Chapter 5 : How to Lead A Team

Leading a software engineering team effectively requires a combination of technical expertise, strong communication skills, and the ability to motivate and empower team members. Key aspects include setting clear goals, fostering collaboration, providing constructive feedback, and promoting a culture of continuous learning and improvement.

A software development team consists of such roles as product manager, engineering manager, tech project manager, software architect, software developers, designers, testers, and team/tech leads.

Roles in a Software Development Team

1. Product Manager

A Product Manager (PM) is the quarterback of your software development team, a pivotal role in a software project. Their responsibilities cover a vast terrain: crafting and implementing strategies, planning and monitoring product KPIs. PMs typically set ambitious goals to achieve company targets through their products. For this, they keep a keen eye on competitors, thoroughly analyze the market, generate ideas, and manage pricing policies.

2. Product Owner

Another important role in a software development team is a product owner. This expert is accountable for maximizing the value of a product. To do so, they need to understand the client’s specific needs to define goals and create a cohesive vision for each member of a software development team.

A product owner’s core responsibilities revolve around:

- crafting a compelling product roadmap
- managing the product backlog
- defining the product vision
- acting as a liaison between the development team and stakeholders
- evaluating feedback
- ensuring maximum value to the end-users

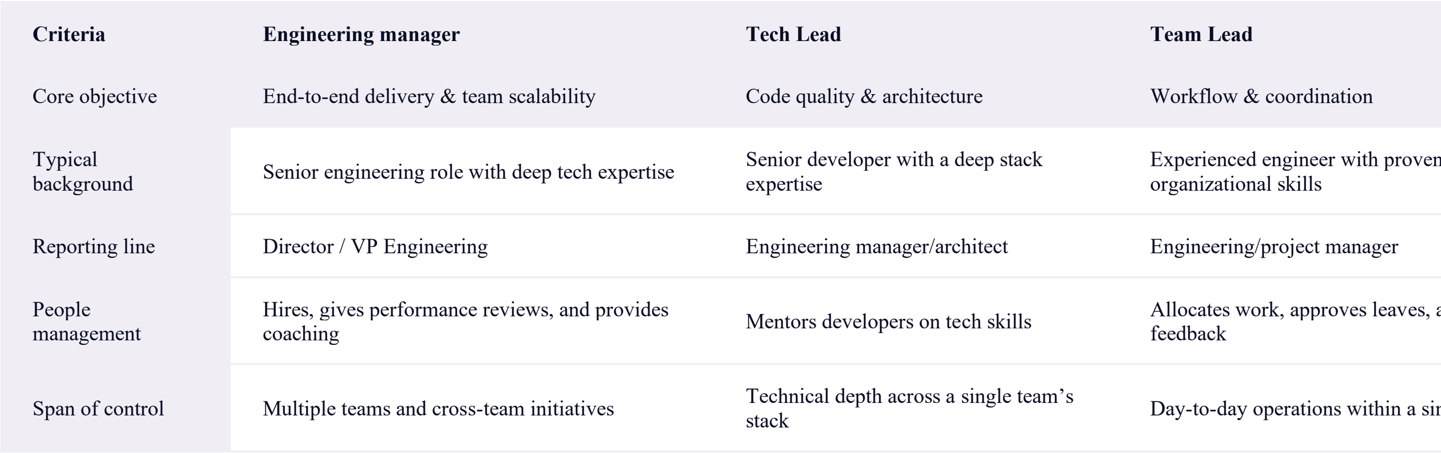
3. Tech Project Manager

- The tech project manager (TPM) has become one of the indispensable roles in a software engineering team. They bridge the gap between technical teams and business objectives, ensuring projects are delivered on time, within scope, and aligned with organizational goals. Depending on the company, this role may also appear under titles like Delivery Manager or Program Manager. TPMs are involved in all phases of a project, from planning and execution to risk management and stakeholder communication.

4. Business Analyst

Business Analysts study and gather all information about the product, outline the target audience's needs, and act as a bonding agent between the market and engineers. They analyze users' behavior/concerns and then compile product enhancement and development recommendations. In many agile environments, where product owners or product managers absorb parts of this role, BAs often focus more on requirements elaboration and process modeling.

