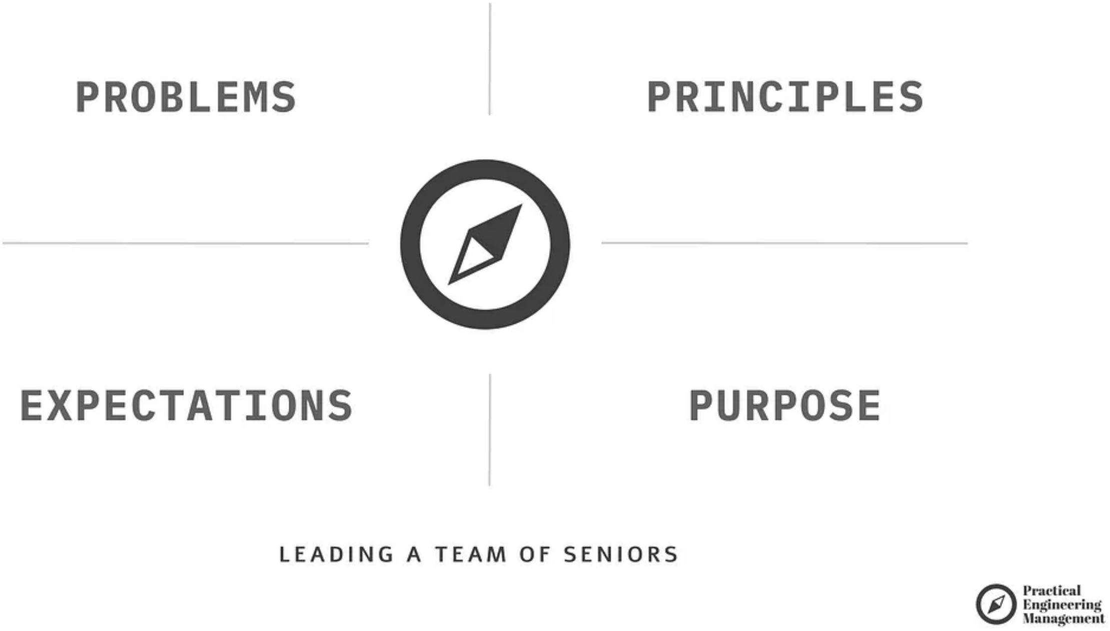


How to lead a team of senior engineers

Managing people who are more skillful than you

Mirek Stanek · Follow

Published in Practical Engineering Management

8 min read · Jan 17, 2024

Listen Share More

One of the responsibilities of engineering leader is to support their teammates with self-development. Growing a high-performing team with knowledgeable and experienced engineers should be the ultimate goal for a manager. But It also comes with the fear (I've got this question multiple times during my 1:1s with engineering leaders):

"What if those people were smarter than me? How am I supposed to manage them?"

In this post, I'm sharing some techniques and materials from Practical Engineering Management that will help you lead a team of senior folks with strong technical skills and rich professional experience, and you cannot constrain them with a command and control management approach.

. . .

According to Wikipedia, here is the etymology of the word "management":

> The English verb manage has its roots in the fifteenth-century French verb mesnager, which often referred in equestrian language "to hold in hand the reins of a horse". Also the Italian term maneggiare (to handle, especially tools or a horse) is possible. In Spanish, manejar can also mean to rule the horses. These three terms derive from the two Latin words manus (hand) and agere (to act).

While "*holding in hand the reins of a horse*" sounds extreme, it somehow reflects the default behaviors of fresh engineering managers.

The administrative part of management is indeed needed (like managing expectations, timelines, resources, costs, etc.). However, there are more effective ways of working with skilled people than telling them what and how to do.

As an engineering leader, your job is less about managing people and more about managing their talent, potential, and growth. Here's how you can do this:

- Manage through expectations
- Manage through problems, not solutions
- Manage through purpose
- Manage through principles

Manage through expectations

Telling people what to do and how to do it has many flaws. By micromanaging, you don't use the full potential of your direct reports. You are also a bottleneck since everyone is dependent on your explicit instructions.

Rather than giving orders, change the narrative to having some specific expectations:

- Expect the team to solve a given problem.
- Expect an engineer to own their work (e.g., monitor the health of their services or keep them up to date).
- Expect particular requirements to be met (e.g., quality KPIs, non-functional requirements, etc.).

