

# Table of Contents

<a href="#">Introduction</a>	1.1
------------------------------	-----

## Engineering Bookshelf

<a href="#">General Management</a>	2.1
<a href="#">Agile, Scrum, XP, and Lean</a>	2.2
<a href="#">Software Management</a>	2.3
<a href="#">Architecture</a>	2.4
<a href="#">Design</a>	2.5
<a href="#">Microservices</a>	2.6
<a href="#">Operations</a>	2.7
<a href="#">Security</a>	2.8

## Engineering Management

<a href="#">The Manager Track</a>	3.1
<a href="#">Team Lead &amp; Developer Resources</a>	3.2
<a href="#">Engineering Hiring</a>	3.3

## Topics in Software

<a href="#">Architecture</a>	4.1
<a href="#">Serverless</a>	4.2
<a href="#">Design (UI and otherwise)</a>	4.3
<a href="#">Microservices</a>	4.4
<a href="#">SDLC (Agile and otherwise)</a>	4.5
<a href="#">Security</a>	4.6

# Introduction

# **General Management**

# **Agile, Scrum, XP, and Lean**

# Software Management

# Architecture

# Design

# Microservices



# Operations

# Security

# The Manager Track

A few links for anyone thinking of hopping to the management track:

## Start Here

- [The Manager's Path](#) (book) - This is the number one best reference I know for technical individual contributors moving to management

## Books

- [The Making of a Manager: What to do When Everyone Looks to You](#) - From Julie Zhou, who went from individual contributor to Facebook VP.

## Getting Started

- [Making the Most of Your First Three Months as a New Manager](#) - Excerpt from **The Making of a Manager**.
- [Advice for first-time managers](#)
- [A Reading List for New Engineering Managers](#)
- [How to fail as a new engineering manager](#)
- [Engineering Management: Lessons learned in first year](#)
- [Becoming a bad-ass engineering leader: 5 tried and true lessons from a woman of color](#)
- [The Seven Areas Of Software Management](#)
- [Everybody's Crazy: Why Management is Hard \(And What to Do About It\)](#)
- [Flavors of Engineering Management](#)
- [How to Lead](#)
- [What is an Engineering Manager?](#)

## Career Paths

- [Engineering Management: The Pendulum Or The Ladder](#) - Not sure whether you want to head for management? Read this first.
- [Engineering Growth Framework](#) - Career paths from Medium
- [Progressing from tech to leadership](#) - things to think about as you move into a leadership role
- [Switching from Engineering to Management](#)
- [I Didn't Want to Be a Manager Anymore—and the World Didn't End](#) - Going back to a technical role
- [On being an Engineering Manager](#) - Another tale of trying management and pulling back
- [Software Roles and Titles](#)
- [Career Growth Frameworks in Software Engineering: A Review](#)

## Coaching, Mentoring, and O3s

- [Coaching](#)
  - [Keeping your 1 to 1s fresh](#) - How to not just turn them into status updates
  - [Ask "How's It Going?"](#)
  - [Engineering Management Distilled: A guide to one-on-ones](#)
  - [How to Deliver Constructive Feedback in Difficult Situations](#)
  - [The Feedback Fallacy](#)
  - [How great managers give and receive feedback](#)
  - [10 Mistakes You Should Avoid During Your One on One Meetings](#)
  - [The one-on-one meeting template for your end of the year review](#)
  - [Mentored by Wizards, Part 1 – “You’re Worthwhile”](#)
  - [Coaching tool: The ORG team building model](#)
  - [Mentoring Developers: Best Practices From Uber - Interview With Gergely Orosz](#)
  - [One-on-ones are my most valuable meetings; here’s how I run them](#)
  - [\(\[https://ajahne.github.io/blog/leadership/2019/07/24/essential-meetings-to-have-with-your-people-as-a-manager.html\]\)](([https://ajahne.github.io/blog/leadership/2019/07/24/essential-meetings-to-have-with-your-people-as-a-manager.html]))  
(<https://ajahne.github.io/blog/leadership/2019/07/24/essential-meetings-to-have-with-your-people-as-a-manager.html%29>)
- [Essential Meetings to Have With Your People as a Manager](#)
- [1 on 1 Meeting Questions](#) - An extensive crowdsourced list.

