

Notebook - The Engineering Executive's Primer

kindle

Larson, Will

Page 5 | Highlight

the hardest part of developing an Engineering strategy in my previous roles was usually building consensus around the solution. As an executive, the hardest part is building conviction that your strategy is right for your company.

Page 8 | Highlight

Engineering executive This is the functional leader for Engineering, who is responsible for both technical execution and people management within Engineering.

Page 13 | Highlight

“Why are you looking for an executive role?”

Page 13 | Highlight

The rationale doesn't need to be particularly compelling, just something positive that expresses your excitement and qualifications for the role.

Page 14 | Highlight

they're trying to filter out candidates with red flags: ego, jealousy, excessive status-orientation, and ambivalence.

Page 15 | Highlight

In executive hiring, there is a selection bias for confidence.

Page 16 | Highlight

it's harder to transition internally than externally.

Most executive roles are never posted on the company's jobs page.

Page 17 | Highlight

It's important to note that the most desirable roles, and roles being hired by a well-networked and well-respected CEO, will never reach an executive recruiting firm. If you try to enter your search without an established network and rely solely on executive recruiters to find roles, you are almost certain to be selecting from second-tier opportunities.

Page 19 | Highlight

Consequently, Engineering executive interviews depend more heavily on perceived fit, prestige, the size of the teams you've previously managed, how personable you are, and how well you would navigate the specific, concrete concerns of would-be direct reports and peers.

Page 20 | Highlight

You'll be evaluated primarily on your background, your preparation for the discussion, the quality of your communication, and perceived excitement for the company.

Page 22 | Highlight

Equity is issued in many formats: stock options, Restricted Stock Units, and so on.

Page 24 | Highlight

Severance packages can be negotiated, guaranteeing compensation after you exit the role.

Page 28 | Highlight

You will never have more leverage to get questions answered than during the hiring process: if it's important and they won't answer, be willing to walk away.

Page 29 | Highlight

the goal of your search is not to find an executive job, but rather to find an executive job where you will thrive.

Page 31 | Highlight

prioritize learning

Page 31 | Highlight

limit the number of early changes you make

Page 31 | Highlight

build trust and a support system

Page 31 | Highlight

organization's health and processes

Page 31 | Highlight

how work (and hiring, in particular) is executed Getting a grip on your technology when you have a limited amount of time to spend in your codebase

Page 32 | Highlight

your discovery process—figuring out what is important—requires some structure,

Page 32 | Highlight

How does the business work? Where does the money come from? Where does the money go?

Page 32 | Highlight

What defines the culture? What are the company's true values? What are some key decisions that

| the company has made recently?

Page 32 | Highlight

| How can you establish healthy relationships with peers and stakeholders? What do your peers need from Engineering and how can you help them succeed?

Page 32 | Highlight

| How can you build a relationship before you encounter your first source of conflict? Is the Engineering team executing effectively on the right work?

Page 33 | Highlight

| Is technical quality high? How well are tools supporting the team's day-to-day work?

Page 33 | Highlight

| Is it a high-morale, inclusive Engineering team? Who is successful within your team? Who isn't finding success?

Page 33 | Highlight

| Is the pace sustainable for the long haul?

Page 34 | Highlight

| durable improvements depend on building systems that create changes, not performing tactical actions that create the ephemeral appearance of improvement.

Page 34 | Highlight

| This is why to be a truly effective executive you must start by understanding the organizations and systems at play before you move toward changing them.

Page 34 | Highlight

If I had to create a short list of the top onboarding traps that new executives fall into, rushing to make changes before understanding the problem's shape is undoubtedly the first. The two other frequent mistakes are judging without context ("Ah, this technology is terrible, what sort of fools made this decision?") and reminiscing about past employers (the infuriating refrain of "At my last job, we..."). You Only Learn When You Reflect

Page 35 | Highlight

"20-40 rule." Always spend at least 20 minutes trying to solve a problem before asking for help, and never spend more than 40 minutes before asking for help.

Page 37 | Highlight

Ask your manager, probably the CEO or CTO, to write their explicit expectations for you.

Page 37 | Highlight

"What will success look like for my role?"

Page 37 | Highlight

Figure out if something is really wrong and needs immediate attention.

Page 38 | Highlight

Go on a listening tour.

Page 38 | Highlight

Set up recurring 1:1s and skip-level meetings.

Page 39 | Highlight

Share what you're observing.

Page 39 | Highlight

I've found weekly emails to your organization to be an effective format for this (more to come in Chapter 11), but you could also use a monthly all-hands meeting or other venue. Attend routine forums.

Page 39 | Highlight

Shadow support tickets.

Page 39 | Highlight

The most scalable way to hear from your customers early on is to plug into support tickets.

Page 40 | Highlight

Shadow customer meetings, partner meetings, or user testing.

Page 40 | Highlight

Find your business analytics and learn how to query the data.

Page 40 | Highlight

Build an external support group of folks in similar roles.

Page 41 | Highlight

Get an executive coach.

Page 41 | Highlight

Create space for self-care. Invest in yourself so that you have the energy to invest in others. Attend therapy, prioritize getting enough sleep, and schedule regular physical exercise.

Page 44 | Highlight

| Document existing organizational processes.

Page 45 | Highlight

| Implement at most one or two changes.

Page 45 | Highlight

| Plan organizational growth for next year.

Page 45 | Highlight

| Set up communication pathways.

Page 46 | Highlight

| Pay attention beyond the product engineering roles within your organization.

Page 46 | Highlight

| Spot check organizational inclusion.

Page 46 | Highlight

| Understanding Hiring

Page 46 | Highlight

| personally sourcing and closing senior candidates, and structuring an effective overall Engineering hiring process.

Page 46 | Highlight

| Track your funnel metrics and hiring pipeline.

Page 47 | Highlight

You'll want to get a recurring recruiting-oriented meeting onto the calendar as well, either weekly or bi-weekly, that pulls recruiters and hiring managers into a room to operate and improve together. Shadow existing interviews, onboarding, and closing calls.

Page 47 | Highlight

Decide whether an overhaul is necessary.

Page 47 | Highlight

Identify your (three or fewer) key missing roles.

Page 47 | Highlight

Offer to close priority candidates.

Page 48 | Highlight

Kick off Engineering brand efforts.

Page 48 | Highlight

getting an Engineering blog up, getting the team speaking at conferences, establishing an active presence on Twitter (X), and so on—all with the goal of exciting folks to work with your team.

Page 48 | Highlight

The biggest mistake a new leader can make is disrupting the current systems of execution before they have a working alternative.

Page 48 | Highlight

Figure out whether what's happening now is working and scales.

Page 49 | Highlight

Establish internal measures of Engineering velocity.

Page 49 | Highlight

Establish external measures of Engineering velocity.

Page 49 | Highlight

Consider small changes to process and controls.

Page 49 | Highlight

the two changes that I'd suggest considering are adding a weekly operational review meeting and moving into an organizational structure that cleans up any impediments to execution.

Page 49 | Highlight

Understanding the Technology

Page 50 | Highlight

Determine whether the existing technology is effective.

Page 50 | Highlight

Learn how high-impact technical decisions are made.

Page 50 | Highlight

Build a trivial change and deploy it.

Page 50 | Highlight

Do an on-call rotation.

Page 50 | Highlight

Attend incident reviews.

Page 51 | Highlight

Record the technology history.

Page 51 | Highlight

Most bad decisions today were great decisions within a context that no longer exists:

Page 51 | Highlight

Document the existing technology strategy.

Page 51 | Highlight

too many opinions or a lack of genuine curiosity about the rationale behind decisions.

Page 53 | Highlight

If only we had an Engineering strategy, their eyes will implore you, things would be OK.

Page 53 | Highlight

Stripe's aspirational approach to balancing technical standardization and exploration),

Page 53 | Highlight

an Engineering strategy is a document that defines: The what and the why of Engineering's resource allocation against its priorities The fundamental rules that Engineering's teams must abide by How decisions are made within Engineering

Page 54 | Highlight

down—is your constitutional document for running Engineering, and writing it is one of the most

valuable things you'll do as an Engineering executive.

Page 55 | Highlight

According to Rumelt, a strategy is composed of three parts: Diagnosis This is a theory describing the nature of the challenge. Its purpose is to identify the root cause(s) at play. For example, "high work-in-progress is preventing us from finishing any tasks, so we are increasingly behind each sprint" might be a good diagnosis. Guiding policies These are the approaches you'll apply to grapple with the challenge. Guiding policies are typically going to involve implicit or explicit trade-offs. For example, a guiding policy might be "only hire for the most urgent team with the most urgent need; do not spread hires across all teams." If a guiding policy doesn't imply a trade-off, you should be suspicious of it (for example, "working harder to get it done" isn't really a guiding policy). Coherent actions This is a set of specific actions directed by a guiding policy to address the challenge. This is the most important, and I think the most exciting, part because it clarifies that a strategy is only meaningful if it leads to aligned action.

Page 56 | Highlight

The reason most written strategies don't apply is that they're actually visions of how things could ideally work, rather than accurate descriptions of how things work today. This means they don't help you plot a course through today's challenges to the desired outcome.

Page 58 | Highlight

product engineer to infrastructure engineer

Page 58 | Highlight

tech spec review process.

Page 61 | Highlight

To write an Engineering strategy, write five design documents, and pull the similarities out. That's your Engineering strategy. To write an Engineering vision, write five engineering strategies, and forecast their implications two years into the future. That's your Engineering vision.

Page 61 | Highlight

consensus around. That consensus is the basis for enforcing the strategy going forward. As an Engineering executive, you don't need to rely on consensus to drive enforcement, which means you can take a much more direct path to writing your strategy document. Rather than enforcement, your biggest risks are writing a weak diagnosis or ineffective guiding policies.

Page 61 | Highlight

Commit to writing this yourself! Delegation is an extremely valuable executive skill, but the Engineering strategy will significantly shape how the Engineering organization functions. As the company's Engineering executive, you have the unique perspective to write this strategy.

Page 62 | Highlight

Focus on writing for the Engineering team's leadership, both your senior-most Engineering managers and senior-most engineers.

Page 62 | Highlight

Identify the full set of stakeholders you want to align the strategy with.

Page 62 | Highlight

It should likely include the executive team, the senior-most Staff-plus engineers, the senior-most Engineering managers, and potentially a few additional business and product leaders. From within that full set of stakeholders, identify three to five who will provide early, rapid feedback.

Page 62 | Highlight

strategy working group.

Page 63 | Highlight

steer away from including the CEO in this smaller working group. Write your diagnosis section.

Page 63 | Highlight

Write your guiding policies.

Page 64 | Highlight

get private feedback from two to three external Engineering executives.

Page 64 | Highlight

Now share the combined diagnosis and guiding policies with the full set of stakeholders.

Page 64 | Highlight

Companies are designed to spread context, not to conceal it, and you just can't fight that tendency. Instead, make sure to edit out anything too sensitive to share widely, particularly changes that directly impact individuals or teams. Write the coherent actions. The actions are usually less complex and less controversial than the guiding policies themselves, although not always.

Page 65 | Highlight

Partner with your working group to identify the individuals in the wider team who are most likely to be upset or to strongly disagree with the strategy.

Page 65 | Highlight

Share the written strategy with the Engineering organization.

Page 65 | Highlight

Finalize the strategy, send out an announcement, and commit to reviewing the strategy's impact in two months.

Page 66 | Highlight

A year later, I came back and documented three core, guiding principles: choose boring

technology, resolve conflict with curiosity, and prefer vendors for commoditized functionality.

Page 66 | Highlight

Are you confident in your diagnosis or do you trust the wider Engineering organization to inform your diagnosis?

Page 66 | Highlight

Are you willing and able to enforce the strategy?

Page 67 | Highlight

Are you confident the strategy will create leverage?

Page 67 | Highlight

an ineffective strategy will quickly wear down people's trust in you.

Page 67 | Highlight

every bad strategy rollout I've seen has been rooted in the executive's inability to listen to clear feedback.

Page 69 | Highlight

Some of the questions that I've found valuable to explore in these draft strategies are: What are the cash-flow targets? What is the investment thesis across functions (e.g., sales and marketing, research and development, general and administrative)? What is the intended role of mergers and acquisitions? What is the business unit structure? How do the business units support one another? How are costs expensed across business units? Who are your products' users, what do they need, and how are you prioritizing in relation to those users? How will other functions evaluate success over the next year? What are your current distribution mechanisms, and how are you trying to change them? What are the most important competitive threats? What about the current strategy is not working?

Page 70 | Highlight

Your diagnosis is your strategy's foundation.

Page 71 | Highlight

Don't skip writing the diagnosis, even if it's very tempting to do so.

Page 71 | Highlight

If you don't start with your diagnosis, you can't evaluate whether a solution is appropriate.

Page 71 | Highlight

When possible, have two or three leaders diagnose independently.

Page 71 | Highlight

Staff-plus product engineer, a Staff-plus infrastructure engineer,

Page 71 | Highlight

Diagnose with each group of stakeholder skeptics.