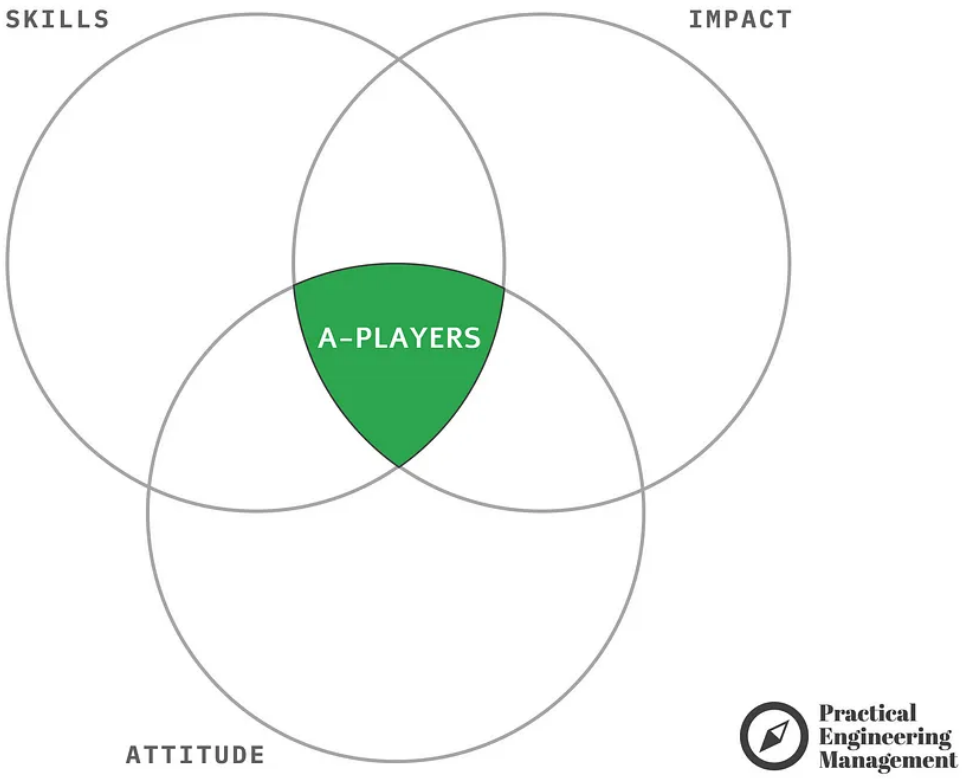
By going this way, you show the team what the expected results are while it's on them to figure out how to do it properly.

. . .

To extend your expectations management practices, go beyond just goals assessment.

Here are three fundamental aspects you should take into consideration when discussing performance with your teammates:

- Skills
- Impact
- Attitude



Great software engineers should be a mix of all of these. It's not just about being a good developer or software architect (**Hard Skills**). It's neither about being a great public speaker or a decent negotiator (**Soft Skills**).

Hard and soft skills are critical but only matter when appropriately allocated (**Impact**). And your teammates' willingness, openness, and proactiveness (**Attitude**) are the ways to maximize this impact.

To read more about expectations management, check this article: Intro to Expectations Management

Intro to Expectations Management

Shifting from Effort to Impact when Leading Engineering Teams

www.practicalengineering.management

Manage through problems to solve, not solutions

One of the common pitfalls in software engineering is that we jump into solutions too quickly before we even define a problem we're trying to solve. If you, as a manager, give your strongest people solutions to implement, not problems to solve, you deprive them of the opportunity to find the right solution.

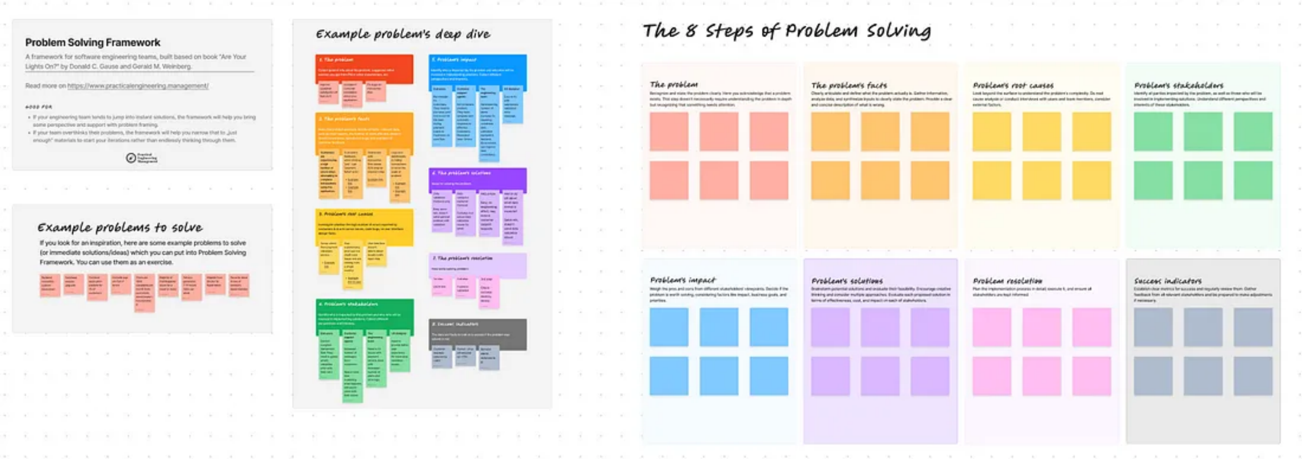
In the complex world we live in, the initial solution coming to your head is rarely the optimal one. Complex problems may have multiple definitions, none of which is ultimately the right one. A problem may have multiple stakeholders, each with their own solution, and these stakeholders may have different power to articulate their preferences (the loudest one in the room, the most senior on the org chart, or those who prefer to remain silent).

That's why, before diving into any problem, task, or challenge, try answering these questions first:

1. What is the problem?
2. What are the facts underlying the problem?
3. What are the root causes of the problem?
4. Who is affected by the problem?
5. Is solving the problem worthwhile? Is it in alignment with broader goals and strategy?
6. What are the options for solving the problem? Which are the optimal ones?
7. Implement the solution.
8. Are success metrics defined and met through the solution?

