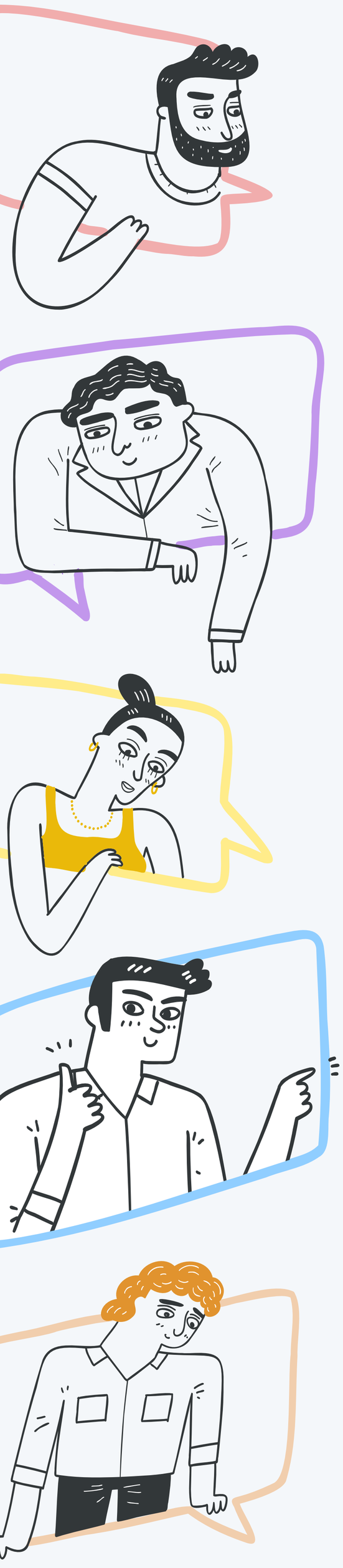
How to Be an Effective Engineering Manager

Mastering the Fundamentals of Engineering Management

Tips and insights on leading your developers
with empathy, context, and a sense of direction.

hay



Throughout my career, I've firmly believed that software engineering managers refine their skills through experience, and my journey was no exception. From the very beginning, I found myself immersed in the heart of management. I was responsible for recruiting the right engineering talent, guiding and overseeing engineers, managing projects, and making critical decisions that shaped both team dynamics and project trajectories. These experiences were instrumental in molding my leadership style and strategic thinking.

Years later, with a wealth of experience behind me, I now have a comprehensive understanding of what it takes to be an effective engineering manager. It goes beyond mere people management or technical proficiency. Let me share my insights into unraveling the complex and vital role of an engineering manager in the software development process.

Let's begin with the many roles of a software engineering manager.

The Role of an Engineering Manager

The Engineering Manager wears many hats, playing a multi-dimensional role that encompasses:

- Team Leadership and Development: Lead and nurture the software team, foster individual and collective growth, mentor, and promote continuous learning.
- Technical Oversight: Offering technical expertise and guidance and steering the team through technical challenges.
- Wellness Monitoring: Monitoring the team's wellbeing, fostering a supportive environment that values work-life balance and mental health.
- Performance and Output Assessment: Measuring team performance, work output, and maintaining a high quality without sacrificing morale.
- Strategic Vision: Guiding the engineering team's efforts to align with the organization’s goals, ensuring technical strategies bolster business objectives.

- Facilitate Innovation: Promoting innovative thinking, keeping up with industry trends, and implementing cutting-edge technologies and practices to boost productivity.
- Communication and Collaboration: Encouraging clear, open communication within the team and across different roles and departments.
- Culture and Environment Building: Creating a collaborative company culture focused on continuous improvement and collective success.

Understanding and Enhancing Developer Experience (DX) for Future-Ready Teams

In today’s rapidly evolving tech landscape, understanding and enhancing Developer Experience, commonly referred to as DX, is far from optional — it's essential for the success of both developers and the organizations they belong to. Let's explore how DX came to be an important part of building technology.

- Rapid Technological Expansion: Over the past few decades, we've seen an unprecedented increase in technologies, tools, and methodologies in software development. This proliferation has equipped developers with an incredibly diverse and extensive range of resources, significantly enhancing their capabilities and options.
- Fragmented Experience: However, this rapid expansion in the field has resulted in a fragmented DX. Developers frequently face the challenge of integrating these varied tools and technologies into a seamless and efficient workflow.
- The Need for a Holistic View: The industry has acknowledged the necessity of adopting a more holistic approach to DX. This involves crafting an environment that fosters collaboration and streamlines workflows for greater efficiency.

Why is Developer Experience Important?

- Foundation for Confidence and Impact: A thoughtfully designed DX empowers developers to operate with increased confidence and deliver more impactful results, leading to heightened levels of job satisfaction.
- Complexity in Software Development: Today's software development involves managing complex environments. A strong DX provides consistency across environments, processes, and workflows.
- Driving Business Success: Enhancing DX leads to better business results. Companies with superior DX outperform their competitors, irrespective of the industry.
- Statistical Evidence: Studies show that improving DX can lead to increased developer productivity, shorter time to market, and a significant impact on revenue growth.
- Wellness (Satisfaction): A positive DX significantly boosts developers' satisfaction and wellness. When DX is prioritized, it reduces stress and burnout, leading to a more engaged and productive workforce, and ultimately drives business success through increased innovation and efficiency.
- Mental and Emotional Wellbeing: In DX, wellness extends beyond physical health to include mental and emotional satisfaction in the workplace. It's about creating an environment that fosters empathy, psychological safety, and a healthy work-life balance.
- A Satisfied Developer is a Productive Developer: The goal is to ensure developers feel valued, respected, and heard. This not only enhances their productivity but also contributes to the overall success of the project and the organization.
- Teamwork (Collaboration): A robust DX cultivates a collaborative environment, essential for efficient and innovative software development. Streamlining communication and project management allows developers to synergize their efforts effectively, boosting productivity and the overall quality of work.