Page 72 | Highlight

Be wary when your diagnosis is particularly similar to that of your previous roles.

Page 72 | Highlight

There's one other understated value of writing an effective diagnosis: it makes it clear to readers that you're committed to your strategy, and acknowledging and resolving the issues at hand.

Page 73 | Highlight

Structuring Your Guiding Policies

Page 73 | Highlight

What is the organization's resource allocation against its priorities? (And why?) Competition can be healthy, but competing internally on budget and headcount tends to reward empire building rather than effectiveness.

Page 73 | Highlight

your Engineering strategy clearly articulates resourcing and priorities. As an Engineering executive, it's particularly important to think about the priorities that no one else is asking for (especially security, reliability, compliance, and developer productivity), and to ensure your investment thesis addresses those.

Page 73 | Highlight

We aim to maintain a ratio of four product engineers for every one platform engineer (e.g., security, reliability, infrastructure, developer productivity).

Page 74 | Highlight

What are the fundamental rules that all teams must abide by? (And why do they matter?)

Page 74 | Highlight

requiring all backend projects to be implemented in Golang would greatly narrow your security, compliance, and tooling needs. Similarly,

Page 74 | Highlight

Folks are much more open to following rules if you explain why the rules are valuable, so I strongly recommend you explicitly explain why each rule is important. Things that are obvious to you may not be obvious to others.

Page 74 | Highlight

All development must use our standard development stack (background services use Golang, frontend services use TypeScript, storage is in a service-isolated instance of Aurora PostgreSQL) and our development lifecycle (standard code review, linting, and deployment processes)

documented in the Development Lifecycle wiki). Exceptions to these rules must be approved by both tech spec review and the CTO. How are decisions made within Engineering? (And why do we work this way?)

Page 75 | Highlight

You want teams to know what they can decide themselves, what they should optimize for when making those decisions, and how to move forward with decisions that they can't make independently.

Page 75 | Highlight

Technical decisions that deviate from the standard development stack or standard development lifecycle should be approved by tech spec review and the CTO. Changes to those two standards should similarly be approved by tech spec review and the CTO. Changes to organizational structure, hiring prioritization, and general people processes should be approved by the CTO. All other decisions should be made by the teams and leaders closest to the decision. If anyone believes we are making a meaningfully suboptimal decision, please escalate that decision using our Escalation Process.

Page 76 | Highlight

ensure your strategy is operating at the right altitude,

Page 76 | Highlight

Applicable: It can be used to navigate complex, real scenarios, particularly when making trade-offs.

Page 76 | Highlight

Example: We generally prioritize stability of the existing product over new product work. If stability work takes less than a week, teams should self-approve the work. If it takes longer, they should review sequencing one step up their management chain. Example: We prefer SaaS vendors over building our own commodity solutions, but we only consider SaaS vendors with current SOC 2 Type 2 compliance. Build-versus-buy decisions should be reviewed by tech spec review. Exceptions to our SOC 2 Type 2 policy should be approved by the CTO (but won't be granted).

Enforced: Teams will be held accountable for following the guiding policy.

Page 77 | Highlight

yourself: are you willing to enforce this policy? If not, look for something else that you're willing to enforce. Often the gap between unenforceable and enforceable can be bridged by a simple nuance (such as "unless approved by the CTO"). Creates leverage: Create compounding or multiplicative impact.

Page 78 | Highlight

However, some approaches must be deployed at the Engineering strategy layer to be impactful, particularly standardization strategies that require organization-wide commitment (e.g., everyone uses TypeScript for frontend development). Your Engineering strategy also needs to solve for scenarios in which no team is capable of prioritizing a given effort, despite the effort being very valuable, such as a compliance or privacy initiative that doesn't fall cleanly into any given team's scope but is necessary for continued business operation. Example: Google historically constrained development to four languages: Python, C++, Go, and Java. They enforced this fairly rigorously and it created leverage in their development tooling. Each new project happening within that ecosystem increases a centralized tooling team's impact on the company. A closely related example is Dan McKinley's "Choose Boring Technology", a strategy that advocates building leverage by constraining technology choice, which was heavily enforced during Kellan Elliott-McCrea's era of Etsy Engineering leadership. Example: Uber (in 2014) had an implicit technology strategy, related to its "Let builders build" value: letting teams select their own tools. This aimed to create leverage by allowing teams to select the best tool for the job at hand and was enforced through both Engineering leadership's absence of an enforced counter-policy, and the use of a permissive service tool that merrily ran any Docker container. This approach is implicitly grounded in the theory that teams' individual gains will outweigh the inability to operate a high-leverage developer productivity team.

Page 80 | Highlight

Enforcements

Page 80 | Highlight

Enforcement actions explain the process used to maintain policies, as well as a clear, ongoing

process for evaluating exceptions.

Page 80 | Highlight

Example: Tech spec review will meet weekly and review requested exceptions to our standard development stack. Escalations How should folks constructively argue that a guiding policy doesn't work or doesn't apply to their case?

Page 80 | Highlight

Example: If you believe our guiding policy is meaningfully wrong, escalate up the technical or managerial reporting chain. Transitions How do we transition between the current state and the new state?

Page 81 | Highlight

Example: We are winding down our proposed migration to services, and instead recommitting to our monolithic service. Team structure won't be impacted, but priorities will be. First, individual teams will stop work on service migration. Second, our developer productivity team will make stable builds their top priority.

Page 81 | Highlight

Long-lasting guiding policies require fewer one-off actions, and more enduring actions to maintain the policies over time.

Page 81 | Highlight

Even with that efficiency argument in mind, there's a certain type of person who encounters a strategy and immediately argues that it's disempowering. The argument comes in a couple different flavors: Strategies are top-down, removing autonomy from teams doing the work. We value autonomy, so teams should determine their own strategies. Managers, including the Engineering executive, shouldn't set strategy; it should be owned by engineers rather than managers.

Page 82 | Highlight

When I hear arguments like these, usually the concern is really about how strategy is being set (e.g., did our migration to Go ignore certain perspectives?) or with a specific strategy (e.g., is our monorepo strategy not working?). If you can find the root concern, there's almost always something interesting to learn there!

Page 85 | Highlight

James Carse's *Finite and Infinite Games* (Free Press) proposes that you can view most things in life from two different perspectives. The first is seeing life as a series of finite games, with clear rules and specific ways to win. The second is seeing life as an infinite game, with rules that are evolved over time by its players, and where the goal is continuing to play. Although planning is often viewed as a finite, rules-heavy process, I find it much more useful to approach planning as an ongoing game with dynamic rules.

Page 86 | Highlight

but in practice, individual executives privately advocate to the CEO to increase their function's headcount. If the CEO rewards those private escalations, then that becomes the real process and the jointly created financial plan will become a polite fiction.

Page 88 | Highlight

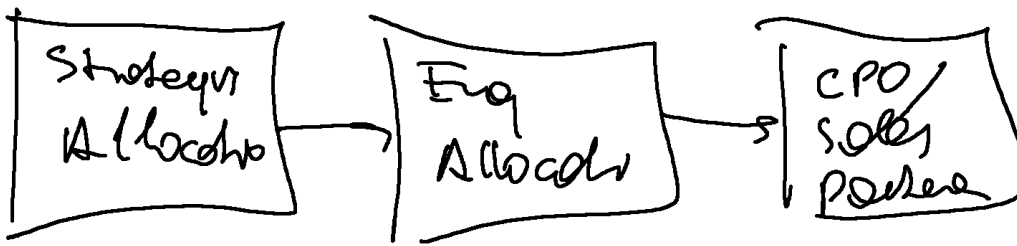
Your planning process will be effective to the extent that it can do three things well: Set your company's resource allocation across functions, as documented in an annual financial plan.

Page 88 | Highlight

Refresh your Engineering strategy's resource allocation, with a particular focus on Engineering's functional portfolio allocation between functional priorities and business priorities.

Page 88 | Highlight

Partner with your closest cross-functional partners, particularly Product and Sales (or whatever the leading go-to-market function is within your company), to establish a high-level quarter or half roadmap.



Page 89 | Highlight

Get comfortable with the reality that planning is dynamic!

Page 89 | Highlight

These constraints create better decisions because constraints drive innovation; removing constraints drives overly simplistic magical thinking

Page 90 | Highlight

Phase 1: Establishing Your Financial Plan

Page 93 | Highlight

How to Capitalize Engineering Costs Capitalizing software costs is a financial concern, not an Engineering one, driven by the Finance team's obligations to follow the Generally Accepted Accounting Principles (GAAP).

Page 93 | Highlight

The general guidance on capitalizing software engineering costs is straightforward: development of new software capabilities should be capitalized, and everything else should be expensed. In practice, there's a great deal of flexibility in deciding what is or isn't a new capability. This is the same challenge as agreeing on whether a given task is really a new feature and bug fix: most changes can be reasonably argued either way.

Page 93 | Highlight

three core criteria that your approach to capitalization must address to keep everyone involved happy: It must be easy to explain and justify to an auditor. It must be available in a timely manner for Finance's financial reporting. It must minimize both the Finance and Engineering teams' ongoing efforts.

Page 94 | Highlight

one of three approaches: Ticket-based Track the capitalization status and time for each ticket.

Page 94 | Highlight

Project-based Track hours for each project and determine a capitalization weighting for each project (80% capitalized, 0% capitalized, etc.)

Page 94 | Highlight

Role-based Set a fixed percentage of time for each role.

Page 95 | Highlight

The Reasoning Behind Engineering's Role in the Financial Plan

Page 95 | Highlight

your primary contribution to the financial plan is managing Engineering's expenses.

Page 95 | Highlight

Headcount expenses within Engineering Production operating costs (most cloud costs, vendor costs related to production, etc.) Development costs (vendor and hosting costs related to CI/CD, running test, development environments, etc.)

Page 95 | Highlight

For any business line where expenses accelerate faster than revenue, you and the entire executive

team should have a clear, documented hypothesis for when and how you'll reverse the setup.

Page 96 | Highlight

Adjusting your financial plan too frequently makes it impossible to grade execution, because your targets will keep moving. Making significant adjustments to your financial plan is a planning-intensive activity that requires a great deal of time across many functions. As such, changing the plan creates a great deal of churn, and often requires the reworking of other planning phases. Like all good constraints, if you make the plan durable, then it will focus teams on executing effectively within it. If you make it flexible, then teams will instead focus on moving the constraint (e.g., asking for more headcount). The former is much preferable to the latter.

Page 96 | Highlight

it's much easier to run an effective company with a stable financial plan.

Page 97 | Highlight

In that case, it's easy to attribute Enterprise-focused product engineers to the Enterprise business unit, but less clear how you should attribute product engineers building the core product.

Page 97 | Highlight

when executives are focused on their function rather than the executive team's shared success, allocating expenses is a zero-sum game. Combined with the tendency for struggling executives to point at an insufficient budget as the rationale behind their underperformance, it's easy to end up with either broken executive relationships or out of control spending.

Page 98 | Highlight

Should Engineering Headcount Growth Limit Company Headcount Growth? In general, I do believe that most companies should constrain overall headcount growth based on their growth rate for Engineering.

Page 98 | Highlight

Informing Organizational Structure

Page 99 | Highlight

Divide your total headcount into teams of eight. Each of these should have a manager and a mission. Group those teams into clusters of four to six.

Page 99 | Highlight

Continue recursively grouping until you get down to five to seven groups, which will be your direct reports.

Page 101 | Highlight

Vendors generally only value a few things: Money, preferably guaranteed A long-term commitment,

Page 101 | Highlight

Help meeting their internal goals,

Page 101 | Highlight

Sometimes timing of payments

Page 101 | Highlight

Sometimes co-marketing

Page 101 | Highlight

A volume-based discount so that as you spend more, their rates decrease. You should always pay a lower rate as your spend increases. A commitment length that aligns with your potential timeline to consider other vendors. Generally, prefer one year unless you're getting something meaningful in return. Discounts based on service and delivery issues during the previous contract. If they had

Page 101 | Highlight Continued

frequent outages, this should be reflected in the following contract to some extent.

Page 103 | Highlight

Phase 2: Determining Your Functional Portfolio Allocation

Page 104 | Highlight

Once a year or so, review your full set of Engineering investments, their impact, and the potential investments that you haven't made yet.

Page 105 | Highlight

Why Do We Need a Functional Portfolio Allocation?

Page 106 | Highlight

Things like compliance, security, reliability, and so on should really be fundamental company work, not invisible Engineering functional work. Bringing Engineering work out of functional allocation is much more effective than trying to bring non-Engineering executives in.

Page 106 | Highlight

Changing your allocation feels like progress, but each change is fairly disruptive, so there's a significant penalty to making frequent changes.

Page 106 | Highlight

Be Mindful of Allocation Granularity

Page 106 | Highlight

The larger the granularity that you allocate, the more you empower your team to lead their teams.

Page 107 | Highlight

Don't Over-index on Early Results

Page 108 | Highlight

Phase 3: Agreeing on the Roadmap

Page 108 | Highlight

it's rarely the format that causes roadmapping to fail.