Including your senior engineers in answering these will bring you needed perspective and help with framing the problem.

These 8 Steps of Problem Solving come from the framework I built based on my experience as well as the great story — "Are Your Lights On" by Don Gause and Gerald Weinberg.



Problem Solving Framework — example template in FigJam

You can read more about it here: The Problem Solving Framework

The Problem Solving Framework

Define your problem before jumping into immediate solutions

www.practicalengineering.management

lving Framework

ftware engineering teams, built based on book
ald C. Gause and Gerald M. Weinberg.

s://www.practicalengineering.management/

eam tends to jump into instant solutions, the framework w
nd support with problem framing.
ks their problems, the framework will help you narrow tha
start your iterations rather than endlessly thinking throug

Manage through purpose

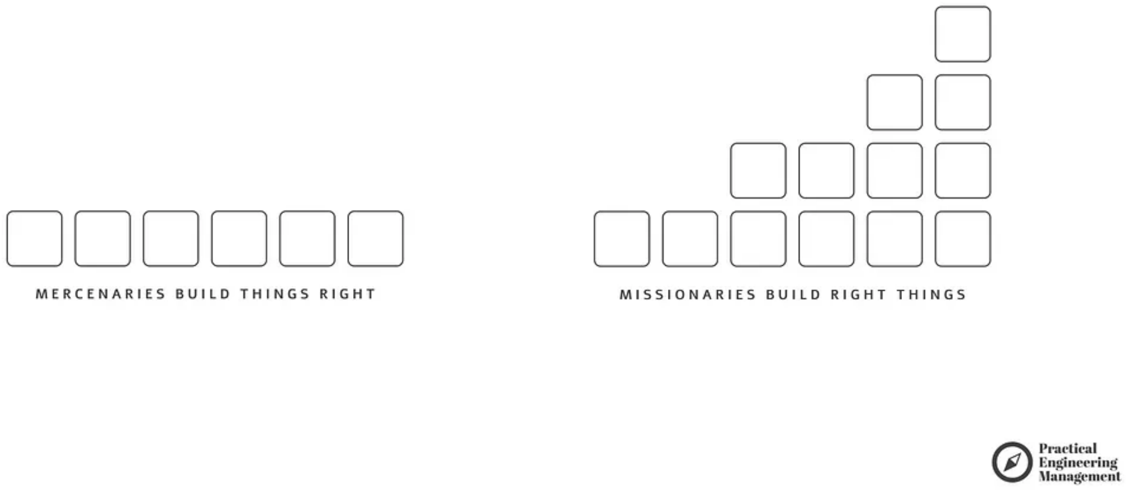
In materials on Practical Engineering Management, I often share this statement:

The role of technology in a product company is to solve customers' problems.

This is especially critical for skilled engineers, who, thanks to their expertise, can have a huge impact on the product. Your job is to ensure that they don't just build features for stakeholders but are an integral part of the product organization.

It means your strongest engineers should take part in discovery and delivery processes as well as own their work post-launch. To achieve that, assess their work not by tasks accomplished but by achieved outcomes.

I covered these with some details in the article: Build a team of missionaries, not mercenaries.



MERCENARIES BUILD THINGS RIGHT
MISSIONARIES BUILD RIGHT THINGS
Practical Engineering Management

Build a team of missionaries, not mercenaries

Driving a team with outcomes requires some level of the organization's maturity. If you think your company is not yet ready for that, you can try to define the purpose of your team through the factors that are universal for most product companies:

- growth
- expansion
- customer satisfaction
- costs

You can read more about it here: Four Factors Essential to the Success of Any Product

Manage through principles

A company's mission and vision are often high-level and not easily translated into tangible actions and decisions. To keep the organization aligned, a company can promote a principled leadership culture. The concept of principles serves as a framework for decision-making and pursuing the company's strategy.

Principles provide alignment and autonomy for your team. Even with strong software engineers, there are countless ways to solve a problem. Principles guide decision-making without constant supervision. For instance, if considering an additional security layer for an API, Apple's principle of "Privacy is a fundamental human right" would make the decision a no-brainer. In contrast, Facebook's "Move fast" might lead to a more nuanced discussion.