5-7. Engineering Manager vs Team Lead vs Tech Lead



Criteria	Engineering manager	Tech Lead	Team Lead
Core objective	End-to-end delivery & team scalability	Code quality & architecture	Workflow & coordination
Typical background	Senior engineering role with deep tech expertise	Senior developer with a deep stack expertise	Experienced engineer with proven organizational skills
Reporting line	Director / VP Engineering	Engineering manager/architect	Engineering/project manager
People management	Hires, gives performance reviews, and provides coaching	Mentors developers on tech skills	Allocates work, approves leaves, a feedback
Span of control	Multiple teams and cross-team initiatives	Technical depth across a single team's stack	Day-to-day operations within a sin

8. Software Architect

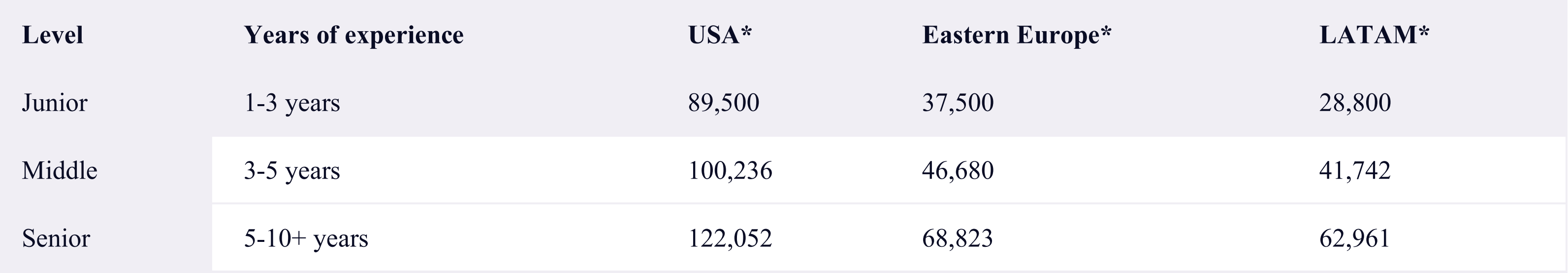
This specialist takes the lead in orchestrating the internal arrangement of the software and aligning it with product requirements and available resources. Grounded in business needs, architects outline technical and functional instruments essential for product development. But they aren't mere planners – architects roll up their sleeves and get hands-on in the development process. Not to mention that they craft the very design of the system.

These experts should have a strategic vision as they plan for software enhancements, extensions, and the addition of new features. A master of optimization, a perfect software architect navigates the complexities to find the optimal technical solutions that align seamlessly with the product, while also driving adherence to coding standards and influencing long-term tech direction through internal governance practices like tech radars.

9. Software Developers

While discussing important roles in a software development team, programmers are crucial. These are the wizards who bring the code to life. They use various programming languages, frameworks, and libraries to make it possible.

Due to their experience, coders have three ranks, and our latest salary benchmark reveals what each level costs in the USA, Eastern Europe, and LATAM:



Level	Years of experience	USA*	Eastern Europe*	LATAM*
Junior	1-3 years	89,500	37,500	28,800
Middle	3-5 years	100,236	46,680	41,742
Senior	5-10+ years	122,052	68,823	62,961

Alcor 2025 developer salary monitoring (average annual, USD, gross)

They all collaborate closely with designers, testers, product engineers, and other team members. This ensures a seamless parallel workflow where everyone's expertise intertwines to build exceptional software, with developers also taking ownership of unit testing and participating in code reviews to uphold quality across the codebase.

10. UX/UI Designers

Among the key roles in the software development team are also UX/UI designers. User experience (UX) and user interface (UI) may be distinct, but they are harmoniously intertwined. Both demand a blend of creativity and technical prowess.

UX designers ensure the browsing experience and features are accessible to users. They meticulously analyze the target audience's behavior to sculpt an interface that guides users effortlessly through the product. This often includes close collaboration with product managers and product owners during discovery workshops to align on user needs, goals, and product direction.

On the other hand, UI designers translate those insights into pixels – choosing fonts, colours, motion, and micro-interactions that express brand and boost usability. Although these are two team roles in a software development process, the UX/UI designer often wears both hats.

