

MISSION ■ BUILT

V2.0 · SECOND EDITION

■ THE FIELD GUIDE

MISSION BUILT.

— A FIELD GUIDE FOR BUILDING
THINGS THAT MATTER.

BY
MIKE NICHOLS

EDITION
SECOND · 2026

||| SECOND EDITION · 2026 |||

MISSION BUILT.

A FIELD GUIDE FOR BUILDING
THINGS THAT MATTER.

MIKE NICHOLS

Mission Built: A Field Guide for Building Things That Matter

by Mike Nichols

Copyright © 2026 Mike Nichols.

L I C E N S E

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

creativecommons.org/licenses/by-nc-sa/4.0

F I N D T H E L I V E E D I T I O N

missionbuilt.io

C O N T A C T

mike@missionbuilt.io

SECOND EDITION · 2026

CONTENTS

00	PROLOGUE	BUILT FOR MORE
01		MISSION BEFORE METRICS
02		BUILT THROUGH REPS
03		RITUALS OVER RULES
04		FEEDBACK IS A SUPERPOWER
05		PROGRESS ISN'T PRETTY
06		THE MISSION DEMANDS RECOVERY
07		TRAIN THE ENGINE, NOT JUST THE OUTPUT
08		DECISIONS ARE MADE UNDER LOAD
09		SHIP IT LIKE YOU SHOW UP
10		THE TEAM IS THE TOOL
11		STRONG ENOUGH TO LISTEN
12		THE WEIGHT IS REAL. OWN IT.
13		AI IS THE NEW OS. THE MISSION IS THE ONLY APP THAT MATTERS.

A NOTE ON THE SECOND EDITION

If you read the first edition of *Mission Built*, welcome back. A few things changed.

Most visibly, the second edition adds a chapter. Chapter 13 — *AI Is the New OS. The Mission Is the Only App That Matters.* — exists because 2026 forced the question. AI raised the floor of average work in a way no tool ever has. The book's central argument — that mission, reps, feedback, recovery, and ownership are what produce real strength — needed to be tested against the loudest tool of the year. So I tested it. Twice. Once under the bar, once in the boardroom.

A few other additions you'll notice. The Drift, named once in the first edition, now recurs as a deliberate framework — Boeing's Drift in Chapter 1, recovery-shaped Drift in Chapter 6, legacy-shaped Drift in Chapter 12, and the AI-shaped Drift in Chapter 13. Three appearances of an idea is when it becomes a reader's framework. Four is when it becomes the spine of how you think about work.

The give-a-shit motto now has its source on the page. Marine squadron VMM-364, the Purple Foxes. CASEVAC pilots flying into enemy fire to bring wounded troops home. Three words on their patch. The motto came to me through Nate Fick at Endgame, and it has been the working motto of my career.

Some examples are paired with 2024–2026 updates. Boeing's 737 MAX with the 2024 Alaska Airlines door plug. Cyberpunk 2077 with the Humane AI Pin. The principle holds. The evidence is current.

The book is shorter than the first edition by about ten pages, despite the new chapter. The prose was tightened. Less padding, more spine.

What didn't change: the voice, the mission, the give-a-shit. Those are the work.

— Mike

WHO I WROTE THIS FOR

This book is for anyone who builds things.

GMs and PMs trying to figure out what matters most. Engineers who want to know why the work is what it is. Designers who care more about the user than the demo. Leaders trying to lead instead of manage.

If you spend your days deciding what to ship and how, this book is for you. It is not a methodology. It is not a framework you have to adopt. It is a field guide for the why beneath the what. For the principles that hold when the deadline shifts, the strategy changes, the tooling evolves, and the team turns over.

I am also a powerlifter and a coach. I have spent more than a decade under the bar. I lean on that world throughout this book, but not because I expect you to lift. The gym is one of the cleanest mirrors I know for the work we actually do in product. Honest input. Reps that matter. Systems tested under load. The bar is just the example. The principle is the point.

It's also a leadership book, in the way the best leadership books are: a collection of specific stories about what actually mattered when the work got hard. Endgame. Elastic Security. The military. The barbell. The CAOC at Nellis. Lived examples, not borrowed wisdom.

Most of all, this is a book about empathy.

That word has gotten lighter than it should be. We live in a time when caring for others gets called weak. When standing up for someone besides yourself gets called foolish. When the loudest voices argue that strength is what you take and softness is what you give away.

I disagree. I have spent my whole career proving the opposite. In uniform. Defending networks. Building products. Coaching lifters. The same lesson, every time. Real strength is what you give. The mission lives in the people next to you. The user is on the other end of every decision. Empathy isn't softness. It's clarity. It's what makes the work worth doing in the first place.

If that resonates with you, this book is for you.

If it doesn't, I hope it changes your mind.

— Mike

BUILT FOR MORE

July 2016. The Combined Air Operations Center at Nellis Air Force Base. Red Flag is about to begin. My engineering lead and I are standing silent on the floor. Our agent is deployed on systems across the room. Pilots are about to be in the air. Controllers will guide them. Air defense will track them. A crashed laptop in a startup is annoying. A crashed laptop here is a different category of problem entirely.

We had bet the company on this moment. Six months of rebuilding every layer of the product. Front end language. Agent communication. Back end architecture. Six months of late nights, early mornings, and days that bled together. Six months of couches, conference room floors, and pizza at 2 a.m., for one customer, in one room, on one exercise we could not afford to fail.

That moment is what this book is about.

Before the titles. Before the roadmap, the metrics, or the barbell plates neatly stacked on a rack. There was the mission.

I've worn uniforms and hoodies, sat in war rooms and boardrooms. I've served with soldiers and shipped with engineers. And the one thing that always stood out — the one thing that always worked — was giving a shit. About the work. About the people. About the outcome.

That motto belongs to VMM-364 — the Purple Foxes, a Marine tiltrotor squadron whose pilots fly CASEVAC into enemy fire to bring wounded troops home. Three words on their patch. They have run my life ever since. You will meet them in their full weight in the chapter on AI.

This isn't a memoir. It's not a manual.

It's a field guide for anyone trying to build something that lasts, whether that's a body, a product, a company, or a life with integrity.

You won't find shortcuts here. Just reps. Principles. Stories from the weight room and the war room, forged in pressure, refined in reflection.

Real strength doesn't come from just lifting the weight. It comes from knowing why you're lifting it.

■ CHAPTER 01

01

MISSION BEFORE METRICS

THE MISSION IS THE MAGNET

Before I ever built products, I served in the U.S. Army as an Airborne intelligence sergeant. That experience of working in service of something larger than myself, of making decisions under pressure with lives on the line, shaped everything that came after. I learned early that mission comes first. Not ego. Not recognition. Mission.

That mindset followed me from the military to cybersecurity, and into leadership roles where the stakes changed but the values stayed the same. Whether it was securing critical systems, helping build Elastic Security, or coaching in the gym, the goal has always been the same: real strength is lifting others.

One of the proudest chapters of my product career was building Endgame. We entered a brutally competitive market, going head-to-head with massive players like McAfee and CrowdStrike, and carved out real ground. Not because we had more money or brand recognition, but because we had something harder to copy: a clear mission. Protect high-value targets from nation-state level attacks. That focus, and the small, fierce team who rallied behind it, made all the difference.

At the heart of it was Nate Fick, a Marine officer turned tech CEO, and later the U.S. Ambassador for Cyberspace and Digital Policy. He led with conviction, clarity, and a deep respect for the mission. In all-hands meetings, Nate would remind us that we were an "elevator asset company" — that if the building burned down, the most important assets could still fit in the elevator. It wasn't the code or the tools. It was the people: the ones who understood the user's mission and had the passion to serve it.

That idea stuck with me. Nate's example reinforced what I learned in uniform: the success is the user's success. Your mission is their mission.

Nate also brought something else with him out of uniform — a motto from the Marines. Three words from the patch of VMM-364, the *Purple Foxes*, a tiltrotor squadron whose pilots fly CASEVAC into enemy fire to bring wounded troops home. *Give a shit*. About the work. About the people. About

the outcome. It became the working motto of my career, and it has been the spine of everything I have built since. We will come back to it in full in the chapter on AI.

Metrics are the outcome of making your user successful. Yes, we need to measure them. But they are the result of serving the mission, not the reason for it.

There's a moment in every product meeting when the question slides in like it always does: *"How will we measure success?"*

It's a good question. Just not always a good first question.

In lifting, it's the same story. People chase PRs every week like the number on the barbell is the whole point. Add five pounds. Hit record. Post the clip. Repeat.

But metrics without mission? That's just noise. Pressure with no direction. Goals with no guts.

"The weight on the bar isn't the goal. It's the evidence of progress, not the destination."

We've all seen what happens when this mindset takes over. It's not a failure of talent. It's a failure of alignment. *Cyberpunk 2077* didn't initially flop because the devs didn't care. Far from it. The development team poured years into building something ambitious. But the pressure to hit a holiday launch window, a decision made at the executive level, overrode the mission of delivering a complete, polished experience. The result was a rocky release, millions in refunds, a reputational hit, and a stock crash.

To their credit, the team stuck with it. Years later, after patches and a reimagined DLC, the game has earned back much of the trust it lost. A testament to what happens when talented people are finally given the space to do the work right.

But not every team gets the chance to earn trust back. Humane raised over \$230 million to ship the AI Pin in April 2024, a wearable that was supposed to replace your phone with voice and laser projection. The reviews were brutal. Battery life, overheating, latency, hallucinations. The mission of building something a user actually wanted got buried under the mission of shipping what investors had already paid for. Less than a year later, HP bought the

company's IP for a fraction of its peak valuation and shut down the servers. Every Pin in the wild became a brick on the same day. There was no mission left to recover.

And we've seen what it looks like to protect the mission, even when it means stepping back. Simone Biles did exactly that in front of the entire world at the 2021 Olympics. Under unimaginable pressure, she chose long-term purpose over short-term performance. She knew something was off, and she honored that instinct.

"I have to focus on my mental health... if you don't, then you're not going to enjoy your score and you're not gonna succeed as much as you want to." — Simone Biles

Her move wasn't retreat. It was leadership. And in time, she returned to competition stronger, on her terms, and more respected than ever. A different kind of comeback. One powered by mission, not metrics.

Success isn't about hitting every metric. It's about refusing to lose yourself trying.

MORE THAN JUST GOOD INTENTIONS

Mission-driven isn't a poster in the break room or a bullet in a pitch deck. It's how you move. How you decide. How you show up when it's hard.

In a world that celebrates velocity, mission is quiet. But that doesn't make it weak. Mission gives you clarity when things get blurry and stamina when things get hard.

Patagonia famously ran a campaign telling customers not to buy their jacket unless they truly needed it. SpaceX targets goals across decades, not quarters. Mission alignment shows up in the timelines a company is willing to commit to. And it pays off. Research shows people who believe in the purpose behind their work stay longer, burn out less, and deliver more.

WHEN METRICS ECLIPSE MEANING

Let's be clear: metrics matter. But only when they serve the mission, not when they become it.

Here's where teams lose the plot:

- They ship fast instead of shipping right.
- They chase signups instead of learning why users leave.
- They brag about launches and ignore long-term usage.

When dates drive development, teams start cutting corners. Feedback loops close. Energy fades. You build momentum toward a number, not toward value.

In the gym, this is ego lifting. In product, it's just as risky. It looks like burned-out engineers, brittle systems, and growth that collapses the second you stop pushing it uphill.

THE FULFILLMENT FLYWHEEL (POWERED BY PURPOSE)

There's a better model. One that's as relevant in combat as it is in code, or in the squat rack.

It's called the OODA Loop: Observe, Orient, Decide, Act. Developed for fighter pilots, adopted by startups, powered by clarity.

Without a clear mission, the whole loop spins out. With mission, observation knows what matters, orientation filters cleanly, decisions line up with the work, and action carries resolve. Without it, you collect noise, over-prioritize the wrong metrics, chase vanity wins, and execute half-heartedly.

In lifting, that means trusting the plan instead of maxing out because you feel good that day. In product, it means waiting to ship because your users aren't ready, even if your OKRs are.

Mission turns chaos into clarity. It makes every rep count. Every release matter. Every decision directional.

This is the real flywheel of fulfillment:

Mission fuels clarity. Clarity powers resilience. Resilience drives real progress. And real progress reinforces the mission.

That's the through-line. That's what keeps us going.

Metrics follow. But the mission leads.

THE DRIFT

At first, the metric is a mirror. It reflects the mission. Clean, focused, and true.

But over time, something shifts. Quietly. Gradually. The mirror warps. And without noticing, we begin steering toward the reflection instead of the road.

This is the Drift.

It doesn't announce itself. It rarely arrives with bad intent — it begins with a goal:

- Launch the feature by Q4.
- Hit 405 on deadlift.
- Increase MAUs by 20%.

Reasonable. Measurable. Actionable. These are the numbers we hold up as evidence that we're making progress, and for a while, they are.

But then: the goal becomes the game.

In product, we've seen it time and time again. Shipping becomes more important than solving. Teams crunch to meet a date set quarters ago, long after the user problem has evolved. Metrics chase headlines. Investors want news. Executives want motion. The dashboard glows green while user trust fades red.

Think of Boeing. In the race to beat Airbus, they needed the 737 Max on runways fast. Shareholder pressure mounted. Deadlines became immovable. On paper, the metrics looked great: deliveries met, costs controlled. But beneath the numbers, safety systems were skipped, warnings dismissed.

Two crashes. Hundreds of lives lost. The drift wasn't just technical, it was cultural. Speed overtook scrutiny; the metric overtook the mission.

And here is the hard part. The Drift does not heal on its own. In January 2024, a 737 MAX 9 lost a door plug at 16,000 feet on Alaska Airlines flight 1282. No one died, but only because the seats next to the missing door were unoccupied. The plug had been reinstalled at the factory without the four bolts

that were supposed to hold it in place. Same company. Same speed-vs-scrutiny pressure. Eight years after the first MCAS crash, the cultural drift was still doing its work. The Drift, once it sets in, takes a generation to clear.

In strength, the signs are physical, and brutal. Hafthor Björnsson, one of the strongest men in history, returned to powerlifting after two years of boxing. He set his sights on breaking the raw total world record. Big lifts were stacking fast. But recovery lagged behind. He felt the warning signs — tightness, fatigue — but kept pushing. He wasn't recovering between sessions, and as he later admitted, not listening to his body. Then came the snap. Attempting a 556-pound bench press, his pec tore clean off the bone.

The Drift had found him too. Progress misaligned with process. Recovery sacrificed to reach a number that no longer served the mission. The result was pain, delay, and a lesson carved in scar tissue.

The damage isn't just physical. It's psychological. Because when we treat metrics as the mission, missing them feels like failure. And so we hide the truth, inflate success, or worst of all, stop trying.

The Drift corrodes not through force, but through inversion. It flips process into performance. It turns care into compliance.

And it burns people out.

I've worked with brilliant people who could have built anything, but they left. Not because they failed, but because the system stopped valuing why they showed up in the first place. When we reduce contribution to a dashboard, we forget the soul behind the keyboard.

Research shows mission-driven employees are significantly more loyal. A LinkedIn survey found mission-motivated employees were 54% more likely to stay with their company for five or more years; companies with purpose-driven programs saw 52% lower turnover among newer employees.

So what's the antidote?

Not to ignore metrics. To anchor them. Use them as signal, not steering. Build systems that reinforce why we do the work. The real goal isn't a launch date or a deadline. It's built through the reps, not measured by them.

REPETITION WITH INTENTION

You don't get strong by lifting heavy once. You get strong by showing up again. And again. And again.

But if you do the same thing forever, you don't get stronger. You get stuck.

That's the tension of progress: it demands ritual, but it punishes repetition without variation.

Whether in the gym or in product, growth comes through iteration. Small, deliberate cycles of effort. You log your training. You tweak your form. You try a new tempo. In software, you ship. You measure. You talk to the user. You try again. The process isn't glamorous. But it's generative.

Rituals compound. Rules confine.

A rule says *"do this."* A ritual says *"do this because it matters."* One is brittle. The other bends with you.

Agile isn't magic. It's a set of rituals: standups, retros, demos, sprints. They only work when they're anchored in meaning and adapted to context. Treat retros like checkbox theater and you're looping, not iterating. If your standup sounds like *"I did stuff, I'll do stuff, no blockers,"* you've got a ritual without reason.

The same failure shows up in the gym. People find a powerlifting program online and follow it to the letter, even when it doesn't fit their recovery, their age, their job, their life. They wonder why they plateau, or get hurt. It's because they forgot the principle beneath the plan.

That's where most systems fail. Not because the framework is flawed, but because the user is forgotten. They're treated like an input to a method, instead of the reason for its existence.

Let's be honest: there's no shortage of books telling you how to do things. And yes, here we are, writing another one.

But this isn't a blueprint. It's a philosophy. The implementation is on you.

What works for one team, one lifter, one body, won't work for another. That's not failure. That's reality.

Stefi Cohen didn't become a record-breaking lifter by blindly following a single powerlifting template. Her training evolved through experimentation: strength phases, hypertrophy cycles, hybrid athletic blocks. She didn't just lift heavy. She studied her feedback, varied her stimuli, and rewrote her rituals to fit her mission. Progress didn't come from rigidity. It came from rhythm and reinvention.

In product, that same spirit of experimentation lives in hack weeks. Short, sacred breaks from routine where teams can stretch in new directions.

Atlassian, Spotify, and Elastic have all embraced this. At Elastic, hack weeks became a celebrated ritual — not just for innovation, but for joy. Some of Elastic Security's most creative features were born not in sprint planning, but in that carved-out chaos.

Sometimes, the best way to realign with the mission is to deliberately step outside it.

Rituals work when they're shaped by the user, not imposed on them. They're tools, not commandments.

And when they're working, you feel it. Not just in output, but in outlook. A team that trusts its rituals doesn't need micromanagement. A lifter that trusts their program doesn't need motivation hacks. The rhythm carries you. The meaning sustains you.

But rituals aren't static.

They must flex to your fatigue. They must bend to your bandwidth. They must evolve, not just to avoid boredom, but to stay honest with the mission. Same input, same outcome. If the goal has changed, so must the reps.

That's what separates the lifter who grows from the one who stalls. That's what separates the product team that adapts from the one that burns out.

Because if the mission is the magnet, rituals are the rails.

They don't tell you where to go. They keep you from sliding off the path while you find it.

Rituals give you direction.

But progress? That comes from the work itself. The grind, the reps, the effort no one applauds.

■ CHAPTER 02

02

**BUILT THROUGH
REPS**

THE MYTH OF OVERNIGHT SUCCESS

We've all heard the stories. The product that "took off overnight." The lifter who casually pulls four plates like they've always been able to. The founder in a garage who changes the world with a single keynote.

It's tempting to believe that mastery happens like that. In a flash.

But that's not really how it works.

Take the iPhone. The 2007 reveal made it feel like it dropped from the sky fully formed. Behind that moment were years of trial and error: touchscreens that didn't quite work, software that crashed mid-demo, a mountain of prototypes that never saw the light of day. The final product wasn't a stroke of brilliance. It was relentless iteration.

Same with Instagram. What looked like an overnight success was actually the second version of a too-close-to-Foursquare check-in app called Burbn. The team just kept listening, trimming, trying again. One update at a time.

Or Cursor. The AI coding tool that seemed to dominate developer feeds in 2024 was the second product from Anysphere, a small team that had spent the previous two years pivoting from a different idea entirely. The "overnight" moment was years of quiet work on a forked IDE before anyone outside the team noticed.