## Communicating

- [The Senior Engineer’s Guide to Helping Others Make Decisions](#)
- [Management bugs: 18 months later](#)
- [How transparent should you be as a leader?](#)
- [The five types of communication problems that destroy company morale](#)
- [The Secret to Being a Better Boss: Create a “How to Work With Me” Manual](#)
- [BLUF: The Military Standard That Can Make Your Writing More Powerful](#)

## Ladders, Evaluations, Promotions and Related Topics

- [A software engineering manager guide to measuring a software engineers performance](#)
- [Square's Growth Framework for Engineers and Engineering Managers](#)
- [Stripe’s Will Larson on Designing a Performance Management System from Scratch](#)
- [Multi-team Software Delivery Assessment](#)
- [Engineering Career Development at Etsy](#)
- [Etsy Engineering Career Ladder](#)
- [Job Titles & Levels: What Every Software Engineer Needs to Know](#)

- [Individual Performance Appraisals, Just Say No!](#) - If the team does the work, why are you grading the people?
- [How I Do \(Hopefully\) Fair Performance Reviews for Software Developers](#)

## Remote Teams

- [Awesome Remote Job](#) - A big curated list of resources for remote jobs.
- [Best Practices for Managing Remote Teams: A Psychological Perspective](#)
- [Managing Remote Teams - A Crash Course](#)
- [12 Tips For Managing a Remote Team \(And Loving It\)](#)
- [Employee Engagement in a Remote World](#)
- [Better Remote Meetings](#) - Tips to equalize participation if you're split between in-person and remote.
- [Why remote work is inclusion work](#)
- [How to build social connection in a remote team](#)
- [Managing Remote Developer Teams: How Buffer Set the Gold Standard - Interview with Katie Womersley](#)
- [Remote Mob Programming](#)
- [Working with distributed teams](#) - "To be able to work well with distributed teams, we need to pay very close attention to our communication, in order to always strive to convey our message as clearly as possible, and in doing so, avoid having different views regarding the same subject. In addition to effective communication, we also need to efficiently use the tools that are available for this work context."
- [What is it like to work remotely as a software developer?](#)
- [Working from home: the yin and yang](#) - Advice on dealing with the physical and mental challenges of being out of the office full-time.
- [Tools That Make Work Faster](#) - "...when developing software in a remote team"
- [Asynchronous Communication: The Real Reason Remote Workers Are More Productive](#)
- [How to be a better remote worker](#)
- [The Manager's Schedule Is Holding Back Remote Work](#)
- [What Most Remote Companies Don't Tell You About Remote Work](#) - "Isolation, anxiety, and depression in the remote workplace and what we're doing about it"
- [Managing From Afar: How This Engineering Manager Tackles the Challenges of Remote Work](#)
- [Long-Distance Relationship: How To Work With Clients You've Never Met Face-to-Face](#) - Ebook from Toggl.
- [Remote Work Encyclopedia](#)
- [Scaling Culture: Retain Your Developer Team's Soul During Fast Growth](#)
- [A Distributed Meeting Primer](#)
- [A guide to distributed teams](#)
- [Working as a Remote Product Designer from far far away](#)
- [Are You Ready to Work Remotely?](#) - Guidance for newer developers.
- [How to Work From Home and Actually Get Stuff Done](#)
- [There's One Major Reason Remote Work Can Go Horribly Wrong](#)

- [The tools and tricks that let Ars Technica function without a physical office](#)
- [How I Work From Anywhere in the World](#) - With advice on travel as well as hardware/software issues.
- [Tracking the future of remote workplaces: Apps, communication, and liability](#) - Crystal-ball gazing from Ars Technica
- [Setting Up Your Webcam, Lights, and Audio for Remote Work, Podcasting, Videos, and Streaming](#)
- [Surviving meetings while remote](#)
- [Busting the common misconceptions about working from home](#)
- [The science behind self-care and practising it when working remotely](#)
- [Reworkin](#) - Social networking site for remote workers.
- [Open Offices Inhibit Remote Work](#)
- [Dear Founder: So You're Thinking About Building a Remote Team](#)

## Staying Technical

- [From Engineer to Manager: keeping your technical skills](#)
- [How \(and why\) Should Managers Code?](#)
- [The Fallacy of needing a technical manager](#)