11. QA Engineers

Quality assurance anchors theapplication development team's roles and responsibilities, ensuring every release is thoroughly vetted before it reaches users. In a high-velocity software engineering team, QA Engineers partner with the coding crew, DevOps, and product management to keep quality metrics in the green and rework in check.

A QA Engineer is a common role in the software development team, typically divided into two specializations: Manual and Automation QA – both essential to safeguarding software quality from ideation to deployment.

Manual QA Engineers take on the role of the user. They walk through real-life scenarios, verify functionality, and identify usability gaps that automation may miss. Their meticulous testing uncovers edge cases, validates new features against requirements, and provides detailed bug documentation to steer developers away from costly mistakes.

Automation QA Engineers, on the other hand, build scripts and frameworks to scale testing efforts. They apply coding skills to automate regression checks, run performance tests, and monitor system behaviors under pressure. After executing automated test runs, they analyze results and compile reports – ensuring fast feedback loops without sacrificing depth.

Tips for Building a Perfect Software Development Team

1) Have a clear scaling vision

Having a clear product vision is a solid start – but it won't get you far without a plan for execution. That means figuring out the structure of your future team: size, tech stacks, and hiring location.

2) Prioritize hiring T-shaped programmers

Then, you can move on to filling different roles in your software development team. As you may know, the workforce can be divided into two categories based on their skills: generalists and specialists. My advice is to get the best of both worlds and focus on hiring generalists who possess specialties. Such experts are typically referred to as T-shaped developers. Not only do they have in-depth technical knowledge and hands-on coding experience, but they also demonstrate advanced collaborative abilities, emotional intelligence, proactivity, empathy, and problem-solving skills.

3) Tap into a foreign talent pool for rare expertise

Scroll a global tech job board and you'll see Kyiv, São Paulo, and Bogotá posting the same niche roles once found only in San Francisco. According to our IT market analysts, emerging Europe boasts over 1.8 million tech professionals – making it the second most concentrated region for ICT talent after Western Europe – while venture investment in LATAM start-ups jumped 26% in 2024 and is forecast to grow again in 2025, reflecting a deeper bench of experienced engineers.

4) Set ambitious yet realistic goals for your developers

Having business goals is a no-brainer, but you shouldn't overlook the real MVPs of your success – your software engineers. Getting a handle on each team member's career goals is crucial while keeping an eye on the overall growth of the development team. Consider using the SMART approach to set goals for software engineers. You've got a buffet of choices: career, technical, interpersonal, leadership, and community goals, which can be divided into long-term and short-term ones.

5) Establish clear communication

Clear and constant communication is a cornerstone of a highly productive and successful development team. This holds even truer for remote teams that miss the luxury of instant communication. Tools help (yes to Slack, Notion, Jira, etc.), but the rituals make the difference.

Weekly department check-ins keep everyone aligned. Monthly one-on-ones provide an opportunity to discuss what's working and what's not. Feedback should be honest and regular – sugarcoating kills productivity faster than conflict ever will.

6) Create a pleasant work environment

The software development process can be quite overwhelming, exacerbated by the risk of burnout from working overtime. According to a report by Haystack Analytics, the pandemic increased burnout in 81% of software developers. This led to decreased performance, a lack of motivation, and even instances of quiet quitting. As a manager, it becomes crucial to prioritize and create a healthy work-life balance for your programmers. This may include flexibility in working hours, fair compensation adjustments, enhanced benefits packages, and opportunities for growth.

Another thing to consider is corporate culture. According to Eagle Hill's survey, 76% of workers believe culture significantly influences efficiency. Therefore, focus on clearly defining your company's mission, core values, and goals, and promoting transparency. This attracts top talent and improves retention, creating a positive and thriving work environment.

7) Competitive employee value proposition

A first-rate Employee Value Proposition (EVP) is no longer optional when you're competing for senior developers who can pick from a dozen offers. Salary still matters, but the edge often comes from benefits that speak to long-term upside and daily quality of life.

Why developers say yes:

Market-matched basics: remote-first policies, learning budgets, and private health cover keep you level with top competitors

True ownership: stock-option plans turn engineers into partners, not renters. The promise feels real when backed by clear vesting schedules and fair exercise windows.