- Collaboration as the Lifeblood of DX: Collaboration is key to effective DX. It involves breaking down silos, fostering open communication, and creating an environment where ideas can be freely exchanged.
- Multiplying Effect of Collaboration: Collaboration enhances every other aspect of DX, leading to increased productivity, satisfaction, and innovation.
- Work Output (Tools, Processes): Effective DX optimizes work output by equipping developers with streamlined tools and clear processes. This leads to enhanced productivity and innovation, as developers can focus more on quality work and less on overcoming blockers, thus accelerating project timelines and aligning outcomes with business goals.
- Integration of Tools into Workflows: It's not just about having the latest tools; it's about how these tools and processes integrate into developers' workflows. The focus should be on reducing friction and making life easier for developers.
- Streamlining for Efficiency: By streamlining tools and processes, we can create a more cohesive and efficient workflow, enabling developers to focus on creativity and innovation.

DX is a complex yet vital aspect in determining the success of software development projects and teams. By emphasizing wellness, effective teamwork, and optimized work output, we can cultivate an environment that not only drives innovation and productivity but also ensures that our developers remain content and engaged. As we progress further into this technological era, it's imperative to focus on refining DX, laying the groundwork for building robust, resilient, and future-prepared teams.

How to Monitor Developer Experience?

To monitor DX effectively, it's crucial to blend direct feedback from developers, obtained through one-on-one meetings or check-ins, with performance metrics. Regular surveys and discussions provide insight into their challenges and needs, while data on development cycle times and tool usage offer a quantitative measure of workflow efficiency. This dual approach helps identify areas for improvement, ensuring developers remain productive and satisfied.

❤️ Wellness (Satisfaction):

Wellness extends beyond the scope of daily tasks. It encompasses the developers' contentment with their current projects, their rapport with team members, and their overall sentiment towards the company. In one-on-one meetings, I ask about their general wellbeing, inquiring if there are any concerns or if they're feeling good overall. It's imperative to create an environment where they can openly discuss any topic, ensuring their voices are heard, and their feelings are valued.

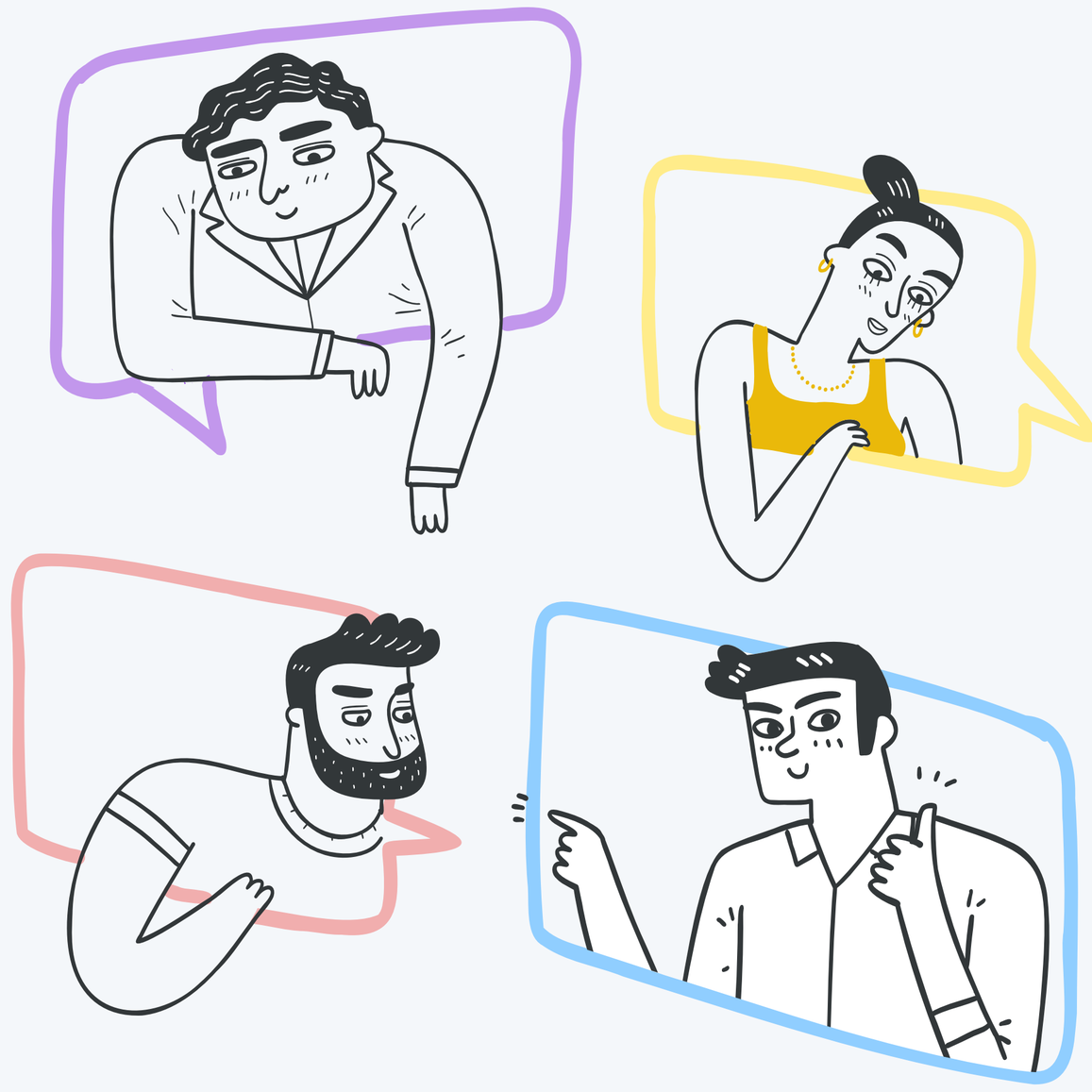
👩‍👧 Teamwork (Collaboration):