That's a story I've lived from the inside. In July 2016 at Endgame, we shipped a product into a live Air Force exercise after six months of rewriting every layer of the platform. Front end language. Agent communication. Back end architecture. All of it, for one customer in one room, on one exercise we could not afford to miss. The press release made it look clean. The reality was conference room floors, 2 a.m. pizza, and a team that never went home. Polish hides process. We will come back to that exercise in the chapter on shipping. The grind that built it belongs here.

And that big deadlift on your feed? Probably not magic. More likely: a hundred quiet sessions, some of them rough. A lot of small choices to show up, tweak form, trust the program.

We love the idea of sparks. Fast wins, big leaps, sudden breakthroughs. But progress usually doesn't feel like that. Most of the time, it feels a lot more like repetition.

The engineer fixing the same piece of code, again. The founder rewriting their pitch for the fifth time. The lifter doing the same warm-up cues every session, no matter the weight.

It's not flashy. But it adds up.

That's part of what makes the first few months — or the first year — so exhilarating. In the gym, you make gains almost every week. In a startup, your product evolves daily. Everything feels fast, and the feedback is loud.

But then you hit the plateau.

The easy wins dry up. Strength doesn't come as quickly. Users get harder to surprise. Suddenly, it's not about chasing sparks anymore. It's about showing up and pushing through.

This is where the real work begins.

In lifting, that means grinding through the middle — not maxing out, but mastering the basics under load. In product, it's the shift from scrappy innovation to enterprise-grade reliability. Less fanfare, more focus. It's the part that actually defines mastery.

It's not about more weight or more features. It's about honing your form. In lifting, that might mean dialing in your brace, fixing a subtle shift in your squat, finally feeling your lats engage in a deadlift. In product, it's refining an initiative until it truly solves the right problem.

Reps aren't just about volume — they're about attention. That's what makes progress sustainable.

REPETITION IS NOT REDUNDANCY

In Section 1, we looked at the myth of overnight success — how real strength, in the gym or in product, is built through showing up again and again. But not all reps are created equal. The real magic isn't just in the repetition. It's in how you use it.

Not all reps count the same.

Anyone who's coasted through a gym session, just moving weight, knows the difference. One set burns time. Another builds awareness, precision, and strength, even if the numbers don't change.

The same is true in product.

Repetition isn't redundant when it's intentional. When it's used to test assumptions, refine interfaces, tune performance, or build something just a little closer to what users actually need. That's the kind of rep that moves things forward.

But repetition alone isn't enough. Variation is what makes repetition transformative.

In strength training, that's where Renaissance Periodization's framework lives. Dr. Mike Israetel teaches a structured progression — Minimum Effective Volume (MEV), Maximum Adaptive Volume (MAV), Maximum Recoverable Volume (MRV) — that lets lifters build under load without burning out. The concept isn't about working hard. It's about working smart, within boundaries that adapt to you.

In training, we use blocks for different goals: hypertrophy to build muscle, strength to build expression, deloads to recover and grow. We manipulate movement, load, and tempo to avoid stagnation. A paused bench press, a deficit deadlift, or chains added to a bar — each variation targets a different adaptation, even if the base movement stays the same.

Product is no different.

You might run a block of sprints to achieve a specific outcome: refining onboarding, improving performance, or tackling tech debt. Then shift into a new block with a new goal. Like training, it takes enough intentionality to complete a cycle, maybe one, two, or three sprints, before varying the focus. Progress comes not from flailing in every direction, but from focused variation over time.

And just like strength doesn't grow without tension, product insight doesn't grow without diverse input.

You can't just loop with your own team forever. Sometimes you have to stretch — to talk to sales, to customers, to analysts, to skeptics. Even when it's uncomfortable. Especially when it is.

That's the principle behind Frans Johansson's *Medici Effect*: stepping into an intersection of fields, disciplines, or cultures lets you combine existing concepts into extraordinary new ideas. In the gym, that might mean lifting with people stronger or different than you. In product, it means bringing in voices from outside your echo chamber — support calls, sales objections, the person who almost churned.

Reps alone build endurance. Smart variation builds power.

This is how you break through the plateau. Not by abandoning the reps, but by evolving them.

WHEN THE SPARK FADES

The first reps are easy. Not physically, but emotionally. You're fired up. Everything's new. Progress is loud and obvious.

But what happens after the rush?

In lifting, it's the long middle. The early PRs stop coming. Your form stalls. You show up, grind through the same sets, and wonder if you're actually moving forward. You're not always chasing your one-rep max. And you shouldn't be. Strength isn't built by living at your limit. It's built in the space between peaks, when you train with intention and recover with discipline.

In product, it's the same. You can't always be doing the flashy, innovative thing. Sometimes the most important work is foundational: fixing backend debt, improving accessibility, tightening up performance. The kind of work that creates capacity for brilliance later.

And for many, this is where the wheels come off.

Because motivation, that spark, is unreliable. It's not designed for the long haul. And it doesn't care about your goals.

That's why systems matter more than sparks.

Systems are how you keep showing up when the dopamine dies down.

Morning routines. Logbooks. Standups. Progress reviews. They don't need to be rigid. But they do need to be real. Reps don't get done by accident.

You don't need hype. You need structure.

In training, that structure might be a coach, a program, a calendar alert that says *"get under the bar."* In product, it might be a rhythm of sprint planning, async demos, or check-ins with customers. Externalized accountability is often the only thing that keeps momentum moving.

James Clear, author of *Atomic Habits*, wrote:

"You do not rise to the level of your goals. You fall to the level of your systems."

The work still has to be done. But when your environment supports your actions, it gets done more often.

And here's the quiet truth. The people who make the biggest progress aren't usually the most intense. They're the most consistent.

They build when no one's watching. They train when it's not fun. They keep caring, even when the spark is gone.

But what if you need to find the spark again?

One of the fastest ways to reignite your drive is to reconnect with the people you're building for. Talk to your users, especially the ones who aren't shouting. You might think the work you're doing isn't flashy, but somewhere out there, someone is craving exactly what you're building.

Stability. Accessibility. Visibility. These aren't buzzwords. They're lifelines for users with real problems. Find them. Listen to them. Let them lift you up.

Because nothing recharges momentum like hearing someone say: *"This made my day better."*

And that brings us to the next section: the power of the quiet reps. The multiplier of boring work.

THE MULTIPLIER OF BORING WORK

There's a kind of work that doesn't make headlines. No one posts their warm-up sets. No one celebrates shaving 100ms off load time.

But that's the work that wins.

For every PR pulled in competition, there are hundreds of days of grinding behind it: submaximal sets, long pauses, light reps, mental resets. The same is true in product. Every effortless-looking release rests on a foundation of something much deeper. Months of planning, iteration, bug-fixing, and late-night Slack threads.

In lifting, it's the mobility work you do alone at 6 a.m. The back-off sets you don't skip. The deload week you take seriously. It doesn't look impressive. But it makes everything else possible.

In product, it's building out role-based access controls — not because it's exciting, but because your biggest customers expect it. It's mapping audit logs across services so your platform isn't a compliance risk anymore. It's the 10th conversation with a user about the same rough edge in the UX. These aren't "big bets," but they're the reason your big bets land.

That's what boring work does: it compounds.

Each rep you don't skip, each ticket you don't shortcut, each problem you refine instead of avoid — it stacks. Quietly. Relentlessly. And over time, it becomes your edge.

You don't need to go viral. You need to be trusted.

And trust is built in the boring work.

The warm-up that prevents injury. The small fix that prevents churn. The five-second improvement that gives a user five minutes back.

This is what separates the strong from the strong enough.

It's not what you do once. It's what you do without applause.

You do it for the growth. For the discipline. For the user whose day you quietly made better. Not for the accolades.

The new wrinkle, in 2026, is that AI can do some of the boring work for you. The release notes can be drafted. The audit log can be summarized. The user transcript can be tagged. That is real, and mostly good. But the reps that build *you* are not the ones a model can take. They are the ones you choose to show up for. We will come back to this in the chapter on AI.

That's what separates long-term success from short-term effort. Not glory, but the passion to do the work for its own sake. The features and the gains? They're just symptoms. What matters is the mission that fuels them.

That's why I'm so passionate about product management — and about lifting. Because the best PMs and the best lifters don't just show up for themselves. They show up for the team, for the user, for their own growth. Not in a selfish way, but in a way that elevates everything and everyone around them.

And if you've made it this far — through the reps, the plateaus, the quiet work — you already know:

This isn't just about shipping or lifting.

It's about becoming the kind of person, or the kind of team, that keeps showing up.

That's the real win. And that's where we end this chapter. Not at the peak, but at the foundation.

THE WORK BECOMES THE WIN

At some point, the reps stop being something you have to do. They just become something you do.

You stop chasing motivation and start trusting momentum. You stop asking when it gets easy and start asking how to keep showing up.

And that's the shift. Not just in the gym or in your sprint board, but in your mindset. You lift because you care. You build because it matters. Because you love it.

It's not about PRs or product launches. It's about what they represent: the hours you logged. The patterns you learned. The people you helped.

That's what it means to be mission built.

You're not doing it for the spotlight. You're doing it because you've seen the power of the process — and you're not walking away from it.

Progress doesn't shout. It stacks.

One quiet rep at a time.

■ CHAPTER 03

03

RITUALS OVER RULES

THE RITUAL IS THE RAIL

If the mission is the magnet, the ritual is the rail.

Rituals keep us on track. Not by forcing us forward, but by guiding our momentum. Like rails under a train, they don't power the engine, but they make sure we're headed in the right direction. They keep drift from becoming derailment.

In the gym, rituals aren't written on the whiteboard, but everyone knows them. The way you chalk your hands before a heavy lift. The nod from your training partner. The slap on the back before a max attempt. These moments don't make you stronger by themselves. But they signal that it's time to lock in. They tune your mind and body to the work ahead.

In product, rituals shape how we build and why. A startup begins with a weekly sync focused on shipping a differentiated capability. As the company grows, the rituals evolve: structured roadmap reviews, tighter standups, the shift toward shipping "boring" features. Role-based access control. Data retention workflows. RBAC and audit logging won't make the cover of TechCrunch, but they're what your biggest customers need to trust you.

These shifts aren't a loss of agility. They're a deepening of commitment. The rituals grow because the mission does.

The difference between a rule and a ritual is intention. Rules are imposed. Rituals are earned. A rule says *"follow this or else."* A ritual says *"this is how we move with purpose."* When teams stop questioning why they follow a process, it calcifies into dogma. But when the process stays tied to the mission, it adapts. It grows. It stays human.

Elite lifters know this instinctively. Even as their programming changes — higher volume, lower intensity, or a shift from competition prep to recovery — their rituals stay intact. The warmup flow, the music choice, the way they approach the bar. These things evolve, but they never disappear. They guide the body into readiness. They keep the lifter connected to something deeper than the day's numbers.

And that's the point.

You don't need more rules. You need rails that carry you toward the mission, and flex with the turns ahead.

The mission is the magnet. The ritual is the rail. Not to restrict your path. To keep you from drifting when the pressure builds.

LISTEN TO YOUR SIGNALS

Rails only work if you're paying attention to where you are on them. Rituals are no different.

You can't run the same program forever. Not in the gym. Not in product. Not in life. Progress demands feedback. And feedback starts with listening.

Lifters learn this early. You might show up ready to deadlift heavy, but your grip feels off, your back's tight, your CNS just isn't firing. (CNS, Central Nervous System, the part of your body responsible for strength output, coordination, and neural drive. If it's fatigued, you'll feel it, even if your muscles are technically rested.)

That's not failure. That's information. A smart lifter doesn't abandon the workout. They adjust. They keep the ritual, shift the intensity. Maybe you pause at 70%, maybe you pivot to accessories. Listening doesn't make you weaker. It keeps you in the game longer.

In product teams, the same truth holds: rituals without awareness become liabilities.

You can run the ceremonies — sprint planning, retros, standups — but if you're ignoring signals from your team or your users, you're performing process theater. Burnout doesn't show up in Jira. Disengagement doesn't flash red in a dashboard. You feel it in the delay before someone unmutes. In the tension after a roadmap shift. In the quiet attrition of both teammates and customers.

Nowhere is this more visible than in the game industry's long-standing reliance on crunch time. Late-stage death marches, where teams work 60, 70, 80-hour weeks to hit a ship date. Executives cite passion. Teams call it what it is: avoidable. The rituals of *"just one more sprint,"* of all-hands war rooms, of praise for pulling all-nighters — they're treated as signs of commitment, when they're actually signals of failure.

Failure to listen. To plan. To build sustainable systems. Failure to treat the team as human, not just headcount.

This is where product management has a second, often overlooked role.

Yes, product is the voice of the user inside the development team. But product is also the shield of the team against the wrong voices from above. A good PM doesn't just absorb pressure from the top and pass it down. A good PM pushes back. Uses data to say no to date-driven development. Advocates for pacing, not panic. Protects the team's ability to think, breathe, and build well, even when the deadline is loud.

Because our goal isn't to meet an investor's timeline or an executive's forecast. It's to build the best product to solve our user's mission.

And the only way to do that — sustainably, meaningfully, and well — is to listen.

To your body. To your team. To your users. To the mission.

CHANGE THE PATTERN, NOT THE PRACTICE

Sometimes listening reveals the ritual itself needs to change.

When things stop working, you don't abandon the ritual. You adapt it.

Progress isn't a straight line. It's a cycle of push, plateau, pivot. And when you hit that plateau — when the usual ritual no longer delivers results — it's not a sign to quit. It's a sign to evolve.

In the gym, this happens all the time. You run a successful squat cycle, adding five pounds a week like clockwork. But then you stall. Your knees cave, your speed slows, and the bar starts winning. The answer isn't to scrap the movement. It's to change the pattern. Maybe you shift to pause squats, change your stance, or drop the volume to focus on recovery. The ritual — showing up and squatting — stays. The shape of it changes.

The same holds true in product.

The standup that once helped your team sync becomes a box-checking chore. The planning meeting that used to set priorities now spirals into status updates. That doesn't mean you stop planning. It means you shift how you plan. You move from live check-ins to async threads. You replace calendar fatigue with focused, flexible rituals that match your team's needs and their current phase of growth.

Rituals that don't evolve become rules. And rules, as we've seen, break people.

This is especially true in globally distributed teams. At Elastic, our workforce spans time zones and continents. We had to learn early that rituals built for co-located teams don't translate. Sync meetings at 10 a.m. in California are 7 p.m. in Berlin, and 2:30 a.m. in Sydney. So we changed the pattern.

We moved toward asynchronous rhythms:

- Key decisions live in documents, not meetings.
- Slack threads replace sidebars.
- Recorded meetings include transcripts and tagged callouts.

- Big calls are followed by quiet time. Space for global teammates to reflect, respond, and contribute.

We didn't stop collaborating. We just restructured how and when collaboration happens. The ritual of cross-team communication stayed. Its shape changed. And it made space for more voices, not fewer.

We are not the only team that publicly evolved its rituals as it grew. Linear has documented how they shifted from rigid sprints to six-week cycles, lightweight async-first standups, and tighter scope-cutting rituals. The cadence changed. The mission did not. Their team guide reads like a working document of how rituals should evolve as the company evolves around them.

The newest version of this is happening in 2026 with AI. Standups now run with AI-generated summaries instead of live recaps. Retros surface patterns from team data nobody had the time to read on their own. Mock reviews include an AI co-reviewer. The ritual is still the ritual — show up, learn, adapt — but the form is shifting fast. That is not a problem. It's a sign that rails can take new shapes when the train evolves. We will come back to what AI changes, and what it doesn't, in the chapter on AI.

What matters isn't how you do it. It's that you keep showing up with purpose. You hit a wall, you adjust, you don't abandon the work — you change the pattern. That's the lesson: rituals serve the mission, and when they stop serving, reshape them.

Don't confuse rigidity for discipline.

Don't let the form matter more than the function.

(And please, don't treat your favorite product management book like doctrine. Those frameworks weren't written for your team. Your mission was.)

Rituals aren't constraints. They're commitments. But only if we let them evolve. The strongest systems, the longest-lasting teams, the most resilient lifters — all share this in common: they don't cling to routine for routine's sake. They adapt. With intention. With feedback. With mission in mind.

■ CHAPTER 04

04

**FEEDBACK IS A
SUPERPOWER**

Trust is built one cue at a time. One adjustment. One shared win.

CUES, NOT CRITIQUES

A distinction worth naming first. *Criticism* and *critique* are not the same word.

Criticism announces what's wrong. Critique tries to make the work better. Criticism is judgment. Critique is investment. Criticism comes from someone who wants to be heard. Critique comes from someone who wants you to succeed.

The same words can deliver either one. What changes is the care underneath.

This chapter is about a third thing entirely. Even the best critique can be too much in the moment. Under the bar, mid-swing, mid-presentation, you don't need analysis. You need something tighter. You need a cue.

My mentor Nate Fick used to talk about the power of laser focus. Not just attention. Precision. That phrase gets thrown around a lot, but he meant it in the truest sense.

Laser focus isn't about doing one thing. It's about doing the right thing. The move that creates the most impact. The shift that makes everything else click into place.

I saw that principle in action during my early days at Endgame.

When I joined, we had four different products, all in different markets, serving different users. Built with a small team. Sold with an even smaller one. The problem wasn't that the ideas were bad. They weren't. In fact, each had real potential.

But potential doesn't scale without focus.

Nate made a tough call. He chose to end-of-life three of the products, even though they were generating short-term revenue, and put every ounce of our talent behind a single one.

It was a risk. But it paid off.

We built a best-in-class endpoint protection product in one of the most competitive markets in cybersecurity. Not because we chased everything. But because we aligned everything to a singular mission.

That's what laser focus looks like in the real world. And feedback should work the same way.

When you're under pressure — under the bar, under deadline — you don't need a list of everything that's off. You need one cue. One phrase that slices through the noise and sticks when it counts.

POINT THE FLASHLIGHT

Like in baseball.

Coaches used to flood hitters with advice mid-swing. *"Keep your hands back."* *"Level the bat."* *"Don't bail out."* All technically true. None actually helpful in the moment.

Then came this:

Picture the knob of the bat as a flashlight. In the "flashlight position," you're aiming that light beam right at the catcher's face. This cue helps you set the correct bat angle, slightly upward, with the knob directed toward the target. It's called the flashlight position because you're holding and pointing the bat the same way you'd shine a flashlight in someone's eyes.

"Flashlight!"

It's weird. It's vivid. And it works. The batter stops overthinking and starts moving with intention. The hips rotate. The swing tightens. Power follows.

That's the magic of laser-focused feedback. Not more, just clearer.

TWO WORDS THAT CHANGED THE LIFT

The same thing happens under the bar.

Stefi Cohen is one of the most accomplished powerlifters on the planet. But even the best get stuck. During a heavy squat cycle, she was losing depth and collapsing forward, the kind of breakdown that could invite a checklist of mechanical critiques.

Her coach didn't do that. He gave her one cue:

"Knees out."

Two words.

Her glutes fired. Her hips opened. Her position locked in. No overcorrection. No overthinking. Just a precise adjustment that reconnected the movement. That's what real feedback does. It doesn't flood you with everything that's wrong.

It's a spotlight, not a floodlight. Aimed at what matters most, right now.

FROM BACKLOG TO BREAKTHROUGH

Similarly, in product development, we're inundated with feedback. Users, developers, sales teams, executives — everyone has ideas, and most of them are shared with genuine excitement. Everyone wants the product to be better.