Moving from an Individual Contributors role to a Leadership Role

Introduction:

Are you an Individual Contributor (IC) aspiring to step into a leadership role? The journey might seem daunting, but in this article, we'll explore the qualities and qualifications essential for a successful career as an engineering leader and how to actively cultivate them. Drawing from personal experiences and insights, my goal is to provide clarity on the expectations and responsibilities of an Engineering Manager (EM). Hope it will help you!

Qualities and Qualifications:

Engineering management demands a unique skill set, combining technical prowess with interpersonal abilities, learned through hands-on experience. EMs orchestrate, supervise, and streamline engineering projects, teams, and operations, requiring an intricate understanding of the software engineering lifecycle and business dynamics.

Core Responsibilities:

Let's go over some Key Traits of Exceptional EMs.

1. **Effective Communication:** The ability to convey complex technical concepts to both technical and non-technical stakeholders creates a shared understanding within the team and the organization.

2. **Empathy and Emotional Intelligence:** Understanding and empathizing with team members creates a positive and collaborative work environment, building trust and support.

3. **Delegation Skills and Trust:** Effective delegation empowers the team, emphasizing the importance of trust in leadership.

4. **Flexibility and Negotiation:** Navigating trade-offs and being open to compromise are crucial in challenging situations with multiple stakeholders.

5. **Leadership and Vision:** Providing a clear vision, aligning goals, inspiring the team, delivering transparent and constructive feedback, and having a strategic mindset contribute to long-term success.

Conclusion:

Success as an Engineering Manager goes beyond technical knowledge; it requires a balance of leadership, communication, strategic thinking, and adaptability. By blending these qualities, an EM can navigate the complexities of team management and contribute to both individual and organizational achievements.

How to Lead a Team

- The **engineering manager** is responsible for the performance, productivity, and happiness of every person on their team while still making sure that the needs of the business are met.
- The **tech lead** (TL) is responsible for the technical aspects of the product, including technology decisions and choices, architecture, priorities, velocity, and general project management.
- The TL will usually work hand in hand with the engineering manager to ensure that the team is adequately staffed for their product and that engineers are set to work on tasks that best match their skill sets and skill levels.
- On small teams, the default is often to have a TLM: a single person who can handle both the people and technical needs of their team.
- The job of TLM is a tricky one and often requires the TLM to learn how to balance individual work, delegation, and people management.
- If you've spent the majority of your career writing code, you typically end a day with something you can point to and say, "That's what I did today." But at the end of a busy day of *management*, you'll usually find yourself thinking, "I didn't do a damned thing today."
  - As tempting as it might be to focus on purely the technical health of the team, **the social health of the team is just as important**.

It's time for leadership to grow up

"Before I got married I had six theories about raising children; now, I have six children and no theories."—John Wilmot

There's a tension that keeps repeating over and over—when things don't go well; there's a tendency to blame it on 'the people.'

I help organizations build a culture of change—to become more experimental, innovative, and adaptive. When I kick-off a project, I receive a brief from senior executives. Most of the times, the diagnosis focuses on how their teams are not performing *as they should*.

A collection of the patterns that you don't want to follow if you want to be a successful manager:

- Hire Pushovers.
- Ignore Low Performers.
- Ignore Human Issues.
- Be Everyone's Friend.
- Compromise the Hiring Bar.
- Treat Your Team Like Children.

A collection of positive patterns for successful leadership and management:

- **Lose the Ego:** Apologize when you make a mistake. You need to sincerely mean it. Apologizing doesn't cost money.
- **Be a Zen Master.**

  A Zen master is a spiritual leader and teacher in the Zen Buddhist tradition, recognized for their profound understanding of Zen teachings and practices. They guide students through meditation, koan practice, and direct pointing to help them realize their true nature and attain enlightenment. Zen masters embody wisdom and compassion, serving as living examples of the transformative power of Zen.

- **Be a Catalyst.**

  In the context of team building, "being a catalyst" means acting as a facilitator or agent of change to encourage growth, collaboration, and positive interaction within a team. A catalyst leader helps reveal team members' strengths, aligns their values, and energizes them to work effectively towards common goals

- **Remove Roadblocks.**

  "Removing roadblocks" means eliminating obstacles or impediments that prevent or slow down progress towards a goal. It involves identifying and addressing anything that interferes with smooth workflow or achievement. In a practical context, this might involve resolving logistical issues, clarifying communication, or providing necessary resources.

