right sweet spot for the team.

**Performing-** This is the rewarding stage where the team has matured and has built a synergy of working together to be a high-performing team. As EM, you continue to support your team members and identify areas where any extra help and resources will help them succeed further. This is also the time to recognize the great work of individuals around you to boost that confidence.

The model helps the team build trust and transparency and work together towards a common mission.

We learned key fundamentals to remember as we collaborated with various cross-functional partners. But effective communication is the foundation on which all good projects are built. When it comes to cross-team collaboration, effective communication becomes even more important. Let's learn more about this in the next section.

## **9.3 Effective communication**

Effective communication is the process of exchanging information and ideas with individuals in a clear, concise, and meaningful manner to achieve mutual understanding. Keeping the communication process effective requires careful planning and coordination. Let's look at a few steps which can help with it.

### **9.3.1 Establishing communication framework**

A cross-functional team often involves members from different teams. Each of these teams has different ways of managing communication within their own teams. Some teams prefer the “daily-scrum” method, where each member provides an update on the work they did the day before and what they plan to achieve in the coming day. Some teams prefer the “weekly scrum” method, where the team meets and does the same thing just on a weekly basis. Similarly, your organization may have a set process to provide

upward visibility on the work that your team is doing. This process can look completely different from how other teams manage upward communication. Therefore, it is extremely important for every member of a cross-functional team to be on the same page. To ensure you have a smooth communication framework, at the very least, the following things are required.

- Well-defined project goals, milestones, and deliverables
- Understanding how different dependencies will work together
- Communication channels (Slack/Workchat/Emails/Microsoft Teams, etc.)
- Clearly defined roles and responsibilities of each individual on the project, something like **RACI** model (refer below Did you know section)
- Frequency of project updates for the working team
- Frequency of project updates for relevant stakeholders including your leadership chain
- Tools to be used for Project Management (JIRA/Asana, etc.)

### **Did you know?**

**RACI** ([https://en.wikipedia.org/wiki/Responsibility\\_assignment\\_matrix](https://en.wikipedia.org/wiki/Responsibility_assignment_matrix)) (stands for -Responsible, accountable, consulted, informed) is a Responsibility Assignment Matrix. It is a toolset to help lay clear roles and responsibilities of each member who is part of the project. This helps level set the involvement and the expectations from each member involved. This helps remove the blur lines about the roles and what is expected as part of that role. This also details the activities and project deliverables with owners against each, which is further used for tracking purposes during and after the project launch. This helps set the project up for success. Below is a sample RACI matrix shared to give a visual representation.

**Figure 9.4 A sample RACI matrix**

# RACI Matrix

## Roles and Responsibilities

Deliverables/ Tasks		Status	Project Team			
			Product Manager	Technical Lead	Engineering Manager	Technical Program Manager
						QA Engineer
Phase 1	Deliverable/Task 1		R			
	Deliverable/Task 2		A			
Phase 2	Deliverable/Task 1		A	R	I	I
	Deliverable/Task 2		A	R	I	
Phase 3	Deliverable/Task 1		A	R	I	
	Deliverable/Task 2		C	R	C	
Phase 4	Deliverable/Task 1		A	R		R
	Deliverable/Task 2		A	R	I	R

**R** Responsible  
**A** Accountable  
**C** Consulted  
**I** Informed

Person responsible for delivering the task, assignee  
Person who has final decision-making authority and accountability for completion. Main point of contact  
An adviser, stakeholder, or subject matter expert who is consulted before a decision or action.  
Must be informed after a decision or action.

Initially, team members may have differing views at the beginning of the project. However, accommodating various processes can be challenging. Surveys and polls can help ensure equal participation in decision-making and avoid imposing unpopular procedures. I have used polls in my teams to determine meeting frequency, timing, and communication preferences (online vs. offline). This was effective as the poll results dictated the decisions, and even those unfamiliar with the outcome eventually aligned. Now that you have established a communication channel and aligned on tools and cadence, let's learn how to ensure everyone feels heard.

### 9.3.2 Providing space for opinions and engagement

One of the reasons for cross-functional teams being dysfunctional is the inability of everyone on the team to provide their opinions or influence certain decisions. There are many reasons why this could happen, the most

obvious being that team members who have usually worked with teammates that they know already find themselves in the mix of not only engineers but also other cross-functional partners. Moreover, they may not have interacted with those individuals before. This can have a discouraging impact on members who are not used to working in a cross-functional environment. Ensuring that every team member feels part of the project where they can provide their opinions and feedback on different elements makes everything smoother. This can be done via different mechanisms. Here are some examples of how you can achieve this

- Dedicated communication channels for different purposes - You could have one channel or group just dedicated to technical design discussions and another for general support and Q&A. Encouraging team members to use these channels also helps bring inclusivity in the project and ensures everyone has a sense of belonging for the project
- Team building activities - Such activities can also help with building better habits for the team. At one of my previous employers, it was a common practice for engineering, product, and UX to do “FigJam” sessions where UX created feature mockups for the product feature using [Figma](https://www.figma.com/) (<https://www.figma.com/>). This allowed engineering and product teams to be very involved in the UX design process, especially to provide feedback and their own opinions on the design. Once the mockups were created, user research would follow suit. This simple team-building activity ensured that UX, engineering, and product were always aligned on what needed to be built and how the design would look like.
- Anonymous communication channels - By creating an avenue for raising questions or concerns anonymously, you can encourage your team members to be more involved and ask the hard questions that they otherwise feel discouraged to ask.
- Open project tracker - JIRA or Trello boards where people can see the status of each major task and are encouraged to put in comments and ask questions via that channel
- Bug bounty programs, reward programs for milestones, group lunches, etc. are all great ways to encourage collaboration and engagement

Effective communication is a make or break factor in a lot of cross-functional projects. You need to establish the cadence on how frequently the teams work, how and where issues/blockers are raised, how visibility is provided to relevant stakeholders and the leadership chain, etc. Keeping your communication processes effective requires continuous monitoring. Being on the lookout for communication breakdowns and taking appropriate measures in a timely manner helps alleviate minor hiccups with relative ease.

### **9.3.3 Fostering a culture of recognition**

In the chapter on “Rewards and Recognition,” we have already discussed why recognizing and rewarding good work is important. As an EM, sometimes you tend to get narrow-visioned in recognizing the work of your immediate team members. Cross-functional projects are a great place to reevaluate how you have been recognizing excellent work done by your cross-functional partners. This also means that some of the processes that you have used in your own team to recognize and reward others can be used here as well. None is more important than fostering a culture of recognition where different members of the cross-functional team actively recognize each other's great work. To read or revisit the tools you can use to create this culture, refer to the “Reward & Recognition” chapter. Effective collaboration with our cross-functional partners is a unique skill that lays the foundation of any partnership and successful delivery/launch of a project or product. Learning the skill well can help unite people cohesively to work towards a common goal. Next, let's look at what other leaders have experienced and have to say about it.

#### **What do other leaders have to say about it:**

“One of the major benefits of collaborating with cross-functional partners is the exposure to diverse perspectives. Each team brings unique expertise and knowledge to the table, leading to more comprehensive problem-solving and innovative ideas.”

**~ Sanjay Gupta, General Manager at HCL Technologies**

“The most frequent pushback I see from cross-functional stakeholders is timelines. I keep referring to work visibility, but that has been the best tool to reach cross-functional alignment. By being able to show all upcoming items, ask where they believe their item slots in, and then having the cross-functional stakeholder own some of the negotiations for impact to other stakeholders, you build better ownership from that cross-functional stakeholder and reduce the likelihood of a loud stakeholder flippantly always asserting that his/her work is most important as this now adds personal cost. An example of this was a recent sales conversation where a feature that was considered a “nice to help sell” would have consumed the same resources as a “must-have for the product to be marketable”. By requiring the sales and marketing stakeholders to align and own any possible escalation around lack of alignment, we removed key resources from being in the middle of product discussions.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“Treating partners and your team as the same has proven to help in my teams. Everyone who contributes to the common objective has to be included in all the forums, including team dinners and celebrations.”

**~ Rajakumar Sambasivam, Delivery Manager at Microsoft**

“I believe cross-functional partners are key to our success. Not only can their contributions lead towards us reaching our goal, but they are a great source for ideas and best practices that they observe in other teams, I intentionally try to create opportunities to stay connected with my peers via frequent cadences and corporate social events.”

**~ Sumit Kumar, System EM at Cisco**

“To convince cross-functional partners to prioritize non-product work items like operational excellence over product asks, it is important to highlight the need of allocating capacity on operational excellence for regular maintenance and improve platform capabilities by making XFN partners understand the benefits that customers, product or marketing partners, and

development team will have from such investments. The effort to convince XFN partners to prioritize non-product work reduces once the return on investment is understood.”

~ **Saurabh Gandhi | Sr. Director Software Development Audible | Amazon | Ex-Amex**

## 9.4 Stop & Think: Practice questions

1. Who are my key cross-functional (xfn) partners to work with? How would you describe the working relationship?
2. What are the biggest challenges you face working with them?
3. Think about when you received pushback from your xfn team member? What was the reason for the pushback?
4. How would you convince your xfn team to prioritize non-product work items like operational excellence features over product features in a balanced manner?
5. How do you receive and provide feedback to your xfn partners?
6. Do you see effective communication across all your projects? If you don't feel it's effective, what are your plans to address that?

## 9.5 Summary

- Effective cross-functional collaboration is one of the most important skills to master as an engineering manager.
- Effective cross-functional collaboration helps your team to:
  - Bring the best of different worlds into one project
  - Create the best customer experience possible
  - Provides avenues to both understand and influence strategy and roadmap
  - Expand and build your professional network
- Effective cross-functional collaboration is a hard problem. Studies have shown that roughly 75% of such teams are dysfunctional.
- Cross-team collaboration can be hard because of various reasons. Some of the most important reasons are

- Competing priorities between teams and ill-defined goals
  - Inefficient communication and coordination
  - Unclear roles and responsibilities of different individuals and teams involved in the project
  - Individual and team conflicts
  - Geographical Constraints
  - Missing support from cross-functional or cross-team partners
  - Can simply be overwhelming for some individuals
- Conflicts can arise on different fronts especially when working in large team setups. Unmanaged conflicts are sure to cause disruptions in effective collaboration.
- Different teams have different ways of working. Adapting to a common strategy might be difficult for some in the short term.
- To efficiently manage cross-team collaboration
  - Work with your leadership chain to define clear goals where all collaborators are aligned
  - Bring trust and transparency to the decision-making process by providing a voice to everyone
  - Manage conflicts by working with the individuals or the teams. Ensure everyone's opinions are heard and use neutral parties when needed to take a step forward.
  - Cross-functional projects may require you to provide additional support in different situations. Be ready to anticipate and provide such support when needed
  - Use open channels to gather feedback in a continuous manner
  - Work with non-tech partners to raise important projects which don't directly launch project features but help the company, in the long run,
- Communication breakdowns and inefficient coordination can lead to bad collaboration experience. Effective communication is one of the founding pillars of effective cross-team collaboration.
- To streamline your communication we should establish a communication framework as below:
  - Communicate goals and objectives clearly and transparently
  - Establish a cadence for regular team meetings

- Establish a cadence for regular updates on progress to different stakeholders, including leadership chain
- Define different communication channels (Slack, Teams, Email, Workplace, etc.)
- Communicate different project management tools that will be used throughout the process. Also, ensure to provide a space for voicing opinions and feedback.

# **10 Project management, execution, and delivery**

“Those who plan do better than those who do not plan, even though they rarely stick to their plan.”

~ Winston Churchill

## **This chapter covers**

- The project lifecycle phases and EM involvement
- The four stages of the project lifecycle- pre-planning, planning/project kick-off, execution, and post-execution
- Four stages of understanding the key things to keep in mind

Whether you have experienced dancing firsthand or observed it through videos, you are likely aware of the high level of skill required for this art form. Achieving smooth and fluid movements necessitates not only coordination of your body but also collaboration and coordination with partners if you are not performing solo. This entails extensive practice and rehearsal, spanning days and even months, to master each move and create a plan for executing them flawlessly on the main stage. The timing of the movements is also critical, as missing a beat can disrupt the dance's momentum. Additionally, adapting on the spot is a valuable trait of a skilled dancer, as not everything may go according to plan during a performance.

Project delivery, management, and execution can be compared to a dance, in which the entire team, including the engineering manager, product manager, and technical program manager, are dance partners working to perfect their routine for their next performance. The success of a project hinges on skillful project planning, which involves creating a project plan with clear timelines and priorities, identifying risks and dependencies, setting up sprint ceremonies, providing leadership visibility, and tracking milestones and timelines. If new requirements emerge or things change, the team must remain agile and iterate on the plan to remain flexible while balancing the work expected and resources provided. Success ultimately depends on how

well the team members coordinate and collaborate with one another to achieve their shared goals. In this chapter, we will explore the project lifecycle phases and discuss key considerations to remember as the project progresses.

## 10.1 Project lifecycle phases

The success of a project relies heavily on the planning, execution, and delivery phases, as the effort put into each phase determines the quality of the final product. The process entails:

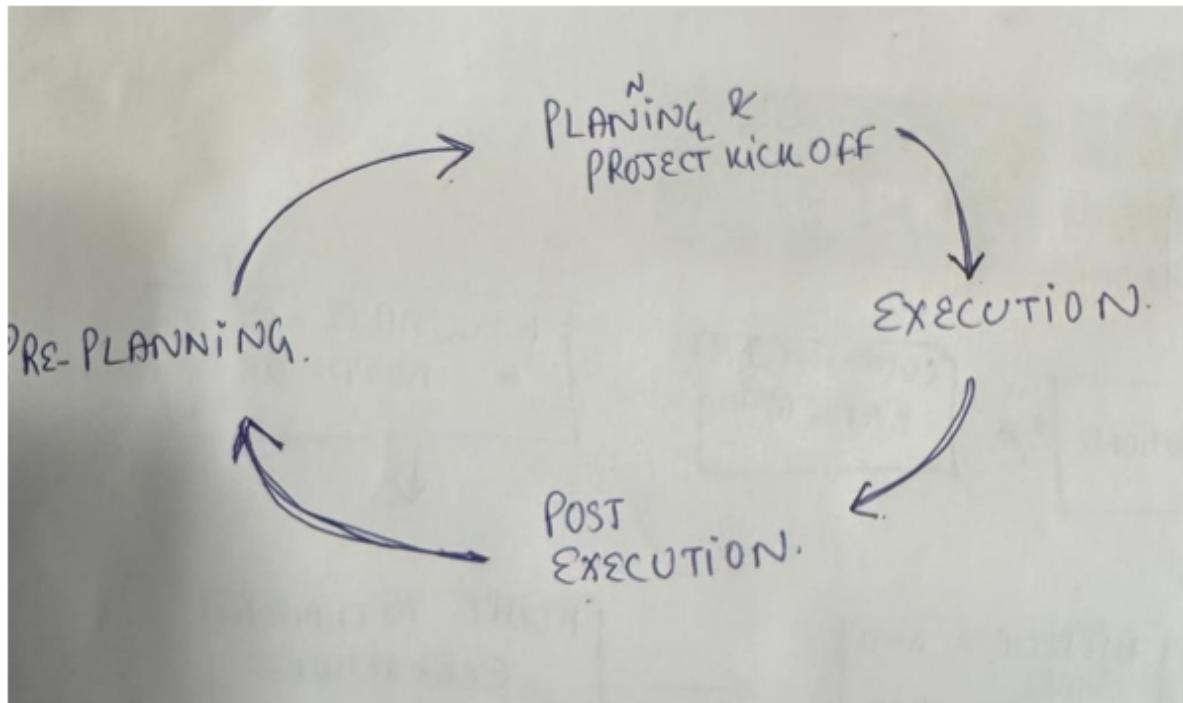
- meticulous planning that takes into account important details
- prioritizing business needs
- defining roles and responsibilities
- establishing a budget, assembling a team
- determining effective communication channels
- training team members
- executing the project
- ensuring operational stability.

### Did you know?

Project portfolio management (PPM) is the concept of managing a set of projects as a collection/portfolio. The focus is to identify the right set of projects, prioritize and ensure the business makes the right investments. It is a step higher to project delivery and execution and encompasses several projects and processes. Each project is part of the portfolio and can be managed from this centralized portfolio. The company's strategy feeds into portfolio management to identify areas for businesses to invest in. Learn more [here](https://www.pmi.org/learning/library/project-portfolio-management-limited-resources-6948) (<https://www.pmi.org/learning/library/project-portfolio-management-limited-resources-6948>).

This requires the collaboration of various stakeholders working towards a shared objective. We will explore the various phases of the project lifecycle as shown in the figure.

**Figure 10.1 The four stages of a project life cycle**



In the subsequent sections, we will delve deeper into each stage of project management and execution.

### **10.1.1 Pre-planning**

Pre-planning in the project lifecycle is the first step to setting the foundation for the project/product launch. This phase involves identifying what the project is, whether it is high in priority or not, looking at the overall vision of the company, defining the project's scope, and identifying the key stakeholders that will be involved in the project. This is the most crucial stage as it is the starting point, and the more efficient the pre-planning is, the smoother the execution of the project. This phase outlines the plan for the project. It helps to identify any bottlenecks or potential obstacles that can come along, a strategy to tackle them and establish a crystal clear, feasible plan for the delivery and execution of the project.

**Did you know?**

Objective and Key Results (OKRs) based project execution is a powerful tool to prioritize and set organizational goals. This helps identify the highest priority for the business by having clear success metrics or results against each objective. Key performance indicators or KPIs are used to measure the success criteria for the projects and keep them objective. So OKRs are like helping set goals and, when combined with project management, help bring the goal to life by focussing on delivery and execution.

Let's take an example of a company XYZ which is in the business of selling toys for infants and toddlers. With increasing competition in the market, the company decides to launch a new line of toys that are accessibility friendly and also cater to kids who are specially-abled. Let's assume this set of new projects to launch this new line of toys is 'Project Accessible.' You are the engineering manager who has been tasked with the launch of this new product line. So as the EM, you start with the pre-planning phase for the project. Let's use this example as we walk through the various phases of the project lifecycle.

## Prioritization

Now that we have identified the new project, it's essential to assess its priority compared to other planned projects for the year. Most companies have a three or five-year vision plan, except startups and small companies still navigating their target market. During the pre-planning phase, teams need to understand where the new project stands in terms of priority among other projects. Prioritization or stack ranking of projects is crucial in setting expectations and ensuring consistency and alignment across teams. Conflicts may arise when teams have different priorities, especially if there is a cross-functional dependency. In such cases, company-wide prioritization or stack ranking can help determine which project takes precedence. If the new project is ranked as the second-highest priority in the company, team B may need to assist in prioritizing the higher-priority project. Overall, prioritization helps align teams toward the company's most important goals and ensures they are working toward them.

So back to our Project Accessible, the team members, including you, get together with the stakeholders to define the prioritization of the project. The product manager helps put numbers around the return on investment. After numerous discussions, the project is decided to fall into the must-have category as being the company's big rock. This ensured all teams were aligned on the prioritization so they could move to the next phase, scope definition.

## Scope definition

In the scope definition phase, the team members take one step forward by getting into defining the scope of the project and documenting it. This starts with defining first of all:

- What are we doing?
- Why are we doing this? Is this going to solve a customer problem or bring in more revenue? What's the value proposition it brings to the table?
- Is there a rigid date this work needs to be done?
- Who's the end customer?
- What is the timeline and the budget?
- How are they going to use this feature or product?
- Will application security review be part of the scope?
- Do we need to do load and performance testing?

This helps get a deeper understanding of what will be included as part of the delivery of this project(in-scope) and what will be considered out of scope to define the boundaries. Also, this is the time when we put forward the success metrics of the project to help understand what defines success for it. This also helps identify any blind spots that can be missed and helps build a shared understanding of the project. By clearly defining the project scope, we can minimize scope creep, reduce project risks, and ensure that the project stays on track and within budget. The [project management triangle](https://en.wikipedia.org/wiki/Project_management_triangle) ([https://en.wikipedia.org/wiki/Project\\_management\\_triangle](https://en.wikipedia.org/wiki/Project_management_triangle)) is an excellent way to understand project constraints and take action accordingly.

Now for Project Accessible, you, as the EM, will take the lead in setting up the meeting to define the scope. You invite all the stakeholders, including product, program, engineering, and business teams. The members go through a long whiteboarding session to break the high-level goal into smaller components to define the scope. Some of the components from the scope definition identified were:

- The product should cater to specially-abled students from age groups 0 to 5
- The product should be lightweight, considering the audience
- Ensuring the product is chemical free and safe for the audience, etc.
- Out of scope: For the first iteration, the product will only be available in the USA and not be available outside of the USA, as there are some foreign certification issues.

The product manager takes the lead in putting numbers around the expected revenue the project will bring. Since another competitor in the market is looking to get into a similar business, the stakeholders decided that the project needs to be launched in the next three months to keep a competitive edge. Since the product will be used by infants and toddlers, testing and security are of prime importance. The next step is to define who is responsible for what as the project execution begins.

## **Roles and responsibilities**

Once the scope is defined, our next step in the project is to define the key players who will be involved and what role they play in the entire process. Having blurry lines of ownership tend to cause confusion and missed requirements. For example, the leadership expects the overall test coverage for core services owned by a team should be above 80%. (Note: Absence of test coverage indicates that there is some issue. However, that doesn't prove that the code is defect-free). They set the goal for the year, assuming the engineering team will get support from the QA organization. Now what if the QA org had a lack of bandwidth? Even if they had the bandwidth, what if there was a blurry line between what engineers would own versus what the QA is responsible for? Who will take care of the load testing? Is security

testing part of this test coverage criteria? This clearly shows how important it is to be explicit with roles and responsibilities and an agreement between all parties. In chapter 9, ‘Working with cross-functional partners,’ we learned about a standard responsibility matrix used- the RACI model, for defining the roles and responsibilities. Below is a sample RACI chart for Project Accessible to help define the key players and their ownership.

**Figure 10.2 A sample RACI matrix for Project Accessible to define the roles and responsibilities**

A	B	C	D	E	F	G	H	I	J	K	L	M
RACI Matrix- Project Accessible												
		Project Team										
Deliverables/ Tasks	Status	Product Manager	Technical Lead	Engineering Manager	Technical Program Manager	QA Engineer						
<b>Phase 1</b>		R										
Kick-off meeting		R										
Project request approved by leadership		R										
Create a project/feature tracker dashboard		C	A	R								
Engineering spike begins		A	R	R								
<b>Phase 2</b>			A	R								
Create a risk mitigation plan			A	R								
Implement deliverables		A	R	A	A	I						
Create status reports for leadership			C	R								
<b>Phase 3</b>												
Demo to customers		R	A	A	A							
Any change management to incorporate		C	C	C	R							
Testing and bug fixing		C				R						
<b>Phase 4</b>												
Live demo sessions for customers		A	R	R	T	I						
Launch of product		A	A	R		I						
Retrospective and lessons learnt		C	C	R	E	C						
Oncall setup		R	A			C						

**ROLES**

- Product Manager
- Technical Lead
- Engineering Manager
- Technical Program Manager
- QA Engineer

**Deliverables/ Tasks**

- Kick-off meeting
- Project request approved by leadership
- Create a project/feature tracker dashboard
- Engineering spike begins
- Create a risk mitigation plan
- Implement deliverables
- Create status reports for leadership
- Demo to customers
- Any change management to incorporate
- Testing and bug fixing
- Live demo sessions for customers
- Launch of product
- Retrospective and lessons learnt
- Oncall setup

**Status**

- Phase 1
- Phase 2
- Phase 3
- Phase 4

**Project Team**

- Product Manager
- Technical Lead
- Engineering Manager
- Technical Program Manager
- QA Engineer

**Legend**

- R Responsible
- A Accountable
- C Consulted
- I Informed

**Definitions**

- Responsible: Person responsible for delivering the task, assignee
- Accountable: Person who has final decision-making authority and accountability for completion.
- Consulted: An adviser, stakeholder, or subject matter expert who is consulted before a decision or action.
- Informed: Must be informed after a decision or action.

Once the RACI chart is ready, ensure to run it by all stakeholders and get a sign-off to ensure everyone is on the same page. Now that we are done with pre-planning activities, let's begin the planning phase and kick off the project.

### 10.1.2 Planning and project kick-off

The second phase in the project lifecycle is the project planning and kick-off stage. In this, the wishlist created from the pre-planning phase is turned into a proper project plan. The planning phase can be done as a program increment (commonly called PI planning process), quarterly(three months), or half of a year fashion(six months). This is where clear milestones are

defined, involving all the team members and creating a working backward plan from the end goal. The idea is to identify the project plan with milestones, build the team to execute it, gear people with the skills they need to execute it, define communication channels, and create a dependency management plan if the project cuts across several teams. This is also where the EM and the program manager would build the project feature tracker that will be the one-stop shop to track project progress and be used to give transparency to leadership. Let's look at the activities involved in this phase in detail below. Depending on the project situation, you can see if they all need to be in serial order or have the opportunity to parallelize them.

## **Budgeting and resourcing**

As you kickstart the project, one of the first steps that come into play is planning for budgeting and resourcing for the project and overall portfolio. Budgeting is thinking about the costs involved with the execution of the project, which includes resources based on expertise, software/hardware expenses, and the time allocation for the project. This helps the leadership plan for the cost and expenditure involved to understand the cost of delivering the project. This is a crucial step to ensure the project is well-funded and does not run out of funds, causing unexpected delays and outcomes. Resourcing involves assigning engineers and cross-functional partners to the project who will be the key players involved in the execution of the project, whether the team is virtual, in-person, or hybrid, facility needs, and the assignment of any peripherals/hardware/equipment needed. For example, if you are in the R & D team, you might need physical devices and be close to the office facilities. This ensures the team is well equipped with the resources needed to deliver the project in the agreed-upon time and cost. Resourcing ensures the team has the right balance of resources needed and that the quality of the project is maintained as promised to stakeholders.

So going back to Project Accessible, being a big rock for the company, the project was well-funded with ample resources to help attain completion in a timely manner. Eight full-stack engineers, product and technical program managers, QA, etc., were on top of the project and assigned full-time. On top of that, the QA team had asked for sample hardware desired for the testing

of betas. Hence a separate budget was allocated to account for that, resulting in a 2 million investment for the next six months.

## Team building

Now that the project has been defined, budgeted, and resources allocated, the idea is to build the team with desired responsibilities assigned. This is where the specific skills and expertise of the team members are taken into account to help get to the finish line. Since the resource allocation and RACI model are already in place, this is the time to ensure team members know each other and understand their roles in completing the project. This also calls for assigning a project technical lead who will collaborate with you from the technical side to deliver the project. You, as EM, assign the engineers at this point to build the team and define clear goals and objectives. While sometimes, the make-up of the team is often done above the EM level. Directors and VPs will typically move chess pieces around to build the team, which you, as EM, might inherit. Read about the [Forming, Storming, Norming and Performing model](#) we learned in chapter 9, working with cross-functional partners to learn about team building.

For project Accessible, we saw eight full-stack engineers allocated as resources; now, your job is to recruit or decide on the eight will to have a balanced team of senior and junior engineers. If the project involves heavy data manipulations and automated jobs, you would like to have one or two data engineer experts to help guide the project from the data side; similarly, if the project is heavy on the backend API developments, you want to keep at least one or two senior backend/complete stack engineers involved, especially during the technical design phase. Sometimes, you might have to hire people externally to fulfill a particular set of skill sets for the project. If that's the case, the team building might take longer as you advertise the job role, interview, and hire the individual. For tips on hiring, revisit the chapter on 'Hiring' to hone your skills. Depending on the size and allocation of the project, the team size and expertise/skills will vary as you build the team. This is also the time to define the decision-making responsibilities and processes for this team; what if a technical conflict occurs between two engineers? You, as EM, proactively communicate that in times of conflict,

the tech lead for the project is to be consulted to help bring the team to alignment and so on. It is on you to help establish a strong team culture for this team to help them succeed, so set the foundation right.

## Communication channels

Effective communication is the key to successful project delivery at the workplace and building relationships with team members. Open communication channels and transparency help build trust and ensure no surprises later in the project development lifecycle. As EM, your job is to define the communication channels that will be useful for the team members to interact and unblock themselves should they hit a roadblock.

Taking the example of Project Accessible, you, as EM, step up to define the communication channels, keeping the cross-functional partners involved in the process.

- **Face-to-face communication** - One of the most common ways, in pre-pandemic times and less common during the pandemic times, face-to-face communication is effective for interaction as one can not only see the facial expressions but rather the body language with it. This also allows for immediate quick feedback loops from the participants. Since your team is primarily located in the same office, you align with the team that significant technical design discussions like whiteboard will happen on a typical day decided by the team in person, face to face. This will be a synchronous communication method for the team.
- **Video conferencing** - Another form of synchronous communication that rose to popularity during the pandemic and work-from-home times is video conferencing. It is beneficial if people are geographically dispersed. One can do a scheduled video conference or start an instant one, depending on the need and urgency of the topic of the discussion. At the same time, I can't stress enough the importance of keeping cameras turned on in video calls to get visual cues and understand facial expressions better.
- **Asynchronous communication** - Of course, there will be times when something is not urgent but needs to be squared out soon. In those

times, one can use async communication that causes less disruption to work, especially if engineers are, let's say, in their focus coding time. For things where the answer can wait a few hours to days, you ensure the team uses the power of async communication, such as emails and messages over chat messengers like Slack or Teams. This also involves documents like one pager written to explain the problem in hand, document technical discussions, and pros and cons using tools like Wiki, Confluence, MediaWiki, and DocuWiki, to name a few. This also helps easily transfer files or documents to be kept for posterity and further discussions.

- **Escalation process-** Yes, this is the time to define the escalation process, as only some things will always go as intended. This helps provide clarity to other team members to know the right people to reach out to in case they have blockers or have difficulty getting things done. This can involve getting a set of approvals for a specific database or application, permissions, or if some team member is underperforming, to name a few. As EM for Project Accessible, you define a straightforward escalation process and also the SLAs to expect as below. Also, you share the expected SLAs at each level; for example, the on-call should respond in 2 business days; if not, the escalation can move to level 2, where EM has 1 business day to respond. Further, if no action is taken, the escalation will move to the Director level to be responded to in 1 business day.

### **Person escalating -> Oncall(Level 1) -> Engineering manager(Level 2) -> Director(Level 3)**

A typical communication channel is also in the issue/ticket tracking system. Jira, for example, is where devs and others (including PMs, QA, etc.) can communicate changes, progress, status, and more. The communication channels help build a process around communication and have action plans in case there are barriers or concerns seen around communication. Also, you want to ensure people adopt the required communication channels, consider the audience when communicating and ensure that the security of the information shared is maintained intact. For example, if the escalation concerns a particular customer, ensure the customer credentials and data are

maintained completely confidential instead of putting it on a public ticket accessible by all. Similarly, ensure we capture all details needed to define the issue, such as:

- Steps to replicate the bug
- Any screenshots or recording
- Environment and build details
- Any other call stack or technical details that might be useful

Hence, clarity in communication is needed to ensure people are on the same page and build trust.

## **Training**

Now that we have the project and the team members assigned, our next objective is to ensure that each individual is well-prepared to contribute their best to the project. This will require some team members to acquire new skills to excel in their roles and boost their confidence. Additionally, we may encounter a lead time for onboarding people, particularly when sending laptops to remote locations, so we must consider this time for training planning.

As an EM (Engineering Manager), you are responsible for identifying skill gaps among the team members and locating the appropriate resources to support their needs. For instance, if the project involves building a new cloud service, it is essential to ensure that the team is proficient in cloud computing and comfortable with implementing services in the cloud. Conducting a skills survey can quickly and efficiently assess the team members' skills and experiences. Once the gaps are identified, the next step is developing a customized plan to address those needs. Remembering that each engineer might have different gaps and requirements is crucial. There is no one-size-fits-all solution for addressing their needs. For instance, if the project involves using React for application development, one engineer might already be an expert in React. At the same time, the other two may require training and resources to become proficient. Moreover, training can extend beyond vocational aspects. For instance, when dealing with a new

team, it might be beneficial to organize sessions led by knowledgeable subject matter experts or technical leads who can guide the team through the system's architecture.

In the example given earlier, it could be helpful to have a session with the product designer, guiding the team on crucial considerations related to cost and materials when building a new toy. Similarly, various training opportunities, such as Project Management Professional (PMP) or the scaled agile framework (SAFE), can be provided to help the team understand different project models like waterfall, agile, and scaled agile, allowing them to determine the best approach given the specific project's circumstances and constraints. Hence, it is vital to create a tailored training plan for each team member, considering their individual skill sets and learning requirements.

For Project Accessible, you, as EM, brainstorm with the team to help them prepare what type of supporting resources will help them succeed. One good outcome was the need for accessibility training for all members involved. Hence, you reach out to the training team in your organization and collect and curate a set of accessibility training to help prepare them. Also, you deep dive into individual strengths and skill sets and identify two engineers who will need additional training on development in the front-end technical stack. Hence, you pair them up with frontend engineers from the sister team plus suggest them some online courses to hone their skills. There is also an upcoming technical conference/webinar in your area on how to develop IOS applications. So you volunteer two engineers to attend that and share the learnings with the rest of the team. These are common ways to help train your team members to prepare for the challenges ahead. In sum, training and learning is a continuous process, especially in the technical sector, where a new language or technical stack might be introduced every few months. Hence, the need is to keep the foundations strong and be willing to learn new things at every step.

## **Project/Feature tracker**

A project or feature tracker helps the team to keep track of all features or items expected and has corresponding owners, status, and priorities against

each. It is like a living document to help ensure each team member is on the same page and acts as a source of truth in case of disagreements. Irrespective of whether the team follows a waterfall or an agile model, the project plan can help with product refinement as you keep the features stack ranked based on the company's priority. This ensures you deliver the highest priority feature to the customer in the spirit of customer obsession. This also helps to keep a superset of items to avoid missing any items later in the project. It is like a powerful organizational tool for the project that helps keep track of the status to give more transparency to leadership in terms of how the team is trending- red (in case there is a blocker or risk), yellow (some impediment but has a path to green) or green (development seems to be on track). Several tools are available in the market, such as Trello, Jira, and Asana, or one can use an in-house tool or as simple as an Excel sheet to create one. One can customize them the way they want, depending on the need of the project and stakeholder expectations. They help with a visual representation to show how far along we are in the project and what seems to be the remaining delta.

