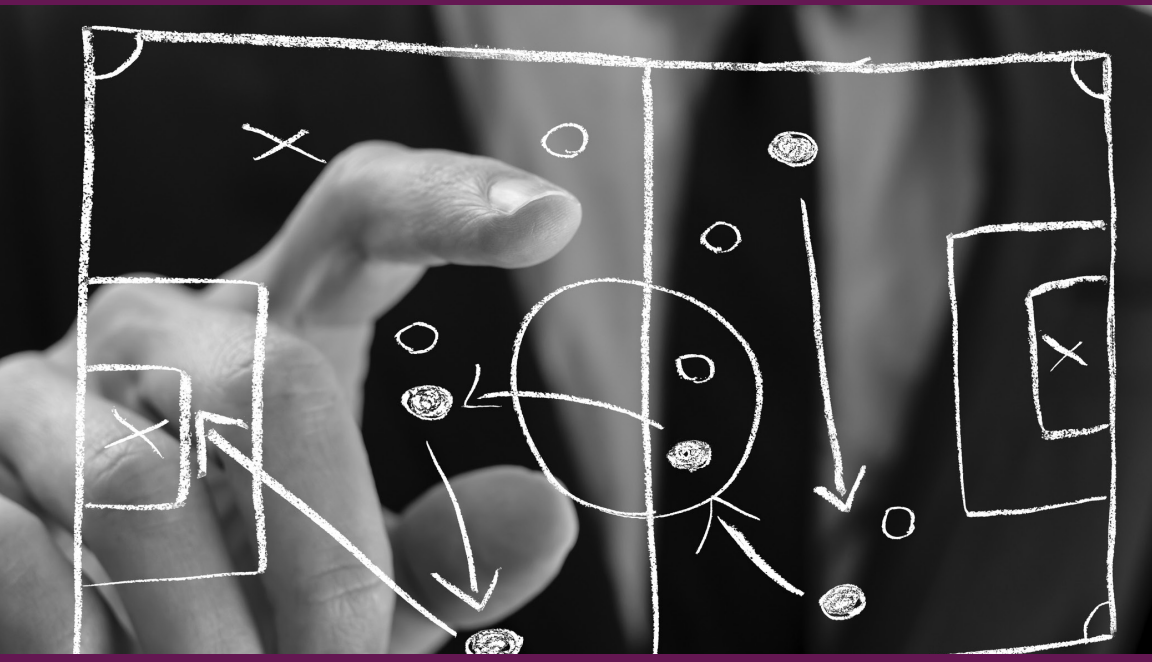
O'REILLY®

The New Manager Mindset

Become a Trusted, Respected & Reputable Leader



Robert Hoekman, Jr.

The New Manager Mindset

Become a Trusted, Respected, and Reputable Leader

Robert Hoekman, Jr.

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

The New Manager Mindset
by Robert Hoekman, Jr.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Laurel Ruma
Production Editor: Nicole Shelby
Copyeditor: Octal Publishing
Interior Designer: David Futato
Cover Designer: Karen Montgomery

June 2016: First Edition

Revision History for the First Edition
2016-05-27: First Release

978-1-491-95976-3

LSI

Table of Contents

1. Introduction **1**
A Note About How I Learned 2

2. The Daily Practices of Leadership **5**
Being Collaborative 7
Steady As You Go 8
Eyes On the Road 9
Speaking Up 11
Institutionalized 11
Snap Out of It 12
Taking Criticism 12
Giving Credit Away 13
Managing Things Away from People 14
Creating Opportunities for Others 15

3. Learning as a Habit **19**
Leaving Your Ego Out of It 20
Learning to Succeed, Not to Embrace Failure 21
Repeated Failure Gets You Nowhere 22

4. Skill Equals Perspective **25**
Upside of Overlap 26
Be Replaceable 27
T-Shaped People 28

5. The What Over the How **31**
Tools, Not Processes 32

Improvising 33
Public Perception 35

6. Saying It with Purpose. 37
Framework for Thinking 38
On Writing Well 39
Enabling Comprehension 41
Mapping Your Message to Their Concerns 41

7. Becoming Convincing. 43
Listening 45
Asking 46
Restating 47
Educating 47
Presenting 48

8. Being Unreasonable. 53
The Advantage of High Standards 54

9. The Philosophy It All Adds Up To. 59

CHAPTER 1
Introduction

So you're a manager now. (Or you'd like to be.)

Odds are, you got here by being good at something other than managing. Odds are, you're a little nervous now about whether you'll be as great at this as you were at that other thing. Odds are, you're already getting tired of platitudes and generic business advice and op-eds about leadership and you're hoping this time things will be different.

It is. It *is* different.

Leadership advice is cheap. It's also loose and vague—just vague enough that you can project whatever you want into it. If you want to hear that it's all going to be okay and that if you just lean in and finally learn to use your project management software, you'll be well on your way to success in no time. You can find someone to say that. Probably a lot of people, in fact. You can find studies to tell you how to deal with Millennials. You can learn how to be a Millennial who's also a manager. You can read up on the latest stats and thinking on employee retention and positive feedback loops and opportunity for growth. You can find a hundred books on all of the aspects of leadership you can imagine, written by every current and former leader you've never heard of. You can do all that, sure.

I don't think you want to do all that.

I think you want to turn off the "Complicator" switch. I think you want to simplify the whole thing and learn how to Zen-out about management and be the kind of manager you wish you had, or like

that one you really admired once a few years back; the one who was really great and who inspired you to become good at whatever it was you were good at before you became a manager yourself. I think you want to develop a mindset that will help you navigate all the situations, not just this one or that one. I don't think you want tactics. I think you want a belief system. One that makes you feel calm, confident, controlled.

I'll say that again: I think you want a *belief system*.

You want a set of guiding principles for good managers. A collection of truths you can latch onto and use as your North Star. You don't want a solution; you want an *approach* to solutions. One you can use all the time, every time.

Okay then. Let's do that.

A Note About How I Learned

Like you, I began my career in subordinate roles and dealt with more than my fair share of bad bosses with remarkably few great ones along the way. Eventually, I worked my way into the Director chair at a good-sized company, where I remained for a while until I'd built up enough consulting work to become self-employed full time. After eight or so years of that, I joined a startup for a few months as a Director, and then returned to consulting work (I didn't want to relocate).

It's a very different thing to be a consultant than to be an in-house manager, I know. Before we get too far, I want to acknowledge that. Whereas I can usually get in, do my work, and get out without getting mired down in the daily life of it all, for example, in-house staffers have to deal with internal politics long term, working with the same people, the same constraints, the same internal issues and roadblocks for months and years. My projects haven't worked that way. When someone has bothered me, I could take comfort in knowing I wouldn't need to deal with that person for very long. On the flip side, internal staff members get to stick around to see the long-range outcomes they've spent so much energy trying to achieve, whereas I'm off on another project. When—if—I've heard about the results of my work a year after the fact, it's because I remembered to get back in touch with the client and ask.

What all this did for me, though, was something that can help you. It put me in a position to watch hundreds, if not thousands, of people wing their way through projects in all kinds of situations, in all kinds of companies, in all kinds of team setups. It put me in a position in which I could study what made the successful ones successful. What made the struggling ones struggle. Who did well, who didn't, why they could or couldn't navigate their way to a good outcome, and what they did, said, and thought that got them there, whether positive, negative, or somewhere between.

It put me in a position to develop an approach to management that could, did, and had to work in every situation.

During my nine combined years as a consultant, most of my work involved serving as the ad hoc strategy consultant and project manager for teams ranging in size from single-person startups to Fortune 500 companies. I did this for dozens of companies on hundreds of projects, temporarily running projects and teams, sometimes for a few days, other times for months on end. Basically, I spent a lot of time being a person who had to walk into unknown situations, take charge of them, and get things done. And I spent a lot of time looking at leadership from all angles—studying the qualities and actions and behaviors that made people effective, trusted, and respected. I've been able to watch these skills play out in every direction.

In client situations, I rarely had more than a day or two for getting to know people, building rapport, earning trust. If I didn't do this well, projects could go wrong. If I did do this well, I had the chance to leave lasting guidelines in place that could shape a product or team for months or years to come.

So over time, I learned to forge a path from the first day I showed up—to position myself as a leader and get others to go with me down that path. I learned to do this well, and I learned to do this fast. I learned how to consistently walk into a situation, no matter how different it might be from the last or all the others before it, and lead it wherever it needed to go.

I had to. My job was to be a leader.

In each and every situation, the approaches I discuss herein have been crucial to success, and not just mine. They are the things I've seen make other people effective, too—the in-house people, whether VPs, PMs, or lowly team members with barely six months under

their belts working just to be heard. These are the qualities and actions and behaviors that have made people successful across the board, regardless of their situations, regardless of their roles.

I offer them here as lessons and approaches for new managers and those who aspire to become managers.

While all of them contribute to being a good leader and to helping you build credibility and trust, a few skills, I think, are also specific to leaders—to those already in leadership roles.

I'm betting you've spotted most of these qualities in people who weren't bosses a few times yourself. I'm betting you've really liked working with them. I'm betting you've wished the people who are your bosses were more like them.

If you were lucky enough to work with one, I'm betting you stayed. If you left, I'm betting it's because you learned so much from this person that you felt it was time to hold yourself to the standard to which that person held you.

If you're reading this now, I'm betting it's because you'd like to learn what made them great. How to become great yourself.

You can do that.

Here's how to get from here to there.

CHAPTER 2
The Daily Practices of Leadership

There are two kinds of managers.

"And only two," said a project manager once, responding to my quip.

A client meeting had just ended. I was finishing up some notes. The client's PM had stayed behind. Someone else on the client team had told a funny story a few minutes earlier about an old boss, and the PM and I were still chuckling over it. He knew what I meant when I said there were only two kinds of managers. He filled in the rest for me.

"The kind who want to control everything, and the kind who hire smart people and get out of the way."

I'd first heard this from an old boss of my own. It stuck with me for a long time, and I sometimes found myself dropping every boss I'd ever had into one box or the other. It wasn't hard to do. I don't think I'd ever thought about it until my old boss had said it. Once I'd heard it, there was no escaping it. Every manager fell to one side or the other.

Not because it was true, but because someone had *said* it was true.

In truth, there are more than just two buckets.