- **Be a Teacher and a Mentor.**

  A teacher acting as a mentor provides guidance, support, and professional development to other teachers, particularly those new to the profession or seeking improvement. This relationship is characterized by trust, mutual respect, and a focus on fostering growth in the mentee. Mentors share their experience, offer feedback, and help mentees navigate the challenges of teaching.

- **Set Clear Goals.**

  Setting clear goals means defining specific, achievable objectives with well-defined boundaries and a concrete vision of what you want to accomplish. This clarity provides direction, motivation, and a framework for tracking progress towards your desired outcomes.

- **Be Honest.**

  In the context of team leadership, "being honest" means acting with integrity, transparency, and truthfulness in all interactions with the team and stakeholders. It involves communicating openly, providing constructive feedback, and building trust through consistent and reliable behavior. Essentially, an honest leader is someone who can be relied upon to be truthful, fair, and ethical in their actions and communication.

- **Track Happiness.**

  Different studies suggest that happiness indeed makes people more productive. The direct relationship between morale and productivity prompts employers to try to measure their own team's happiness. It's not surprising, as there's a lot on the line here, business-wise.

A good simple way to track your team's happiness is to ask the team member at the end of each one-on-one meeting, "What do you need?" This simple question is a great way to wrap up and make sure each team member has what they need to be productive and happy.

- **Hire Pushovers.**The concept of "hiring pushovers" refers to the practice of hiring individuals who are easily persuaded, influenced, or defeated, often lacking assertiveness or a strong backbone. While this might seem appealing for short-term ease or control, it ultimately hinders team growth, innovation, and overall success.

What are the benefits of being a pushover?

Being a pushover, or someone who tends to be overly accommodating and agreeable, can have both positive and negative implications. Here are some potential benefits:

1. **Harmony in Relationships**: Pushovers often prioritize others' needs, which can lead to a harmonious environment and fewer conflicts. This can foster positive relationships with friends, family, and colleagues.
2. **People Skills**: Being accommodating can enhance interpersonal skills. Pushovers may develop a keen sense of empathy and understanding, which can be valuable in both personal and professional settings.
3. **Flexibility**: Pushovers are often adaptable and willing to compromise, making them easy to work with in group situations. This flexibility can be beneficial in collaborative projects.
4. **Avoidance of Confrontation**: For some, being a pushover helps them avoid confrontations and the stress that comes with them. This can lead to a more peaceful day-to-day life.
5. **Supportive Nature**: Pushovers are often seen as supportive individuals, which can attract others who appreciate their willingness to help and accommodate.

- **Ignore Low Performers.**
  Ignoring low performers can negatively impact team morale, innovation, and overall productivity. While it might seem easier in the short term, it's crucial to address underperformance to ensure a healthy and effective work environment.

Ignore Human Issues

The instruction "Ignore Human Issues" likely refers to a directive to disregard or minimize the importance of human-related problems, challenges, or perspectives in a specific context. This could be in the context of a decision-making process, a design project, or a scientific analysis, where the focus is on technical aspects or other factors rather than human impact.

Ignore Human Issues

…………………………………….

Be Everyone's Friend

Compromise the Hiring Bar

Compromising the hiring bar means lowering the standards or requirements for new hires, which can negatively impact the quality of the workforce and potentially lead to future problems. This can manifest as accepting candidates who don't fully meet the necessary qualifications or failing to properly assess candidates' skills and experience

I regret every time I compromised on hiring. Compromised because of pressure, desperation, or lack of better options.

My first hiring decisions were easy—pick a couple of the brightest interns, and leave the one that is the funnest to work with. But as I grew more into management it got more complex: limited candidates, different expertise requirements, remote work, internal competition for the same candidate. And so I rushed and made some mistakes.

I defined 3 avatars:

1. Low performer — instant to hire; produces 50% output;
2. Medium performer — takes a month to find; 100% median output;
3. Great performer(top 10%) — 300% output, but takes 6 months to find;

Also, I took into account the opportunity cost of not hiring — the company loses money without an employee. And for simplicity's sake let's assume performance is static over time.