My focus then shifts to 'Teamwork.' I inquire about their interactions with other team members, seeking to understand whether they collaborate smoothly or face challenges. Questions like, "Are there any issues in working together, or is the team dynamic healthy?" help me gauge the state of team cohesion. This insight is crucial for identifying if any interventions or team-building efforts are needed to enhance our collective performance and workplace harmony.

💼 Work Output:

Finally, I address 'Work Output,' focusing on the specifics of their work. We review their recent accomplishments, discuss any challenging aspects they're encountering, and explore how I can assist. Discussions also surround their personal goals and how these align with our team objectives. Preparation is key for these conversations, and I also encourage them to come equally prepared. Preparation ensures that our discussions are productive and mutually beneficial. It also provides me with vital insights into how I can better support them and enhance our team's overall effectiveness.

Engineering Manager Ceremonies



As engineering managers, we often encounter complex situations with our team members, from performance reviews and conflicts to discussing promotions or continuous learning opportunities. Each of these situations requires a tailored approach, which is where the concept of 'Engineering Manager Ceremonies' comes into play.

In my management practice, I've developed what I like to call 'life jackets' – these are clear, step-by-step ‘algorithms’ crafted into simple, actionable guidelines.

Tailored for specific developer interactions, they are particularly useful when navigating high-stakes or emotionally charged situations, ensuring my actions safeguard my interests and remain balanced. These 'life jackets' are more than mere strategies — they are essential tools that help me respond in a manner that is considerate, effective, and in line with both my objectives and the needs of my team, even under pressure.

I categorize each distinct interaction with a developer as a 'boss ceremony.' These ceremonies are structured exchanges where I implement these guidelines to effectively manage a range of challenges. Be it a ceremony for resolving conflicts, providing performance feedback,

or negotiating retention, each scenario has a customized 'life jacket' to steer me through.

My aim with these 'life jackets' is to empower engineering managers, equipping them to navigate their managerial responsibilities with compassion, confidence, and a sense of direction. This guidance is designed to foster positive outcomes, benefiting both the managers themselves and their respective teams.

The Initial Address

This section will concentrate on mastering the art of addressing your new teammates in a new engineering management position. This ceremony is significant as it facilitates the connection between a new engineering manager and their new teammates.

When you become a new manager, you take charge of a part of the company.

From my experience, sharing what you believe in, what you expect from your team, and your rules right from the start is beneficial for several reasons:

- Clarity: It ensures everyone knows their responsibilities.
- Setting the Culture: Your initial address is an opportunity to establish the kind of work environment you envision.
- Authority: It allows you to set the rules and establish your leadership role.
- Avoiding Problems: Being clear about your expectations can lead to fewer misunderstandings later on.
- Making People Feel Confident: When team members understand your expectations, they can make more informed decisions and feel more confident in their roles.
- Efficiency: Clear guidelines enable the team to work more effectively, reducing the need for constant supervision.

Delivering an impactful introduction to your new teammates involves a clear structure and a straightforward approach. Here's how you can do it:

Show Your Human Side and State Your Intentions:

Addressing teammates as an engineering manager can be an emotional rollercoaster for everyone involved. The engineering manager might be concerned about how their message is perceived, risking unintentionally appearing more formal or unapproachable than intended. Meanwhile, the team might be speculating about the meeting's purpose, with thoughts like, 'Is the engineering manager going to implement strict new rules?', 'Are they aiming to assert more control?', or even 'Have we done something wrong?'

Here's a more practical approach: If you, as the engineering manager, are straightforward and clear about your intentions right from the start, it will help reduce misunderstandings and speculation. Being transparent about the purpose of your message can help your team understand what you're trying to say without all the unnecessary tension.

For example, you could start your address with something like:

'Hey, Team, let's huddle up for a quick chat. I'm not here to drop a bunch of rules on you. Instead, I want to talk about what makes our team awesome and what doesn't fly with me. We've all faced challenges, haven't we? Let me share my take on this...'