There does seem to be a bucket that can be easily filled with people who want to micromanage everything. You know the type. They dish out strict instructions. They disappear for long periods of time while you work out all the details (there's not enough time to micromanage you *and* everything else). Then they do drive-by manage-

ment to tell you how wrong all your decisions were and how they would've done it all differently. Let's face it; those people suck.

But the ones who manage well? They come in a million varieties and have a million different techniques and traits. And none of them simply "get out of the way." Leaders tend to do a whole bunch of things that make us all want to follow their lead in the first place.

Some of their skills are complicated and take a long time to develop, and they help push the entire organization forward with intelligence and quiet persistence. Some are basic, everyday practices that keep teams united and moving and bonded to each other. Practices that nurture an environment of respect and dignity and possibility on a daily basis.

I want you to want to be the second kind of leader. Your team will want that, too. If you want to be a micromanager, you might as well toss out this report right now. It's not going to do you any good. If you want to be the right kind of leader, stick around.

This report is full of observations about that kind of leader as well as approaches I've picked up myself over the years that have helped me be successful in these roles. For good measure, I also include the insights of a couple of *very* knowledgeable friends.

I occasionally make a reference to design work and design teams, particularly in a web context. This is because I spent most of my professional life working in various design roles, especially that of a User Experience (UX) Strategist. (This means I helped web companies figure out what their products should do, why, and how to achieve those objectives.) This report, in fact, spills many of the same points and stories as a book I wrote for UX professionals called *Experience Required: How to Become a UX Leader Regardless of Your Role* (New Riders 2015). (If you want advice specific to leadership in the design profession, you can find it in that book.)

The takeaways, though, can be applied to any professional environment. I've seen them hold true and work in all variety of professions. They aren't industry skills, they're *human* skills. They're principles. They're a mindset.

This is *not* about how to do managerial tasks. It's not about which meetings to hold and how often and how to run them. It's not about how to hire, fire, or promote. This is about the human skills that can help you become trusted, respected, and reputable as a leader.

Before we get into the more involved concepts—the ones that require long-term focus and that develop over time—I want to hand you a set of daily must-haves you can use for years to come regardless of what else is happening. So that on your worst day, you can rely on these few practices and still trust that you've kept yourself from getting carried away in the tide of internal politics and obstacles and objections and managed to somehow still do better as a manager than 90 percent of the rest of the world on their best day.

These take commitment, as well—I don't mean to imply they don't. But they are simple in concept, and, if you can stick to them, they will make up a base that will serve you and your team well *every single day* of your tenure.

Whereas much of what we talk about later are things that will help you manage *out* and *up*—things that will help you initiate and push your company forward—this chapter is for the things that will help you manage your *team*.

These are the things people who work under you will thank you for when you leave.

I highly recommend printing them as a list and then stapling it to your forehead. It's a small amount of pain for a whole lot of benefit.

Being Collaborative

One of the first things a lot of new managers want to do is *manage*. They want to begin inserting themselves as decision-makers, experts, masters of their domains.

Don 't do that. It's a mistake. Collaboration has a lot of benefits. You likely experienced most of them prior to becoming a manager.

First, everyone in a group can benefit from everyone else's knowledge and talent. Teams have backup skills, overlapping know-how, people with specialties that you don't have yourself.

Second, working in groups, with their close working relationships and their innate knowledge of who knows what and who's good at what, invariably makes individuals better than they could be on their own.

Third, rock-star employees are rare. Though a legitimate master within a profession can do some great things while working alone,

teams are far more often the better way to go. Members can help each other out in all sorts of ways.

Rock stars can also be destructive. Pretty much no one wants to be excluded from a project because the rock star on the team can do it all alone. If you want your entire team to look good, no single person can stand out as the lone master without damaging morale and quite likely the product you're all working to create.

Besides all that, collaborating is often fun. If you have the chance to bounce ideas off other people, take it. Even if the ideas to come out of it rarely surprise you, they will sometimes surprise you. That alone makes collaboration worth trying.

There is, however, a trick to making it work well: keep the team on the small side.

A study showed that larger teams perform better than small teams despite the fact that individuals on small teams performed better than individuals on large teams.

On large teams, people don't know each other as well. When a person on a large team needs help or additional information, she is less likely to know who to talk to. But because a larger team has more people on it, it will perform better than the small team overall. An individual on a small team, on the other hand, is more likely to know who to go to for extra information. Team members are also more likely to have better relationships with each other, because they've worked together more closely than they would on a larger team.

Steady As You Go

I found the first great manager I ever knew in an unlikely place: a video store I worked at in my twenties. Her name was Kim. I learned something from her I still believe in today, and have seen in countless other good managers since.

When Kim asked people to do things around the store, she'd stand slightly turned, one foot ahead of the other, with her forward shoulder tilted down and her head down. Then she'd look directly at you and say something like, "When you get a chance, can you please clear that shelf and put out the new releases?" She did the same thing when asking you to complete a task, when she asked how your

life was going, when she mentioned something her family had done last week, when she asked how the event was that you went to last night. She was never condescending or cold, never angry or harsh. When making a request, she was only mildly commanding. It felt like she was telling you a secret.

To Kim, nothing was worth anxiety. Nothing was worth worry, or urgency, or panic. Kim was as even-keeled as a person can be. Deliveries would arrive, customers would ask questions, lines would get long. Her voice never rose. Kim simply did not believe in freaking out.

And everyone loved her for it. Everyone happily did what she asked. For the most part, the staff remained as calm as she did. We looked at and treated her with all the respect we had, because she earned it, every day.

What I learned from Kim is this:

We teach other people how to treat us.

Overreactions teach people to be nervous about approaching us. Anxiety teaches people to avoid us. Pushing things off teaches people they can't trust us. Expressing frustration at a CEO's request to put something fun on hold so you can catch up on lagging projects teaches the CEO to treat you like a nuisance rather than a leader.

When you are calm and consistent, with few exceptions, you teach people you can be relied upon. That you are capable of handling any challenge. That you can be trusted as a leader.

The way you act and react affects the way everyone else will see you the next time they need you to act or react. Stay calm in all situations, and you will have all the respect you'll ever need to get things done when the chance comes.

Yes, this will be hard to do sometimes. Remind yourself. The long-term payoff of staff loyalty and trust and endearment is worth far more than any satisfaction you could derive from spouting off, even just once.

Eyes On the Road

Learning to achieve work Zen can take a while.

Years ago, a few months into my first director-level role, I began to notice a disturbing trend: meeting overload. No matter the day, no matter the workload, I was dragged away from my desk more frequently than I was able to plant myself at it and get things done. It ate at me for a month or so, then another month, and then another, until I found myself angry at every meeting invitation to hit my inbox. I was even resentful of every invite I created myself. I started scheduling them for shorter time slots. I walked in with agendas and skipped the small talk. I made it a requirement to walk out of every meeting with a decision—a takeaway of some kind.

It didn't help. I was still frustrated. The work at my desk was still sitting there undone.

A few months of that and I was decidedly *over it*.

Then I realized something.

Perhaps because there was no real alternative, or perhaps because I convinced myself to believe it, I had the thought that changed my experience for the duration of my spell at this particular company:

Meetings *are* my work.

I don't mean that my job was to have meetings (though, that was certainly part of the revelation). I mean that my job was bigger than a job—it was a mission, a purpose, a goal. I didn't go there to complete tasks, I went there to form and shape and execute on a larger vision. The work I did at my desk was part of it. The meetings I was pulled into were part of it. The daily interruptions by coworkers and questions and emergencies were part of it. Everything I did was *part* of that mission. It all contributed. It all could be tied back in some way to what I was there to achieve.

Many days came and went after that. Other jobs at other companies did, too. I've been just as frustrated by meetings and other distractions at other times as I was back then. But since the revelation, I've been able to remind myself of something:

There are distractions and meetings and demands and questions and emergencies in literally every kind of work, every week, no matter what you do.

They are not distractions. They are part of how you will one day achieve whatever it is you're so focused on. And *not one* of them matters.

Keep your eyes on the road. Every white stripe that flies by on your left side is another mark in the asphalt, another block of time, another stretch of mileage that will get you there.

Internalizing this—owning this—will all but eliminate your frustration. It'll make you easier to approach, less anxious, and *more* productive.

It'll make you the Buddha of your office.

Speaking Up

When you were called in for the interview, you were nervous, but you knew you were qualified—you knew you could do this manager thing. When you met with the people who would hire you, you were humble and gracious and not at all tough, but you knew you had opinions and beliefs about how to be a good manager and what made your last bad manager so bad. When you were offered the job, you knew you'd be on a big learning curve for a while. (Or maybe you didn't. Maybe you thought the fact that you were hired meant you had it all figured out already and that's why they hired you for the spot.)

Then you started, and something changed. Not immediately—not on day one—but soon, and one day at a time. You began to see more of the puzzle pieces that make up your company's decisions. You became one of the voices in the room whose job it was to recognize the constraints and the challenges and to navigate them, to measure priorities against time, resources against abilities. Over a few months, you grew *knowledgable*.

Great. But you also became something else (or you're about to):

Institutionalized

Yes, you learned to focus on the big picture, how to map out projects and divvy up the talent pool to the right places, and to get things done and to know that things must sometimes move slowly. Yes, you did all that. But in the process, you unlearned the things that made you want to be a manager in the first place. Like how important it is to watch over the *quality* of the product, the business, the work you're all doing. You unlearned how to be critical of the thing you

came there to do. In its place, you learned to sacrifice *this* for *that*, and *that* for *another thing*.

Snap Out of It

Your staff cares about quality. They all want to go home and feel good about themselves and their work. They want to believe in what the company that employed them is doing and how well it's doing it.

They'd really love it if you did, too. If you weren't just another lowly manager trying to appease the higher-up managers—another cog in the machine.

Just like questioning a government is one of the most patriotic things a citizen can do, at some point, the only way to truly care about the product your company puts out is to focus on the end result. To them, the product is all that matters. They don't care that Gianni calls in sick a lot or that Sheila isn't that great of a designer or that your boss wanted you to spend your time elsewhere for a while. They definitely don't care about your budget constraints. All your customers care about is whether or not the work you've done was worth doing—whether it or not made their lives better.

Believe in your product, whatever your "product" is. Speak up about it. They hired you because you were good at something. Remember to be good at *that*. Remember to care about *that*.