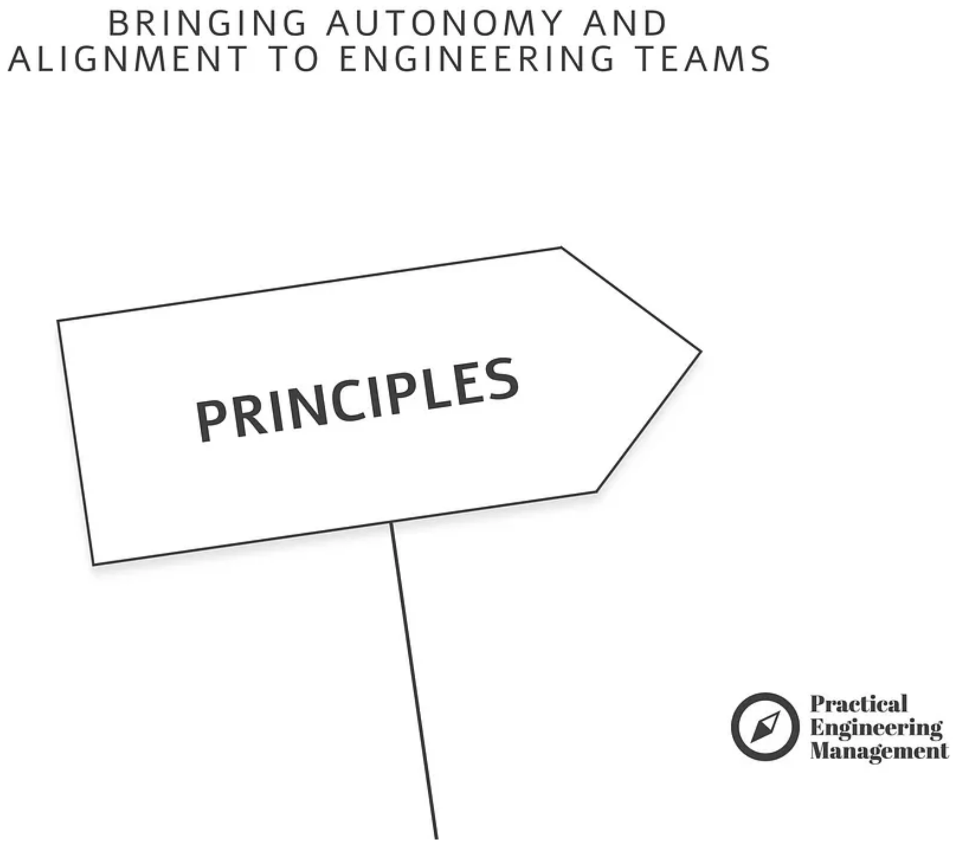
While your team should be responsible for the "HOW" of a given challenge, your principles should provide them with a context on what matters the most in their work.

Here are some example principles from the industry:

- **Google's "Focus on the User and All Else Will Follow":** Emphasizing user-centric design, this principle inspires software engineers to prioritize user experience, ensuring intuitive and accessible products.
- **Amazon's "Working Backwards":** This principle encourages engineers to think from the end-user's perspective and work backward to ensure the product meets its intended goals.
- **Netflix's "Highly Aligned, Loosely Coupled":** A principle of having teams that are aligned in their goals but loosely coupled in their execution can inspire software engineers to work autonomously while still contributing to the broader objectives of the organization.

If your organization hasn't developed clear principles, don't let that stop you from creating them for your team. In a small company, this is an opportunity to lead by example and influence the entire organization.

Practical Engineering Management explores the concept of principles here: Principles — Guidelines for Your Team.



Other materials

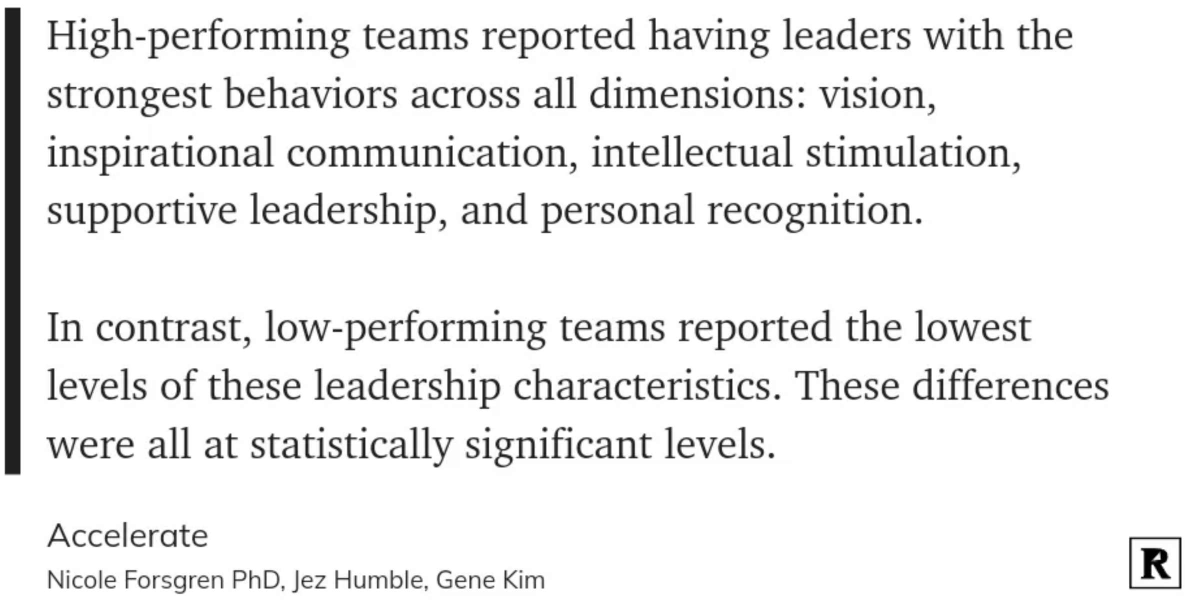
No matter how skilled the engineers you are managing, there are some practices that should be universal for leaders.

5 Characteristics of a High-performing Team Leader

The book "Accelerate" highlights five key characteristics that define a high-performing team leader:

- Visionary,
- inspirational communication,
- intellectual stimulation,
- supportive leadership,
- personal recognition.

Explore these traits and see how they can elevate your leadership skills. Read more here.