Page 108 | Highlight

Disempowering teams by planning too deeply

Page 108 | Highlight

Roadmapping with Disconnected Planners

Page 109 | Highlight

The frequent mistake I see here is executives trying to solve this through a series of one-on-one discussions.

Page 109 | Highlight

Roadmapping Concrete and Unscoped Work

Page 109 | Highlight

planning processes often proactively invite new ideas: The implicit or explicit ask of Planning is often, "Tell us about all your exciting new ideas. We need your creativity to achieve our goals. Swing for the fences! Throw big rocks!" This is a mistake. The challenge is that new ideas are unscoped and unproven, and you end up comparing the raw potential of an unscoped idea against the proven potential of an existing idea.

Page 110 | Highlight

| Agree on an allocation between scoped and unscoped initiatives,

Page 110 | Highlight

| 20% of Product Engineering time should go against validating new projects.

Page 110 | Highlight

| Have a long-running allocation toward validating projects, say 10% of Product Engineering time,

Page 110 | Highlight

| Roadmapping in Too Much Detail

Page 110 | Highlight

| roadmapping that is focused too narrowly on project to-do items rather than on desired outcomes can constrain teams' thinking.

Page 111 | Highlight

| the goal should be to emphasize the target outcome first at a high level of confidence, and then discuss the specific project second with a lower degree of confidence. As long as they remain aligned with the target outcome, the team should feel empowered to change the project.

Page 113 | Highlight

| Platform Engineering, Developer Productivity, or Infrastructure Engineering.

Page 114 | Highlight

| Planning as Ticking Checkboxes Signs that you've delegated planning to someone who is process-oriented rather than outcome-oriented,

Page 114 | Highlight

Planning is a ritual rather than part of doing work.

Page 114 | Highlight

Planning is focused on format rather than quality.

Page 114 | Highlight

The executive provides direction and makes trade-offs, but it's the planning implementor who designs the templates, finalizes the schedule, manages exceptions, checks to make sure initial drafts are useful, and generally keeps the process on track. Planning as Inefficient Resource Allocator Signs that the executive team is optimizing for their function, rather than working together to optimize for the company's overall outcomes,

Page 115 | Highlight

Planning creates a budget, then ignores it.

Page 115 | Highlight

Planning rewards the least efficient organizations. Often leaders will sandbag their organization's capabilities, arguing that they need significantly more headcount to even continue operating at their current level. This culminates in a business where most resources are directed toward your least efficient leaders and organizations.

Page 116 | Highlight

Planning treats headcount as a universal cure.

Page 116 | Highlight

Planning as Rewarding Shiny Projects Signs that your planning process is more focused on energizing the executive team than solving the inherent resource allocation and functional alignment problems

Page 116 | Highlight

Planning is anchored on work the executive team finds most interesting.

Page 116 | Highlight

most executive teams are excited to discuss sales numbers or new product development, but fewer are thrilled to discuss compliance.

Page 117 | Highlight

Planning only accounts for cross-functional requests.

Page 117 | Highlight

For example, some companies view customer satisfaction, security, compliance, and stability as the responsibility of a single function.

Page 117 | Highlight

Planning as Diminishing Ownership Signs that the executive team is approaching planning in a way that is reducing autonomy and ownership within your broader team

Page 117 | Highlight

Planning is narrowly focused on project prioritization.

Page 118 | Highlight

It's more effective for executives to prioritize outcomes, which leave sufficient flexibility in their execution, than to steer projects toward those outcomes that heavily constrain non-executive leadership. Planning generates new projects.

Page 118 | Highlight

sometimes those distant executives will use planning as an opportunity to suggest significant new, unscoped work.

Page 121 | Highlight

Stated values make it clear how you want people to make decisions.

Page 121 | Highlight

Documented values increase cohesion across the new and existing team. They also avoid the scenario in which new hires unknowingly practice their previous companies' values, causing a cultural rift between new and existing teams.

Page 121 | Highlight

want to formalize that change so it persists over time.

Page 122 | Highlight

your documented values help them navigate that change successfully.

Page 122 | Highlight

they are absolutely not an effective way to start the underlying culture change that they describe. Rather, they celebrate change that's already in motion, and help make it permanent.

Page 122 | Highlight

whether Engineering should have values at all.

Page 123 | Highlight

there are three paths forward: add a new company value, introduce Engineering values, or introduce Engineering leadership values (for Engineering managers, Staff-plus engineers, or ideally both).

Page 123 | Highlight

It's much more work to maintain values than to create them, and adding to the company values will allow you to share the maintenance across the executive team.

Page 124 | Highlight

Amazon's Principal Engineering Community Tenets. (Note that these tenets apply really well to anyone in Engineering, regardless of their title.)

Page 124 | Highlight

a useful value is reversible, applicable, and honest. Reversible: It can be rewritten to have a different or opposite perspective without being nonsensical.

Page 125 | Highlight

Uber's value of Make Magic

Page 125 | Highlight

Amazon's Are Right, A Lot.

Page 125 | Highlight

Applicable: It can be used to navigate complex, real scenarios, particularly when making trade-offs.

Page 126 | Highlight

Honest: It accurately describes real behavior. A touch of aspiration is OK, but useful values should explain how effective employees navigate the organization as it exists today.

Page 126 | Highlight

Uber's value of Meritocracy and Toe-Stepping

Page 126 | Highlight

Amazon's Strive to be the Earth's Best Employer.

Page 127 | Highlight

Amazon's Dive Deep asks leaders to engage with the details at hand.

Page 127 | Highlight

Stripe's Seek Feedback asks folks to socialize plans and decisions before finalizing them.

Page 127 | Highlight

Y Combinator's Do Things That Don't Scale

Page 128 | Highlight

Identity values are things like Amazon's Are Right, A Lot, Uber's Champion Mindset and Stripe's Deliver Outstanding Results. Identity values can be reversible and may be honest, but they are very rarely applicable.

Page 129 | Highlight

The best way to think of the relationship between values and a strategy (business, technology, or otherwise) is that a useful value generally can serve as a strategy's guiding principle.

Page 129 | Highlight

Reuse common technologies unless you see 10x improvement

Page 130 | Highlight

Wait. Wait at least six months. If that's too long, at minimum wait until you can evaluate whether a given value is honest or aspirational.

Page 130 | Highlight

the general patterns of good process rollout: identify the opportunity, document potential options, involve stakeholders early to build buy-in, test before finalizing to avoid folks feeling trapped, and iterate until it's useful.

Page 131 | Highlight

Integrate them into your hiring process (including letting candidates opt-out if they don't like them). Explicitly talk about them in new-hire onboarding. Update your career ladder to require value-aligned behavior for promotions. Highlight culture-aligned accomplishments in 1:1s, team meetings, and elsewhere.

Page 131 | Highlight

Some Values I've Found Useful

Page 131 | Highlight

The challenge with copying values is tied directly to the definition of useful values: reversible, applicable, and honest.

Page 132 | Highlight

Create capacity (rather than capture it).

Page 132 | Highlight

Default to vendors unless it's our core competency.

Page 133 | Highlight

Follow existing patterns unless there's a need for order-of-magnitude improvement.

Page 133 | Highlight

Optimize for the [whole, business unit, team]. Pick one of these options to help leaders understand how they should balance decisions between impacting their team and impacting other teams.

Page 134 | Highlight

Approach conflict with curiosity. One of my foundational beliefs is that most professional conflict

Page 134 | Highlight Continued

between reasonable people is driven by asymmetric information.

Page 134 | Highlight

learned to evaluate values according to their reversibility, applicability, and honesty.

Page 137 | Highlight

Measuring for Yourself

Page 138 | Highlight

Measure to Plan

Page 138 | Highlight

You want a single place that shows the Business, Product, and Engineering impacts of work.

Page 138 | Highlight

Measure to Operate

Page 138 | Highlight

measures of execution quality,

Page 138 | Highlight

tracking the number of incidents (each connected to an incident review writeup), minutes of user-facing API and website downtime, latency of user-facing APIs and websites, engineering costs normalized against a core business metric (for example, cost to serve your most important API, calculated simply by dividing API requests per month by engineering spend for that month), user ratings of your product (e.g., App Store ratings), and a broad measure of whether your product's onboarding loop is completing (e.g., what percentage of users reaching your website successfully convert into customers). Measure to Optimize

Page 139 | Highlight

Maintain a list of technical investments that turned something impossible into something so obvious that it isn't really worth mentioning. When you speak at company meetings,

Page 140 | Highlight

The secret of good measurement is actually measuring something. The number one measurement risk is measuring nothing because you're trying to measure everything.

Page 140 | Highlight

don't be afraid to measure something quick and easy today.

Page 140 | Highlight

One reason to focus on quick measurements rather than perfect measurements is that there are a lot of folks within your company who want you to measure something (often to confirm or disprove their intuition about your current trendline).

Page 141 | Highlight

Measure for Your CEO or Your Board

Page 141 | Highlight

This means they want you to measure something, set a concrete goal against it, and share updates on your progress against that goal. They don't have an engineering background to manage your technical decisions, so instead they'll treat your ability to execute against these goals as a proxy for the quality of your engineering leadership.

Page 141 | Highlight

Finance teams generally have three major questions for Engineering. First, how is actual headcount trending compared to budgeted headcount? Second, how are actual vendor costs trending compared to budgeted vendor costs? Third, which Engineering costs can be capitalized instead of expensed, and how do you justify that in the case of an audit?

Page 142 | Highlight

Although I've rarely had Finance ask for it, I find it particularly valuable to align with Finance on Engineering's investment thesis for allocating capacity across business priorities and business units.

Page 142 | Highlight

Measure for Strategic Peer Organizations Your best peer organizations are proactive partners in maximizing Engineering's business impact.

Page 142 | Highlight

Product, Design and Sales functions that are well-positioned for strategic partnership with Engineering.

Page 143 | Highlight

Measure for Tactical Peer Organizations

Page 143 | Highlight

more tactical organizations will generally demand to measure Engineering based on more concrete outcomes.

Page 143 | Highlight

Customer Success organization may push Engineering to be measured by user ticket acceptance rate and resolution time.

Page 144 | Highlight

Some things are difficult to measure, so only measure those if you'll actually incorporate that data into your decision making.

Page 144 | Highlight

Some things are easy to measure, so be willing to measure those to build trust with your stakeholders, even if you don't find them exceptionally accurate.

Page 144 | Highlight

Whenever possible, only take on one new measurement task at a time.

Page 145 | Highlight

Measurement is rife with antipatterns.

Page 145 | Highlight

Focusing on measurement when the bigger issue is a lack of trust.

Page 145 | Highlight

While metrics can support building trust, they are rarely enough on their own. Instead, push the CEO (or whoever remains frustrated despite your bringing more metrics) on their frustration, until you get to the bottom of the concern masquerading as a need for metrics. Letting perfect be the enemy of good.

Page 146 | Highlight

Using optimization metrics to judge performance.

Page 146 | Highlight

Instead, evaluate teams based on their planning or operational metrics. Measuring individuals rather than teams.

Page 147 | Highlight

Worrying too much about measurements being misused.

Page 147 | Highlight

Deciding alone rather than in community.

Page 147 | Highlight

Particularly when you're new to a company, it's easy to project your understanding of your last job onto the new one, which erodes trust. Instead, build trust by incorporating feedback from your team and peers.

Page 148 | Highlight

The reality is that measurement is not inherently valuable; it is only by repeatedly pushing on data that you get real value from it.

Page 148 | Highlight

Review the data on a weekly cadence.

Page 148 | Highlight

Maintain a hypothesis for why the data changes.

Page 148 | Highlight

Avoid spending too much time alone with the data.

Page 149 | Highlight

Segment data to capture distinct experiences.

Page 149 | Highlight

Discuss how the objective measurement does or does not correspond with the subjective experience of whatever the data represents.

Page 152 | Highlight

evaluating acquisitions is some of the most interesting work that Engineering executives do, and that managing Engineering's "no" in the M&A process is uniquely impactful work.

Page 153 | Highlight

Your M&A team may be evaluated by their ability to complete a certain number of acquisitions that contribute a certain amount of revenue to the business. That goal would align them toward moving forward with almost any revenue-contributing acquisition, even if it's an exceptionally painful integration.

Page 154 | Highlight

Business strategy: what is the desired role of M&A in your business? Acquisition thesis: what are the assumptions that need to be true for this particular acquisition to fit into your business strategy? Engineering evaluation: are you able to disprove any of those assumptions, or uncover other significant risks?

Page 155 | Highlight

From an M&A perspective, your business strategy is how you expect acquisitions to contribute to your company.

Page 155 | Highlight

common M&A strategies include: Acquiring revenue or users for your core business

Page 156 | Highlight

Entering new business lines via acquisition

Page 156 | Highlight

Driving innovation by acquiring startups in similar spaces

Page 156 | Highlight

Reducing competition. Finally, you will absolutely never see this in writing, because no one wants to show up in court documents, but many acquisitions are motivated by reducing future competition.

Page 156 | Highlight

any strategy is reasonable as long as it gets written down, and the executive team is aligned around it.