But without focus, we risk becoming a jack of all trades and a master of none. Chasing every suggestion means delivering none of them well.

The best product teams don't just collect feedback. They hone it. They find the thread, the theme, the one insight that can generate the most impact with the least investment.

Because here's the truth: in software, we can build almost anything, given infinite time and budget. But for some reason, executive teams rarely offer those.

So it takes real leadership to sift through the noise, spot what matters most, and act.

Take an example from Dropbox.

Dropbox had a problem most product teams would recognize. Users were signing up but not sticking around. Specifically, they were dropping off before installing the desktop client, the core part of the experience that made Dropbox more than just another web app.

The team started spinning on solutions. Maybe the onboarding flow needed to be redesigned. Maybe they needed more education, more nudges, more screens. The backlog of ideas kept growing.

Then a PM said something simple:

"You're treating the CTA like a destination, not a trigger."

That one line cut through the noise.

They changed the button copy from "Next" to "Install Dropbox." They added a subtle animation to guide the user forward. That was it.

And conversion went up.

No massive rebuild. No big campaign. Just a focused shift, one that made it crystal clear what the user needed to do next.

It wasn't about the volume of feedback. It was about knowing which piece would actually move the needle.

That's product leadership: listening broadly, then acting precisely.

THE FLASHLIGHT, NOT THE FLOODLIGHT

The same principle applies across every domain. Lifting, sport, software.

You don't have infinite time to build a product. You don't have infinite weights to lift and sculpt the perfect physique. You can't act on every idea, every error, every impulse.

So you focus.

Whether you're adjusting your squat, refining your swing, or unblocking a user journey, the best feedback doesn't overwhelm. It illuminates.

*Not a floodlight that blinds.
Not a laser that burns.
Just a flashlight, steady and clear, aimed at what matters most right now.*

That's how progress happens. Not with noise. With precision.

One cue, one rep at a time.

MAKE FEEDBACK A RITUAL

Feedback isn't something you give once a quarter. It's not a Slack comment or a surprise at review time. It's a rhythm. A rep.

If cues are the spark, rituals are the circuit. They keep the current flowing. The strongest teams and the strongest lifters don't just react to feedback when it shows up. They build it in. They expect it. They practice it. They normalize it.

In lifting, you log your sets. You reflect on what moved well, what felt off, what needs attention tomorrow. Feedback isn't an event. It's a ritual. Same goes for product.

And if you're a product manager, here's the simple truth:

If you're not talking to users, you are failing at your job.

Not falling behind. Not missing a step. Failing.

User feedback isn't a bonus. It's the core ritual of the role. If you're not making regular space to listen — directly, personally — you're flying blind.

You're not a scrum master. You're not a backlog babysitter. You are the voice of the user.

The user, and their mission, is why you have a job.

The research, the roadmaps, the revenue — all of it is an outcome of solving the user's problem. Not the other way around. Too many teams fall into the trap of building for elegance, scale, or conversion, and forget to ask the most important question: *Does this move the mission forward?*

That's your job as a PM: to hold that line. Metrics can drift, roadmaps can get noisy, but the mission stays clear if you keep listening.

SLACK: FEEDBACK AS THE PRODUCT

Slack didn't start out as a product idea. It started as an internal tool. The team at Tiny Speck built it for themselves while working on a multiplayer game called Glitch.

The game didn't succeed. But the internal chat tool did, because the team had built it by listening to their own pain.

They weren't running customer interviews or user panels. They were simply using the tool every day, and treating their own experience as data. The engineers, designers, writers, and founders were all giving and receiving feedback constantly. What was slow? What was confusing? What needed to work better?

The users were them. The feedback was constant. And the ritual was built into the way they worked.

They didn't treat those observations as complaints. They treated them as cues. Small signals that, over time, shaped something far more useful than they had originally intended.

So when Glitch shut down, the pivot to Slack wasn't a Hail Mary. It was a progression. The team had already spent years listening to what they needed most, and building it.

Ritualized feedback didn't just improve the product. It revealed the product they should have been building all along.

Slack didn't become Slack by guessing what users wanted. It became Slack by listening to internal signals — consistently, honestly, and early.

ELASTIC: CUSTOMER ZERO AND DESIGN PARTNER RITUALS

We've seen the same thing at Elastic.

Our internal information security team — "*customer zero*" — is often the first to use upcoming Elastic Security features. They're not a test group. They're real users with real problems. When something's unclear, slow, or missing, they tell us, early and often. And because they sit just a few channels away, we can respond quickly and refine before anything reaches broader release.

It's not just internal, either. Some of our most pivotal feedback comes from design partners — external users who've volunteered to be on the front lines with us. They opt into early access, knowing the feature might be raw, knowing it might break, because they care about shaping something real. Something that works not just in a demo, but in a live environment.

These aren't one-offs. They're not spontaneous bursts of goodwill. They're planned rituals — structured partnerships built into how we deliver great products. Embedded loops that make the product stronger long before it ever reaches general availability.

Ritualized feedback isn't about checking a box. It's about building alongside the people who share the mission.

RELATIONSHIP-LEVEL LISTENING

Collecting feedback is the easy part. Getting someone to tell you what's on their mind is just the start. What matters is what you do next.

Do you listen? Or do you nod, say "*totally,*" and go back to your plan?

It's the same in product. And in training.

You can run all the standups, retros, interviews, and training logs you want. But if you're not listening with intent, you're just gathering noise. You're not growing.

Feedback is only useful if it lands.

And that means learning how to listen — not just politely, but purposefully.

LISTENING IS A LIFT

You've built the ritual. You're getting the feedback. Now comes the hard part: listening.

Not passively. Not performatively. But actually listening, in a way that helps you grow.

And here's the first thing to understand:

Being user-obsessed doesn't mean the user is always right.

Sometimes users give you feedback in the form of a solution.

"Add a new button." "Change this flow." "Make it more like what Company X does."

If you chase those suggestions blindly, you're not listening. You're outsourcing product design.

The real job of a product manager is to listen deeper. To get past the request and find the root. Don't treat the feedback as the answer. Treat it as a clue. Your job is to uncover the real problem.

One thing I'm adamant about with my PMs: we don't hand our engineers a solution. We bring them a problem worth solving.

That means doing the work. Talking to users. Studying how others have handled it. Mapping the friction. Listening for what's unsaid.

We work with brilliant, creative engineers. Our job isn't to steal the *"how."* It's to own the *"why."* And collaborate on the *"how"* together.

Because feedback isn't always clear. It's not always clean. It can be noisy, misdirected, emotional, subtle, or even wrong.

And when you're under pressure — in the middle of a hard training cycle or a high-stakes product push — it's easy to hear what you want to hear. Or to hear nothing at all.

Listening is a skill. One that separates people who just collect feedback from those who actually use it.

It's like lifting: you might record every set and log every number, but if you don't know what the soreness is telling you, if you don't know when to push and when to pause, you're not getting better. You're just getting tired.

Sometimes, the most accurate feedback in a training session isn't in the weight or the timer. It's in how the set felt.

How many reps did you leave in the tank? What was your rate of perceived exertion?

RIR and RPE are qualitative cues. And they only work if the lifter is honest.

Just like product feedback only works if the team is actually listening.

WHEN THE DATA LIES, OR WE LIE TO OURSELVES

Take Fitbit.

When they rolled out sleep scores, users were thrilled. Finally, a single number to tell you how well you slept. But the feedback loop backfired. People started optimizing for the score — going to bed earlier, waking up later — and still feeling exhausted.

Why?

Because the metric didn't tell the whole story. It rewarded quantity, not quality. It missed the impact of stress, caffeine, or real rest. And instead of helping people understand their bodies, it created sleep anxiety. People chasing a number instead of listening to what their bodies were actually telling them.

That's the risk with metrics-as-truth: you end up listening to the wrong signal, just because it's the one that's easy to measure.

And the same thing happens at scale.

Take Target Canada.

When Target expanded into Canada, the surface signals looked great: strong brand awareness, excited customers, big opening-week foot traffic. But underneath, stores were suffering. Inventory systems were broken. Shelves

were empty. Pricing was inconsistent. And operational feedback from store teams was ignored or deprioritized.

Leadership focused on the shiny signals — the PR, the launch buzz, the optimism — and missed the messy truths on the ground. They listened to the brand metrics and headlines, not the store managers and logistics teams waving red flags.

Within two years, the entire operation collapsed, costing billions and ending in a full retreat.

When you only listen to what makes you feel good, you stop seeing what's real. And in product or business, that kind of selective hearing is deadly.

You don't need more metrics. You need better listening. Listening that filters noise. That finds the root. That says: *"I hear you. Let's fix the real thing."*

STRONG FEEDBACK, STRONG FOUNDATIONS

We should strive for critique, not criticism. Feedback that's constructive, not combative.

I've learned this over the course of my career. As a passionate product leader, I've had to walk a line: how do you bring intensity without intimidation? How do you make sure your excitement doesn't land as insult?

It's not always easy. Passion can blur the edges. And in fast-moving teams, feedback can come fast and hot, even when it's meant to help.

But here's what I've come to believe:

When feedback is anchored in empathy and structured around the mission, it becomes a lift, not a weight. It pushes people forward instead of pinning them down.

These lessons apply well beyond product, and well beyond management. They show up in how we coach our teammates, how we parent, how we partner, how we grow.

If something matters enough to say out loud, then it matters enough to deliver with care.

Because what you say is only part of it. How it lands is what makes it real.

STRONG FEEDBACK IS OFTEN MICRO

We tend to think feedback has to be a big moment. A review. A reset. A speech.

But most of the feedback that actually changes people, and changes outcomes, is micro.

One cue, one adjustment, one rep at a time.

That's how coaching works on the platform. That's how great PMs guide a team. That's how trust is built, and how performance gets better.

In the gym, the best feedback is short and specific:

"Brace before the pull." "You're losing tension at the bottom."

You don't wait until after the meet to say it. You say it in the warm-up room, in the moment, when it counts.

In product, it's the same.

"This feature's drifting from the core user need. Let's tighten it." "That's a great insight. Let's surface that earlier in the deck."

Micro feedback avoids defensiveness. It's less likely to be taken personally. It's just the next rep, done better.

SHARE EARLY, IMPROVE ALWAYS

Too many teams wait too long.

They polish. They perfect. They fear showing something unfinished.

But by the time the feedback comes, the team is too attached, or too far gone, to adapt.

The best teams share early.

That's the Medici principle: seek feedback while ideas are still flexible, not fixed. Let people shape the work before it's locked in.

You can see this principle in action at Figma.

From the start, Figma built real-time collaboration into their product, not as a feature, but as a philosophy. Designers could share mockups midstream, collect comments in context, and adjust in the flow. No long feedback loops. No over-polished reveals. Just open collaboration, one comment at a time.

At Elastic, we embraced that same mindset, first in design, then across product teams.

Figma became a shared space for early feedback: PMs sketch rough flows, designers wire up concepts, and teams review together before a line of code is written. Mock reviews aren't a gate. They're a ritual.

And like everything else, we've evolved it.

In 2026, AI is reshaping the early-share ritual entirely. We use tools like Lovable to wireframe ideas in minutes instead of weeks. PMs can upload an epic and get a working prototype back before the design review. AI editorial models read drafts and ask the questions a senior PM would have asked. The feedback loop has gone from days to seconds. That's real, and it's mostly good — but only when the human at the center is doing the harder work AI can't do: actually listening to the user, actually carrying the mission, actually hearing what was unsaid in the room. We will go deeper on this in the chapter on AI.

And feedback doesn't always need to come through conversation.

Just look at Duolingo, one of the most successful learning platforms in the world, and proudly Pittsburgh-built.

They've designed feedback directly into the product experience. From streak celebrations to in-app corrections, every interaction is a tiny nudge:

"You're close." "Try again." "You're on track."

There's no giant review. No sit-down critique. Just micro feedback, in real time, helping users improve without ever slowing them down.

And feedback rituals themselves are getting AI-augmented. Tools like Dovetail now synthesize customer interviews automatically — pattern-matching across hundreds of conversations a PM would never have the time to read straight through. The right teams use it the way Elastic uses customer zero: not as a replacement for actually being in the room, but as a way to see what was said across all the rooms at once. AI as editor of the feedback. The PM is still the author of the response.

The earlier you share, the better the outcome. The smaller the feedback, the bigger the impact, if it comes at the right time.

STRONG FEEDBACK BUILDS STRONG PEOPLE

Strong feedback doesn't come from shouting. It comes from someone who's done the work. Someone who's built themselves up, not just to be heard, but to be helpful.

The best coaches don't bark orders. They observe, they care, and they offer what helps, when it helps.

In product, the same rule applies.

You don't build trust by waiting until things break.

You build it one cue at a time. One adjustment. One shared win.

Feedback that fuels, not frustrates. That builds, not breaks.

Even in our personal lives, the principle holds. A daily affirmation. A kind word. A check-in after a hard day.

These are all acts of feedback. They tell someone how they're doing, and that you're still with them.

That's how we get stronger. As partners. As teammates. As leaders.

Not through silence. Not through softness.

When it's anchored in empathy and aimed at the mission, feedback stops being friction, and becomes fuel.

■ CHAPTER 05

05

**PROGRESS ISN'T
PRETTY**

MASTERY IS REPETITION WITH FEEDBACK

Practice doesn't make perfect. Perfect practice makes improvements.

That's the principle that separates a hundred wasted hours from a hundred that actually move you forward. The reps you log have to be deliberate. They have to be honest. They have to teach you something the next time you show up.

You see it clearly in cybersecurity.

Penetration tests are our version of reps under pressure. Simulated real-world attacks run across teams, infrastructure, and products. We don't go easy. We don't telegraph the punch. Sometimes, we don't even tell the blue team it's happening. Why? Because defenders don't get warning in the real world. If we want to build muscle memory, we need to simulate the actual weight.

That's what *perfect practice* really means. Not pristine, flawless action, but high-quality, high-context repetition.

Malcolm Gladwell's *10,000 Hour Rule*, popularized in *Outliers*, offers a compelling headline: do something for 10,000 hours and you'll become world-class. What's often forgotten, and what Gladwell himself has acknowledged, is that the number only matters if those hours are deliberate, structured, and improving. Not just any reps. The right reps.

It's a mindset that transcends domains. In lifting. In product. In leadership.

THE LIFTING LENS: PERIODIZATION AND PROGRESSION

Walk into any well-run gym and you'll see programs. 5/3/1. Texas Method. Juggernaut. Conjugate. Each offers structure, progression, and intentional overload, built on periodization: you cycle through intensities, movements, and recovery phases. It's not sexy. But it works.

Chad Wesley Smith, founder of Juggernaut Training Systems, built JTS by emphasizing smart lifting over heavy lifting. As he puts it:

"Motivation gets you started. Discipline keeps you going. Feedback tells you where to go next." — Chad Wesley Smith, founder of Juggernaut Training Systems

I use the JuggernautAI app myself. It takes in feedback on how I move, how I recover, how I progress, and adjusts the plan accordingly. Every session has a purpose. A coach in your pocket, and a reminder that great outcomes come from smart, consistent reps.

And here's the truth lifters learn the hard way: showing up and doing something beats waiting for the perfect plan. A light session with focused movement and intention will outperform a skipped workout every time. Small improvements, made consistently, create compounding strength.

THE PRODUCT MIRROR: DISCIPLINED ITERATION AT ELASTIC

When Elastic Security shipped its initial security analytics and SIEM product in 2019, it wasn't a complete answer to the market. It was a start, informed by users, opinionated in structure, and designed to evolve. What made it successful wasn't just the launch. It was the loop: weekly user feedback, signal from real-world deployments, and disciplined iteration sprints that followed a clear roadmap.

Each product cycle wasn't a shot in the dark. It was a planned lift. Part of a broader program of growth. Features like detection rules, case management, and timeline didn't just "appear." They were reps, refined under load. We looked at how users engaged, what didn't click, what needed retraining. We built telemetry into the stack, not to track vanity, but to see where improvement was possible.

The discipline scales beyond security. Watch how Anthropic has rolled out Claude from 2023 through 2026 — a deliberate cadence of model versions, capability updates documented in public changelogs, and a clear deprecation path that respects the developers building on the platform. Claude 3 to 3.5 to 4 to 4.5 to Opus 4.6 was not a series of sudden leaps. It was a planned program of reps, in public, refined under load. The market saw the releases. The team felt the loop.

REPS WITH A PURPOSE

Whether you're logging lifts or logging bugs, the idea is the same: feedback closes the loop. Reps without it just reinforce habits, good or bad. Reps with it? That's where you find progress.

Athletes track tempo, range, and fatigue. Product teams track usage, drop-off, and latency. Neither discipline rewards brute force for long. But both reward those who show up, take notes, and return better.

In the gym, a lifter might swap from high-bar squats to pause squats to address sticking points. In product, a team might switch to dogfooding an unfinished feature to catch friction early. In both, the rep is the mechanism. The lesson comes from how you interpret it.

THE GRIND THAT GROWS

The reps that change you aren't the glamorous ones. They're not the PR attempts. Not the launch keynotes. Not the LinkedIn flexes. The reps that build real strength — in the gym and in the product — are the ones nobody sees. The early morning warmups. The late-night bug fixes. The process of showing up, adjusting, refining, repeating.

This is the grind. And it's where the gains are made.

IN THE GYM: WARMUPS, NOT MAX OUTS

You can spot the novices in any weight room: they go straight for the heaviest lift they can manage. They max out on deadlifts, post it on Instagram, then disappear for weeks.

The ones who stick? They warm up. They move methodically. They hit moderate numbers with perfect form. They log their progress. And they show up again tomorrow.

Warming up isn't a sign of weakness. It's the mark of a seasoned lifter. It's protection. Preparation. And part of the process. As Chad Wesley Smith and countless coaches remind us, progress is built on consistency, not bravado.

IN PRODUCT: QUIET WINS THAT MATTER

Product teams can fall into the same trap. We celebrate the big splashy features, but the real loyalty comes from the quiet wins. The "minor" UX updates that shave 10 clicks off a daily workflow. The improved loading time. The default that finally makes sense.

We saw this first-hand at Endgame. Yes, we differentiated on high-end protections like in-memory attack prevention. But the reason our champions stayed engaged wasn't just the cool stuff. It was the continuous improvement in how they used the product. With each release, we delivered smoother experiences and more intuitive workflows. It added up.

The same holds true at Elastic Security. We spent over a year deeply focused on the alert triage experience. We met with analysts. We compiled hundreds of hours of feedback. We made sure everything they needed — context,

actionability, next steps — was right at their fingertips. That wasn't a flashy launch. It was grinding. And it paid off.

Not every enhancement gets a press release. But the users notice, and often celebrate, the small, thoughtful changes more than the big ones.

WHAT THE GRIND TEACHES

The grind teaches humility. Discipline. Attention to detail. It teaches you to care about the user who files enhancement tickets, not just the buyer who signs the deal. It teaches you to find progress in the seemingly minor.

In product, it's tackling the long list of enhancement requests. In lifting, it's refining your bar path or ankle mobility. Both require you to step away from the spotlight, and into the discipline of reps.

THE MYTH OF MOTIVATION

The myth is that motivation carries you. The truth? Motivation gets you started. But it won't carry you through the hard sets, the long roadmap, the friction.

What will? Mission. Ritual. Feedback. Community.

The best product managers and lifters don't chase the end state. They chase the process. They fall in love with it. And that's where mastery begins.

LOOK BACK TO SEE THE GAINS

Progress is sneaky. When you're in it every day, it's hard to see.