Taking Criticism

There's this Neil Gaiman quote I like to use a lot:

> When people tell you something's wrong or doesn't work for them, they are almost always right. When they tell you exactly what they think is wrong and how to fix it, they are almost always wrong.

Right now, let's focus on the first part of that thought.

When people tell us something is wrong or that something is bothering them, most people react as if it were a criticism.

Don 't be one of them. React as if it were a *test*.

Think of everything you say as a first draft of some brilliant future speech. Think of everything you write as the shoot-from-the-hip version of some yet-to-be-polished manuscript full of typos and unclear assertions. Then, when you send them out into the world—

these statements and words, designs and lines of code, products and mantras—think of them as a test. Then think of what you get back as data that tells you whether or not the thing you put out into the world did its job.

Your job as a manager is to manage as well as possible, which is likely far better than you can manage *now*. Especially in the early days, there's little likelihood you are anywhere near as good as you could be. Criticism is the only way to get better. You have to listen to it. You have to think about it. You have to consider what it is that's bothering other people about how you're doing it.

Did someone react badly to something you said? It could be that you said it poorly. Did a department react badly to a new policy you've implemented? It could be that you didn't communicate its value, or include them in the decision-making process, or even collect their input in the first place. Is a customer complaining about a flaw in your product you don't believe exists? It could be that the customer doesn't understand what was *intended*.

Criticism is how you learn. Learn to hear it as the result of a *test* of your work, not as an assault on your work.

Besides all that, if you ever want to be able to offer criticism, you must first learn how to accept it. You have to teach other people that you *can* accept it.

Giving Credit Away

It's supposed to be a good thing when you get credit for good work.

I knew a man once who was fired for doing so. Mostly because the credit he took was for work he had done as part of a team, and the people around him who had done the rest of it were justifiably unhappy with his credit-grabby hands. One week, he was on the cover of a magazine. The next, he was looking for work.

Around that same time, a manager I knew was telling a story about a work success and how she'd been nominated for an award quite ironically after having pinned all the good work on a project to her staff. She claimed, basically, that stepping out of the credit race was the key to her success.

"It's amazing what you can get done when you don't care who gets credit," she said.

I believed her. I began doing it myself. Here's what I've learned:

First, giving credit away is *gigantic* for morale. It helps newer staff members feel good about their progress, and it helps existing team members keep a healthy and positive outlook about their roles within the team and their contributions to the company.

Second, because it makes the entire team look good and not just you alone. Your team becomes a powerhouse. An individual can have a great reputation, sure, but the result is often that the individual becomes overloaded with work because everyone wants to work with that person. When an entire team has a good reputation, the rest of the company wants to work with it, the team as a whole builds a ton of clout that can be used to sway future projects, and when it comes time to hire more people, you can often have you pick of internal candidates.

And third, just like the manager who'd first suggested the power of giving credit away, I soon found that when the team was thanked for its work by the higher-ups in the company, its members tended to point to the *team* as their foundation. No one person tried to hoard credit. Quite the opposite; they wanted to give it to each other.

Rather than claim credit for my team's greatness, my team's work made us *all* look great.

Managing Things Away from People

The word "manage" has a lot of implications, a lot of social baggage. It's an ugly word many people associate either with a bad boss, the act of maintaining the status quo, or pushing papers around a desk while the minions do all the real work. And there's some truth to that perception. When managers manage, they do tend to manage *people*.

It's a mistake, of course. Few people want to be managed. To be *handled*.

To the contrary, most people want to be able to go home at night feeling halfway decent about themselves—to hold up some level of pride over their work, to feel a sense of purpose, to satisfy their own internal instinct and drive to be worthy contributors.

We don't need to *manage* that. It's human nature.

What we do need to manage is their ability to do it. And that means something else entirely.

It's not about allocating resources and handing out assignments and filling in the spreadsheets. Sure, that's part of the job, but it's neither why you signed up nor what anyone really needs you to do for them.

What your team wants is for you to keep stupid crap off their desks. Needless email, hijacked priorities, demands from people with no skin in the game, shifts in strategy that come out of the blue and can undo weeks of work.

Keep it off their desks.

As much as possible, be the person who fends off the relentless frustrations of professional life.

Stop managing people. Start managing *distraction*.

Creating Opportunities for Others

Once in a while, someone wants to break out of the usual and do something exciting.

You know what that's like. So, let them.

Helping others on your team become as well-rounded and skilled as they want to be is a great way to strengthen your team and keep people motivated.

Here's a story about this from my book, *Experience Required*:

> I once had a junior designer on my team who desperately wanted to do more strategy work. He'd told me so at least three times while the team was in the midst of a Herculean effort to rid itself of technical debt. He was at his least challenged, design-wise. He wanted to push himself.
>
> I understood. This isn't what any of us had signed up for. I was just as eager to get back to doing rather than redoing.
>
> A couple of weeks later, a project I'd been pushing for came to fruition. It was something I had been planning for a few weeks and had spent a good amount of time convincing the relevant stakeholders to take on. It was a project I really wanted to work on myself.
>
> Strategy came first. My wheelhouse. My favorite part. But something was gnawing at me.

This designer was really talented. He showed up every day. He slogged through the projects that had kept us all from any kind of forward motion. He did stellar work no matter how tedious the project. From the moment I'd joined the company, I'd wanted to see how far he could go, and here he was, stuck in the mud, not getting anywhere. He wanted very much to do more strategic work, and I very much wanted to give him the chance.

I had to do it. I gave the project to him. I told him I'd be there for guidance and to answer questions if he needed, but that the project was all his.

Secretly, I also committed to asking him loads of questions along the way that would allude to the kinds of things he should be thinking about. But I swore to myself I would give him some distance. Remember: Good ideas come from all kinds of places. We all needed to see what he could do, most of all him. He needed room to step out. To step up.

And so he did.

He did great work on that project. He proved himself to a bunch of people in a bunch of ways. And he learned some things in the process that enabled me to go to him later as a strategic thinker.

Again, people in the web industry already want to do great work. They want to learn more, take on more, build more, design more, ship more. They' re hungry.

So feed them.

When someone wants the chance to improve at something, find ways to let them. The next time a project comes up that appeals to that person, hand it off, even if you really want to do it yourself. Nine times out of ten, that person will step up. And both of you will be better off for it.

Even if you're not in the position to decide which people belong to which projects, you can always find little ways to involve others. If they're interested in the thing you're working on, invite them over to see what you can do together.

A little advice, though, if you're on the other side of this scenario—if you are the person who wants more opportunities: Do something to earn them.

I once invited onto a project of mine a designer who'd been pretty well known for his talents. He had worked at a local agency for a long time and had recently gone out on his own to become a consultant. He'd said to me a few weeks earlier that he, too, wanted to get more into strategic work. So when a project popped up that would involve a lot of it, I asked him to join the client and me for a two-day kickoff session. He accepted. We holed up for a couple of

days in a conference room and cranked out a bunch of ideas, and then planned to refine them over the next week or so. During those two days, he was great. He was full of ideas, we worked well together, the client liked him, and we got a lot done. His next job was to go create a more complete version of the strategy, make sure our initial design ideas supported it, and put together some more refined screens. We'd have a meeting about it the next Monday.

He never showed up to the meeting on Monday. And days later, when we pressed to see what he'd done, he sent us a digital version of a single sketch we'd done on a whiteboard the week earlier.

I haven't invited him into a project since. I'm sure you can see why.

If you want people to include you, be there when the invitation comes. Be present. Be reliable. Be engaged. The only way to get better is to drown a little bit. Be ready to jump in the pool.

CHAPTER 3
Learning as a Habit

As you likely know by now, to the people on a team, managers aren't peers. They're usually the people the rest of the team points to as the ones doing everything wrong, the ones holding progress back.

There's some truth to that.

Managers become managers by previously having been good at something *else*. Our hierarchical professional structures all but guarantee the "Peter Principle "—that we tend to rise to the level of our own incompetence. We get good at one thing after another until we end up managing—a skill that presumably has almost nothing to do with whatever we were good at before, a brand new skill for which you have no prior experience.

Congratulations. As a new manager, you've just risen to the level of *your* incompetence.

Don 't stay that way.

The Peter Principle might be entirely accurate, but it excludes all talk of the idea that we are, in fact, *built to improve*. Ambition isn't an inherent evil; it just tends to advance us to higher and higher positions until we reach one for which we have zero skill. This achievement doesn't render us incapable of learning. We can still learn. We can still do better. We can figure out how to be managers. Then we can figure out how to be *great* ones.

Likewise, we can learn to be better craftsmen at whatever craft it is we were originally hired to practice. We can help push our profes-

sions forward by poking holes in standards and digging for better options. We can remember that it's more fun to get better than it is to stagnate. Not only will it help *you* feel more fulfilled, this leading-by-example approach will motivate your entire team.

Learning as a habit means to consistently aim for something better than the present. Whether your skill level, your ability to manage distractions, your sense of empathy or respect for your coworkers, or something else, age and experience and self-examination will almost always eventually present a different perspective than you have now.

Good thing, too. Because if you'd never adopted a strategy of advancing past your current state, you'd have never arrived at where you are now. Keep that strategy and you could lead yourself into any variety of great situations later on.

Besides that, learning simply helps satisfy your own internal drive to be good at what you do. On a long enough timeline, every job turns boring and routine. Learning new things about it makes your work more fun and interesting. It makes you more useful and beneficial to your team. It gives you the best chance of advancing your *own* career. Not just your job, but your *career*.

Leaving Your Ego Out of It

There's a trick to it, though. You must learn to get your ego out of the way.

As a designer, part of my job was to have people try out my work via usability tests, the goal of this being to find issues in the design that would keep users from successfully completing tasks and being happy with the product we were producing for them. I wasn't designing printed works, after all—I was designing *functionality*.

Usability tests are a tough thing the first times.

Strangers come to the office, sit down in a little room with a computer and a microphone and a video camera. You ask them to try to complete various tasks—really important things that everyone should be able to do with your product. They push buttons and click things and talk about what their guesses are how they're making their decisions. They muddle their way through something you've spent a couple of weeks developing into what you think is a pretty

good design. Something you're confident about. They stumble and trip and get lost. Then, someone gets completely stuck on a page and abandons a task you were so sure was simple and clear. They tell you how hard of a time they're having. And you can't believe it. And you go home feeling lousy.