Page 156 | Highlight

Your acquisition thesis is how a particular acquisition fits into your company's business strategy.

Page 156 | Highlight

what, specifically, needs to be true for this acquisition to fit into your business strategy?

Page 158 | Highlight

Venture's risk model allows it to miss nine times as long as it gets the tenth one right, but as an acquiring company you can't simply write down an acquisition; you're also on the hook for the costly integration necessary for it to have a chance to succeed.

Page 158 | Highlight

Your Finance team will focus on revenue forecasts, your Legal team will focus on intellectual property, and within Engineering you'll need to focus on validating assumptions around implementation complexity, integration, scalability, security, compliance, Engineering costs, and Engineering culture.

Page 159 | Highlight

Don't assume you'll have six meetings; plan for having two.

Page 161 | Highlight

Product implementation

Page 161 | Highlight

you want to verify that the implementation is real and that it is a first-party implementation rather than powered by a third-party capability. For example, OpenAI has spawned a flurry of AI-driven startups whose implementation is primarily calling OpenAI's APIs, rather

Page 162 | Highlight

Intellectual property

Page 162 | Highlight

Security Who's responsible for security at the company, and what are they doing? Are they doing routine penetration tests, and where are their latest results?

Page 162 | Highlight

Compliance What compliance certifications does the company have? SOC 2 Type 2? HITRUST? Nothing?

Page 163 | Highlight

Integration mismatches

Page 163 | Highlight

Costs and scalability

Page 164 | Highlight

make it clear whether there are economies or dysfunctions of scale, which will help you understand the product's long-term revenue and cash-flow contributions.

Page 164 | Highlight

Engineering culture

Page 164 | Highlight

How does the company make technology decisions? How do they structure teams? Does the Engineering contact mention other people, or primarily speak of themselves as the decision maker?

Page 165 | Highlight

A surprising number of acquisitions are completed with only an informal integration plan, which leaves a lot of room for miscommunication between the acquiring and acquired companies.

Page 165 | Highlight

Commit to running the acquired stack “as is” for the first six months.

Page 165 | Highlight

Bring the acquired Engineering team over, with their head of Engineering reporting into your head of Engineering (or the appropriate business unit’s engineering lead if they are a small acquisition contained within a single business unit). Let the acquired Engineering leader know that you will aim to combine vertical teams (developer productivity, infrastructure, and so on) as feasible. Be direct and transparent with any senior leaders about roles where they could step in, and how you would make that decision to maximize their success along with the existing team’s. Even if someone is deeply qualified, you’ll better set them up for success by letting folks get to know them a bit before inserting them as a new boss.

Page 166 | Highlight

most important questions to work through are: How will you integrate the technology? How will you integrate the teams? How will you integrate the leadership?

Page 168 | Highlight

Success depends on both groups working together effectively, and it’s easy to dig a deep hole if

| your existing team feels the acquired team is underqualified for their new roles.

Page 169 | Highlight

| Sometimes this is pretty difficult, and won't make you friends. My best advice is to anchor your feedback to the company's goals, rather than Engineering's, and to make sure your concerns are truly from the company perspective. Ignoring the philosophical question of whether it's fair, Engineering will incur much of any acquisition's operating cost and complexity and owes the company (and itself) a strong, clear point of view.

Page 169 | Highlight

| Sometimes, being an executive means doing something painful for your function that may be impactful for the broader business.

Page 170 | Highlight

| The best thing you can do for the team is fight for their compensation packages before the acquisition finalizes, and then push them to acclimate to the new world.

Page 173 | Highlight

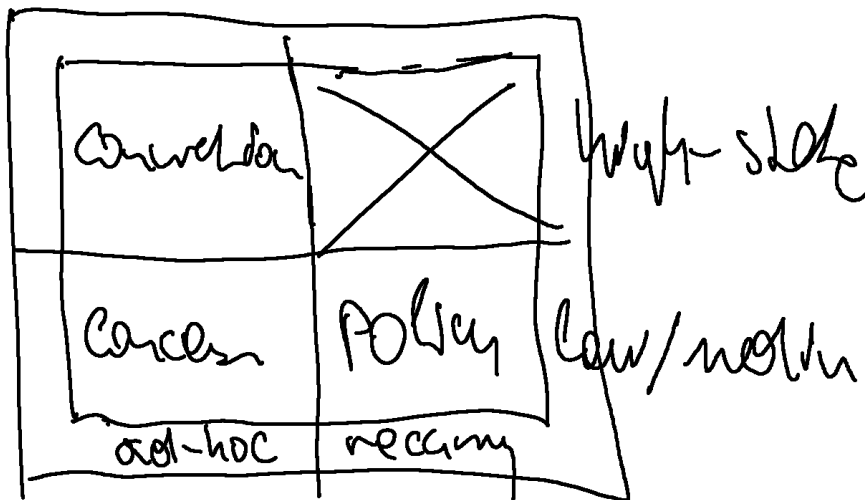
| effective executives have a number of distinct styles that they're able to swap among to solve a given challenge.

Page 174 | Highlight

| three primary styles: leading with policy, leading from consensus, and leading with conviction

Page 175 | Highlight

| Good policy empowers your organization to move quickly without you in the room, and is the foundation of an effective organization with 50 or more engineers.



Page 175 | Highlight

Leading with policy For recurring decisions that need to be made consistently by many individuals within your organization
 Leading from consensus For infrequent decisions with context spread across a number of different stakeholders
 Leading with conviction For decisions without any clear proposal, where involved individuals are deeply at odds with one another, or that have outsized, long-term impact on your organization

Page 176 | Highlight

Leading with policy is creating a documented, predictable means for making a decision.

Page 178 | Highlight

Leading from consensus is pulling together the relevant set of stakeholders to identify a shared approach for a given problem.

Page 178 | Highlight

Consensus has a reputation for being slow, but I think that's an overstated concern. In situations where no one has the entire relevant context, there's no particularly quick solution to making a robust decision. You can't rely on policy for these decisions because you won't have enough repeat exposure to base your policy on.

relevant policy is a candidate for consensus-based decision making, but where consensus shines is when decisions must be made with many stakeholders, none of whom has the full and relevant context.

Page 181 | Highlight

Write a framing document capturing the perspectives you know and encourage other stakeholders to add their perspectives.

Page 181 | Highlight

Identify the most senior, engaged party (or parties) to decide how the group will work together on the decision. Particularly important is aligning on the deadline for making the decision.

Page 181 | Highlight

First, make sure that all stakeholders feel heard. Second, set a tight timeline and hold to it. These two ideas balance building buy-in and moving quickly, which is the key trade-off in building consensus.

Page 181 | Highlight

Leading with conviction is pulling all the relevant context into your head, thinking through the trade-offs, and personally making a definitive decision. It's most useful in scenarios where no one has a clear perspective on making a decision, where the involved stakeholders have wholly incompatible views on the path forward or on how to make decisions with significant, long-lasting consequences (e.g., spinning up or shutting down a business unit).

Page 183 | Highlight

Identify a meaningfully important scenario in which no one has the full context to make a high-quality decision, and where the decision is unlikely to repeat (e.g., performance review happens a lot, a specific acquisition happens exactly once).

Page 185 | Highlight

leaders work on the system, not in the system.

Page 185 | Highlight

being too far removed was just as disempowering as being overly engaged. It's not micromanagement to know what a team is doing, nor to question the thinking behind a decision they're making.

Page 185 | Highlight

If you ever have someone imply that those activities are micromanagement, then it's far more likely that the individual is misaligned with you than that you are too deep in the details. Don't let accusations of micromanagement steer you away from doing your job. To evaluate whether you are micromanaging, ask yourself two questions: Are you carefully considering the feedback of those closest to the work before making decisions? Are you adding friction to a team that would have gotten to a similar quality decision without your involvement? As long as you can honestly answer "yes" to the first and "no" to the second, then don't spend more time worrying about it.

Page 187 | Highlight

most executives have a strong tendency to rely on a preferred style, and it's important to recognize situations in which your own default may lead you astray.

Page 189 | Highlight

External demands have broken the executive's work systems.

Page 189 | Highlight

The executive's pursuit of "enlightened distance" to avoid micromanagement has gone too far.

Page 189 | Highlight

The executive is chasing energy and will continue to drift toward wherever energy accumulates in their life.

Page 189 | Highlight

Admittedly, as you get further into your career, you're increasingly unlikely to have a helpful, attentive manager, but it's still likely that a peer will nudge you

Page 191 | Highlight

your role is too challenging, or not challenging enough.

Page 193 | Highlight

reach the right outcomes by prioritizing the company first, your team second, and yourself third.

Page 193 | Highlight

It also helped me articulate why I disagreed with some of my peers' decisions that violated this hierarchy by placing individual or team preferences over the company's priorities.

Page 193 | Highlight

A long-term career depends equally on being impactful and staying engaged.

Page 195 | Highlight

the most valuable work at a company is rarely the most interesting nor—oddly enough—what the company itself particularly values.

Page 196 | Highlight

folks will accomplish more if you let them do some energizing work, even if some of that work isn't very important.

Page 196 | Highlight

It's not only reasonable to violate perfectly correct priorities to energize yourself and your team, but modestly violating priorities to energize your team in pursuit of a broader goal is an open leadership secret. Leadership is getting to the correct place quickly; it's not necessarily about walking in the

straightest line.

Page 196 | Highlight

There are, of course, rules to breaking the rules. The most important being that your energizing work needs to avoid creating problems for other teams.

Page 197 | Highlight

you'll be a better leader if your decision making modestly factors in your team's energy rather than ignoring it.

Page 197 | Highlight

I think of this new framework as "eventual quid pro quo." The core framework is: Generally, prioritize company and team priorities over my own. If I'm getting de-energized, artificially prioritize some energizing work. Increase the quantity until equilibrium is restored. If the long-term balance between energy and proper priorities can't be achieved for more than a year, stop everything else and work on solving this (e.g., change your role or quit).

Page 199 | Highlight

There is always another layer of context that validates or invalidates any given piece of work. This is why operating from a place of curiosity is so powerful: folks who appear to be doing something nonsensical almost always have more information than you, or are missing a key piece of information that you have.

Page 199 | Highlight

You'll rarely be missing strategic context but you'll increasingly be missing the tactical context around how things actually work. Seniority creates just as many gaps in knowledge as it closes.

Page 199 | Highlight

small amount of work that's orthogonal to the company's needs, you will always regret doing work that is opposed to those needs.

Page 201 | Highlight

I've never really seen high-adherence personalities succeed in senior engineering roles.

Page 201 | Highlight

It's surprisingly common that they're stuck because they are someone who is energized by adherence, views a lack of adherence in others as a character flaw, and consequently alienates those they need to build relationships with to succeed.

Page 204 | Highlight

Why a monthly Engineering Q&A session is my favorite meeting

Page 205 | Highlight

surprising: large meetings are rarely the best communication solution for any particular goal. However, they are a remarkably effective backup solution when there are gaps in your default approaches.

Page 205 | Highlight

distribute context down the reporting hierarchy,

Page 205 | Highlight

communicating culture,

Page 205 | Highlight

surfacing concerns from the organization.

Page 206 | Highlight

six core meetings

Page 206 | Highlight

Weekly Engineering Leadership Meeting

Page 207 | Highlight

it's your forum for forging your direct reports into a "first team" whose top priority is to support each other.

Page 207 | Highlight

My weekly leadership meeting always includes my direct reports and usually includes our key partner in the Recruiting, People, and Finance teams.

Page 207 | Highlight

Maintain a running agenda in a group-editable document, so anyone in the team can add agenda topics over the course of the week.

Page 207 | Highlight

I'd strongly push the core Engineering leadership team to meet weekly.

Page 208 | Highlight

Weekly Tech Spec Review and Incident Review

Page 208 | Highlight

All reviews should be anchored to a concise, clearly written document. Reading the written document should be a prerequisite to providing feedback on it.

Page 208 | Highlight

Good reviews are anchored to feedback from the audience, and discussion between the author and the audience. Bad reviews are anchored to the author presenting their document. You will only have good reviews if you're diligent about steering folks away from presenting their writeups. There

should be a clearly documented, very simple process for getting your incident writeup or tech spec scheduled. An effective approach I've seen is having a dedicated room in Slack to request that your document is added to the next meeting.

Page 209 | Highlight

Prepare first-time facilitators with a practice run with an experienced facilitator!

Page 209 | Highlight

Running these reviews well is time consuming, and I highly recommend finding dedicated owners for each meeting. Although a bit annoying at times, both are highly impactful leadership opportunities that can be run by an experienced engineer.

Page 209 | Highlight

Monthlies with Engineering Managers and Staff Engineers I run two monthly meetings focused on ongoing team development. The first is the Engineering managers' monthly, attended by all Engineering managers, and the second is the staff engineers' monthly,

Page 210 | Highlight

[15 minutes] Ask each member to take one minute to share something they're working on, something they're excited about, and something they're worried about. [30 minutes] Someone, often me, presents on a development topic.