In lifting, that's why we take progress pictures. Not for vanity, but for truth. I've had times where I felt stuck, only to look at side-by-side photos and realize I was growing all along.

Same in product.

One of the best presentations I gave at Endgame was a demo showing the same workflow over multiple years. Each version looked small on its own. But side-by-side? It was night and day. Better design. Better defaults. Better detection. That's the power of consistent improvement.

CHANGE THE PHASE, NOT THE PROCESS

Anyone who's followed a good lifting program knows the feeling: what used to work, stops working. You stall. Plateau. Feel stuck.

The worst response? Panic and start over.

The best? Adjust the phase. Keep the process.

In powerlifting, this is where periodization becomes personal. You might deload. Shift focus to speed or hypertrophy. Swap bars or change stance. But you don't throw away your structure. You modify it with purpose.

The same is true in product development. You don't abandon the mission when progress slows. You evolve your strategy. You change the shape of the work.

PRODUCT EXAMPLE: SCALING FOR WHAT'S NEXT

Early in Endgame's life, we took an almost nine-month pause on new features to rearchitect the platform for long-term scale and performance. It was a scary bet, and there were plenty of sleepless nights wondering if it was the right call.

But when we brought the new build to a major deployment, and saw the blue team "win" their live training exercise for the first time, we knew it had paid off. Their expertise made it happen. But our new platform and agent helped enable it.

Sometimes, changing the phase means sacrificing momentum now to build leverage for later. That takes guts. But it's part of the process.

LIFTERS ADJUST, TOO

Smart lifters know when to change gears. That might mean trying a new program, like shifting from 5/3/1 to Juggernaut. Or letting an AI-based plan like JuggernautAI retune based on fatigue and bar speed. Sometimes it means getting a personal coach, or joining a lifter forum to troubleshoot sticking points.

What matters is sticking to the discipline. You're still logging reps. Still progressing with intention. The ritual stays, the phase shifts.

That's the new wrinkle in 2026. AI doesn't just track your reps. It adjusts the plan based on how the bar moved last week. The program responds to you. Which means the only thing that breaks the loop is whether you show up honest. We will go deeper on that trade in the chapter on AI.

SET A GOAL TO STAY MOVING

During COVID, lifting alone in my garage, motivation came from an unexpected place: signing up for a powerlifting meet. I wasn't planning to set records. But having a date on the calendar gave me focus.

That meet, and the ones that followed, taught me something else: competition isn't about beating others. It's about beating the weight. And the community is there to help you do just that. Even your "opponents" cheer you on.

The same lesson applies in product.

When you're feeling stuck, talk to your users. Talk to your peers. They'll remind you why you're doing it. And they'll help you find the next rep.

Even when you change the phase — the load, the format, the plan — your mission stays. Your ritual stays. The growth continues.

GROWTH THAT HOLDS

There's a difference between strength that shows up... and strength that holds up.

That difference comes from experience. From failure. From learning the hard way.

We saw it earlier in the story of Hafthor Björnsson, the world's strongest man who tore his pec pushing beyond safe limits. He later called it his "biggest mistake" and rebuilt with a smarter plan. The same pattern shows up again and again in high-level lifting. Even Stefi Cohen, one of the most decorated lifters of her generation, has shifted focus toward long-term health, adjusting how she trains, eats, and recovers, not out of fear, but out of wisdom.

These lessons apply everywhere high performance matters.

PRODUCT EXAMPLE: BUILD FAST, LEARN FASTER

At Endgame, we learned a critical lesson well before the world-shaking CrowdStrike outage of 2024. We had shipped a powerful capability: the ability to update our machine learning-based malware prevention engine out of cycle from the main product release. It meant our users got faster protection. They didn't have to wait for a full product upgrade.

It was the right feature. But it had an unintended consequence.

During Black Friday at one of our retail customers, an update disrupted point-of-sale systems. Nothing catastrophic. But enough to make an impact.

We fixed it fast. We added guardrails. We changed our rollout process. And we never made the same mistake again. More importantly, we kept the feature, and improved how it worked for the user, at their scale, in their implementation.

We learned that the best product capabilities don't just "work." They hold up.

They account for real-world usage, edge cases, and the 80/20 rule, that 80% of issues will come from 20% of scenarios. And we built resilience into the system because of it.

THE PRODUCT TAKEAWAY

Your best ideas may cause friction if they're not paired with empathy.

It's not enough to be right. You need to be reliable.

And the only way to get there is through reps that test the system, expose the edges, and give you time to learn and evolve. This is how mature products, and mature lifters, endure.

■ CHAPTER 06

06

**THE MISSION
DEMANDS
RECOVERY**

THE BIOLOGY OF BOUNCE BACK

Recovery isn't retreat. It's reinforcement. Whether you're lifting heavy or leading strong, progress isn't just about what you do. It's about what you recover from. And yet, in both gyms and boardrooms, recovery is often misunderstood as weakness, wasted time, or something you'll "get to later." When stress hits or deadlines loom, recovery is usually the first thing cut.

That mindset breaks things.

It's an often-repeated phrase, *"it's a marathon, not a sprint,"* but even marathons have aid stations. Recovery isn't the opposite of performance. It's what makes performance possible.

You'll notice a pattern: the strongest systems, bodies, and teams are built on rhythms, not sprints. And the ones that fall apart? They usually forgot to rest.

THE YEARS RECOVERY VANISHED

During the pandemic, the commute disappeared. And with it, the last buffer between work and rest. Working from home seemed like a win, until days began with a Slack notification and ended with Zoom still open. Lunch breaks were replaced by back-to-back meetings. Evenings bled into late-night emails. There was no off switch, just a lower screen brightness.

People didn't stop working. They stopped recovering.

It wasn't the difficulty of the tasks. It was the absence of boundaries. Parents toggled between spreadsheets and homework help without leaving their chairs. Managers tried to hold their teams together through webcams and chat threads. Even rest became another thing to optimize, a new podcast, a sleep tracker, a mindfulness app squeezed in before another call.

Burnout crept in sideways. Not as a dramatic collapse, but as a slow erosion of clarity, patience, and presence. Work got done. But purpose got blurry. People didn't crash. They faded. Not because they cared too little, but because they were never given space to come up for air.

Recovery wasn't just missing. It had been overwritten.

MAT FRASER AND THE DISCIPLINE OF RECOVERY

That same lesson echoed far beyond the home office.

Mat Fraser, five-time CrossFit Games champion, wasn't always the strongest on paper. But he outlasted everyone. His secret was obsessive recovery — sleep tracked with Whoop, daily heart rate variability, nutrition managed to the ounce, mobility, ice baths, light movement on off days. Nothing left to chance. He saw rest as the difference between surviving and dominating. In his own words:

"I do everything right, especially the boring stuff."

That "boring stuff" is what let him stay healthy, adapt faster, and build a career that eclipsed stronger but less disciplined competitors.

BASECAMP'S BUILT-IN COOLDOWN

The same mindset showed up in software development. Basecamp built recovery into their operating rhythm. After every six-week product cycle, where teams focused on delivering something real, they took two weeks off from planned work. These cooldowns weren't vacations, but a time to reset, fix bugs, explore ideas, or recover team energy. There were no deadlines, no pressure to deliver. This wasn't laziness; it was strategy. Founders Jason Fried and David Heinemeier Hansson believed sustainable work required ebb and flow. Without that rhythm, quality would slip, and burnout would rise. Instead of asking their team to be superhuman, they asked them to be consistent and gave them the space to do it.

G2 ESPORTS AND THE POWER OF TOURNAMENT RECOVERY

The power of recovery isn't just for traditional sports or software teams. In the high-stakes world of esports, teams that build recovery into their cycle gain a lasting edge.

G2 Esports, one of the most successful organizations in Valorant, is a prime example. After a disappointing showing in 2023, the team deliberately adjusted its approach, incorporating more deliberate cooldown time between matches and reducing the number of high-pressure scrims. The goal wasn't to train less. It was to train smarter. To recover. To breathe.

The results spoke for themselves. G2 won the VCT Americas Stage 1 title in 2024 and reached the Grand Final at Masters Bangkok, pushing T1 to a five-map series. They didn't take the trophy. But their ability to perform tournament after tournament was proof of a system built for endurance.

THE CHALLENGER DISASTER AND THE TRAGIC COST OF SKIPPING THE PAUSE

And when recovery is missing, the cost isn't just performance. It could mean people's lives.

As we wrote in *Rituals Over Rules*: "...our goal isn't to meet an investor's timeline or an executive's forecast. It's to build the best product to solve our user's mission." That perspective doesn't just apply to shipping code. It applies to every system that carries real risk.

In 1986, engineers at Morton Thiokol warned that cold weather could compromise the space shuttle Challenger's O-ring seals. The data was there. The risk was known. But leadership, under pressure to meet a schedule, pushed forward anyway. The launch went ahead, and seventy-three seconds later, the Challenger exploded, killing all seven crew members. The physical cause was the O-ring failure. But the systemic cause was the inability to pause. There was no cultural space for dissent, no moment to reflect. The organizational systems were sprinting — politically, reputationally, and economically — with no built-in mechanism for recovery or review. The result wasn't just a tragedy. It was a preventable failure brought on by the absence of pause.

EVERNOTE AND THE DRIFT WITHOUT REFLECTION

That same cost plays out in the digital world. And here is where the Drift comes back.

We named it in Chapter 1: the Drift is the slow inversion of mission into metric. It does not announce itself. It rarely arrives with bad intent. And it does not stop at launch dates. There is a recovery-shaped Drift, too. It looks like an unreflective sprint that nobody calls a sprint anymore.

Evernote was once the darling of productivity tools, with a loyal user base and a clear mission. But as it scaled, it kept adding: more features, more integrations, more complexity. There was no time to reflect, consolidate, or revisit its core value. Internal teams were stretched thin. Technical debt piled up. Bugs increased. The product lost its identity under the weight of

unchecked ambition. And as the team chased innovation without rest, users quietly left for simpler, clearer alternatives. Evernote didn't fail overnight. It drifted, feature by feature, update by update, away from what had made it valuable in the first place. All because recovery was never part of the plan.

This is the Drift again. Same shape we named in Chapter 1, but recovery-shaped this time. Boeing's Drift killed people. Hafthor's Drift tore his pec. Evernote's Drift cost the company its identity. The mechanism is the same. The mission gets buried under the metric. The pause gets skipped. And by the time you notice, the thing you were building has become something else.

PROTECT THE PAUSE

Recovery doesn't happen by accident. It has to be built in, programmed into cycles, protected by boundaries. Strength isn't built in the moment you push. It's built in the space between pushes, if you protect it.

Recovery is strategy. Not the opposite of work — it's how the work keeps getting done. Growth requires rest. And not just any rest. Intentional, protected, repeated. When you plan for recovery, you're not falling behind. You're investing in your future resilience.

Because the mission doesn't just demand effort. It demands endurance.

RAILS, NOT BARRIERS

Boundaries aren't barriers. They're rails. Not to box us in, but to keep us from going off track.

In lifting, the rails are obvious: rest days, deload weeks, technique sessions. You can grind all you want, but if you don't recover inside the structure, the system breaks. Not always with a snap. Sometimes just with a slow slide into fatigue you can't name.

In product, the rails are harder to see. And harder to hold.

This work is rarely clean. It shifts constantly between strategy and triage: vision decks one moment, sales escalations the next. Context-switching becomes the default. What you planned often gets pushed aside by what the business demands now.

Instead of pretending we can block out perfect "no meeting days," the better move is to create rails that expect the chaos but still protect time to think clearly.

Call them strategy blocks. Step-back time. Mental deloads. The name doesn't matter. The intent does: to build in space that lifts your head above the noise and lets you make intentional decisions, not reactive ones.

Because rails aren't rigidity. They're protection. For focus. For energy. For the mission.

NETFLIX AND THE SPACE TO REFLECT

Even in a high-performance culture like Netflix, boundaries around strategic thinking are built into the system. Product leaders are expected to regularly pause and reflect, not just on what they're building, but on who's building it. Their well-known "Keeper Test" asks a simple but revealing question: *Would I rehire this person for this role today?* Answering that honestly takes courage. But it also takes time. Time to zoom out. Time to get quiet enough to think clearly. In a culture that thrives on speed, those reflections are what keep the team aligned, accountable, and evolving.

Step-back time doesn't have to be silent. It just has to be sacred.

SUSTAINABLE STRENGTH

Recovery is not a reward. It's part of the plan.

In strength training, the most disciplined athletes don't just train hard. They track what fuels the work. Calories. Sleep. Macros. Because performance isn't just about what you do in the gym. It's about what you support outside of it. That's why tools like MacroFactor have become essential for serious lifters: not for vanity, but for visibility. To make sure every training block is fed with what it needs to grow.

In product, the same principle applies. You can't build a resilient team without knowing what you're feeding it. That's where tools like Jira, Aha!, or Linear come in, not just as ticket trackers, but as visibility engines. They let you zoom out. Spot overtraining. Identify where strategic work is undernourished. Recovery in product isn't just about taking breaks. It's about sustaining the energy to build well over time.

Because burnout doesn't usually come from sprinting. It comes from sprinting without a cycle.

That's why elite lifters use periodization — planning not just the next workout, but the next month, quarter, and year. They alternate intensities. Deload. Peak. Rebuild. Not because they're lazy. Because they're playing the long game.

Sustainable product teams follow the same rhythm. They don't live in permanent crunch. They push when necessary, then deliberately create time to reflect, retool, or recharge. They respect the difference between urgency and importance. And they recognize that more activity doesn't always mean more progress.

When you track your macros, you don't just know what you're eating. You know what you're aiming for.

When you track your product macros — the balance of innovation, maintenance, user research, and downtime — you build not just velocity, but viability.

One example of this kind of tracking in action comes from Atlassian's Team Health Monitors. These are not daily scrums or sprint reviews. They are recurring, focused check-ins designed to assess the long-term health of a team. Each one reviews key areas: clarity of roles, alignment on goals, confidence in decisions, pace of delivery, and engagement with stakeholders. Teams score themselves on a red-yellow-green scale to visualize patterns over time. The point isn't to be perfect. It's to stay aware. To notice stress points early, to invest in what's working, and to surface risks before they derail momentum. It's macro tracking for the makers. Less about this week's burndown, more about whether the system is still healthy enough to support what's next.

Because the strongest teams aren't the ones who go the hardest. They're the ones who stay strong the longest.

WHEN YOUR TEAMMATE DOESN'T SLEEP

There's a new variable in 2026 worth naming. AI doesn't sleep. It doesn't deload. It doesn't take Q4 off to recover for Q1. The pace of an AI-augmented team is now set, in part, by a teammate that has no biology and no boundaries. The risk isn't that AI burns out. It's that the humans pairing with AI start absorbing AI's pace as their own. The pace of the AI is not the pace of the mission. We will come back to this in the chapter on AI.

RECOVERY IS A LEADERSHIP SKILL

If you want to go the distance, as an athlete, a builder, or a leader, you can't just train intensity. You have to train sustainability.

Recovery isn't about stepping back. It's about stepping up, with clearer focus, steadier energy, and decisions made from purpose instead of panic.

In the gym, you build this with sleep, nutrition, deloads, and reflection.

In product, you build it with boundaries, cooldowns, strategy days, and macro awareness.

You build it when you protect the pause. When you track more than just tickets. When you care not just about the sprint, but about who's still standing after it.

Because the mission won't slow down. But you can build systems, and teams, that last.

RECOVERY CREATES ROOM FOR PERSPECTIVE

Halfway through the book, and at a chapter all about recovery, it's worth pausing on something that too often gets misunderstood or misused in product work: diversity as a design principle.

The acronym DEI has been politicized and weaponized. The principle behind it is older than the acronym, and more useful than the politics. Because the enemy of clarity isn't chaos. It's assumption. And when your product is only shaped by people who already know how to use it, you stop seeing what's broken.

Think about the average enterprise tool. Bloated. Confusing. Clunky. Not because the teams behind them don't care, but because they've learned to work around the flaws. They stop noticing friction because they've internalized the shortcuts. And then they add more complexity without questioning the base layer. That's not innovation. That's calcification.

Remember Apple's infamous iPhone 4 moment? When reception dropped if the device was held a certain way, the official response was: *"You're holding it wrong."* That wasn't just a PR misstep. It was a failure of empathy. A product built with a single use case in mind, and a team unwilling to question it.

This is where diverse inputs become a design discipline. Not a slogan. Not a checkbox. An actual practice. When you bring in voices that don't match your defaults — different backgrounds, different abilities, different ways of thinking — you create friction that reveals blind spots. You build products that more people can actually use.

The Medici Group teaches a powerful example of this in their innovation workshops: hospitals learning from Formula 1 pit crews. Two entirely different fields — surgery and motorsport — sharing best practices to reduce patient handoff time in operating rooms. The result? Faster triage. Fewer errors. Lives saved. That's what happens when you step outside your bubble and let diversity expand your playbook.

And sometimes, that kind of inclusion is quieter, but just as critical.

RESPECT THE PAUSES

Diversity comes in many forms, including language. At Elastic, where teams span the globe, English is the default for most business communication, but it's not the first language for many Elasticians. That matters more than most people realize.

In a leadership training, one of our engineering leaders walked the room through what happens cognitively when you're a non-native English speaker in a fast-moving meeting. You listen. You process. You translate. You form a thought. You translate again to respond. That's not delay. That's effort. And if you interrupt the pause where that work happens, you erase the contribution before it can surface.

Respecting the pause allows full participation. It gives space not just for language translation, but for different processing styles, communication rhythms, neurodiversity, and accessibility needs.

Internally, we called it the curb-cut effect. Curb ramps were designed for wheelchairs. But now they help everyone — people with strollers, luggage, bikes, injuries. What was built for access became a better default for all.

Respecting the pause works the same way. It's a small behavior that opens the door to a much wider room.

■ CHAPTER 07

07

**TRAIN THE ENGINE,
NOT JUST THE
OUTPUT**

WORK CAPACITY OVER WINDOW DRESSING

There's a difference between looking strong and being strong. You see it in the gym all the time: the lifter who chases their one-rep max every week, treating PRs like progress reports. Impressive, until the plateaus hit or the pain sets in. What you don't see as often is the lifter dragging a sled after squats. Doing walking lunges in the cold. Rowing in the morning and hitting their accessories at night. That's the engine work. That's capacity.

"If you want to go far, build the lungs before the legs."

In the training world, this is called GPP, General Physical Preparedness. It's not just a warmup or conditioning. It's the foundation for everything else.

GPP improves recovery between sets and sessions. It builds joint integrity, tissue resilience, and work tolerance. It makes your system ready, ready to train harder, recover faster, and adapt more consistently. And yet, in many strength circles, it's dismissed. You'll hear things like:

- *"Combo sets are my cardio."*
- *"I don't have time for fluff. Just the main lifts."*
- *"If it doesn't add pounds to the bar, it's a distraction."*

That's the specialization trap. And it's seductive.

Specialized programs promise fast results. Just do the thing you want to improve, more of it. Until it stops working — until your joints hurt, your energy fades, your momentum disappears, and you burn out.

Repetition without variation isn't just physically risky — it's psychologically draining. Without variation, even the most disciplined lifter starts dreading the next workout. Even the most motivated developer begins to disengage.

We've seen this in teams that only work on performance. Or only ship new features. The work becomes mechanical. Morale suffers. Curiosity fades. They're strong in one motion, but brittle under real-world load.