### **Did you know?**

A famous approach to tackling requirements prioritization is [the MoSCoW technique](#) ([https://en.wikipedia.org/wiki/MoSCoW\\_method](https://en.wikipedia.org/wiki/MoSCoW_method)). **Moscow** stands for:

M - Must haves, these are non-negotiable product items that are necessary for the launch of the product- e.g., product features, security

S - Should haves, these are essential product items that add value to the product- e.g., performance, low latency, the authentication system

C - Could haves, these are good to-have items, they will also impact the launch- although at a small scale - e.g., feature for comments to add as a journal for users is not necessary for a product launch but can have a small impact and make customers happy

W - Won't have/ Will not have, these are like a wishlist, and the launch doesn't hinge on these items - e.g., features that are not priority 0 or 1 can be implemented in later iterations or versions.

The technique helps teams prioritize what features and launches are most important to the company and helps with stack ranking for the features. This can be used for newer product launches and existing feature enhancements. This is useful in constraints like resources, bandwidth, budget, competing priorities, etc.

Let's look at a sample project tracker for Project Accessible to get a rough idea of how one can make their own project/feature tracker. (I am providing a generic toy example, feel free to customize it to fit your project).

**Figure 10.3 A sample project tracker for Project Accessible to track a list of project items along with the priority, status, and other critical items**

SNO	Project Feature	Goal	Priority	Assigned To	Status	Expected Delivery Date	Dependencies	Risks + Mitigation Plan	Testing complete or not	Comments/Notes
1	Defining the usage or rules of using the toy	The goal of this feature is to ensure the toy defines the usage and precautions of using the toy.	P1	Jack	Green	15-Mar-23	Yes, the feature involves collecting data from upstream accounts services to get customer accounts	The upstream service has an active bug in production where if the customer somehow has two accounts associated with it, it randomly returns the account information for one of them. Mitigation Plan- The team is actively discussing with the upstream accounts team to fix the bug and ensure consistency in the response returned.	Yes	Here we can have a link of related documents
2	Adding texture and patterns to be recognizable by the touch and feel	The toy should have exterior touch and feel so kids can identify the toy and be able to recognize the shape of it.	P1	James	Green	25-Mar-23	Not Applicable	Not Applicable	No	-
3	Program the toy to speak the rules of using the toy	The toy should have an audio transcript to speak the rules and precautions of using the toy.	P2	Emily	Red	1-Apr-23	Yes, the feature needs to integrate with an text to speech converter library in order to add the audio.	Not Applicable	No	Links to text to speech library
4	Inclusion of soft audio to calm the kid	The toy should have a background soothing voice so as to help calm the kids in times of distress.	P3	Sam	Yellow	15-Apr-23	Not Applicable	Not Applicable	No	-

Looking at the project tracker above helps provide a consistent blueprint towards which the entire team works and helps hold owners accountable for the work assigned to them. In addition, it can be used by QA and other

testing persons to help build test cases and verify traceability. Next, let's look at how to manage dependencies as part of the project.

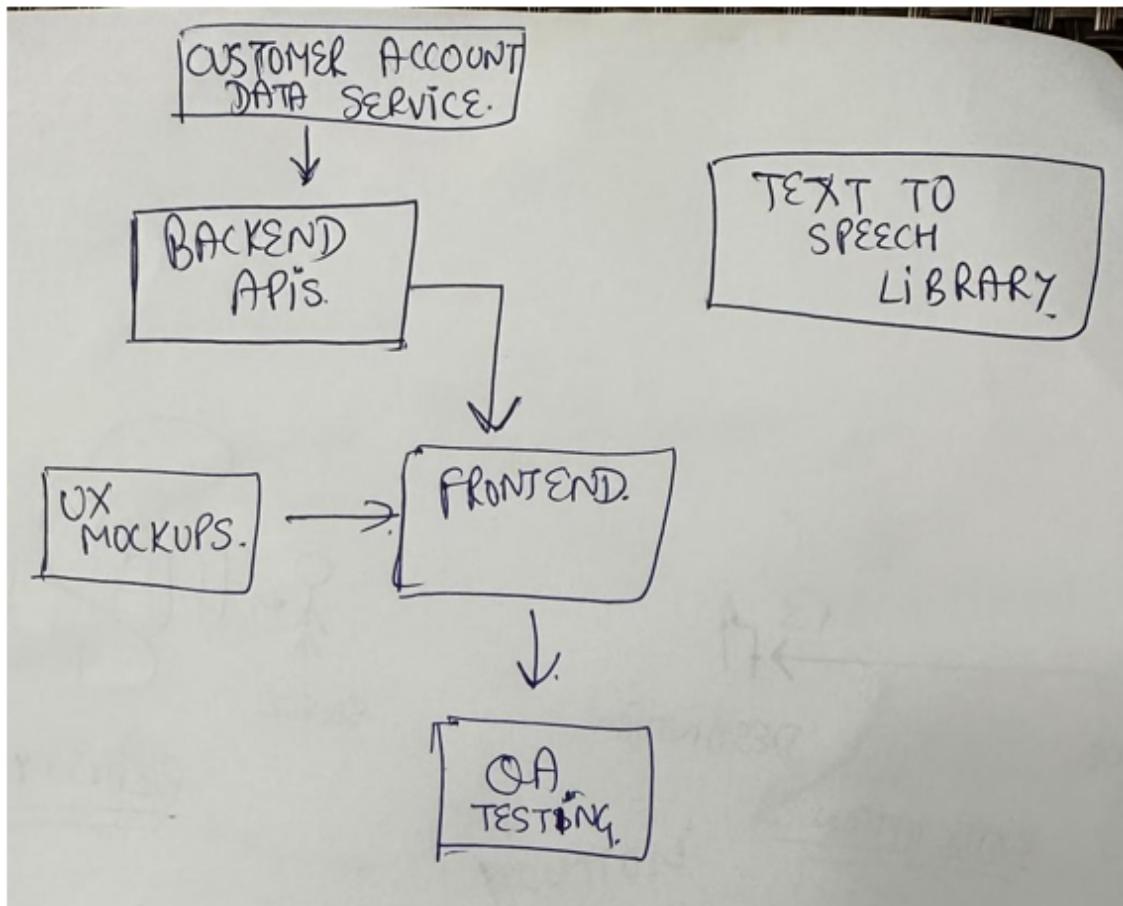
## Dependency management

Dependencies occur when the sequence of activities involved depends on each other in the sense one is dependent on the other. They maintain a relationship that needs to be respected for the successful completion of the project. Dependency management is identifying, organizing, and tracking these dependencies to lay out the sequence of activities. Initially, you start with identifying the dependencies to find the missing pieces, followed by organizing them into a logical sequence to show the relationship between the dependencies. The last step is the tracking of dependencies which involves continuous monitoring and check-ins to ensure the team is on the right track. When tracking, if we find a slippage, it might require a re-allocation of resources or a change in the plan of execution to get back on track. Now, dependencies can be within or across teams, resourcing-related (dependent on an engineer from another team), or external (such as procuring a license for a particular tool). Hence, it is essential to have clear timelines expected against the dependencies tracked in the project/feature tracker shared above.

### Did you know?

A [RAID log](https://asana.com/resources/raid-log) (<https://asana.com/resources/raid-log>) is a form of a project planning tool for Risks, Assumptions, Issues, and Dependencies. It helps proactively identify the risks to the project, any assumptions the engineering team is taking, track the issues which are more active risks that need to be addressed, and lay out the dependencies involved. This helps provide transparency to the team and leadership to plan for strategies to mitigate them and prepare backup plans.

**Figure 10.4 A sample dependency management visualization for Project Accessible to show all dependencies**



For Project Accessible, various levels of dependencies were identified. One basis was the backend needed to be ready before the front end could integrate with the APIs. Things can be developed based on the API specification or contract where the front end can mock and build the actual API contract, and the back end can implement it in parallel. This means the front-end engineers use mock test data as a stop-gap solution to avoid a waterfall execution of the project. Similarly, the team depended on the upstream team to collect customer accounts data, QA for testing, and the UX team to provide UX mockups and wireframes. Also, the team was deciding on using an open-source library for text-to-speech conversion or building an in-house solution, another candidate for a dependency on the tool to be used. Remember that the lag incurred in delivering item 1, first in the dependency sequence, will have a ripple effect on the other items laid out in the dependency management plan. Once identified, the dependencies were organized in the logical sequence expected, adding them to the feature tracker and using other tracking tools to give them a visual representation.

As EM, the technical program manager and you keep track of those dependencies using the feature tracker through frequent check-ins with the teams involved and helping unblock them.

### **10.1.3 Execution**

The next stage is the execution, where actual implementation and action begin. This involves the engineering team implementing the solution, testing, and spending time on defect/bug triage. As EM, this is the stage where you might be tasked with several tasks that you need to juggle between - active contributions to roadmap execution, leading the engineering team, participating in technical design review sessions, helping manage conflict, doing frequent check-ins with the team members and the stakeholders and of course managing the risks.

#### **Lead the team**

As EM, you are the team captain, and each individual would look up to you as their leader/coach. You set the foundations and lead by example. Since the execution begins now, one of your first steps might be to get the team together and align on the various agile sprint ceremonies the team would like to have to stay on track for the project. This can involve having regular product grooming sessions in partnership with the product team, sprint planning where the team gets together and decides what tasks to focus on for the next set of sprints and assigning owners against those tasks, regular sprint end retrospectives to continuously learn and iterate and have demos to share the progress. I have seen teams successful with daily stand-ups or having a standup once a week, so it's on you to help the team develop team norms to align and see what works best for them.

Time will always play a key role in leading the team. Sometimes there can be development delays due to reasons which can either be in your control or not. These might blow up the launch schedule. In such scenarios, staying calm and developing a path to the green is essential. Adding more resources can help to cover the delay, simplifying the design, negotiating a trim-down version of initial requirements, and lastly, there may be a need to delay the

launch, with proper heads up to leadership. Sometimes you may not get a date from a team you depend on in terms of when they can give the inputs, so your team is unblocked, for such situations, you can ask a date for a date to have some plan in place. With tight deadlines, you and the team can end up in fire-fighting mode, so keeping a balance between the expectations and the resources available is a must. One tip is to use your cross-functional partners to the fullest as they are your support system and partners in crime. Refer to the chapter on 'Working with cross-functional partners' to learn more.

Another aspect that comes into play as you lead your teams is to hold people accountable, mainly when team members are based in different time zones and are geographically dispersed. This means members have to be accommodating and take a call between what meetings are needed in real-time (synchronous) versus asynchronous. As EM, you will be responsible for setting a conducive environment for the team. You will be a facilitator, advocate for the team, and arbiter of consensus when the team(s) are stuck or need direction/advice. This will help them function and give their hundred percent.

## **Participation in technical meetings**

As an EM, you play an essential role in being part of technical design discussions and sometimes reviewing the team's code. This includes team-facing meetings and meetings with enterprise architecture or teams that work on organization-wide security, guard rails, and direction. As an enabler to the team, you partner with your team's technical lead and help identify any blindspots the team could have missed or help bring eyes to risks and dependencies based on your experiences. This also helps instill confidence in the team and shows you are part of the team. Team members tend to appreciate a technically strong EM and feel more comfortable approaching them if they were to hit a roadblock. Also, this is the time to help identify not just the current project's design in hand but also coach the team to think about maintainability and extensibility for future enhancements. Of course, in the end, team members will have to go on-call for the services and features your team owns, hence every effort should be made to make them

operationally stable. This also calls for emphasizing QA/testing and documenting robust runbooks to help ease on-call and provide better documentation for posterity- keeping the operational bar high.

## **Vendor/Third party (tools) management**

Sometimes, your team will be tasked to integrate with third-party tools and services, or you, as EM, will have to manage a set of vendors for your business. You will be responsible for maintaining those relationships with the vendors to ensure you get the best price and effective customer service. If the integration is new, you might be tasked to identify the vendor for the company, which means collaborating with cross-partners like the product manager and legal, procurement, and finance departments to shop around and find an apt solution for the organization. This also means you would need to be involved with paperwork to ensure negotiations happen over prices and services and the vendor stays compliant with all regulations to ensure the safety of the employees and the customers. There should also be precise alignment regarding an action plan in case of conflict, with clear service level agreements signed for both parties. Next, let's explore managing conflicts during project execution.

## **Managing conflicts**

Conflicts are no surprise when working at a workplace, especially in projects where the teams might have dependencies, and everyone involved would like to move fast. The conflict can happen inter-team (between your own team members) or cross-team (let's say, a conflict between an engineer from your team and the technical program manager). Feel free to go back and read the section on Conflicts in the chapter 'Working with cross-functional partners' to refresh the concept. In such situations, if the conflict does not land anywhere on its own, you, as the EM of the team and project, will have to step in, acknowledge the conflict. Let's take an example of a conflict with respect to Project Accessible and see how one could navigate through it.

So for project accessibility, your team engineer Jane and an engineer from the other sister team, Henry, ended up in a conflict. The root cause was the

two different schools of thought regarding how dashboards should be implemented for the project. Jane from your team wanted to build a new set of dashboards specific to this project, while Henry believed there was an opportunity to reuse the templates of existing dashboards. They both called for a team meeting, but the discussion landed nowhere. Now you, as EM, have to step into this situation to help clear the air and bring in some alignment. Firstly you step in and acknowledge the conflict and schedule a 1 on 1 with each of the members. You go in sync with a blank slate to understand the perspectives of each of the engineers. You learn that Jane wants to implement a few best practices in the new dashboards she plans to implement, while Henry wants to move fast and not reinvent the wheel. As you do these 1-on-1s, you make sure to ask the right questions to each of them, in terms of:

- What is the ask for the project?
- Given the tight project timelines, how confident do they feel about implementing the solution?
- Is there a better way to make the dashboards extensible for the future?

These sample questions surely intrigue Jane and Henry to think through the solution and understand the other perspectives. Next, you call for a sync-up with the three of you and the project lead in the room, and as a neutral coach, you try to steer the conversation to a convergence keeping a data-driven mindset. With these individual and collective discussions, Jane realizes the overall efforts to create dashboards from scratch. She understands that the best practices could be introduced to existing templates, creating a broader impact as it will also fix existing dashboards. Thus, by keeping a calm, data-driven approach and resolution through consensus, you could steer the discussion to alignment and help resolve the conflict. Next, let's look at the importance of frequent check-ins with the team members for a smooth execution of the project.

## **Check-ins/Feedback**

Frequent check-ins and a robust feedback loop help ensure the team stays on track and within common goals. During the execution of the project, it is

essential to set frequent check-ins with your team engineers and the cross-functional partners involved. This serves two primary purposes- firstly, you can stay aligned with each team member, ensuring you understand each other's expectations and maintain an open communication channel.

Secondly, it helps gather feedback for your team and yourself early to avoid surprises later. Similarly, you can share constructive feedback with others to help grow.

Also, this is an excellent opportunity to engage in any negotiations if needed. For example, for the project accessible, there was a last-minute request from the product team to change the user experience (request for change or RFC button from left to right) of the toy. Now this changed several things and was last minute, hence you engage your engineers and technical program manager to help estimate the new effort and see if it can be accommodated within the constraints of bandwidth and timelines. If not, this is a chance to negotiate with the product and see what can be punted out to accommodate this high-priority task. With the benefits of early check-ins and feedback, several companies always go on to release a beta version of the apps, sometimes doing an employee release so that internally the product can go through dogfooding before being released to the end customer. The idea is similar, gather early feedback and fix bugs before it makes it to customers.

## **Risk management**

Risks are common in projects and can crop up anytime, anywhere. The earlier the risks are identified and assessed, the better it is for the project to avoid drastic consequences later. So for risk management, you start with identifying the risks, which usually involves getting the project team together and brainstorming all things to be cautious about that can blow up as a risk. Once risks are identified, the next step is to categorize them based on the severity of the threat they pose. Given the resources and time, the team should align on what all risks will be addressed as part of the project and assign owners against each with a timeline against them. The owners are responsible for carving out a risk mitigation plan/path to green and helping drive them to completion. Frequent check-ins will be done as part of the regular sync-ups to ensure the team is progressing in the right direction.

## **Did you know?**

A well-known lightweight model used for risk management is the **ROAM model**. ROAM stands for Resolved, Owned, Accepted, and Mitigated.

**R(Resolved)** - This is when the risk has already been resolved by the team and is no longer open. **O(Owned)** - This means the risk now has an owner assigned who is responsible for resolving the risk. They will work with the team to devise a resolution strategy in a time-box manner.

**A(Accepted)** - This is when the team decides to accept the risk and live with the consequences, perhaps because it is not a severe risk for the project.

**M(Mitigated)** - This state is when actions were taken by the team to help reduce the severity of the risk. The risk is now mitigated and might not be fully resolved, but the impact is reduced severity, and we plan to live with it.

The approach helps the team group the risks in the categories above and develop a game plan to address the open risks. This also promotes proper risk tracking continuously so teams consciously consider addressing risks.

The idea is to ensure the risk does not turn into an issue, which is an active item in the present that the team is dealing with. Hence the idea is to identify risks as a proactive approach and address them before they become active issues. The team will make several decisions as they progress through the process, so ensure to document them so there is a reference to return to later. The risks can also be tracked in the feature tracker. Sometimes it is a separate tab in the spreadsheet, and other times it's a column on the appropriate row, but it helps to see them listed, along with any resolutions, if there are any.

## **Transparency to stakeholders and leadership**

Transparency and visibility to leadership and business stakeholders help build trust and ensure the team gathers feedback early in the process. Let's say you demo the team with the first iteration of the product, now the visual

feel of the product can help gather feedback and ensure the engineering team and stakeholders are on the same page regarding what is expected as an end goal. Treat it like a quick feedback loop that helps iterate faster on the product. Another way to bring transparency is sharing the dashboards, sprint burndown charts, sprint demos, project updates, enablement sessions, and preparing a launch plan to get a go, no-go from the leadership.

Scrum of scrums is another sprint ceremony that can help with better transparency to leadership. It brings the updates from individual team scrums to a bigger picture to see how different parts fit together. This also helps ensure the dependencies and risks are being catered to with no blockers on the critical path of the project. For project accessibility, you want the leadership to get first-hand updates on progress. You partner with your technical program manager to build status reports sent on a weekly cadence to the leadership to provide updates and show the remarkable progress the team is making and keep them informed.

#### **10.1.4 Post-execution**

Post-execution tasks are the ones that are a set of actions performed towards the end once the project is complete. They are like a closure protocol to ensure we have met the project goals and evaluate its success. The activities help us not only in the present to close the project but also set the foundation for future iterations of the project and document the learnings from it. Let's explore the post-execution activities in detail.

##### **Recognizing the team members**

As humans, we naturally tend to incline towards feeling notable and recognized. It boosts our confidence and morale and makes us feel valued. Recognizing your team members and cross-functional partners for their contributions to the project is a valuable gesture to show their efforts were recognized. This can be sending accolades on team chat groups, shout-outs to members in a large setting, nominating members for spot bonus programs(if your company has one), or sending some goodies to them. This helps acknowledge and appreciate all their contributions from start to finish.

Revisit the chapter on ‘Rewards and Recognition’ to refresh how you can value your team members and the people around you. This is also an opportunity to provide constructive feedback to weak performers and help them identify potential growth opportunities.

Taking the example of Project Accessible, let's say we had two engineers who exceeded expectations and helped resolve a significant bug identified in the last stages of the project. Since the bug was critical for the launch, they spent a weekend debugging one of the issues with a text-to-audio converter component and wrote a script to consolidate stake records and helped resolve the issue. As the EM, you appreciated their contributions and ensured their work got visibility. You send them a shout-out in the leadership forum where you have your manager and the skip as part of the group. Next, you recognize their efforts in the team meeting setting and share how they went above and beyond the call of duty. These efforts helped boost their morale, and the team appreciated the recognition culture you have set in the team.

## Success metrics

The launch of any project is complete by comparing the results to the objective success metrics aligned by stakeholders and team members at the start of the project. We need to analyze the data per iteration and make course corrections (in agile development). They help evaluate how the project and the team performed against expectations. As a project team, multiple metrics can be used to evaluate the success criteria. Let's look at a few common ones:

- **Project timelines-** We evaluate if the project was launched successfully at the decided timelines, using the aligned resources and milestones. Where timelines or milestones weren't met, it is helpful to know the why and to document that. It could be as simple as "We just under-estimated the work involved," or as complex as "External vendors couldn't provide the parts we needed by the time they were expected." The difference could be huge, though.

- **Budgeting**- Did the project stay within the range of the budget set aside for the project? This is where we evaluate the cost invested in the project.
- **Return on investment**- Return on investment on a project is the evaluation of how well the project was received in the market and if the company can meet or exceed the revenue goals expected from the project's launch. The idea is to see if we achieved the value predicted in the planning phase.
- **Customer score/Product satisfaction score**- In the spirit of customer obsession, you want to evaluate how your end customers feel about your product and what their sentiments are after using the product. This can be performed with user research studies, sending customer feedback surveys, or evaluating the customer complaints received.
- **Think out of the box**- This involves evaluating if the product line is innovative and is new of its kind in the market vs something that already existed.
- **Employee survey**- Yes, a survey and understanding how our employees felt with the launch of a product is equally important as evaluating end-user satisfaction. This helps plan if the employees feel connected to the vision and goals of the company and are here to stay to give their best.
- **Dashboards to evaluate usage**- This is where we take a data-driven approach and evaluate the number of hits on a page or how many users are using our product, how they are using the product, any friction points seen in the application, etc. We must build observability into the application to analyze customer trends and gain insights for future improvements.
- **Operational stability and analytics**- Here, we evaluate the stability and performance of the backing infrastructure and services that power the customer experience. Metrics such as latency and how much load the service can handle come into play to ensure a smooth customer experience.

For Project Accessible, you, as EM, ensured the project met the stipulated requirements and timelines, along with delivering the project (project timelines) with the initial set of engineers allocated (budgeting). The product

manager helped evaluate the return on investment, where in the first three months, the project sold 20K toys, bringing revenue of 250K (return on investment). This was compared against the estimated metrics set at the start by the leadership of selling 15k toys in three months of launch. The user research team did a survey where the product was well received and appreciated in the market, giving a customer satisfaction score of 95 over 100 (customer score). The product line was something familiar to the market, but it was well received due to better quality and price point against the competitors in the market. You, as EM, decided to do an internal anonymous survey for team members involved to gauge how they felt about the project execution and launch (employee survey). The survey returned an employee satisfaction score of 70% and some constructive feedback. Voila—the opportunity to learn for next time. The team members had suggested having retrospectives early and often to raise issues earlier. For example, the team struggled with some testing initially and wanted to iterate on the testing process early. This caused some delay as the first retro happened 4 weeks into the project. Such metrics that suggest things did not go as expected are equally essential to capture to understand the overall success of the project and learn for future projects. For monitoring, the team created multiple dashboards to evaluate customer trends, which location sold the maximum toys, incoming trouble tickets, and root causes for customer complaints.

While these metrics are a great way to objectify if the project was successful, sometimes subjective criteria such as hype in the market or customer sentiments also help evaluate the success. The success metrics feed in as input for the entire project team and leadership as they plan ahead for further improvements and technical roadmap.

## **Operational maintenance and technical debt**

Well, as a continuous delivery process for any agile development project, operational maintenance/production support becomes the next oversized item on the list, which has all eyes from the team members to the leadership. You don't want your service to go down or the application latency to be so high that it hampers the customer experience. At the same time, you want to plan for the infrastructure cost and optimize the resources to avoid hefty

cloud services usage or third-party bills. In this phase, you want to ensure the right set of alerts and monitoring so the team on-call can get an immediate alert if something is suspected to go wrong. At the same time, not everything will have a huge blast radius, hence the importance of defining service level agreements (SLAs) for the service everyone abides by. Let's understand this through an example. For Project Accessible, the organization followed the standard trouble ticket and SLA agreement practiced in the company, with four levels of severity for any trouble ticket or issue.

**Table 10.1 Different severity levels for trouble tickets**

Severity 1	This occurs when the business gets impacted, and everyone should stop working on anything else and focus entirely on fixing this issue as soon as possible. The blast radius, in this case, can be that each company customer is impacted. Example- like order drops or a security breach of customer data. SLA can be like a 4-8 hours turnaround.
Severity 2	This is also critical and needs immediate attention, but the blast radius, aka the number of customers impacted, might be smaller(maybe 40% of customers are impacted). Example- during the order checkout, the applying code option is broken, so any customer with a discount code cannot use it and hence needs to pay the total price- not a happy customer experience, right? SLA, in this case, can be 24 hours.
Severity 3	This is not critical as the above two and the on-call can take 1-2 days to triage the issue and get back—for example- an application that needs to go through a security review before launch. Since the service has not been launched yet, there will be no production impact and hence the SLA of 2 business days.
Severity 4	This is not urgent at all and something that can wait. This has the

4

least priority for the on-call or team and will be addressed once other trouble tickets of higher priority are addressed. Example- a marketing manager requesting access to your internal platform to show sale numbers. In this case, SLA can be in days such as 5 business days.

So we have defined the SLAs and turnaround times expected from the team for the trouble tickets and escalations; how about having standard on-call runbooks per service to have a faster troubleshooting mechanism? Yes, on-call runbooks are another way to optimize the speed at which the team can cater to high-severity issues and use them to at least document the most common customer issues and how to address them. This can become a repository where all the team members contribute to enhance the team documentation and help improve the on-call. Again, this will be a living document that will continue to grow as the team encounters new challenges and issues and ways to resolve them. This also comes in handy when the service/project needs to be handed over to a different team post-launch, along with knowledge transfer sessions. On a side note, I would recommend that teams have a "warranty period" where they support what they built for some time before production operations take over entirely and are handed to a different team. These warranty periods can vary between 30 and 90 days, but either way, it is an excellent way to have the two teams work together and significantly reduce the stress of both teams.

Sometimes when enhancements or new features are launched with additions to the legacy services (Yes, organizations do have some services, let's say, built 20 years ago), it can be hard to maintain them, especially if the code quality is not up-to the mark. There is missing documentation, calling for tribal knowledge lost over the years. In such cases, maintaining the services might be a little more complicated as changes in one place can cause a ripple impact on others, which might be unknown to the developer team. If you own legacy services and are building on them, advocate for documentation in your team and emphasize robust testing before the feature's launch.

Technical debt is a crucial topic arising from old legacy code or when launching new products and features. When time is of the essence, and a fast

release of a prototype is necessary for gaining a competitive advantage, the regular six-month development route may need to align better with the product organization. In such cases, as an EM, you would negotiate features and collaborate with the engineering team to explore alternative technical approaches. It's important not to shy away from discussing technical debt, as it is a common aspect in every organization. Even if your team had to take shortcuts to meet production deadlines, resulting in technical debt, it should not be ignored. As an EM, you should openly communicate with stakeholders and all involved parties about the pros and cons of each approach. It is advisable to get written alignment that once the primary product is released in production, the engineering team will have the bandwidth to address the technical debt.

Using tracking tools like Jira or Asana, you can proactively create tasks to address the technical debt and keep them organized in the project backlog. If necessary, you can seek additional resources to help the team resolve the technical debt and maintain the quality of systems. Enterprise architecture plays a significant role in managing and tracking technical debt. They can be involved in decisions related to technical debt and assist teams in managing and resolving it over time. In conclusion, addressing technical debt is crucial for maintaining a high-quality product and should be openly discussed and actively managed throughout the development process.

## **Project retrospective**

A retrospective ceremony is like a learning opportunity to acknowledge both sides of the coin, celebrate the wins, and learn from failures by identifying opportunities for the future. This can be conducted at the project level at the end of the project but is also sometimes done as a project mid-check-in to steer in the right direction early in the process. The idea is to have all key stakeholders and players involved reflect on the project and understand the opportunities for the future. In the table next, we will explore a sample retrospective board for Project Accessible.

**Table 10.2 Sample retrospective board for project Accessible to document kudos, what went well, opportunities, and action items**

Kudos	What went well?	What could be improved?	Action items
Kudos to Jack for going above and beyond for building the volume control feature in a short span of time	Text to speech API integration with UI complete	Too many meetings, takes away builder time	Move team standups from daily cadence to three days a week
Shout out to Jessica for documenting a runbook with common troubleshooting mechanism for frequently occurring customer issues	QA testing made good progress		
...	...	...	...

The project end retrospective helps get all stakeholders together to discuss the project challenges and lessons learned for the future and helps identify improved working methods. The action items can then be prioritized, and an action plan can be laid out to reap immediate benefits. As EM, you can help set a positive tone and foundation for the retrospectives so people use it as a learning experience instead of pointing fingers at each other in case something did not go as planned.

### What do other leaders have to say about it:

“Choosing the right tool for tracking and managing your projects is important. Getting visibility across team projects becomes much easier if you all choose one tool for progress updates. Every time you start a project with a cross-team project, you start with getting the POC from each team and setting up a weekly rhythm with them. Once you close on high-level approaches, ensure that every dependency on cross-teams is assigned to them in the tool. One important part is to share monthly progress with all stakeholders and ensure everyone is on the same page as part of those progress reports.”

~ **Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

“For the last 3 years, I have worked with small teams at small startups. Success metrics are different than when I was at larger organizations. We seek market fit and iterate quickly, sometimes faster than a 2-week sprint can accommodate. My guiding principle stays the same across organizations: “Is

my team executing the most valuable work for the company? In practice, at small startups, this usually skews to building the Minimal Iterable Product (instead of Minimum Viable Product) and then rapidly iterating with feedback from sales/marketing/product to hone into the MVP. This sometimes means making quality tradeoffs to answer critical market questions knowing that we will have to clean up later.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“The key to tracking progress on multiple projects simultaneously is establishing project leaders or product owners from the beginning. Projects can often go sideways when ownership is unclear. As a leader, we must acknowledge that one person cannot keep track of all projects simultaneously and that delegation empowers other leaders on your team. Even in the case of cross-team projects, appointing one single owner responsible for negotiating timelines, capturing dependencies, identifying risks, and managing the overall health of the initiative is my not-so-secret sauce of a successful outcome.”

**~ Nishat Akhter, Data Product & Engineering Leader, AWS**

“For a project's success metrics, defining success criteria early before prioritizing the project is important. This usually helps in identifying the metrics that could measure success criteria. Success metrics vary from team to team and company to company; could be related to business drivers (revenue, acquisition, engagement) or technical metrics ( performance, latency, errors)”

**~ Saurabh Gandhi | Sr. Director Software Development Audible | Amazon | Ex-Amex**

## **10.2 Stop & Think: Practice questions**

1. What is your most memorable project delivery experience? Why is it most memorable?

2. What is your most forgettable project delivery experience? Why is it most forgettable?
3. How would you navigate a project with strict deadlines?
4. Share a project experience where the delivery failed in the timelines decided. How was it taken by the stakeholders? What did you do next?
5. How would you plan for operational maintenance for a newly launched service by your team?
6. What are the most unique findings that came to your attention as part of a retrospective? Why was it surprising for you?
7. Have you faced difficulty managing operations for a newly released project/product? What do you think was the problem? How did the team navigate through it?

## 10.3 Summary

- Smooth and fluid movements require coordination and collaboration, not just within your own body but also with partners if you're working with others. Project delivery, management, and execution can be likened to a dance, where the entire team, including the engineering manager, product manager, and technical program manager, act as dance partners striving to perfect their routine for the next performance.
- The success of a project relies heavily on the planning, execution, and delivery phases, as the effort put into each phase determines the quality of the final product. This requires the collaboration of various stakeholders working towards a shared objective.
- The four phases of the project lifecycle include Pre-planning, Planning and project kick-off, execution, and post-execution.
- Pre-planning is the first step to setting the foundation for the project/product launch. This phase involves identifying what the project is, whether it is high in priority or not, looking at the company's overall vision, defining the project's scope, and identifying the key stakeholders involved in the project.
  - Prioritization - Here, we understand the project's priority over the other projects in the pipeline. This is where prioritization or stack ranking of projects comes in handy across the company and plays a significant role in setting expectations across teams.

- Scope Definition - In the scope definition phase, the team members take one step forward by getting into defining the scope of the project and documenting it. This helps get a deeper understanding of what will be included as part of the delivery of this project(in-scope) and what will be considered out of scope to define the boundaries.
- Roles and Responsibilities - Once the scope is defined, our next step in the project is defining the key players involved and their roles in the process. Having blurry lines of ownership tend to cause confusion and missed requirements.
- Planning and Project Kickoff - The second phase in the project lifecycle is the project planning and kick-off stage. The wishlist created from the pre-planning phase is turned into a proper project plan.
  - Budgeting and Resourcing - Budgeting involves considering the costs associated with project execution, including resources, software/hardware expenses, and time allocation. This allows leadership to plan for costs. Resourcing involves assigning engineers and cross-functional partners as critical players in project execution.
  - Team Building - This is where the specific skills and expertise of the team members are taken into account to help get to the finish line. Since the resource allocation and RACI model are already in place, this is the time to ensure team members know each other and understand their roles in completing the project.
  - Communication channels - Having open communication channels and transparency helps build trust and ensure no surprises later in the project development lifecycle. As EM, your job is to define the communication channels that will be useful for the team members to interact and unblock themselves should they hit a roadblock.
  - Training - This will involve individuals learning new skills to be effective in the role and feel more confident. As an engineering manager, your job is to identify the skills gaps and find the desired resources to support your engineer's needs.
  - Project/Feature Tracker - A project or feature tracker helps the team to keep track of all features or items expected and have corresponding owners, statuses, and priorities against each. It is

like a living document to help ensure each team member is on the same page and acts as a source of truth in case of disagreements.

- Dependency management - Dependencies occur when the sequence of activities involved depends on each other in a sense one is dependent on the other. Dependency management is identifying, organizing, and tracking these dependencies to lay out the sequence of activities.
- Execution - The next stage is the execution, where actual implementation and action begin. This stage usually takes the longest time as it involves the engineering team implementing the solution, testing, and spending time on defect/bug triage.
  - Lead the team - As EM, you are the team captain, and each individual would look up to you as their leader/coach. You set the foundations and lead by example.
  - Participating in technical discussions - As an engineering manager, you play an essential role in being part of technical design discussions and sometimes reviewing the team's code. As an enabler to the team, you partner with your team's technical lead and help identify any blindspots that the team could have missed.
  - Vendor/Third party management - Sometimes your team will be tasked to integrate with third-party tools and services, or you as EM will have to manage a set of vendors for your business. You will be responsible for maintaining those relationships with the vendors to ensure you get the best price and effective customer service.
  - Check-ins/Feedback - Frequent check-ins and a robust feedback loop help ensure the team stays on track and within the common goals. During the execution of the project, it is essential to set frequent check-ins with your team engineers and the cross-functional partners involved.
  - Risk management - Risks are a regular part of projects and can emerge anytime. Identifying and assessing risks as early as possible is crucial to mitigate potential consequences. Risk management begins with identifying risks, often involving gathering the project team to brainstorm potential areas of concern that could pose risks.