But then you get some sleep, and you wake up a little less anxious. You play back the video from the day before and you try to see what they saw. And slowly, you begin to understand how their interpretation was so different from your own.

And you learn from it.

Do this enough times and you realize something vital: they're not attacking your design; they're telling you what *they* need in order to understand it the same way you do. They're telling you how you can help them. They're giving you the chance to learn something, to do something better than you did before. They're telling how to reach a point in the future when you can make better guesses in the first place and end up with fewer problems in your work than you have now.

They're teaching you to take your ego out of the equation.

It can take a long time to teach yourself that it's nothing personal. That it's not about being right or not, or being perfect or terrible on the first try. It's about learning.

To do that well, you have let your ego be crushed, and then focus on what you can learn rather than how you feel about having been humbled.

Learning to Succeed, Not to Embrace Failure

All that said, the mindset you have about learning is key.

An idea has been going around the business world for the past few years that failure is a good thing. Failure, so the idea goes, means you're making the good attempt to break past the typical and achieve something more. If you're not failing, you're not trying hard enough. "Fail early, fail often," they say.

It's a lot like the aforementioned Peter Principle, in fact—rising to the level of your incompetence.

But I have a problem with this notion of failure as a positive goal. Not so much with the idea behind it—you'd have never become a manager had you not been willing to push yourself. My problem is with the phrasing.

Repetition is a core tenet of the persuasion industry. Say something over and over again, according to some strategies, and people often begin to believe it, regardless of whether it has any merit. For example, we've all heard for years now that pit bulls are lethal, and we've learned to accept it as truth whether the notion has any real merit or not (hint: statistically, it really doesn't).

You can use this for positive purposes, like building enthusiasm for a product vision for the team of people responsible for building it. But repetition also reinforces your own beliefs. The more you say something, the harder it is to change your own mind about it later.

Failing is a negative idea. Even if you know that the intent of "fail early, fail often" is the opposite of its wording, having the word "fail " sitting in your head all day can be outright self-destructive. Your brain is hardwired to feel negatively about negative words.

It's far better to set a goal you want to achieve.

Like *succeeding*.

Rather than fail early and often, try examining every situation for what good can come out of it, what insight can be gleaned from it. Focus on the wins, and celebrate them.

Repeated Failure Gets You Nowhere

Besides that, failing only teaches you how to fail. Success teaches you how to succeed.

When startup entrepreneurs succeed once, they are more likely to succeed again. When they fail, however, they are just as likely to fail the next time around as they were the first time. This is what venture capitalists have found to be true, anyway.

Some people believe you can learn as much from failing as you can from succeeding. This is just not true. There are a billion ways to do both, but only one of these outcomes is desirable. You're not trying to learn how to fail. That's absurdly easy to do. There's nothing to learn there. You're trying to learn to succeed.

Fail once and you've learned one way to fail. But succeed once and you've learn at least one way to succeed. And if you learn to succeed one way, your odds go up for learning to do it another way. Because in the process of succeeding, you pick up a few skills necessary for doing it again. You learn how to shift strategies. How to prioritize. How to rethink. How to sit back and stare at the rafters for a minute to reimagine how you'll achieve the goal. You learn how to push through the obstacles. How to align the variables that will give you a positive outcome.

Most of all, you learn that success is possible, which is essential to believing it can happen again.

CHAPTER 4
Skill Equals Perspective

One of the most significant things you can do for your staff as a manager is to know their work—to understand what they do, how they do it, and what it means to them to be tasked with those particular demands on a daily and yearly basis.

First, and perhaps most obviously, this helps you to build rapport with your staff, but this isn't because it means you can now chat about it over drinks at the company happy hour. Rather, it's because understanding their work helps you respect their work. The difficulties, the constraints, the sophistication, the basic knowledge it takes to do the job.

Meet my friend Christina Wodtke.

Christina has been the GM of Social at MySpace, Principal Product Manager at LinkedIn, GM at Zynga, and publisher of revered web magazine *Boxes and Arrows*. She has also been involved with countless startups as a Silicon Valley veteran. She points to a lack of respect as one of the biggest mistakes a team member can make.

"I would say the one thing that I really badly want," she says, "and I hate to say it's rare, but it is, and it kills me that it's rare, is genuine respect for other disciplines."

Big companies, especially, tend to have multiple people in the same roles, and when that happens, they tend to stick together and forget to consider all the other aspects of the projects they're working on. She says she meets a lot of people who simply don't understand or care what that other people on their team do, or what their manag-

ers do, or what the general manager does, or what Marketing does. And Christina notes that these people who spend their time ignoring everyone else's contributions are, ironically, the same people who most often say, "How do I get people to respect me?"

She says the key is to not only to be respectable, then, but also to respect others.

Very often, she points out, people who have similar jobs tend to stick together. "They all go to lunch together, and they all sit together, they all leave at the same time."

This keeps them from gaining solidarity with the rest of the people on their projects—the ones doing all the other jobs necessary for successful execution.

I've seen this myself. A lot.

Understanding the work others do helps build chemistry, rapport, *respect*. As a manager, these things will change your life.

Second, understanding your team's skills is essential when it comes to planning, work allocation, and strategic improvement. This much is obvious: if you don't know what the people on your team do all day, you can't possibly plan a project around them. Instead, you'll end up making assumptions about ability levels, time considerations, and all sorts of other aspects of a project that will only lead it to a point of failure. At the very least, you should be able to identify which people can pick up where others leave off—where you have overlap.

Finally, to make yourself even *more* useful as a manager, you can actually gain some of those same skills. There are major benefits to overlap.

Upside of Overlap

That small teams are more effective than individuals is common knowledge. No individual can think of everything, do everything, know everything. Teams make people better. When the people on those teams share skills, they collectively have the benefit of shared interest. They read different articles, different books, go to different conferences and local events. You bring those insights back to the group and share them. You learn from each other. You change each

other's minds. You get the benefit of multiple perspectives. You get the benefit of debate.

This is great for a company. It's great for the products you're working on. And it's fantastic for users because it's not a single-minded effort that brings a product into their lives, but a collaborative, considered, deliberate effort managed by a group of people who come up with more ideas, poke holes in more arguments, anticipate more problems, and solve more issues before they get out into the world.

As a manager, your own ability to overlap skills with the people on your team means that you can get something you're going to need a lot: extra help. Individuals—especially the stronger performers on a team—invariably become maxed-out at one time or another. Bringing your own skill to the table, you can contribute to a project directly when the team needs some extra hands. This gives you more credibility, too—it means the team can trust that you understand everyone's work.

Likewise, anything you can do to encourage overlap among your team members will only help you in the long run. Does your company offer ongoing education benefits? Are there workshops and conferences around that would be useful for helping your team build new skills? Odds are, at least some of the people on your team already want to delve into new subjects and further their skillsets. You won't need to convince them. Wherever you find this drive, nurture it. Figure out how to get your company to pay for these mind-expanding opportunities and make them happen for your team.

When one person needs help, others will be there.

Be Replaceable

Before we move on, let's talk about something that probably popped into your head while reading the last section: namely, the fear that redundancy on a team makes it sound like you're replaceable.

You are. You should be. To elaborate on this, here's a quick snippet from *Experience Required*:

> No matter how good you are or how well you fill a niche, or how useful you are in whatever situation comes up in a day, you are replaceable.
>
> This should comfort you. You need to be replaceable.

> If you're absolutely perfect for one thing, you're less perfect for a bunch of other things. It'll be harder for you to find the perfect job. It'll be hard to find any job. If you're replaceable, you're hirable. Besides that, you want to be able to move on at some point. Test out your other skills. Develop new ones. Work on a different product you care about more than this one. If you're replaceable, a company won't trap you into staying by throwing more and more money at you until it's impossible for you to leave. Every job eventually turns into the wrong situation, whether because the job changes or because you do. The web industry isn't built for lifers. It's built for people with an endless sense of adventure. An incessant will to take on the next project. The next challenge. When that feeling strikes, you need to be able to leave. Don't get trapped by money. Make too much cash in one place for that perfect thing you do, and your desire to get out will drive you mad.
>
> There's also the reality of emergencies. You're a human being, and you're going to experience a few of them. A health emergency. A car accident on a day full of deadlines. A death in the family. Sick kid. You name it. You need to be able to take a day off, a week, three weeks, and not spend that time worrying about what you're not getting done. Let them get done by someone else. No UX project will ever be more important than your life.
>
> Be replaceable.

T-Shaped People

All that said, there's a problem with generalists, and because of it, I hope you'll consider attempting to go beyond a basic level of skill. Here's my case:

The hard work of professions far and wide has been distilled and reduced and simplified into how-to articles and videos and workshops that are now absurdly easy to find online. If you follow anyone in your professional circles on social media, it's hard to even look at your phone without tripping over another "5 Ways to Do This, That, or Something Else" article.

Great. This is something the Internet is supposed to be good at—enabling people to empower themselves to learn more. And it's doing a bang-up job.

That information, though—those insights, all that "thought leadership"—has to come from somewhere. It has to be dug up, made meaningful, turned into teachable, lasting content the rest of us can use to get moving more quickly. *Your* profession probably became

accessible by way of the people who spent years and decades shaping it. The experts. It seems easy now because it's been simplified, communicated, learned, and then interpreted and *repeated*.

Again: great. Our ability to stand on the shoulders of giants means we can more readily change careers, enhance our current skillsets, and do all kinds of other things a lot more easily than we used to.

It also means that if *you* were to become one of the experts, you could help move *your* profession forward. There's still opportunity to improve on what you know, what you've learned, what's been taught and perpetuated by the old experts. There's always time to challenge the norms and help make your profession better.

Expertise is like bird food. Someone needs to go get it and bring it back to the others. When everyone's a generalist, there' s no expertise.

Because of this, I highly recommend deep-diving into some aspect of your work to find out how it can be improved—to find out how the current standards were established and see if there's a better way. If there is a better way, try it out. If it works, talk about it. If people listen, maybe write some articles or even a book. Help the people who will come after you.

There are a couple of key reasons to do this.

First, most work is short-lived. The effects of what any one of us does in a day rarely outlast the year. But improving the profession as a whole? That leaves a mark. Few things are more rewarding than knowing you were able to contribute to potentially decades worth of effects by digging into your profession and helping to raise its proverbial bar.