> High-performing teams reported having leaders with the strongest behaviors across all dimensions: vision, inspirational communication, intellectual stimulation, supportive leadership, and personal recognition.
>
> In contrast, low-performing teams reported the lowest levels of these leadership characteristics. These differences were all at statistically significant levels.

Accelerate

Nicole Forsgren PhD, Jez Humble, Gene Kim

1:1s

It was mentioned countless times on Practical Engineering Management — if you can pick a single leadership technique to work with your team, let it be 1:1s.

1:1 meetings are where:

- you make these abstract concepts tangible.
- you support your team in reaching their potential and succeeding professionally.
- you translate strategic objectives into daily actions.
- you foster the company's culture.

Starting your first 1:1 meetings can be challenging. You must bring structure and continuity to these sessions, as well as ensure they are valuable for your team members. On Practical Engineering Management, you'll find materials on running these meetings, their core principles, and focus areas.

Explore these resources here:

- 1:1 — The Essential Tool for Leaders

- (Premium Content) First Ten 1:1s for Engineering Leaders

First Ten 1:1s for Engineering Leaders

Every successful leader says that 1:1 is one of the essential things in their job. But for fresh managers, these…

www.practicalengineering.management

And more

I covered some tools and techniques to guide engineering leaders through the challenges of their roles in the article Engineering Leadership: Tools and Techniques for a Successful 2024. I hope it'll help you succeed in your role and take much satisfaction out of it!

Engineering Leadership: Tools and Techniques for a Successful 2024

Practical Engineering Management materials for you

blog.practicalengineering.management

End Notes

This compilation is designed to help you lead senior software engineers, who are supposed to be more skillful than you in some areas. My mission is to help engineering leaders make great ideas happen.

Explore more content on practicalengineering.management. You will find there practical strategies for effective engineering leadership. Join the community of impactful leaders to bridge the gap between inspiration and implementation with actionable steps that empower your team, boost trust, and drive real-world results.

If you would like to discuss any of your challenges, don't hesitate to reach out to me at mirek@practicalengineering.management.

Leading Engineers Engineering Management Leadership Startup
Management

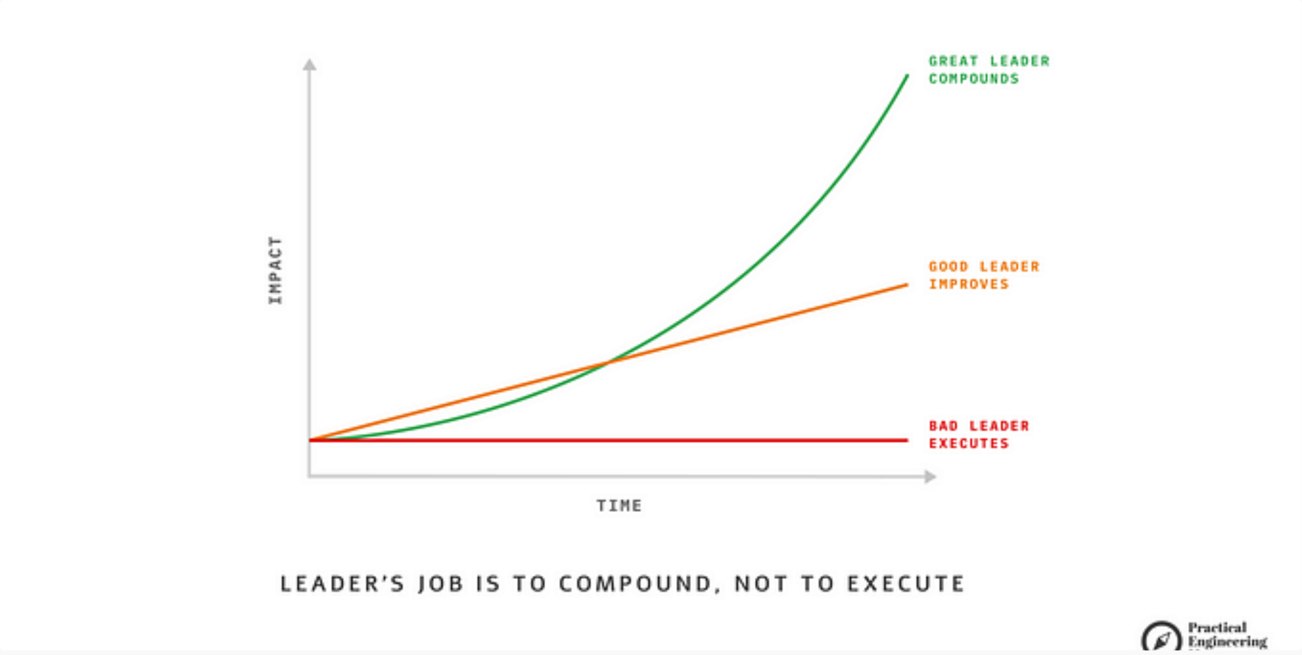
Follow

Written by Mirek Stanek

6.3K Followers · Editor for Practical Engineering Management

I empower leaders through practice 🛠💡✨. Site Leader and Director of Engineering at Papaya Global. Ex-Head of Engineering at Azimo.

