SOFTWARE ENGINEERING RADIO

Editor: **Robert Blumen**
SalesForce
robert@robertblumen.com

Johnathan Nightingale on Scaling Engineering Management

Travis Kimmel

From the Editor

In Episode 352 of "Software Engineering Radio," guest Johnathan Nightingale talks with host Travis Kimmel about scaling engineering management. Portions not included here go into more depth about the organizational structure of engineering teams, seniority in management, career growth, management titles, predictability in management, and goals. To hear the full interview, visit http://www.se-radio.net or access our archives via RSS at http://feeds.feedburner.com/se-radio.—*Robert Blumen*

Travis Kimmel: What is the role of head of engineering at a small but rapidly growing company of, say, 20 people?

Johnathan Nightingale: In a classic tech start-up of 20 people, engineering is about half. The head of engineering is still a practitioner. You don't need full-time management because too much development has to happen. Even if they're not writing code, the head is choosing AWS services and building toolchains. A lot of their work is driving engineering progress: putting the right practices in place, iterating quickly, building engineering velocity, and establishing a tight understanding with product or the CEO [chief executive officer].

What does a day in the life look like?

What it is often is running very fast. There are fires to fight and new problems to discover. You're writing code, doing code reviews, and also interviewing and making decisions about things like text editors, data warehouses, and instrumentation. By the time you have a 10-person team, you should be thinking about your role as a manager: "How do we establish clear norms and expectations and help people build something sustainable?"

When do you start hiring another layer of management?

Eight to 10 people, regardless of function, is often where most people max out in their ability to be thoughtful, conscientious managers. When you have 10 or more people reporting to you, your top priority is to get that number down. So much will be going so fast that you can't be adaptable and responsive to 16 people—it's too much. You need to find people you can trust with day-to-day management.

In an eight-person engineering team without effective management, you're getting about four engineers' worth of work. You don't bring in a manager just to feel good; you do it because that person can get your eight engineers to provide at least seven engineers' worth of productivity. And if you let them go for a while and they're

Digital Object Identifier 10.1109/MS.2019.2899530
Date of publication: 16 April 2019

good at it, you can get eight, nine, 12 engineers' worth of productivity as that investment starts to pay off.

Does engineering experience mismanagement with greater frequency than other industries?

Underequipped management is not something that tech invented. In the business we run, Raw Signal Group, we train managers, and a lot of the groups we work with are start-ups because we come from tech and it's our wheelhouse. Putting people on their payroll whose only job is to tell other people what to do feels to them like [a] waste. And by the time we get to them, they've often grown to 100 people plus, and nobody knows how to manage well.

I mean basic stuff, like what is a one-on-one for, how do I know if it's working well, and how do I give people feedback? At the end of a workshop or multiday training with them, their eyes are open. They want to learn more about management because suddenly it's a discipline that has a system to it instead of just something you pay people to do when they stop knowing how to engineer.

What team sizes and organizational structures work best for engineering, and what's the standard way to design an engineering team?

All the founders we have worked with have said something like, "I just want to keep it flat. Don't want a lot of structure. Ideas should win over titles." It's healthy to not want people in positions of authority who are attracted purely to the authority they have over other people. But when flat stops working because nobody knows who to go to for anything, what they own, or where accountability lines are, they often start inventing their own systems. And a lot of those are a mess.

They create weird, multidimensional matrix structures where you've got a functional manager and a program manager and a geographical manager. They do feature teams with management lines around four engineers and a product manager and half a designer. And people don't know who their boss is.

> Eight to 10 people, regardless of function, is often where most people max out in their ability to be thoughtful, conscientious managers.

People in a growing organization should not innovate on this. When it comes to org structure, use the stuff that we've been using for a long time, like sensible load factors for management, clear functional reporting chains, and well-trained management that can be held accountable for the work of their teams.

Engineers know that tree structures grow efficiently. If you like having engineers who still work, when those people express an interest in management, put them into a lead structure, managing four engineers. Those leads report in to a manager, who will have at most six to eight direct reports. Those managers then report to a director of engineering, with no more than four to six direct reports for each director. And then maybe there is one VP who collects all the directors. This tree structure can carry 250 people without any of the leaders feeling overtaxed, with everyone knowing who their boss is, and with none of them being too many hops away from the CEO. And it allows for a surprising amount of scalability.

What specific skills or background experience should you look for in management candidates?