Second, and arguably more important, mastering a skill means finally understanding how deep everyone else's professions might go, as well. You can gain sympathy for what everyone *else* faces in *their* roles. Everyone thinks they know something well enough until they go headlong into it with all their might and discover just how little they knew before.

As a result of all this digging, you gain a perspective not a lot of people these days have: what it's like to have both a *breadth* and a *depth* of knowledge.

When you can stack expertise in one area on top of a whole bunch of general knowledge in others, you become a T-shaped person. This is just what it sounds like. The horizontal top line of the "T" is a set of skills in which you have reasonable understanding and knowledge, and the vertical center line is a *single* skill in which you have significant depth.

This is intrinsically rewarding; although a great many people can do a great many things, few of them are great at anything until they specialize in something. It's also extrinsically valuable; specialization in at least one subject makes you incredibly useful to your team, and later on, it makes you incredibly employable. When companies need deep knowledge, they need to have some specialists around to bring it. If you can demonstrate that you know your stuff, you can become the go-to for your subject.

Besides all that, being *well-rounded* means never getting to a point of greatness, never seeing what you can do at your best.

But even without all these benefits, expertise makes you useful to your team, and a manager who is as useful as another team member is a lot easier to respect than one who just *manages*.

CHAPTER 5
The What Over the How

A lot of industries have passionate professionals. A lot of those professionals like to hang around in discussion groups, asking for advice, offering it (frequently without even being asked), bouncing ideas off each other, and debating the ins and outs of a hundred different aspects of the job, whatever the job might be.

A lot of those debates—a *lot*—end up being about process. They end up being about the way you do things, what order you do them in, and how it all turns out at the end. Process, a lot of people say, is the key to success. To do anything well on a consistent basis, you must be able to repeat what you did well previously. If you can do something well twice, the thinking seems to go, you should write it down, wrap it up, send it off to the textbook writers, the authorities, the experts, and record it in the annals of history. You, after all, now have it all tied up in a pretty red bow. You've mastered process.

Trouble is, nothing about the success of process is provable. Not in soft-skill-focused professional work, that is. Not in office life, where projects come with so many variables it's nearly impossible to find two that are even all that much alike. The number of people, the people themselves, their skillsets, everyone's time constraints, the budget, the deadline, the reason for the deadline, vacations, health emergencies, holidays, technology, you name it—every last detail of a project is as variable as the last one.

In a couple of decades worth of projects, I've never seen a "repeatable process" that could truly be repeated, let alone to the same effect.

The only two things consistently in common are the two principal constraints of time and money.

Also, this process thing has been studied. Turns out, it's not successful at all.

Tools, Not Processes

Boxer Mike Tyson once said, "Everyone's got a plan 'til they get punched in the mouth."

This is exactly what it's like to work under tight deadlines and tight budgets. It's like getting punched in the mouth. And when time and money rear their ugly heads, process is the first thing to go. People pick and choose what and how something will work, and they do it quickly. *Forget process*, they say. *I have things to do.*

Instead, they look around, pick up their proverbial tools, and start hacking away in whatever approach makes sense at the time. They take their tools, and they do something really cool with them.

A respected designer and design manager colleague of mine, Stephanie Troeth, saw this firsthand while working for several years with an otherwise very smart and talented team who couldn't seem to avoid the pitfalls of process.

"Sadly, they made one major mistake,"she said. "They tried to put in layers and layers of formalized process. When you have a roomful of smart people, you have to acknowledge that they might be able to short circuit the process and arrive at a different way of working."

I wrote about this in an article for *UXPin* called "When the Best UX Design Process Isn't A Process At All". Here's a snippet:

> Jared Spool has done some research on this. You know the guy—he's the one who's been running a design research company and think tank since the '90s called UIE. Few people have spent more time talking to designers and companies than Jared.
>
> Know what process he's learned is the one employed by the most consistently successful design teams? It's the same one I find every time I walk into one of those conference rooms with a great team.
>
> The improvised process. The one defined specifically for that project, and often not defined at all.
>
> As it turns out, the teams that do measurably good work on a regular basis, with the fewest issues and the fewest failures, are the ones

without a de facto process. They're teams which devise a custom process for every project, and frequently go into a project without one at all. Conversely, the teams which most consistently accomplish the mediocre—that dirty, awful word—are the ones who live and die by their processes.

I know. It's nuts. But is it? Think it through.

The teams who succeed the most are the teams full of people who work well together—who communicate well together—and who have a broad range of skills and tools they can apply at any given moment. They approach problem-solving in a way that works with the constraints of the situation. They have all the UX tools you can find—strategy definition, user research, UI design, prototyping, usability testing, data analysis, and on and on. To do their work, they pull out what they need, when they need it.

- At the beginning of a project, gather around and decide together what design activities will be appropriate for the situation, whether thorough user research combined with rapid usability testing with paper prototypes or some other combination of things that will give you the best insights.
- At every point of the project, hold design brainstorms and reviews so that everyone can continually keep each other in check with regard to the goals of the project and how well the team's ideas support those goals.
- Ensure that everyone knows who has final design decision-making power, and that this person relies heavily on evidence and solid rhetoric rather than whim.

Improvising

All that said, the incredible willingness of teams to forego formality and begin scrapping their way through a project is far from the best reason to let your team focus on tools rather than process.

The best reason is that it gives them *autonomy*, and teams need autonomy to feel good about their work. It's crucial to a person's sense of intrinsic motivation.

Daniel Pink talked about this in depth in his book, *Drive* (Riverhead Books, 2011). In it, he posits that autonomy is one of the three things needed for a person to feel fulfilled and motivated by a job or task, besides mastery and a sense of purpose (seeing how one's work maps to the company's larger intentions). The ability to make deci-

sions—to choose how to approach a challenge, and to execute on it with a hand in the decision-making process, is key. Minus autonomy, people are just following orders. And that's not fun for anyone.

Autonomy makes people better, which makes outcomes better. Nurturing this approach within your team can mean better results and a happier staff. Orders are for drive-thru fast food joints. If you want great work from a team, the people on it need the autonomy to use their tools in whatever way they see fit to achieve success.

Of course, this right to improvise has other benefits, too.

It means your team will be able to invent new ways to get things done, which goes right along with your new mission of finding ways to improve your profession (see the section about T-shaped people in the previous chapter).

I wrote about this in *Experience Required*:

> Improvisation can get you more resources out of a slim budget. It can get you impromptu coding tests to validate someone's guesswork. It can get you through a usability test with a crashed laptop by sketching screens on napkins and stepping through a task flow anyway.
>
> Your ability to improvise doesn't just help you. It helps the other people involved trust you. Depend on you. It makes them come to you when they can't think up what to do next.
>
> Psychological studies have shown it's more difficult to have good ideas and make good decisions when you're under stress. Staying aloof to a certain extent can give you the brain space you need to bust out a good idea under bad circumstances. Being able to improvise shows you can keep a clear head when a project goes wrong, and approach situations with a level of calm not everyone has.
>
> It also gives the impression of experience, even if you don't have any. So many designers leap to solutions devised in a panic. An experienced designer stays calm, shuts up for a minute, thinks through the angles and asks questions about the constraints and looks around for anything that might help in the moment, then pitches something no one else has been able to think up.
>
> I worked with a programmer once who could do this.
>
> I've always thought it takes a certain kind of mind to be a brilliant engineer. This guy was the poster child for it. When people got stuck, he'd sit next to them, stare at their screens for a few minutes, ask some questions, and then tell you to do something that soun-

ded like it'd come out of left field and which worked on the first try. It was virtually the opposite of everything you'd tried. You'd spent five hours on the effort only to realize you'd been approaching it the same way the whole time and that your approach wasn't going to get you anywhere. (Granted, I was a mediocre programmer. But even if I'd tacked up five more years of experience, I'm pretty sure he'd still have been faster than I was in situations like that.)

It was downright satisfying to watch someone do this. It still is. It's the kind of thing that makes you sit back and ask, "How did you do that?"

After a bunch of years of experience, people might start asking you that question. In the meantime, you can fake it. And faking it will help you learn how to do it.

Improvisation shows competence. It shows situational awareness. It shows intelligence. It shows you truly have a grasp on a bunch of different concepts and skills, and are able to pluck them out of the sky at will, rearrange them, and make them into something new. It means you understand concepts over prescribed actions and can apply them to any situation, dire or stable. You're not the kind of person who sees one way to build a table and runs to Home Depot as soon as it goes awry. You're the kind of person who walks back into the shop, picks up a screwdriver, a piece of scrap wood, and some glue, and sorts it out.

You're the one who gets things done.

This makes you especially useful when deadlines are tight. Which is most of the time.

Public Perception

There is one other reason to focus on *what* you do over *how* it gets done. It's this:

What gets done is the only thing anyone really cares about.

In other words, there's a public perception—the perception of the higher level managers, the execs, the customers, the shareholders, and everyone else—that matters far more than how you do something, and that's simply *that* you did something. That you are actively doing something now. That you will do more in the future.

This is what people who sign checks care about. And as a new manager, it's a great way to show them you're focused on the business, not just whatever skill you'd gained competency in before leveling up. Now that you're a manager, your job is to think about how the

things your team does now will help the company later. Once the team does those things, your job is to go tell someone about what was done.

It's rare that someone will argue against your methods for having achieved said outcome. If they even ask, it's usually out of curiosity. Everyone else is winging it, too. They just want to know if you came up with something cool that they can try out on their own teams.

Odds are, absolutely no one is watching your every move to make sure you followed some strict process letter by letter. They have their own problems, their own work to get done, their own anxieties about whether or not they're doing it well. All you need to worry about is being able to say you *got the thing done*. That's what will demonstrate your team's abilities. That's what will earn you successes. That's what will earn you the respect you want so that you can get in on more interesting projects later.

CHAPTER 6

Saying It with Purpose

I can practically guarantee that you do more writing in a day than you do design work. You do more talking. You do more pitching. More wondering aloud with coworkers and peers. More speculating. More defending and convincing. So few of these moments, in my experience, are seized with the same vigor for clean communication that you put into a design. So few of them are planned for their purpose, considered for their effectiveness, or evaluated for their results.