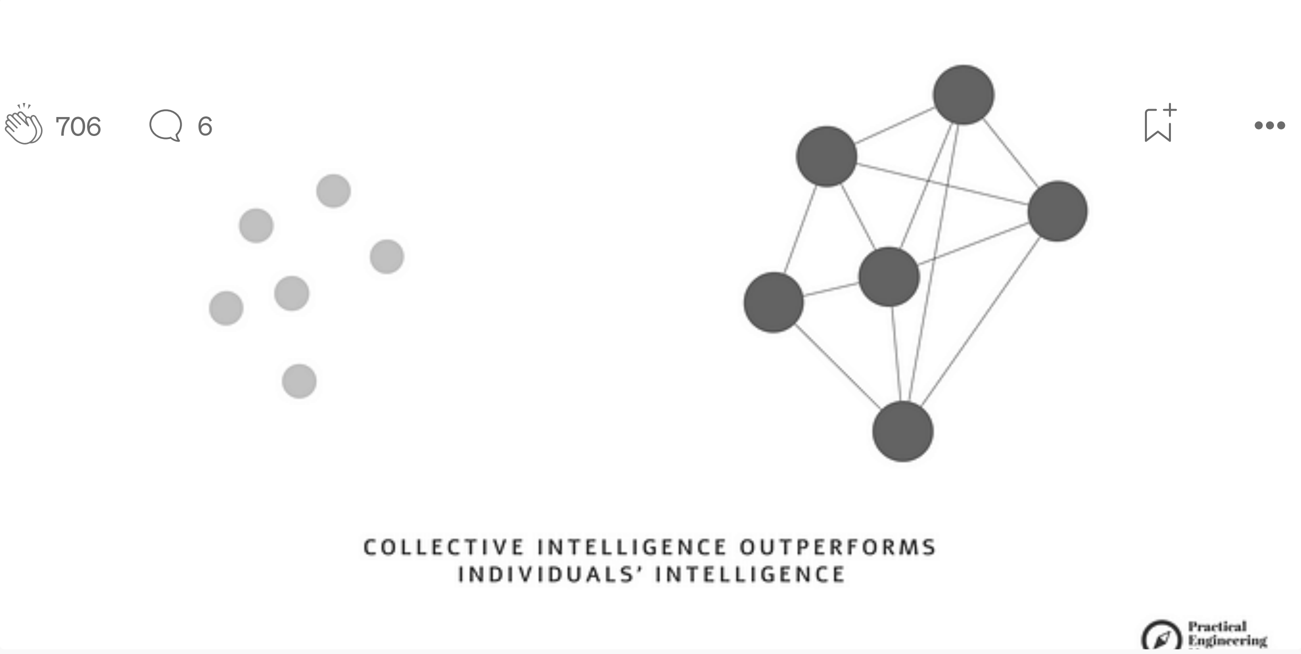
More from Mirek Stanek and Practical Engineering Management



Mirek Stanek in Practical Engineering Management

How to Be a Better Software Engineering Leader

Strategies for Elevating Your Engineering Leadership


COLLECTIVE INTELLIGENCE OUTPERFORMS INDIVIDUALS' INTELLIGENCE
Practical Engineering

706 6

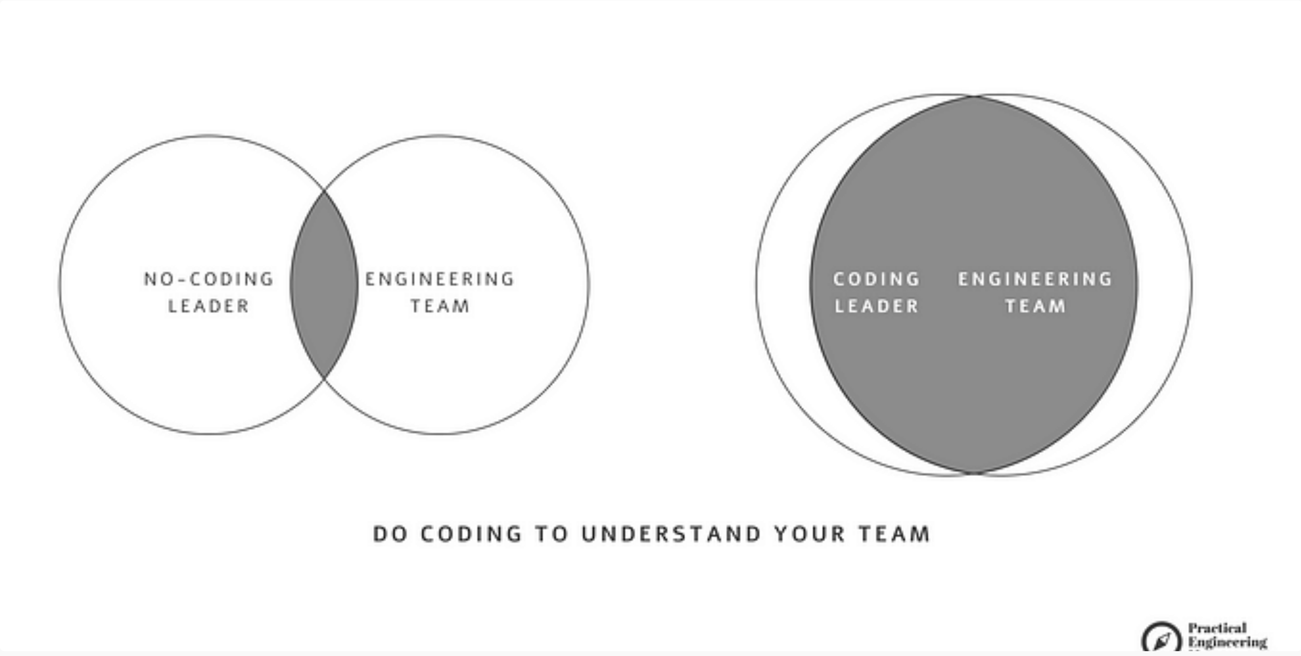
Mirek Stanek in Practical Engineering Management