Treat Your Team Like Children.

Treating a team like children, particularly with an emphasis on control and micromanagement, is generally detrimental to team morale and productivity. Instead, leaders should foster a mature, respectful, and trusting environment where team members feel empowered and accountable.

Why Manager is a Four-Letter Word to Me

I have a personal disdain for the word "manager." When I think of things to manage, I think of a 401(k), irritable bowel syndrome or a crazy ex. When I think of the word manager, I think of someone **who is in charge that asserts their authority**. Often this authority is used negatively reacting to adverse situations. That's not who I am and not what I do. Yet I am a manager. This internal conflict is the reason why "manager" is a four-letter word to me.

That's what got me to thinking. People don't want to be "managed" or "**retained**" at all, they want to be engaged, taught and supported.

Engineering managers are experienced engineers who currently manage technical activities in both large and small organizations. No matter the size of the company, an engineering manager makes decisions that are part of the management chain of command and impact the success of technical activities. Making data-based technical decisions is at the core of what an engineering manager does. The success of an engineering manager's career depends on these decisions. And in today's ever changing technology-based world, these decisions are not easy.

Thus, engineering management is an in-demand, growing field that's constantly evolving to accommodate modern technical concepts like artificial intelligence (AI) and big data and other technologies emerging from the fourth industrial revolution. Engineering managers are needed in project management, director, and executive roles across engineering fields, including mechanical, civil, environmental, chemical, biomedical, biotech/pharma, industrial, aerospace, aeronautical, and petroleum engineering. Aspiring engineering managers will play a valuable role in the key technical decisions that will help these industries adapt to an ever changing technological world.

If you aspire to advance your engineering career, an engineering manager role can pave the way for many diverse leadership opportunities. The duties of an engineering manager include planning, designing, and overseeing projects, as well as managing finances and supervising one or more engineering teams. The higher the engineering manager rises up the technical management ladder, the more engineering teams and managers fall under him/her. The work of an engineering manager requires a combination of skills and knowledge including engineering expertise, personnel management, leadership skills, and financial and business acumen.

What is an Engineering Manager?

An engineering manager is an experienced engineer who manages technical activities and team(s) of engineers, related professionals, and support personnel. This type of manager has likely already proven herself/himself to be a highly competent engineer, who showed a sense of vision, teamwork, and responsibility.

At some point, an early-career engineer may face a career decision: either stay technical (instead of leaving the engineering and technology directorates) and remain an individual contributor engineer, or expand his/her skillset in order to become an engineering manager. To climb the individual contributor ladder, that engineer must typically earn graduate degrees in a field of engineering like the Master's degree or a Ph.D., in mechanical, electrical, chemical, civil, or architectural engineering or in the computational sciences (i.e., computer science, statistics, and computational and applied math). Those who decide to become engineering managers can first earn a Master's in engineering management (MEM) degree, or as it is known at Rice, the Master's in engineering management and leadership (MEML) degree. Figure 1 below shows these two technical career tracks for the engineer at a technology company. It should be noted here that in both of these tracks, the engineer has decided to remain technical and work amongst top technical talent.

For the engineering manager, he/she has decided to lead engineering without leaving engineering. Yet, the engineering manager is a part of an organization's management chain of command, the levels of management that make both strategic and operational decisions. The strategic decisions require that an engineering manager communicate and 'manage up' to executives. The operational decisions require that an engineering manager lead and manage down with the teams doing the technical work and interact horizontally with management peers in other divisions such as finance, marketing, human resources (HR), and information technology (IT).

Across industries, an engineering manager's strategic responsibilities involve communicating with key stakeholders to determine project goals and objectives and report on outcomes. These key stakeholders are normally executives and customers.

An engineering manager's operational responsibilities involve overseeing engineering projects and teams. They help strategize, design, and manage projects at every stage, ensuring that the projects are completed efficiently, correctly, and on schedule. Engineering manager duties also include working with and managing each member of the engineering team(s).

While the strategic and operational responsibilities of an engineering manager are consistent across industries, the role will vary across engineering specializations.