- Transparency to stakeholders - Transparency and visibility to leadership and business stakeholders help build trust and ensure the team is gathering feedback early in the process.
- Post-execution - Post-execution tasks are the ones that are a set of actions performed towards the end once the project is complete. They are like a closure protocol to ensure we have met the project goals and evaluate its success.
  - Recognizing the team members - Recognizing your team members and cross-functional partners for their contributions to the project is a valuable gesture to show their efforts were recognized. This can be sending accolades on team chat groups, shoutouts to members in a large setting, nominating members for spot bonus programs, etc.
  - Success Metrics - The launch of any project is complete by comparing the results to the objective success metrics aligned by stakeholders and team members at the start of the project. They help evaluate how the project and the team performed against expectations.
  - Operational maintenance and technical debt - During this phase, it is essential to establish appropriate alerts and monitoring systems to notify the on-call team of potential issues promptly. It's worth noting that only some issues will have a significant impact, which is why defining service level agreements for the service is crucial.
  - Project Retrospective - A retrospective ceremony serves as a valuable learning opportunity to recognize successes and learn from failures by identifying areas of improvement for the future. It involves key stakeholders and team members reflecting on the project to gain insights and identify growth opportunities.

# 11 Managing expectations

“If you do not know where you are going, any road will get you there.”

~ Lewis Carroll

## This chapter covers

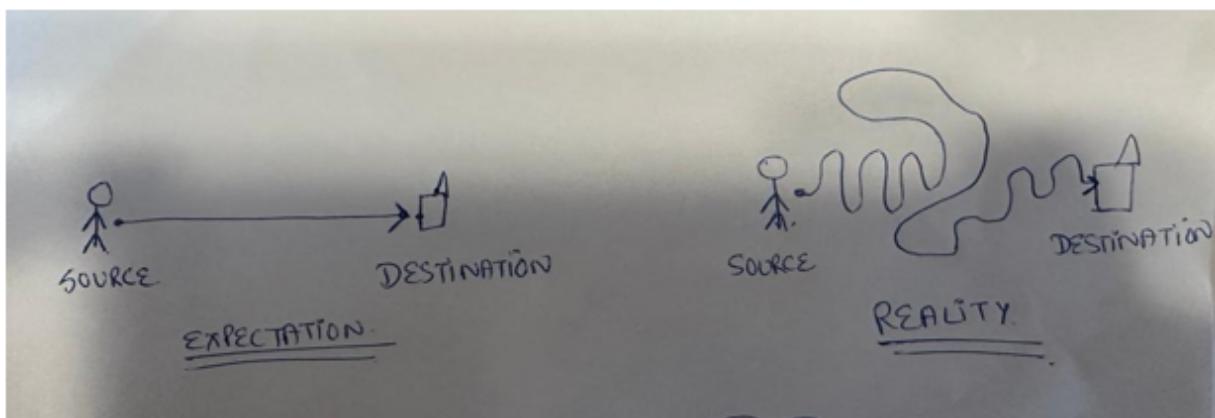
- Importance of setting and managing expectations
- Common challenges faced during expectation management
- Framework to manage expectations
- Tips on managing expectations at all levels- your manager, peers, and your team

Film-makers have used storyboarding for years now, which is the stepping stone between a finished movie script and rolling cameras to record the action. The storyboards essentially help to visualize the chronology of the scenes and help level set the expectations involved in the movie-making process. Further, as you go through the storyboards, you might realize the need for some refinements that need to be made to the script and adapt to them. This is all done early in the process to avoid surprises in the end. Well, on similar lines, setting and managing expectations early in the process plays a crucial part in the success of the task at hand. One starts with setting the expectations to provide clarity, as they talk through it, there might be a need to update expectations to help set the ground for execution. Again, this all needs to happen proactively in the process to avoid repercussions in the end that might arise due to unclear requirements and ambiguity.

Expectation management is like an agreement between two parties, they discuss the inputs and expected outputs based on the situation and use it as a mental model to describe what is expected out of the process. It helps you set the foundation, tone, and expected way of working together toward common goals. Expectations can have a significant impact on how one perceives things, and it is important to understand the distinction between realistic and unrealistic expectations. Realistic expectations comprise tasks

that are data-driven and follow the SMART criteria. Past experiences help shape the expectations for the future, ensuring the goals are attainable- given the time, constraints, and resources. On the other hand, unrealistic expectations might not be fact-based- confuse the EM and team members, and can be tangential to the situation or unattainable given the constraints. Here one side may over-expect things while the other is certain that it is not achievable. Such expectations are usually never welcomed and lead to demotivation and frustration. There can be a stark difference between how the work is perceived versus what it actually entails as shown in the image next. Hence it is important to stay grounded as you set expectations and aim for a win-win realistic situation for all parties involved.

**Figure 11.1 Expectations can be very different in terms of what is perceived in the first go vs what is actually expected/reality**



Another callout is that the buck does not stop with setting and managing expectations, a third dimension to this is managing perceptions. Oftentimes, given the human psychology of being judgmental, people build perceptions related to how the other person works and reacts. These optics impact the prediction of future behavior based on past experiences. For example, when an EM joins a new team and meets the PM, they initially have a positive outlook. However, after two months, they observe a lack of collaboration and proactivity from the PM. There is a missing business requirements document, causing frustration. The EM gives feedback to the PM and their leadership, leading to positive changes in their behavior. Despite this

improvement, the EM still carries a biased perception of the PM, influencing their collaboration. Setting clear expectations early on can prevent the formation of incorrect perceptions that may harm the relationship. While change takes time, the PM should recognize the situation and work on managing perception to establish alignment.

Managing expectations is about a shared need to build understanding, alignment, and cooperation at all levels. In the next set of sections, we will learn about the

- Importance of managing expectations
- Some common challenges faced
- A framework to set and manage them
- End on the note of managing expectations at all levels, starting from your manager, peers to your team members who report to you.

## **11.1 Importance of managing expectations**

Managing expectations is an important skill set as it helps ensure all parties involved have a clear understanding of what is expected, and that those expectations are realistic and achievable. This can help to prevent misunderstandings and disappointment and can foster better communication and collaboration. Additionally, managing expectations can help to set clear goals and objectives, and can help to ensure that resources are used in an efficient manner. In all, managing expectations in an effective manner can help to improve the chances of success for any project or endeavor.

### **11.1.1 Avoid ambiguity**

One key benefit of setting expectations beforehand is it avoids ambiguity or multiple interpretations of a problem statement. This means you clearly state the tasks and expectations and also share the level of support you will provide to the other person as they try to accomplish the tasks. This helps both parties- it helps you to understand that you are well versed with the expectation and that's what you want. At the same time, the other person knows what is expected of them. This also serves as a contract or an

agreement that both sides align for to avoid confusion later. This helps provide clarity of objectives, helps one prioritize the assigned tasks and thoughts, and helps disambiguate ambiguity.

For example, Alice expects Bob to deploy a set of code changes to production. Now if this is the only information provided by Alice to Bob, this can lead to multiple consequences depending on how Bob perceives it. Although the outcome is straightforward that the changes by the end of this should be in production, how it is executed can vary a lot.

- Can Bob simply go ahead and push forward the changes?
- Does it require Bob to do a specific testing set before changes are deployed?
- Do we need to involve any QA engineers for this deployment?
- Do changes need to go through some specific staging environment?
- Once changes are deployed, is there a need to inform any team members?

As you can see, the lack of clarity in the expectations can lead to multiple interpretations. It probably has led to frustration on Bob's part, slowed him down in the process, and certainly led to a less-effective result. On the other hand, as EM, we can't be too prescriptive, lest we fall into being a micromanager. A framework or guideline, professed as an expectation, rides a comfortable line in between. Hence it's important to set them in the start itself, align with the team members, and set them right, to avoid re-do of work later.

### 11.1.2 Build relationships

Setting expectations loud and clear helps build strong relationships between the setter and the doer. With a clear alignment on both sides, there are higher chances of the doer successfully finishing the task and even asking for the desired support. This helps build rapport, transparency, and trust with each other. It also leads to **higher engagement** and provides a **conducive environment for growth**. From the example above for Alice and Bob, Alice has shown to trust Bob as they share the expectation and the support they

will provide as Bob works on the task. Bob feels engaged and reciprocates the trust bestowed on him by focusing on the task and asking for help. This also opens up clear communication channels between the two.

### **11.1.3 Risk Mitigation**

Setting and managing clear expectations help plan for contingency measures if a problem arises. As you talk through the expectations, the setter and the doer can walk through the scenarios and discuss what Plan B is if Plan A fails. This is possible by frankly talking about the expectations and aligning on a course of action (identifying a path to green in case you hit a dead end). This also helps promote a growth mindset.

Using the example above, let's say Alice expected Bob to deploy the code changes to production, and Bob frankly communicated that being his first production deployment, he does not feel very confident doing the deployment alone. This ensured both parties have clearly stated the expectations, and Alice could now plan to have another engineer in the team reverse shadow Bob during the deployment. This ensures Bob gets the opportunity to learn the deployment process, and at the same time ensures that Alice trusts but verifies it by providing a support person to help him during the deployment. This is how one can plan for risk mitigation and have a plan in place, in case things do not go as planned.

In the realm of contemporary agile methodologies, the primary objective is to consistently provide customers with valuable outcomes. Consequently, the process of managing expectations occurs continuously, whereby you establish initial expectations, deliver a portion of the product, incorporate feedback through iterations, and subsequently manage expectations once again, perpetuating this cycle. The concept behind this approach is to frequently experiment and adhere to the principle of quickly recognizing failures in order to acquire knowledge swiftly. This methodology facilitates rapid learning and the continual delivery of value, encompassing various aspects such as project progress, expectations, and potential risks. In this section, we looked at the importance of managing expectations and how they help both parties to set a plan of execution. In the following section, we will

examine some frequently encountered difficulties in establishing and maintaining expectations.

## 11.2 Challenges when managing expectations

Setting and managing expectations is not as easy as it sounds. While at the face of it, it might look as simple as someone saying what is expected and another person acting on it, but let's be real- we are all human and not a robotic machine. When someone asks us to open the door for the home, one may simply go and open the door straight, another might first peep through the window before opening the door, and maybe a third person might interpret this message differently and ask more questions and then proceed to do the action. As human beings, we may have varying perceptions of what is expected of us, so it's crucial to clarify expectations when setting and managing them. In this section, we will examine some of the common challenges that can arise in the process of establishing and maintaining expectations.

### 11.2.1 Ambiguity

A common challenge to setting and managing expectations is ambiguity. When the setter of a task expects the doer to do something but provides inefficient or incomplete information, it leads to open-ended questions, and ambiguity creeps in. In the example of Alice and Bob for avoiding ambiguity, we saw how simply asking Bob to deploy code changes opens up Pandora's box of questions and interpretations. This leads to several open-ended questions and hence is one of the most common challenges faced. Especially for someone who is new to the company or the role, until they build context and adapt to the company culture, it is more common not to give clear instructions when setting expectations.

Although the flip side of this is the opportunity the doer gets to ask questions to clarify the expectations, which is important to avoid any assumptions. It is important to ask the right questions to get accurate details.

- Can you take external help when trying to perform the task?

- Do you see any prerequisites or dependencies for the task?
- What is the expected outcome?
- What is the timeline expected?
- Do you need to provide frequent reports or progress on the task?
- If you have another task in hand, ask for help to stack rank the priority of tasks.

This also helps the setter of the expectation understand what all pieces encompass in the complete set of the expectations they are trying to set. Also, this entails that you not only clarify what is expected but also how you will support them and when the work is expected to ensure it is time-boxed.

### **11.2.2 Inability to say ‘no’**

It is a human tendency to feel intimidated in a situation where you are to say ‘no,’ especially at a workplace. Let's say your manager has asked you to attend a technical talk on iOS development. Now, you are a QA manager and might not have an interest in attending this technical talk, especially during the week when the majority of your time is spent writing performance reviews for your team members. Now, when the time comes to say ‘no politely,’ you might be speculating how your manager might react to your response and if it will impact your work relationship with them. You want to remain likable, in their good books, and avoid conflict. This is a widespread problem and is seen when employees overcommit to work because they did not say ‘no’ at the right time. This is also prevalent with professionals early in their careers, who might feel they will not get promoted if they were to say ‘no’ to something asked by the boss. As they gain more experience and mature, they eventually learn to say ‘no.’ The inability to say no at the right time often leads to burnout and frustration on the job. Learning the power to say ‘no’ at the right time helps keep one focussed on the vision and overall goals. This does not necessarily mean you must respond to the situation immediately. Maybe you take time to sleep over it and come back the next day to respond with a ‘no’ with objective reasons and rationale to back your message. In the short term, we focus on being nice and protecting our relations, but in the long term, it might prove destructive- lost time, burnout,

and frustration. Similar situations can arise when you, as EM, have to say ‘no’ to the end customers but maintain working relationships with them.

### **11.2.3 Making it personal**

One of the common challenges faced with expectation management is taking things to heart and making them personal. Making things personal occurs when in a conversation, one party passes off inappropriate comments given the situation, and the people involved or the other person in the conversation interprets a different meaning to what is said and takes things personally due to misunderstanding. This has the potential to cause offense and hurt each other's feelings, causing a rough patch between the two parties.

Let's understand this through an example. A project kick-off meeting is being held where David walks the audience through the business requirements document. One of the team members, Julie, raises her hands to ask questions. David gracefully answers the first set of questions, but eventually, Julie starts to question the project's feasibility, given the limited resources and engineering constraints. David is now unable to tackle the questions as his thoughts are blurred with him taking things personally. He continues to cross-question Julie and even started sounding defensive. Julie gets up and walks out the door in the middle of the conversation as the argument gets heated. A project kick-off meeting soon turned out to be a battleground, with David taking the questions personally and indirectly treating them like criticism of the technical design. This is a classic example of someone taking things personally at face value and reacting, or maybe overreacting to the situation. The end outcome is a pin-drop silence and an awkward situation in the room full of team members. Some basic strategies of how David could navigate this situation are perhaps

- Step away from the room a bit to catch a breather
- Try to understand the perspective from Julie's shoes and not necessarily treat it as blame, or think the intent had a negative connotation, rather treat it as a learning opportunity
- See if there is a better way to communicate thoughts to the other person

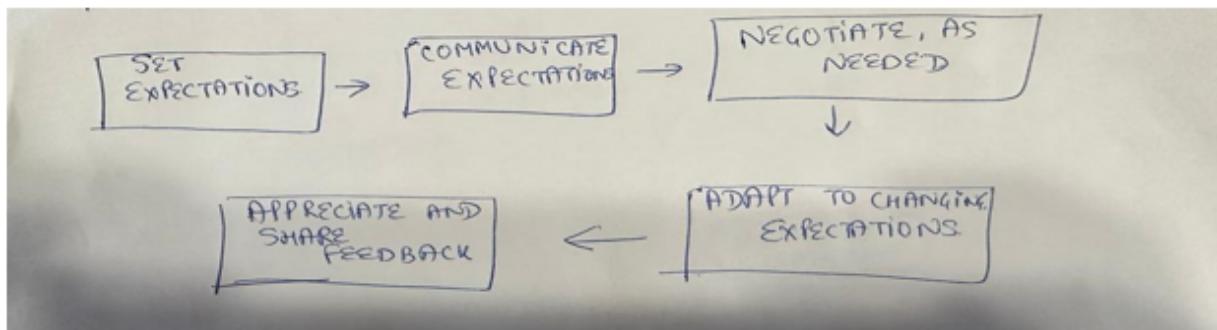
- Try to come to a common ground instead of pointing fingers at each other

Above are some of the many ways one can navigate such situations. As an engineering leader, you might benefit from staying away from making things personal and also intervening as needed if you see someone in your team hurting others' feelings. The idea is to encourage people to be polite and amicable to help come to a resolution as, in the end, we all know, all humans are not cut the same.

## 11.3 Framework to manage expectations

Managing expectations is one of the core competencies of an EM. It is like a recipe that needs to be mastered over time to make it perfect, by learning from past experiences. Let's look at a step-by-step framework that will help you guide through the process of effectively setting and managing your next set of expectations with your boss, peers, or subordinates.

**Figure 11.2 A five-step framework to manage expectations**



In the next subsections, we will explore all the five steps of managing expectations, namely- setting expectations, communicating them, negotiating if needed, being agile with changes expected, and appreciating and sharing constructive feedback with the team in detail, so let's get started.

### 11.3.1 Set expectations

First step in the process is to understand what is expected from the other person(s) and set the expectations clearly. The idea is to avoid making assumptions or keep ambiguity out of the process. At this step, clearly state ‘what we are doing,’ ‘why we are doing it,’ and sometimes ‘how we are doing’ to set the context. Even before engaging with others, the idea is to sit down for yourself and think through in your head (like a storyboard) what realistically you expect from the other person. If you are the one(giver) giving expectations to another person(doer), your goal is to ensure you are clear with the expectations in your head. This should define what will be the input and what is the expected outcome. Also, this is the stage where you identify the resources and support you will provide to enable the doer to complete the task successfully in the stipulated time. Yes, tasks should have a stipulated time so they are time-boxed and not open-ended. Setting expectations can vary from what you expect someone to do at work, in terms of expected behavior, any docking guidelines to be followed in the team, balancing work with personal life, and so on.

Jason is an EM for a full-stack team of 8 software engineers. As part of the quarterly program increment planning, he has been tasked to plan ahead a concrete roadmap for his team for the next three months. This plan will define the features and products the team will work on for the next three months to launch by the end of this time frame. Jason collaborates with his product manager and technical program manager to gather the product requirements and also consolidates them with the engineering team tasks that are coming bottom up. Since Jason will be working closely with his team and cross-functional partners to plan for the quarter, he sets up a kickoff meeting with the people to set the expectations. In the meeting, he shares the need for the quarterly planning process, and how everyone will collaborate to come up with the plan for the next quarter. He further shares the sizing scale that will be followed to identify how many sprints worth of effort are needed across each feature.

### **11.3.2 Communicate expectations**

As the next steps, once the expectations are set, the idea is to communicate and sometimes over-communicate to avoid confusion later. This helps

provide clarity to the tasks beforehand to avoid throwaway work. Think about it like reinforcement learning, where one might repeat and remind others of what has been discussed so that everyone aligns on the same page with similar interpretations of the outcome. As part of communication expectations, ensure to make any crucial or difficult conversations early instead of deferring them for later. This helps provide clarity, transparency, and plan for a risk mitigation plan if things do not go as intended. One key thing to remember is to document what everyone aligned on so things are in black and white, and one can come back and refer to the agreement everyone aligned on. Plus, this provides an opportunity to visualize the expectations to help with a better understanding of the nuances involved. This is especially useful for projects that cut across various teams and go on for longer durations, sometimes multi-year projects. Documentation also helps build accountability and ensures we have a clear understanding of what was agreed upon and summarized.

As part of communication, Jason needs to communicate expectations to both parties- first, his team members should have clarity in terms of what the overall roadmap asks and start sizing the work based on the scope. Second, is the product leadership to confirm them based on the sizing that all the team can pick in the next quarter. This is also the time when you collaborate with the product team to have a clear stack ranking of product feature tasks. Jason sets a meeting to ensure everyone is aligned and on the same page. He then sends a written email of the minutes of the meeting in terms of what was discussed and what everyone aligned on to communicate the expectations clearly.

### **11.3.3 Negotiations**

Not always things go as expected, so it is important to prepare for unexpected situations and handle them in a calm and composed manner. As humans, we can all make mistakes, be it a mistake in the initial estimation of the work, during implementation where we over-engineered a technical design, or maybe even during testing where defective code was made into production. When such an unexpected or conflicting situation arises, the idea is to dissect the root cause and develop a solution or path to the green. The [5](#)

[whys approach](https://en.wikipedia.org/wiki/Five_whys) ([https://en.wikipedia.org/wiki/Five\\_whys](https://en.wikipedia.org/wiki/Five_whys)) is a great tool for root cause analysis. You should ensure the team members avoid a blame game and instead focus on correcting and learning from their mistakes. Also, at the same time, not all expectations will be met the way you think they should be. This is also the time to root cause the reasons and work together to devise a solution. Many times, a minimal viable product helps with negotiations as it allows a team to show a draft version of what is being asked for and get feedback early in the process to avoid surprises later. Not only it helps the customer, but it also helps the developer to calibrate their estimations and assumptions of workload.

Going back to Jason, let's say he received a list of ten product features that the product leadership wants. He ensured alignment on the priority of the ten features by confirming the stack ranking for them. Next, his engineering team pitched in two engineering efficiency focussed features. Jason set up a sizing and planning meeting with the team where each of the total twelve features was discussed in detail in terms of the scope involved and what it would take to take the features to the finish line. Next, the sizing was marked as S, M, or L depending on the number of sprints it will take for the feature. For example, S meant 1-2 sprints worth of work, M- 3-4 sprints, and L meant 5-6 sprints worth of work (the entire quarter assuming each sprint is 2 weeks) for one engineer. In the next figure, is a sample product feature list which has been estimated by the engineering team to understand the level of effort involved.

**Figure 11.3 A sample feature request list with estimated developer sizing from the team**

Stack Ranking	Feature/Initiative	Priority	Sizing	Translation in # of days
1	Feature 1	P0	M	4 sprints * 2 weeks * 5 days = 40 days
2	Feature 2	P0	L	6 sprints * 2 weeks * 5 days = 60 days
3	Feature 3	P0	L	6 sprints * 2 weeks * 5 days = 60 days
4	Feature 4	P1	M	4 sprints * 2 weeks * 5 days = 40 days
5	Feature 5	P1	L	6 sprints * 2 weeks * 5 days = 60 days
6	Feature 6	P1	M	4 sprints * 2 weeks * 5 days = 40 days
7	Feature 7	P1	S	2 sprints * 2 weeks * 5 days = 20 days
8	Feature 8	P2	M	4 sprints * 2 weeks * 5 days = 40 days
9	Feature 9	P2	L	6 sprints * 2 weeks * 5 days = 60 days
10	Feature 10	P2	L	6 sprints * 2 weeks * 5 days = 60 days
11	Feature 11	P3	M	4 sprints * 2 weeks * 5 days = 40 days
12	Feature 12	P3	M	4 sprints * 2 weeks * 5 days = 40 days
<b>Total</b>				<b>560 days</b>
		<b>S</b>	<b>1-2 sprints</b>	
		<b>M</b>	<b>3-4 sprints</b>	
		<b>L</b>	<b>5-6 sprints</b>	

Once these estimates were ready, the next step for Jason was to get the capacity estimates from the engineers. This means roughly estimating the number of working days in the upcoming quarter minus the holidays, buffer days (in case someone gets sick), planned vacation by team members, on-call capacity, meetings time, etc. The idea is to get accurate estimates of how much development capacity is available to understand what the team can commit and what falls below the commitment line. In the figure, you will see a sample capacity planning estimate to understand how much engineering work hours are available.

**Figure 11.4 A sample capacity planning sheet for the team to get available capacity**

Total company working days(excluding holidays)	100		
Meetings buffer(in days)	10		
On-call capacity buffer(in days)	60	(Assuming 12 weeks in quarter, 5 working days each week, so $12 \times 5 = 60$ days)	
<b>Team member</b>	<b>Planned vacation</b>	<b>Available capacity</b>	
Alex	4	$100 - 4 = 96$	
Bob	2	$100 - 2 = 98$	
Charlie	8	$100 - 8 = 92$	
Daniel	4	$100 - 4 = 96$	
Engo	5	$100 - 5 = 95$	
Farley	3	$100 - 3 = 97$	
<b>Total capacity for the team</b>		$574 - 10(\text{meeting}) - 60(\text{oncall}) = 504$ days	
Miscellaneous 10% buffer(sick day, unplanned days etc)		~50 days	
<b>Total Available capacity for the team</b>		<b>454 days</b>	

Once the capacity estimates were received, Jason could see that the entire work expected (remember the twelve features) is not something that is realistic to finish in the stipulated quarter. He saw a delta of work requested for 560 days, but the team had a capacity of 454 days, hence short of 106 days. So as the next step, he worked with the cross-functional partners to negotiate what can be punted for the next quarter, given the priority and stack ranking. This also ensured Jason and the team are not over-committing to the work and are working towards it in a planned manner. This opened up a floor of discussion and negotiations as, in reality, the engineering capacity available has to be somewhat equal to the estimated work in engineering hours, being a zero-sum game.

### 11.3.4 Adapt to changing expectations

As we saw above, change is the only constant. So things will change and not always go as intended. Hence, as engineering leaders, we need to hone our muscles to adapt to changing expectations. Most companies today have shifted from the traditional waterfall model to more of an iterative/agile model. This allows early feedback during product development or planning so that changes can be incorporated in an iterative manner instead of waiting till the end. This means one needs to be flexible and adapt to changing requirements and, of course, changing expectations. Expectations, like any other thing, can evolve over time. When you were in high school, your expectation of success might be getting good grades and then getting into your dream college. Once you get into your dream college, your expectation

of success might be to learn the skill sets and practical knowledge to be a successful professional. Eventually, as time goes on, you might start thinking more about family and having a home, and hence at each stage, your expectations keep evolving.

Jason feels confident with the negotiations he has done to help set up a solid technical roadmap considering the product and engineering requirements plus the available engineering capacity. His team begins executing the aligned roadmap. Now two weeks into the quarter, your company's competitor launched a cool feature that is gaining popularity. Immediately, your company updates the strategy to launch a new product with better capabilities to compete with the competitor. What does it mean for you? Yes, your team has been asked to pivot from the original roadmap and provide development support to build this new product. This means you and your team have to adapt to the new plan as the expectations of the leadership chain and the business, in general, have changed (keep in mind that there should be a document listing the new strategy and why the pivot would be helpful to bring the team on the same page). At the same time, take a moment to ask the right questions about why we are doing this, what we are doing, how we are doing, and when we plan to launch. This will help you understand the business value and share that with your broader team to help them understand why a change in expectation makes sense. Remember, change is for good, and every new change makes you learn something new.

### **11.3.5 Appreciate and share feedback**

So we learned that changes will happen, and it's important to embrace them with open arms. You, the team, and cross-functional partners came together to adapt to the changing expectations to help ship an awesome product. This calls for two main things- feedback and appreciation. Frequent periodic check-ins help to gather feedback. As part of feedback, it is an opportunity to learn from the experience and understand what you could do differently to do it better. The idea is to identify the strengths and growth opportunities from the lens of your peers. We have read about 360 feedback in Chapter 3, Managing people, team(s), and yourself, section 3.2.7. It is useful to take a pause, introspect, retrospect with the team, and share constructive feedback

with each other. A retro will help us understand how expectations were communicated, how well the team did matching those expectations, and whether we could learn or do better for future projects. On similar lines, this is an awesome opportunity to show appreciation to your team members and cross-functional partners who went above and beyond their roles to help manage expectations and take the project/product to the finish line. We learned about this in detail in the ‘Rewards and Recognition’ chapter. Go back and read it to get a refresher on the topic.

Jason worked with his team to adapt to the changing expectations and tweaked the technical roadmap to incorporate the change for the new product launch initiative. Two engineers in his team really went above and beyond to dive deep into the requirements and helped launch the product earlier than expected. Also, his cross-functional partner- the technical program manager (TPM), helped Jason plan for the capacity changes and build a project execution timeline. Jason appreciated the work of the two engineers and the TPM in the team slack channel and the leadership channel by giving them huge shout-outs. He also used an internal appreciation tool in the company to recognize the work of his team members against the company’s leadership principles to help boost their morale.

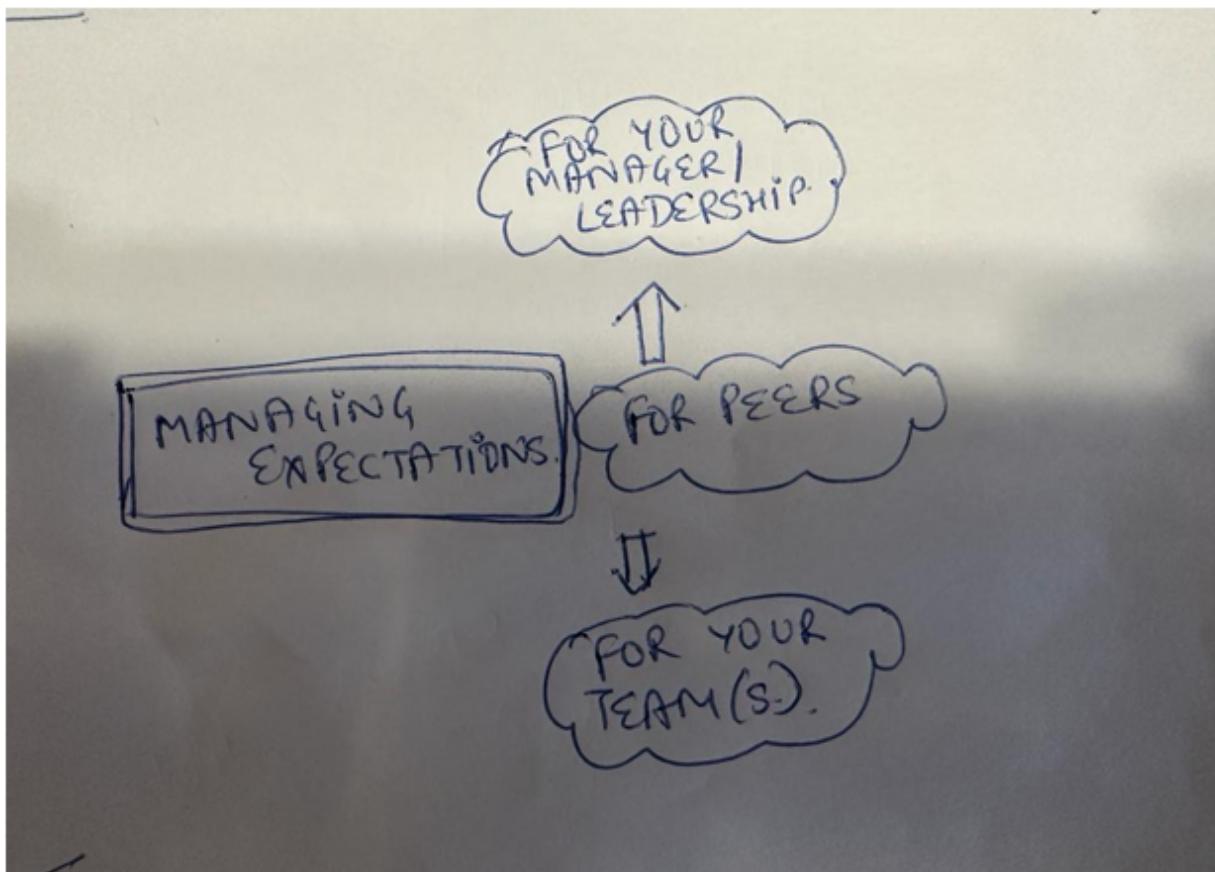
Managing expectations is an ongoing task that requires skill and patience to get it right. Above, we looked at a step-by-step framework to follow, keeping your team and cross-functional partners in mind. In the next section, we will look at how to manage expectations at all levels- with your boss, your peers, and your team members.

## 11.4 Managing expectations at all levels

Having a clear understanding of expectations at all levels of the organization, both vertically (managing up and down) and horizontally (managing peers), is vital to achieving success in both personal and professional settings. Inadequate expectation management can result in confusion, misinterpretation, disappointment, and ultimately, unsatisfactory outcomes. Conversely, when expectations are effectively managed and explicitly defined, individuals are likely to be engaged, inspired, and have a

greater chance of success. This also fosters a positive work environment and reduces the risk of misunderstandings.

**Figure 11.5 It is important to manage expectations at all levels(horizontally and vertically) to succeed**



Expectations are not just about superiors giving directives to subordinates, but rather a two-way flow of communication, where both parties can set and adjust expectations to ensure mutual alignment. For example, the person setting the expectation may task someone with a certain responsibility, but the recipient has the ability to request support to fulfill the responsibility. This type of mutual expectation management leads to better outcomes for all involved. In the following sections, we will explore expectations management with managers and leadership, peers, and team members. Let's get started.

### **11.4.1 For your manager/leadership**

Engineering management is a multi-faceted role where one needs to wear several hats simultaneously and be self-aware. Managing expectations from your leadership, particularly your manager, can be daunting at times. The idea is to set clear and realistic expectations with them to ensure there is clarity on what is expected, why it is expected, and how it needs to be done. The way you support your team members, it is a fair expectation to expect some support from your leadership and manager. Few things to keep in mind:

- Set clear expectations and realistic goals with your manager. Regular communication is the key to managing expectations. Clarify the mutual expectations shared with each other, it is always a good idea to reiterate what you understood to ensure you both are aligned. Ensure to schedule regular check-ins with your manager to discuss your progress and any issues that may arise. This ensures any concerns can be addressed early in the process and surprises can be avoided.
- Have an open communication channel between you and your manager/leadership where constructive feedback is provided proactively rather than waiting till the last minute.
- Understand your management- There are times when you might have to adapt based on what your leadership prefers. For example, let's say you have a manager who is more of a visual person, then it makes sense to present the data to them through charts and dashboards for ease of communication and clarity. Further, knowing the leadership style of your manager, helps you understand better from their shoes and give an idea of what their preferences would be.
- Keep the management team informed about any roadblocks or impediments you face instead of sitting on them. Calling out risks earlier ensures the team has the required transparency and can plan ahead for a mitigation plan, in case of a slipover. This also shows your leadership capability and skills to see through problems. Giving transparency to leadership fosters a relationship of trust.
- Learn to say ‘no’ and not be a ‘yes’ man- a lot of first-time managers struggle with this fearing they might look incompetent if they say no or

push back on asks from upper management. Overpromising and under-delivering can harm your relationship with your manager. The idea is not to simply say no, but rather share what makes you say that, backed up with objective data. This always opens up the door to negotiating to come to a common ground.