I have to be honest: Most people *suck* at communicating. (Not all of them, of course, but yeah, *most*.) And remarkably few people know how much it's affecting their ability to lead.

This happens because people tend to be bad at the thing at the very core of good communication: clear thinking. The human brain just isn't very good at turning off all the noise of the day so it can focus on a concise, clear message. It's just not an easy thing to do.

But failing to communicate well leads to countless problems. *Daily* problems. *Financial* problems. *PR* problems. An unclear message results in an inaccurate interpretation, that leads to guesswork, and next thing you know, you're wasting hours and hours trying to clarify details that you thought should have been understood from the beginning.

That's why it's so important to master communication. To learn to say things with purpose—a purpose of clarity, concision, and completeness.

And you do that by using tools.

This chapters covers tricks to good communication.

Framework for Thinking

The first thing you get when thinking about pretty much *anything* is a bunch of scattered thoughts that threaten to ramble and confuse anyone who would hear them. These are not the thoughts you want to put out into the universe. If they're rambling to you, they'll be insufferable to anyone else.

First drafts are notoriously bad. The better communication is the one several drafts further down the road. Sadly, most people stop at the first draft.

Clear thinking comes from applying some structure to those thoughts. You achieve coherent communication by applying a sort of framework to your otherwise disorganized, stream-of-consciousness-style internal monologue.

The framework is like a college essay—the kind with a thesis and supporting statements.

Set a direction
: Decide on a main point you'd like to get across and state it right away. If information is like a hand-drawn map, this initiating idea is like the part where you take out the paper and write "North" at the top.

Go there
: Spell out all the ideas that lead to the takeaway. If your direction is that you think the company should consider fixing a backlog of problems with a current product, here is where you explain what the problems are, how much they cost the company, how much facing them could make the company later on, and how long it all might take.

Tie it all up
: You opened with a direction; now you've reached it. Restate your main idea and either end with a call to action or a question. Or simply stop talking. (That one's really difficult to do for a lot of people.)

Not everything that needs to be communicated is this complicated. When it is, however, such as when you're initiating a project, describing a plan or approach, or pitching an idea, this clarity in

thought is *crucial*. Knowing your destination makes your communications deliberate.

I asked Christina Wodtke what she thought were the most valuable skills a designer could have that were also the least represented in the design industry. On her list:

"Communication—verbal and written communication—is incredibly high and remarkably rare," she said, "and I'm not sure why. If you want your work to go live, you have to present it, and I'm amazed at how poor presentation skills are—setting people up, telling them what kind of feedback you want, explaining your choices, bringing them on the journey."

What she's talking about is the idea of pointing your design skills at yourself. Learning to apply your design chops to your own day-to-day communication.

Christina continued:

"It's just a matter of saying, 'You know all that stuff I just did with the interface?'—or whatever you're working on—'Now let me apply it to my own personal interface.' Let's try to be clear. Let's give a clear call to action. Let's be user friendly. Many designers aren't as user-friendly as they should be."

On Writing Well

Despite this being an era in which we use the written word to communicate more than at any point in human history, people seem to have worse writing skills than ever. But it has so many advantages that we should all take it more seriously.

First, writing is a thinking exercise. Many, if not most, professional writers consider writing to be the tool they use to develop ideas, not just relay them. They write to *find out* what they think. Writing forces you to have complete thoughts. It forces you to stay on a subject, to think through it completely. Then it allows you to see your whole idea in front of you where you can revise it. *Improve* it.

Here's my advice from *Experience Required*:

> If writing is something you struggle with, consider ways to improve at it. Simply getting a handle on the basic tools of writing can significantly affect every project you work on. Sentence structure, grammar, punctuation, organization—all of that helps you be

clearer with your colleagues about project details. To be more persuasive. To be more deliberate.

Look at writing classes at a local community college. Check out writing courses online, like those offered in the Gotham Writers' Professional Development section. Find writers you like and study the ways they write, the same way you study great designers. Dig up some books on the subject.

My personal favorites are:

- *The Elements of Style*, by William Strunk Jr. and E.B. White (Longman): Even if you just got out of a college English course—especially if—start here. This is a classic book that gets you cutting down on the flourish and passive voice they so appreciate in academia and truck commercials, and get you writing straight, clean, functional sentences that cut to the chase. But be warned: If you stop with this book alone, you'll regret it.
- *On Writing Well*, by William Zinsser (Harper Perennial): Everyone has a book that changed their lives. This is one of mine. It focuses on the principles, methods, and forms of nonfiction writing, and even breaks down what's specific to various types of nonfiction.
- *How to Write a Sentence: And How to Read One*, by Stanley Fish (Harper Paperbacks): If you hated Strunk and White's The Elements of Style, this book offers a nice counter to it, tossing out the standard arguments for sheer minimalism and looking into structures and styles that can make your writing not only functional and purposeful but beautiful.
- *Several Short Sentences About Writing*, by Verlyn Klinkenborg (Vintage): If you get into writing at all—even if you don't—nothing will blow your mind more than this one. Nothing will make you a more concise and elegant communicator. Every sentence begins on a new line, a fact that illustrates at every step the effect of a sentence standing on its own. Each one is as pure as can be. It not only teaches you how to write with mastery, it leads by example. (If you've read my tiny tome The Tao of User Experience, know that I wrote it immediately after reading Several Short Sentences About Writing. I haven't been the same since.)

The books in this list offer a superior, and very *quick*, education on how to get your ideas across, and how to form them well in the first place.

Enabling Comprehension

After you practice all these things for a while, with any luck you'll notice a couple of interesting benefits. The first is that you'll learn to better enable comprehension for your recipient. (Of course, their willingness and ability to read for comprehension are beyond your control.)

You'll notice two things in particular:

- The most understandable sentences are the short ones—the ones with a single subject, a single verb, a single point.
- Lengthy sentences are more difficult to track. Often, by the time you get to the end of one, you've forgotten what happened at the beginning. People do this a lot when they're trying to demonstrate they're good communicators. They use sentences so dense with facts and big words that everyone struggles to read them. It's ironic and sad.

Ernest Hemingway wrote at a fourth-grade reading level. Not because he *read* at a fourth-grade level, but because simplicity begets understanding. In fact, very few well-known writers write above an eighth-grade reading level. When the point is to be clear, short sentence containing plain language and straightforward information always does the job better than flowery, complicated sentences jammed full of information that would be best split out into multiple sentences.

Once you notice this fact as a *recipient* of information, you can employ it as a *conveyor* of information. Learning to read and listen for comprehension teaches you to write and speak for comprehension.

Mapping Your Message to Their Concerns

One key to saying something with purpose is making the recipient more eager to hear your message. And the best way to do *that* is to map what you're talking about to what they want to hear.

I'm not suggesting you learn to pander—to simply tell people whatever will appease them the most handily. I'm suggesting you find the connection between what you need to know or to get done and the concerns and cares your recipient is most likely to have. The CEO of

a company larger than a startup rarely cares about the specifics of a design, for example. Rather, that kind of CEO cares about how the design will draw in new customers and retain existing customers. When you talk about a design to the CEO, then, spend very little time on how it works, and more time on the data and other insights that brought you to the conclusion that it will help the company and on precisely *how* it will help. Likewise, if you're trying to talk the CEO out of a bad idea, focus on demonstrating how the idea might negatively affect dollar signs later on.

As a manager, this mind set will help you justify your recommendations to both your team and to your own bosses. It will give you the power to get more things done.

This will take some diligence. It's not something you can remember from time to time. It's a *practice*. (Arguably, it should've been included in Chapter 2. But it makes more sense here in context of discussing communication.) I say this because it's the kind of behavior that helps you exponentially. Over time, the more you stick to this mind set, the more people will inherently trust your recommendations. The more they'll know you are worth listening to. The more they'll know you are a person they want at the meeting, at the table, on the project, out in front.

Do yourself a huge favor:

Once you get the hang of this approach to communication, teach the rest of your team how to do it, too. *Everyone* needs to be able to speak the language of the other roles in the company, and doing so is yet another thing that will build your team' s reputation.

CHAPTER 7
Becoming Convincing

One of the best things you can do for yourself as a manager is become convincing. Higher-ups, team members, other departments, customers—you name it, you're going to be spending a lot of your time either pitching ideas, defending them, or trying to demonstrate the value of a decision.

Ironically, this is just the kind of thing they don't teach you in school. So I thought I'd offer a little assistance.

First, a quick story I told in *Experience Required* that demonstrates why this matters:

> In 2005, I was running a UX team and in need of another designer or two. I wrote up a job description, handed it off to HR to post to the company website, and waited for the fantastic candidates to come running, résumés in hand. The first one I got was from a guy on one of the product development teams—a programmer who had recently become enthralled by design and wanted a chance to dig in. Seemed he'd been hovering around the Jobs section of the company site waiting for just such an occasion. As soon as it showed up, he clicked the button.
>
> I'm the type who hires from all directions. The way I see it, a person can be better at the more readily learnable aspects of UX as long as that person has the intense curiosity and other qualities so crucial for a good designer. That same year, for example, I hired someone with fairly poor design chops, but who was hooked on it like a drug and who had a cognitive psychology degree—a combination that's pretty hard to find. I'm not afraid, in other words, to teach someone who has the built-in gusto and smarts. A designer can always

get better. They can't always grow more brain cells or become more naturally ambitious.

Besides that, I was a programmer once (a three-year case of designer, interrupted). I could identify with a guy who wanted to break out and challenge himself.

So I scheduled the interview.

He got through some of the questions well enough, albeit a bit nervously. He clearly had no design experience, but he'd been reading books and had become somewhat awestruck by the complexities of form design and the notion of making everything simpler. He was at that point where we all start—the one where he would see bad things and point them out to other people and complain about them. When you don't know what happens next, that's what you do—complain. It was fine. I could work with that as long as he cared.

After a while, I said to him that a lot of design was about being able to demonstrate that your recommendations are worth considering, and I asked how he approached convincing people of the merits of his ideas. He was a programmer, after all—surely he'd had to do this before.

Part of his response was a sentence that's stuck in my memory ever since:

"Well, I'm usually pretty good at waving my arms around and screaming until I get people listen to me."

He waved his arms around while saying it as if to prove he could. And indeed, he seemed capable of some perfectly suitable arm-waving.