Or perhaps,

'Morning, everyone! Let's keep this casual. I'm here to clear

the air about how we do things around here. It’s not about laying down the law – more like setting us up for success. What behaviors help us, and which ones don’t.'

Another way could be,

'Alright everyone, I’d like to discuss something that has been on my mind. You know how it is, some things help us shine, and some ... well, they don’t. I don’t want to lecture anybody or point fingers; I just want to ensure we're all on the same page. What works, what doesn’t – let’s talk about this, huh?’

Define Your Minimal Limits:

After outlining your intentions, the next step is to establish your key boundaries. It's beneficial to communicate these in concise, unambiguous sentences that clearly convey your stance. Begin each statement with firm language that underscores the repercussions for team members who do not respect these boundaries. Here are some suggested formulations:

- I will not tolerate any behavior that undermines our team’s integrity...
- Should these boundaries be crossed, I will take the appropriate steps...
- Actions contrary to our values will be actively discouraged...
- In case of non-compliance, I will recommend other approaches....

Every manager must find the proper phrasing for themselves. My experience has taught me that it’s tough for a manager who oversees a group of independent specialists to use the phrase: 'I will draw consequences.' It is too reminiscent of school, tyranny, and lack of partnership relations. The sentences: 'I will discourage... I will kindly lead towards... ' are much easier for them.

Many managers shy away from explicitly stating potential

sanctions, preferring to use broader admonitions and requests. Common phrases include 'This is unacceptable...', 'Please remember...', 'I urge you to consider this seriously...', or 'The company's wellbeing depends on...'. However, this indirect approach is generally not recommended.

This approach weakens your authority, showing your team that you hesitate in confrontations and are reluctant to enforce consequences. An initial address should be more than a collection of wishes; it should communicate decisions clearly. An engineering manager's minimal boundaries must be specific and measurable in behavior, not just general attitudes.

Engineering managers will often address ‘lateness, disorganization, ineffective task performance, disloyalty, lack of cooperation, along with deficits in creativity and initiative.' But it's a challenge for team members to understand if they are truy meeting expectations or where they're falling short. How can they know if they've messed up or are performing poorly? What are the clear indicators for teamwork, initiative, responsibility, or creativity? To avoid future misunderstandings, refer to specific examples in this part of the speech.

- I will not tolerate delays in executing official tasks...
- If someone refuses to assist a teammate, there will be consequences...
- Repeating the same mistakes will be addressed...
- Providing inaccurate information in reports will lead to consequences...
- Consistent negativity or unconstructive complaints will not be overlooked...

As an engineering manager, I encourage you to acknowledge those behaviors that mainly affect you in your address. Of course, everyone has their quirks.

Name Your Maximal Preferences:

In this part of the address, you'll discuss your desired behaviors and attitudes. If you were to list all the behaviors that reflect qualities like independence, responsibility, or creativity, the list would be incredibly long and still incomplete. By consistently rewarding various manifestations of a particular attitude, your team will learn what matters and how they can earn your appreciation. I suggest using the following sentence starters when presenting your list of critical preferences:

- 'I will particularly highlight new ideas that contribute to improving our team's workflow.'
- 'I will most appreciate helping out teammates. I'll reward availability and commitment.'

I believe rewarding is a more essential and effective way to motivate your teammates than punishing them. If you agree, highlight this in your speech. Say that you prefer to reward rather than punish. Express your belief in the benefits of several attitudes that are important to you.

Emphasize that you will pay special attention to various ways of appreciating those who do more than what their job description entails. This will also help alleviate the team's fears when you discuss minimum standards and punishment.

Positive reinforcement allows for cultivating attitudes within the team that are important to a manager, while harmful consequences should only deter detrimental behaviors from a manager's perspective. You can't build anything of significance using only punishment.

Finish Your Address:

In your address, skip the fancy summaries. You've laid out your rules, so ensure your teammates have understood

Ithem correctly. Avoid asking for their opinions or opening discussions about the content of your speech, as it can become counterproductive. Instead, focus on clarifying any misunderstandings and ensuring they're clear on what will lead to rewards or punishments. Conclude with something like:

- 'Is everything clear? Is there anything I should add to make things more understandable?'

If faced with negative comments or tricky questions from the team, don't engage in arguments or get offended. Simply and calmly reiterate your rules and explain the interests you are protecting. Remember, such a speech is vital to forging a precise, secure manager-direct report relationship, and is crucial for setting up a consistent recognition system.

Accountability Session

After an engineering manager delivers the initial address, outlining what's expected and what's not, sticking to these rules is crucial. It's not just about handing out big punishments or rewards, but about being fair and consistent. It's like being in a band: each member's role is vital to create harmony.

Accountability sessions are tough and need practice, given the stress they bring for an engineering manager.

In an era where developers are often seen as rock stars, enforcing these standards gets trickier, especially when you value the person involved. But this is exactly what I'm here for – to lead these tough conversations.

Consistency is key in management, not the severity of punishments or rewards.

When talking with my friends or former colleagues who run IT companies, common concerns arise, like 'How do I punish or reward without the right tools? How often can I reprimand, scold, or threaten to fire? How do I show appreciation with limited funds?' I get these concerns, so I always prepare a thorough list of ways to acknowledge good work and viable sanctions. A manager needs a variety of informal punishments and even more forms of rewards for effective motivation.