How to Support Collective Intelligence and Decision-Making

Example Templates and Tools for Software Engineering Leaders

8 min read · Mar 26, 2024

308 1



Mirek Stanek in Practical Engineering Management

Go back to coding, engineering leader

Software Engineering Management in 2023

8 min read · Feb 23, 2023

1.3K 16

Mirek Stanek in Practical Engineering Management

Engineering Leadership: Tools and Techniques for a Successful 2024

Practical Engineering Management materials for you

6 min read · Dec 11, 2023

705 4

See all from Mirek Stanek

See all from Practical Engineering Management

Recommended from Medium

Somnath Singh in Level Up Coding

The Era of High-Paying Tech Jobs is Over

The Death of Tech Jobs.

✦ · 14 min read · Apr 1, 2024

8.7K 251

RESEARCH
TECH EXCELLENCE
METRICS
1:1S
OUTCOME FOCUS
HOW TO BE A SUCCESSFUL LEADER IN THE COMING YEAR
Practical Engineering

Mirek Stanek in Practical Engineering Management

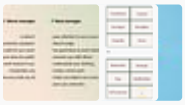
Engineering Leadership: Tools and Techniques for a Successful 2024

Practical Engineering Management materials for you

6 min read · Dec 11, 2023

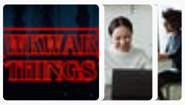
705 4

Lists



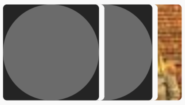
How to Lead Well as a New Manager

14 stories · 546 saves



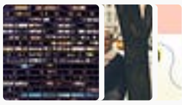
How to Run More Meaningful 1:1 Meetings

11 stories · 377 saves



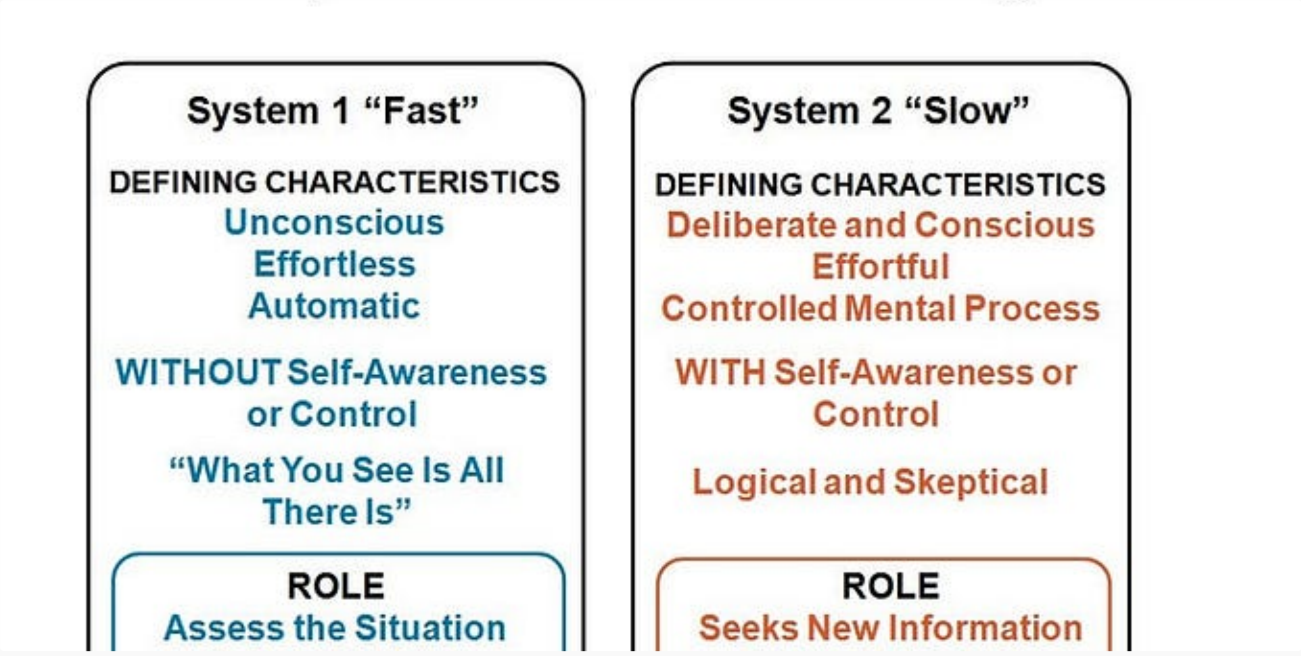
How to Give Difficult Feedback

7 stories · 570 saves



Intro to People Ops: Not Your Mama's HR

8 stories · 74 saves



System 1 "Fast"