Page 210 | Highlight

Engineering Q&A. I run it monthly for an hour, starting with introductions of new team members, and then taking a few minutes to share any messages I'd like the team to think about. Then we move into Q&A for the remainder of the session. If we run out of questions, we end early, but I find that rarely happens.

Page 211 | Highlight

I start every Q&A by saying, "I'm glad to talk about anything, no questions are off limits. If your

| question is too awkward or private, I'll just avoid answering it. Ask whatever you want."

Page 211 | Highlight

| Having a good tool for taking questions goes a very long way. I recommend three core features: to be able to ask questions before the meeting starts, to vote on which questions are most important to answer, and to ask anonymous questions. I would always prefer difficult anonymous questions over folks feeling too uncomfortable to ask them, even if it makes for an awkward session. Remind folks the day and hour before the meeting that the Q&A is coming up and steer them toward the Q&A tool to maximize the incoming questions and the number of people asking them. Use the Q&A as an opportunity to highlight individuals doing important work or those who are key leaders in your organization. When a question comes in that touches on someone's work, I warn them that I'm going to share my thoughts quickly before passing the question to them, giving them a few minutes to prepare. Every organization has a small number of folks who can't help but ask questions. Nurture those folks to ensure you get at least some questions! Good tooling (a question-voting tool) and hygiene (creating pauses even when hands are up to allow others to ask questions, too) will mitigate the potential downside of only answering their questions.

Page 212 | Highlight

| A. I generally recommend spreading sessions across time zones instead of increasing frequency. I also generally recommend against recording Q&As, as question quality tends to drop.

Page 213 | Highlight

| 1:1 meetings I highly recommend weekly 1:1s with your direct reports and the peers you work with most closely.

Page 213 | Highlight

| Skip-level meetings

Page 214 | Highlight

| Execution meetings You absolutely want some kind of weekly or monthly execution review meeting, to track and debug execution.

Page 214 | Highlight

Show-and-tells or demos

Page 214 | Highlight

Frequency across companies varies a bit from weekly to monthly. Tech talks or lunch-and-learns

Page 215 | Highlight

Engineering all-hands

Page 217 | Highlight

You cannot scale large meetings without holding participants responsible for their behavior.

Page 219 | Highlight

Maintain the drip Communicate on a regular basis, even if you don't have something novel to share. Test before broadcasting Review your communication with a few people before sending it, to improve clarity and avoid blind spots. Build the packet Ensure every communication has a concise summary, and link to background context and an explicit way to ask questions. Keep it short Edit messages for brevity. Use every channel Distribute important information across every relevant channel, including meetings, email, chat, and so on.

Page 220 | Highlight

you can close the distance between yourself and the team by sending a short, weekly email.

Page 220 | Highlight

Over the course of each week, I would accumulate my weekly updates in one document, then spend about 20 minutes compiling them into the final email.

Page 221 | Highlight

One to two sentences about something human.

Page 221 | Highlight

One sentence summarizing any key reminders for upcoming deadlines or dates. One paragraph for each important topic that has come up over the course of the week.

Page 221 | Highlight

A bulleted list of brief updates.

Page 221 | Highlight

Close with an invitation for folks to reach out with questions, thoughts, and concerns.

Page 221 | Highlight

Dense updates are good, but it's the brief updates that reassure the team that you would have updated them if there was something they needed to know.

Page 222 | Highlight

Even if they convince the CEO that a custom model is the better choice, the CEO will be annoyed that their real concern about the schedule wasn't addressed.

Page 223 | Highlight

getting caught up on the incidental details in their question.

Page 223 | Highlight

At least half of the self-inflicted messes that I've seen executives create are caused by unsympathetic or confusing communication,

Page 223 | Highlight

recommend directly asking individuals for feedback, as that tends to inspire more useful feedback than asking a group

Page 224 | Highlight

testing the message speeds up the pace of communication, because it reduces the time spent explaining the message.

Page 225 | Highlight

Link to the best place for more in-depth information, generally a document or wiki.

Page 226 | Highlight

Email and chat

Page 227 | Highlight

Everything important should be communicated across both. Meetings

Page 227 | Highlight

Meeting minutes

Page 227 | Highlight

Weekly notes

Page 227 | Highlight

share weekly notes with their organization.

Page 227 | Highlight

Decision log

Page 228 | Highlight

Summary

Page 229 | Highlight

the value of brand building is overemphasized, but that the impact of enhancing prestige is underrated, and that there's a relatively straightforward playbook to increase your personal and organizational prestige.

Page 230 | Highlight

Your brand is a deliberately crafted, sustained narrative that is widely known about you.

Page 230 | Highlight

few of the most successful folks I know are well-known online, and many of the most successful folks I know don't create content online at all.

Page 231 | Highlight

The prestige is the passive-awareness counterpart to your brand. Rather than being what someone already knows about you, it's what someone can easily discover about you if they do some searching. Many interviewers won't know anything about me, but a few minutes of research will find my writing, conference talks, and work history. You can build prestige in a couple ways: As an individual By attending a well-respected university, joining a well-known company, or giving a recorded conference.

Page 231 | Highlight

As a company By focusing on a problem that's immediately attractive to software engineers, finding an attractive way to approach that problem, or retaining prestigious employees.

Page 233 | Highlight

Are you able to hire senior candidates to work in your organization?

Page 234 | Highlight

Does your team seek you out for career advice and advice beyond the immediate scope of their current work?

Page 234 | Highlight

Are you able to start the interview process for jobs you're interested in?

Page 234 | Highlight

Is an executive recruiter able to match you with interesting roles?

Page 234 | Highlight

Is your network expanding by default, allowing you to reach out further and to more senior individuals?

Page 235 | Highlight

Prestige is an ambient, positive familiarity. This doesn't require an organizational program or a strict content calendar; rather, it depends on building awareness of a small amount of noteworthy accomplishments.

Page 235 | Highlight

doing a small amount of writing or public speaking, and then ensuring that work is discoverable.

Page 235 | Highlight

I argue that effective onboarding is the key constraint in hypergrowth companies rather than hiring.

Page 236 | Highlight

ten people committing to share your article online around the same time is a fine distribution plan.

Page 238 | Highlight

prestige. I recommend making a small, timeboxed investment, tracking how often content comes up in hiring processes (both as a candidate and as a hirer), and avoiding spending any more time on measurement.

Page 239 | Highlight

it's much easier to drive pageviews by being controversial than by being thoughtful, but controversy reduces prestige rather than enhancing it.

Page 242 | Highlight

When a new Engineering executive is hired, they are usually brought in by a CEO who believes that the Engineering function is underperforming.

Page 243 | Highlight

Understanding whether you're supported, tolerated, or resented Navigating the implicit power dynamics Bridging narratives across the CEO, board, peers, and your function Avoiding anchoring to previous experience Fostering an alignment habit Focusing on a small number of changes at a given time Embracing transient conflict and preventing persistent conflict Surviving a panicking peer executive

Page 243 | Highlight

It may sound obvious, but the most effective executives are proactively supported by their peers, the CEO, and their functional team.

Page 244 | Highlight

Supported This is when others proactively go out of their way to make your efforts successful.

Page 244 | Highlight

Tolerated This is when others are indifferent to your work.

Page 244 | Highlight

Resented This is when others view your requests as a distraction from their actual work.

Page 245 | Highlight

CEOs hire new executives when there is a problem to solve, which almost always means they've been telling you all about Engineering's problems, but probably not about everything that's going well, and their diagnosis of the problem may have some gaps.

Page 245 | Highlight

Just because Engineering is having problems doesn't mean Engineering is the problem,

Page 245 | Highlight

prone to treating the top-level symptoms as if they are the underlying causes,

Page 247 | Highlight

Many new executives accidentally alienate their peers and functions by assuming their new company will work the same way that their old company worked.

Page 248 | Highlight

You should instead build a habit of inviting feedback and making sure you receive it well. This might look like reaching out to participants in an executive meeting to ask if there's anything you could have done better or should not have done at all.

Page 248 | Highlight

As long as your previous ideas have been effective, then folks will generally be glad to support your next batch.

Page 249 | Highlight

is to focus on delivering a small number of changes with meaningful impact.

Page 249 | Highlight

Conflict isn't inherently negative: experiencing new kinds of conflict is an important sign of growth.

The sort of conflict that you want to avoid is unresolved, recurring conflict.

Page 250 | Highlight

keep an open mind on whether you are the person preventing conflict from resolving.

Page 251 | Highlight

Structured Escalations

Page 251 | Highlight

Most companies have unstructured, undocumented escalation processes that can make escalating feel awkward. But LinkedIn developed a structured escalation process to encourage escalations that build trust. LinkedIn's process is as follows: If you identify a disagreement, agree with the counterparty (e.g., whoever you disagree with) that you'll resolve it in the next five days. Prioritize time with the counterparty to understand each other's perspective and try to agree on a path forward together. If you cannot, then perform a "clean escalation" where you both escalate together with a shared document, rather than escalating separately and losing important context. Commit to following the direction from whomever you both jointly escalated to.

Page 252 | Highlight

This is particularly messy to deal with as a newly hired executive, who is trying to build a relationship with a peer who's more focused on short-term survival than building a long-term relationship with you.

Page 252 | Highlight

it's usually less disruptive for you and the company to improve that executive's execution than to find and hire someone new.

Page 254 | Highlight

it's much harder to be a good teammate to your peers than to your direct reports.

Page 254 | Highlight

In this chapter, we'll discuss: Debugging the Engineering leadership team after stepping into a new role
Gelling your leadership into an effective team
What to expect from your direct reports in that leadership team
Diagnosing conflict within your team

Page 255 | Highlight

developing your point of view on your functional leadership team.

Page 255 | Highlight

Are there members of the team who need to move on immediately?

Page 255 | Highlight

Rather than underperforming, these are individuals actively causing damage.

Page 256 | Highlight

Are there broken relationship pairs within your leadership team or among members of your team and their key stakeholders?

Page 256 | Highlight

Does your current organizational structure bring the right leaders into your leadership team?

Page 256 | Highlight

Once you have a first opinion, spend time trying to disprove it.

Page 257 | Highlight

Start by moving out the individuals who won't be part of the team going forward. Each time you change members of your leadership team, you will have to gel that team afresh, so it's important to quickly put together a plausible leadership team.

Page 257 | Highlight

establishing clear expectations sends the message that you're here to help them perform.

Page 258 | Highlight

When it's possible, it's well worth your time to consider having one or two senior engineers to report to you, as opposed to only managing Engineering managers.

Page 258 | Highlight

work to establish a group of senior engineers that you interact with frequently to ensure you're hearing their perspectives.

Page 258 | Highlight

Operating your team effectively means boiling down to four themes: Define team values. How do you want this team to make decisions? How much should they focus on enforcing consistent policy versus solving escalations?

Page 259 | Highlight

The most important values to establish are those that explain navigating conflict within the team

Page 259 | Highlight

Establish team structure.

Page 259 | Highlight

Find space to interact as individuals. Many senior leadership teams become so transactional that they feel impersonal.

Page 260 | Highlight

Your role as the team's manager is to create some unstructured space to interact as informal, unstructured humans.

Page 260 | Highlight

Referee defection from values.

Page 260 | Highlight

Usually this can be traced back to one or two members of the team violating the team's values without consequence.

Page 260 | Highlight

A central part of creating your team's structure is setting expectations for how members participate with the team.

Page 261 | Highlight

Leading their team.

Page 261 | Highlight

It's essential that your leadership team understands however you and your company will judge them. Communicating with peers.

Page 261 | Highlight

If you don't tell your team what you expect, they'll default to their natural state, which probably won't be what you want. Staying aligned with peers.

Page 262 | Highlight

Should they escalate together, or is it more important to escalate quickly, even if it's a bit messy?
Creating their own leadership team.

Page 262 | Highlight

Learning to navigate you, their executive, effectively.

Page 262 | Highlight

“personal READMEs,”

Page 262 | Highlight

your team will spend a significant amount of energy figuring out how to work with you, and you should do what you can to make it easy for them. Are

Page 263 | Highlight

Competition Amongst Peers

Page 263 | Highlight

collaboration has slowly faded into internal competition.

Page 263 | Highlight

your team believe they'll be rewarded more for capturing capacity from each other than creating capacity for the organization overall.

Page 263 | Highlight

A perceived lack of opportunity The application of poor habits learned in the context of shrinking or bureaucratic companies The failure of a leader to referee their team

Page 264 | Highlight

For senior leaders, I recommend giving the problem back to them: the next step in their career is being an executive, and certainly no one is going to tell them how to advance their scope at that point. Particularly, encourage them to look at opportunities to grow themselves and their career in ways that aren't zero-sum.

Page 264 | Highlight

Your goal is to convince them that today's capacity represents a small fraction of the future

capacity.

Page 266 | Highlight

In most of my roles, I've learned more from my peers than from my manager.

Page 266 | Highlight

spent time building a learning circle of industry peers, which was fundamental to my success.