Many bosses, myself included, hesitate to punish destructive behaviors, often out of fear or helplessness, leading to anger and disproportionate punishment.

I believe in graduated consequences to motivate change. Start with mild punishment for first offenses, leaving room to escalate if needed. For example, for missed deadlines:

- First time: A verbal warning, expressing dissatisfaction (Feedback and symbolic punishment).
- Next time: More detailed task reporting or extra documentation (Increased control and extra work).
- Further offenses: Removal from project, mundane tasks like unit tests or legacy code (Reducing responsibilities).
- Only as a last resort, use severe measures like bonus cuts, formal reprimands, or dismissal (Severe, formal punishments).

This approach ensures fair, progressive discipline, encouraging change rather than fear, and maintaining a balanced, productive work environment.

Addressing Underperformance and Behavioral Issues

Effectively addressing underperformance and behavioral issues involves a clear structure and a straightforward approach. Here's how you can do it:

Sample Scenario

Show Your Human Side: Start by being kind.

“Greg, this is a tough conversation for me. You're a thoughtful, sensitive person, and an engineer I've always admired. Your love for people and computers makes you a real hacker. It's hard for me to have this talk because I respect you.”

Address Reason for the Meeting: Say you need to talk about a work problem.

“We need to discuss some issues with your recent work that are concerning.”

Say What’s Wrong: Point out what the developer did that's not okay.

“You've taken on many initiatives but haven't been delivering on them, leading to a lack of reliability in your work.”

Remind Them of the Rule: Talk about the work rule that wasn't followed.

“As a PM, you're expected to lead projects effectively. Unfortunately, in projects like Gatsoma, RPCM, and GL Info, there have been significant lapses.”

Explain Why It’s Bad: Say how their action(s) caused a problem.

“Your approach has led to client dissatisfaction and internal challenges, impacting our team's performance and reputation.”

Say What Happens Moving Forward: Tell them what the consequence will be.

“I need you to improve in project management and execution.”

Hear Them Out: Let the developer respond, but just once.

“I'd like to hear your perspective on these issues.”

Offer to Help Later: Mention that you can talk more another time to help them improve.

“I'm here to support you. Let's schedule another meeting to discuss this further.”

Warn About Next Time: Make it clear that if this happens again, things will be more serious.

“Continued problems in these areas will force me to make serious decisions.”

Ask to Fix the Problem Behavior: If they can make things right, tell them to do so.

“I expect you to focus solely on delivering the Fatsoma project successfully in the next two months.”

Finish the Conversation: End the meeting.

“This situation is challenging for both of us. If our goals and expectations are not aligned, it might be better to part ways before more frustration sets in. However, I believe in your abilities and would like you to succeed.”

Respectful Dismissals

Let’s get into a topic that is both challenging and essential in the role of a manager - the process of employee termination. Letting an employee go is more than just a business decision; it's a significant moment that reflects the values and culture of our organization.

The Importance of the Termination Process:

Let's begin by addressing a fundamental question: why does the way a company handles the departure of one of its developers matter so much? The answer lies in the reflection of our humanity and professionalism. A properly handled termination process not only shows respect for the individual leaving but also conveys a sense of consideration and empathy to the remaining employees who observe these actions.

Emotional and Ethical Difficulties:

Terminating employees is an emotional challenge. Even experienced managers might feel discomfort when faced with the task of delivering such news. It's important to remember that employees often perceive termination as a personal failure or rejection. Our role is to minimize the negative emotions associated with this.

The Importance of Preparation:

We cannot underestimate the importance of preparation. Improvisation in such a delicate matter as termination can lead to unintended consequences, both legally and in terms of the company's reputation. It's crucial to have a clear plan and standards that help conduct conversations with employees in a dignified manner.

Extending Care and Support:

In addition to preparation, extending care and support during this process is vital. This involves not just the moment of termination but also the aftermath. Offering resources such as career counseling or assistance in job searching can go a long way in softening the impact of the news.

Communicating with Empathy:

The way we communicate the decision is just as important

as the decision itself. Using empathetic and clear language, avoiding jargon, and being honest yet respectful can help maintain the developer's dignity. It's about striking a balance between being firm about the decision and compassionate about its delivery.

Managing Team Morale:

The termination process doesn't end with the individual leaving; it continues with the team that remains. It's essential to address any concerns or questions they may have, reinforcing the company's values and their importance to the organization.

In conclusion, as managers, our responsibility extends beyond business outcomes to the people affected by those outcomes. The termination process, though difficult, is a test of our ability to maintain ethics, empathy, and professionalism.

How to Communicate the Decision to Dismiss

The algorithm of communicating the decision to dismiss is a way to tell someone they're being let go from their job in a kind and clear way. Each part of this method is important for making sure the conversation is fair and respectful. Here's what each step does:

Showing Kindness and Explaining Why You're Having the Meeting:

This step is about being kind and understanding. It helps the person feel respected and shows that you didn't make this choice easily. Telling them why you're having this talk helps them understand what's happening and prepares them for the news.

Telling Them They're Being Let Go and Why:

Saying directly that they're being let go makes sure there's no confusion. Explaining why helps them understand the reasons. This is important for being open and honest, which can make the person feel they're being treated fairly.

Sticking to Your Decision:

If you keep saying the same thing, it shows that the decision is final. This stops the conversation from turning into an argument and makes it clear that the decision won't change.