That's why smart teams, and smart lifters, build in variation. Like the 6/2 cycle popularized by Basecamp: six weeks of deep, focused work followed by two weeks of cooldown. Capacity training, not a break. We explored this in Chapter 6, and other companies have since adopted and evolved it because it works.

GPP isn't glamorous. But it's what lets you train hard, and show up strong, again tomorrow.

That's what Alex Viada understood. A biomedical engineer and hybrid athlete, Viada rejected the false binary between strength and endurance. He trained for both, and thrived. While most lifters feared that conditioning would eat their gains, Viada built an aerobic base robust enough to support heavy squats and sub-5:00 miles. His results weren't just impressive. They were repeatable. Durable.

You see the same principle in product teams. Linear didn't try to dazzle with a full feature suite out of the gate. Instead, they trained their engine: performance. Every click, every transition, every load state was built to feel instant, even offline. That invisible investment became their advantage. Users stayed focused, fast, and frictionless. And while competitors stacked features, Linear scaled loyalty.

But the obsession with output can take an even darker turn, in the gym and in the market.

Some lifters now chase not even performance, but the appearance of performance. Injections like synthol or site-enhancing oils artificially inflate muscle volume — biceps, shoulders, traps — to create the illusion of strength. No increased output. No increase in health. Just ballooned tissue under the skin. These are vanity muscles, pure and simple. And in too many cases, they don't just harm the lifter's reputation. They destroy their body. Heart failure. Sepsis. Amputation. Even death.

When looking strong matters more than being strong, the mission is already lost.

And in product? It's not so different.

Teams inject vanity metrics into their roadmaps: flashy dashboards no one uses. AI features for the pitch deck. Slideware that outpaces software. It might look good in the moment, but behind the scenes, the engine's breaking down. We will come back to the AI version of this problem in the chapter on AI — when the model can generate any output you ask it for, the question of *what's actually under the hood* becomes the only one that matters.

You can't fake work capacity. You can fake a feature. You can fake a max lift. But you can't fake the foundation.

Burnout is what happens when you try to specialize your way out of exhaustion. Capacity is what happens when you train for the long haul.

The ones who last don't just train the output. They train the system behind the output.

INTERNAL STRENGTH FOR EXTERNAL LOAD

Strength isn't just about what you can lift. It's about what you can withstand.

Anyone can show up strong once. But the mission demands more than that. It demands repeatability. Resilience. The ability to show up strong again and again, under pressure, under uncertainty, under load.

That kind of strength doesn't come from testing yourself every day. It comes from building yourself for the test, from the inside out.

We know this intuitively in lifting. If you neglect your stabilizers, your joints, your breath control, your bracing, it doesn't matter what your numbers look like. You'll break. The biggest total in the gym means nothing if your back gives out the next week. True strength isn't just output. It's the integrity of the system supporting it.

The best lifters don't chase the max. They train the margin.

That's what Amanda Lawrence understands. A multi-time world champion powerlifter, Amanda is famous not just for her records — squatting over 600 pounds raw at 84kg — but for her poise under pressure. She visualizes every meet lift weeks in advance, rehearses the rituals, manages recovery like a professional sport. She doesn't just train her muscles. She trains her nervous system, her mental process, her ability to stay composed under load.

That internal strength — not just quad size or grip — is what lets her hit massive lifts on the world stage, even when the bar bends and the cameras are rolling.

THEY TRAINED THE TENDONS OF THE PRODUCT, NOT JUST THE MUSCLE

The same applies in product.

Notion wasn't always a darling. In 2015, they were nearly dead. No traction. Technical debt piling up. The team had two choices: patch the pain and push forward, or stop, go dark, and rebuild the core from scratch.

They chose the hard route. Two engineers rewrote the entire editor engine over 18 months. No releases. No hype. Just deep technical investment in a stronger, more flexible system. They bet that the internal strength of the product, not surface-level features, would determine their long-term viability.

They were right.

That rebuilt core became the foundation for everything that followed: blocks, embeds, team permissions, scalable collaboration. The invisible architecture made possible everything the user eventually saw. What saved Notion wasn't a sprint. It was a rebuild.

Throughout my career, I've seen how scale and stability work is often dismissed as unexciting, not flashy enough for slides, not "pressable" enough for headlines. Some leaders fall into the trap of always needing something new to talk about. Something shippable. Something that demos well.

But that pressure leads to bloated products, incoherent strategies, and worst of all, users who suffer when the product finally gets real-world load.

When everything matters most, the engine breaks.

Here's the truth: capability refinement is not the boring work. It's the hard work. It's where the most difficult engineering problems live. It's where the most intricate, high-stakes user requirements get applied. Not in the pitch deck, but in the heat of usage.

Building a reliable core doesn't mean every startup needs to design for a Fortune 10 on day one. But it does mean they need to carve out time in the roadmap to move deliberately toward that direction. They need to ask the hard questions early. Not just *"What are we building?"* but *"Who are we building it for?"* And just as critically: *"Who are we not building it for?"*

You cannot build for every user in every vertical in every geo all at once. Trying to do so leads to brittle systems and vague products that serve no one well.

As we said in earlier chapters: laser focus is strength. Define your product requirements. Define your user. Define the scale. Define your edge cases and your tests. Then do the work, especially the work that doesn't make the press release.

Because that's where real durability comes from. Not the big reveal. The boring reps.

A STORY I'VE LIVED

As I described in the prologue, in July 2016 my engineering lead and I stood on the floor of the Combined Air Operations Center at Nellis, watching our agent run through a live Air Force exercise. The story behind that moment starts here.

Early at Endgame, after CEO Nate Fick made the difficult but necessary decision to narrow our product portfolio to one core offering, engineering leader Tony Meehan and I found ourselves at a crossroads.

The product we had was powerful. It delivered real results and we had major enterprise deals on the horizon. But the architecture wasn't built for scale. The team had shown clearly that it wouldn't hold as we moved into larger opportunities.

We could have coasted. Closed deals. Deferred the hard part.

But we knew that if we built a base of users on that foundation, the eventual transition would be expensive and painful.

So in early 2016, we made the call to rebuild. We moved the front end to a more modern language, redesigned the back end with a new queuing architecture, and rewrote how the endpoint agent communicated with the platform. All of this had to be done in under six months, alongside new features, for a critical delivery with the U.S. Department of Defense.

It was brutal. Long hours. Hard conversations. Constant doubt.

But participating in that live DoD exercise, watching operators succeed with what we had built, it was clear. The effort had been worth it.

The new architecture held. It scaled. It supported the mission until we joined Elastic in 2019.

Internal strength isn't flashy. But it's what gets you through the weight that would break someone else.

WHAT POWERS YOU?

The fastest engines in the world still go nowhere without fuel.

You can have world-class systems, perfect technique, and an unbreakable schedule. But without something driving you — a reason to push through when it's hard, a mission that makes the effort matter — the engine stalls. Strength without fuel is just potential. It goes nowhere.

This is the third layer of performance. After physical preparation and mental resilience comes purpose. The why behind the work. The force that powers the engine when the load gets heavy and the results are still far away.

You can see this difference on the platform. Some lifters are chasing numbers. Others are lifting for something deeper — to test themselves, to prove something to a younger version of who they were, or to inspire someone watching from the crowd. The same weight moves differently depending on what's driving it.

The same is true in product work. Teams that ship the fastest aren't just well-organized. They're aligned around a purpose they care about. They understand the mission. They know what their work means to the people who use it. And that makes them more resilient, not just more productive.

When the fuel runs low, purpose drives on.

Take Kristy Hawkins. Before she was a record-breaking powerlifter, she was a biochemical engineer at Genentech. Whether in a lab or under a barbell, she trains with the same energy source: a relentless drive to understand, improve, and own her progress. She's powered by process, not comparison. That's what makes her strength sustainable.

The same goes for teams. Look at how Superhuman built their product. Founder Rahul Vohra didn't just want to make a faster inbox. He wanted to build the fastest, most delightful email experience ever created. They didn't launch until users said they'd be "very disappointed" if they lost the product. That wasn't marketing polish. It was purpose, measured. Every decision —

from UI transitions to onboarding flow — was shaped by that internal fire: delight. Clarity. Joy. The product knew what it was, and the team knew why it mattered.

Capacity is mechanical. Mental strength is adaptive. But purpose is emotional. It's fuel.

That's why it matters what you're building strength for. You can build a massive work capacity and still burn out if you're not connected to what it's in service of. Even strong people lose their way when they forget why they're moving in the first place.

The question isn't just: *Can you go further?* It's: *Why would you?*

Ask any seasoned lifter what keeps them going, and the answers get quieter over time. It's not the meet. Not the total. It's that the ritual grounds them. It makes them better. It reminds them of who they are.

Ask a product team what keeps them shipping when everything breaks, and if they're still smiling, it's because the mission means something to them. They've felt the user's pain. They've seen the real-world impact. They've tasted the outcome of building something that matters.

So, what powers you?

Not just when things are working. Not just when people are watching. But when the bar is stuck. When the sprint is a slog. When the road is long and there's no applause in sight.

That's the kind of test SERE training is designed for. In the U.S. military's Survival, Evasion, Resistance, and Escape course, individuals are pushed to their psychological limits — deprived of sleep, isolated, interrogated. It's not just a test of skill. It's a test of resolve. Of remembering the mission when the environment is engineered to make you forget.

The ones who endure don't have better gear or more talent. They have clarity. They know what they're fighting for.

That's your fuel. That's your engine.

That's the difference between showing up when it's easy, and showing up when it counts.

■ CHAPTER 08

08

**DECISIONS ARE
MADE UNDER LOAD**

STRESS TESTS THE SYSTEM

Stress doesn't break the system. It exposes what was already weak.

We often talk about "testing under load" as a strength-building concept. In the gym, it's simple. Your ability to move weight when you're tired is the real signal of progress. But in product leadership, systems design, and incident response, stress testing is often overlooked until it's too late.

The Log4j vulnerability in December 2021 made that brutally clear. A zero-day exploit in a widely used Java library allowed remote code execution with minimal effort. Suddenly, nearly every organization had the same urgent questions:

- Are we exposed?
- Where is Log4j deployed in our environment?
- Do we have visibility into historical access or exploitation attempts?

This wasn't just a code issue. It was a test of organizational readiness — technical, operational, and strategic. Teams that had practiced incident response, mapped their architecture, and built cross-functional trust could move with purpose. Others stumbled, not because they didn't care, but because they'd never trained for the moment.

It was the backup problem, writ large. Many companies back up data religiously, but few test whether they can actually recover it. Even fewer know how long recovery will take, or what their RTO (Recovery Time Objective) or RPO (Recovery Point Objective) really are. The lesson was hard-earned: backups are only as valuable as your ability to restore fast.

Log4j forced a parallel reckoning. Malicious scanning began months before the public disclosure, but most organizations didn't retain accessible data that far back. They had cold storage — frozen or tiered off for cost savings — but had never practiced bringing it back online quickly for investigation.

When the pressure hit, teams had to figure out, often in real time, how to thaw data, rehydrate logs, and search for infection markers across old records.

Some had tested this process before. They knew the commands, the timing, the quirks of their tooling. They'd trained like operators. Others hadn't. Some couldn't search more than 30 days back. Others found that recovering historical logs was so cumbersome it may as well not have existed. Those companies weren't just missing data. They were missing a disaster recovery plan for threat analytics.

At Elastic, customers who had previously rehearsed these data flows — archiving, restoring, and hunting — were able to quickly triage and verify impact, drawing from months-old telemetry. They weren't just ingesting logs. They were building institutional memory.

And this mirrors what elite military units do when they train decision-making under fatigue. In Navy SEAL training or Army Special Forces SERE programs, candidates are made to perform operational tasks after sleep deprivation, cold water exposure, and physical exhaustion.

Not to break them, but to reveal what's already broken.

In those moments, you don't rise to the level of the manual. You fall to the level of your defaults. And the only way to improve those defaults is to train them, deliberately, under controlled stress.

We have lived the same shape since. In February 2024, the Change Healthcare ransomware attack took down claims processing, prescription routing, and billing across most of U.S. healthcare. The vulnerability wasn't novel. The organizational readiness was. Hospitals that had drilled their incident response could keep dispensing prescriptions on paper. Others lost weeks to manual workarounds because no one had practiced operating without the platform. Same lesson, different load. Stress finds the truth.

THE 2026 STRESS TEST IS MACHINE-SPEED

The Log4j-shaped stress test was bad. The 2026 version is worse, because the attackers are no longer human-speed.

The mean time from vulnerability disclosure to confirmed exploitation has fallen from 2.3 years in 2019 to under one day in 2026. Breakout times — the window between initial compromise and lateral movement — are measured in seconds. Phishing campaigns crafted by large language models hit click-through rates 4.5 times higher than human-written ones. Capabilities that once required nation-state expertise are now available to anyone with an API key.

If you are not using AI to battle AI, you will lose. Defensive AI is the new minimum entry fee just to stay level. We will go deeper on this in the chapter on AI.

STRESS FINDS THE TRUTH

It's no different in product and platform work. You might have a dashboard for monitoring, a workflow for restores, a playbook for incident response. But unless you've tested them when it matters, they're just ideas. Stress finds the truth.

So whether you're squatting on trembling legs after five working sets or trying to rehydrate petabytes of logs to confirm an intrusion:

The system only works if it works under pressure.

This isn't a new idea. We introduced it in Chapter 2 with James Clear:

"You do not rise to the level of your goals. You fall to the level of your systems."

Under the weight of stress and urgency, that line deepens.

Goals vanish in crisis. Roadmaps collapse under pressure. What remains is the infrastructure you've trained, tested, and trusted.

Whether that's a disaster recovery pipeline, a muscle memory forged in fatigue, or a culture that knows how to move without waiting for permission, it's your system that carries you.

Resilience isn't reactive — it's built. Deliberately. Daily. And under load.

CLARITY BEATS CERTAINTY

You don't need to know the outcome to know how to act.

There's a myth in leadership, and in lifting, that success comes from always having the right answer. That if you just gather enough data, wait long enough, or plan hard enough, certainty will arrive and carry you through.

But the best teams, the best lifters, the best products — they don't run on certainty. They run on clarity.

You see it clearly in moments of disruption. In early 2020, when the COVID-19 pandemic rewrote every assumption about business, teams that waited for complete information froze. But the ones who moved quickly didn't do it recklessly. They did it with intention.

- **Zoom** opened up free access to educators worldwide. Daily meeting participants skyrocketed from 10 million in December 2019 to 300 million by April 2020.
- **Peloton** overhauled its entire supply chain when demand outpaced capacity. By Q4 2020 it reported 232% year-over-year revenue growth and added over 1 million new subscribers.
- **Notion** refocused messaging for the remote work moment. Its user base grew from 1 million to over 4 million in 2020.

None of them knew how long the shift would last. But they had clarity of mission. That clarity became a compass.

This is the same clarity that high-level lifters use when training with RPE (Rate of Perceived Exertion). Instead of sticking to a fixed number on the bar, they assess how each set feels in real time. On some days, that means pulling back to preserve recovery. On others, it means pushing heavier than planned.

It's not guesswork. It's informed adaptation. It's clarity in action, even when conditions change.

This is where the OODA Loop earns its place. We met it in Chapter 1: Observe, Orient, Decide, Act. The point isn't perfection — it's momentum. You act, assess, and adjust faster than the environment can overwhelm you.

This model doesn't just apply to military operations. It applies to product releases, user feedback, platform incidents, and even organizational change. You're not waiting for every signal to align. You're moving with awareness, grounded in your principles, and fast enough to respond before inertia sets in.

And this is where clarity beats certainty.

- Certainty waits.
- Clarity prepares.
- Certainty asks, "*Are we sure?*"
- Clarity says, "*Here's what we do next.*"

You don't always need a perfect map if you've trained your compass.

Let's be honest: Agile is a loaded word.

Some teams swear by it. Others roll their eyes. And more often than not, the problem isn't with the philosophy. It's with the implementation.

Too many teams have had Agile forced on them by someone who read a textbook but never wrote a line of code. The rituals become rigid. The flexibility becomes performative. And the spirit gets lost in the ceremony.

I'm a firm believer in the Agile mindset, but not the dogmatic, by-the-book version. True agility means responding quickly to new information. It means iteration with intent. But that only works if the team has clarity. Without it, Agile devolves into a flurry of half-finished tickets, context switches, and directionless pivots.

The point was never to abandon focus. The point was to stay mission-aligned, laser-focused on the "*why*," while remaining flexible on the "*how*."

Where I most often see Agile fail is when it becomes a cover for indecision. When teams confuse agility with ambiguity. When the lack of a clear goal gets labeled as "*keeping our options open*."

I also see it fail when people demand certainty before committing to action. Even today, with more data than ever before, certainty is rare. We have telemetry, user interviews, funnel analysis, Salesforce records, analyst feedback. But the decision on what to build next is still more art than science.

If there was a playbook to follow, we'd be project managers. But we're product managers.

Our craft isn't in perfect foresight. It's in translating user pain into progress. It's in surfacing the "why" beneath the "what." It's in seeing the pattern behind the request.

And as Nathaniel Fick wrote in *One Bullet Away*, his memoir about the transformation from Ivy League student to Marine officer leading combat missions in Afghanistan and Iraq:

"Complex ideas must be made simple, or they'll remain ideas and never be put into action."

That's our job. To take everything we know — the signals, the inputs, the intuition — and turn it into action. Not after a perfect analysis. Not when every detail is known. But when it's time to move.

We do that not by being certain, but by being clear.

Clear on the mission. Clear on the user. Clear on what matters most right now.

Because clarity moves. Certainty stalls.

And in the fog of delivery, feedback, and shifting priorities, the job is not to predict perfectly. It's to navigate intentionally.

TRAINING FOR CHAOS

We don't rise to the level of our plans. We fall to the level of our preparation.

In the last chapter, we explored the idea of training the engine, building general physical and mental capacity, not just specific skills. But that alone isn't enough. What happens when the plan vanishes? When you're under pressure, short on time, and forced to act without hesitation?

That's when preparation has to become instinct.

I learned this firsthand as a U.S. Army paratrooper. Before you ever board a plane, you train on the ground for weeks. You drill every movement until it's embedded. How to hook up your static line, how to check your gear, how to exit, how to land. You jump from towers. You simulate the aircraft. And you practice failure, not because you want it, but because it's coming.

You want to fail when you're four feet off the ground, not 800.

We don't take our first flight until the basics are second nature. Because stepping out the side door of a C-130 at 800 feet, in pitch black, isn't natural. It's against every self-preservation instinct. And in that moment, you don't want to be thinking through a checklist. You want to be moving.

If your main chute doesn't deploy, you have 9 to 10 seconds to respond before impact. That's it. There's no time to think. You just act.

I still remember my first jump. The roar of the engines. The rush of the open door. And when we stood up, hooked up, and waited for the green light, I didn't feel fear. I didn't feel stress.

I felt clarity.

I knew what to do. And I executed. Because we had trained for it.

That same mindset exists in the best teams, even in tech.

At Netflix, Chaos Monkey was built to deliberately shut down random services in production, to ensure that both systems and people could adapt under pressure. They didn't just hope their systems were resilient. They trained them to be. And the results were measurable:

Netflix prevented 80% of potential outages through the learnings surfaced by Chaos Monkey and the broader chaos engineering discipline. These weren't hypothetical flaws. They were real vulnerabilities, caught early because someone had the courage to test failure on purpose.

This wasn't an act of sabotage. It was an act of preparation. You don't find out if you're ready in the middle of a real incident. You find out beforehand, if you're willing to simulate the storm.