- **Mechanical engineers** work with a team and oversee the execution of technical and design operations. Mechanical engineering professionals are also tasked with administrative aspects of the project including financial planning and budget and coordinating work schedules. As the engineering field continues to advance, mechanical engineers will leverage I4 technologies like Big Data, AI (machine learning), digital twins, 3d printing, and thermal-mechanical sensors. These technologies enable engineering managers to use simulation and data-driven innovations to test products and troubleshoot potential problems.
- **Chemical engineers** work with teams and technicians to design and implement processes to manufacture chemicals and other products. In addition to managing teams and personnel, chemical engineering managers develop the procedures that define how a project works to ensure teams are meeting deadlines and following best practices and compliance regulations. Chemical engineers will use I4 technologies including chemical and biological sensors, AI, and simulation based engineering to help digitally transform operations. These advanced technologies will also help to create smart supply chains and chemical processing plants and streamline internal processes.
- **Civil engineers** lead technical and design teams on civil engineering projects, like buildings and infrastructure. The civil engineering manager is charged with supervising team members and personnel and coordinating budgets and schedules. As the engineering field continues to evolve with I4 technologies like building and environmental sensors, AI, and mobile device based smart and connected platform products, civil engineers will be

able to digitize and automate various aspects of their jobs, leverage real-time data processing thanks to IoT, and plan projects virtually before arriving on site.

Types of team leads at Google

- There are two types of engineering team leads at Google: engineering managers and tech leads. They focus on people's needs and technical needs, respectively.
- Engineering manager is responsible for the performance, productivity and happiness of every person on their team, including their tech lead. Engineering manager needs to find the careful balance between the needs of the product and the team.
- Tech lead is responsible for the technical aspects of the product, including technology choices, architecture, priorities, velocity, and general project management. Most tech leads are also individual contributors.
- On small teams, there can be a "tech lead manager" handling both the people and technical needs of the team. It's very hard to do both of these jobs well in a growing team, however.
- All engineering team leads at Google are expected to have an engineering background.

ANTIPATTERNS

Antipatterns are common, yet ineffective or counterproductive solutions to recurring problems in software development, project management, or business processes. They are the opposite of best practices, often leading to maintainability issues, errors, and inefficiencies.

Management antipatterns

Antipattern 1: Hire pushovers

If you as a manager feel insecure, one way to ensure that no one questions your authority is to hire people you can push around. Strive to hire people who are smarter than you and can replace you. This way, you can learn and grow. If your team doesn't need you anymore, you've succeeded.

Antipattern 2: Ignore low performers.

The most difficult part of dealing with humans is handling someone who isn't meeting expectations. The most difficult cases are when someone isn't capable of doing their job no matter how long or hard they work. Here applies the Google SRE motto: "Hope is not a strategy." Do not hope that low performers magically improve or go away. You need to act quickly, before the team morale gets sour or you're so frustrated you can't help them. Set up a time frame and some very specific goals to achieve in that period. Make the goals small, incremental, and measurable. Either the low performer will notice themselves they can't keep up, or they'll up their game to meet the expectations.

Antipattern 3: Ignore human issues.

It's common for managers to be stronger in the technical side and ignore human issues. Managers can forget to treat their reports as individuals with specific needs and, for example, enforce the same working habits for everyone.

Antipattern 4: Be everyone's friend.

Many newly-promoted leads don't want to lose the friendships they've cultivated with their teams and therefore try very hard to maintain friendships after becoming team leads. This can be a recipe for a disaster. You can lead a team and build consensus without being a close friend or a monumental hard-ass. You can be a tough leader without tossing friendships to the wind. One way to stay socially connected without making the team uncomfortable is to have lunch with the team. Avoid the situation where you end up managing a close friend who is not self-managing and not a hard worker.

Antipattern 5: Compromise the hiring bar.

A common (bad) approach to hiring is that when a team needs to hire five engineers, it sifts through a pile of applications, interviews 40-50 people, and picks the best five candidates regardless of whether they meet the hiring bar. This results in a mediocre team. The extra cost of finding the appropriate person pales in comparison to the cost of dealing with an employee who should not have been hired in the first place. This cost manifests in lost team productivity, team stress, time spend managing the employee, and the paperwork and stress involved in firing the employee. Without the raw materials for a great team, you're doomed as a team lead.