Page 267 | Highlight

Here are a few of the ways that I've leveraged my network over the past several years: To collect data points on the size of platform engineering teams at different companies, to better benchmark the size of my own platform engineering team. To understand pricing when negotiating renewals for our highest-cost vendors. To solicit advice on navigating complex situations within my executive team, which I'd prefer not to discuss with anyone within my company. To calibrate job offers when there's a repeated gap between compensation data and candidates. For example, has the market moved or are these candidates getting unusual offers? To decide whether to move on from a job, including understanding the state of the job market for executive roles. To gather backchannel feedback on senior leadership hires who got mixed feedback during our interview process when we had a short timeframe to extend an offer due to an outstanding offer from another company.

Page 269 | Highlight

I highly recommend going into networking with the mindset that it's going to take some work, and that you're going to need to be useful to the folks who will network with you.

Page 270 | Highlight

Let's take a look at some specific tactics for connecting with the people you're most interested in. Working Together By far the best way to build your network is to spend time working at a large, fast-growing company in a central tech hub like San Francisco, New York, or London.

Page 270 | Highlight

Cold Outreach The quickest way to build out your network is to send out quick, concise notes to folks near, but not quite in, your network (e.g., someone you don't know but with whom you have a mutual friend). This works because people want to be helpful, as long as you make it easy.

Page 271 | Highlight

Engineering executives learning circle.

Page 272 | Highlight

One way to build your network is by writing and speaking in public.

Page 272 | Highlight

I generally believe that creating content is a time-inefficient way to build your network,

Page 272 | Highlight

There are a number of very large engineering leadership communities out there, most notably the Rands Leadership Slack.

Page 273 | Highlight

They are too open for a genuine discussion.

Page 273 | Highlight

There is an inverse relationship between ease of access and quality of advice.

Page 274 | Highlight

Three patterns that are particularly unlikely to work well for you are: Ambiguous requests, particularly for time ("could we grab some time to chat?"),

Page 274 | Highlight

Confusing requests that force the recipient to decode the message.

Page 274 | Highlight

Asking for introductions or connections when there is not clear, real mutual value.

Page 275 | Highlight

In addition to building your network of engineering leaders, there are a few other networks worth investing in as well: founders, venture capitalists, and executive recruiters.

Page 275 | Highlight

Founders Founders are a powerful, oddly shaped resource.

Page 275 | Highlight

If you're struggling with your own founders, other founders are particularly helpful to chat with.

Page 275 | Highlight

Venture capitalists are the most connected people and best pattern-matchers in the industry.

Page 277 | Highlight

To develop your network, set a small goal, like meeting one new person each month, and slowly build your network up over time with a focus on the challenges you're working on at that moment. Don't make it your top priority, but don't forget it either.

Page 278 | Highlight

onboarding new executives tends to be an ad hoc, chaotic affair because it's used too infrequently to ever become excellent.

Page 278 | Highlight

every executive and role is unique, and it's hard to create a repeatable program to handle one-of-a-kind onboardings.

Page 278 | Highlight

Your job as a peer executive is to help the new executive succeed, even if no one tells you it's your job.

Page 279 | Highlight

It's unfortunately true that you can be perceived as a high-achieving executive without accomplishing much.

Page 279 | Highlight

but you'll only be exceptional if every part of the executive team functions well.

Page 279 | Highlight

Without a good onboarding process, even strong executives will make missteps, sometimes unrecoverable ones.

Page 280 | Highlight

When onboarding a peer software engineer, your goal is to help them understand your current process and one specific onboarding project well enough to implement that one project.

Page 280 | Highlight

about existing processes before ramping up on them. When onboarding a peer executive, your goal is to help them understand the company's landscape (business, team, project, and process), with a particular focus on the most critical, current issues.

Page 281 | Highlight

bring a positive energy (even if things are an absolute mess). Bad signs are inaction on key issues, changing areas that are already working well before addressing higher priorities, or immediately falling back onto previous companies' approaches that fit poorly in their new circumstances.

Page 282 | Highlight

Where can the new executive find real data to inform themselves, rather than relying on existing narratives?

Page 282 | Highlight

What are the top two to three problems that they should immediately spend time fixing?

Page 282 | Highlight

What is your advice to them regarding additional budget and headcount requests?

Page 283 | Highlight

What are areas in which many companies struggle, but that are currently going well here?

Page 283 | Highlight

What is your honest but optimistic read on their new team?

Page 284 | Highlight

the head of Design moved from an embedded model to a centralized studio,

Page 284 | Highlight

Who do they need to spend time with to understand the current state of the company and its implicit power structure?

Page 284 | Highlight

| What is going to surprise them?

Page 284 | Highlight

| What are the key company processes?

Page 285 | Highlight

| One of the magical moments you'll experience as a senior leader is the first time you get support from an executive assistant.

Page 285 | Highlight

| There's a real learning curve to partnering effectively with executive assistants.

Page 286 | Highlight

| Managing time

Page 286 | Highlight

| Drafting communications

Page 287 | Highlight

| Coordinating recurring meetings

Page 287 | Highlight

| Planning off-site sessions

Page 287 | Highlight

| Coordinating all-hands meetings

Page 288 | Highlight

The three questions you should work together on are: What are your respective roles?

Page 289 | Highlight

How do you handle public conflict?

Page 289 | Highlight

What is the escalation process for those times when you disagree?

Page 289 | Highlight

When people complain about politics, often it's a subtle disagreement about an implicit escalation contract: you can just make it explicit instead.

Page 290 | Highlight

Spend some time knowing them as a person.

Page 290 | Highlight

Hold a weekly hour-long 1:1, structured around a shared 1:1 document.

Page 291 | Highlight

Identify the meetings where your organizations partner together to resolve prioritization and work through any conflict.

Page 291 | Highlight

Mediocre but very nice executives last much longer, often remaining indefinitely.

Page 295 | Highlight

When you join a company as an executive, you join with a large reservoir of trust from others based

Page 295 | Highlight Continued

on your having successfully navigated the hiring process.

Page 295 | Highlight

That initial stockpile of trust doesn't go particularly far if you start burning it immediately.

Page 296 | Highlight

Their argument is that it's empowering to give you room to make mistakes while ramping up in your new role; to do otherwise would be stifling you when you need support the most. They'll keep talking about empowerment as you thrash around your new organization, lose trust with your peers, and eventually dig yourself into a hole that's too deep to dig out of. A

Page 296 | Highlight

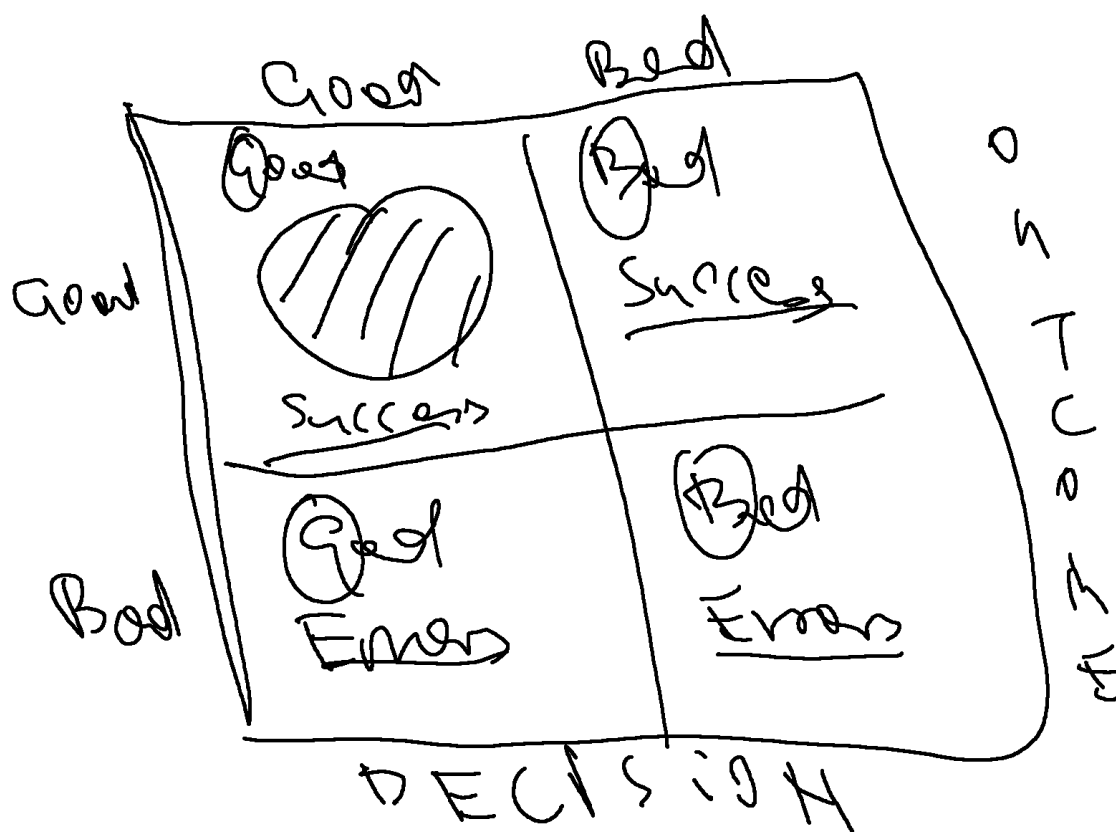
trust on its own isn't much of a management technique.

Page 297 | Highlight

Good errors Good process, good decision, bad outcome (e.g., missing a project's release date because a vendor had an outage that day) Bad errors Bad process, bad decision, bad outcome (e.g., missing a project's release date because you shifted unfamiliar engineers onto the project and predictably slowed it down) Good successes Good process, good decision, good outcome (e.g., hitting

Page 297 | Highlight

release date through good planning and execution) Bad successes Bad process, bad decision, good outcome (e.g., hitting a project's release date because a vendor unexpectedly released functionality that the project was missing)



Page 297 | Highlight

Inspected Trust Is Better

Page 298 | Highlight

“trust first, then investigate before acting” approach.

Page 299 | Highlight

Earlier in my career, I felt bad when folks viewed my inspection as a lack of trust, but now I view that concern as an indicator of a deeper relationship issue between me and the individual I am working with. Your goal in working together is to help each other succeed, and that includes catching each other’s mistakes.

Page 300 | Highlight

The default inspection strategy for established companies is setting and tracking goals, typically in a weekly or monthly metric review forum.

Page 300 | Highlight

After getting a sense of a problem, drill deeply into the problem by talking to as many folks involved as possible.

Page 301 | Highlight

Many senior leaders rely exclusively on heavily prepared data to understand challenges, but prepared data includes the bias of its preparers. Instead, accumulate a small set of data sources whose preparation you understand in detail

Page 301 | Highlight

Some leaders have a deep fundamental intolerance for confusion and misalignment. They'll hear a slight bit of confusion in a meeting and keep digging on it until they understand what caused it.

Page 302 | Highlight

Don't expand everywhere all at once; instead, focus on one or two critical problems.

Page 302 | Highlight

new project-oriented team structure rather than the expertise-oriented team structure.

Page 302 | Highlight

Figure out your peer executives' tolerance for inspection forums before trying to roll them out.

Page 303 | Highlight

Explain to your Engineering leadership team what you're doing and your thinking behind it. Being explicit about your rationale will go a long way, so tell them that your inspection is a long-term

investment in their success, not a demonstration that you don't trust their work.

Page 304 | Highlight

If there are members of your team who are so uncomfortable about making errors that they can't receive feedback, then they aren't the right leaders for your organization.

Page 304 | Highlight

shift away from relying exclusively on unvalidated trust, and instead build more robust trust with your team by pairing trust with inspection to detect errors early and learn from those errors.

Page 305 | Highlight

Managing teams has taught me a lot about my own behaviors and motivations.

Page 305 | Highlight

This culminated in joining Stripe with the explicit goal of pacing myself to be more valuable after four years, rather than getting exhausted after two years like I had at Uber.

Page 305 | Highlight

high standards, generally considered an obviously good thing, frequently cause friction for executives

Page 305 | Highlight

important to match your standards with the standards that your organization upholds, rather than assuming that high standards are obviously right

Page 305 | Highlight

escalating in regard to perceived low standards can get you in conflict with your CEO

Page 305 | Highlight

role modeling is an effective tool for shifting an organization's standards What to do when your standards are not serving you well in your current situation

Page 306 | Highlight

The socially acceptable vice that I've seen harm the most executive careers is having higher standards for those around them than their organization supports.

Page 307 | Highlight

The main character is trying to do their job effectively, but can't because of the low performance of a peer. They escalate to the appropriate manager to address the issue, but that manager decides to view the performance issue instead as a relationship issue: it's not that the peer isn't performing, it's just that the two of you don't like each other.

Page 308 | Highlight

you're actually doing is trying to hold the CEO accountable for not solving a known problem, which they may not appreciate.

Page 308 | Highlight

they simply believe the given problem merits a different quality standard than you do.

Page 309 | Highlight

If you're struggling with your peers not meeting your standards, it's important to recognize that you're disagreeing with how your CEO is doing their job.