You can look at a résumé that shows experience managing, citing things, such as low retention and high-performing teams. But you still need to probe with questions, such as, "How do you think about one-on-ones? Walk me through how you handle an underperforming employee," and so on. Such questions flush out cultural assumptions.

Everybody in an organization sets culture. Although the CEO has to be brought in on major cultural initiatives, culture is set much more at the line-manager level. Most people's experiences of working in your company [are] dictated by their manager. The only way to screen for that is with an in-depth conversation.

You need to do more than just hire someone who is good with people, charming, of high emotional intelligence. I've met lots of charming sociopaths. You can teach the core skills of how to manage well: running meetings, figuring out ownership in a cross-team project, hiring, distributing seniority, and getting the right mix of junior and senior engineers.

Character is part of it, but I also want to see that you have skills.

What does seniority look like in management?

Managing individuals is different from managing managers. When you manage individuals, you hear about what's going on in their lives. If you're a VP of a 200-person organization, although some stuff gets filtered, more of that bubbles up—like, "Alex's mom is sick, and Alex will be off for a couple weeks." So one element of the difference is that the higher you are in a tree structure, the more of that reaches you.

We also expect a shift in operator context, a gradual blending in of strategic perspective, aligning teams so that they work on the right things. Although engineering is a crucial part of a VP of engineering's work, their principal obligation is to the other members of the senior leadership team: "What are we doing as a business? What are we committed to? What are the strategic choices that we're making?"

You shift from being team oriented at the more junior-manager level to being organizationally oriented as you get more senior. Everyone should be laddering their work up into the organizational goals, but practically, that shift tips at around the director level, where we expect more strategic ownership and less day-to-day operational management.

How do you do onboarding right for managers?

A manager has one job: to make their team more effective. A manager needs to know what effective looks like. So number one in onboarding is, "Here's what your team's here to do." That conversation could take 20 min, but it often doesn't happen. Second is, "Here are our norms, and here's how we make choices." And third is, "Here's how we do work. Here are our processes." The first two tend to get ignored a lot.

> If you like having engineers who still work, when those people express an interest in management, put them into a lead structure, managing four engineers.

How do you plan for the career growth of managers?

A junior engineer can be moved out of the junior category quickly. There is a big difference in productivity between someone fresh out of school and someone who has worked for a year in software development. In management roles, career growth will not go as quickly. Tree structures scale well because you can have a small number of people at the top to handle a lot of people at the bottom. So there are not many opportunities for managers who want to be directors. It's important to set honest expectations about the career path and timeline.

Dealing with employee personal problems, mental health issues, or toxic work situations require[s] a depth of expertise that takes time and experience to develop from unconscious incompetence to unconscious competence. Career progress is also dependent on growth of the organization and internal changes that create an opening. Career development can focus on broadening context and deepening insight to prepare managers for promotion should such openings occur.

When I worked at IBM, they had 10 bands, and engineers started at band six, which was entry level. Band seven required more sophisticated work and mentoring of some junior band sixes. Band eight assumed that you could be counted on to be a reliable individual contributor and coordinate the work of small teams. Beyond band eight, it was possible to change assignments and get increases in compensation, but to be promoted to a higher band, the engineer had to take initiative and solve a difficult, persistent problem.

For managers in tech organizations other than IBM, this is typically the path to promotion: people see that you are taking the business seriously and want to fix an important piece of it, and you are given resources to do so. "This person is doing great stuff. This thing is working. Let's pour gasoline on it."

What should you do when you give someone a promotion?

You should articulate expectations. People who were just promoted are receptive to a management conversation because at that moment, their life options have changed. Many managers will just say, "Congratulations." But they should keep talking and explain how the new job is different and what the new expectations are.

Do titles matter?

There's a reason people use them. They're internal and external signifiers. A lot of start-ups will use standard titles, such as director and VP, only externally. Standard titles are useful for calibration. Large companies, like Facebook and Google, develop their own title nomenclature, but when a 200-person start-up uses nonstandard titles, it requires more work in the job description to calibrate the degree of seniority and autonomy in a role. And why add additional stressors by taking out certainty where it's easy to add it back in?

I've seen promotions where the person's job description doesn't change. What's your view of those kinds of title changes?

There should be a difference that is clear internally. Contributors know if there's a difference or not. Title is a useful signifier only if it signifies something other than tenure. Meaningless title changes waste an opportunity to distinguish between different levels of responsibility.