I didn't hire him, though. The arm-waving just wasn't as compelling as I wanted it to be. It wasn't the fluidity or style so much. It was the sheer lack of rhetoric in his response that his arm-waving was trying to keep me from noticing. If his rhetoric was a total non-starter in an interview—where he was supposed to be presenting an argument for why I should hire him—it's unlikely he'd be able to help me convince the massive engineering team at this company that design mattered.

Why has this story been stuck in my head for so long?

Because for all the ridiculousness of this story, I haven't met many managers who could've answered the question any better.

Each of the following sections discusses a piece of what I consider effective tools for persuading others of your recommendations. They lack any hint of deception. To the contrary, they're pretty

straightforward and honest, and simply need to be remembered and considered and practiced.

Before we move on, I should mention there's a pretty great resource on the subject of rhetoric that goes well beyond what I can do here. It's the book *Thank You for Arguing*, by Jay Heinrich (Three Rivers Press, 2007). It's a quick read that introduces all the basics, and in it, the author references conversations with his kids a few times as real-life examples of rhetoric, which makes the whole thing personable, too. Highly recommended.

Listening

I worked with a design agency once that had a client who sold fencing. The client wanted the agency to design a new website that would basically serve as a brochure for the fencing company. About Us, Contact—the usual.

When I spoke to the client directly, however, I asked for a general rundown of what it was they wanted—to make sure we all agreed on what was going to happen. I asked about the client's goals for the site. That's when the person on the other end of the call said something a little incongruent with the agency's assessment of the situation:

"We want to use the site to replace the brochures we use now when we're walking around the yard with customers."

My next question was something like, "I'm sorry, what?"

The client didn't want a simple website. They wanted to replace the paper brochures their salespeople used with customers. They wanted salespeople to be able to walk around the yard with iPads and going over all the options a customer could ever dream about, and have the option to order things right away.

I called the account manager at the design agency.

"Did you know the client wants to put its entire brochure online and enable its salespeople to use iPads to put together estimates and submit orders?"

"Um, no, " he said. "That would be a much bigger project than what they told us before."

Actually, no. It wasn't a bigger project. It was the same project the client had wanted all along. The agency just hadn't heard it before. The guy at the fencing company hadn't known how to communicate what he needed, and the account manager at the agency hadn't really heard what the fencing company was after.

It was a major lesson for the agency in the value of listening. In this case, it was actual, *quantifiable* value. The agency had won a much bigger, far more lucrative project than they'd thought. It was thousands of dollars more valuable.

Before you can make a case for any kind of solution on a project, you need to know what the problem is. And that means listening.

Listening helps you to determine what the constraints of a project really are. What the client's concerns and goals really are. It helps you see whether or not you already have the right argument in hand or if it needs revising.

Asking

Part of listening requires drawing out the right information. And that means asking questions. One of the key ways, in fact, to spot the most curious person in the room—the one most likely to subsequently do great work on a project—is to look for the person asking all the best questions, whether in interviews, conference rooms, or by email and phone.

Why?

First, because people who ask questions get the most information, and they get the best information. People rarely know which details of an idea or situation are going to be most relevant when trying to pass it along to someone else. A computer programmer, for example, who is intimately familiar with the programming languages used within a company can easily forget what it's like to *not* know how they work. In a discussion about "technical debt" (old problems that need attention), the programmer can end up making all kinds of assumptions, leaving the other person in the dust. Asking questions means drawing out more and more of the facts that will lead to a complete, knowledgeable, useful, meaningful understanding of what's really being discussed.

Second, asking questions helps the person being questioned think through all the details. Let's say, for example, that a marketing person is debriefing a designer about a new campaign concept. The marketing person has one understanding, the design another. The marketing person knows the purpose of the campaign, and some basic details, but not necessarily where it will be targeted, what precisely it will try to communicate to potential customers, or when it needs to be done in order to get all the necessary approvals before launching it. A designer who asks questions can determine that all this pretty important information is missing. One who goes without asking questions will just run off and start designing, potentially creating work that completely misses the point of the campaign and shows up a week late.

Finally, asking questions is key to being persuasive because it demonstrates to the other person that you're focused on the details. And because when there are *gaps* in those details, you can suggest your own solutions and help push the idea along. And once you have skin in the game, you have more opportunity to influence the outcome.

Restating

Some quick advice to take full advantage of the answers you get:

Restate them. As in, out loud. For several reasons.

- It helps the person answering your question know you've heard them correctly.
- It ensures you understand the answer.
- Restating the answer can cause the person who provided the answer to rethink it. Simply hearing it said back forces the person to recognize some flaw in the logic they couldn't hear when the answer was still hidden away inside their own

Educating

Whether you're recommending a project to your boss, a change to your team's structure, or something else, taking the time to educate people around you on the rationale for the recommendation can make all the difference in how it's received.

Yes, it takes time to explain things like this. It can take even more time to educate yourself well enough to know what recommendation to make. But it's always worth it. It buys you respect, and it shows the people around you that you respect them, because what they hear is that you care enough to explain your rationale. It also demonstrates that you *have* a rationale —a deep, considered rationale—for everything you do. It builds trust.

In all, educating your clients and coworkers and stakeholders with every recommendation you make has several major effects.

- It guarantees you have a reason for your recommendation. If you can explain it, you've thought it through.
- It gives everyone else a good reason for the recommendation. (Often, they just need to know there is a reason.)
- It has a great long-term effect: It teaches people to think about design. To think like a designer. To think like a user. It teaches them that every decision has an impact on a user's experience and therefore should be considered. Do this well, and over time you won't need to form an argument for your recommendations nearly as often. The people around you will have learned to make better design decisions in the first place.

Sometimes, your best arguments don't mean a thing. Your clients, coworkers, and other stakeholders want to do what they want to do no matter how potent your case is for doing it another way. Your research goes wasted. Your evidence goes limp in the face of whim and obstinacy.

This is usually a timing problem (see the section "Not Just What, but When" later in this chapter). Generally, though, taking the time to explain your rationale will settle any nerves a stakeholder might have.

The benefits of education are far more valuable than the time it takes to do the educating.

Presenting

In your career, you'll get plenty of chances to explain your work—to *present* it—whether to your team or to other managers or departments. You might be in a conference room full of stakeholders who need to hear your proposed solutions. You might be on stage at a

web conference. You might be in a meeting with a lone design manager who just wants to see how a project is developing. No matter. You should seize every one of these chances. If you're not presenting your work, you're throwing it over the proverbial wall where other people get to misinterpret it and decide on their own that it's terrible. Be proactive; present it so that it is communicated on your terms.

If you can present your case well and do it up front, you don't need to argue about it or defend it later on. Your narrative will address every concern before it even comes up.

First, remember what I said in the chapter "Saying It With Purpose": It helps to apply an essay-like structure to your communication. When you're presenting work or ideas to someone, that college essay structure can be pure gold. Besides giving you a template to follow, it's a template that works well. It's time-tested. Its structure of thesis–support–conclusion tells a story to the recipient.

To learn more about a timeless storytelling technique that can help you structure all kinds of presentations, check out Christina Wodtke's blog post, "The Shape of Story".

Holding the Questions

A lot of times, your audience—likely a small group in a conference room—will want to ask questions along the way. This is fine if it's a minor question with a quick answer. But there's always the risk they'll nail you with something you need to think about or that a bunch of other people have an opinion about. When that happens, you can find yourself in the weeds in no time. There is no quicker way to derail a coherent argument than to let something like this throw you off.

The tip is pretty simple:

Ask people to hold their questions until the end.

Tell them to write their questions down so they can remember to ask them later, and that you'll get to them at the end. Tell them you've planned for that. Then stick to your promise; leave plenty of time for them to ask.

In many cases, especially if you've done your job well of anticipating what questions they'll have, you'll end up answering most of the

obvious questions as part of your presentation, making their questions a moot point. Even then, it's practically a guarantee that someone will ask you something you haven't yet considered. This is what Q&A is for.

If your meeting is an hour, leave 10 minutes at the end for questions. If you're running more of a feedback or review session, leave more time.

Whatever the case, leave time. Questions are the only way you'll know what you've missed.

Backing It Up

One of the most convincing things you can bring to a conversation is *evidence*.

It can come from a lot of places. It can be a study you read. It can be personal observation (which becomes more credible the more experienced you get). It can be the results of a study. It can be data you recalled from a previous project that involved a similar situation as the one you're facing now. It doesn't really matter where it came from as long as it's credible, its conclusions are relevant, and you can connect the dots between the evidence and your current project.

Evidence has, of course, several major benefits. For starters, it means you can make a case to *yourself*.

More from *Experience Required*:

> It can become really easy over time to accept your previously learned lessons as standing truths. Not only that, but universal truths. You think that thing you leaned five years ago is not only still true, it's true in every situation. Let's say, for example, you once witnessed five people in a row ignore a line of instructions on a form. It's easy to believe it will keep happening. But as you work through more and more projects, you'll need to test this notion many times. A line of instructive text isn't just a line of instructive text. It' s a font. It 's a color. It's something a user can skip past, stuffed between two fields in light gray and a 6pt font.
>
> It's a visual element as much as it is text. A user's bypassing of it is no sign of universal truth. Moved to a different spot, given a different color, its font size bumped up a notch or two, it could be just what the user needs. Displayed inside a big purple box right next to the form as the user clicks the field it relates to, it could be impossible to ignore. In a different application—one where the information

the form asks for is complicated and needs to be looked up—it could be consistently sought out. Every standard has exceptions. Designers tend to think users avoid reading while they're performing tasks. But it's certainly not always true. And if you're working on such a project, ignoring this fact could be a major detriment. Gathering some data on your assumptions early on can mean big differences in your design's effectiveness.

No matter how much you believe something, data can prove you wrong. Every suspicion, every assumption, every guess can be validated or debunked with a little research. And the last thing you want is to recommend a false truth. When you feel like you know you're right, take some time to make sure. See if the studies covered in articles online still back you up. See if the data you have access to can verify your belief.

I can't tell you how many times—especially early on in my career—I learned something only to unlearn it later. The older I've gotten, the more I've recognized that there are no hard and fast answers. Every decision you make is a guess. Your job is to mitigate the risk of that guess as much as you can. If you're unsure about something—if you are sure about something—find evidence to prove it out, one way or another. This will give you a great deal of confidence about your recommendations. All you have to do to convince someone after that is relay the facts.