You see this in sport, too. During the CrossFit Open, athletes receive surprise workouts announced just days before competition. Movement patterns shift. Equipment combinations vary. Sometimes it's a short, explosive sprint. Other times, it's a long, grinding engine test. You don't know what's coming, and that's the point.

This unpredictability isn't accidental. It reflects one of CrossFit's founding principles: prepare for the unknown and the unknowable. Because in real life, just like in combat, product development, or incident response, you don't get a playbook with advance notice. You're asked to perform under pressure, with incomplete information and limited control.

By designing chaos into the format — new standards, awkward movement pairings, strange pacing demands — CrossFit forces athletes to confront not just their physical readiness, but their emotional and mental adaptability. The top performers aren't just strong. They don't just move well.

They stay composed when the plan disappears.

And that's the test. That's the transferability. The Open doesn't just reward fitness. It rewards clarity under fatigue, pacing under uncertainty, and strategy when the rules shift.

The chaos isn't a bug. It's the feature. It reveals what structure alone can't.

This belongs in the same category as paratrooper training or chaos engineering. You don't need bullets or breakage for chaos to be real. All you need is volatility, pressure, and a demand for action.

And when that moment comes, the best don't panic. They breathe. They move. They decide.

I want to pause here and share something personal.

I've never claimed to have done the hardest thing in the military. Many others have faced more direct danger, made life-or-death decisions in the field, and carried far heavier loads. Nate Fick, whose story we referenced earlier, is one of many. But I've also learned that you can't live your life comparing hardships. Chaos shows up in many forms. And every one of us has moments when our training is tested by stress.

For me, that came during my deployments in support of Operation Enduring Freedom and Operation Iraqi Freedom, where I served as an intelligence analyst at the Combined Air Operations Center. My job was to represent U.S. Army interests in daily joint targeting and mission planning, to advocate for the air support our soldiers needed downrange.

That meant walking into rooms where I was often the lowest-ranking person present, trying to secure limited airframes for missions that couldn't afford to fail. There were more missions than aircraft. Everyone in the room wanted to help, but everyone had missions to support. The real pressure wasn't the data. It was the conversation. The negotiation. The weight of knowing that if I couldn't make our case clearly, my brothers in arms might not get the support they needed.

That stress taught me the true value of clarity. Not just in numbers, but in stories. It wasn't the PowerPoint or the packet that made the difference. It was being able to explain why this mattered, what the risk really was, and who would be impacted if we didn't act. I learned young, and under pressure, how to speak across rank and service. How to translate intelligence into insight. And most importantly, how to give a shit, loudly, clearly, and with purpose.

I didn't have a name for the motto then. As I shared in the prologue, the words live on the patch of VMM-364, the Purple Foxes. I would learn the squadron's history later in my career. But under the load of those rooms in the CAOC, the principle was already running the work.

One of the awards I received, the Army Commendation Medal, noted that I served in multiple roles normally held by ranks far above mine, including as the BCD Intelligence NCOIC (a Sergeant First Class role) and as an Intelligence Plans Officer, a position typically held by a Major. I don't say that to boast. I say it to underscore this truth:

You don't perform under stress unless you care deeply about the mission and the people it impacts.

That lesson stayed with me. In uniform. In product. In leadership.

It's not process that drives you when the pressure hits. It's principle. It's preparation. And it's the ability to act with conviction, because you know exactly what's at stake.

You don't have to be on the front lines to know what chaos feels like. You just have to care enough to move when it matters most.

■ CHAPTER 09

09

**SHIP IT LIKE YOU
SHOW UP**

Empathy isn't softness. It's clarity. It's knowing what to say no to.

INTEGRITY IN SMALL THINGS

How you do the small things is how you do everything.

Greatness doesn't usually show up where we expect it. It isn't built in the spotlight. Not in PR lifts or flashy product launches. Not in press releases or headlines. Most of what truly earns trust and builds strength happens quietly. In the warmups. In the follow-through. In the care no one notices but you.

From the outside, powerlifting looks simple. Three lifts: squat, bench, deadlift. Anyone can walk into a meet and give it a shot.

But the difference between showing up and winning isn't about size or intensity. It's about discipline. The small things. The lifters who win are the ones who move with precision.

The bar comes out of the rack the same way every time. The breath is controlled. The bar path is tight and deliberate. The brace is locked in. The foot pressure is balanced. And they do it the same way whether it's 135 pounds or a max-effort lift.

That's where mastery comes from: consistency under load.

Watch Olympic lifter Lu Xiaojun and you'll see it. People celebrate his explosive strength, but what makes him exceptional is his control. Every rep is intentional. Every rereck is clean. There's no ego in his movement, only precision. He lifts like someone who knows that how you finish matters just as much as how you start.

This is what the best product teams understand too.

Anyone can ship a feature. But not everyone takes the time to check whether the error message makes sense. Not everyone notices if the cursor lands in the right input field. Not everyone asks if the experience feels smooth or rushed.

But when someone does, users feel it. Even if they don't know why. It's the difference between something that works and something that feels right.

At GitHub, engineers fix what they call papercuts: small friction points most people would ignore. They don't wait for permission. They just fix them. And when they do, the product feels better. Not flashier. Just more thoughtful.

At Apple, designers obsess over pixels and motion. Scroll behavior, bounce physics, shadow softness. These are the details most people never consciously notice. But they notice how it feels. And that's the point. The little things aren't just polish. They're the experience.

This isn't fluff. It's care, made visible.

So why doesn't the most full-featured product always win?

Android phones offer more toggles, more options, longer spec sheets. Yet Apple, with fewer features and tighter control, continues to lead in loyalty and user satisfaction.

Think back to Blackberry, when the iPhone first launched. *"No one wants to type on glass,"* they said. They were wrong. The first iPhone didn't have copy and paste, an App Store, or video recording. On paper, it looked incomplete. In practice, it felt revolutionary. Apple's teams spent months tuning details most users would never name — scroll speed, button animation timing, the physics of how things moved on screen. Not for the demo. For the experience.

Other companies added features. Apple focused on how it felt to use.

BMW takes a similar approach. They don't sell the highest horsepower. They sell the experience of driving — the way the wheel feels, the way the chassis responds. It isn't just what it can do. It's how it makes you feel while doing it.

That's the difference.

Great products don't win because they have the longest list of features. They win because they've been shaped by people who care about every detail. People who understand and obsess about the mission.

That mindset shows up outside of product and sport too.

John Cena, before the fame, was a regular at Gold's Gym. He trained hard, but that's not what stuck with people. What they remember is how he re-racked every plate. Wiped down every bench. Left the space better than he found it. No cameras. No applause. Just quiet respect for the work and for the people who would follow.

You don't earn trust with a launch. You earn it with the work no one claps for.

Because when the moment comes, when the lift is on the platform or the product is in the wild, the real question isn't, "*Did it work?*" The question is, "*Does this reflect who we are when no one's watching?*"

THE SHIPPING STANDARD

A clean lift on the platform tells you more than any gym session ever could.

You can move big numbers in training, hit personal records, and feel confident under your own rhythm. But step onto the platform at a meet, and everything changes. The commands are faster. The lights are brighter. The room is louder. The pressure is real. And even with the same weight on the bar, your form starts to slip.

Not because you lack strength, but because you didn't train for that moment.

Building is like training. Shipping is competing.

And competing demands more than power. It requires control, presence, and precision when the world is watching. It also introduces more variables than training ever does. In the gym, you control the pace, the setup, and the rhythm. But on the platform, you might face a cold bar, unfamiliar flooring, a tight timeline, or commands that arrive quicker than expected.

Shipping is no different.

In development, teams test the happy path. In production, users rarely follow it. They skip steps, input bad data, use features in unexpected ways, and still expect everything to work. These edge cases aren't theoretical — they are reality. That's the difference between building in private and shipping in public. That's what it means to compete.

At Netflix, shipping isn't something saved for the end. It's part of every step. Engineers own their code from the first line to long after it's live. They test it, monitor it, tune it, and fix it when needed. There's no handoff. No deflection. Accountability is shared.

Staging environments mirror production closely, simulating real user behavior and traffic patterns. Netflix teams know you can't expect consistent performance unless you train in the conditions you'll actually face. They don't celebrate heroics after a failure. They celebrate readiness before one. That's how they maintain quality under pressure.

Shipping is a team sport, and the teams that win align around a shared standard of care.

You see that same commitment at NASA.

Before every launch, they hold a Flight Readiness Review. Each subsystem lead, whether in engineering, mission ops, safety, or comms, walks through their area of responsibility and gives a go or no-go decision. If even one person says no-go, the mission halts. No debates. No pressure to push through. Just a clear respect for the standard.

When the risks are that high, trust isn't built on optimism. It's built on preparation, discipline, and the shared courage to pause until it's right.

THE DAY THE STANDARD HELD

That's the standard we held ourselves to before our biggest deployment.

In July 2016, our agent — the product we had spent six months rewriting from the inside out — went live for Red Flag at Nellis Air Force Base. As I described in the prologue, my engineering lead and I stood silent on the floor of the Combined Air Operations Center as the exercise started. Pilots in the air. Controllers guiding them. Air defense tracking them. A crashed laptop in a startup is annoying. A crashed laptop in that room is a different category of problem.

There's a saying in these exercises: *it's more fun to be a pirate than to be in the navy*. The red team — the adversaries — always have the upper hand because they only have to be right once. The defenders have to be right twenty-four hours a day, seven days a week. In 2016, the blue team and our agent changed that.

The exercise ran through. The teams reported no issues. The system held. And then the red team came back and asked us to turn down our protections so they could continue to train. That is the greatest honor of any defender in these exercises. The pirates had been forced to ask permission. One of our operators earned an award from the exercise commander for performance under load.

We didn't get applause for the rebuild that made it possible. We got something better. We got the trust to be there next time.

*That is the shipping standard. Not what you ship.
Who you are when it goes live.*

EVEN WHEN THE LIFT DOESN'T LAND

Even with all that care, sometimes the lift still doesn't land.

In powerlifting, you might feel like you hit a perfect rep, but the judge flashes red. Maybe your depth was just short. Maybe the lockout wasn't fully controlled. Maybe you rushed the pause. The judge doesn't grade your effort. They grade your execution.

You don't get to argue. You adjust.

The best lifters don't spiral when they miss. They listen, learn, and step back on the platform with more precision than before. You miss. You learn. You lift again.

Shipping is no different.

You might release something you believe in, only to hear that it confused users or didn't work as expected. Maybe the problem it solved internally doesn't translate externally. That's not failure. That's feedback. And the only real failure is refusing to respond.

The best teams are not perfect. They are resilient. They listen to signals and respond with care. They don't just ship once. They ship again and again, with more awareness every time.

We've all seen what happens when something goes out just to meet a deadline. Corners are cut. Quality drops. Bugs slip through. The follow-up work takes longer than doing it right would have in the first place. And users notice.

That isn't shipping. That's recovery.

A real shipping standard doesn't mean moving slowly. It means moving with intention. The checklist exists before the pressure hits. Dry runs happen before the launch. And everyone knows what "ready" really means. Not just functional, but finished.

Your standard is your signature.

If you wouldn't sign your name to it, it's not ready.

Lifting the weight is only part of the story. Anyone can move it once. But the lifter who shows up, hits depth, follows commands, and finishes clean is the one who earns respect.

In product, it's the same.

You don't earn trust because you shipped something big. You earn trust because you shipped something you stand behind.

That's why elite teams don't rush. They prepare.

As the U.S. Marine Corps teaches:

"Slow is smooth. Smooth is fast."

It isn't about hesitation. It's about precision under pressure. You train the way you want to perform. You ship the way you want to be trusted. Speed without control leads to chaos. But when you move with purpose, readiness compounds into confidence.

That's what creates momentum you can build on, not just scramble after.

EVERYTHING BUILT IN

At the highest level, strength isn't brute force. It's total integration. Each breath, angle, and cue refined until nothing is wasted.

Lu Xiaojun isn't dominant because he trains harder. He's dominant because his training is complete. His grip, his breath, his core, and his positioning are aligned. There is no wasted motion. Every part of his movement supports the next. He's not just strong. He's unified.

Product teams face a similar challenge. It's easy to chase what's next, the shiny feature, the headline for the release notes. But when something new is added without regard for the whole, it often unsettles more than it improves. Instead of adding value, it reveals what's missing. Instead of delighting users, it creates friction in places that once felt natural.

That's why product managers need to pause. Not just to evaluate whether something can be built, but whether it belongs. Does it complete the user's experience? Does it connect meaningfully to the rest of the platform?

The goal isn't just to deliver functionality. It's to ensure everything in the system makes more sense because it's there.

Lu doesn't just train the bench press. He trains the setup, the position of his feet, his breathing under load, and his recovery. Strength alone doesn't make a champion. What matters is the integration of every part.

Product should be no different.

A new feature can introduce more problems than it solves if it isn't thought through holistically. It can expose what's missing more than it reveals what's new. PMs need to take a step back and evaluate not just the capability, but the completeness of the capability across the entire product suite. That completeness is what creates trust.

One of the clearest illustrations of this lives far from software. It lives in a hospital.

At Great Ormond Street Hospital in London, a pediatric cardiac surgery team noticed something troubling. Their operations were world-class, but the transition from surgery to intensive care, the most critical moment, was prone

to error. Equipment delays, missed steps, unclear roles. Despite skilled professionals and excellent tools, the system itself wasn't working.

To solve it, they looked beyond healthcare. They studied the Ferrari Formula 1 pit crew.

In an F1 race, a pit stop lasts seconds. Tires change, fuel flows, adjustments happen. All with zero confusion. The choreography is clear. Roles are defined. Everyone knows their job, and every motion supports the next.

The hospital team brought in Ferrari's pit crew to analyze their process. Inspired by what they saw, the team adopted similar principles. Clear roles. Repeatable sequences. Coordinated language. They didn't add more people or machines. They built better integration.

The result? Fewer errors. Faster transitions. Stronger outcomes. Not because they moved faster, but because they moved together.

That kind of completeness is rare but unmistakable.

Fred Rogers understood it deeply. His television program wasn't flashy or fast. It was thoughtful, quiet, and purposeful. From the music to the pacing, from the way he entered the room to the way he asked questions, every detail was considered. Nothing was accidental. Everything contributed to the experience of a child feeling seen, safe, and understood.

Mr. Rogers didn't ask how to entertain children. He asked what they needed. He aligned his words, his tone, and his format with the emotional weight of that need. He didn't simplify to the point of distortion. He clarified. He built trust by being consistent. And he built it in Pittsburgh.

Completeness is not a checklist. It is a standard. You feel it in a product that anticipates what you need. You see it in a lifter whose movement is so fluid it barely looks like effort. You experience it in a team that moves with confidence because nothing is missing.

But completeness doesn't begin with execution. It begins with empathy.

The best teams do not just ask what they can build. They ask what someone is trying to accomplish. They understand the pressure, the environment, and the mission that lives outside the interface. They recognize that a feature is not

just code. It is part of a system meant to help someone do something that matters.

Empathy isn't softness. It's clarity. It's knowing what to say no to. It's designing for the full context of the user's day, not just the ideal case in a wireframe. It's building for the hard path as well as the happy path.

Mr. Rogers didn't deliver entertainment. He delivered care. He didn't just talk. He listened.

That's what the best products do. They don't overwhelm. They understand. They don't compete for attention. They earn trust.

What *complete* means is changing in 2026, because AI is shifting the interface layer itself. We will go deeper on this in the chapter on AI.

EVERYTHING BUILT IN

Everything built in. Nothing in the way.

That is what makes a system strong. That is what people come back for. Not because it does everything, but because it does the right things, completely.

■ CHAPTER 10

10

**THE TEAM IS THE
TOOL**

SPOTTERS SAVE MORE THAN REPS

Spotters are often seen as the quiet figures standing near the bench or squat rack, but their role is anything but passive. A good spotter wears many hats: a safety net ready to intervene if failure would cause harm, a form checker watching for breakdowns in technique, a source of subtle cues like *"tighten up"* or *"drive through."* Yet the essence of a spotter's job is restraint. They give just enough help, if any, to let you challenge yourself safely and build real strength.

"A spotter's purpose is to help you lift, not to lift for you."

This same principle shows up in the best peer reviews. When a teammate reviews your code, design, or plan, they are not there to rewrite your work. Their goal is to provide insights you might have missed, highlight potential risks, and offer alternative perspectives. Done well, a review leaves the author better prepared to own and improve their work, not dependent on the reviewer to fix it. The best peer reviewers, like the best spotters, are vigilant but hands-off until needed.

Similarly, in product work, the feedback we gather from users should act like a spotter's cues. Engaged users tell us what is confusing, what is delightful, and what is missing, helping us see where to focus. But they do not dictate the solution. They guide our decisions with their lived experience. If we lean too hard on user feedback as a script, we lose our strategic intent. Like a lifter expecting the spotter to lift every rep, we stop growing.

Even in the boardroom, we do not go in cold. Before a major strategy session or executive presentation, we run documents by trusted colleagues and dry-run meetings to anticipate hard questions. These are forms of spotting. The spotter's goal is the safety and growth of the lifter, not their own performance — and the same is true of the colleagues who review your work.

Skipping spotters by rushing in without preparation is like dropping the weight in front of your peers. You overlook a key stakeholder's concern, fumble a critical metric, fail to align on goals.

The risks of skipping spotters are real, both in the gym and in our work. Attempting a near-max squat without a spotter or safety pins risks catastrophic injury. Likewise, shipping straight to production without a second set of eyes can let bugs, security flaws, or critical oversights slip through. Even when alone, you can set up tools like spotter arms or safety straps in lifting, and automated linters or security scans in software, to catch failures before they cause damage. These tools do not replace human judgment but add a crucial layer of resilience.

This watchful mindset extends to active and adversarial roles like red, blue, and purple team exercises in security. Here, the spotter is not quietly observing but actively challenging the system. Red teams attack assumptions and infrastructure to expose weaknesses. Blue teams defend and shore up vulnerabilities. Purple teams integrate both perspectives to accelerate learning. The spotting role becomes dynamic and strategic, applying deliberate stress so the team grows stronger, smarter, and more resilient before a real adversary arrives.

"Great spotters know when to speak up, when to stay silent, and how to encourage without undermining."

To be a better spotter — in the gym, on your team, or in the boardroom — focus on staying attentive, offering clear and timely feedback, and resisting the urge to take over. Remember, you are not there to lift the weight, give the presentation, or write the feature. You are there to assist, protect, and create the conditions for someone else to succeed. As explored earlier, a key aspect of strong product management is defining the why, not prescribing the how, empowering engineering teams to discover innovative solutions rather than trying to do the lift alone. Above all, great spotters listen deeply, because understanding what someone needs in the moment is the difference between helping them grow and accidentally holding them back. And listening, both as a skill and a form of strength, is exactly what we will explore in the next chapter.

TRUST IS A SHARED PR

In powerlifting, a personal record, or PR, is celebrated not just by the lifter but by everyone who helped them reach it. Spotters, coaches, training partners. Each person who encouraged, corrected form, or simply showed up to support shares in the success. Trust grows strongest in these moments because the lifter knows they could not have done it alone, and the team feels ownership in the accomplishment.

"A shared PR is a shared story, one that bonds people together and makes them eager to keep showing up for each other."

In product work, trust is forged the same way. When teams win together — landing a major customer, resolving a critical outage, or shipping a complex feature — the shared achievement becomes a foundation for deeper collaboration. Trust grows not because of individual heroics, but because people see each other invest, struggle, and succeed side by side.