- Also, it is completely ok to ask for additional help if you feel you are overwhelmed at work. You are a manager, but that doesn't mean you need to do all the work yourself, use the power of delegation to its full advantage
- In the end, even though they are your upper level, remember the power of recognizing them and showing them how their support and involvement helps boost the morale of the team. Whatever the level, most humans prefer to be appreciated, even though their preferences for recognition can be different. To learn more, visit the chapter on 'Rewards and Recognition'.

To illustrate this further, consider the scenario where Sydney, an EM employed at an e-commerce firm, is approached by her supervisor Charlie, who informs her of an intriguing project that is one of the company's major initiatives for the upcoming quarter roadmap. Sydney is initially thrilled to hear about the opportunity, but as the discussion continues, she realizes that Charlie expects her to handle this project and the other projects her team has planned, without any additional support or resources. Given the team's pre-existing roadmap for the next quarter, taking on this project would require engineers to sacrifice their work-life balance due to a lack of resource allocation. It appears that Charlie is an exceedingly optimistic boss who is making unrealistic demands given the constraints of the situation. Therefore, Sydney must decline and negotiate to find a project that can be swapped if a new scope is introduced.

Assuming Sydney has moved into the negotiation phase with the product manager of the new project, she has been able to leverage the company's priority list to identify a low-priority project that can be swapped for the big rock project. Afterward, she arranges a meeting with her supervisor to discuss the execution plan for the project, including expected outcomes and timelines. To ensure that there is no confusion in the future, Sydney takes the

extra step of documenting the agreement she and her boss have reached. She also highlights a risk associated with the project, namely the team's dependence on another data team to source data. Finally, they agree to biweekly check-ins to assess progress on the project. Despite starting with unrealistic expectations from her boss, Sydney was able to navigate the situation and reach a mutually agreeable solution.

As we saw in the example above, it is important to clarify the expectations, raise any risks or red flags early in the process and gain alignment before getting into the execution phase. This also ensures everyone is working towards the same goal. Next, let's look at how to manage expectations regarding our peers.

#### **11.4.2 For peers**

When it comes to managing expectations with your peers, one key distinction is that neither of you will be in a direct reporting structure to the other. This means working with them is more of a partnership and cooperation where both parties work together as a team and sail together. Now, I will not repeat the fundamental framework of managing expectations, as a lot of it is common when it comes to working with peers. What can be different is if you both represent different teams, you both might have a different roadmap that each team would be working towards. Hence, cross-collaboration will need each party to acknowledge the others' roadmap and be realistic about expectations from each other. RACI matrix (refer to chapter on 'Working with cross-functional partners') and customer demos can come in handy in such scenarios where you can clearly align the roles and responsibilities of each member and party involved and know what will be the inputs and outputs for each team. Demo visuals help provide early feedback so the team can iterate on the product instead of following a waterfall model.

Let's dive further using an example. Dave and Rich are the two EMs for the pricing and promotions team, respectively. Most key projects require close collaboration between the two teams as the two pricing and promotions go hand in hand. Another key stakeholder in the picture is the marketing team

that comes with a request for a project that will help with the increase of customers, both from an acquisition and retention point of view. Now as Dave and Rich get into discussions with the product manager and marketing team, there is contention around the resource allocation for the project. Dave has fewer engineers on his team and is only willing to put one engineer on the project (with others already assigned to other initiatives). Rich, on the other hand, has suggested three engineers in his team to work on the project. The overall plan will make Dave's team a bottleneck as pricing work needs to be done before the promotion services could be implemented. What do we do next?

Rich and Dave get together to discuss and manage each other's expectations by having a candid discussion and discussing the pros and cons of various approaches. Next, they negotiate on the resources and suggest that Rich can loan one of his engineers to work on pricing to help expedite the project, with the promise of later using the engineer from Dave's team for some dev QA testing. They clearly document the alignment and, in the process, help each other to come to a common understanding. As said, managing expectations with peers is like a partnership, the idea is to get on the same page and reach the finish line. In the next section, let's look at what can be unique when managing expectations for your own team.

### **11.4.3 For your team members**

Managing the expectations of your team members is crucial for the success of the team and the overall motivation of the team members. Setting clear expectations will provide a supportive work environment to the members where they can feel a sense of belonging. Also, expectations can come from within the team too; not just from the EM. The team will have some expectations of their own for how work is to be done, and communications exchanged. These should be brought to light, and recognize that this is another source of expectations. Things to keep in mind for your team members are:

1. Don't just give one directional expectation. Your job as the EM is also to provide them with the support in the form of training resources and

guidance they need to finish the job successfully.

2. Keep an open communication channel with your team members where you also share your goals with them, this will help avoid a strict manager-employee relationship and instead foster a mutual trust relationship. Also, over-communicate wherever necessary. Keeping a bi-directional flow of information will ensure they feel comfortable asking you questions and help whenever they hit a roadblock.
3. As an EM, you act as the shock absorber for the team and let them focus on their day-to-day work. You can set this expectation with your team members, which helps build trust.
4. Since some of the team members might be new to the jobs(fresh grads) or are still early in their careers, ensure to help each member break down the expectations you have shared into actionable items that follow the SMART criteria and are aligned to the individual's career goals. Let them know how their contributions will impact the bigger picture and help with company-wide business metrics. When members understand the reasoning behind the expectations, they tend to feel more valued and vested in the tasks.
5. Look for continuous feedback from the team members. This can be achieved by sending frequent surveys to gauge the overall engagement and morale of the team. Also, ensure to meet the team members on a regular cadence to check on their progress and pitch in to provide support where needed.
6. Lead by example- This, in my opinion, is one of the most important leadership skills and can really help set an example for the team. For instance, if you would like to build a culture of work-life balance in the team to avoid burnout, set an example. As an EM, stop sending late-night emails to your team members so you demonstrate what you believe in.
7. Reinforce expectations with the team members to ensure people do not end up on a tangential path and are aligned with what was agreed upon.

Let's say you, as an EM, expect your team members to invest in building a new skill as part of their career plan. Now would simply asking them to do the trick? Perhaps some motivated individuals in the team would be proactive and identify courses that interest them and start learning, but there

will always be a good chunk of members who might not take any action. So, what do you do? You can lead your team by example by stepping up to learn a new skill yourself and share it in the next team meeting. This will demonstrate that you can walk the talk and build a stronger relationship with the team. Since we would like to set up SMART goals with our team members, let's define the criteria for this goal for the team:

**S(Specific)**- The goal is specific in the sense you would like each of your team members to build at least one new skill. Based on the interests of the individual, you can set specific goals such as doing AWS certification, doing a security certification, or learning React skills.

**M(Measurable)**- The goal needs to be measurable. Doing a certification helps quantify(score) and hence is a measurable goal. One can also set up measurable goals by providing a way to put theoretical knowledge to practical knowledge by building a small project using the skill they learned.

**A(Achievable)**- The goal should be achievable, and not unrealistic. As an example, if you expect the engineer to learn a new skill, but they already have three projects to juggle between and no time is allocated for the learning of a new skill, the goal is not achievable. As EM, you have to provide the desired support so that the team member can actually work towards achieving the goal.

**R(Relevant)**- The need for the goal should be relevant to the team member and the company as a whole. For example, if the team member works for a technology company and learns a new skill in baking class, that is not directly relevant to the work they are doing and expected to do at the workplace. On the other hand, if the member is working on cloud computing, then doing a cloud certification will definitely help them in the workplace.

**T(Time-bound)**- We should time-box the goal not to keep it open-ended. If you set a goal for learning React or doing a certification, set a time limit, perhaps 3 months or 6 months. Don't let the discussion happen without aligning on a clear end date.

At one point for myself, when in such a situation, I basically shared with my team in the team meeting how I spent time learning a new skill and getting AWS certified. I also shared my learnings and what resources I used to hone my cloud skills. This helped with open communication as the team started to brainstorm a few ideas together.

Similarly, one of the situations I faced at my workplace was an overwhelming amount of meetings scheduled for me and my engineers to attend. Some of them were definitely useful and fell into the urgent and important categories, while some meetings could have easily been handled in an async fashion through documentation. Most of the meetings that could be handled async came from the product team(we found a pattern). My team and I got together in the sprint retrospective, discussed the pain point, and brainstormed a few ideas on how to tackle and set the expectations right for the product team. We came up with the idea of ‘meeting bill of rights’, where we started advocating for any meeting invite to have a clear agenda and expected outcomes from the meeting, sent by the meeting organizer. We promoted the idea without cross-functional partners to set the expectations, and this eventually helped us get in control of the meeting's fatigue.

Managing expectations at any level involves an amalgamation of setting clear expectations and goals and also coming up with a path to achieve them. Throughout the journey, you collaborate with others to work towards the common goal and try to get to the finish line successfully. Of course, there will be impediments on the way, but using the managing expectations framework shared above can help you navigate through the situation and keep people around happy.

### **What do other leaders have to say about it:**

“In times of conflicting expectations, work towards finding common ground through negotiation and compromise. Seek solutions that are acceptable to all parties involved. Never make assumptions about what others expect from you or what you expect from them. Strive for clarity to help everyone be on the same page.”

**~ Sanjay Gupta, General Manager at HCL Technologies**

“One tip I keep in my back pocket is to make sure that stakeholders see what is being delivered at a regular cadence so that when difficult conversations need to happen about a slip or an increase in scope, stakeholders consider it to be an exception. When stakeholders ONLY interact when there is a scope problem and are only involved in escalations, they start to believe that all work is misscoped and escalated. On a previous team, this meant a “this week in engineering” email to the entire company to discuss what had been accomplished and frequent messages in Slack. When a stakeholder sees general progress all the time, it’s much easier to negotiate when a smaller set of items is off track. Additionally, make sure to communicate early. You have missed a communication opportunity if you wait until a project is way off track or at the missed date. Communicate that a project is trending off track or something is at risk of being off track.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“Managing expectations is key to everything we do. We need to be clear about what we can or cannot achieve given the resources, time, and budget. Setting expectations with our customers allows them to negotiate trade-offs and ultimately helps us build an optimal solution. Setting expectations with your team provides clarity on team culture, creates a positive workplace, and also helps inform their career paths. Finally, establishing expectations with your leaders builds transparency and trust between you and your leadership.”

**~ Nishat Akhter, Data Product & Engineering Leader, AWS**

“The biggest challenge when managing expectations is identifying your stakeholders; if you identify new stakeholders later in the project then it’s already too late as the ship has sailed. The other common challenge is being realistic while setting expectations. Oftentimes best guess estimates are used for initial prioritization due to urgency of ask, lack of time or resources. Later these are not revised, resulting in project delays.”

~ Saurabh Gandhi | Sr. Director Software Development Audible |  
Amazon | Ex-Amex

## 11.5 Stop & Think: Practice questions

1. How important is managing expectations for you?
2. Do you understand how your work/role fits in the bigger picture of the organization/company?
3. Think about a time when you had a conflict with your manager/peer/cross-functional partner because of unclear expectations? What could you have done differently?
4. What are the most common challenges you have faced when setting and managing expectations?
5. Think about a time when you helped someone in the team get to the finish line by providing support and resources to them. What was the outcome?
6. Can you think of a scenario where you failed to set the expectations clearly and the outcome?

## 11.6 Summary

- Storyboarding is commonly used by filmmakers to visualize the sequence of scenes and align expectations during the movie-making process. Similarly, setting and managing expectations early on is vital for the success of any task. Initially, expectations are set to provide clarity, but as discussions progress, there may be a need to update and refine expectations to establish a solid foundation for execution.
- Expectations greatly influence our perception, and it's crucial to differentiate between realistic and unrealistic expectations. Realistic expectations are based on data and past experiences, clarifying requirements and making them achievable within time and constraints.
- On the other hand, unrealistic expectations lack factual basis and may be overly optimistic, making them difficult to achieve within the given time frame and available resources. Unrealistic expectations can lead to frustration and disappointment and be counter-productive for people.

- Managing perceptions is critical to expectation management. This happens when bias and perception come into the way as people start predicting the behavior and outcome of future situations with the person involved, given past experiences.
- Managing expectations is an important skill set as it helps ensure all parties involved have a clear understanding of what is expected, and that those expectations are realistic and achievable. It helps avoid ambiguity, builds strong relationships, and helps plan ahead for risk mitigation.
- There are several challenges that come with expectation management, namely
  - Ambiguity - When the setter of a task expects the doer to do something but provides inefficient or incomplete information, it leads to open-ended questions and ambiguity creeps in.
  - Inability to say ‘no’ - As a human tendency, one might be intimidated to say ‘no’ at the workplace and end up overcommitting for work because they did not say ‘no’ at the right time. This often leads to burnout and frustration on the job.
  - Making it personal - Taking things personally happens when people interpret comments inappropriately or misunderstand their meaning in a conversation. This can lead to offense and hurt feelings between the parties involved.
- Managing expectations is one of the core competencies of an engineering manager. A framework for managing expectations can help one navigate through some of the difficult situations.
  - Set expectations - The first step in the process is to understand what is expected from the other person(s) and set the expectations clearly. The idea is to avoid making assumptions or keep ambiguity out of the process. At this step, clearly state ‘what we are doing,’ ‘why we are doing it,’ and sometimes ‘how we are doing’ to set the context.
  - Communicate expectations - The idea here is to communicate and sometimes over-communicate to avoid confusion later. This helps provide clarity to the tasks beforehand to avoid throwaway work. One key thing to remember is to document what everyone aligned

on so things are in black and white, and one can come back and refer to the agreement everyone aligned on.

- Negotiate, as needed - As humans, we are prone to making mistakes, whether in our initial estimations, overengineering technical designs, or allowing defective code into production during testing. In such unexpected or conflicting situations, it's important to analyze the root cause and find a solution or path forward. This highlights the need for collaborative negotiations.
- Adapt to changing expectations - Since change is the only constant, as engineering leaders, we need to hone our muscles for adapting to the changing expectations and be agile.
- Appreciate and share feedback - It's important to show appreciation for team members and cross-functional partners who go beyond their roles to manage expectations and successfully complete a project or product. Take the opportunity to express your gratitude and provide constructive feedback to support their growth in their roles.
- Having a clear understanding of expectations at all levels of the organization, both vertically and horizontally, is vital to achieving success in both personal and professional settings. This helps avoid confusion and misunderstandings and helps strengthen working relationships.
  - For your manager/leadership - when managing your manager or leadership, be crystal clear with the requirements, reiterate as needed and make sure to ask for the desired support you need to take the task to the finish line.
  - For peers - When it comes to managing expectations with your peers, neither of you will be in a direct reporting structure to the other. This means working with them is more of a partnership and cooperation where both parties work together as a team and sail together. Make use of tools like regular check-ins, RACI matrix, and product demos to collaborate with peers.
  - For your team members - Do not give one-directional expectations, instead keep the flow bi-directional and support your team members. Set realistic SMART goals with them and help them understand how their contributions fit in the bigger picture

for the company goals. Lead by example so you walk the talk and people can trust you as a leader.

- Managing expectations at any level involves an amalgamation of setting clear expectations and goals and also coming up with a path to achieve them. Throughout the journey, you collaborate with others to work towards the common goal and try to get to the finish line successfully.

# 12 DevOps and operational excellence

“It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.”

~ Charles Darwin

## This chapter covers

- Importance of DevOps and operational excellence
- Tools to get you started on DevOps
- DevOps and Operational Excellence are treated as a continuous process

You are an engineering manager for a team of eight engineers. Lately, you have seen your team struggle to meet customer expectations and deliverables. The code deployment frequency is once every two weeks, which takes forever for any change to land in production. Often the code shipped by your team is buggy and has caused customer impact. In the last month, your team had 4 critical high-impacting issues which caused a significant impact on company revenue. What do you see as a problem here?

Ok, now that you have gathered your thoughts, here we go, this could be a typical case of lack of operational procedures (while this could also be attributed to poor communication among team members or a junior engineer who is not following good coding standards, or a lack of solid QA testing, etc. As EM, you would have to identify the root cause and ensure you have checked some of the boxes to get to the root of the problem) lacking cooperation between the development and operations team and unstable systems. This is precisely where DevOps and Operational Excellence (OE) come to your rescue. DevOps refer to a marriage of development and operations to avoid friction between the two and provide greater collaboration and earlier communication between those creating and those deploying or supporting the application. DevOps, in layman's

language, is a combined form of software development and software operations. They are usually the initiatives run by the engineering team to prevent shaky, unstable systems. The mindset is to be proactive in handling issues vs being reactive. Also, once the issue has occurred, the idea is to identify the root cause to avoid bandages and fix it from the bottom so that it does not occur again. They help track progress on operational issues and hold the team accountable to raise the bar on OE. Things like improvements in the on-call process, OE runbooks/documentation, including security during development, coding best practices, throttle services where needed, defect triage exercises, reporting, and so forth all constitute the umbrella of DevOps and OE. At the same time, it does not mean the time goes and build a brand new system for each service, you have to keep the lights on on existing systems and think about improving the same system or building something new.

DevOps and OE make an organization agile and help ship products faster to the end customers. Recognize that it is not a silver bullet or one size fits all solution. What may work for one company may not directly work for the other, so it is essential to be transparent with the goals, roadmap to get there and advocate the idea of continuous improvement. Two sources I would suggest going through are the [Phoenix project book](#) (<https://www.amazon.com/Phoenix-Project-DevOps-Helping-Business/dp/0988262592>) and the [AWS blog](#) (<https://aws.amazon.com/devops/what-is-devops/>) on DevOps. In the next section, let's examine the benefits of introducing/practicing DevOps and OE.

## **12.1 Importance of DevOps and operational excellence**

DevOps helps a business to incorporate consistency, reliability, and stability into the systems and helps bring a change in mindset to think about these continuously for improvements. Steps taken in this direction benefit the engineers, the organization, and the end customer using the products. OE is defined as the roadmap toward continuous improvement. It will consist of

actionable items to help the organization improve. It helps the business grow and gain a competitive advantage against competitors. Let's look at some of the top reasons why organizations should practice DevOps and OE.

### **12.1.1 Stability of systems**

DevOps and OE help the systems to be stable and resilient for the future, especially when the security aspect is introduced (we will read more about DevSecOps in the next section shortly). Every system will eventually break down under unprepared or unforeseen load, DevOps ensures it breaks down gracefully. This helps in the present by providing ease of operations, and for future use cases where given an outage, we aim for faster recovery times. For example, let's say your services are onboarded to OE tools that can understand and analyze the memory dumps used. Now in a situation where suddenly one of your services starts to fill up disk space and is not highly available, you can quickly triage the issue by consulting the logs and results of the analysis of the memory dumps using some analyzer/profiler tools and see what areas of the service code are causing trouble. Now, once this process is adopted as a regular practice where the team ensures to run the new code through such tools to identify memory issues earlier in the development cycle, we have essentially introduced more stability and maturity in the development and operations of the services. This will help in faster turnaround time for fixing the problem.

### **12.1.2 Improved on-call**

DevOps and OE help with improved communication and collaboration with team members by bringing everyone together on the same page for this one common cause- improving the operational and engineering excellence of the organization. This promotes closer communication channels between Dev and Ops. Instead of having separate orgs that only communicate during handoff points, a good DevOps org will embed themselves with the development teams so that they are aware of what's coming, can give advice or answer questions about production operations, can clear roadblocks, prepare in advance of delivery, and so on. The steps taken (we will discuss more tooltips in the subsequent section) in this direction help stabilize the

on-call with reduced failures in production, keeping the employees and end customers happy. Also, this ensures the team members can focus on things that are important and need immediate attention by root causing the failures to prevent them from recurring. Overall, the engineering team can have a higher confidence level when releasing new changes to production.

### **12.1.3 Optimize resources and processes**

We aim to optimize resources and processes as part of the DevOps initiative. This means teams can focus on automation versus manual tests, eliminate waste of resources and identify areas where improvements can help save the company money. This also involves planning for third-party tools and infrastructure resources- whether we use cloud versus on-premise machines, create an in-house security testing framework or a third party, and so on.

### **12.1.4 Faster time to market**

One of the topmost advantages of bringing in a culture of OE in the organization is room for faster innovation and increased speed to get the products faster to the customer. OE opens the doors to proactively thinking about security, automation, shorter deployment cycles, etc that all lead to optimizing developer velocity and benefits to the services and infrastructure. Some standard incident metrics to be aware of are:

- **Mean time to recovery or MTTR:** It represents the average time it takes to recover a system from failure. The lower the MTTR, the more stable the system is.
- **Mean time between failures or MTBF:** It represents the average duration between repairable failures in an application. This metric serves to monitor the product's availability and reliability. A longer MTBF indicates a more dependable system, reflecting a higher time interval between failures.
- **Mean time to failure or MTTF:** The metric represents the average duration between non-repairable failures in a technology product.

- **Mean time to acknowledge or MTTA:** This is the average duration between triggering an alert and work commencing on the corresponding issue. It proves valuable in monitoring your team's responsiveness and evaluating the effectiveness of the monitoring system in place.

Now that we looked at some of the reasons why we should focus on DevOps and OE, let's learn some tools to keep handy as we practice DevOps.

## **12.2 Tooltips to get you started on DevOps and operational excellence**

Focus on development operations (DevOps) and OE are key indicators for the success of an organization. Having a culture of continuous improvement where people value long-term development decisions over the short term tends to exist for longer periods of time and has the extensibility to launch faster products eventually over the long run. While understanding DevOps is easy, as engineering leaders, we should ensure that at the team level and organizational level, we are making a continuous stride to advocate for it and ensure it is actually being practiced. There are several resources available online that give in-depth information on DevOps tools and best practices, so I would not repeat that. What I would provide you here are practical tooltips to get you started thinking in the direction of DevOps and introducing small guardrails that can help bring a culture of engineering excellence to the team. So let's look at some of the tooltips that can come in handy and give you a launchpad to get started on DevOps and OE.

### **12.2.1 Acknowledge and reduce technical debt**

Technical debt ([https://en.wikipedia.org/wiki/Technical\\_debt](https://en.wikipedia.org/wiki/Technical_debt)) refers to a set of short-term decisions made in deference to some other need (time, cost, convenience, code quality, etc.) or where an organization prioritizes speedy delivery of a project over project quality. This is usually a side-effect of projects that have tight deadlines or pressure from leadership to launch

them with inadequate resources. When shortcuts are taken during the design development and implementation phase, the overall architecture design suffers, which later needs to be revisited and refactored. While organizations might shy away from technical debt and, in the long run, might pay a price for it, it is omnipresent. Instead of trying to run away from it, the need of organizations is to face it and balance them with product feature tasks.

A few ways as an engineering leader, you can play your part is by active participation in the tech design process and asking the right questions to your team(s). A few questions could be:

- Are we introducing any duplicate code?
- Is the code modular and extensible for the future?
- Has the code been reviewed by senior engineers?
- Are there any other design implications or blindspots we could have missed?
- Have we prioritized the right set of tasks and introduced guardrails to control code quality?
- Are we laying emphasis on test coverage?
- Are we producing good documentation for knowledge sharing and posterity purposes?
- Are we introducing more latency or friction for our customers with the new feature changes?
- How much time will it take to make a change to this code and deploy it in production?
- By making this choice, what are we negatively impacting elsewhere in the system?
- Is there a financial cost to this decision today that will only get worse or increase in the future?
- If we roll this out today as-is, when will it need to be refactored or rewritten? Is there a drop dead date (i.e., an end of life of a product or system or a supported version that now goes out of support)?

These are some of the sample questions to get you started thinking about engineering and OE. Once you have tried answering the questions above,

tech debt tasks should be treated like any other product tasks, where a specifically assigned capacity is kept aside each sprint to work on the prioritized OE backlog. This ensures the entire team is aligned and working towards reducing technical debt. An easy way is the prioritized backlog of tasks from which the team members can pull a few tasks per sprint. Another thing that I have seen work well is forming an architecture board at the organizational level composed of senior engineers from multiple teams. If any team is planning to work on a new feature or component, they can present the technical design to this board of senior engineers and get proactive feedback before getting into the implementation phase. This helps to ensure best practices and consistency bar is being followed across the teams.

Let's say your team is tasked with adding a new dynamic pricing feature for the e-commerce department for your team where based on the competitor's price, your service will provide a dynamic price to lure the customer. Now your entire pricing architecture is a monolithic legacy system which has been a pain to work on (Note: no architectural style is superior, it is the problem domain that decides the type of architecture). You have a choice now, whether to go ahead and add this new functionality to the old monolithic structure or think about future extensibility and create a new dynamic pricing micro-service. Depending on the scenario and resources available, you will negotiate delivery timelines and try to avoid technical debt in the already existing legacy system. This will allow you in the future to add more enhancements to the service with ease if you go the microservice route. On the contrary, if for reasons you decide to go with the monolithic legacy architecture, you are adding technical debt to your systems.

This is a common example of how technical debt and OE play their part. If you make the right choices, you have good OE, wrong choices can leave you with poor OE. Even spending time on documenting a wiki or document for onboarding new hires to your team can be categorized as an OE/engineering excellence task and help the team in the long run. Technical debt is like a health check for how a team is doing. Ensuring engineers do not implement a design feature and move on without recognizing it,

accounting for it, or fixing the debt or hacks(fire and forget) is the responsibility of us as engineering leaders.

### **12.2.2 Identify an operational excellence ambassador**

OE is very important for the team, but that does not mean you, as an engineering manager, can do it all yourself. At the same time, it is not something that only a single engineer in the team can take the burden of. What you do need is someone to work with you to help prioritize the OE backlog and ensure the team is making progress every sprint by sprint. This is where the concept of OE ambassador or champion comes into play. This simply means identifying an engineer in the team who is passionate about OE and can keep the ball rolling forward. As EM, you need to align with this individual on the responsibilities of the OE ambassador, such as ensuring OE tasks are groomed every sprint and a prioritized backlog is maintained. A lot of times, smaller OE tasks can be picked up by the on-call person as well, and that's another way to move faster with addressing the OE backlog of items. As an engineering leader, you can:

- Help identify an OE ambassador/champion for the team
- Work with them to keep a prioritized OE backlog that is groomed and some tasks picked up every sprint
- Provide the desired coaching and mentoring to the OE ambassador
- Bring in a culture of regular operational reviews with the team to align all members and collect feedback
- Review metrics and assess progress
- Hold retros to review and improve in a continuous manner
- Eliminate roadblocks to successful execution
- Help nurture the OE best practices in the team by advocacy and showing the importance of it

Getting ahead of OE with the help of an OE ambassador can be fruitful and helps promote an engineering excellence culture in the organization. Next, let's look at how DevSecOps developed from DevOps and how to get ahead of it.

### 12.2.3 DevSecOps

The tech industry has undergone a significant cultural shift with the emergence of DevOps, which refers to the convergence of **development** and **operations**. This union has led to significant progress in establishing a relationship of mutual trust between the two areas. At the same time, security threats and breaches are common in the workplace and in newspaper headlines. [Gartner's research](#)

(<https://www.gartner.com/smarterwithgartner/is-the-cloud-secure>) suggests that the majority of cloud security failures until 2025 will be caused by the customer's fault, with an estimated 99% of failures attributed to them. As a result, developers should ensure that their applications do not leave any gaps that could make an enterprise and its customers susceptible to such attacks. [1] In light of the growing number of security threats and breaches, enterprises should prioritize application security from the beginning of the development process.

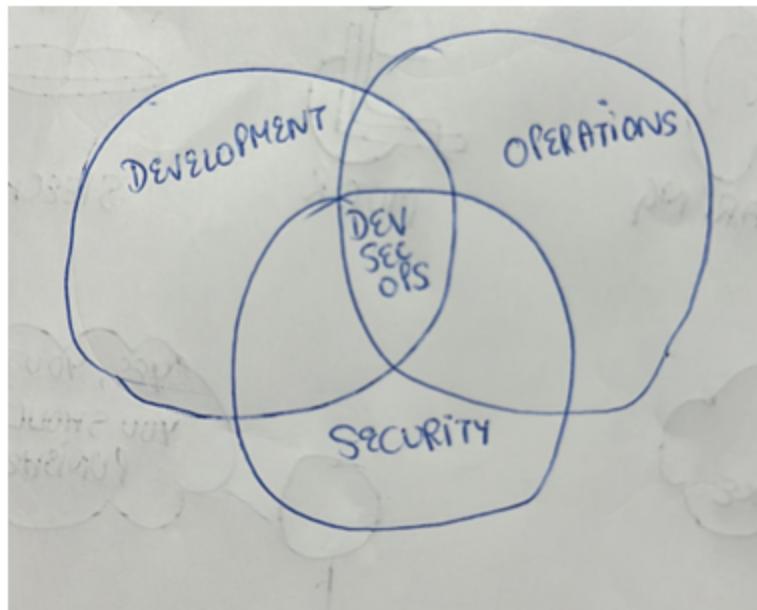
#### Did you know?

[Open worldwide Application security project \(OWASP\)](#) (<https://owasp.org/>) is a nonprofit organization that focuses on the security aspect of applications and software. It is free to be sued by all and published top security threats and ideas on how to combat them to make your web applications more secure. Check out OWASP (at least OWASP top 10) to learn more about the security aspect of the application and how you can ensure there are no loose ends in your applications.

DevSecOps is an extension of DevOps where we include the security component along with DevOps. DevSecOps emphasizes proactive security considerations during the development process rather than reacting to security issues after they arise. This is also called the “shift left” of security at the earliest stages of the development cycle. As part of the development cycle, security testing and bug fixes are integrated to detect security vulnerabilities early in the software development lifecycle. This approach

has facilitated innovation, streamlined developer velocity, and enabled rapid release cycles while prioritizing security.

**Figure 12.1 DevSecOps combines security with DevOps to ensure the development of a secure product**



One such food for thought for proactively addressing security threats in the code is making use of security tools, such as [Fortify](#) ([https://en.wikipedia.org/wiki/Fortify\\_Software](https://en.wikipedia.org/wiki/Fortify_Software)) or [Veracode](#) (<https://www.veracode.com/security/static-analysis-tool>), early in the development cycle. These allow for static application security during the development phase and can address security threats early in the implementation phase. Some companies have built internal tooling to address security threats as part of their code deployment pipelines which is also a great way to address potential security breaches. Another way is to do a detailed application security review before the launch of any production service. This is usually done by security engineers in the company to look for loopholes in the system and ensure the customer's confidential data is well protected. Embarking on the adoption of DevSecOps for an enterprise is a lengthy journey that spans several years,

and initiating it at an earlier stage will prove beneficial for the organization in the long term.

#### **12.2.4 Test coverage**

Test coverage is an important component of DevOps that ensures the code is well tested (be it manually or in an automated fashion) and is reliable. Test coverage impacts the code quality and reliability of the code deployed and helps raise the bar on the OE front. Further, it helps give confidence to the team during on-calls and that the changes are safe for the customers (reduced likelihood of introducing bugs in the code). It's also a good tool for EMs to assess the quality state the team/code is in. Please note that the absence of code coverage indicates a problem and risk in making changes without a safety net! Meanwhile, the presence of code coverage doesn't guarantee that the code is bug-free. It is better to know at least where you are, so you can see if things are (minimally) improving or getting worse. It also enhances the development process, where bugs/defects can be caught early in the development phase.

Test coverage commonly includes line coverage(amount of lines of code covered through tests), branch coverage(ensures all branches in the code are at least tested once), and method or function coverage(this checks if the method was entered at all during test execution).

Some common types of testing are:

- **Unit testing** - These are the tests written to test the functionality of the smallest unit, such as the method or function block. These should be must-haves and be part of the definition of done for a task assigned to a developer. As an engineering leader, you can work with your team to identify testing standards such as 90% unit testing code coverage for any new code deployed to production.
- **Integration testing** - Integration tests are a superset of unit tests and involve the end-to-end integration of the systems. They are more real to customer experience and can help mimic customer actions and

interaction with the system. This helps to ensure there is no system breakage once the basic building blocks are integrated together.

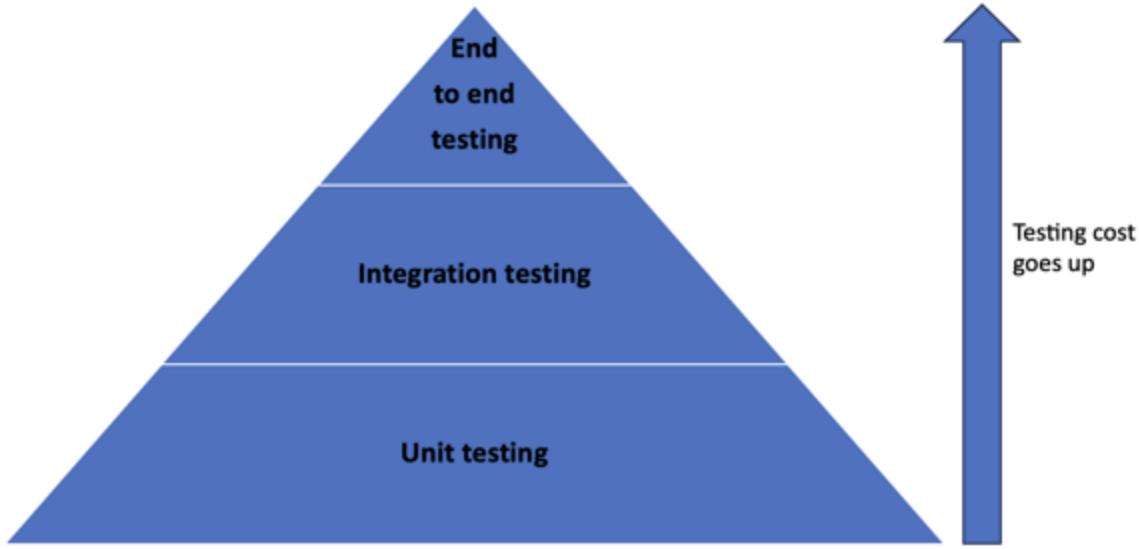
- **End-to-end testing** - End-to-end testing examines the extensive codebase encompassing the entire application. By simulating real-world production functionality, it validates the system's behavior as a whole. However, it comes with the highest maintenance costs and operates at a comparatively slower pace than other testing approaches.
- **Load testing** - Load testing includes testing where loads from sample customers are generated on the system to mimic production-like traffic. This helps the development team to understand if the system is scaled well to handle peak traffic in times of heavy usage. Some companies call it [gameday testing](#) (<https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.concept.gameday.en.html>). This helps identify any bottlenecks or single points of failure in the system and helps prepare the system for better scalability and reliability.
- **Performance testing** - Performance testing is somewhat related to load testing, where the load generated in the system focuses on the performance aspect of the software application. For example, let's say you generate a load of 1 million customers coming to your application and downloading a short video. Now, you want to measure how this load impacts the system's performance- such as increased latency, responsiveness, and download speed to give an example. Performance testing reflects the code quality of the system and if it is optimized for heavy usage of the system.
- **Regression testing** - Regression testing is a form of testing to ensure new code changes do not impact or break the existing functionality of the system. It can include both functional and non-functional tests that are done to verify the new changes. These are must-have tests for any new big change-making to production. This is usually done by a combination of software engineers and the QA for the team.
- **Sanity testing** - Sanity testing is like a subset of regression testing that focuses on testing the new set of code changes being deployed. This is to ensure the application works as expected and the new functionality behaves as expected.