Antipattern 6: Treat your team like children.

You must respect your team and trust it to get the job done. If it's permanently necessary to micromanage people because you don't trust them, you have a hiring failure in your hands. If you hire people worthy of trust and show these people you trust them, they'll usually rise to the occasion (assuming you've hired good people). Google trusts their employees enough to give them access to self-service "tech stops" where they have free access to all sorts of technical equipment.

Positive patterns

Positive pattern #1: Lose the Ego.

Humility comes first in "humility, respect, and trust". Being humble is not the same as lacking confidence.

Cultivate a strong collective team ego and identity. Trust your team and respect their abilities and prior accomplishments. Giving your team ownership gives them a greater sense of accountability and responsibility.

You do not need to have all the answers. If you act like you do, you will lose the respect of your team. Find people who give you good constructive criticism. Avoid the urge to be territorial.

Apologize when you make a mistake. You need to sincerely mean it. Apologizing does not make you vulnerable. Instead, you will earn respect.

Positive pattern #2: Be a Zen Master

Engineers typically develop an excellent sense of skepticism and cynicism. As a team leader, you need to be less vocally skeptical while letting your team know you're aware of the intricacies and obstacles involved in the planned work.

Always maintain your calm. The higher you up in the company hierarchy, the bigger effect your behavior has on the organization. The leader is always on stage and always being watched. Do you show confidence or fear? You attitude spreads infectiously to your team.

A VP of engineering at Google, Bill Coughran, was always so calm that people used to joke that if someone told Bill that 19 of the company's offices had been attacked by space aliens, Bill's response would be, "Any idea why they didn't make it an even 20?"

As an engineer, you're always solving problems and answering questions. In a leadership position, you need to leave the solution mode and enter the asking mode. Help your team find the answers.

Positive pattern #3: Be a Catalyst

Catalyze your team. Help them build consensus. Do not rely on authority, unless that's what the team accepts and needs.

Positive pattern #4: Remove Roadblocks

Help your team to get moving by removing technical or organizational roadblocks. For example, help the team get the server resources they need. If an engineer is having trouble with an arcane bit of Java code, help them connect with another engineer who can help them. Knowing the right people is more valuable than knowing the right answer.

Positive pattern #5: Be a Teacher and Mentor

It can be very difficult to watch a team member spend three hours working on something that you know you could finish in 20 minutes. Giving people a change to learn on their own is a vital component of effective leadership.

To be a good mentor, you only need three things: experience with your team's processes and systems, the ability to explain things to someone else, and the ability to gauge how much help your mentee needs.

Positive pattern #6: Set Clear Goals

To make your team move rapidly in one direction, you need to make sure that every team member understands and agrees on what the direction is. You need to set clear priorities and help your team decide how it should make trade-offs when the time comes.

Create a concise mission statement for the team. After the direction and goals are clearly set, give the team autonomy, and periodically check in to make sure everyone is on the right track.

Teams can and do succeed without clear goals, but they typically waste a great deal of energy.

Positive pattern #7: Be Honest

Honestly say when you don't know the answer or you're not able to say it. Tell your new team members, "I won't lie to you, but I will tell you when I can't tell you something or if I just don't know."

Giving redirecting feedback is hard. Do not use the "feedback sandwich", because people can easily miss the critical message. Instead, employ respect. Kindness and empathy are critical if you want the recipient to hear the criticism and not go on the defensive.

As an example, if someone is bluntly criticizing every decision made in the team, do not tell them to "stop being a jerk", but honestly tell them the effect their behavior is having on the team.

Positive pattern #8: Track Happiness

Take time to gauge your team's happiness. Make certain your team is happy.

For example, keep a spreadsheet of all the grungy, thankless tasks that need to be done and make sure they are evenly spread across the team. In one-on-ones, do not only discuss technical issues but ask how they're enjoying the work and what they're looking forward to next. At the end of one-on-ones, ask "What do you need?" Carefully gauge also their happiness *outside* of the office. Be sensitive to personal situations and give extra slack to team members going through a tough time.

Most people have goals for the next five years: being promoted, learning something new, launching something important, or working with smart people. Whatever it is, help your team work towards their goals and show you're paying attention to them.