DEFINING CHARACTERISTICS

Unconscious
Effortless
Automatic

WITHOUT Self-Awareness or Control

"What You See Is All There Is"

ROLE

Assess the Situation

System 2 "Slow"

DEFINING CHARACTERISTICS

Deliberate and Conscious
Effortful
Controlled Mental Process

WITH Self-Awareness or Control

Logical and Skeptical

ROLE

Seeks New Information

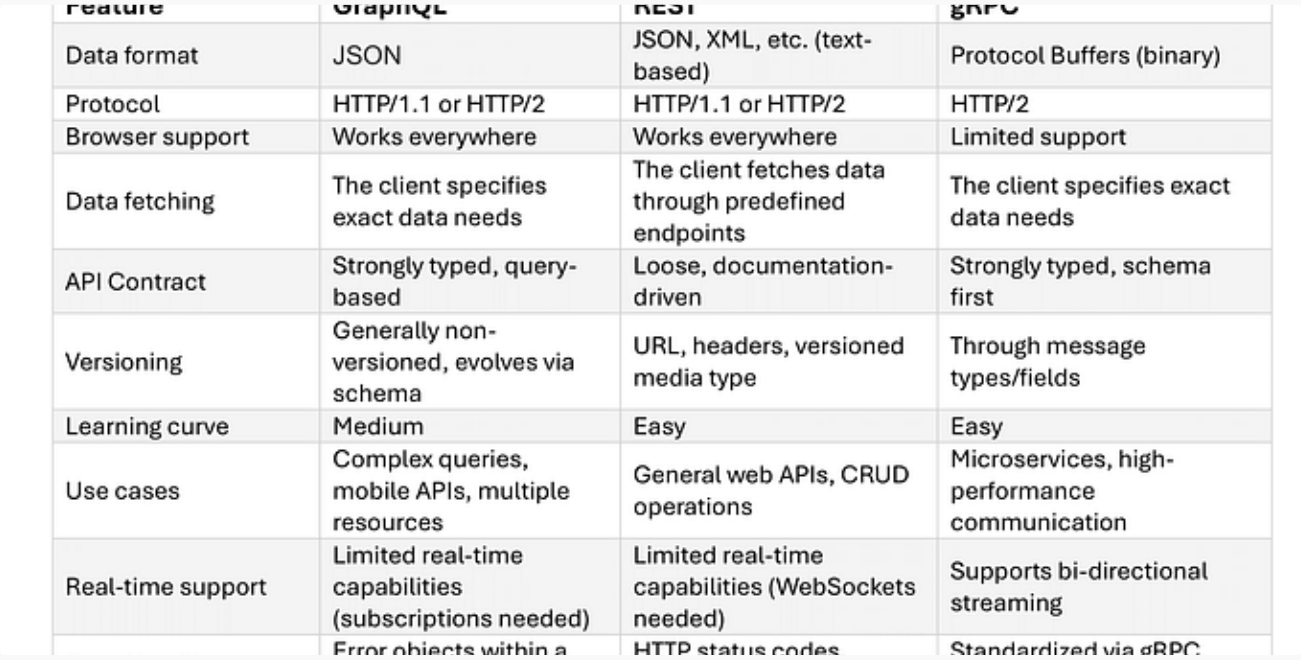
Devansh in Startup Stash

What I Learned From Thinking Fast And Slow

Quite possibly the single most important book ever written for data scientists, decision makers, and everyone else.

16 min read · Apr 1, 2024

3.6K 56



Feature	GraphQL	REST	gRPC
Data format	JSON	JSON, XML, etc. (text-based)	Protocol Buffers (binary)
Protocol	HTTP/1.1 or HTTP/2	HTTP/1.1 or HTTP/2	HTTP/2
Browser support	Works everywhere	Works everywhere	Limited support
Data fetching	The client specifies exact data needs	The client fetches data through predefined endpoints	The client specifies exact data needs
API Contract	Strongly typed, query-based	Loose, documentation-driven	Strongly typed, schema first
Versioning	Generally non-versioned, evolves via schema	URL, headers, versioned media type	Through message types/fields
Learning curve	Medium	Easy	Easy
Use cases	Complex queries, mobile APIs, multiple resources	General web APIs, CRUD operations	Microservices, high-performance communication
Real-time support	Limited real-time capabilities (subscriptions needed)	Limited real-time capabilities (WebSockets needed)	Supports bi-directional streaming
	Error objects within a	HTTP status codes	Standardized via gRPC

Dr Milan Milanović

When to use GraphQL, gRPC, and REST?

Building APIs is one of the most important tasks for developers in modern engineering. These APIs allow different systems to communicate...

11 min read · Apr 8, 2024

620 4

Are You Smart Enough To Work at Apple?

There are three boxes: one with only apples, one with only oranges, and one with both. Each box is wrongly labeled. If you pick one fruit from a box without seeing inside, how can you then correctly label all boxes?

Bella Lockhart in ILLUMINATION

Can You Pass This Apple-Orange Interview At Apple 🍎?

The iPhone Company’s Interview Question

✦ · 3 min read · Mar 14, 2024

2.2K 64

Jake Page

The guide to Git I never had.

🩺 Doctors have stethoscopes.

13 min read · Apr 11, 2024

2.8K 25

See more recommendations