## Team-Building and Scaling

- [The Essential Guide to Building Balanced Development Teams](#) - with a particular focus on junior/senior developer balance & mentorship
- [Building Distributed Engineering Teams](#) - Patterns and antipatterns from Doximity.
- [Scaling Engineering Organizations](#) - Stripe is way bigger than your organization, but there are some good thoughts on hiring, onboarding, and related topics here.
- [10 Tips to Boost Developer Productivity in Your Team](#)
- [How to Create a Feedback Culture](#)
- [Introducing Engineering Management to a Growing Organization](#)
- [Designing Engineer Onboarding at Affinity](#)
- [Camille Fournier on Scaling, Structure, and Growing as an Engineering Manager](#) - It's even more complex when the company is growing while you're getting more responsibility. Also has some material on composing an engineering track.
- [How to evolve an engineering organization](#). - Advice on scaling up from someone who's been there.
- [Scaling Technology and Organizations Together with Randy Shoup](#)
- [An Alternative Approach to Re-Orgs At Your Company](#)
- [Accelerate](#)[How to build a generative engineering culture] - If you read (agile\_scrum\_xp\_and\_lean) and the idea of a Westrum generative culture resonated with you, here are some ideas of things you as a manager can do to encourage one to flourish.
- [How to run a successful team retreat](#)
- [Organizing team offsites.](#)

## Values

- [Engineering Values](#) - A list of values for one team at Medium
- [Just Culture](#) - A healthcare quality concept that more software organizations should be familiar with.
- [How we align our goals](#)

## Your Influence

- [How to Make Sense of Your Impact When You're No Longer Coding](#)
- [Influence](#) - Pretty much everything on the Engineering Manager blog is worth reading, but this one stood out
- [Why Great Managers Matter](#)
- [Reasoning about Leverage in Engineering Organisations](#)
- [How to stop micromanaging and give your team autonomy](#)
- [Developer Burnout: How Can a Manager Spot It and Stop It](#)
- [It's Not All About You: Truly Empowering Your People](#)

## Beyond First-Line Manager

- [The Evolution of Management](#) - "Transitioning up the ladder"
- [What do executives do, anyway?](#) - Looking further up the ladder

## Unsorted Links

- [The Change Gap: How to Unleash Your Desired State](#)
- [How Google sets goals: OKRs](#)
- [How do managers get stuck?](#) - Managing down, up, and sideways for greater impact & visibility
- [The Geek's Guide to Leadership](#) (video specifically aimed at team leads)
- [Design Patterns for Managing Up](#) - "Four challenging work situations and how to handle them"
- [Learning by fixing — the value of triage engineer rotations](#)
- [Engineering Management Philosophies and Why They Matter Even if You are Not a Manager](#)
- [11 Practical Steps Towards Healthy Power Dynamics at Work](#)
- [Fire Fixation](#): "Fire fixation is the pattern of patching urgent problems as quickly as possible, and then immediately moving on to the next."
- [Targets and Estimations in Software Development](#) - You need to understand the difference.
- [Supermanagers](#)
- ["10x engineers": Stereotypes and research](#)
- [Clearbit's approach to management](#)
- [Lessons from Stripe](#): What one manager learned about ambition, optimism, and recruiting.
- [Starting an Engineering Management Book Club](#)

- [Markers of Progress in Incident Analysis](#) - How do you know you're actually learning from experience?
- [Lessons on leadership: The 10 most impactful lessons I've learned from 1,000+ managers in 2019](#)
- [How to build your company's engineering brand.](#)



## **Team Lead & Developer Resources**

# Engineering Hiring

## How to Hire

- [Buiding Intersectionality Into Your Hiring Strategy](#)
- [Great developers are raised, not hired](#) - Trying to hire "rock stars" is a waste of time.
- [How to make your organization attractive to engineering talent](#)
- [How NOT to hire a software engineer](#)
- [You probably don't factor in engineering time when calculating cost per hire. Here's why you really should.](#)
- [How Splice Builds Globally Distributed Engineering Teams](#)
- [How to Fix Your Tech Interview to Increase Diversity](#)
- [Assholes: A Probing Examination](#) - "Whatever you do, don't hire assholes at your company."
- [Want to hire the best programmers? Offer growth.](#)
- [The Ultimate Guide To Startup Hiring](#) - Hiring more than just engineers
- [Hiring is Broken and Yours is Too](#) - Pros and cons of all the common techniques for hiring developers.
- [I never liked technical interviews](#)
- [How we doubled the representation of women in Engineering at Clio](#)
- [Eric's Guide to Hiring {Software Developers}](#)
- [What's Wrong with the Tech Interview Process?](#) - A look at some of the issues, along with questions to ask yourself if you're trying to hire.
- [The Qualified Manifesto On Hiring Software Developers](#)
- [Reducing the Impact of Unconconscious Bias in Our Hiring Process](#)
- [How to Make Tech Interviews a Little Less Awful](#) 0 "What is your goal in interviewing candidates? If you answered 'to hire the best candidate' you should reconsider. Forming the best team

# Architecture

# Serverless

## **Design (UI and otherwise)**

# Microservices

## **SDLC (Agile and otherwise)**

# Security