Talking About What Happens Next:

Discussing things like their last day and any money they'll get helps the person know what to expect next. This is important for ending their job in a fair and organized way.

Trying to End on a Good Note:

Offering help like a reference letter or advice shows you still think well of them. This can make the person feel better about the situation and helps keep a good relationship even after they leave.

Each part of this method helps make sure that letting someone go from their job is done in a caring, fair, and clear way. This is really important for the person leaving and for keeping a good feeling with the rest of the team.

Sample Scenario

Showing Your Human Side, Intentions, Difficulties, and the Purpose of the Meeting:

“Chris, this is one of the hardest conversations I've had to have. I truly value your contributions and who you are as a person. I want you to know that this decision was made with a lot of thought and consideration for everyone involved,

including you. It's never easy to reach this kind of decision, and I understand the impact it can have. The reason we're meeting today is to talk about your future with our company."

State Your Decision to Let the Employee Go and Offer a Brief Explanation:

"I've made the difficult decision that it's time for us to part ways. We've talked before about the challenges in project management, fulfilling mentorship roles, and your individual contributions. Despite our efforts, these issues have continued. This isn't a sudden decision, but a conclusion reached over time, considering various factors."

The Broken Record: Standing by Your Decision:

"I understand this is hard to accept, but my decision is final. It's important for both of us to handle this situation with the professionalism it deserves. I'm here to talk through this decision with you, but it's important to understand that the decision won't change."

Negotiate the Terms of the Termination:

"Let's discuss the next steps, including your notice period and any final settlements. I'm open to hearing any concerns you might have about the termination process. I'd like to make this transition as smooth and respectful as possible for you.

Bargain to Give Your Fired Developer Satisfaction:

"I'm willing to provide a positive reference for your future job endeavors based on your strengths. We will handle the communication of your departure to the team in a way that maintains your dignity and respects your privacy. I believe in your skills and potential, and I'm confident you'll find a role that's a better fit. I'm here to offer advice or feedback if you'd like".

Recognition and Rewards

Using rewards instead of punishment at work is really important. When you reward your developers for doing well, it works better than punishing them when they make mistakes. Punishments might make people do what you want right away, but they can also make the workplace unhappy and make people not trust each other.

Rewards can be things like saying "good job," giving someone a better position, or bonuses. These make people feel good and want to work harder. They know their hard work is seen and valued. When people are happy at work, they do their jobs better and get along better with others.

So, giving rewards instead of doling out punishment makes the workplace a better place. Everyone feels more supported and valued, and this helps the whole company do better.

How to Recognize and Reward

Start with Positive Emotions and the Purpose of the Meeting:

Begin by sharing how happy and proud you are of the developer's achievements. Make it clear that the purpose of the meeting is to recognize their hard work.

Remind Them What They Are Being Rewarded For:

Be specific about what actions or results the developer is being rewarded for. Highlight the particular tasks or projects that were done exceptionally well.

Explain How Their Actions Benefited You and the Company:

Tell them about the good things that came out of their actions, both for you as the boss and for the whole

company. This could be better results, a happier team, or happy customers.

Refer to the Initial Address (How the Desired Behaviors Stood Out):

Recall previous conversations or guidelines (the 'Initial Address') where you outlined what you expect and value. Show how the developer's actions fit into these expectations.

Define the Reward:

Clearly state the reward they are receiving. This could be a bonus, a promotion, extra time off, or another form of appreciation that suits the developer.

Listen to the Developer's Comments and Discuss the Sources of Their Success:

Give the developer a chance to talk about their feelings and thoughts about the reward. Also, have a conversation about what helped them succeed.

Promise to Keep Rewarding Such Behaviors and End the Meeting Positively:

End the meeting by saying that such behaviors will continue to be rewarded in the future. Make sure the ending is positive and leaves the developer feeling satisfied and appreciated.

This algorithm focuses on clearly communicating appreciation for specific actions or achievements, highlighting the importance of these actions for the boss and the company, and ensuring the developer feels valued and motivated for further growth.

Handling Complaining Developers

As an engineering manager, encountering developers who complain about something is a common challenge. It's important to have a strategy for addressing this, considering the various reasons behind such attitudes. Understanding the root cause of grumbling can guide your response and maintain a positive work environment.

Types of Complainers and Managerial Responses:

External Negativity:

Developers might grumble due to external influences that dampen their enthusiasm and positive work attitude. This could stem from criticism of the company, product, or service by colleagues, or unsettling internal rumors.

Manager’s Role: Reinforce the developer by focusing on their strengths and the positives. This 'detoxifying' process is crucial. Be ready to remind them of the positive aspects of your company and their role in it.

Problem Identification:

Complaints here might indicate a problem the developer faces but can't solve or clearly articulate. Their grumbling is a sign of helplessness and a desire for change, yet not know how to achieve it.

Manager’s Role: Assist in clearly identifying the problem, framing it in terms of impacted interests, and supporting the developer in finding solutions. Encourage a collaborative approach to problem-solving.

Chronic Complainers:

Some individuals are habitual complainers, viewing their constant criticism as a sign of engagement with the

company. They often resist participating in finding solutions, claiming it's not their job.