Hence, my second point:

Data helps you make the case for your recommendation to everyone else, especially after you've vetted it yourself. Remember: This book is about how to lead as a UX professional. If you're out front with all the facts in your hand, and you've considered your recommendations, and you can demonstrate their validity, people will believe you. They'll believe in you.

Finally, putting evidence up for examination with every recommendation you make will build your reputation over time. It'll become easier and easier to get past whatever obstacles you face now. The objections. The politics. People will learn they have a trustworthy source of reliable, accurate information in their midst, and they'll come to rely on it rather than their guesses.

This will take a long time. But if you stick to it, it will work.

Always point to evidence. Always have evidence to point to.

Do What You Can

Communication is always incomplete. Short of climbing inside a person's brain, you'll never get all of what they mean to say, nor will you ever be able to fully predict all of what you need to say to be 100

percent understood. But holding yourself accountable for as much of it as you can will certainly change the nature of your conversations. If you walk into every interaction with the idea that it is your job to be complete, your job to understand the recipient's needs as well as your own, and your job to receive information as well as you deliver it, you will put yourself in a position to lead.

Don't take this lightly. Every business depends on communication. To be good at anything, you'll need to be good at this.

If you can think clearly, you can communicate clearly. And if you can communicate clearly, you can lead. Because communicating well buys you something you can't get in any other design deliverable:

Trust.

CHAPTER 8
Being Unreasonable

I want to wrap up here with an idea that's not just about being a good manager, but about being a good human. It starts with this quote:

> The reasonable man adapts himself to the world. The unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man.
>
> —George Bernard Shaw

The version of reasonableness Shaw is talking about here isn't the kind that refers to getting along with people or respecting your coworkers or handling the daily challenge of internal politics and bosses. He's not suggesting you become unreasonable to all of those people. You need to keep working with them. It would be a huge mistake.

Shaw is suggesting, quite logically, that achieving something better—a better product, a better team environment, a better company—means setting a much higher bar than the bar you have now and then relentlessly working toward it.

There's a certain amount of stubbornness in the idea, and necessarily so, but I doubt Shaw is suggesting you become difficult to work with, the kind of person who digs your heels in and clenches your face muscles and glares at people, refusing to budge. I think he's suggesting we sort out how to make other people want to get to the new place as much as we want to get there ourselves.

This isn't about being difficult. It's about believing in something and striving for it, even when—*especially* when—no one else can imagine it.

This is how companies are born. It's how products are invented. It's how laws are enacted. It's what allows teams to do great work. A bold vision for a better future and an unreasonable dedication to achieving it is how all real progress is made.

Without this kind of dedication to worthwhile beliefs, we wouldn't have the Internet. We wouldn't have space shuttles. We wouldn't have seat belts. The United States wouldn't exist; it was formed, after all, in resistance to British rule. Few things demonstrate more unreasonableness than forming an entirely new country in pursuit of better government.

On a day-to-day level, as managers, unreasonableness is what can drive us to form, run, and treat our teams with a higher regard than a mere schedule-pusher, task-assigner, project-runner. It's what can make us *better*.

It also happens to have a few great benefits. Because being unreasonable means having high standards, and high standards make all sorts of things happen.

The Advantage of High Standards

One of the biggest obstacles to great work—the kind you can be proud of, the kind that actually does let you go home feeling good about yourself—is low standards. It's easy to be run-of-the-mill. To be mediocre. To do just enough. And when you do all that, you never really achieve great work.

Mediocrity is nothing you want to put your name on, nothing you'll be remembered for achieving.

High standards, on the other hand, help you to get past obstacles, like people and politics (more on this in a minute). They help drive you to become better. They help drive you to the people who can help get you into better situations and projects.

This happens in three simple ways.

High Standards Lead to Prowess

Earlier, I described the benefits of skill overlap, and the benefits of having some level of expertise in at least one subject. I described how this helps your team and how it helps them respect you as a manager.

Assuming you've internalized this idea and now want to make it a goal of yours, high standards are the trick to getting there. The reason?

A devotion to high standards is a powerful force. It will push you past the typical lessons and drive you to relearn the meaning of quality.

It works in a process.

Early in your career, everything is new and interesting, so anything that even resembles your view of *quality* looks like something you should aspire to. And achieving that level of quality is really difficult, because you have little skill, little knowledge. You're a long ways off from being able to do the great work that inspires you about other people.

Slowly, you learn more, you do more, and you gain context, and insight, and start to form opinions. You also start to see that what you thought was amazing before is in fact fairly routine and typical, and the really great stuff is actually much more advanced and requires much more skill and experience. In other words, your target for greatness moves further away. Rather than leaving you disheartened, though, this fact motivates you.

As you build up skill and gain experience, you slowly become competent. Perhaps not much more than that for a while, but at least *competent*. If you have a certain knack for whatever it is you're doing, maybe you evolve more quickly. Still, it takes time to gain any level of real mastery. (Malcolm Gladwell, in his book *Blink* (Back Bay Books, 2007), posits that mastery takes something around 10,000 hours. Talent can speed that along, I think, but it's not as common as you'd think; *talent* is often another word for *years of practice*.)

Eventually, your standards go way up. And up again.

Assuming you *care* about the work you're doing, the target will keep moving, but each time, your commitment to quality will move you

forward. It will be the thing that gets you to ask questions, to envision something better, to try to achieve it.

As a result of all this, while you aren't even looking, you'll gain more *skill* than you ever thought you'd have. At each step, you'll have learned more, done more, and become more adept. High standards lead to prowess.

As a human and as a manager, you'll feel better about yourself and your work, and your team will recognize both.

High Standards Make You More Persuasive

Another way high standards help is by making you more persuasive.

Internal politics can bury you on any project, and politics can come in so many forms that you don't always even know which direction they're coming from. You might need to deal with a stakeholder who's always trying to "manage up" to his boss. You might encounter an exec whose whims are treated as more important than good, well-researched decisions. You might encounter a peer trying to take over one of your projects and introducing a lot of bad ideas in the process.

Sticking to your guns—committing to high standards—means you'll find ways to work through the situation. Having such standards will compel you to ask more questions, do more research, and show more evidence, all in the interest of producing a better outcome.

Couple that drive with the advice in the Debate School chapter, and you'll become a driving force, able to take any great idea from concept to completion.

High Standards Lead to People

And finally.

You know those people who are really great at their work and who inspire you with their intelligence, cunning, insight, all that great stuff?

High standards lead you to them. And this is arguably the most exciting effect of being unreasonable.

In the beginning of your career (presumably long before becoming a manager), your high standards can hinder you. The tendency for a

lot of people is to see what's possible but have no idea how to achieve it other than to wave their arms and complain a lot. This doesn't get anyone anywhere. If anything, it just makes people resent you. I can't begin to count how many managers I've met who've uttered the words, "Bring me solutions, not problems." They say this because they've all met more than their share of newbie professionals who only know how to complain. It takes a while to sort out how to have high standards *and* lead people to pursue them along with you rather than getting grouchy when they don't. It takes experience and skill and knowledge. As you gain those things, high standards start to become a positive.

Eventually, you start to have good ideas. You start to share them, to learn to speak up without stepping on toes, to work toward solutions.

Eventually, people start to ask for your opinions more frequently. You become the person whose thoughts matter most on a particular subject, someone who knows the most about it and has a track record for handling it well.

Keep that up and you'll get recommended for things.

Those people you respect will hear about you on their own. You'll end up working with them. You'll become *one* of them.

This happened in my career. I started as a complainer, a guy who could spot a design problem from a mile away, but who had no idea how to make it better. As I learned, I built up a record of successes. I began to write about them. This led me to conferences, which led me to all the people I respected and admired most. In the years that followed, I got to work with many of the best minds in my industry. I got to have long conversations with them about their approaches and their experiences and their insights.

It's that simple. High standards move you toward *other* people with high standards, and they attract people who have high standards right back to you.

Can you see why this would matter in a managerial context?

Passion like this is contagious. Your belief in *better* can make people want to come along with you, to be *part* of the better.

When they are part of the better, they feel more motivated. When they are more motivated, they do better work. When they do better work, they feel better about their lives and their work.

This is what it means to lead by example.

Have passion, get passion.

Long-term, this will put you and your team in high demand. It will make you successful.

Embrace the highest standard you can imagine and aim for it. If you do, your career could be wilder than you ever expected.

CHAPTER 9

The Philosophy It All Adds Up To

I've hit on a lot of themes in the preceding chapters. To close out, I'd like to take a moment to extrapolate those themes and piece them together into the philosophy I promised you in the introduction.

It has just as few major tenets:

1. Be the same person as manager that you were when you aspired to become one—the hungry kind, the kind who collaborated and always wanted to learn more and get more involved.
2. Don't manage people, manage things *away* from people. Don't ask people to do good work; they already *want* to do good work. Clear the path so that they can. Don't tell them what you need; ask what they need from you.
3. Communicate well (and frequently). Listen, ask questions, clarify, restate, absorb, and contribute. And when it's time to talk, present your ideas rather than leaving them open to interpretation.
4. Always consider and be able to explain how today's actions map to tomorrow's outcomes. This shows higher-ups that you're thinking about the business, and it lets your team keep the overarching vision in mind so that they can contribute in a meaningful way. It helps your team connect their work to the company's goals, which is a major factor in feeling like they have *purpose*. It will also help you stay calm and measured as a manager; your eye for the long-term reduces the frustration of distractions.

5. Don't just get out of the way. *Lead*. Work with people to establish direction, communicate that direction (frequently), and then support everyone's efforts to get there.

About the Author

Robert Hoekman Jr. has authored several books and dozens of articles for a range of publications, including *Fast Company* magazine's "Co.Design" blog. He is a columnist for the revered motorcycle culture and lifestyle magazine Iron & Air, where he is also a contributing editor. He has spoken to packed rooms at dozens of events all over the world.

Robert's talents for questioning and challenging conventional wisdom have earned him success in a myriad of professional interests, including design, product strategy consulting, freelance writing, editing, and public speaking. As a veteran of the web industry, he was among the few who practiced User Experience before it became a household term, and is widely considered to have written several of the profession's defining guidebooks.

Learn more about Robert at *rhjr.net*.