- **No harm testing** - "No harm testing" refers to a testing method where teams check that the introduction of a new feature does not have any negative impact on their metrics. Often used in product experimentation, this approach involves releasing the feature gradually and monitoring metrics at each stage.
- **User acceptance testing (UAT)/Beta testing** - Also referred to as dogfooding sometimes, this testing is one of the last forms of testing that is done by a subset of real customers to test the functionality and play around with the new software. This is closest to real-world testing as, at this point, you have basically handed over the product to real users to test and confirm the robustness and utility of the software.

### Did you know?

[Test automation pyramid \(<https://www.headspin.io/blog/the-testing-pyramid-simplified-for-one-and-all>\)](https://www.headspin.io/blog/the-testing-pyramid-simplified-for-one-and-all) is an industry-wide framework to develop robust high-quality software. It helps define the types of tests that should be included in the automated test suite. It plays a critical role in times of regression testing as well. It helps with time and cost savings due to automated testing and helps remove human error.

**Figure 12.2 The test automation pyramid helps the engineering team to produce high-quality software**



It works at three levels, namely, unit testing(base step), integration testing(mid-step) and end-to-end testing(top of the pyramid).

The aforementioned testing approaches serve as protective measures and quality checkpoints for the modifications made by your engineering team before delivering them to end users. You might be concerned about the added scope these testing methodologies introduce to your current project. Indeed, that's where the significance of automation testing becomes apparent. Automation testing enables us to create test cases once and then rerun them using test tools and frameworks in the future. Although there is a substantial initial development investment, the benefits lie in its repeatability and reusability for subsequent iterations with minimal to no manual intervention required.

### **Did you know?**

[Feature flags/toggles/switches](https://www.atlassian.com/continuous-delivery/principles/feature-flags) (<https://www.atlassian.com/continuous-delivery/principles/feature-flags>) is a powerful software development technique to control features in production by switching them on or off on a need basis. This is especially useful to understand customer usage trends to decide between one experience over another and allows the power of experimenting with customer behavior. This allows the engineering team to

control features as toggles, so as a basic example, you can control showing a Submit button in one experience and not the button at all in the other experience. This also acts as a kill switch if the feature is not working as intended in production. Major advantages of the technique include more power in the hands of the development team, reducing risks, iterative launch of features, and performing analytics on customer behavior. Feature flags are often tied into [A/B testing](#) ([https://en.wikipedia.org/wiki/A/B\\_testing](https://en.wikipedia.org/wiki/A/B_testing)), so you can direct some percentage of users towards one flag state (on or off) and some percentage to the other. This gives you that customer behavior analysis too.

A famous industry example of test coverage is the Netflix tool Chaos Monkey which uses the concept of [chaos testing engineering](#) (<https://netflix.github.io/chaosmonkey/>). The chaos monkey tool will randomly terminate production machines to test for resilience and failover mechanisms in the applications. Of course, this needs a controlled environment and engineers to execute this as it takes place directly in production. But this helps prepare the development team to identify issues earlier and prepare for large-scale outages that are unexpected events in advance. Next, let's look at how one can check operational costs.

### **12.2.5 Reduce operational cost**

Another key aspect to look into when thinking about OE is operational expense or opex. This can include infrastructure costs (cloud spending, on-premise machines), employee pay, travel and perks for employees, office space, third-party tools and vendors being used, and so forth. This ensures the company is making the right investments and spending on areas that can help maximize profitability. Work with the expenditure team to conduct a detailed audit of the expenses, and identify areas that have the potential to reduce cost- maybe you do not need a specific third-party tool or have an opportunity to renegotiate the contract, maybe your services are using extra memory leading to increased cloud expenditure and so on. Most importantly, advocate for optimizing operational costs to bring awareness to the team and organization and have people volunteer on this initiative.

## 12.2.6 Managing Deployments

Deployments are crucial to taking our code from the implementation phase to production. While we all have heard about the emphasis on continuous integration and continuous deployment and it being a Northstar for most companies, it is important to focus on safer deployments. Safer deployments simply mean that deployments are bug-free, do not cause any regression in production, have no quality or security concerns, and at the same time, the deployment window is time efficient. Higher code test coverage can organically help with safer deployments.

As the engineering manager for the teams(s), you can try a few of the tools from the below list and see what works best for you and your team(s).

- **Code Reviews-** Code deployed to production should go through a proper code review cycle where peer engineers review code, and provide comments to ensure another pair of eyes. This helps identify any blindspots that the original engineer might have missed. They can be done in person(live with humans, whether remote or local), or they can be done asynchronously through various software tools with code check-ins and pulls to be reviewed and shipped/passed by an ‘x’ number of engineers before pushing the code to production.
- **Deployment in phases -** Production deployments should follow a phased approach. That means the code should first be deployed in early staging environments like alpha/beta, then moved to pre-production environments that mimic production and eventually to production. Between the stages, we can include bake time (let's say the code stays in pre-production for 24 hours before it is moved to production to ensure stability) to ensure a gradual rollout of changes.  
[Canary deployments](#)  
(<https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.concept.canary-deployment.en.html>) where the deployment also happens in stages and is first released to a set of customers, and based on feedback and stability, the changes are moved to all other customers is also another way to safer deployments. We can also use

feature toggles to enable the feature for a few customers and then enable it for all. If there is an issue, we can rollback with ease.

- **Communication** - Communication is the key to any small or big change making to production. The teams can send proactive communication to all team members, cross-functional partners, and other stakeholders about the planned deployment of changes, dates, and what to expect (in case any downtime might occur). Some companies have the role of release manager, who is tasked to coordinate the overall communication and deployment for production. This ensures everyone is aware and knows of a pending deployment, and if something goes during that time frame, they know who to reach out to. Similarly, once the deployment is complete, another communication can be sent to the audience, sharing the outcome of the deployment (success/failure) and reasons along with the release notes for the changes. You can also create chat rooms during deployment cycles so people can bring faster attention to any bugs they encounter during the rollout period. This way, your product team won't be surprised or stay in oblivion for when a particular change is being rolled out to production.
- **Deployment checklists** - A deployment checklist is a comprehensive document detailing the modifications intended for production, encompassing code or infrastructure changes, along with the responsible individuals, deployment date, and time. It includes preparatory measures taken before deployment, an in-progress checklist, and post-deployment tasks. Additionally, the document outlines the types of testing conducted on the alterations to ensure confidence before implementing them in the production environment. The checklist acts as a blueprint reference for the changes to align everyone on the same page and also can go through approvals to ensure we have someone overseeing the changes. These checklists can further be integrated into issue tracker systems and automated. For example, let's say you have your deployment and rollback checklists embedded within Jira. So if the checklists are not filled out, the Jira ticket cannot be promoted to the next step in the workflow or can't move to the next stage of the pipeline. In any case, the deployment is

happening in the production system, so you want to be fully serious and not keep any loose ends.

- **Monitoring dashboards** - Dashboards serve an important purpose-bring transparency and insights for all. With the dashboards in place, the deployments can be managed well by keeping an eye on any changes in metrics as the code makes its way to production. At the same time, alerts should be in place so that the on-call or engineers can be informed early about a suspected issue with the new changes deployed.
- **Quality gates/ Pipeline blockers** - Sometimes introducing quality gates and blockers in the pipeline are helpful to ensure high-quality code is deployed to production. Pipeline blockers can include test coverage rules (90% and above test coverage for code deployed), date and time blockers (for example, no Friday deployments or off-office hour deployments), product launch dates (we want to avoid any code deployment during the day a new product is launched), etc. These can act as guardrails to protect any changes made into production if they are not ready or the timing is off. Some things to read about are:
  - **Software composition analysis (SCA)**: An application security methodology that enables development teams to swiftly monitor and examine every open source component integrated into a project.
  - **Software application security testing (SAST)**: It is the static automated code analysis, whether compiled or uncompiled, conducted to detect security vulnerabilities. It parses the code to look for vulnerabilities.
  - **Dynamic application security testing (DAST)**: This involves actively interacting with your operational application to discover and handle potential vulnerabilities it may possess.
- **Rollback procedure** - Rollback procedures refer to a set of actions to be taken in case a deployment made to production causes unwanted customer impact and needs to be restored to its previous state as soon as possible. These steps should be part of the deployment checklist and kept handy even before the deployment begins so that one can take timely action on a need basis. If there is a customer impact, then the rollback procedure should be triggered to restate the system to a stable

state. Examples of such scenarios can be if a deployment caused duplicate records to be added to the customer table. A rollback procedure for such a scenario can include a set of database queries written to identify duplicate records and delete them.

- **Automated deployments** - As it is said, to err is human, we all can make mistakes, especially when the task involves a manual set of tasks to be performed in a specific sequence. Automated deployments of continuous integration and continuous deployments help remove human intervention/errors and can help with faster, safer deployments to production. As engineering leaders, it will be beneficial to set goals for your teams to raise the bar on automated deployments.
- **Regular deployment cadence** - We want to push changes to production and push them often. The biggest advantage of this approach is that it avoids piling up changes in the prep-production environment and avoids a big-bang deployment to production. A big bang deployment has been shown to cause more confusion if an issue is encountered to isolate what could have caused it. Hence, with a DevOps mindset, get together with your team to identify a regular cadence of deploying changes to production.

Managing deployments is important and crucial in aiming for deployment safety. Using some of the best practices shared above, you can find a model that works best for your organization. To sum up, as one of my leaders said, on-calls and deployments should be boring tasks - you deploy changes, and everything just works as expected.

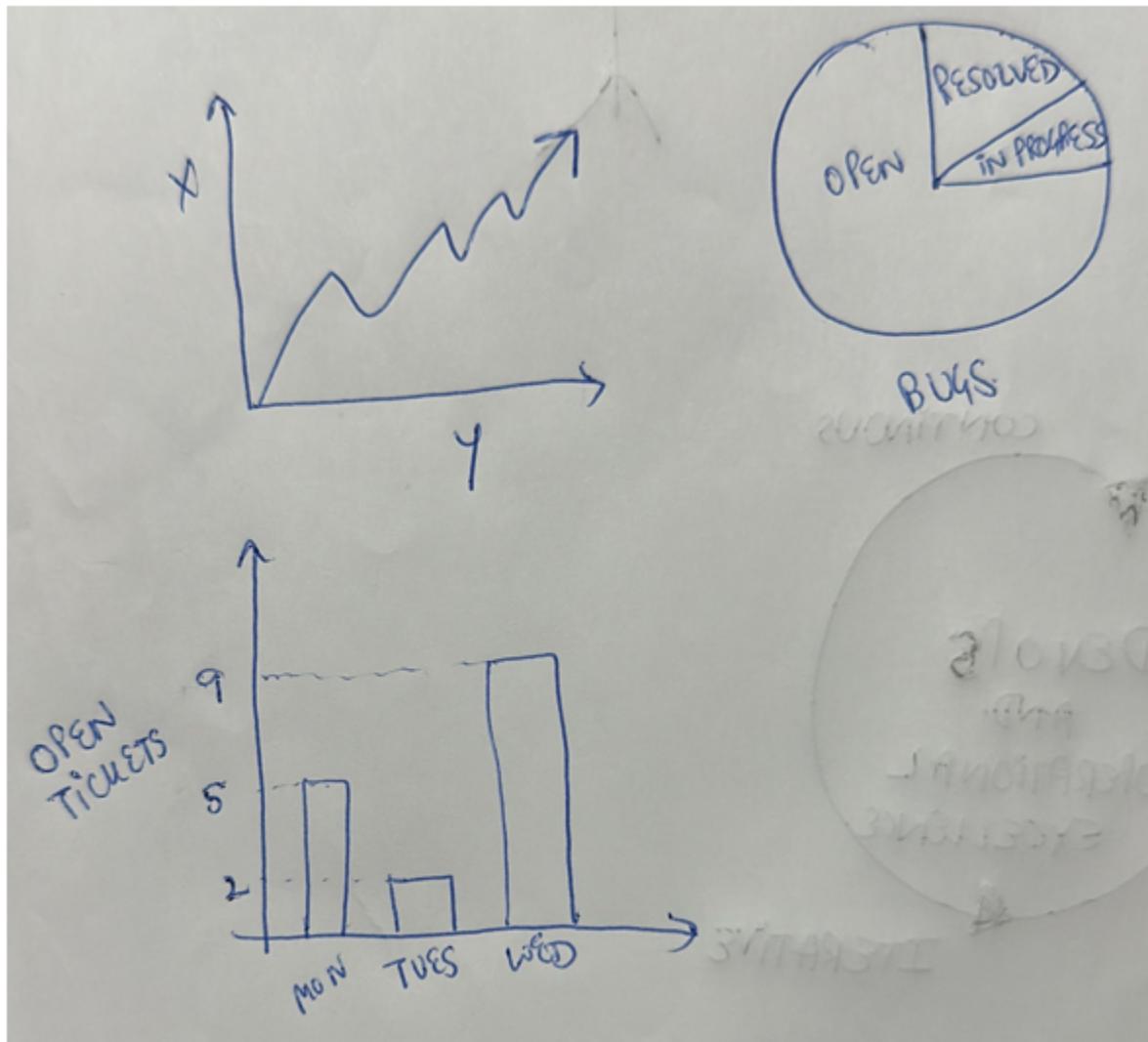
### **12.2.7 Logging, monitoring, and dashboarding**

One of the critical components of any software development process is the logging and monitoring system that powers it. Such systems help with transparency to get visibility into the overall health and performance of the system. It is especially useful to identify any bottlenecks in the system by tracing the issue and triaging a customer-impacting issue. Logging systems store detailed information about actions performed, by whom, date and time stamp, system errors, and help with troubleshooting. Monitoring tools help

with detecting outages and taking steps to mitigate customer impact. Some of the tools that can come in handy are:

- **Alerts and Dashboards** - These include dashboards created to provide insights into the overall health of services, gain insights into customer usage trends and help set up monitors and alerting systems in case the values reach a particular threshold. This is specifically useful to identify a problem and later do a root cause analysis on the issue. Once the issue is mitigated, they help to ensure the values go back to the normal expected range and the system is back to a healthy state.
- **Runbooks and documentation** - None can deny the benefits of better documentation detailing the services and features along with technical design diagrams to help gain insight into how different system components interact. Runbooks are similar in nature in that they contain information about ownership, as the overall system can have components owned by different teams. You can even have alerts from monitors pointed to the runbook for faster debugging of the issues.
- **Service level agreements (SLAs)** - SLAs help define the expectations on services expected, such as turnaround time in case of customer issues, mean time to resolution, or something like the number of days taken to respond to a code review. This helps set a clear agreement between the parties involved and especially is useful in cases where the team is working with third-party vendors. This also involves ensuring each party maintains an agreed-upon standard of procedures and a bar for the services provided for quality control.
- **Reporting summaries** - Transparency is important at all levels, sharing it with your team in terms of where we are and the progress made, and at the same time, sharing it with leadership to get the visibility your team deserves. Reporting summaries come in handy in such scenarios to provide snapshots of progress made and celebrate the work of individuals. This provides a visual representation to leadership to also analyze what areas to focus on and invest in.

**Figure 12.3 Monitoring and dashboarding help provide visibility into the health of the systems**



Efficient logging, monitoring, and dashboarding allow for engineering teams to make data-driven decisions to enhance security, reliability, and customer experience. Some tools that teams can explore are [OpsGenie](https://www.atlassian.com/software/opsgenie) (<https://www.atlassian.com/software/opsgenie>), [Splunk](https://www.splunk.com/en_us/home-page.html) ([https://www.splunk.com/en\\_us/home-page.html](https://www.splunk.com/en_us/home-page.html)), [AWS X-Ray](https://aws.amazon.com/xray/) (<https://aws.amazon.com/xray/>), [page speed insights](https://pagespeed.web.dev/) (<https://pagespeed.web.dev/>), [New Relic](https://newrelic.com/) (<https://newrelic.com/>), [Veritas](https://www.veritas.com/) (<https://www.veritas.com/>), [AppDynamics](https://www.appdynamics.com/product/cloud-native-application-observability) (<https://www.appdynamics.com/product/cloud-native-application-observability>) and others depending on the use-case.

## 12.2.8 Away team work

[Away team](https://pedrodelgallego.github.io/blog/amazon/operating-model/away-team-model/) (<https://pedrodelgallego.github.io/blog/amazon/operating-model/away-team-model/>) work refers to a work agreement between engineering teams where one of the team(called the away team) implements features in the codebase owned by another team(called the host team). This concept addresses any resources or time constraints that hamper the delivery of a critical project. Of course, the first attempt by the away team will be to get the work done by the host team as they are subject matter experts of their codebase and might be able to do the work faster. But most times, the roadmap of one team can be quite different from the roadmap of the other team, and accommodating more work with the given resources might not even be possible. Hence, in such scenarios making use of an away team model comes to the rescue. This model is heavily used in companies like Amazon where several independent teams own different services, and other teams can have work dependencies on each other. But with away team power comes great responsibilities. It does not mean the away team can simply go ahead and change the codebase for the host team.

**Table 12.1 A sample away team model agreement**

<b>Host team + point of contact(POC)</b>	Team Alpha, POC: Dave Chen
<b>Away team + point of contact</b>	Team Beta, POC: Frank Ruth
<b>Project/Initiative</b>	Project X
<b>Timelines</b>	Jan 15 2023-April 6 2023
<b>Architecture</b>	<Link to document>

<b>design document</b>	
<b>Changes to expect in codebase</b>	The away team will implement a new API in service ABC to return a list of items ordered by a particular customer.
<b>SLA for response from host to away team</b>	2 business days
<b>Coding best practices document</b>	<Link to document>
<b>Code deployment ownership</b>	Away team and host team will have a joint ownership for deployment, led by host team on-call
<b>On-call handoff</b>	The handoff will include: knowledge transfer sessions conducted in the end week of March. The away team will be part of a secondary on-call for 2 weeks before a full hand-off to the host team.
<b>Escalation Route</b>	SDE-> Team on-call-> EM-> Director
<b>Meeting notes trail</b>	Jan 22, 2023 - Architecture design document started

Jan 15, 2023

- Kick-off meeting conducted to align on away team model

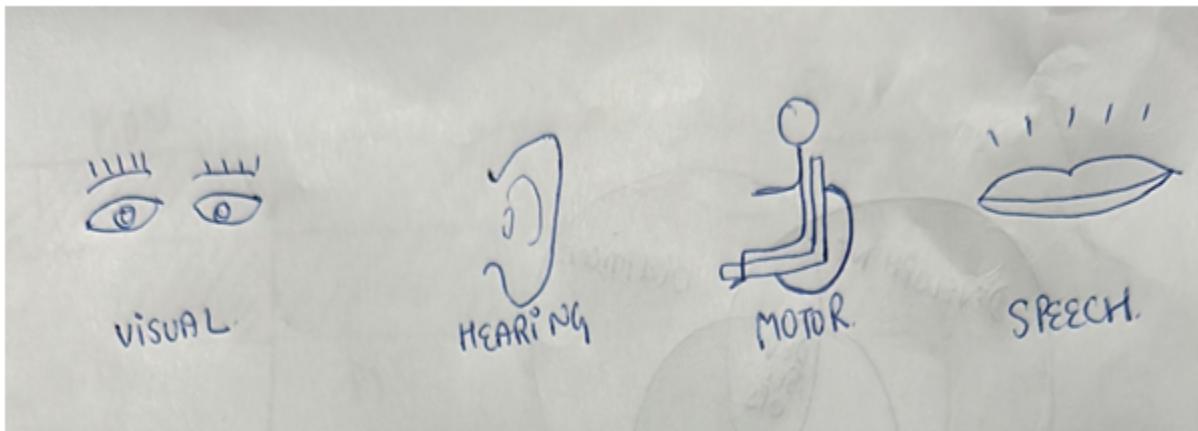
The work is planned and agreed upon using an agreement between the two teams. Usually, a document should be prepared that details what has been agreed upon, timelines, the type of changes to expect in the codebase, who will have access from the away team to make the code change, code review mechanism, who will review the architecture design from the host team, etc. Such guardrails ensure the away teamwork does not impact the code quality bar set up by the host team and that the code/service integrity is maintained. The away team model really empowers the engineering teams to execute on their roadmap without blockers from dependencies on other teams. There will be a knowledge transfer component involved here as the ownership and on-call of the service remain with the host team, so it is important to produce detailed documentation of changes made in the codebase and how to troubleshoot them in case of a high-severity issue. As an engineering leader working with a big company or a startup, if you see several instances of dependencies on other teams that seem to block your team roadmap, introducing an away team model might help you move fast and is a good opportunity to bring everyone together to unblock each other.

### **12.2.9 Accessibility**

Accessibility ensures every individual, irrespective of their status and abilities, gets access to the software products that meet their needs. It is empowering each individual, even if they are especially able, to be able to use the product by identifying barriers or friction points and working on removing them. Improvements done on the accessibility front really send a positive signal to our customers and ensure they trust our product. Accessibility is something that's mostly deferred or ignored when developing software and hence is crucial to reach out to a larger audience and maintain an engineering excellence bar. As shown in the figure,

accessibility can take various forms such as visual, hearing, motor skills or speech.

**Figure 12.4 Accessibility can come in various shapes and sizes and is a critical component for end customers**



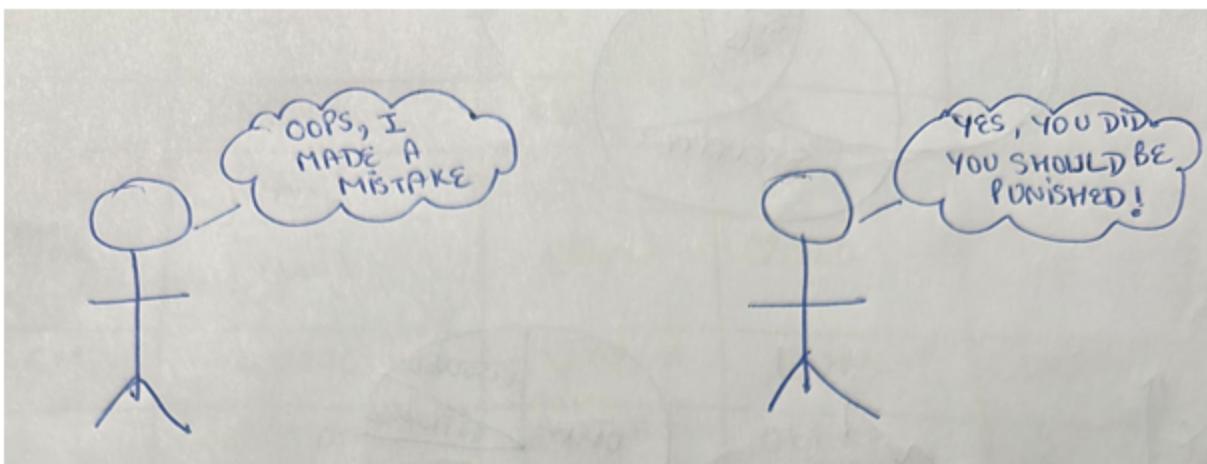
As an engineering leader, it is on you to advocate for it, whether it means providing training resources for your engineers and using forums to spread the word. As an example, suppose you are building a web application, it is important to follow the laws and regulations enforced around accessibility. Next, from a product perspective, you can think about providing the functionality of screen magnifiers to zoom and/or screen readers to help the visually impaired audience. Accessibility helps improve your audience reach and is a critical component to be included in software development to help raise the bar on DevOps.

### 12.2.10 Learn from mistakes

Unexpected things are bound to happen. Let's say a deployment to production was planned. You followed a deployment checklist and made sure proper testing was performed, but when the code hit production, customers were unable to render the web page. What do you do next? Well, we learn from our mistakes. The idea is to immediately jump on the issue, perform a root cause analysis, assess the customer blast radius, and, once the issue is mitigated, document the troubleshooting steps. Next, most

companies advocate for learning from the mistakes by creating documentation to understand the root cause/postmortem of the problem and answer the why behind it using the [5 whys approach](#) ([https://en.wikipedia.org/wiki/Five\\_whys](https://en.wikipedia.org/wiki/Five_whys)). This then feeds into the next set of action items the engineering team can take to avoid having this issue repeated in the future. The idea is to avoid blame games and instead focus on the learning aspect of the issue at hand. So go for failure fast, learn fast mindset.

**Figure 12.5 One should stay away from the blame game and rather focus on learning from mistakes**



### Did you know?

The [5 Whys approach](#) ([https://en.wikipedia.org/wiki/Five\\_whys](https://en.wikipedia.org/wiki/Five_whys)) is a famous iterative approach to help with root cause analysis and get to the bottom of the problem. It is generally considered sufficient to go to five levels of why to get to the cause, although one can go deeper as well as need be. As a basic example, let's say one of the company's coffee machines broke.

- 1) Why? - The coffee filter was not cleaned
- 2) Why? - The service team did not show up for the weekly cleaning

3) Why? - The company is focussing on cost savings and perhaps the bill was not paid

4) Why? - The coffee service company had jacked up the prices with the new year

5) Why? - There is a shortage in the workforce at the current coffee servicing company - Root cause

As you can see, the approach helps identify the root cause with an iterative set of questions in the form of a way to get to the bottom of it. Similar techniques can be applied in software development to learn from mistakes and the root cause of the software issue.

With the above tooltips in your arsenal, as an engineering leader, you can help introduce and mature the processes around DevOps and OE. This will help bring a culture where it's not a few sets of people running the DevOps initiative, rather as a team/organization or business unit, people make it part of the work culture. Next, explore how OE should be viewed as a continuous iterative process.

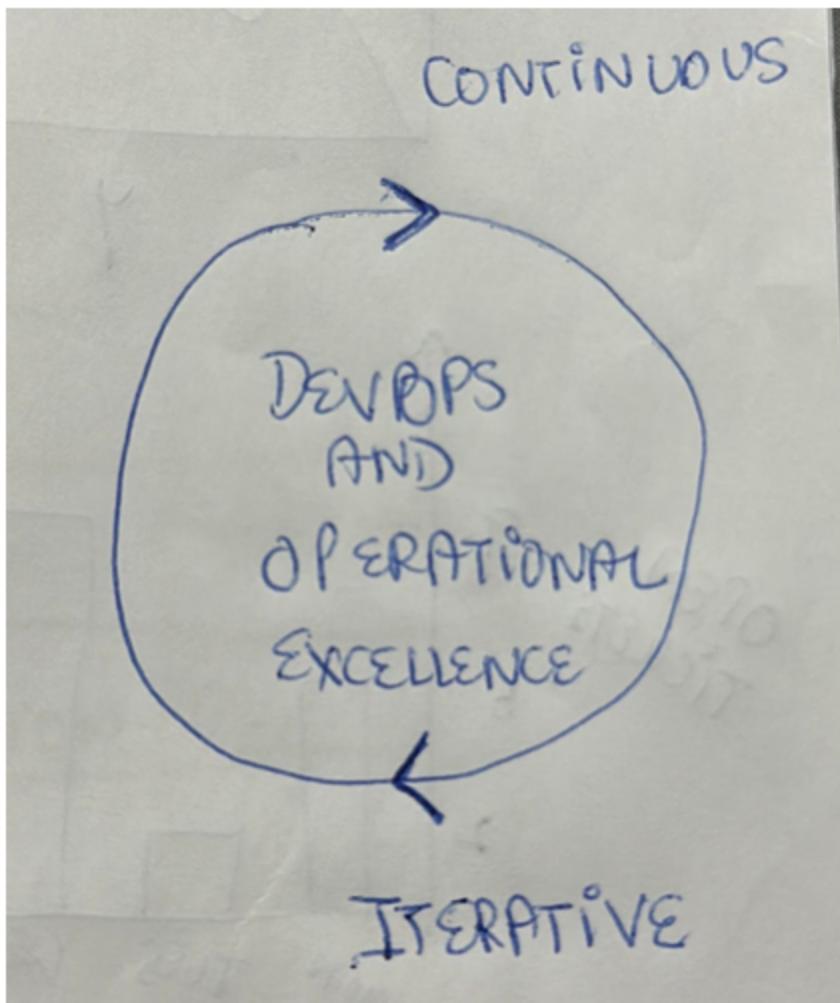
## **12.3 DevOps and operational excellence as a continuous process**

Adopting a DevOps and OE culture is not something that happens overnight. It is a strategic commitment and mindset change that is a continuous learning process. The idea is to continue to ship high-quality products and services at a faster go-to-market rate, along with a focus on operational stability, security, and monitoring of them. During the strategy-building phase for DevOps with your OE champions/ambassadors, your focus is to optimize processes, identify single points of failure, and optimize for cost and resources. Now this is not a waterfall model, rather in the process, you gather feedback from the engineering team and end users and go back to strategize again to incorporate the feedback. Working with OE ambassadors lets you create an operational roadmap and keep a prioritized backlog of OE tasks which can act as a working queue from which the

engineering team can pick up tasks during on-call or sprint-assigned OE capacity. Hence, consistency and continuity is the key to OE and DevOps. This also involves providing resources to your engineering team to learn industry trends and hone their skills. Remember, the idea is not for your team member to only be fit for the team, but rather fit for the organization and the industry.

Let's take an example to understand this better. Like any other engineering team, let's say your team follows the rules of having two engineers from the team review code, ship it/provide the green light before the code is merged into production. Now as part of an OE initiative, a team of engineers develop a plug-in integrated with a free open-source static code analysis tool that can proactively detect code bugs and code smells during the code implementation phase itself- what a wonderful plugin! To note, tools, frameworks, and plug-ins should be approved by the Infosec team to ensure they meet industry security standards. This can really help ease the code review process as the developer can first address the basic code bugs and smells called out from the static code analyzer and then send the code for review to other engineers- saving time and energy. Now, this operational process was introduced, and the team started using the plug-in. One month into using it, you, as an engineering leader, send out a survey to the teams using it to gather feedback on the plug-in and the process. What you find out from the survey is that the tool has several false positives where it detects code bugs and smells that are really not bugs or smells. This is causing the developers to spend more time on implementation as they now have to sieve through the false positives and ignore them before sending code for review which has added more friction in the development process. Next, you take this feedback and iterate with the original engineers to see how the pain points could be addressed by continuous learning. The engineers were able to deploy a fix where an engineer could mark a detected false positive with a flag in the UI, and it will be stored so as not to prompt again for it in the next iteration. This greatly improved the efficiency and addressed the pain point called out, and helped the team use the power of the new operational plugin and process in place to its fullest power. This is an example of OE being a continuous process, even on the smallest of perceived improvements, also shown in the subsequent image.

**Figure 12.6 DevOps and operational excellence are continuous and iterative steps for a better experience for end users**



Adopting a continuous improvement mindset for DevOps and OE helps the engineering teams to identify bottlenecks in the system, improve them, introduce efficient processes, and focus on overall performance and stability of the application. These bring in benefits not only to the immediate engineering team in the form of smooth on-calls and operational maintenance but rather help provide a better experience to end customers as well.

**What do other leaders have to say about it:**

“I believe in automation as much as possible. Any work which does every week/month and takes your 5% of total capacity should be automated asap.”

~ **Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

“I’ve been lucky to build trust with my leadership so that when I say “this is a business risk, we need to address”, I’m usually respected in that opinion. What helps me here is to think in thin interactions of improvement as I advocate for thin interactions on products. I find leaders are able to stomach an overhaul if it’s done in smaller stages mixed with other business priorities. I would also say that as I’ve matured in my career, I tend to be more critical of what is considered debt. Some engineers have a tendency to want to work with the newest framework, the newest pattern, but hardened code in an old framework may carry less risk than new code EVEN IF that new code would be considered better practice.”

~ **Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“Since I primarily own data products versus a traditional high availability system, OE isn’t confined to generic metrics like latency and availability. Our metrics revolve around the quality and freshness of our data. Ultimately, we work backward from our customers to understand what will matter to them most, tie that to our operational tenets and our operational excellence.”

~ **Nishat Akhter, Data Product & Engineering Leader, AWS**

“To prioritize operational excellence, encouraging teams to focus on challenges/problems of their respective teams. As leaders, spending more time aligning on the problem statement and impact. Prioritizing OE enhancements based on highest return in investments (infrastructure cost, performance, productivity improvement, developer time, etc.)”

**~ Saurabh Gandhi | Sr. Director Software Development Audible |  
Amazon | Ex-Amex**

“Monitoring is part of a DevOps transformation, but that means alerts and lots of them.”

**~ Larry Gordan, Managing Director at Emtec Inc.**

“If there is one skill or practice that all engineers should add to their repertoire, it is a thorough understanding of practices of DevOps and Site Reliability. These are integral parts of any modern development effort and should never be considered a "toss it over the wall" situation - the success of a good DevOps/SRE engagement is in collaboration and a shared understanding of goals. Learn about the tooling, techniques, and integrations and how they work together with your development efforts to improve reliability, efficiency, and performance.”

**~ Bruce Bergman, Manager at Lytx**

## **12.4 Stop & Think: Practice questions/follow-ups for this section.**

1. How will my team/organization benefit from DevOps?
2. What are the current pain points in my team(s) that perhaps focusing on DevOps can solve? Are the goals tangible?
3. Can I identify OE ambassadors/champions in my team(s) to help advocate for OE?
4. Does your team have a roadmap for OE items?
5. What channels in the company can you use to bring awareness about the adoption of DevOps?
6. How does my organization recognize and/or track tech debt?
7. How is the tech debt measured and reviewed in my team and organization?
8. Do we have an opportunity to introduce OE tools to help with OE?
9. What are your thoughts on success metrics to capture for OE efforts?