Manager's Role: Learn to 'mute' such behavior by offering clear choices and outlining consequences. Address persistent negativity firmly and consistently. This approach, referred to as 'muting the complainer,' involves setting boundaries and enforcing them.

Understanding the different reasons behind complaints allows engineering managers to respond appropriately, fostering a constructive and positive team environment. Each type of complainer requires a different approach, but the key lies in being clear, consistent, and supportive while addressing these challenges.

How to Handle Complaining Developers

I'd like to introduce you to a four-step procedure for talking with a complaining developer.

Listen

The manager should encourage developers to fully express their concerns.

This means attentively listening without interruptions or hastily offering solutions. Providing a space for them to air their frustrations is essential, as bottling up these emotions can be detrimental.

Key active listening techniques include:

- **Nodding and Acknowledging:** Non-verbal cues like nodding show you are engaged.
- **Paraphrasing:** Restate their points in your own words to demonstrate understanding.

- **Asking Open-Ended Questions:** Questions that can't be answered with a simple 'yes' or 'no' encourage deeper discussion.
- **Reflecting Feelings:** Acknowledge the emotions behind their words, such as saying, 'It sounds like that was frustrating for you.'
- **Summarizing:** Finally, summarize the key points to ensure clarity and mutual understanding.

Detoxify the Jaded Developer

After listening to the developer's concerns, the manager should help them see the bright side. This means talking about what makes the team and company great.

For example. Our team is really good at a few things:

- We're skilled in different areas, which makes us creative and good at solving problems.
- We talk to each other well, which helps us do our projects better.
- We stay calm and do well even when things are tough, showing we can handle challenges.

Also, our company offers great things for everyone:

- Training and Learning: We have lots of programs to help you grow in your career.
- Health and Happiness: We care about your health and offer support, like a gym membership or sessions with mental health experts.
- Recognizing Hard Work: When you do something great, we make sure it's noticed and celebrated.
- Helping the Community: Being part of our company means you can help with important community projects.
- Working With the Latest Technology: We use really cool and new tech, which is exciting and good for your skills.

All these things make our workplace unique and remind us why it's good to be here.

Identify the Problem:

The manager needs to help the developer articulate their specific problem. Often, developers may feel overwhelmed but are unable to pinpoint the exact issue. The manager can move the conversation from vague complaints to specific, solvable issues by guiding them to identify the core problem.

Assist in Finding and Implementing Solutions or 'Mute' the Persistent Complainer:

If the developer is willing to cooperate, the manager should assist them in finding and implementing solutions to their problems. This collaborative approach encourages developers to take responsibility and actively engage in problem-solving.

However, if the developer remains a persistent complainer without the intention to cooperate or find solutions, the manager might need to 'mute' them. This means setting clear boundaries and consequences for continued unconstructive behavior. The manager should be firm in addressing this behavior to prevent its negative impact on the rest of the team.

Each step transforms unproductive complaining into constructive problem-solving and engagement, fostering a more positive and proactive workplace environment.

Here’s An Example of 'Detoxifying' a Developer:

Manager (after listening to the developer): “I understand that you're facing challenges and it's a tough period for you. I want to offer you some support in this situation. Please, listen without interrupting, as my goal is not to argue or to persuade you, okay? Let me remind you of a few positive aspects that you might have overlooked: Remember that

you are part of a stable and renowned software development company. Our products are high-quality and highly regarded by a broad client base. You also work with talented, supportive colleagues in a collaborative environment, and the company is there to support you in various aspects. The company is committed to your professional growth, providing training opportunities and clear career advancement paths. Consider that, in comparison to our competitors, you have access to better technical resources and a more supportive work environment."

And the list goes on...

Executive Negotiation Interface

Effective communication with our team members is crucial because it highlights our role as leaders within the team. Additionally, another critical aspect of an engineering manager's role involves a different facet of communication: the intricate ceremony of negotiating with the CTO. This interaction goes beyond a mere routine exchange; it's a crucial ceremony that reflects our team's aspirations and needs.

The essence of these negotiations lies in acknowledging the human element of your superior. To establish common ground and successfully negotiate with your CTO, it's imperative to view them first and foremost as individuals, with their own perspectives and challenges. This human-centric approach lays the foundation for effective and empathetic negotiations, facilitating a dialogue that is both meaningful and productive.

Approaching high-level discussions with empathy is crucial, especially when aiming to devise solutions that resonate with both your team's interests and the organization's

broader objectives.

Embracing this empathetic mindset not only sharpens your negotiation skills but also solidifies your position as a crucial link between your team and upper management. This role involves ensuring that your team's needs and goals are not just conveyed but also adequately addressed within the larger framework of the company.

Let's look into this algorithm, from initiating the request to the actual process of negotiation.

The Request:

Begin with a clear and concise request. It's important to be straightforward to ensure your boss fully grasps what you're asking for. Avoid complex language and get directly to the essence of your request.

Get a Clear Answer:

Often, bosses may not give a direct 'yes' or 'no.' In such cases, you might need to persist in getting a clear answer to understand your position fully.

Understand Why They Said No:

If your request is declined, refrain from immediate counterarguments. Instead, seek to understand the reasons behind their decision. Inquire about their concerns, focusing on gaining insight into their viewpoint rather than convincing them to change their mind on the spot.