Title changes can be motivated by the need to change compensation, leading to title inflation. This is just a failed compensation strategy. Managers should be able to recognize hard work and growth without title changes. There's a difference between saying, "Great work as a senior engineer. Here's a 10% raise," and, "You're promoted to principal engineer, and here's a 10% (or more) raise." Those should have different meanings.

What is the difference between a manager, director, and VP in a standard setup?

SOFTWARE ENGINEERING RADIO

Visit www.se-radio.net to listen to these and other insightful hour-long podcasts.

RECENT EPISODES

- 357—Felienne speaks with Adam Barr, author of *Why Smart Engineers Write Bad Code*, about code quality.
- 356—Tim Coulter, the founder of Truffle (Ethereum DApp development framework), discusses the Truffle framework for Ethereum Smart-Contracts and Decentralized App development with Kishore Bhatia.
- 355—Host Travis Kimmel speaks with WeWork's VP Engineering at WeWork, Randy Shoup, and discusses how to scale technology and organization together.

UPCOMING EPISODES

- Edanea Salinas talks to Pete Koomen about A/B testing.
- Jean-Denis Greze and host Nate Black discuss engineering maturity.
- Daniel Berg joins host Nishant Suneja for a conversation about Istio and service mesh technology.

A manager's job is to make a team effective at accomplishing part of their director's obligations. The manager is a fundamental building block of organizations with more than eight people. The job collects people around a similar function. The manager does one-on-ones, mentorship, work assignment, and tradeoffs with other teams to situate work within a broader organization.

Orchestration happens at the management level. Directors inform strategy. A director is the basic level of deconstruction of a business; a director owns a single function. The VP gathers everyone in a function as a member of the senior team who is charged with an entire division of the organization when divided by function. They are fully responsible and accountable for the work of their divisions.

> Tree structures scale well because you can have a small number of people at the top to handle a lot of people at the bottom.

Where does an architect fit?

How you provide career growth for senior engineers who do not want management responsibility is a cultural

ABOUT THE AUTHOR

TRAVIS KIMMEL is the chief executive officer and cofounder of GitPrime, the leading provider of data-driven metrics for software engineering organizations. As a Y Combinator alum and former engineering leader, he is focused on helping software teams use data to become more effective and bridge communication between engineering and the rest of the enterprise. Contact him at t.kimmel@gitprime.com.

decision: How much seniority do we accord someone who doesn't manage? In some organizations, a chief architect is a C-level role that is speaking with the board and in the weekly steering committee meetings. The rationale is that engineering is so core to our function that the person charged with the architecture of the entire system needs to be part of all the strategic and planning meetings for the business.

I've also seen chief architect used as a title that sounds better than principal for people who outgrew their titles. But there shouldn't be more than one chief architect. Expectations should be spelled out crisply.

How do you manage a senior employee up, and when do you decide to manage them out?

Managers, directors, and VPs are rarely put on a performance-improvement plan (PIP). If a manager is underperforming, probably no one has told them so clearly. Because so many managers fear confrontation and don't have well-developed feedback skills, the PIP often lands as a surprise. The value of a PIP is that somebody wrote it down. A manager's job is to make his or her team more effective, and effectiveness is visible. Everyone in an organization knows which teams are effective and which are not.

Junior staff [members] are measured on effort: "I gave you that work. You did that work." But, at a VP level, if outcomes aren't what was promised or expected, I don't care how hard the team worked. Where was the communication in Q1 that we were lagging? Where was the communication in Q2? Where were the contingency plans? How did you help the senior team plan and adapt? If that stuff's not there, we shouldn't manage you up. At the VP level, we expect you to be self-managing.

IEEE Software (ISSN 0740-7459) is published bimonthly by the IEEE Computer Society. IEEE headquarters: Three Park Ave., 17th Floor, New York, NY 10016-5997. IEEE Computer Society Publications Office: 10662 Los Vaqueros Cir., Los Alamitos, CA 90720; +1 714 821 8380; fax +1 714 821 4010. IEEE Computer Society headquarters: 2001 L St., Ste. 700, Washington, DC 20036. Subscribe to *IEEE Software* by visiting www.computer.org/software.

Postmaster: Send undelivered copies and address changes to *IEEE Software*, Membership Processing Dept., IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854-4141. Periodicals Postage Paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail Agreement Number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8, Canada. Printed in the USA.