## 12.5 Summary

- DevOps, in layman's language, is a combined form of software development and software operations. They are usually the initiatives run by the engineering team to prevent shaky, unstable systems.
- Things like improvements in the on-call process, OE runbooks/documentation, including security during development, coding best practices, throttle services where needed, defect triage exercises, reporting and so forth all constitute the umbrella of DevOps and OE.
- DevOps and OE make an organization agile and help ship products faster to the end customers.
- DevOps help a business to incorporate consistency, reliability, and stability into the systems and help bring a mindset change. Some of the major advantages of DevOps and OE are:
  - Stability of systems - DevOps and Operational Excellence contribute to the stability and resilience of systems, particularly when security is a priority. This enhances present operations and facilitates faster recovery times in case of outages, ensuring preparedness for future use cases.
  - Improved on-call - They also improve communication and collaboration among team members, aligning everyone towards a common goal of enhancing the operational and engineering excellence of the organization. This leads to more stable on-call operations, reducing failures in production and ensuring satisfaction for employees and end customers.
  - Optimize resources and processes - This means teams can focus on automation versus manual tests, eliminate waste of resources and identify areas where improvements made can help save a buck for the company.
  - Faster time to market - OE opens the doors to proactively thinking about security, automation, shorter deployment cycles, etc that all lead to optimizing developer velocity and benefits to the services and infrastructure.

- There are several tools that can come in handy in your arsenal as you plan about DevOps and OE.
  - Acknowledge and reduce technical debt - Technical debt refers to a set of consequences in the form of code quality where an organization prioritizes speedy delivery of a project over project quality. Instead of trying to run away from it, the need of organizations is to face it.
  - Identify an OE ambassador/champion - This simply means identifying an engineer in the team who is passionate about OE and can keep the ball rolling forward. As EM, you need to align with this individual on the responsibilities of the OE ambassador, such as ensuring OE tasks are groomed every sprint and a prioritized backlog is maintained.
  - DevSecOps - DevSecOps is an extension of DevOps where we include the security component along with DevOps. It emphasizes proactive security considerations during the development process rather than reacting to security issues after they arise.
  - Test coverage - Testing can act as guardrails and quality gates for the changes made by your engineering team before they are handed to end users and can be of various forms like unit testing, integration, performance, and load testing, to name a few.
  - Reduce operational cost - This can include infrastructure costs (cloud spend, on-premise machines), employee pay, travel and perks for employees, office space, third-party tools and vendors being used, and so forth. Advocate for optimizing operational costs so you can bring awareness to the team and organization and have people volunteer on this initiative.
  - Managing deployments - Safer deployments simply mean that deployments are bug-free, do not cause any regression in production, and at the same time, the deployment window is time efficient. Higher code test coverage can organically help with safer deployments. Managing deployments is important and plays a crucial role in aiming for deployment safety.
  - Logging, monitoring, and dashboarding - These systems enhance transparency by providing visibility into the overall health and performance of the system. They are particularly useful in

- identifying bottlenecks, tracing issues, and triaging customer-impacting problems. Logging systems aid in troubleshooting. Monitoring tools help detect outages, generate alerts, and more
- Away teamwork - Away teamwork refers to a work agreement between engineering teams where one team (called the away team) implements features in the codebase owned by another team(called the host team).
  - Accessibility - Accessibility ensures every individual, irrespective of their status and abilities, gets access to the software products that meet their needs. It is empowering each individual, even if they are especially able, to be able to use the product by identifying barriers or friction points.
  - Learn from mistakes - Learning from mistakes is essential. The process involves promptly addressing the issue, conducting a root cause analysis, and evaluating customer impact. Once the problem is resolved, documenting the troubleshooting steps is crucial, and learn from mistakes.
- The [5 Whys approach](#) is a famous iterative approach to help with root cause analysis and get to the bottom of the problem. It is generally considered sufficient to go to five levels of why to get to the cause, although one can go deeper as well as need be.
  - Adopting a DevOps and operational excellence culture is not something that happens overnight. It is a strategic commitment and mindset change that is iterative in nature and is a continuous learning process.

[1] Learn more: Akanksha Gupta, "An Integrated Framework for DevSecOps Adoption," International Journal of Computer Trends and Technology, vol. 70, no. 6, pp. 19-23, 2022. Crossref, <https://doi.org/10.14445/22312803/IJCTT-V70I6P102>

# 13 Organizational change management

“Change is the law of life, and those who look only to the past or present are certain to miss the future.”

~ John F. Kennedy

## This chapter covers

- Reasons for re-organizations
- Framework for organizational change management
- How to handle a change in leadership
- The aftermath of changes in the workforce and how to deal with them

In today's fast-paced and ever-evolving business landscape, change has become an inevitable force that organizations must navigate to stay competitive and relevant. Businesses face a constant stream of internal and external factors that demand adaptation and evolution. As the world around us transforms, it is imperative for individuals and organizations to embrace these changes rather than resist them. However, human nature often inclines us to push back against the unfamiliar and cling to the comfort of the status quo. This chapter explores the critical discipline of organizational change management, delving into the reasons that lead to re-organizations, and strategies to go about an organizational change that empowers us to survive and thrive in times of change. It emphasizes the significance of working together, understanding that change is not an individual journey but a collective effort that requires collaboration, communication, and a shared vision. By embracing change, adapting to new circumstances, and leveraging the power of effective change management, organizations can seize opportunities, foster innovation, and achieve sustainable growth. So let's get started!

## 13.1 Motivations for reorganizations

Reorganization, which refers to overhauling the current state and bringing in changes such as moving teams, business units, or layoffs, is an unavoidable process in technology companies. In the fast-paced modern world, companies continually confront the imperative of outpacing their competitors to maintain the relevance of their products, features, and services. Tech companies, in particular, face this challenge to an even greater extent than other industries. To stay ahead in the game, companies employ reorganizations and restructuring of their operations as a means of response. Reorganizations can manifest in various ways, such as departmental restructuring, the establishment of new business and technical units, or realignment of the company's structure based on a fresh vision. While one might dread reorganizations, it has its advantages as well, like a more remarkable ability to react, discouraged complacency, allowing for trying new ideas and processes, and more. It is important to embrace changes and reorganizations and look for their positives. Within this section, we will explore different rationales behind company reorganizations, and we will also analyze the advantages and challenges associated with such transformations. Understanding these reasons will enable you to grasp large companies' functioning and ongoing efforts to remain at the forefront of the industry, ultimately facilitating your alignment with the company's mission.

### **13.1.1 Market conditions or external factors**

Rapidly changing market conditions such as shifts in customer preferences, new technologies, emerging competitors, regulatory changes, geopolitical events, recessions, or fluctuations in the business cycle and more can have a significant impact on the way a company executes its operations.

Technological advancements or breakthroughs, shifting consumer needs, growing competition, or a new area of opportunity can all contribute to this change where companies need to realign and reorganize their execution model. For example, let's say an audiobook company, XYZ begins to see a drop in revenue. Further exploration suggests the reason is other new players in the market that are providing similar and sometimes cheaper products to the customers. The XYZ company may do at least one or both of the following things as part of the strategic move.

- Figure out and build new key features and products that differentiate their company. In this case, XYZ decided to introduce a new catalog of podcasts to attract new customers. Further, they introduced a new tier in their subscription model to provide a less-priced subscription for a limited audiobook collection.
- Find new opportunities to compensate for the loss of revenue because of growing competition in one market or cost cutting. For this, XYZ decided to diversify the revenue stream and get into the video streaming business to attract a new set of customers.

Either of these scenarios would require the company to realign its focus which would require a reorganization.

Moreover, if a new product or new technological advancement threatens to disrupt the company, altogether this realignment gets even more pressing. Ultimately market conditions are a dominant factor that constantly requires companies to be aligned with their vision and adapt to shifting circumstances. All of this ultimately trickles down to reorgs.

### **13.1.2 Streamline operations**

Consider a company with a department dedicated to developing recommendation infrastructure for various products. Initially, this department may have been focused on targeting all use cases within a specific domain. However, if multiple teams within the company are independently setting up similar infrastructure stacks for similar use cases, it indicates a duplication of efforts. In such cases, a reorganization would expand the scope of the original team to support use cases beyond the initially targeted domain. This ensures that recommendation systems across different products work synergistically, leveraging a shared infrastructure stack and providing long-term value to the company by promoting maintainability. Reorganizations are frequently undertaken by companies to streamline their operations. One of the critical advantages of reorganization is the identification and elimination of redundancies, as well as the consolidation of functions that may be duplicating efforts across different departments with similar objectives. This consolidation helps companies

reduce inefficiencies, lower execution costs, and increase productivity. You, as EM, can help identify such redundancies to executive leadership to help streamline operations.

Reorganizations can also clarify the organizational structure of the company, enabling employees to understand their roles and responsibilities better. For instance, as reporting hierarchy changes, there can be more precise lines of authority and improved communication channels, making it easier for employees to understand who they report to and who they are responsible for in their roles. Companies might also revisit the job description and duties expected in the role and create more awareness of them. This, in turn, leads to faster decision-making, increased performance, and improved overall productivity. Another way reorgs prove beneficial is by facilitating closer proximity of teams that require frequent communication within the organizational structure. This helps break down communication silos and allows the company to reorganize around specific products or services. Consequently, the execution of new features related to these products becomes streamlined, and the company becomes more responsive to market conditions. Overall, reorgs are a vital tool companies use to optimize their operations, making them more efficient, productive, and capable of staying competitive and achieving their goals.

### **13.1.3 Unlock new potential opportunities**

Reorganization can also help companies capitalize on new business opportunities in several ways. First, reorganizing companies can create the resources and capacity required to support this new initiative. For example, a company could consolidate its investments in existing products and free up resources for a new initiative. This helps the companies redistribute how they spend the same amount of money overall to support more initiatives simultaneously.

#### **Did you know?**

"[Crossing the Chasm](https://www.amazon.com/Crossing-Chasm-3rd-Disruptive-Mainstream/dp/0062292986/)" (<https://www.amazon.com/Crossing-Chasm-3rd-Disruptive-Mainstream/dp/0062292986/>) is a marketing book by Geoffrey A. Moore. It delves into the challenges faced by innovative products in the market, paying particular attention to the critical "chasm" or adoption gap between early adopters and the mainstream market. The book suggests a significant gap between the initial adopters of a product, who are typically technology enthusiasts and visionaries, and the subsequent majority of users, known as pragmatists. This expectation disparity between visionaries and pragmatists is a crucial point addressed in the book. The author seeks to delve into these distinctions and provide insights on effectively navigating this "chasm." The book offers guidance on selecting a target market, comprehending the concept of a comprehensive product, positioning the product appropriately, developing a marketing strategy, determining the optimal distribution channel, and establishing suitable pricing strategies.

Second, reorganizations can help create an organizational structure that explicitly supports this new initiative. This is a crucial aspect of unlocking new business opportunities, as dedicated effort is required to capitalize on such an opportunity. For example, a company may create new business units focussing on developing and integrating technologies for this new investment area. This can help ensure that this new business unit has the support and expertise required to succeed. Likewise, a company may reorganize to eliminate a business unit because of market conditions or technology changes. Witness the recent [ChatGPT](https://openai.com/blog/chatgpt) (<https://openai.com/blog/chatgpt>) explosion, and how it is already killing some small companies (for example, marcomms companies losing contracts because they can now ask ChatGPT to write the press releases for them).

Third, reorganizations can help ensure that their organizations are lean, efficient, and agile, which can be crucial to setting up and scaling new business units. By creating more flexible organizational structures, companies can quickly respond to market changes, make faster decisions and adapt to changing situations.

Companies can better align their technical departments to the overall strategic vision and underlying goals using the above. This can be highly

beneficial when creating new business opportunities as it helps companies realign the vision based on these new opportunities and brings a holistic picture to the consumer.

### **13.1.4 Change in leadership**

A change in leadership is another common reason a company can go through a reorganization. A company can aim to get to a flatter organization if there are multiple layers of leadership and hence would go for a reorganization. Another instance can be if one of the directors left the company and the company decides not to hire a replacement, they may distribute the existing team to the other two existing directors. Some other reasons for leadership changes could be the company pivoting to a new technology or service, moving to the cloud from a traditional on-premise approach, newly acquired companies with different technology stacks or services, integrating and combining divisions or companies, or implementing enterprise CRM or ERP systems. Whatever the case, such a change in leadership creates a ripple effect and impacts many. Such changes should be dealt with carefully as it involves several factors, such as how the new leadership is, what their management style is, and so forth. We will discuss this in more detail in section 13.3 to learn how to handle a change in leadership. Let's explore a framework we could use to navigate organizational change management.

## **13.2 Framework for managing organizational change**

Managing organizational change is a complex endeavor that requires careful planning, strategic execution, and effective leadership. Organizations must adapt to stay competitive and thrive in evolving market conditions, technological advancements, and shifting customer demands. The framework for organizational change management serves as a guiding tool to navigate the intricacies of change, enabling companies to minimize resistance, maximize employee engagement, and achieve successful outcomes. This framework provides a structured approach that outlines key

stages, strategies, and considerations necessary to manage change initiatives within an organization effectively. By understanding and applying this framework, leaders can foster a culture of agility, resilience, and continuous improvement, positioning their organizations for long-term success in today's dynamic business landscape.

### **13.2.1 Understand the need for change**

The first and foremost step in any organizational change is understanding the ‘why’ behind it. This involves understanding the need for the organization’s change, what investments it would need from the employees and the company, and the expected outcomes or benefits to it. One needs to ensure all the other alternatives have been explored. Further, any organizational change comes with a component of time to it, the time spent in planning, implementing the change, and reaping the benefits. Hence it is essential to time-box the change management and prepare in advance if any friction or pushback happens. Remember, we need to thoroughly create a change management strategy and analyze it to understand if the changes that come out of it are a one-way door or a two-way door.

For instance, you work for an audiobook company that has a subscription-based model for customers. Now each month, the standard subscription provides two coins to customers, which are suitable to be used for two audiobook downloads of their choice. The customers can further purchase additional coins for \$10/coin. Your company has new competitors in the sector, so it decides to differentiate itself from the competitors by providing more value to its customers. As part of that, a significant company change is being discussed, which would entirely remodel the subscription-based model. Now let's say you have two choices- first, the standard subscription model now offers four coins instead of two to provide extra value to customers at the same price, and the second choice is to reduce the value of extra coin purchase from \$10 to \$8. The two strategies look equally good as they give customers more value, but a deeper dive might suggest that choice one might be like a one-way door while choice two can be a two-way door. In choice one, we are changing the base subscription model, and hence once the change is made, the customers will be used to the four coins and would

not take less. For choice two, since it is more of a top-up value on the base plan, in case something does not work out, the company can increase the price for additional coins back. This shows that for the organization to survive competitors and thrive, it needed to understand the ‘why’ behind it. There is new competition in the market, and the company needs to differentiate itself from competitors, hence the new changes in the subscription model. They time-boxed the changes to, say, four months to move fast and deliver value to customers.

Now let's look at how the people aspect of champions is essential for any organizational change.

### **13.2.2 Recruit champions/ambassadors for bringing change**

Any change to be introduced needs a team of passionate people we can call champions or ambassadors to help us advocate for the change. They are like the acting force behind the implementation and spreading the word on changes brought in the company and hence need to be onboarded with the changes. As an engineering manager (EM), if you are tasked with planning an org change, one of your primary jobs will be to recruit from existing folks the set of champions. Look for self-driven employees, who believe in the mission of the change and are effective communicators who can help spread the word. Since they will help bring about the change, consider involving them early as you work on shaping the org change. Make them part of the brainstorming phase to craft the strategy and then run the show during the execution phase. Making them feel heard will keep them motivated toward the vision, and providing the set of desired training resources will provide them with the support they need to succeed.

### **13.2.3 Craft a strategy and roadmap for change management**

After assembling the appropriate team, the subsequent task involves formulating a strategy and roadmap that outlines the necessary steps to translate organizational change from mere ideas to tangible actions. This stage entails defining the criteria for success and establishing the expected outcomes, considering the available resources and time constraints.

Moreover, critical decisions must be made, such as determining whether the change should be implemented through a comprehensive "big bang" approach or a more iterative process. This forum provides an excellent opportunity to initiate discussions on the fundamental principles that are the basis for the change and the best practices that teams should adhere to. The strategy should include an effective communication model with other teams and organizations to promote easy adoption of the upcoming changes. Many of these decisions rely on a comprehensive understanding of existing pain points, as well as the strengths and weaknesses of the organization.

### **Did you know?**

The **SWOT** analysis is a technique to evaluate the business and create a strategy. The analysis is done at four dimensions, namely:

**Strengths-** Strengths encompass the areas where an organization demonstrates exceptional performance and distinguishes itself from competitors. These may include a strong brand, a devoted customer base, a solid financial position, innovative technology, and other advantageous factors.

**Weaknesses -** Weaknesses are the areas that hinder the growth of the organization and need improvement.

**Opportunities -** Opportunities represent advantageous external circumstances that have the potential to provide a competitive edge to an organization.

**Threats -** Threats refer to factors that have the potential to harm an organization. For instance, if a company manufactures phones and procures the chip from a foreign country and perhaps the trade between the companies stops, the company is at a threat of being able to fulfill the demand based on the supply.

A SWOT analysis aims to enable an objective and evidence-based evaluation of an organization, its initiatives, or its industry, highlighting its

strengths and weaknesses. To ensure accuracy, the analysis should avoid preconceived notions or ambiguous aspects and instead concentrate on practical and tangible situations.

It is crucial to consider both short-term and long-term plans when crafting the strategy, ensuring that it encompasses the necessary actions for immediate improvements while aligning with the overall vision for the future. The strategy will help you get a buy-in from the leadership team, so put your best foot forward. Next, let's look at how to go about executing the change.

### **13.2.4 Execution**

Execution is the core stage of bringing the organization change, where the team brings words to action. You start with the strategy and execute it in phases to keep your eyes and ears open for early feedback. Constructive feedback will help shape your execution plan and make people feel heard. At the same time, remain open to the fact that you could have missed something, or your plan is not optimal, and be willing to flex with the feedback and changes coming your direction. Also, as part of human tendency, no change is easy, so there are chances to get friction and pushback in the start. Keep your actions data-driven and effectively communicate as much as possible with the stakeholders. Few ways to tackle pushback:

- Keep discussions data-driven and objective. Good documentation will help serve as a blueprint for times of conflict.
- Keep your doors open for feedback. You might have to think from other people's shoes to get what they are saying and why they are saying so
- Ask the right set of questions. Disagreements are bound to happen, but promote a healthy discussion by asking the right questions to help come to an agreement or clarify the confusion. Don't forget to acknowledge and thank the person for the discussion, irrespective of who was right or wrong.

- Provide training resources and support to the team members to help bring their best. This also reflects that you are vested in their growth and development.
- Negotiation is critical to resolving conflicts in the workplace around change management. Try to come to a common ground by finding the sweet spot and agreeing.

While all this sounds reasonable, there can be times of conflict where there is a deadlock, and the other person refuses to budge (authoritative style). Such situations call for enforcement if it is the right thing for the people and the organization. An excellent way to handle execution is to work on a proof of concept and demo an early version of the product, process, or structure to check on feasibility and help prove the idea to others. With that, let's look at how to monitor and measure the outcome of the efforts.

### **13.2.5 Monitor and measure success**

What is the result of the steps you have taken so far? The monitoring and evaluation of actions during change management is a crucial stage. The results are assessed against the predetermined success criteria established during the strategy development phase. It is essential to actively seek feedback through surveys and retrospectives to identify successes and areas for improvement. Additionally, it is important to show appreciation for the hard work of ambassadors and champions and acknowledge their contributions. Cultural and mindset changes are essential to bring about any changes. It is crucial to recognize that change does not occur overnight, requiring patience as adoption and results take time to manifest.

### **13.2.6 Iterate on opportunities**

This step follows the monitoring and measuring success phase. The idea is to identify the gaps from experience, consolidate feedback, organize retrospectives and iterate on opportunities for the future. If things can be changed, move back to step three of crafting the strategy and the roadmap for change management and incorporate the changes in the strategy, reevaluate, and re-execute. Identify the strengths, weaknesses, and trends

you see from the feedback. Based on the evaluation, make necessary adjustments, refinements, or pivots to your opportunities.

Remember that iteration is an ongoing process, and embracing a mindset of continuous improvement and agility is essential. By iterating on opportunities, you can optimize their potential, address emerging challenges, and uncover new avenues for growth. In the next section, we will learn about tackling leadership changes, whether it is your boss who changed or you are a new boss to the team.

### 13.3 How to handle a change in leadership?

When a change in leadership occurs, whether due to a new boss taking over or becoming a new boss to a team, it is common to feel some level of apprehension, it is essential to approach this change thoughtfully and embrace it effectively. By applying the change management framework discussed earlier, you can work towards achieving a seamless transition. Some fundamental tenets to keep in mind as you go about a change in leadership are:

- Understanding the reason behind the change in leadership to help communicate it to your team members further. Learn what will change, the timelines, expectations, plans, and milestones for the upcoming changes. For instance, if Dave will be the new Director of technology for commerce, when sharing this with the team, be aware of the reasoning given behind the change, and what Dave has managed in the past to share with the team when the change takes into effect
- Lay emphasis on communication by creating channels for candid communication. Be there for your team members, should they have any questions or concerns
- Try to convey your view and, simultaneously, be a patient listener to make others' opinions heard. Remember, changes lead to learning in both directions.
- Take ownership and accountability for the changes. Don't be someone who simply repeats what is told to them, participate actively and be part of the process so you earn the trust of your team members.

- Risks are an essential part of any change management. Be transparent and discuss what you fear about the change, the risks involved, and the mitigation plan. This is an excellent opportunity to invite people to participate in the process.
- Be vocal about the resources and training you need to work efficiently. If you are the new boss of a team, ask your team members what will help them for a smooth transition.
- Don't shy away from leading by example. If the new leader expects people to be in the office three days a week, first see if you are on board with the change, start by showing up in the office and reinforce the message of how it helps the team with collaboration. Don't expect something from your team that you are not ready to do yourself.
- Keep a growth mindset as it allows one to envision a change in leadership as a chance to explore new perspectives, acquire new skills, and expand their horizons.
- Collect feedback to be shared with leadership to understand what is working well and what needs further change.

### **Did you know?**

#### The Lewin change management model

Lewin's change management (<https://www.mindtools.com/ajm9l1e/lewins-change-management-model>) model divides the change process into three phases. It is widely utilized because it clearly outlines the steps and the role of management at each stage. These phases are as follows:

**Unfreeze:** People generally have a natural inclination to resist change, so the initial phase of the change management process focuses on initiating change by motivating employees to embrace it.

**Transition:** The organization goes through a transitional period once the change process is initiated. During this time, it is crucial for the company's leadership to provide strong guidance and continual reassurance.

**Refreeze:** After the change has been implemented and accepted by the organization's workforce, everyone begins to view the new guidelines as the norm, and the organization regains stability. This phase involves solidifying the new state and ensuring its integration becomes the established practice.

If there is a change in leadership directly impacting you, it is crucial to recognize and comprehend how this change will affect your daily activities. This involves initiating meetings with the new leader to gain a clear understanding of their motivations and expectations. Take the time to clarify their management style and work collaboratively to assist them in transitioning to their new role. Remember, the change affects you and the new leader, so fostering mutual understanding is essential. Once you have a better understanding of the new leader, it becomes your responsibility to communicate the changes to your team and assure them of your support during this transitional period. Additionally, help the new leader by familiarizing them with the team structure, core principles, organizational culture, and key stakeholders.

On the other hand, if you are the new manager joining a team, start by comprehending the strengths and weaknesses of the team members you will be managing. Collaborate with the outgoing and incoming leaders to create a transition plan outlining key responsibilities, timelines, monitoring, escalation routes, and goals for the handover period. Ensure there is proper knowledge transfer so nothing is lost in the transition. It is equally important to manage the people aspect of the transition effectively. For example, if a team member is ready for a promotion, ensure you gather all the necessary information to support their case. At the same time, clarify any over-promises made by the previous leader. From my own personal experience, I once took over a team from a manager who was about to exit the company. To ensure a smooth transition, I set up conversations with each team member where I invited the previous manager and the team member so the three of us could have a candid conversation and ensure nothing was lost. Next, I set up a one-on-one conversation with the team member to ensure we both are on the same page, once the previous manager exits. This ensured clear communication and transparency. One can also

give the new leader anonymous (guided, led) questions from the existing team, to help better understand the perspective and plans of the new leader.

### **Did you know?**

### **The ADKAR change management model**

The [ADKAR change management model](#) (<https://www.prosci.com/blog/applying-the-adkar-model-when-change-management-is-new>) is a famous model that involves educating company members about the significance of the changes being introduced and its effects on their daily routines, as well as providing continuous support and reinforcement. This enables them to comprehend and wholeheartedly accept the value of the change.

The ADKAR model encompasses the following five stages:

**Awareness** - make them aware of the change

**Desire** - the desire to change and to join the mission

**Knowledge** - how the change will take effect

**Ability** - demonstrate the skills required for the change, provide desired resources for support

**Reinforcement** - for the sustained impact of the change, recognition is part of the reinforcement

Let's say there's a company called ABC that wants to move from a third-party project management software to an in-house tool across their organization. They decide to use the ADKAR model to guide their change management process.

**Awareness:**

ABC starts by creating awareness about the need for the missing capabilities of current project management software. They send a company-wide email explaining the current system's limitations and the new software's benefits. They also organize a town hall meeting where the head of engineering addresses the importance of improving project management efficiency and encourages employees to embrace the change.

### **Desire:**

To create a desire for the new software, ABC showcases success stories from other companies that have implemented similar tools in-house and the financial savings that come with it. They highlight how the new software will streamline processes, improve collaboration, and simplify employee jobs. They also emphasize that training sessions and resources will support learning and using the new software.

### **Knowledge:**

ABC provides comprehensive training programs to ensure employees have the necessary knowledge to use the new software. They offer online tutorials, in-person workshops, and access to a dedicated support team. They also create a knowledge base with FAQs and step-by-step guides to address common challenges.

### **Ability:**

ABC understands that simply knowing is not enough; employees need the ability to use the software effectively. They assign project mentors to each team who guide and assist during the initial implementation phase.

### **Reinforcement:**

To reinforce the change and make it stick, ABC recognizes individuals or teams that effectively use the new software. They create a feedback loop where employees can share their experiences, suggestions, and challenges related to the software. ABC considers this feedback and makes continuous improvements to the software and support processes.

By following the ADKAR model, ABC successfully implemented the in-house solution, ensuring employees are aware of the change, desire its benefits, have the knowledge and ability to use it effectively, and are reinforced for their efforts. The ADKAR model is a valuable tool that helps set the foundation for enterprise change management.

Allow your team members the time and space to adapt to the leadership change, avoiding unrealistic expectations. Understand the reasons behind the team's current practices before introducing any changes or imposing your own preferences. Aligning the team with goals, mission, and guidelines ensures everyone is on the same page, leaving no room for ambiguity. Remember, each leadership change is unique, and it's essential to tailor your approach based on the specific circumstances and needs of your organization and team members.

## **13.4 Aftermath of changes in workforce**

Changes in the workforce often present themselves as pivotal moments in an organization's journey, bringing both challenges and opportunities. Whether driven by the need for cost-cutting measures or strategic realignment, these changes reshape the employment landscape, leaving lasting effects on individuals, teams, and entire companies. Understanding the intricacies of these workforce transformations and uncovering strategies to navigate these times of change is essential. In this section, we focus on two significant changes: reduction in workforce and mergers or acquisitions.

### **13.4.1 Reduction in workforce**

Workforce reduction, also known as a reduction in force, involves the elimination of positions within an organization without any intention of hiring replacements. This can occur due to various factors, both internal and external. External factors may include unfavorable market conditions, leading the organization to cut a particular workforce segment to avoid expenses associated with a team that may not generate the expected

revenue. Internally, a company may merge two teams into one, resulting in the redundancy of specific roles, such as engineers and leaders, leading to their elimination. As an EM, you may encounter critical situations where team, division, or company-wide layoffs are announced.

To navigate these challenging circumstances, one can employ some strategies such as after the announcement of upcoming layoffs, promptly reach out to your team members to convey solidarity and support. Keep an open line of communication, ensuring they feel comfortable approaching you as a sounding board and seeking assistance. If you are involved in layoff planning, exercise caution as you can access confidential information. In such cases, treat all individuals honestly and fairly, maintaining professionalism by preserving confidentiality while remaining accessible to your team members.

Layoff information is typically communicated by the HR department; however, in some companies, EMs may be entrusted with delivering this difficult news. Approach affected employees with empathy, allowing them time to process the information. Share pertinent background details about the decision-making process, mindful of legal constraints. Familiarize yourself with the severance policy and other benefits, redirecting impacted employees to the appropriate HR representative for further inquiries. Assist impacted employees by leveraging your professional network to share their profiles. This can be done through platforms like LinkedIn or other social media channels.

Following layoffs, it is natural for the remaining employees' morale to be affected. Organize team meetings to reassure them of their unaffected status and provide a supportive environment for discussion. Maintain frequent communication to demonstrate compassion and strive to inspire them by presenting the future vision. If your team has been impacted, develop a plan for redistributing workload to ensure project deadlines are met. Proactively communicate any risks resulting from the reduced workforce to stakeholders in a timely manner, mitigating potential surprises down the line. Workforce reduction is a challenging period for all involved. It is

essential to handle the situation professionally and approach it with maturity, considering the prevailing circumstances.

### **13.4.2 Acquisitions or mergers**

When two or more companies come together under one umbrella and join together, it is referred to as a merger. The acquisition is when company A controls company B through a purchase deal or another method. Such changes in the workplace also lead to significant changes in the workforce and impact the employees. For example, if a bigger company A buys a smaller company B, there is a sudden change in leadership across company B. Now this change can be positive or negative and, at the same time, involve a few roles that are eliminated as part of the acquisition. In such situations, use the strategies shared in the preceding section on reduction in the workforce along with understanding the context behind the change, why it is happening, what will be the impact, and how it is being executed, and maintaining effective communication with employees to ensure they feel heard and understood.

Acquisitions and mergers bring together companies with unique cultures, systems, and workforces, forging a new entity that can be greater than the sum of its parts. However, amidst the promise of growth and synergy, challenges arise, including the integration of disparate workforces, the alignment of goals and values, and the navigation of potential redundancies. There are often disgruntled employees on both sides. Salaries may be different, benefits differ, and even the working structure and team norms may be different. It's essential to look at all angles and take both points of view (acquirer and acquiree) into consideration.

In conclusion, change management is a complex and ever-evolving landscape of organizational transformations. The framework presented offers a guiding light, empowering leaders to navigate the complexities of change with strategic clarity and sensitivity to the needs of their workforce. Furthermore, we have delved into the critical topic of change in leadership. By embracing empathy, transparency, and open communication, you can guide the teams through these transformative moments, ensuring a smooth

transition and inspiring employee resilience. Ultimately, organizational change management is a delicate dance, requiring a balance of strategic foresight, compassionate leadership, and a deep understanding of the human element within the organization. You, as EM, must navigate the winds of change with confidence and drive meaningful transformation.

### **What do other leaders have to say about it:**

“Effectively handling organizational change can pose challenges, particularly when a new leader enters the company and implements immediate changes. As an engineering leader, it is crucial to invest time in understanding past practices and identifying the gaps that necessitate addressing during the reorganization process.”

**~ Sanjay Gupta, General Manager at HCL Technologies**

“Mostly, developers fail to understand the motivation behind these reorgs and find them meaningless. Sharing a business aspect of how/why it aligns with a long-term plan helps them understand and grow as a contributor.”

**~ Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

“As part of the team restructure, we had to move people around to different squads. The change was important for future projects. Started working with the leaders of each group to define the objectives for each team. The next step was delivering the message, working with the teams to gather feedback, and documenting any process changes. Team members hesitated to change the existing process and let go of their squads. It was important for the leaders first to accept the change and be the proponent of the new model.”

**~ Rajakumar Sambasivam, Delivery Manager at Microsoft**

“Often pushbacks happen when there is a lot of ambiguity in the process, and the impact is unknown. If we do not know who and how a change will

impact us, we tend to reject the idea of the change itself.

When there is pushback, I empower my team to dive in and figure out the answers to who and how and, most importantly, educate themselves on why the change is important.”

**~ Nishat Akhter, Data Product & Engineering Leader, AWS**

“Re-orgs are unavoidable in a growing organization. A team structure that made sense in the past may become less productive or dysfunctional due to interdependencies. Accepting and addressing this openly helps people understand the challenges and the need for reorg. As a leader, I feel it is most important to show my commitment to continuous growth and emphasize that individuals not lose their progress with the previous manager or team even after reorg.”

**~ Saurabh Gandhi | Sr. Director Software Development Audible | Amazon | Ex-Amex**

“Embrace change! Not only does it keep you on your toes and constantly learning, but it shows that you are adaptable and flexible, and open to new ideas and new ways of achieving value for your team, your department, and your company. You have probably worked with someone who was the opposite - NOT open to change - and experienced how difficult that made just about any task. Don't be that person!”

**~ Bruce Bergman, Manager at Lytx**

## **13.5 Stop & Think: Practice questions**

1. What questions would you ask your manager if they told you about an upcoming organizational change?
2. What are some of the ways you have navigated through an organizational change?
3. If you were to go back in time, what would you change about your experience from above?

4. Have you gone through a leadership change in the past? If yes, was it a positive experience or not? Why?
5. What do you think is an effective way to handle a change in the workforce professionally?