Find a Common Problem:

Once you've grasped their reasoning for refusal, formulate a problem statement that encompasses both your needs and your boss's concerns. This approach reframes the issue as a shared challenge, calling for a collective solution.

Work Together to Solve It:

With a common problem identified, invite your boss to jointly seek a resolution. This not only demonstrates your cooperative spirit but also turns the negotiation into a collaborative effort.

Highlighting Shared Goals:

In the final step, underscore the shared objectives and interests between you and your boss. Stress how reaching a solution is mutually beneficial, aligning with the team's and the organization's broader goals. This reinforces the cooperative nature of the negotiation, aiming for a win-win outcome.

These steps offer a structured approach to evolving from a simple request to an engaging, collaborative negotiation, promoting a constructive and mutually advantageous problem-solving process.

On Helping and Responsibility

Helping your team take charge and solve their own problems is a big part of being a good engineering manager. Consider this role similar to that of a gardener: while you provide care and support to your plants, ultimately, they must grow independently.

Sometimes new managers struggle to find the right balance. Some may attempt to handle everything for their team, while others might push their team members too hard. The ideal approach lies somewhere in the middle – offering assistance and support to your team, but also allowing them space to carry out their tasks and learn from their experiences.

Here's how you can do this:

Listen:

When your team tells you about a problem, listen carefully. Help them think of ways to solve it.

Give Them Control:

Ensure they know it's their job to get things done. You're there to help, but they do the work.

Teach Them to Learn From Their Mistakes:

When things go wrong, use it as a chance to learn. This helps everyone get better.

As a manager, your job is to help your team be independent and strong. It's about giving them the right help so they can solve their problems.

The Boss as a Mediator

Navigating workplace conflict and helping your teammates through their disagreements is an important part of being an engineering manager. It’s not always easy, but your approach can make a difference in how your team works together.

Here's our approach to evolving into effective mediators, going beyond the role of mere supervisors. These aren’t just techniques but a mindset that can positively influence our approach to conflicts and team dynamics.

Guide, Don't Judge:

The first thing to remember is not to take sides. You're like a coach, not a judge. Your role is to encourage everyone to talk it out and find their own solutions. So, instead of saying who's right or wrong, help them focus on how they can resolve their issue.

Listen to All Sides:

Each person in the disagreement has their reasons. Ask them, "Why do you feel this way?" or "What's important to you here?" It's all about understanding their perspective. By showing that you're listening, you help create an environment where everyone feels valued.

Create a Framework for Resolution:

It's helpful to set some ground rules for resolving the conflict. Let your team know the consequences if they don't work things out, and give them a clear deadline. This creates a sense of urgency and shows that resolving the conflict is important.

Highlight Common Goals:

Sometimes, reminding your team of their shared goals can help them see past the conflict. It could be as simple as saying, "Remember, we all want this project to succeed." This can shift their focus from the conflict to what they can achieve together.

Be Neutral and Fair:

One of the most important things is to stay neutral. Don't favor anyone. Your role is to facilitate a fair discussion. Make sure everyone has a chance to speak and be heard.

Encourage Solution-Finding:

Encourage your team to come up with solutions. Ask them, "What can we do to solve this?" This empowers them to take ownership of the issue and the solution.

Based on these five guidelines, I've developed a procedure that I try to remember when I find myself in a situation where I need to act as a mediator

Briefly Listen to the Conflict:

Hear out the disputing teammates briefly to understand the gist of their conflict.

Establish Meeting Rules and Define Your Role:

Set the rules for the mediation and clarify your role as a mediator, not a decision-maker.

Understand Starting Positions:

Compel both parties to clearly and concisely state what they want from you.

Identify Interests of Both Sides:

Ask each party, "Why is this important to you?" to understand their interests. Paraphrase their needs to ensure clarity.

Enforce Paraphrasing of Each Other's Needs:

Require your teammates to paraphrase the needs of their counterparts to promote understanding.

Formulate the Negotiation Problem:

Define the problem that your conflicting teammates need to resolve. Write it down as they might forget what they need to work on.

Outline Consequences and Set a Deadline:

Inform them of the consequences of continuing to fight versus cooperating and set a final deadline for resolving the issue.

Remind Teammates of Shared Interests if Necessary:

If needed, remind your teammates of their common

interests to encourage collaboration.

To be an effective mediator and instill a sense of responsibility in your developers, it's important to avoid favoring one side's reasoning over the other. Acting as a neutral facilitator, rather than a judge who delivers verdicts or determines who is right, is crucial, as taking sides can ultimately have negative repercussions.

The mediation algorithm clearly shows that as a boss, your main task is to define your teammates' problems, not to solve the problems for them.

In Conclusion

The role of an engineering manager, as we've seen, extends far beyond the confines of traditional management or technical expertise. It's a dynamic blend of leadership, strategic vision, and a mastery of rituals that turn good managers into great ones! Of course, this can't be achieved without understanding the human element that drives successful software development.

As you move forward in your career, remember that each challenge is an opportunity to grow, and every decision you make shapes not just your path, but that of your team and your organization.

May this guide serve as an inspiration in your ongoing pursuit of becoming an exceptional engineering manager 🚀

If you have any questions, write!

Tomek

tomek@howareyou.work

CEO @ HAY

Handy links:

Blog · Course · Contact Us · HAY Slack Community · Linkedin · Youtube · Twitter · Facebook