Shared PRs in the workplace are not just milestones. They are reminders that we are stronger as a team than as individuals.

A personal example comes from working alongside my former engineering peer, Tony Meehan. We had a strict rule when communicating to the team: never use I statements. Our reasoning was simple. There was nothing we accomplished in that job that was done alone, so saying *"I did this"* or *"I fixed that"* was incorrect. Everything became *"we"* or *"the team,"* even in small wins. We also made it a habit to take blame when things went wrong and defer credit when things went right. This simple discipline created a culture of shared ownership, where trust was built through consistent signals that no one succeeded or failed alone.

This mindset helped us build a world-class endpoint at Endgame by sharing the mission, the credit, and the responsibility with everyone involved. We avoided the false pretense that leaders single-handedly make everything

happen. The truth is, we are only as strong as our team. Our real job as leaders is to clear the path for our team's success, remove obstacles, and ensure everyone can do their best work.

This was the heart of what Nate meant when he said we were an elevator asset company, as shared in the first chapter. Our mission was not to build products for our resumes but to lift others — users, teammates, and the organization — so they could rise higher. Every decision, every feature, every fix was grounded in the belief that we succeed only when we help others succeed first.

When I entered the world of competitive powerlifting, I was intimidated. I was surrounded by people stronger than me, lifting far more than me, and for much longer. I expected to be mocked or ignored because of my starting lifts. But it was the total opposite. The entire community celebrated every lift, no matter the weight, because they were there for the love of growth.

"True communities of strength do not measure value by the weight on the bar but by the willingness to show up, put in the work, and support each other along the way."

That experience solidified the fact that real strength lifts others.

The best teams lift the weaknesses of others and amplify their strengths. The best leaders know their own challenges and intentionally build teams that fill those gaps, challenge them, and make everyone better. Abraham Lincoln famously surrounded himself with people who disagreed with him because he understood that true leadership does not fear dissent. It embraces it to sharpen decisions and expand perspectives. I have made the same practice a part of how I build teams. But I have also been on the other side of it.

When I joined Elastic after the Endgame acquisition, I came from a classic endpoint enterprise model where everyone charged per endpoint deployed. That was the industry standard. Elastic leadership was adamant that endpoint security should fold into Elastic's consumption pricing, eliminating per-endpoint pricing entirely. I disagreed, based on my years in the classic space. My bias was rooted in my experience — which is exactly the kind of bias this book has been warning about.

I listened. I aligned. The shift turned out to be a massively positive move. We saw instant growth in endpoints under management, and we had users tell us, "Wait — I just get a world-class endpoint now?" as part of their existing

Elastic spend. The leaders I have learned the most from were the ones who insisted on being told they were wrong. That is the kind of decision I am proud of. The kind I was wrong about first.

Weak leaders hire yes-men and cling to their authority, afraid of being replaced or overshadowed. But greatness comes from wanting your team to be better than you, from helping them grow to surpass what you could achieve alone. That is real leadership. Creating an environment where others succeed, thrive, and carry the mission forward stronger than you ever could on your own.

I see the same lesson every time I coach in powerlifting. I help lifters who can move more weight than I ever will, and I love it. There is no jealousy in their success. Only pride. Their growth does not diminish me. It fulfills me. That is the joy of building strength in others, of seeing people achieve things they once thought impossible. This is the mindset great leaders carry into every team they build: knowing that their true legacy is not what they personally accomplish, but what they enable others to do.

Yet many organizations unintentionally erode this trust by rewarding only the person who lifts the heaviest. The loudest voice in a meeting or the contributor who works unsustainably long hours. These incorrect rewards extend to structures like executive bonuses tied solely to short-term metrics or exclusive club trips awarded only to individual sales reps. While intended to motivate, they create a culture of isolated effort, where teammates compete for personal recognition instead of collaborating for shared success. It is like a lifter chasing PRs without ever spotting others, focused only on their own numbers while the team stagnates around them.

Building a culture of shared PRs means celebrating contributions at every level. The engineer who asked a critical question in a design review. The product manager who aligned stakeholders. The customer success team that surfaced insights changing the roadmap. When we recognize that each lift, each launch, and each success is the result of collective effort, we shift the spotlight from individual ego to shared mission.

Trust grows fastest in tough moments. When mistakes happen or plans fall apart, how a team responds determines whether trust is built or broken. Blame creates distance. Honesty, accountability, and a commitment to move forward together deepen trust and keep the team resilient. Shared PRs aren't only about winning. They're about recovering and learning side by side. Leaders who

highlight team wins, acknowledge behind-the-scenes work, and remind everyone that no victory is a solo achievement build cultures where people feel seen and valued.

I do not believe in the myth of the single person shaping the world. Stories of figures like Musk or Jobs often overlook the teams of brilliant designers, engineers, and product managers who imagined, built, and refined what those leaders presented. Apple's iconic products were crafted by extraordinary groups whose work was brought to Jobs, not invented solely by him. Musk has made careers out of buying others' visions and marketing them as his own. Ego has no place in product management. Our role, as I shared often with Tony at Endgame, is to take the blame when things go wrong and defer the credit when things go right.

"No single person defines a product or a company, and those who claim otherwise have a personal reckoning they need to face."

Real strength, real leadership, and real success come from recognizing that only by lifting each other do we reach heights worth celebrating.

THE TOOL YOU INVEST IN

Every lifter knows that your tools matter. Barbells with tight tolerances, smooth bearings, and well-maintained knurling do not just make lifting more enjoyable. They make it safer, more effective, and more consistent. Investing in quality equipment is not about appearances but about performance and longevity. Neglecting your tools leads to unpredictable outcomes, unnecessary risk, and wasted effort.

The same is true in building products and teams. The most important tool you will ever invest in is your people. Yet in too many organizations, especially those led by egocentric founders or executives, the energy and focus go toward building a cult of personality instead of investing in the team that actually delivers results. These leaders chase personal brand moments — keynotes, interviews, social media spotlights — while their people struggle with lack of resources, shifting priorities, and unclear goals. Companies like these are not designed to lift everyone up. They are designed to keep one person at the top.

WeWork under Adam Neumann is a prime example. Neumann's obsession with his personal image and extravagant lifestyle overshadowed the needs of the company and its employees. His behavior bred chaos and instability, culminating in WeWork's spectacular collapse, which hurt thousands of workers and investors who had trusted in the vision of one person rather than a strong team.

I preach the complete opposite. Our role as leaders is not to cultivate an image of ourselves as saviors or visionaries. It is to clear the path so our teams can do their best work, remove obstacles, and set them up for success without making ourselves the story. A leader who needs to be the hero will always sacrifice team health for personal glory. A leader who makes the team the hero builds lasting strength.

Consider Microsoft under Satya Nadella. By shifting focus away from internal politics and combative silos to a culture of empathy, shared success, and growth mindset, Nadella turned a stagnating giant into one of the world's most innovative and collaborative companies. His investment in people became Microsoft's most powerful tool, reigniting both morale and performance.

This investment shows up in simple but powerful ways. Regular one-on-one conversations that go beyond status updates. Clear career paths that match individual goals with organizational needs. Genuine encouragement when someone steps up to face a tough challenge. Just like oiling the bearings of a barbell or checking the safety pins, these small actions add up to a culture where everyone feels they can perform at their best.

When you fail to invest in your tools, or your team, things break. People leave, quality suffers, and trust erodes. Frustration festers when problems are ignored or when people are treated like cogs rather than craftspeople. Teams become brittle, unable to adapt when stress increases. Over time, just like a neglected piece of gym equipment, they become unsafe and unreliable.

The best leaders recognize that the team itself is the ultimate tool. They make proactive investments in learning opportunities, mentorship, collaboration rituals, and mental health support. They advocate for their team's needs with the same urgency they would apply to fixing a broken production system. They know that an investment in people pays dividends in creativity, resilience, and sustained performance.

And they remember that this investment is ongoing. Just as a well-used barbell requires regular maintenance, people require consistent care. The strongest teams are not built by one-time efforts or flashy perks but by steady, thoughtful attention that helps every member feel stronger, safer, and more capable.

Empathy is at the heart of great product management. We spend time listening to users, understanding their pain, and building solutions that make their lives better. Yet too many leaders forget that empathy must start with our own teams. The same patience, curiosity, and care we extend to customers should

guide how we support those we work with every day. When leaders apply empathy both outward to users and inward to their teams, they create a culture where everyone feels seen, valued, and motivated to do their best work.

"Titles do not matter. Relying on explicit authority — your rank, your title, your power over someone's paycheck — reveals weakness, not strength."

The strongest leaders do not force people to follow them. They create an environment where people want to follow because they trust the mission, believe in the work, and feel supported in becoming their best. This is implicit authority. The quiet confidence earned by consistently empowering others. It is the difference between being a boss people fear and a leader people choose.

I saw this firsthand in the US Army. Those who demanded respect purely through their rank were often the ones who deserved it the least. True leaders like Nate Fick inspired teams who wanted to follow them into battle, not because they had to, but because they believed in the leader, the mission, and each other.

WHEN AI JOINS THE TEAM

The team is changing in 2026. AI agents are now teammates. They draft, summarize, prototype, prioritize, ship, and increasingly, decide. You manage them the way you would manage any direct report: clear mission, honest feedback, real ownership when something breaks. The team is still the tool. The tool just got bigger. We will go deeper on this in the chapter on AI.

The strongest teams are not defined by the brilliance of a single person, or the cleverness of a single model, but by how fiercely they lift each other, day after day. When we invest in people with patience and purpose, we build not just products, but missions worth believing in.

■ CHAPTER 11

11

**STRONG ENOUGH
TO LISTEN**

The hardest lift you will ever make is putting your ego down.

LISTENING AS LEADERSHIP

In lifting, the strongest athletes don't just follow cues. They seek them out. They crave feedback on form, timing, and control, knowing every correction is a chance to grow. But this hunger for insight doesn't end at the gym doors. The same mindset defines exceptional leaders: they don't listen to reply. They listen to understand.

Listening as leadership begins with presence. It's easy to nod while you plan your response, but transformative listening happens when you quiet your thoughts long enough to hear what the other person truly needs. Whether it's a training partner describing discomfort under the bar or a teammate raising a concern in a product sprint, the act of listening says: *I value you, and I value what you see.*

We've all interacted with someone who claims to want feedback but argues or shuts down when it's given. Or worse, nods along while rehearsing their rebuttal. In lifting, these athletes plateau or get hurt. In leadership, the consequences are subtler but just as damaging: a culture where honesty fades, mistakes repeat, and innovation stalls.

When people feel their leadership is unwilling to listen or routinely ignores feedback, they stop engaging. Leadership becomes something to avoid rather than trust. Interactions with managers feel like a tax on progress instead of a path to improvement, and silos begin to form as people retreat to spaces where they feel heard or simply left alone.

Feedback exists for a reason. It's an outside perspective that reveals what we can't see ourselves. Day after day, trapped in our own heads, we start to self-reinforce ideas, sometimes bad ones. Without external input, flaws harden into habits.

But feedback must also be given responsibly. Some people weaponize "feedback" as an excuse to be cruel or condescending. Great leaders know that effective feedback is clear, focused on actions, not personal attacks, and delivered with the intent to build, not break. It's the difference between saying, "You suck at this," and "You could improve by keeping your elbows tighter during the press." One demoralizes. The other empowers.

In previous chapters, we explored how product managers can get lost in tactical execution, missing the bigger picture of the workflow or user journey. The same happens under the bar. You might obsess over your sticking point in a press, never realizing your bar path is drifting forward. Stepping back, and welcoming constructive feedback, is what breaks through these blind spots.

Listening also creates psychological safety, the foundation of high-performance teams. When people know their ideas and fears will be heard without judgment, they stop guarding their words and start sharing the truth. That candor becomes the bedrock of innovation and resilience.

Leaders who listen multiply their perspective. They gain dozens of lenses on the mission. It's not passive. It's proactive. They ask clarifying questions. They invite dissent. They recognize that the best solutions rarely come from a single voice, but from many voices shaped into one direction.

And just as a lifter needs to listen to their own body, leaders must listen to themselves. Noticing fatigue, frustration, or ego creeping in is the first step to recalibrating. It's leadership turned inward. Staying aware of the signals that could compromise decisions or derail momentum.

*"Most people do not listen with the intent to understand; they listen with the intent to reply." — Stephen R. Covey, *The 7 Habits of Highly Effective People**

It's also expected that there will be days when you screw up. Not every meeting, Slack message, or email will land perfectly. We're human. Part of being a great leader is accepting this, taking responsibility for mistakes, and being transparent with your team when it happens. Saying something like, *"Yesterday I didn't give you enough time to clearly make your point. I'm sorry. Can we continue the conversation?"* shows strength, not weakness.

Just as empathy isn't a sign of softness, openly identifying your mistakes builds trust and models the behavior you want in your team. Too often, leaders cling to the illusion that they must stay on a pedestal, mistaking authority for respect. These explicit-authority leaders demand deference without earning it, forgetting that respect is given when it's first shown.

We are all human. Titles are made-up constructs.

Be a human, don't be an asshole, and you'll be amazed at how well your team can operate.

Listening as leadership is a daily discipline. It demands humility, curiosity, and the courage to act on what you hear, and the wisdom to give feedback that lifts others up, not tears them down. It's the difference between a leader who demands compliance and one who inspires commitment, and between a team that survives and one that thrives.

LISTENING BETWEEN THE LINES

Good leaders listen to words. Great leaders listen beyond them. Because what people say rarely tells the whole story. The subtext — the hesitation in a voice, the glance across a table, the silence after a loaded question — often reveals more about what's really happening than the words alone.

In lifting, an athlete might claim they're fine while shifting uncomfortably under the bar. You can hear it in their breath, see it in the wavering setup, or sense it in the way they rack the bar with relief instead of resolve. A coach who misses these cues risks pushing an athlete to injury or letting a lack of confidence fester unaddressed.

In the workplace, the same dynamics unfold. A teammate may say *"I'm good with this plan"* while their tone flattens or their posture closes off. They might contribute less in meetings, or their messages become shorter and less engaged. These are signals leaders must learn to read, because they're early warnings of misalignment, burnout, or growing resentment.

Listening between the lines isn't about mind-reading. It's about active listening. Active listening is the set of techniques that helps you tune into more than words:

- **Observe nonverbal cues.** Pay attention to body language, facial expressions, eye contact, posture, and gestures. Discomfort, stress, or disagreement often shows up physically before it's spoken.
- **Reflect and paraphrase.** Repeat back what you heard in your own words. It shows you're tracking, and it gives the other person a chance to clarify or expand. For example, *"So you're saying the timeline feels tight, but you think it's still doable?"*
- **Ask open-ended questions.** Questions like *"How do you feel about this approach?"* or *"What would you change?"* invite honest responses and help surface unspoken concerns.
- **Pause intentionally.** Silence can feel uncomfortable, but giving space after someone speaks often encourages them to share more, especially when they sense you're patient enough to truly listen.

- **Listen for tone and pace.** Speeding up, slowing down, or changes in pitch can signal anxiety, excitement, or doubt.
- **Eliminate distractions.** Close your laptop, put your phone away, and make eye contact. Active listening demands full presence, and split attention sends the message that what they're saying isn't important.

These techniques help you catch the gaps between words and intent, giving you a clearer picture of what your team actually needs.

This kind of listening isn't theoretical. When Alan Mulally became CEO of Ford in 2006, he started "Business Plan Review" meetings where executives reported with green, yellow, or red indicators. Despite Ford losing billions, everyone initially reported only green. Mulally noticed the nervous laughter, fidgeting, avoidance — clear signs people were afraid to share bad news. He called it out. The next week, an executive showed a red indicator. Mulally publicly praised the honesty, breaking the culture of fear overnight. By recognizing the disconnect between words and behavior, he changed Ford's culture, surfaced real problems, and helped the company avoid bankruptcy without a government bailout.

A similar lesson comes from the boxing ring. During the 2020 fight between Deontay Wilder and Tyson Fury, Wilder's corner threw in the towel in the seventh round, despite Wilder insisting he wanted to keep fighting. His coach, Mark Breland, saw the subtle signs: Wilder's legs were unsteady, his reflexes dulled, and his punches lost their snap. Clear indicators he was in real danger. Breland's choice was controversial, but many praised him for protecting Wilder from potentially devastating injury. His willingness to act on what he saw, not just what his fighter said, put the athlete's long-term health ahead of pride in the moment.

When you can identify misalignment early, you can address it before it grows into conflict or disengagement. When you can spot uncertainty, you can offer clarity. And when you show someone you see beyond their words, you prove you care enough to truly understand them.

AI can hear what gets said. It cannot hear what gets meant. We will go deeper on this in the chapter on AI.

THE EGO LIFT

Some leaders believe the biggest threat to success comes from the outside: competitors, market shifts, or resource constraints. But often, the greatest danger to a team or organization is internal. An unchecked ego at the top. Ego convinces leaders they alone have the answers, blinds them to new information, and makes them resistant to change. It transforms what should be shared missions into personal crusades, suffocating the creativity and collaboration needed to adapt and excel.

The ego lift shows up when leaders believe their title grants them absolute knowledge or when athletes refuse coaching because they think they know best. These moments look like confidence, but they're actually insecurity disguised as bravado. And they poison teams faster than almost anything else.

Listening doesn't work when ego stands in the way. A leader might nod along waiting for their turn to speak, or worse, interrupt to redirect the conversation back to themselves. The effect is the same: people stop sharing. Meetings become performative. Feedback withers. Silos grow. The organization slows down.

This is the dark mirror of what we explored at the end of the last chapter: leaders who claim they alone have the vision and who force alignment through dictation instead of inspiration. They believe respect comes from authority, not credibility. These explicit-authority leaders declare a path without inviting discussion, mistaking fear for alignment. Under their command, teams comply outwardly while disengaging inwardly, working to avoid conflict rather than achieve greatness.

It's the leadership equivalent of "*because I said so.*" Seniority doesn't always mean smarter. Experience brings valuable insight and intuition, but it is not infallible. Fresh perspectives can break stale patterns. The famous example of Formula 1 pit crews inspiring hospital surgical teams to improve patient handovers — healthcare teams saved lives because the mission mattered more than their egos.

Leadership is also a constantly evolving role that demands continuous learning and adaptation. In *What Got You Here Won't Get You There*, Marshall Goldsmith describes how the very traits that once fueled a leader's success can become blind spots if they fail to grow. Markets change. Requirements shift. What worked in a previous era cannot simply be applied wholesale to new challenges. These past learnings are critical for growth, but they should be treated as guides rather than rigid blueprints.

This challenge is captured perfectly in what's known as the founder's dilemma. Founders often excel at the vision, hustle, and risk-taking needed to start a company. But as organizations grow, the same command-and-control habits or unstructured creativity that once fueled early wins can become obstacles. Companies begin to need process, delegation, and collaborative leadership. When founders cling to their old ways, convinced their instincts alone will scale the business, they risk undermining the very mission they set out to achieve. The most successful founders recognize this shift and evolve by seeking mentors, building diverse teams, and accepting that their role must change if they want the company to thrive beyond the startup stage.

When everyone agrees the mission is the most important thing, these decisions become easier. Reaching out for help or inviting alternative viewpoints is the obvious choice when it improves the mission. Those unwilling to listen or unwilling to try usually act from fear and are often motivated by personal agendas rather than collective success. Product management has no place for ladder-climbers or title-chasers.

The mission must come first, and when it is your guiding light, every other decision becomes clearer.