Page 309 | Highlight

it is unfair to be forced to overcome a peer's poor performance—that's the manager's job—but work isn't a perfect place.

Page 309 | Highlight

| Grinding yourself down in frustration about the paths that don't exist doesn't solve anything.

Page 310 | Highlight

| a playbook that I call "Model, Document, Share".

Page 310 | Highlight

| Model Identify the area you want to improve and go about role modeling a high-standards approach.

Page 310 | Highlight

| Document After you've gotten the approach working, write it up into a clear document.

Page 311 | Highlight

| Share Send the documented approach to the teams you want to influence, letting them know what you've tried, what you learned, and why it was an effective approach for your team.

Page 311 | Highlight

| a useful technique for trying to make progress in areas where authority figures have decided not to make improvements, without alienating those authority figures.

Page 311 | Highlight

| "lessons not worth learning."

Page 312 | Highlight

| some lessons aren't worth learning, I've also come to appreciate that some penalties aren't worth paying.

Page 313 | Highlight

Uber ran a tech spec review process called the DUCK Review.

Page 313 | Highlight

a good process is a deliberate trade-off between quality and overhead.

Page 314 | Highlight

There's no right pattern, just the most appropriate pattern for your current constraints. (However, some patterns do seem to go wrong more often than not.) Early Startup

Page 315 | Highlight

the Early Startup pattern: either a founder or the functional executive does something for their function, or it doesn't happen.

Page 315 | Highlight

The next phase is the Baseline pattern, which is first adopted around 50 hires and is the final pattern for most companies. Company-scoped processes (e.g., company planning, performance management, and budgeting) are run by a central People (or Human Resources) function and Engineering-scoped processes (e.g., the tech spec review, the incident review, and the developer productivity survey) are run by engineers and Engineering managers.

Page 315 | Highlight

As companies grow beyond 200 engineers, Engineering-scoped processes often start to take enough time that they benefit from full-time support.

Page 315 | Highlight

the Specialized Engineering Roles pattern because most companies address this problem by spinning up a new function—usually either Technical Program Management or Engineering Operations.

Page 316 | Highlight

Ensuring that processes work properly despite all these extended requirements often leads to Engineering splitting away from the company's default mechanisms for hiring, performance management, and so on, such that there are dedicated recruiters and members of the People team working with Engineering. I call this the Company Embedded Roles pattern.

Page 316 | Highlight

There is a dual-sided career ladder across engineers and Engineering

Page 316 | Highlight

There are often one or more specialized roles along the lines of Developer Relations, Technical Program Management, Quality Engineering, or Security Engineering. Engineers and Engineering managers tend to be slower and more expensive to hire and retain than most other functions. A long ramp-up time, from starting to becoming highly productive, places a higher value on retention.

Page 317 | Highlight

the Business Unit Local pattern, where Engineering reports into each business unit's leadership and there is no longer any one leader who is responsible for the entirety of Engineering.

Page 318 | Highlight

Cons: Often valuable stuff doesn't happen because everyone is too busy; sometimes the quality of a process is very low due to founders or managers having had limited time or experience running that process.

Page 319 | Highlight

Cons: Outcomes are highly dependent on the quality of these centralized functions; changing company-scoped processes becomes challenging.

Page 319 | Highlight

organizational planning, incident management, and engineering onboarding.

Page 319 | Highlight

Cons: While having specialized engineering roles is more efficient, it's usually more expensive as these roles introduce additional headcount. It's a lot of work to effectively support specialized roles like Engineering Operations or Technical Program Management teams. Specialized roles often freeze a company in a given way of working. Whereas engineers are incentivized to eliminate a process, the specialist is incentivized to improve it.

Page 320 | Highlight

Cons: Expensive to operate and quality is heavily dependent on the embedded individuals.

Page 320 | Highlight

Sometimes this pattern is adopted but one business unit is so much larger than the others, it becomes the de facto Engineering hub for the wider company, including leading cross-organizational developer tooling and infrastructure investments.

Page 321 | Highlight

me to pick one without any additional context, I would always recommend the Baseline pattern.

Page 321 | Highlight

What I've called the Baseline pattern is the most common approach to running processes, with a centralized People organization running company-scoped processes and a mix of engineers and Engineering managers running the Engineering-scoped processes.

Page 322 | Highlight

most budgeting discussions are cost-driven rather than effectiveness-driven.

Page 322 | Highlight

Rotate the work. Calm has had great success with six-month rotations, where individuals serve in one capacity (e.g., running incident review) for six months before swapping off. We increase continuity by having two people serve together in each capacity with overlapping but offset periods of service. Make the process interesting for the people running it.

Page 323 | Highlight

strongly preferring that engineers serve a term in one of your processes before moving into Staff-plus or managerial roles.

Page 324 | Highlight

most budgeting processes are anchored to relative dollars rather than absolute impact.

Page 324 | Highlight

Most companies that successfully shift up the patterns' cost curve do so when they are rapidly growing. Such companies are comfortable with rapidly growing headcount costs and are generally tolerant of adding roles to support that growth.

Page 324 | Highlight

Many companies operate to a portfolio allocation target across departments.

Page 324 | Highlight

A surprising number of budgeting processes operate in headcount rather than budget.

Page 326 | Highlight

My advice is to design your organization around processes grounded in one specific set of foundational beliefs and stick solid to those beliefs for at least three to four years before changing them again.

Page 328 | Highlight

you'll spend more and more time designing and debugging your overall interview process.

Page 328 | Highlight

Establishing your overall hiring process, including job descriptions, rubrics, and hiring loops

Page 329 | Highlight

Applicant Tracking Systems (ATS) A good ATS is the core mechanism for coordinating your interviewing process.

Page 330 | Highlight

Interview definition and rubric

Page 330 | Highlight

Interview loop documentation

Page 330 | Highlight

Leveling framework

Page 330 | Highlight

Hiring role definitions

Page 331 | Highlight

Job description template

Page 331 | Highlight

Job description library

Page 331 | Highlight

Hiring manager and interviewer training

Page 331 | Highlight

The most effective process I have seen is having new interviewers shadow several trained interviewers, combined with one reverse-shadow interview where an experienced interviewer

| shadows (and gives feedback to) the new interviewer.

Page 332 | Highlight

| The three clearest indications that you've over-optimized your hiring process are: Recruiters hire fewer than five engineers per recruiter per quarter (excluding the scenario where they are constrained by headcount). You frequently need new interview loops or new interview questions. It routinely takes more than two weeks from a candidate's first interview to you making an offer to that candidate.

Page 334 | Highlight

| Once you've built the hiring process, your job as an executive is generally to monitor and debug it, rather than serve within it.

Page 334 | Highlight

| Include Recruiting in your weekly team meeting

Page 334 | Highlight

| Conduct a hiring review meeting

Page 335 | Highlight

| Maintain visibility into hiring approval

Note:

input	Wlen	offe	deeph
0000 0000 00	000 0	000	0
000 00	00	0	
000 000 000	000 0	00	0
000 000 0	00 00	00 0	

Approve out-of-band compensation

Review monthly hiring statistics

Page 335 | Highlight

It's particularly helpful to understand throughput (hires per recruiter), time to hire, offer rate, and acceptance rate.

Page 336 | Highlight

While it's unscalable to remain as an interviewer across all loops, it is particularly valuable to remain engaged in helping close senior candidates.

Page 337 | Highlight

Leveling Candidates

Page 338 | Highlight

I recommend provisionally leveling candidates before they start the bulk of your interview process.

Page 338 | Highlight

I recommend relying on a simple provisional leveling heuristic, such as a combination of technical phone screen performance and years of prior experience.

Page 338 | Highlight

My experience is that committees appear less biased, but generally introduce a bias of their own, and are less efficient than wholly accountable individuals.

Page 341 | Highlight

The biggest trap to avoid is prioritizing recruiters based on hiring needs. This often steers all recruiting capacity toward your least-effective hiring managers.

Page 342 | Highlight

There's a handful of particularly common hiring problems that are usually due to untrained or

inexperienced hiring managers. These problems include: Demanding unrealistic, unicorn candidates from the recruiting team, such that you never make offers. Allowing any concern from any interviewer to block a candidate from getting an offer, such that you never make offers. Pushing for non-standard compensation on every candidate rather than learning to close candidates within the standard compensation bands. Being indecisive on candidates and asking for additional interviews until the candidate withdraws from the process. Refusing to talk to candidates early in the process and then blaming the recruiting team that senior candidates aren't interested in finishing the process.

Page 343 | Highlight

Effective hiring processes hire candidates. Hiring managers are accountable for their hiring process. Any argument to the contrary is a flawed argument.

Page 344 | Highlight

hiring. Hiring exclusively from your network will convince your existing team that they and their networks aren't wanted for important roles at your company.

Page 344 | Highlight

the pattern of prioritizing external hires will encourage your team to seek career advancement elsewhere, draining your organization of context and continuity.

Page 344 | Highlight

because of a missing hiring skill that's either of the following: An inability to hire outside their network (or within it), or An inability to fairly evaluate an internal candidate (or external candidate)

Page 345 | Highlight

It's very common for these programs to focus entirely on representation of newly hired employees, but it's at least equally important to track retention and promotion of the employees you've already hired. Indeed,

Page 346 | Highlight

and this poor outcome is driven exclusively by inability to retain diverse talent.

Page 346 | Highlight

systems where the Recruiting team is accountable for diversity, rather than Engineering management.

Page 347 | Highlight

Many companies introduce centralized (Engineering-scoped) or semi-centralized (Product Engineering or Infrastructure Engineering-scoped) hiring committees as part of maintaining a consistent hiring process.

Page 347 | Highlight

Hiring committees are a useful tool, but I'd caution against introducing them as the obvious solution. They're useful, but they come with their own problems. I generally dislike committees as they introduce ambiguity around who should be held accountable for outcomes.

Page 348 | Highlight

On the positive side, hiring committees are a great mechanism for training hiring managers' judgment on what makes a good candidate.

Page 348 | Highlight

Committees are slower than a responsive hiring manager, but faster than a disengaged or very busy hiring manager.

Page 349 | Highlight

You should try to solve your problem within the process you've designed, but don't get so blinded by your process that you think the process is always more important than your problem at hand.

Page 351 | Highlight

Most companies say that it takes three to six months for newly hired engineers to fully ramp up.

Page 351 | Highlight

Running Engineering onboarding means optimizing two closely related goals: Increasing the percentage of engineers who are reasonably productive in their first three months Setting the foundation that supports engineers to become more extremely impactful within a few years from now

Page 351 | Highlight

effective onboarding is the highest-value investment you can make into Engineering productivity, but somehow it's often dismissed as a secondary concern.

Page 352 | Highlight

Examples of Engineering onboarding programs are: A startup with 30 engineers

Page 353 | Highlight

It's a bit messy, but a surprisingly large number of startups have effectively no onboarding beyond showing up, getting a laptop, and getting pointed to teammates who can help. A startup with 150 engineers New hires at Stripe spent their first day focused on laptop setup, security practices, and getting familiar with development tools by making a small commit to the product website (eventually adding oneself to the company website's About page). New Engineering hires were then run through a curriculum ranging from architecture to observability. The final step of Engineering onboarding grouped new hires into teams of four to six engineers. Each team worked on a project together, with one existing Stripe engineer serving as the project's tech lead. These projects generally lasted one to four weeks and were completed before releasing the hires to join their long-term teams. A company with thousands of engineers Facebook runs a six-week Engineering Bootcamp attended by every new Engineering and Engineering management hire.

Page 354 | Highlight

A structured onboarding process comes down to staffing a handful of roles (especially your role as the executive sponsor and someone to orchestrate onboarding) and deciding on the specific curriculum to include in your program.

Page 355 | Highlight

the executive sponsor, program orchestrator, team manager, and onboarding buddy.

Page 355 | Highlight

Select the program orchestrator to operate and evolve the program. Align with the orchestrator on how to monitor the program over time (certainly through surveys, but ideally through some objective data, too). Monitor the program data on a monthly basis (e.g., look for quantitative feedback from session attendees, or the number of sessions given). Meet directly with new hires who have exceptionally good or bad onboarding experiences to understand where things might be falling through the cracks. Celebrate the program on an ongoing basis so that people continue to prioritize its success; this includes supporting onboarding work as a valid contribution toward being promoted.

Page 356 | Highlight

it's the program orchestrator who translates those resources into a living program. Most frequently, this is an Engineering manager who runs onboarding in addition to other responsibilities, but at a scaled company this might well be someone in Engineering operations or a technical program manager (covered in Chapter 19).

Page 357 | Highlight

Selecting an effective program orchestrator is the single most important decision you'll make for an effective onboarding program, and you'll make it over and over again as folks shift out of this role.

Page 357 | Highlight

Being the onboarding orchestrator is generally a part-time role until a company gets rather large (20 or more engineers joining per month), but it's not a small task. Whoever runs this program should be able to devote at least 20 hours a month to its success. An Engineering manager could take this on and manage a team, but it wouldn't leave them time for much else.

Page 358 | Highlight

Create a personalized onboarding document for each new hire, starting from a template created by the program orchestrator. Start weekly 1:1s with the new hire.