## 13.6 Summary

- In the dynamic and rapidly changing business environment of today, organizations face an undeniable reality - change is an unavoidable and constant presence. To remain competitive and stay relevant, organizations must adeptly navigate the ever-evolving landscape that surrounds them.
- In the tech industry, reorganization is crucial. Companies must stay ahead of competitors and keep their products and services relevant. To do this, they must be agile and adaptable. Reorganizations and restructuring are common approaches used to address this challenge.
- Several factors can contribute to organizational change management, such as:
  - Market conditions or external factors - Technological advancements or breakthroughs, shifting consumer needs, growing competition, or a new area of opportunity can all contribute to this change where companies need to realign and reorganize their execution model.
  - Streamline operations - Companies reorganize to improve efficiency and optimize operations. One key advantage of reorgs is the identification and elimination of redundancies.
  - Unlock new potential opportunities - Reorganization can catalyze companies to seize new business opportunities by opening up various avenues for growth and innovation.
  - Change in leadership - A change in leadership creates a ripple effect and impacts several people. Such changes should be dealt with carefully as it involves several factors, such as how the new leadership is, what their management style is, and so forth.
- An organizational change management framework is a guiding tool to navigate the intricacies of change, enabling companies to minimize

resistance, maximize employee engagement, and achieve successful outcomes. It involves:

- Understand the need for change - The first and foremost step in any organizational change is to understand the ‘why’ behind it. This involves understanding the need for organizational change, what investments it would need from the employees and the company, and what are the expected outcomes or benefits to it.
- Hire champions/ambassadors for bringing the change - To successfully implement change, it's essential to gather a dedicated group of individuals known as champions or ambassadors. These individuals advocate for the change and act as driving forces behind its implementation.
- Execution - The process begins with devising a strategic plan and executing it in iterative phases while remaining attentive to early feedback. This approach ensures that organizations remain open and receptive to valuable insights.
- Monitor and measure success - Outcomes are assessed based on the predetermined success criteria set during the strategy development stage. Gathering feedback through surveys and retrospectives is essential to identify achievements and areas for improvement.
- Iterate on opportunities - The idea is to identify the gaps from experience, consolidate feedback, organize retrospectives and iterate on opportunities for the future. If things can be changed, move back to step three and incorporate the changes in the strategy.
- When there's a leadership transition, such as a new boss or someone taking on a leadership role, it's normal to feel some apprehension. To navigate this change successfully, it's essential to approach it with a mindful mindset and embrace it.
- Some fundamental principles to remember: understanding the reason for the change, communicating effectively, taking ownership of the changes, identifying risks early on, providing training resources, leading by example, maintaining a growth mindset, and more.
- Changes in the workforce frequently manifest as critical junctures in an organization's trajectory, ushering in a blend of challenges and

possibilities. This can be due to:

- Reduction in workforce - Workforce reduction is a challenging period for all involved. It is essential to handle the situation professionally and approach it with maturity, considering the prevailing circumstances.
- Acquisitions or mergers - When companies merge or acquire each other, they bring together different cultures, systems, and workforces, creating a new entity with the potential to achieve more than its individual parts. However, this process also presents challenges, including integrating diverse workforces, aligning goals and values, and managing potential redundancies.
- Change management represents a multifaceted and continuously evolving realm encompassing organizational transformations. As an EM, it is imperative for you to navigate this ever-changing landscape with self-assurance, propelling impactful and purposeful transformation.

# 14 Time management

“Lack of direction, not lack of time, is the problem. We all have twenty-four-hour days.”

~ **Zig Ziglar (Motivational speaker)**

## This chapter covers

- Why time management is essential for engineering managers
- Tips for better time management
- Tools like the Eisenhower Matrix to manage your time better

Amidst a day brimming with consecutive meetings and no time to spare for even a quick meal, the prospect of cooking and eating seems out of reach, mainly if you are working from home. You steal a moment between meetings to glance at your calendar and realize that not only is your day packed with meetings, but the entire week offers little time for your own work. Despite feeling exhausted, you soldier on and make it to the end of the day. However, before logging off, you quickly check your emails and discover that an essential planning document was due two days ago, which you had completely forgotten about. Inevitably, you must spend another hour responding to the email, but you also want to create a list of tasks to tackle tomorrow. As you jot down the to-dos, you realize that there are over 40 items that require your attention, and prioritizing them in such a short time frame is difficult. You feel the stress mounting as you know you won't be able to give your best effort to everything. Over the next few weeks, you try to clear your backlog by dedicating unreasonable amounts of time, starting early in the morning and pushing late into the night, until burnout begins to set in. Although things are finally under control, it is challenging to be productive in your regular office work. In an attempt to manage things in a sustainable and ongoing manner, you block out time in your calendar to work on your own tasks. Still, just 15 minutes into your first-time block, you receive a message requesting your immediate assistance, and the cycle starts all over again.

Situations like these are easy to happen for an EM who is unable to manage their time effectively. As an engineering manager (EM), you wear lots of different hats. This means you are involved in various different things related to managing your team both on the execution front as well as on the people management front. At the same time, EMs are also responsible for their own deliverables. Some examples of these deliverables could look like this:

- Developing a process that improves the hiring standards for your organization
- Introducing processes that help load balance your team's on-call load
- Establishing practices for how new code is landed in production, minimizing regressions caused by them

Therefore, it is imperative to manage your time efficiently to make progress on all fronts—engineers, unlike managers, often like to work on the maker's schedule. Paul Graham, a renowned computer scientist, explains the difference between a manager's schedule and a maker's schedule. A manager's schedule will involve several things simultaneously that lead to context switching, while a marker's schedule can be more deep work with focus blocks where concentration is on a few things at one point in time. Constant context switching for a manager's schedule poses a big challenge. This means your productivity takes a huge hit when you switch contexts, and time can seem to fly by. Therefore, EMs need to be extra aware of how they are utilizing their time and how things on their plate are getting prioritized to make constant progress. First, let's start with diving deeper into understanding the importance of time management for EMs.

## 14.1 Importance of time management

Effective time management is essential for EMs. EMs are frequently engaged in multiple tasks and must frequently switch between contexts. However, if an EM is unproductive, it can adversely affect the productivity of the entire team. Similarly, EMs need to lead by example. If you are sending late-night emails to the team, you are setting the wrong example that it is ok to send late-night meetings. It is on you to use your time effectively

and set the team culture. Therefore, it is critical to examine why time management is crucial for EMs, as well as the challenges that may arise while attempting to do so.

### **14.1.1 Successful project delivery**

The first and foremost responsibility of an EM is to ensure the successful delivery of the projects that their team has committed to. Managing multiple projects at the same time can be challenging. Not managing your time well can make this challenging task nearly impossible. Effective time management helps you prioritize your time among these different projects and other responsibilities. This process helps you allocate enough time to finish your work on all the projects, avoids procrastination, reduces stress, and limits your burnout.

### **14.1.2 Achieving personal goals**

EMs are responsible for leading teams of multiple software engineers. Consequently, a significant portion of their time is devoted to assisting individuals in achieving their personal goals while aligning them with the team's objectives. Additionally, EMs have various deliverables to fulfill, which typically do not entail coding but may take on various shapes, such as facilitating teams to complete their work, collaborating with cross-functional partners to establish the team's roadmap, and monitoring the team's goals. Without adequate time management, prioritizing all aspects of work becomes challenging, and productivity may suffer. There may be periods when most of the time is spent supporting others while personal tasks take a back seat, or vice versa. In either scenario, stress levels remain high due to an ever-growing backlog of tasks. This may lead to a vicious cycle where a substantial amount of time is spent supporting others, causing a delay in personal tasks. In an attempt to catch up, EMs may over-prioritize personal tasks, leaving the team feeling neglected. Consequently, EMs may need to redirect their energy towards supporting others, and the cycle continues, hindering personal productivity and ultimately affecting the ability to achieve personal goals.

### **14.1.3 Rule your calendar**

EMs work on a variety of different time constraints. Since you are wearing many hats as an EM, your presence may be required in many different things. EM's, therefore, are always tight on time. There are many reasons why you may feel you always have tight deadlines.

- **Meetings** - Meetings for everything, as the expectation may be that you are required everywhere
- **Interruptions** - Constant interruptions can decrease your productivity significantly, you need more time to finish a task than usual.  
Interruptions can occur because of a routine schedule or sometimes ad hoc. “We need you now” is a line that almost every EM is bound to face in their careers more than once. These ad hoc interruptions can put you significantly back and increase your time to finish your tasks.
- **The expectation of high quality** - As an EM, you are the team leader. Therefore, there is a general expectation from you to produce high-quality deliverables. These deliverables lay the foundation for how the team functions, and the team members often evaluate your deliverables as the quality bar for the team. High-quality work requires time. There is no way around it, and hence you may need to spend more time doing tasks than others would.
- **Procrastination** - EMs can quickly feel overwhelmed with the amount of things they have to manage. Therefore, it is easy to procrastinate at a later time. This short-term strategy can have significant consequences later and shoot your stress through the roof as your deadlines approach.

### **14.1.4 Effective decision-making**

EMs are tasked with making numerous decisions for their teams. To make informed and well-considered decisions, effective EMs often invest a significant amount of time in analyzing the problem statement and correlating it with relevant data. Additionally, decisions must be communicated effectively and in a timely fashion. As such, EMs are also accountable for ensuring that any decision they make aligns with the teams they collaborate with. Decision records play a crucial role in managing and

referring back to decisions made in the past with specific constraints and environmental factors. Proper time management allows EMs to fulfill their responsibilities with due diligence, which saves the company money that would otherwise be expended in correcting unsound decisions.

## **14.2 Tips for better time management**

We have already considered why effective time management is an important skill to master. Better time management helps you increase your productivity, keeps you involved in things that matter, and creates opportunities for others to uplevel themselves in the team. Let's look at some tips on how you can better manage your time.

### **14.2.1 De-clutter your calendar**

As we have already discussed, an EM's participation is required in a lot of different things. It is easy to get overwhelmed by the number of meetings you may need to attend. Hence it is essential to keep your calendar as less cluttered as possible. Keeping a clear calendar has several advantages.

To begin with, streamlining your calendar allows you to prioritize critical tasks that may go unnoticed. When you have a jam-packed schedule, you quickly lose sight of the most significant things you must accomplish. By simplifying your calendar, you can quickly identify such tasks and take concrete steps to address them. Additionally, a less cluttered calendar boosts productivity, as you can devote more time and energy to important tasks. By prioritizing your work more effectively, you can spend more time on crucial matters rather than irrelevant ones, significantly increasing your productivity.

Furthermore, decluttering your calendar is essential for effective communication. Although it may seem counterintuitive to think that skipping specific meetings can improve communication, in my personal experience, it has indeed been the case. In such situations, it can be challenging to stay on top of things, particularly the most critical ones, which can result in poor communication with your partners.

Start by identifying and defining **clear priorities**. One tool you can use here is the Eisenhower matrix, which we will look into in the next section. This will help you to focus on what matters the most and avoids wasting time on less critical activities. Even within those priorities, there may be meetings that are unnecessary and avoidable. Identifying and eliminating such meetings can save a lot of time.

Another way to help is to **create focus blocks** on your calendar (and you can color code them as well for better visualization- how cool). Remember, as someone in the technology sector, you protect your own calendar, so it is essential to have these breathing room focus blocks to start your day or keep them at a point in the day when you feel most productive. A few things that worked for me are creating these focus blocks in the morning before my meetings kick in to give me a head start for the day. At the same time, you need to give yourself space and create blocks to have a lunch break, walk the dog, or take a quick walk/stretch to disconnect and refresh quickly.

Similarly, try to have blocks on your calendar for Friday to reflect on what was achieved this week, document it and plan ahead for the next week to stay on top of the game. This is especially important in our job roles where we are bound to a chair and desk for most of our working day.

**Figure 14.1 Declutter your calendar by creating focus blocks, lunch breaks, and free time blocks**

	MON	TUES	WED	Thur	FRI
9:00am- 9:30am					
9:30- 10am					
10am- 10:30am	1 on 1	M	M	M	M
10:30- 11am	1 on 1	1 on 1	1 on 1	M	M
11am- 11:30am	PROJECT MEETING	1 on 1	1 on 1	M	AGILE SESSION
11:30am- 12	MEETING	1 on 1	1 on 1	DEMO	F
12-12:30pm	LUNCH	LUNCH	LUNCH	LUNCH	LUNCH
12:30- 1pm	M	OPERATIONAL REVIEW	SECURITY REVIEW	OFFICE HOURS	SS 2 GET BLOCK
1-1:30pm	M	F	F	F	MEET
1:30- 2pm	F	TEAM MEETING	F	ARCHITECTURE BOARD	NO MEET
2-2:30pm	F	TEAM MEETING	1 on 1	F	
2:30- 3pm	M				
3-3:30pm	BREAK	BREAK	BREAK	BREAK	BREAK
3:30-4pm	F	M	M	AGILE SESSION	
4-4:30pm	M	F	M	M	
4:30- 5pm			M		

 FOCUS BLOCKS  
 MEETING..  
 FREE BLOCK

Another thing that has worked well in my teams is to designate a day in the week as no meetings day. This is a day that you, as EM, together with team members, can decide where you try to protect your team from having meetings on that day of the week to give enough makers time to engineers. This also helps build trust and ensure they are getting the time they need to bring their potential out. Make sure your team members feel heard by sending frequent surveys to help gauge the sentiments of the team members and discussing action items with the team. This leads us straight to providing more transparency to everyone in a meeting.

## **14.2.2 Meeting Bill of Rights**

An essential instrument in making any meeting efficient and worthy of its attendee's time is establishing a Meeting Bill of Rights, which gives a bunch of rights to all meeting attendees. These rights help establish clear expectations for what to expect in the meeting and how everyone should come prepared. Let's look at some simple rules that can be included in this Bill of Rights:

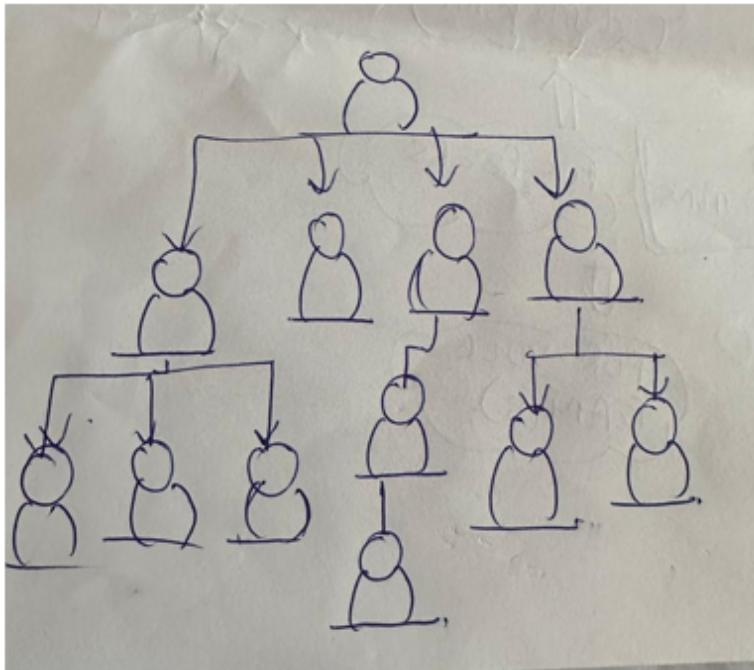
- Right to know the list of attendees
- Right to know the agenda, which is shared before the meeting
- Right to know the expected outcome of the meeting
- Right to ask for more details to prepare for the meeting
- Right to decline meeting if the attendee feels they are not the best person to provide inputs. Identify optional and informational people, so they can choose to attend or not or send a delegate.
- Right to drop off from the meeting if late by, say, ten minutes to the meeting
- Right to have their opinions heard

The above provides a list of some example rights that can be included in your meeting bill of rights based on your situation. Establishing such ground rules helps avoid wasting time caused by inefficient meeting runs. Also, having the agenda made pre-available to all attendees helps them figure out how they should prepare before they go into the meeting. Crucially as well, it gives the attendees time to make sure they are the right person who should be attending the meeting. If specific meetings require intense technical discussions, having an engineer from your team lead those discussions instead of you makes a lot more sense. It is important as an engineering leader to not spend all the time in meetings you are invited to, rather be available for your team members when they need you. This can be in the form of you organizing office hours where anyone can join and talk to you or open team forums to discuss together as a team.

## **14.2.3 Use the power of delegation**

Learning how to delegate effectively is an essential skill for EMs as it enables them to leverage their team's strengths, skills, and expertise to achieve shared goals while also prioritizing their own work more efficiently. However, it can be challenging for new EMs to master this skill. Take, for instance, the case of Jacob, a software engineer who recently transitioned to EM. Although he is well-versed in the code base, he encountered an issue he felt only he could resolve due to his previous knowledge. Jacob spent two days working on the problem, which, while done with good intentions, could have caused two issues. Firstly, the two days Jacob spent resolving the issue could have been used for other essential tasks that required his attention as an EM. Secondly, delegation provides opportunities for different team members to learn and grow, and this particular issue would have been an excellent opportunity for someone else to ramp up deeper into the tech stack.

**Figure 14.2 Delegation can leverage teams' strengths and skills**



Therefore, delegation was a critical tool that Jacob neglected to use in this situation, but he could have leveraged his strong engineering team to take on the task. By thinking strategically about roles and responsibilities and offloading tasks that others in the team can do and learn from, Jacob could

have managed his time more efficiently. To recap the key learnings about delegation, please refer to the chapter on "Delegation" in section 1.

#### **14.2.4 Learn to say 'no'**

Understanding where your time investment is unnecessary and learning to say '**no**' is an important skill to master. One of the biggest challenges you, as an EM, will face is over-commitment. This over-commitment can take various shapes and forms. Sometimes it can be overcommitting to the amount of work your team can do. At the same time, on other occasions, it could be overcommitment of what all you can do outside of your regular team deliverables. We all get 24 hours in a day. Taking on more than needed is bound to send you down a rabbit hole filled with underperformance, low productivity, and increased stress.

Saying no in a timely manner can help you prioritize your work more efficiently, as this frees up time for you to work on what matters. By saying no to low-priority and low-importance items, you free up time for things that are either high priority or high importance or both. This helps you focus on your core responsibilities as an EM more efficiently, which is managing the team and ensuring high-quality work. Lastly, by saying no and freeing up your calendar, you often get more time to introspect and reflect on the team's general health. Similarly, it is okay to bail out early from a meeting where the meeting isn't matching the goals of the meeting, and the attendees are just dragging the topic. In such situations, it is ok to permit folks to say, "You know what, I think we need to go offline and discuss, and I'll schedule a follow-on meeting, so let's adjourn, and you can have some time back." This gives you time to consider the team's long-term goals, including maintaining high-quality deliverables and process improvements that can streamline project delivery. Overall, prioritizing your calendar ruthlessly, identifying low-priority tasks which don't need your involvement, and just saying no can make a huge difference in how you manage your time efficiently. If your team is taking advantage of the RACI matrix (refer to Chapter 9: Working with cross-functional partners), if people are part of the 'informed' category, they can often skip the meeting and just catch up using the meeting minutes afterwards to protect the time.

## **14.2.5 Up-to-date to-do list**

There are times in the year when as an engineering leader, your meetings might exponentially shoot up for a short burst of time. Imagine a time when the annual planning coincides with the annual performance review. You will be inundated with a series of meetings and, at the same time, need focus time actually to write feedback and plan for the roadmap. In such scenarios, having a well-managed to-do list is a savior.

### **Did you know?**

#### [Getting things done \(GTD\)](#)

([https://en.wikipedia.org/wiki/Getting\\_Things\\_Done](https://en.wikipedia.org/wiki/Getting_Things_Done)) is a famous time management system developed by David Allen. The idea is to identify the critical tasks and create a list with clear time-boxed limits. Each should have an actionable item against them. This helps to keep the attention on the actionable items so one can make progress.

An updated To-Do list is an effective way to streamline your work and optimize your time. First and foremost, it helps you identify the tasks that need to be completed. This means you have a centralized place to refer to when deciding what to work on next. By ordering the list according to priority, deadline, or other factors, you can easily prioritize your tasks. To-Do lists have a range of benefits. As we previously discussed, delegation and the ability to say "No" are essential skills for an EM. A To-Do list can assist with both things, promoting timely communication, providing transparency in your work, and holding you accountable for completing tasks. Various apps, such as Trello, Apple Notes, and Reminders, can be used to maintain an updated list. If you prefer a more visual approach, sticky notes or a manual list in a notebook can also be effective.

## **14.2.6 Use feature/project tracker documents**

Project management tools like Jira, Asana, etc., are vital tools to help you manage your time efficiently. Such tools provide a comprehensive view of the projects that your team is supposed to deliver, inter-dependencies of

various tasks that are required to deliver such projects, their deadline, and progress all in one place. If you have such a tool (if not create one), with one click, you can see if your team is on track to fulfill all the commitments made. These tools highlight other important things early, saving you much time later. For example, looking at the time remaining and the task's progress, you can easily understand whether a particular initiative needs more resources to complete than initially planned. Of course, knowing this resource allocation at the start of the project is ideal; however, in most scenarios, this is not the case, as unknown complexities alter the initial investment plan. Not knowing about this almost till the end can put you under a lot of stress of failing to deliver, and you end up introducing more thrash in your team compared to not. Project management tools can provide you with this information holistically in one place. This helps bring your attention to areas that otherwise may have gone unnoticed and helps you reprioritize your work early, saving a lot of time later (refer to chapter 10: Project Delivery, management, and Execution to refresh the concept of project/feature trackers).

#### **14.2.7 Use async communication to the fullest**

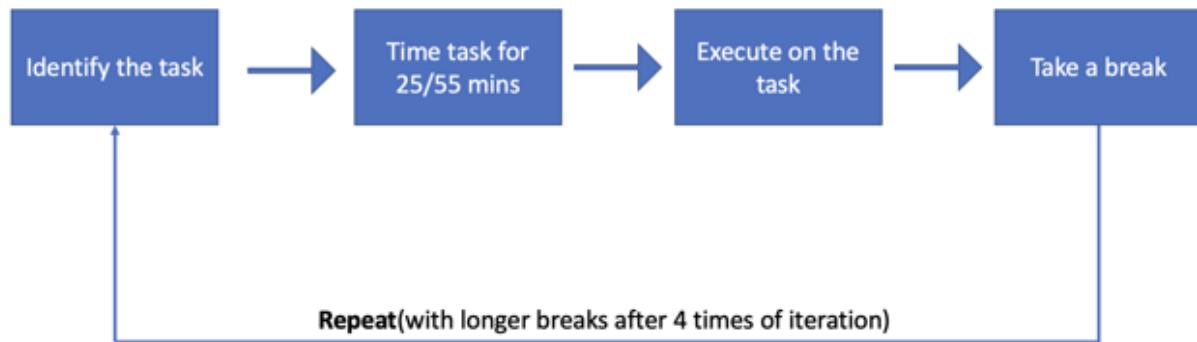
Asynchronous communication is a powerful tool for managing your time more efficiently. Unlike synchronous communication, which needs your attention now, async communication gives you the flexibility to look at a particular issue at your own pace and your own prioritization (obviously within a reasonable time to respond, which should be clarified given the task's priority). The main advantage of using async comms is that interruptions are limited significantly. We have already discussed that, as an EM, you must wear different hats. This means you are constantly switching between work and wearing different hats. As much as we would like to give ourselves credit for doing many things at once, at the base level, humans are single-threaded beings. In [research](#) (<https://www.apa.org/topics/research/multitasking>) published by American Psychology Association, it was conclusively found that switching between two tasks, especially two complex tasks, takes a significant toll on your productivity. You are bound to take a far more extended time for some work if you are constantly bogged down by interruptions. So avoid the urge to

respond immediately when reached out through async chat channels and respect your time blocks if the matter is not super urgent. One can always page/send direct messages to you if something is urgent and needs your immediate attention. Using async communication is a great way to give yourself this flexibility of responding in your time and getting things done simultaneously.

### Did you know?

Developed in late 1980 by Francesco Cirillo, a university student struggling with his own time management, the [Pomodoro Technique](#) ([https://en.wikipedia.org/wiki/Pomodoro\\_Technique](https://en.wikipedia.org/wiki/Pomodoro_Technique)) relies on breaking your work time into 30-minute chunks where you spend 25 minutes working on something with complete focus and then take a 5 minute break. After 4 such 30 min sessions, you take a more extended break of 15 mins.

**Figure 14.3 The framework for the Pomodoro technique**



This technique is widely regarded as a great productivity booster by minimizing your interruptions when working, decreasing your mental fatigue, and makes you more conscious of how you are spending your time. As a pro tip, try to schedule your meetings for 25-minute blocks or 55-minute blocks to give the attendees enough time to take a quick water or bio break or even a quick stretch as needed to avoid mental and physical fatigue.

Creating focus blocks on your calendar, which are just intended for you to get things done without interruptions, is a great way to utilize this process to its fullest. Use Slack, work chat, or emails to respond to things where you don't have to put out a fire. Before scheduling a meeting or accepting a meeting invite, ask yourself if this needs to be a meeting or if this discussion can be done asynchronously over team messaging apps. This mechanism can significantly decrease the amount of synchronous communication you need, which ultimately leads to lower interruptions and higher productivity.

## 14.3 The Eisenhower Matrix

“I have two kinds of problems: the urgent and the important. The urgent are not important, and the important are never urgent”

~ Dwight D. Eisenhower

As an EM, it's common to become trapped in the cycle of being busy all the time. You might find yourself constantly needed for various tasks related to project execution and people management. However, being constantly occupied doesn't necessarily mean that you're being productive. Even if you spend several days working on something, it's possible to feel like you haven't accomplished much. This process can be costly for both the company and yourself personally. The pressure to catch up can cause significant stress, impacting your professional and personal life. The primary reason for this is poor prioritization. It's easy to prioritize tasks with shorter deadlines but neglect the work that can benefit you and your team in the long term. This creates a cycle where you're constantly reacting to problems without being able to work productively. To manage your time more effectively, it's crucial to distinguish between urgent and important tasks. The Eisenhower matrix is a tool that can assist you in this process and in managing your time more efficiently.

### 14.3.1 Prerequisites for using the Eisenhower Matrix

The Eisenhower Matrix is a simple decision-making process that helps you prioritize your tasks based on their importance and urgency. Understanding

the difference between these two is highly crucial to make sound prioritization judgments and use the Eisenhower Matrix effectively. At first glance, every urgent task may seem necessary. However, a closer analysis can often lead you to very different results.

## **Urgent**

These include tasks that *demand* your immediate attention. If something is urgent, it means not doing them now will have clear consequences in some form or the other. These tasks can no longer be pushed to a later date, and delaying would mean more stress for you and your team. Some examples of such tasks are:

- Resolving a high-impacting regression in your services
- Delivering a high-priority business need with a due date that is very near
- Delivering a project that enforces legal compliance for the company

## **Important**

These tasks might not need immediate attention but help you achieve your long-term goals. Even though some essential tasks may not feel urgent, they are equally important to work on to make any noticeable progress in the long run. Some examples of such tasks include:

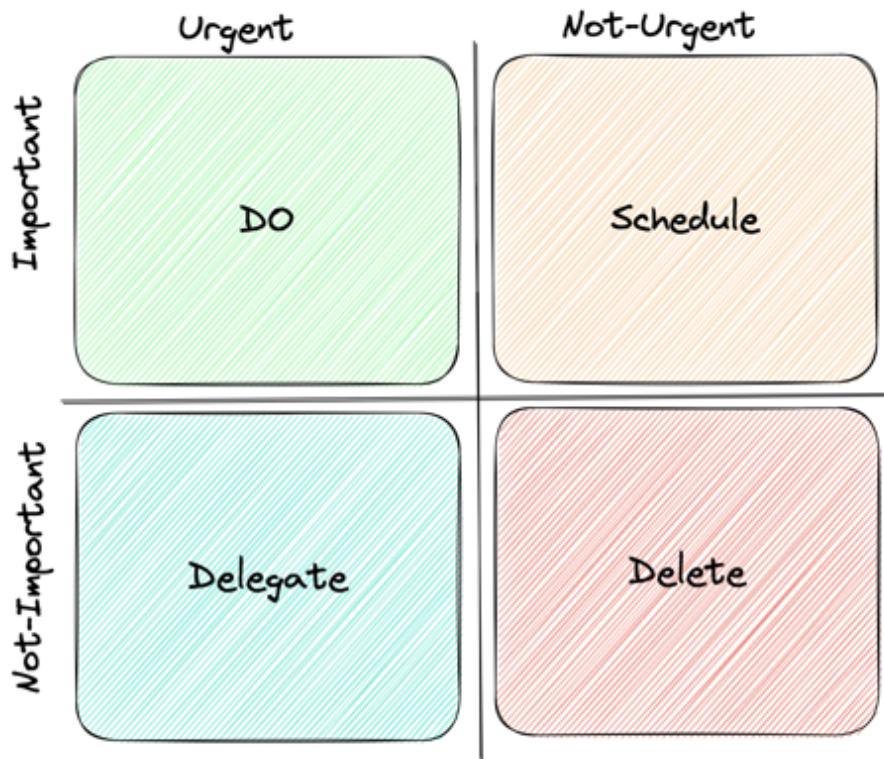
- Redefining how the on-call process looks like for your team to reduce the on-call load
- Planning a long-term project which achieves a significant goal
- Long-term maintenance strategy for the products and features your team supports

Once you are aware of this distinction between important versus urgent tasks, you can now use the Eisenhower Matrix to its full potential.

### **14.3.2 The four quadrants of the Eisenhower Matrix**

Looking at your long ToDo list may initially feel overwhelming. The Eisenhower matrix consists of four quadrants. The goal is to go through each item in your to-do list and categorize them into one of the four quadrants as shown below.

**Figure 14.4 The four quadrants of the Eisenhower matrix for time management**



### **Quadrant 1: Do**

These are tasks that are both urgent and important. These tasks require your immediate attention, have clear consequences, outcome and affect your long-term goals. These tasks are the easiest to identify as they are likely already at the top of your mind stressing you out.

### **Quadrant 2: Schedule**

These are tasks that are important but not urgent. When you encounter a task that doesn't extinguish a fire immediately but has a clear impact on your long-term goals, place them in this quadrant. You don't have to work on these tasks immediately, so you schedule them for later. This is the quadrant that often brings you the most satisfaction. People often don't spend enough time working on tasks from this quadrant because either they don't know what is important or have too many urgent requests that they are working on. A simple retrospective look on a regular cadence can help you understand how much time you have spent working on this quadrant. If it is not enough, then something needs to change. One important thing to remember about Quadrant 2 tasks is if you delay such tasks long enough, they will eventually get recategorized to Quadrant 1. Quadrant 1 is where all the stress lies, so don't delay such tasks indefinitely.

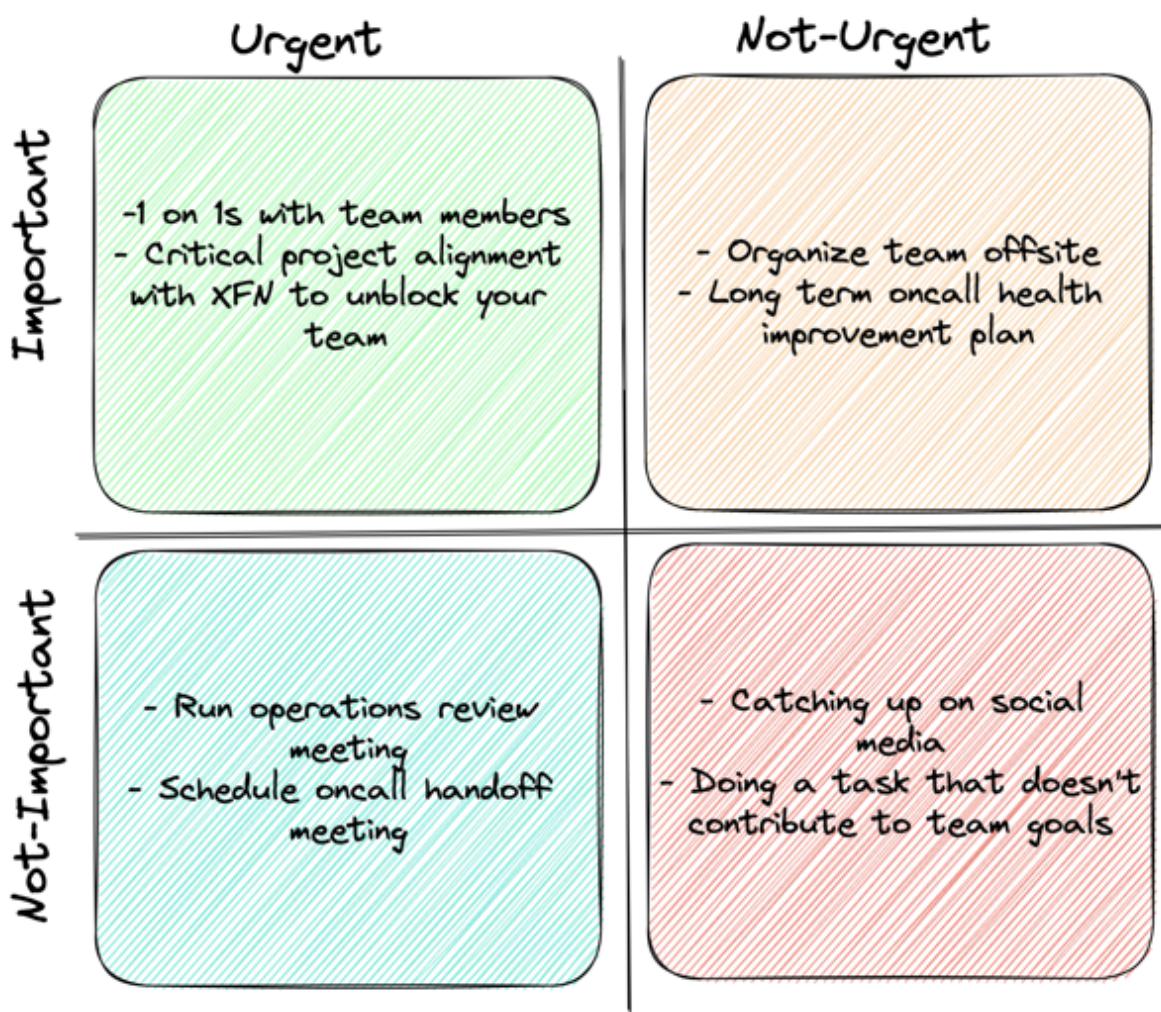
### **Quadrant 3: Delegate**

These are tasks that are urgent but not important. This is the place where people tend to spend most of their time. These are tasks that need someone's attention right away but have no impact on your team's long-term goals. The first question about any task that falls into this category is - can someone else work on this instead of you? Delegating is a key skill to master to manage your time efficiently, and almost all tasks that fall in this category can be delegated. Delegation will not only help you manage your time efficiently but will also help others in your team achieve their goals.

### **Quadrant 4: Delete**

While going down the list of To Do items, if you encounter any task that is neither urgent nor important, categorize them in this quadrant. Tasks from these quadrants are just distractions. Spending time on these tasks often gets in the way of achieving your goals. Therefore just delete these tasks. If doing a task doesn't really achieve any purpose, then there is no point in spending time on it.