Page 358 | Highlight

Assign an onboarding buddy to the new hire.

Page 358 | Highlight

Partner with the onboarding buddy to identify a new hire's first several projects.

Page 358 | Highlight

Make sure the new hire has actually been added to the various team meetings, chat rooms, email groups, and so on.

Page 359 | Highlight

The buddy can answer specific questions from the new hire such as writing software on their team, setting up the development environment, and working through their initial project. A buddy is also a fallback for when the new hire's manager might be unavailable or slower to respond.

Page 359 | Highlight

Bring the new hire along with you to meetings, to grab lunch, and to meet other members of the company. Don't let them linger alone!

Page 360 | Highlight

Schedule daily time with the new hire for their first several weeks. This can be 15 or 30 minutes with some time blocked afterward in case it runs long. The goal is to resolve any blocking issues or topics for the new hire.

Page 360 | Highlight

After one to two weeks, chat with the new hire about whether it's helpful to continue the meetings for another month or two.

Page 360 | Highlight

of deciding on your curriculum—what should you actually focus on teaching?

Page 360 | Highlight

three segments: Engineering values and strategy Technical architecture Spinning up the development environment

Page 360 | Highlight

Survey new hires a month after onboarding to ask what they wish had been covered. Survey managers and tech leads about areas where new hires are struggling. These might be technical but are even more likely to be cultural.

Page 361 | Highlight

whether to include a project component in your onboarding program, like Facebook's programming components of Bootcamp or Stripe's onboarding project teams.

Page 361 | Highlight

these projects work very well if you put a lot of energy into them, and otherwise don't go particularly well.

Page 361 | Highlight

the alternative to running an onboarding-driven project is that teams select projects for their new hires, which usually go well and don't require centralized coordination.

Page 361 | Highlight

one implicit topic in your onboarding curriculum is getting to know the people who joined the company around the same time.

Page 361 | Highlight

My recommendation is to require all engineers and all Engineering managers to attend, regardless of level.

Page 362 | Highlight

most programs fail because someone—with extremely good intentions—creates a complex, involved program.

Page 363 | Highlight

When you consider revising your onboarding process, make sure it's something the orchestrator can run for the long term.

Page 363 | Highlight

The worst onboarding programs come from trying too hard, rather than not trying hard enough.

Page 364 | Highlight

you generally should continue running onboarding from within Engineering.

Page 364 | Highlight

the people who replace the program orchestrator will be less focused on effective Engineering onboarding and will not have experience operating as an engineer at your company.

Page 364 | Highlight

Once another function (such as HR) is responsible for running your onboarding process, experimenting with improvements will become much more difficult,

Page 365 | Highlight

executive leadership is about taking the best available path, not dying on every sword that presents itself.

Page 365 | Highlight

Onboarding is particularly impactful when you're hiring a solid chunk of new people every month.

Page 365 | Highlight

The only case where I'd recommend against investing in onboarding is when you're hiring so infrequently that onboarding materials go out-of-date before they're used a second time.

Page 367 | Highlight

Uber's original performance review process was called "T3B3" and was remarkably simple: write the individual's top three strengths and top three weaknesses, and share the feedback with them directly in person.

Page 368 | Highlight

The conflicting goals between those designing, operating, and participating in performance and compensation processes
How to run performance review processes, including calibrations, and how to navigate their challenges
How to participate in a compensation review process effectively
How often you should run performance and compensation cycles
Why your goal should be to run an effective process rather than a perfect one

Page 368 | Highlight

Uber's T3B3 performance process, which identified an employee's top and bottom three areas for a given six-month period, what was most remarkable was its radical simplicity. It was focused exclusively on providing useful feedback to the recipient.

Page 368 | Highlight

A typical process at a given company tries to balance all of these distinct stakeholders:
Individuals They want to get useful feedback so they can grow. They want to get promoted and maximize their compensation, and they

Page 369 | Highlight

Managers They want to provide fair and useful feedback to their team and promote their team as

| appropriate

Page 369 | Highlight

| People (or Human Resources) team They want to ensure individuals receive valuable feedback and create a “floor” for the quality of that feedback. They want to document feedback so that it can be used in performance management and legal scenarios later, to support the company’s perspective.

Page 369 | Highlight

| Executives They need to decide who to promote based on inconsistent evaluations across managers.

Page 370 | Highlight

| Feedback Sources The baseline performance process requires each manager to provide written feedback for each of their direct reports, including a decision on whether to promote them, but there are quite a few details and variations to consider. The first variations to consider are whether to include peer and upward feedback (on each individual’s manager).

Page 371 | Highlight

| Peer feedback can take up a significant amount of time, particularly for highly connected folks who may be asked to provide peer feedback on 10 or more individuals.

Page 371 | Highlight

| More importantly, my experience is that peer feedback is very inconsistent. I’ve managed teams that feel peer feedback is too uncomfortable to give honestly, and those teams have provided useless peer feedback: in those cases, it’s not worth collecting peer feedback.

Page 371 | Highlight

| Most companies have paired titles and levels, such as: Entry-level Engineer (Level 3) Software Engineer (Level 4) Senior Software Engineer (Level 5) Staff Software Engineer (Level 6)

Page 372 | Highlight

recommend using the fairly typical progression of Entry-level Software Engineer, Software Engineer, Senior Software Engineer, Staff Software Engineering, and Senior Staff Software Engineer.

Page 372 | Highlight

I'd recommend skipping the hardest part by picking a reasonably good rubric that's available online, creating a working group to tweak the details, and then refining it after every performance cycle to address issues that come up.

Page 373 | Highlight

Prefer concise leveling rubrics over comprehensive ones.

Page 373 | Highlight

many folks are exceptionally good at gaming specific criteria.

Page 373 | Highlight

more focused on preventing bad behavior than providing clear guidance for the well-intentioned.
Prefer broad job families over narrow job families.

Page 374 | Highlight

Capture the how (behavior) in addition to the what (outcomes).

Page 375 | Highlight

another. Most companies rely on managers to make a tentative promotion nomination, then rely on a calibration process to ratify that nomination. Calibration is generally a meeting of managers who talk through each person's tentative rating and promotion decision, with the aim of making consistent decisions across the organization.

Page 377 | Highlight

Demotion with compensation adjustment

Page 377 | Highlight

Demotions without compensation adjustment

Page 378 | Highlight

Title demotion without level adjustment

Page 378 | Highlight

the concept of “constructive dismissal” means that demotions need the same degree of documentation as dismissals.

Page 381 | Highlight

Companies typically build compensation bands by looking at aggregated data acquired from compensation benchmarking companies.

Page 382 | Highlight

Compensation benchmarking is always done against a self-defined peer group.

Page 382 | Highlight

Once you have benchmarks, you’ll generally discuss compensation using the compa-ratio, which expands to “comparative ratio.”

Page 383 | Highlight

Every company has a geographical adjustment component of their compensation bands.

Page 384 | Highlight

For most companies, I recommend a twice-annual process.

Page 387 | Highlight

they're a valuable mechanism for earning credibility with your team by following through to address their concerns.

Page 388 | Highlight

Focus more on absolute scores than on relative scores (a high score is still high, even if it's relatively down a bit). Take the time to identify all issues, rather than getting caught up in the first couple you notice.

Page 388 | Highlight

If your executive team leads the company but isn't trusted to view the whole company's survey results, then something doesn't make sense there and it should be fixed. The executive team may find it uncomfortable to see each other's survey results, but that transparency to one another is part of an executive team gelling together.

Page 389 | Highlight

start to group insights into three categories: things to celebrate, things to proactively address, and things to acknowledge.

Page 391 | Highlight

My goal was to better evaluate the polished senior leaders who always said the right thing. I wanted to find the real beliefs and behaviors underneath all the polish. One interview focused on a direct report sharing a mediocre strategy proposal for review, and getting a signal on whether the candidate could give useful feedback on improving the document. I knew that interview worked when a candidate said that they wouldn't even give feedback because the proposal's quality was below their bar to even discuss. Another interview I created focused on how candidates reviewed cultural survey results.

Page 392 | Highlight

If you identify any acutely serious issues (e.g., a manager whose team is in open revolt, or an ethical issue is raised), take action immediately.

Page 392 | Highlight

select two to three areas you want to invest in until the next survey is run

Page 393 | Highlight

For the areas you want to invest in, make sure you have explicit, verifiable actions to take.

Page 393 | Highlight

Share the document with your organization (via email, chat, or whatnot) and schedule a meeting to discuss the findings and priorities with the entire Engineering organization. I often repurpose the Engineering Q&A sessions (see Chapter 10) for this discussion. Follow up on a monthly cadence on progress against your action items. I often include the updates in my weekly updates to the organization (see Chapter 11). Make sure to mention these improvements when you introduce the next round of cultural surveying.

Page 394 | Highlight

Executives are rarely held accountable unless they expose themselves to accountability and this is a valuable opportunity to hold yourself accountable to your team.

Page 394 | Highlight

I generally think the default questions are good enough, and that it's not worthwhile to change.

Page 395 | Highlight

The first sort of question is expensive to change because each change wipes out any historical context on the question.

Page 396 | Highlight

However, as your organization grows, they become more and more useful. I find Dunbar's number, roughly 150 people, to be a good point to ensure you're running one of these surveys.

Page 398 | Highlight

Having problems is part of being an executive, but when you're that second executive who can't turn off the frustration, it's time to start thinking about leaving. Departing an executive job is much messier than leaving an individual contributor role and will significantly impact the team and company around you.

Page 399 | Highlight

succession planning.

Page 399 | Highlight

In performance reviews, think about what your direct reports are missing to thrive in your role and give them at least one of those areas to focus on in each review. As they make those improvements, talk to the CEO about the growth you're seeing in your team. For the one or two with the best chance of eventually succeeding into your role, facilitate them building a relationship with the CEO. Every quarter, run an audit of the recurring meetings you attend. For each, can you delegate it to someone on your team?

Page 400 | Highlight

Go on at least one long vacation each year (aim for two weeks, if you can) and explicitly delegate your roles across your team rather than slowing things down.

Page 401 | Highlight

For many, being an executive is an important pillar of their identity, and that's hard to walk away from.

Page 401 | Highlight

Has your rate of learning significantly decreased?

Page 401 | Highlight

| Are you consistently de-energized by your work?

Page 401 | Highlight

| Can you authentically close candidates to join your team?

Page 402 | Highlight

| Would it be more damaging to leave in six months than today?

Page 403 | Highlight

| In “Work on what matters”, I wrote about Hunter Walk’s idea of snacking: doing work that is easy to complete but is low impact.

Page 404 | Highlight

| under pressure, most people retreat to their area of highest-perceived historical impact, even if it isn’t very relevant to today’s problems.

Page 406 | Highlight

| If it’s more than two years, you’ll be able to find another role as long as some of your previous roles have been three-plus years. As long as there’s a strong narrative, any duration is long enough. For example, if your company is acquired by a larger company, the narrative is that you weren’t excited being in the larger environment.

Page 406 | Highlight

| You’ll always have more leverage negotiating from an existing role, and if your primary goal is advancing your career, then I would recommend finding your next role before departing your current role.

Page 407 | Highlight

If you are open to changing your mind, then you should be having a career discussion with the CEO, not a departure discussion.

Page 408 | Highlight

Departure timeline

Page 408 | Highlight

What you intend to do next

Page 408 | Highlight

Why you're departing

Page 408 | Highlight

Recommended transition plan

Page 410 | Highlight

A shared description of why you're leaving

Page 410 | Highlight

When each party will be informed of your departure

Page 410 | Highlight

Drafts of emails and announcements to be made to larger groups,

Page 411 | Highlight

The most common mistake I see outgoing executives make is trying too hard to help. At this point, you're a consultant, not an executive.

Page 411 | Highlight

They are usually motivated by a sense of guilt rather than a clear plan to help, and they can undermine the person who replaces you.

Page 414 | Highlight

while you should prepare, read, and study, the most valuable way to become an effective Engineering executive is doing the work.

Page 432 | Highlight

important to understand where a table shifts from historical data to forecast data.

Page 457 | Highlight

The trade-offs and timing between standardizing on what works, exploring for superior technology, and supporting adoption of superior technology are the core of Engineering strategy. An effective approach is to prioritize standardization, and explicitly pursue a bounded number of explorations that are pre-validated to offer a minimum of an order of magnitude improvement over the current standard.

Page 458 | Highlight

Narrow standards simplify the process of writing great documentation, running effective training, providing excellent tooling, etc.

Page 458 | Highlight

Investing into new technologies creates unknowns that the organization must invest into understanding; investing more into the existing standards is at worst neutral.

Page 458 | Highlight

I am unaware of any successful technology company that doesn't rely heavily on a relatively small set of standardized technologies. The combination of high leverage and low risk provided by standardized is fairly unique.

I've been in quite a few iterations of "standardize or explore" strategy discussions, and this is the best articulation I've found so far of what works in practice: mostly standardize, make sure that explorations drive order of magnitude improvement, and limit concurrent explorations.