**Figure 14.5 Some real examples for the four quadrants of the Eisenhower matrix**



The Eisenhower matrix is a powerful time management tool if used correctly. It can help you as an EM to evaluate the calendar and prioritize tasks. It further helps identify tasks that can be deleted or delegated so you can focus on things that are a good use of your time.

### 14.3.3 Spend time to find time

Prioritizing your time well is an important skill to master. However, this is easier said than done. One technique that can help you do that is to retrospectively understand your time spent distribution amongst these four quadrants. There are at least two ways to do this.

At the end of each workday, you can look at all the tasks that you spend time working on and write down a rough estimation for each of such tasks. Over a week, you will have a good idea of how much time you are spending on each quadrant. The main advantage of this approach is you don't have to spend a lot of time recording how much time you have spent on each task. You do it once a day, and a rough approximation is good enough for you. At the same time, the main disadvantage of this approach is the accuracy of your recorded time can vary greatly depending on how accurately you record your time. You are likely to miss certain things, over-count the time you spent on certain tasks, and undercount others.

Alternatively, and more accurately, break your workday hours into 30 min chunks where you work for 25 mins, note down in 2 mins at the end of each chunk to record what you worked on during those first 25 mins, and keep the end 3 mins for a quick break, can give you a far more accurate reading. As you continue recording how you spend your time, over a week or two, you can clearly see how much time you have spent in each quadrant.

Let's look at how you can rebalance your time to achieve your goals better and faster.

If you are spending too much time on tasks from Quadrant 1, this means you need to plan better. Of course, things will break, and escalations will happen. However, reducing tasks from this quadrant because of more advanced planning and anticipating what can go wrong, can dramatically reduce your stress and give you more time to work on things that you actually want to work on. If you are spending too much time on tasks from Quadrant 2, then congratulations. Keep on doing what you are doing, as this is the place where you want to be.

If you are spending too much time on tasks from Quadrant 3, then you need to delegate better. Delegating better means either you are able to find a person who can do the task instead of you, or you train someone on your team to step up and do such tasks for the team in the future. Effective delegation is key here, so work with your team to find a point of contact for different topics and work with them so that when the time comes, they can fill in for you.

If you invest excessive time in tasks from Quadrant 4, it indicates a significant waste of time. Although there might be some occasional benefits to completing tasks from this quadrant, it's best to eliminate them if you're struggling with time management. Review the importance of each task in this quadrant with your leadership chain or stakeholders, and discard them if they don't meet a specific threshold. If you already have a lot on your plate, these tasks should not be prioritized. Ensure to communicate this decision to the task requester or any other relevant stakeholders, but don't hesitate to decline and proceed with other tasks.

Time management is a unique and critical skill to master as an engineering leader. While there is no silver bullet solution to manage your time better, the sooner you acknowledge the importance of time and make planning for it a habit, the better. Procrastination of time management is like hurting yourself in the long run.

### **What do other leaders have to say about it:**

“I frequently say that journaling your to-do list is great (I bullet journal) but that it’s easy to fall into a trap of reducing the list size over doing the most important items on the list. It can be beneficial to cross off a few items to build psychological momentum to complete the longer and harder items.

The biggest thing is to acknowledge that there will always be more requests for your time than you can service, and you have to triage what needs to be done now and what can wait (both to take care of in the future and to see if it REALLY is something that needs to be done), what can be delegated, and what can just be dropped or declined. Your personal bias can sneak into that prioritization (doing things you like over needed things), so be careful.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“I maintain my To-Do and reprioritize the same every day I start my day.

Once a month, I retrospect my calendar to look out for no essential meetings and move them for offline discussions over live documents.”

~ **Madhur Kauria, Engineering Leader @Microsoft | ex Moengage, Oracle | IIT**

“I am a planner at heart. I review my calendar in advance and make changes as needed. Let’s say I need to block time for writing, find an optimal slot, and move meetings to another day/time. Distinguishing between deep focus time versus day-to-day tasks is important and helps me prioritize my day. Finally, I recognize that not every email and Slack needs to be responded to immediately and that each day will not be the same in productivity.”

~ **Nishat Akhter, Data Product & Engineering Leader, AWS**

“The most important aspect of time management is to ensure one spends the time on the most impactful tasks. Use a priority management tool like ‘The Eisenhower matrix.’ Block the calendar for every task to avoid distraction. Allocate the time you think each task will take you. Stop the task if you don’t complete it when the time you allotted ends and come back to it later. Assess why you failed and repeat the step again.”

~ **Saurabh Gandhi | Sr. Director Software Development Audible | Amazon | Ex-Amex**

## 14.4 Stop & Think: Practice questions

1. Do you feel you have a grip and control of your work time?
2. What tasks are taking away most of your time? Can they be delegated?  
Can they be automated?
3. Do you feel symptoms of burnout? If yes, do you feel you are racing against time?
4. What meetings on your calendar can be canceled or deferred?
5. How much time do your team members spend in meetings in a day?  
Are they getting enough maker's time?

6. If you could change one thing about your office calendar, what would it be? In what ways can you tackle it?

## 14.5 Summary

- Learning to manage your time is probably the most important thing you will need to learn to be a successful EM
- Managing time for EM's is particularly important because,
  - Helps manage successful execution of project deliverables
  - Helps you achieve personal goals in addition to team goals
  - You are working on a lot of time constraints such as constant meetings, constant interruptions, expectation of high quality work to be a role model for the team, so much work can be overwhelming and can lead to procrastination
  - Effective Decision Making
- Some tips that can help you manage your time efficiently
  - By decluttering your calendar and creating focus blocks for your own work
  - Establishing meeting bill of rights for all attendees to get the most of the meeting
  - Using the power of delegation to both load balance your work as well as provide opportunities for others to step up
  - By learning to say No when the ask is not worth your time
  - By creating a concrete ToDo list which can help you in prioritization
  - By maximizing the use of project management and project tracking tools
  - Use async communication to minimize the interruptions that you face
- Using the Eisenhower Matrix
  - A tool to manage your time more effectively
  - To use this tool, understand the importance between important vs urgent tasks. Not all urgent tasks are important and vice-versa
  - Classify your tasks into one of the following 4 buckets: Important and Urgent - Do them now, Important and not Urgent - Schedule them for later, Urgent but not Important - Delegate to others, Not

important and not urgent - Save your time and get rid of them altogether

- Spend time to find more time
- Use retrospective sessions to understand how your time distribution looks across 4 buckets. Either, at the end of the day, note down rough estimations of how much time you spent on each task, Or, for more accurate results, log what you have done in the last 29 mins at every 30-minute interval, whichever method you choose to do this for a few weeks to understand long term trends
- Rebalance and try to maximize working on tasks from bucket 2(Important and Not Urgent). These often tend to give you the greatest satisfaction and help you achieve your long-term goals

# 15 Beyond this- grow yourself

“Live as if you were to die tomorrow. Learn as if you were to live forever.”

~ Mahatma Gandhi

## This chapter covers

- Review and recap of what we learned so far
- Importance of continuous learning and improving skill set
- Valuable resources for the future that can come in handy in the journey

Finally, we have reached the concluding chapter of this book - a journey that has been quite a ride. Along the way, we have explored several concepts. As an EM, an integral part of the role is to facilitate learning, and guide others and yourself toward career development. It's about investing in the people and teams around you, rather than just yourself. Building, nurturing, and mentoring teams toward achieving greatness is incredibly rewarding. In this final chapter, we will review the key takeaways from the book so far and provide you with additional learning resources for the future.

## 15.1 What we learned so far

Now that we are at the end chapter, let's look at what we learned and recap the journey. We started with the understanding that an engineering manager (EM) role comes with various roles and responsibilities that differ from that of an individual contributor (IC) or the software engineer. We divided the book into three major sections:

### 15.1.1 Start with the people

In this section, we started with the basics of the role and purpose of engineering management and why we need it in the first place. We learned

about what traits help differentiate a good EM from a not-so-good EM so we know our do's and don'ts list and the various leadership styles that exist and how we can find our own style that fits us best. Next, we learn the apparent differences between EM and IC roles and how to decide what suits us. Also, if we are willing to move to EM, how to plan the move and set ourselves up for success? We learned about common pitfalls as a new EM and what to watch out for. Next, we focussed on the people aspect of managing our teams, ourselves, and the critical component of performance, which includes both high and underperformers. We learned about ways to keep them motivated and if we were to do the critical conversations of managing under-performance, how to go about navigating such situations using frameworks. Further, we learned the art of delegation and how to let go, a delegation framework to use in our day-to-day life, trust others and create the multiplier effect by teaching delegation to others. We also focussed on the critical aspect of building teams as an EM by learning about hiring frameworks and handling attrition to stay ahead of the curve.

### **15.1.2 Projects and the cross-functional world**

In the second part of the book, we learned about working with cross-functional partners and answering questions like how to work and collaborate with cross-functional partners for smooth execution and delivery of critical projects, building a roadmap for the team, the importance of collaboration and learning to communicate including technical and non-technical partners effectively. Next, we learned about project lifecycle phases and the EM involvement - pre-planning, planning and project kickoff, execution, and post executions steps. We ended this section on learning to manage expectations at all levels (up, down, and peers). It is about a shared need to build understanding, alignment, and cooperation at all levels. We learned about the common challenges faced when managing expectations and a framework to manage expectations.

### **15.1.3 Learn the process**

In the third section of the book, we learned about DevOps and Operational Excellence (OE)- its importance, tooltips to get you started, and learning to

practice DevOps and OE continuously. Next, we learned about the organizational change management process- learning about re-organizations, a framework for change management, and navigating through a change in leadership and workforce. We learned about the importance of time management, tips for better time management, and an understanding of the Eisenhower matrix for time management. In the next section, we will learn the importance of continuously learning and polishing the skills to keep an open learning mindset.

## 15.2 Continuously learn and hone skills

This EM journey you are on is all about you, you are in the driving seat, driving your career. It is essential to iteratively learn and continuously hone your skills to get better at leadership. Especially in this fast-paced world, where technology is evolving, it is essential to keep your skills up to date, be aware of your surroundings, and polish your EM skills. We saw throughout the book that I have laid emphasis on iterative learning, improvements, feedback, and much more. The idea is to set clear goals and expectations and build a roadmap to get to the goal state. As you go through this journey, you must translate the roadmap into actionable items. For example, learning about technology can take various forms- you can read a book, watch online learning videos, and can go a step further by attending or rather speaking at technical conferences related to the subject. You walk a few steps, find what you are missing, build on them, and continue the journey. I would recommend you to check out the triangle of Knowledge (<https://nealford.com/memeagora/2015/09/08/knowledge-breadth-versus-depth.html>) for learning, whether you want to go breadth-wise or depth-wise, based on what you want to do and where you want to start. As you take the EM journey, focus on foundational learning and practicing what you learned. For instance, yes we all have heard about delegation, but unless you practice it yourself as an engineering leader, you can not master it. Also, practicing and showing that to your team helps set an example that you can walk the talk so your team member believes more in you. In sum, I would say that it is important to keep an open mind to learn from your own experiences and the people around you so you are not wasting time re-

inventing the wheel and instead can focus on going broader and more profound by expanding your knowledge base and expertise. In the next section, look at some of the available learning tools in your inventory.

## **15.3 Some learning resources for future**

Yes, we are almost at the end of this book, and hence I would like to share some resources to keep handy to learn and grow as an engineering leader continuously. Below are some of the resources that have come in handy to me and some of my peer EMs, so let's explore them.

### **15.3.1 Find a mentor**

We discussed the importance of having a mentor in Chapter 2, IC to EM transition (revisit section 2.4 to refresh the concept). Mentors play a critical role in our lives by helping us in various aspects such as growing professionally, helping us with networking within and outside the organization, sharing their experiences and wisdom, providing career advice, and much more. It can be a series of meetings regularly (let's say meeting your mentor every month), ad-hoc check-ins, or even one-time informational chats. Usually, a mentor should be someone who is not in your immediate leadership chain or even in your organization. Some people confide their manager as the mentor and again, your manager is already part of your career development plan, but a mentor is someone who is there to enhance the experience and provide you that outsider perspective. Hence, it is better to find a mentor who is either in a similar role, a few levels above you or has walked the talk of something you would like to achieve in the future role and is not part of your immediate team. Another food for thought is that you can always have more than one mentor, maybe one inside the organization and one outside to give you candid feedback and also help you stay up to date on what's happening outside in the industry without keeping it within the four walls of your organization. Or perhaps one can be your technical mentor, and another can help you guide with business acumen or communications skills like public speaking and so on.

Now how to find a mentor? There are several ways you can get started to find a mentor for yourself.

- The first is to explore mentorship programs within your company. Most companies today usually have a mentor-mentee program where you can sign up and get paired. This is a wonderful opportunity to utilize the company program and also share feedback on the program for future improvements.
- [Linkedin](https://www.linkedin.com) (<https://www.linkedin.com>) is also another tremendous professional network to learn more about the professional journey of an individual and request mentorship or one-time informational chats. Some sample mentorship request messages can be found on [Indeed](https://www.indeed.com/career-advice/career-development/how-to-ask-someone-to-be-your-mentor) (<https://www.indeed.com/career-advice/career-development/how-to-ask-someone-to-be-your-mentor>) and [GrowthMentor](https://www.growthmentor.com/blog/how-to-ask-someone-to-be-your-mentor-templates/) (<https://www.growthmentor.com/blog/how-to-ask-someone-to-be-your-mentor-templates/>) to get you started.
- Some other forums that can be super useful are mentorship platforms like- [PlatoHQ](https://www.platohq.com/) (<https://www.platohq.com/>), [GrowthMentor](https://www.growthmentor.com/) (<https://www.growthmentor.com/>), [First Round Fastrack](https://fasttrack.firstround.com/) (<https://fasttrack.firstround.com/>), [ADP List](https://adplist.org/) (<https://adplist.org/>), [MentorPass](https://www.mentorpass.co/) (<https://www.mentorpass.co/>), [The Muse](https://www.themuse.com/coaching) (<https://www.themuse.com/coaching>), and several more.
- Another way to connect with potential mentors is by attending technical meetups in the area or other networking events where you get exposure to several successful professionals from the field and can learn from them.

A mentor can help you grow, hone skills, show you blindspots that you might have missed, or career opportunities that you might not have thought of. If you don't have a mentor, get started on finding your guiding star.

### **15.3.2 Learning platforms**

One can leverage some learning platforms to grow EM skills. Some of the common ways that you can get started are:

- Reading books on software engineering, leadership, and management. Some books apart from this book that I would recommend are:
  - [Turn the Ship Around- A True Story of Turning Followers Into Leaders by L. David Marquet](https://www.amazon.com/Turn-SHIP-Around-Turning-Followers/dp/B08V4TFFCK/) (<https://www.amazon.com/Turn-SHIP-Around-Turning-Followers/dp/B08V4TFFCK/>)
  - [The 7 Habits of Highly Effective People: Powerful Lessons in Personal Change by Stephen R. Covey](https://www.amazon.com/Habits-Highly-Effective-People-Powerful/dp/B0006IU4C0/) (<https://www.amazon.com/Habits-Highly-Effective-People-Powerful/dp/B0006IU4C0/>)
  - [Radical Candor: Be a Kick-Ass Boss Without Losing Your Humanity by Kim Scott](https://www.amazon.com/Radical-Candor-Be-Kick-Ass-Boss-Without-Losing-Your-Humanity/dp/B07XVQB7XV/) (<https://www.amazon.com/Radical-Candor-Be-Kick-Ass-Boss-Without-Losing-Your-Humanity/dp/B07XVQB7XV/>)
  - [The Managers Path by Camille Fournier](https://www.amazon.com/The-Managers-Path-audiobook/dp/B07SV4VDWC/) (<https://www.amazon.com/The-Managers-Path-audiobook/dp/B07SV4VDWC/>)
  - [The First 90 days by Michael Watkins](https://www.amazon.com/First-Days-Updated-Expanded-Strategies/dp/B00CH7FE1O/) (<https://www.amazon.com/First-Days-Updated-Expanded-Strategies/dp/B00CH7FE1O/>)
  - [Your next five moves](https://www.amazon.com/Your-Next-Five-Moves-Business/dp/B084B2X9TX/) (<https://www.amazon.com/Your-Next-Five-Moves-Business/dp/B084B2X9TX/>)
  - [Good strategy/Bad strategy](https://www.amazon.com/Good-Strategy-Bad-Strategy-audiobook/dp/B07R6XQ8YP/). (<https://www.amazon.com/Good-Strategy-Bad-Strategy-audiobook/dp/B07R6XQ8YP/>)
  - [Measure what matters](https://www.amazon.com/Measure-What-Matters-audiobook/dp/B07BMJ4L1S/) (<https://www.amazon.com/Measure-What-Matters-audiobook/dp/B07BMJ4L1S/>)
  - [The Making of a Manager](https://www.amazon.com/Making-Manager-What-Everyone-Looks/dp/0735219567/) (<https://www.amazon.com/Making-Manager-What-Everyone-Looks/dp/0735219567>)
  - [The First-time Manager](https://www.amazon.com/First-time-Manager-Jim-McCormick/dp/1400233585/) (<https://www.amazon.com/First-time-Manager-Jim-McCormick/dp/1400233585/>)
  - [The First time manager by Jim McCormick](https://www.amazon.com/dp/B07C66KTSD/) (<https://www.amazon.com/dp/B07C66KTSD/>) and many more.
- There are manager courses available on learning platforms like [Linkedin learning](https://www.linkedin.com/learning-login/) (<https://www.linkedin.com/learning-login/>), [Udemy](https://www.udemy.com/) (<https://www.udemy.com/>), [Coursera](https://www.coursera.org/) (<https://www.coursera.org/>), [EdX](https://www.edx.org/) (<https://www.edx.org/>), [Pluralsight](https://www.pluralsight.com/) (<https://www.pluralsight.com/>) and more. Such platforms usually have a subscription model with

unlimited courses to view and test your skills. You can use them to gain technical knowledge as well as management skills.

- Blogs are one of the ways to learn and also share your knowledge and experiences with others. It is also a powerful weapon if you are trying to build a personal brand for yourself. [Medium](https://medium.com/) (<https://medium.com/>) and [Harvard business review](https://hbr.org/) (<https://hbr.org/>) are one of the best platforms where you can go ahead to read blogs from other experts in the field and also start your own blog(on Medium). Some other platforms you can explore are [Wix](https://www.wix.com/) (<https://www.wix.com/>) and [Wordpress](https://wordpress.org/) (<https://wordpress.org/>). To get you started, you can share how you have learned and grown in your career, navigated challenging situations ([Norex](https://www.norex.net/) (<https://www.norex.net/>) is a great platform to explore), etc.
- A personal reflection on what you did, and what you could improve going forward is one of the easy low-hanging fruits to learn and improve as a leader continuously. It helps connect your past and present experiences to learn about future opportunities. Also, it helps dig deeper inside you to help identify your own strengths and weaknesses.
- If you want to go one step further, you can get access to executive coaches such as those provided by platforms like [BetterUp](https://www.betterup.com/) (<https://www.betterup.com/>).

Above are some ways to get started on learning and understanding what the EM role entails and to improve your skills as an engineering leader. What may work for one may not work for another, so see what interests you.

### 15.3.3 Pet projects

There is no age for learning and growing. As an EM, you will slowly get away from day-to-day coding skills and be part of more technical discussions. Pet projects in layman's terms are simply working on a small project in your (free) time to develop something small and valuable. It can also include making prototypes and evaluations within your company to keep you technically challenged, but without jeopardizing production code. This does not mean you must do a large-scale open-source project or build a

whole application with ten functionalities. The idea is to keep it simple and focus on the learning and foundational aspect of honing your technical skills. Pet projects are one of the best ways to learn and keep yourself updated with basic coding skills and industry technical standards. This habit helps you stay up to date on technology and at the same time helps you during technical rounds of EM interviews (usually done for first-line and mid-level EMs). This also helps build trust with your team of engineers that they are being led by someone who is technically sound. It is also an excellent opportunity to share it on [Github](https://github.com/) (<https://github.com/>) and discuss it during interviews. Another good resource for this is [Taproot](https://taprootfoundation.org/) (<https://taprootfoundation.org/>) where you can contribute pro-bono advice to non-profits, in the way of web development, architecture, and software development. This can be a rewarding experience and helps to keep your technical skills up to date. The last one I would share is [Kaggle](https://www.kaggle.com/competitions) (<https://www.kaggle.com/competitions>), where you can team up with other people and build connections. These have actual tasks and deadlines and are a gamified version of learning skills.

#### **15.3.4 Express opinions and work on effective communication**

Mastering effective communication is a vital skill for any EM. This goes beyond language fluency and involves the ability to articulate thoughts, opinions, and ideas to diverse audiences, including both technical and non-technical partners. Effective communication encompasses both spoken and written forms. New EMs may face challenges in this area, as their messages can be perceived differently by team members, not due to ill intent, but rather due to how they are expressed. For instance, consider a scenario where a new engineer-turned-EM is discussing a technical task with a team member. If the EM was an expert on that topic and believes they could complete the task faster, expressing concerns about the team member taking more time than expected can significantly impact morale.

How this crucial communication is conveyed plays a crucial role. If the EM comes across as boastful and self-centered, it could have disastrous consequences. On the other hand, if the EM shares their own learning

experiences and deep dives into the subject when they were new to it, the message will be completely different and more positively received.

Some of the ways you can work on honing your communication skills are:

- Joining a [toastmaster](https://www.toastmasters.org) (<https://www.toastmasters.org>) group that helps focus on effective communication
- Take advantage of attending and speaking at technical and leadership conferences. This helps you learn how other influential leaders break the ice and get the audience to make the journey with them and also gives you forums to practice and grow your speaking skills
- Another way is to attend a few online courses on Linkedin learning or other learning platforms(like TedX talks (<https://www.ted.com/watch/tedx-talks>)) to understand the basics and practice effective communication
- Publish articles through blogs or magazines, or even write a book. Some common platforms for writing content are google blogspot, medium, and Linkedin

We learned some ways to take action to be an effective communicator, following how interviewing regularly can help as an EM.

### **15.3.5 Interview frequently**

This might seem non-traditional advice to you, but it is essential to interview frequently for similar roles as you are in, even though you are not actively looking for a job change. Also, keep your resume up to date. This helps in various ways and can help boost your confidence as an interviewer and interviewee.

- Firstly, regular job interviewing helps you stay current on what is expected in your role, not just from your company but the entire industry. This also gives you a fair idea of what skill sets to focus on.
- It helps find potential prospective employers that align with your career potential. One of the casual interviews may be a better match for what you are looking for from your job and employer.

- There is no harm in learning about a new company and a new product through these interviews. It helps you be more aware of what products and services other companies offer. At the same time, you can learn so much about the industry, and what companies offer to their employees regarding work culture, perks, and compensation- perhaps you will feel more happy about your current role or feel like you should move.
- This also acts as an opportunity to build confidence, improve communications skills and help build a network. If you liked someone in the interview process, you can contact them on LinkedIn and request if you would like them to be your mentor or stay connected through the platform.

Ultimately, it is all about how much you crave to learn about new companies, products, and services and stay aware of your surroundings. Next, let's look at the power of embracing feedback to help you hone your skills.

### **15.3.6 Embrace feedback**

Feedback is one of the most important gifts we can get from others around us, be it personally or professionally. Accepting and receiving feedback is like a skill that helps us grow as individuals and gives us a chance to improve ourselves continuously. As humans, we have a natural tendency to think about feedback as negative criticism, but staying positive and optimistic can help us break that barrier and instead treat them as constructive golden words. Feedback can take various forms- personally, your close friends and family can help guide you to be a better human, similarly at the workplace, we have frequent reviews to help us understand what we are doing well and where we need to improve/adjust. At the same time, feedback can either be positive or negative, constructive or in the form of positive reinforcement. Feedback can help us be more self-aware and understand how others perceive our actions.

Feedback can help turn our weaknesses into strengths. Thinking about perfectionism is good, but no one in the real world is perfect. Hence, having a coaching mindset and being receptive to feedback opens up many

opportunities for us. As EMs, we should ensure not to sound defensive when receiving feedback and instead use it to our advantage to improve ourselves. Ask the feedback giver to share specific instances and keep a data-driven approach, so you know exactly what to improve and request suggestions. Feedback can be formal or informal. Informal in that you can ask how others think you are doing or more formal that is well documented. Again, feedback is a golden opportunity to help identify your blindspots, learn and grow- so make the most use of it to be a successful engineering leader.

## 15.4 Thank you

So with some tips on learning resources on how to continuously learn and grow your skills as a software engineering leader, we come to the end of this chapter and this book. I am grateful for your company in this journey. We started with learning about the role of an EM and how it differs from that of an IC or software engineer. We next deep-dived into the crucial skill sets that make an EM- managing your people, performance, learning to delegate, recognizing the individuals around us, hiring and building teams, and of course ensuring our people are happy to stay ahead of attrition. This all evolved the critical aspect of people management that makes up a significant share of the responsibilities of an EM. Next, we shifted gears and focused on the project delivery and execution aspect, learning about the skills needed for successful project delivery, working with our cross-functional partners, and managing expectations at all levels. In our final section, we focused on processes such as learning about DevOps and operational excellence, handling organizational changes, and managing our time well (yes as EM, your day will be driven by meetings a lot).

I hope you enjoyed reading the book as much as I enjoyed writing it. This is a reflection of my own journey as a software engineer moving to a software EM role and navigating through the ups and downs that came along with it. I hope you gained something in the form of insights, frameworks to use in your day-to-day, lessons learned, or simply a sense of camaraderie in your chosen profession. I wish you the very best of luck in your career and

journey to decide what role is the best fit for you. Again, I am sure you will do great.

You can always connect with me over [Linkedin](#) (<https://www.linkedin.com/in/akankshaguptamgr/>).

Thank you!

### **What do other leaders have to say about it:**

“I’m a lifelong learner and spend a lot of time on learning, besides access I’m a continuous reader for HBR (Harvard Business Review), Medium, and Linkedin. I’m an avid listener of podcasts and youtube. Chat GPT is on top of my favorites on my phone and computer. I frequently use it to understand technologies and assist with day-to-day work. I would recommend “Coaching Habit: Say, Less, Ask More & Change the Way You Lead Forever” written by Michael Bungay Stanier. It provides valuable insights and practical advice for becoming a more effective coach and leader by asking the right questions.”

~ **Sumit Kumar, System Engineering Manager at Cisco**

“There are two books I seem to be referencing a lot lately. The first is a novel, “The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win” by Gene Kim, Kevin Behr, et al. When working in an office, I had a stack of this book and would hand it out to anyone who would listen. It packages a lot of concepts that I care about into a story that resonates with nontechnical stakeholders in a way that works. I had a previous CEO as me “Am I [character who randomizes the team and then complains when the team misses deadlines].” It led to such a valuable conversation. The second is Patrick Lencioni’s “The Advantage.” In “we have to do everything” organizations, the six critical question exercises have teased out what is the best bang for the buck for the business. It also helps raise flags when an organization doesn’t know what it is, putting a magnifying glass on that question needing to be answered before we can reason about what work to do.”

**~ Nathan Bourgoin, Chief Technology Officer- Alakazam Inc, Technical Advisor, Engineering Leader**

“Do not stop learning. Stay humble and understand the business needs. Technology skills

alone will not be enough to forge beyond a certain organizational level.”

**~ Rajakumar Sambasivam, Delivery Manager at Microsoft**

“Hacker News is my main tech news source because of the high signal-to-noise ratio. If something is interesting, it’s usually there.”

**~ Jean Bredeche, Head of Engineering at Patch, (ex) Robinhood, Quantopian, Hubspot**

“As an EM one needs to be comfortable with playing different roles, and switching between them with ease. That also means that the EM’s learning path is diverse, and could range from being up to speed with the latest tech to learning how to build a high-performing team. That translates to converting every moment into a learning moment and diving deeper as needed. It’s amazing how much you can learn from daily interactions with your own team, stakeholders, and your customers.”

**~ Nishat Akhter, Data Product & Engineering Leader, AWS**

“Some suggested reading resources are: The 5 AM Club, The Coaching Habit, and Six Thinking Hats.”

**~ Saurabh Gandhi | Sr. Director Software Development Audible | Amazon | Ex-Amex**

“Beyond finding a good mentor in your industry, find peers at the same level as yourself, and one level above you. Local meet-ups are a great opportunity for this kind of thing. You can learn much from just a few opportunities to meet, chat, and share ideas and challenges. As you move up

the ladder in your career, likewise make yourself available as a mentor and coach to aspiring EMs and those you're grooming to take your place.”

~ **Bruce Bergman, Manager at Lytx**

## 15.5 Stop & Think: Practice questions

1. How confident do I feel about my role as an engineering leader in my organization?
2. What are some of the ways I continuously learn and grow myself? Are they enough?
3. Do I have a mentor or sounding board to help me improve and provide career guidance?
4. How is my attitude when my boss or peer gives me advice or feedback? Is there an opportunity to improve?
5. What are some of my resolutions to improve and hone my EM skills continuously?

## 15.6 Summary

- As an EM, an integral part of my role is to facilitate learning and guide others toward career development.
- We learned and recapped the three sections of the book:
  - **Start with the people** - In this section, we learn about the people aspect of the EM role, how it differs from that of a software engineer, challenges faced, transitioning from IC to EM, managing people, team, performance, learning to delegate, reward and recognize the people around you, hiring and getting ahead of attrition.
  - **Projects and the cross-functional world** - In this, we learned about working with cross-functional partners, project delivery, and managing expectations at all levels as EM.
  - **Learn the process** - Here, we learned about DevOps and operational excellence, handling organizational changes and restructurings, and the importance of time management.

- This engineering manager journey you are on is all about you, you are in the driving seat, driving your career. It is essential to iteratively learn and continuously hone your skills to get better at leadership.
- We learned about some resources to keep handy to learn and grow as an engineering leader continuously.
  - **Find a mentor** - Mentors are invaluable in our lives, assisting in professional growth, networking, sharing wisdom and experiences, and offering career advice. They can help identify blind spots, uncover hidden career opportunities, and refine skills.
  - **Learning platforms** - One can leverage some learning platforms to grow EM skills using Linkedin Learning, Udemy, Medium, and many more. What may work for one may not work for another, so see what interests you.
  - **Pet projects** - In simple terms, pet projects refer to working on small projects in your free time to create something useful. They are an excellent way to learn and stay updated with coding skills and industry standards.
  - **Express opinions and work on effective communication** - A vital skill for an engineering manager is effective communication. It goes beyond language fluency and encompasses the ability to convey thoughts, opinions, and ideas clearly to technical and non-technical stakeholders. This includes both verbal and written communication.
  - **Interview frequently** - It is essential to interview frequently for similar roles as you are in, even though you are not actively looking for a job change. This helps in various ways such as- boosting your confidence as an interviewer and interviewee, staying up to date on industry standards, finding potential employers, and learning about a new project or service.
  - **Embrace feedback** - Accepting and receiving feedback is like a skill that helps us grow as individuals and gives us a chance to improve ourselves continuously. Embracing feedback helps us learn and improve ourselves frequently.
- As we conclude this chapter and the book, I'd like to provide some learning resources to help you continually develop your skills as a software engineering leader. I appreciate your company throughout

this journey, and I hope you have gained valuable insights, frameworks for daily use, lessons learned, or a sense of camaraderie in your profession.

# welcome

Thank you for purchasing the MEAP for *Think Like a Software Engineering Manager*. I hope that the book's content is of immediate use to you and, with your help, the final book will be awesome! This book has been written by a software engineer at heart who eventually made the transition to the engineering manager role. Software engineering manager is a widespread role in the industry where usually you are tasked to manage a set of software engineers and sometimes even cross-functional partners. There are several resources available telling you how to hone your skills once you are in the role. But what about people who are learning about the role and maybe in future see themselves potentially in this role.

When I first decided to make the move from a senior software engineer to a software engineering manager, it hit me hard with the fact that there are few resources to help people like me, a software engineer, to understand about the role of engineering manager and make an informed decision if it's right for me or not. Also, once in the role, learning about what skill sets to hone to be effective at my job.

This book is precisely a reflection of my struggles and issues faced on a day to day basis as I worked on my role transition and how I navigated through them. In this book, my attempt is to provide you with real stories and exhaustive examples from the workplace to help you understand the fundamentals of what this role entails and once in the role, how to thrive in the role. Throughout the book, we will cover the core fundamentals of engineering manager and how to effectively think like one, given any situation. Also, I will share a set of frameworks and templates I have used to deal with challenging situations at work with ease. For a smooth reading, the book has been divided into three parts that dive deep into the three important pillars of an engineering manager role- the people, the projects/products, and the processes.

Each chapter is equipped with a short set of exercises- Stop and think questions to reflect back on what we learnt and answer some of the

questions on the topic and how we perceive it. Further, this book shares insights from experts in the field on what they feel about the topic and any personal anecdotes they have used in their jobs. By the end of the book, you should have learnt:

- What is management and how engineering management is unique
- Learning skill sets around people aspects(managing, delegation, recognition, hiring, attrition, etc.), projects(project delivery and execution, collaborating with cross-functional partners and managing expectations) and learning the processes(stay on top of DevOps and operational excellence, dealing with time and change management and learning how to continuously improve as an engineering manager)
- In case you are thinking about this path as a career move, it should provide you with additional data points to decide if this role is right for you or not.

Thanks again for your interest and for purchasing the MEAP! Please be sure to post any questions, comments, or suggestions you have about the book in the [liveBook discussion forum](#). Your feedback is essential in developing the best book possible.

-Akanksha Gupta

## In this book

[Copyright 2023 Manning Publications welcome brief contents](#) [1 The engineering manager role](#) [2 Individual contributor to engineering manager](#) [3 Managing people, teams, and yourself](#) [4 Managing performance](#) [5 Delegation - Learn to let go](#) [6 Rewards and recognition](#) [7 Hiring](#) [8 Handling attrition](#) [9 Working with cross-functional partners](#) [10 Project management, execution, and delivery](#) [11 Managing expectations](#) [12 DevOps and operational excellence](#) [13 Organizational change management](#) [14 Time management](#) [15 Beyond this- grow yourself](#)

THINK LIKE A  
**Software  
Engineering  
Manager**



Akanksha Gupta

MEAP

MANNING