When a team truly cares about the mission, ego fades. And in a satisfying paradox, personal success often follows naturally as a symptom of the mission's success.

The hardest lift you will ever make is putting your ego down.

The strongest teams are led by people who do not let their ego, vanity, or fear get in the way.

■ CHAPTER 12

12

**THE WEIGHT IS
REAL. OWN IT.**

DON'T DODGE THE HEAVY SETS

The heaviest sets in the gym teach you what you're made of. Your technique frays, your breathing shortens, and every fiber of doubt tries to talk you out of the next rep. But if you dodge those sets, if you forever add weight "*next time*" or cut the range of motion when it gets tough, you never find the limits you're capable of surpassing.

Like a calendar laid out with intention, the night before a big lift becomes a ritual of mental preparation. Scanning tomorrow's workout, knowing there's a 1RM attempt or a set taken to true zero Reps in Reserve, sparks both nerves and excitement. These are the days you celebrate the grind, the day-to-day discipline that built your capacity. The moment the bar bends under load is your chance to prove how far you've come. When you finally log that PR in your fitness app, it's not just a number. It's evidence of your commitment, and the right to earn a well-deserved deload.

Product work has its heavy sets, too. The planning sessions, the heated debates over priorities, the painstaking designs, all culminate in one make-or-break moment: the release. When the new feature goes live and real users experience your work, it's the equivalent of stepping onto the platform. The high isn't just about shipping. It's the culmination of every rep of preparation, and the point where your investment meets reality.

One of my favorite parts of being in product management is visiting my users. My career has been spent building cybersecurity software, but before that, I was a defender who relied on those very tools to protect my teams. That experience infused me with deep empathy for the difficulty of the job, the importance of effective tools, and a healthy resentment for the snake oil promises that plague our industry.

Every time I step on site at a customer's Security Operations Center, I get a mini-experience of my old life. For those few hours, I live in their shoes, seeing the threats they face, the urgency they feel, the challenges that keep them awake at night. I love being in the fight with them, brainstorming new ways to integrate with the rest of their stack, and, most of all, witnessing the

moment when what we built helps them win. Because when their mission succeeds, it's proof that every heavy set was worth it. Their mission becomes my mission.

It's tempting to skip the hard work. To avoid deep discovery with customers. To half-heartedly push a feature to "*check the box*." To let organizational friction excuse mediocrity. But every time we dodge the heavy sets, we trade short-term comfort for long-term weakness. We miss the chance to sharpen our instincts, build resilience, and develop the relationships that make real wins possible.

Like hitting a PR at a competition, seeing a user win with our product feels like hitting my personal best as a product manager. Their success is my lift, and the mission we share is the weight worth carrying. Because the mission doesn't get lighter. Only you get stronger.

THE RESPONSIBILITY TO FINISH

In strength training, a lift isn't counted until it's locked out. Until control is clear, the bar is steady, and the rep is complete. A squat that's buried deep but racked before standing tall, or a deadlift that's ripped off the floor but dropped before lockout, doesn't earn a place on the scoreboard. The intent doesn't matter if the execution isn't finished. The responsibility to finish is about more than effort. It's about seeing your mission through until the work is undeniable.

The same truth echoes through product management. A brilliant idea or well-laid plan means nothing if it stalls at 90%. Too often, teams sprint through exciting early stages — brainstorming, whiteboarding, the rush of discovery — but lose steam when the grind sets in. The last mile demands relentless focus on details that don't win applause: validating integrations, testing obscure edge cases, polishing onboarding flows, aligning final documentation, preparing enablement, and ensuring stakeholders are ready. These tasks can feel tedious, but they are the difference between a product that earns trust and one that frustrates users.

One of the most powerful commitments I made to myself came when Endgame joined Elastic. I promised I would never stop advocating for our Endgame users. That I would ensure their mission remained as critical and deeply embedded within Elastic as it was before we joined. To Elastic's credit, they readily accepted this challenge. Together, we strengthened our platform until our Endgame customers were not only supported, but measurably better protected than ever before. That was the responsibility to finish in action. Not letting the work pause at the point of acquisition or integration, but carrying it forward until the promise made to users was fully honored.

When a user buys a product you helped create, they're placing their trust not just in the brand, but in you. Not every product carries the same stakes. Misprinted playing cards might frustrate a game night but won't bankrupt a family. But in cybersecurity, you're building your customers' shield. They rely on your expertise and vigilance to help defend their business, their livelihoods, and the security of the people who depend on them. When you build security

tools, you're shouldering part of that weight alongside them. And if you let the lift drop before lockout, if you stop short of complete, tested, dependable solutions, you leave them exposed.

It's tempting to skip the late nights of final QA, to brush off corner-case bugs as "*edge scenarios*," to let organizational friction excuse mediocrity. We've already named the trade. Short-term comfort for long-term weakness. In a field where real adversaries wait for missteps, that weakness becomes an open door.

WHEN THE AGENT DID THE WORK, YOU STILL OWN IT

There's a 2026 version of this responsibility that the book has been pointing at since Chapter 1 of v2 and that lands hardest here.

The most dangerous line in any review meeting next year is going to be "*the AI did it*." If you shipped it, you own it. If you signed off on it, you own it. If you let an agent loose in production and didn't read what it was doing, you still own it. AI is a teammate, and like any teammate, it can let you down. The responsibility to finish does not transfer. The model did not ship the feature alone. The model did not pick the customer to deploy to. The model did not write the contract you signed. You did.

In February 2024, Air Canada's AI chatbot told a grieving passenger he could buy a regular fare and apply for a bereavement refund afterward. The policy didn't actually allow that. When the customer tried to claim the refund, Air Canada argued it wasn't responsible for what the chatbot said — the bot was, in their words, "*a separate legal entity*." The tribunal disagreed. The airline owned what its chatbot put in writing. The most dangerous line in any review meeting is "*the AI did it*." A court already disagrees.

We will go deeper on this in the chapter on AI.

Taking personal responsibility for everything you build is not about perfectionism. It's about integrity. It's knowing that when you commit to finishing, you honor the trust your users place in you. Let that trust drive you. Let it fuel you to embrace their mission as your mission. Because what you start may inspire, but what you finish defines you. And only when you lock out the lift does your work count.

THE LIFT AND THE LEGACY

From the outside, both powerlifting and product seem attainable by anyone — and in reality, they are. Anyone can step into either and make an attempt.

However, both require focus and nuance to excel. Minute details like the path of the bar or a user's non-verbal cues separate the truly elite from everyone else. In lifting, a subtle shift in balance or a half-second of hesitation can mean the difference between a successful PR and a missed attempt. In product, a poorly timed question or a missed signal in a customer conversation can derail alignment or cause you to build something that solves the wrong problem.

The barbell is a teacher of both moments and memories. A heavy lift can change your understanding of what you're capable of in a single instant, but the deeper value comes from what that lift means over time. The discipline to prepare, the courage to attempt, and the commitment to finish leave a legacy. Not just in your personal record books, but in the culture you build around you. Those who watch you lift learn not just about strength, but about character.

In product work, the same legacy unfolds. Our choices ripple beyond immediate wins or losses. The way we engage with teams, the care we put into the final details, the honesty of our conversations with customers — these moments compound into reputations that either build trust or erode it. When you choose to embrace a user's mission as your own, you send a message: that their success matters, that they are not alone, and that their mission deserves the best you can give.

I've found some of my proudest moments in product come not from flashy announcements or big conference stages, but from quiet emails and calls months after release. Customers sharing how what we built helped them solve a problem, achieve a goal, or simply feel heard and supported. Those stories remind me that while our work can feel abstract behind code and roadmaps, it becomes real and meaningful in the hands of the people whose lives it touches. Their victories are our legacy.

But legacy can also turn negative if we forget the responsibility we carry. A lift done sloppily in front of new lifters teaches them that form and safety don't matter. A rushed feature full of bugs teaches a team that deadlines outweigh quality. In both lifting and product, we model behaviors that others will follow. Every shortcut becomes permission for the next person to cut corners. Every moment of excellence sets a standard that raises everyone.

There's a quieter version of negative legacy worth naming, because we've already met it twice in this book.

This is the Drift, one more time. We named it in Chapter 1 — the slow inversion of mission into metric, the dashboard glowing green while user trust fades red. We met it again in Chapter 6, recovery-shaped, when Evernote drifted away from itself by sprinting without reflection. The version that shows up here is legacy-shaped, and it's the slowest of the three. It looks like wisdom from a distance. It feels like steadiness from inside. It's the quiet pull of reputation, of *what worked last time*, of becoming protective of the version of yourself that built the success in the first place. The Drift, once it sets in here, takes a generation to clear. The mission is still the antidote. Still.

The lift and the legacy are inseparable. Every rep you finish, every mission you help complete, contributes to a chain of progress that extends far beyond your time under the bar or your name on the release notes. The hours you spend refining the details, the late nights spent debugging stubborn issues, the patience to listen when it would be easier to assume — these aren't just tasks. They're bricks laid in a foundation others will stand on. Your commitment today makes it possible for someone else to rise tomorrow.

Legacy is not built by a single heroic effort. It's shaped by the consistency of your care, knowing that the lift you grind through today, or the product feature you refine until it sings, becomes part of a culture where people believe in giving a shit because they've seen you do it. The real prize of the mission isn't personal glory. It's leaving behind a stronger, more resilient team, product, and community than you found. And legacy is also about ensuring everyone — and I mean everyone — has the same opportunity to excel that you have had. It's about opening doors, sharing knowledge freely, and fighting for a culture where every person has the chance to grow, thrive, and contribute their best.

This is the lift: the act of taking on challenges bigger than yourself, of shouldering weight that matters, of proving what's possible when preparation meets purpose. And this is the legacy: the stories told by those you helped, the

culture of excellence you leave behind, and the quiet confidence that what you built will outlast you and keep serving others.

When you step back at the end of your mission, whether it's a lift on the platform or a product release that lands with quiet power, you'll know the truth: that every rep you finished, every promise you kept, every time you chose the harder right over the easier wrong became part of a legacy worth inheriting. The mission doesn't end when you're done. It lives on through those you've taught, supported, and inspired.

Own the weight. Finish the lift. And leave a legacy that proves giving a shit works.

■ CHAPTER 13

13

**AI IS THE NEW OS.
THE MISSION IS THE
ONLY APP THAT
MATTERS.**

The chapter you've been reading toward.

Across the previous twelve chapters, AI has shown up at the edges of almost every principle in this book. In Chapter 2, it could do the boring reps for you, but the reps that grow you still belong to whoever shows up willing to feel them. In Chapter 6, it became a teammate that doesn't sleep, doesn't deload, and doesn't take Q4 off, which means the recovery problem is now also a problem of deciding when to walk away from the available output. In Chapter 8, it changed what stress-tests look like — from the months-long Log4j shape to attackers moving in seconds. In Chapter 10, it joined the team. In Chapter 11, it surfaced a question we were not yet ready to answer: it can hear what gets said, but can it hear what gets meant? In Chapter 12, it became the most dangerous line in any review meeting: *"the AI did it."*

This is where those threads converge. Not as a defense of AI, and not as an attack on it. As an honest test of whether the book's principles still hold when the work itself is changing under our feet.

The motto that runs this chapter is the same one that runs the rest of the book. *Give a shit.* I planted it in the prologue, where I named its source — VMM-364, the Purple Foxes, the Marine tiltrotor squadron whose pilots fly CASEVAC into enemy fire to bring wounded troops home. I traced it through Nate Fick at Endgame in Chapter 1. I lived a version of it in my own Army intel work in Chapter 8. Now I have to test it against the loudest tool of the year I finished writing this book.

So we will. Twice. Once under the bar. Once in the boardroom.

§13.1 · AI IS THE NEW OS. THE MISSION IS THE ONLY APP THAT MATTERS.

AI UNDER THE BAR

I program with JuggernautAI. It watches my sets, learns from them, and reprograms when it needs to. Fatigue shows up, the plan adjusts. Tweak a hamstring, it backs me off. Hit a PR, it pushes harder next cycle. It is the closest thing to a real coach in your pocket that has ever existed.

But it does not lift the bar for me.

That is the part most people miss when they first hear about AI in training. They hear "*AI program*" and assume the work just got easier. The work did not get easier. The work got smarter, and the responsibility shifted.

The whole system runs on one input: the truth. How many reps did I really have left? Did I actually hit depth, or did I cheat the angle because the camera was off? Did I sleep, or did I tell the app I slept? AI cannot guess. It can only respond to what I tell it. If I lie to the app, the program lies back to me. If I show up honest, the program finds me ground I would not have found alone.

That is the trade. AI gives you a coach. You give AI the truth.

Here is what I have come to believe: AI does not replace the work. It exposes the people who were never doing the work in the first place.

For years, getting a real strength program meant one of two things. Either you could afford a coach you trusted, who knew your body and would adjust your block when your knee was angry, or you got lucky finding the right template online and disciplined enough to follow it without modification. Most lifters never had either. They followed something they read in a forum in 2014. They never deloaded. They never adjusted volume around life. They trained for years and barely moved their numbers.

They were doing the reps. They were not really training.

That is the failure AI just removed. Not the failure of effort. The failure of access.

A good AI program is not magically better than a great human coach. A great coach who knows you, watches you move, and pushes you at the right moment is still the best version of this work. But great coaches are rare, expensive, and hard to vet. Most of the people lifting in garage gyms and corporate gyms and basement bars are never going to find one. AI did not beat the great coach. AI made expert-level programming available to the lifter who was never going to find one in the first place.

That is what democratized expertise actually means. Not the death of the coach. The end of the gatekeeping.

We love to see the numbers go up. That is why we lift. The dynamic, expert-led program is what makes the numbers go up, and AI just put it in everyone's pocket. The lifter who plateaued for three years following a ten-year-old PDF can now run a program that actually responds to her. The dad training in his garage at 5 a.m. between school runs gets feedback he would never have paid for in cash. The fifty-year-old coming back from injury gets a return-to-load curve that no textbook would have written for his body.

The floor of the work just got higher. The ceiling is still the same place it has always been. As far as you are willing to grind for it.

Here is what AI does not change. The bar still bends under load. The night before a heavy set still keeps you up. The platform still asks the same question it has always asked: *what do you actually have, today?* AI cannot answer that for you. It can only set the stage. The lift is still yours.

The motto belongs here too. As I described in the prologue and Chapter 1, *give a shit* is the working motto of my career, traced from VMM-364 through Nate Fick into my own work. AI does not give a shit. That is the part that has not changed and will not change. Hitting depth still requires you to give a shit. The PR still requires you to want it. The grind still requires you to feel it. AI is the program. Giving a shit is the lifter.

So if you are using AI to train, three things are now your job and only yours. Three forms of giving a shit, under the bar.

Show up. Every day, even when the spark fades. Especially then.

Tell the truth. To the app, to your training log, and most of all to yourself. Two reps in reserve means two. Not one and a maybe.

Own the lift. The program did not squat the weight. You did. The PR is yours. The miss is yours. The progress is yours.

That is the trade AI offers under the bar. A smarter program, in exchange for a lifter who still gives a shit. It is the best deal strength training has ever seen.

But only if you take it seriously.

§13.2 · AI IS THE NEW OS. THE MISSION IS THE ONLY APP THAT MATTERS.

AI IN THE BOARDROOM

I run product with Claude as a teammate. It reads my epics and asks me what I missed. It builds wireframes from a written brief in minutes. It summarizes a competitor's last six press releases before my coffee is cold. It drafts sections of a roadmap I would have spent an evening on. It is the closest thing to a senior editor and a junior designer and a research analyst in your pocket that has ever existed.

But it does not feel what the user feels.

That is the seam in this whole conversation. The discourse around AI in product management has been quick to declare the role obsolete. Engineers can prompt their way to an epic. AI can summarize a customer call. Anyone with the right tools can ship a working prototype before lunch. The argument goes: if all of that is true, what does the PM still do?

Let's be honest about the answer. *Robotic* is already the word we use when a person responds without empathy. "*That answer was robotic.*" "*Don't be so robotic with the customer.*" It is not a compliment when we apply it to a human. So why would it be the goal when we apply it to how products get built?

We do not want robots building experiences. We need empathy. We need relation. We need a human.

That motto — *give a shit* — is the line. A model can render empathy convincingly. It cannot live it. It can mimic the language of caring. It cannot actually care. The Purple Foxes flew into enemy fire because someone on the other end was bleeding. That is not a workflow. That is not a prompt. That is what humans do for other humans when nothing else will do. It is also what product work, at its best, is. And it is the part of the work AI is not crossing.

The whole system runs on one input: presence. Did you actually sit with the user, or did you let AI summarize the call? Did you watch them try the workflow, or did you ask the model what frustrated them? Did you carry their problem home with you, or did you tag the recording and move on?

AI cannot guess. It can only respond to what you give it. Give it your assumptions and it gives back a confidently written version of your assumptions. Give it a real problem you have actually witnessed and it gives back leverage. The difference between a PM who is good with AI and one who is great is whether they have actually been in the room.

That is the trade. AI gives you a teammate. You give AI your time with the user.

Here is what I have come to believe. AI does not replace the PM. It exposes the people who were never doing the empathetic work in the first place.

Melissa Perri, who has trained more product leaders than almost anyone in the field, frames it cleanly: *AI excels at the how. You own the why.* That is the right line. AI is a stunningly good *how* engine. Specs, drafts, competitive scans, design variations, prototype builds, edge-case enumeration — all faster, all cheaper, all more thorough than what a human alone produced two years ago. But the *why* has not moved. The user. The mission. The right problem to solve in the first place. That is still a question only a human in the room with another human can answer.

This is also why the LinkedIn experiment is worth taking seriously. Tomer Cohen, LinkedIn's Chief Product Officer, has been replacing traditional PM roles with what he calls *Full Stack Builders* — generalists who design, prompt, prototype, and ship end-to-end with AI rather than handing off across functions. The structural argument is sound. The role boundary is doing less work than it used to. I do not disagree with him on the mechanics.

But the question is not whether you call the role PM or full-stack builder. The question is whether the person doing the work can feel what the user feels. A full-stack builder who can do that is a product creator. A full-stack builder who cannot is shipping faster mediocrity in a smaller org chart. The title is downstream of the empathy.

Marty Cagan has made this point for years, and AI made it urgent. The title PM is not protected. The work of *product creation* is. People in product manager roles who never developed product sense are vulnerable. People in any role who can sit with a user, name the real problem, and own the outcome are the ones who matter. AI raised the floor on execution. It did not raise the floor on judgment.

The PM's epic itself is changing because of this. A great epic in 2026 should be readable to AI. It should explain the problem clearly enough that an AI editor can ask sharp questions back. It should describe the experience cleanly enough that an AI prototyper can wire a working version. It should position competitively in language an AI strategist can stress-test. The best PMs I work with use that as their bar. If I uploaded this epic to Claude, would the AI understand the problem? Would it know what experience solves it? Would it know what competitors are doing nearby? When the answer is yes, you have written a real epic. When the answer is no, you have written a paragraph that hides what you do not yet understand.

But here is the catch. Pawel Brodzinski, who writes about this more clearly than most, names a structural truth the discourse keeps missing: AI by definition pulls toward the median. The model produces what the data's center of gravity says is most likely. Creativity, by definition, lives at the fringes of the distribution. So an AI-generated epic is by definition a median epic. Confident. Clean. Average. The PM's job is to be the deviation that finds the actual user insight the median was never going to surface.

AI gives you a faster median. The human is the difference between median and meaningful.

What AI gives the PM in 2026 is access. The PM at a five-person startup can produce at the pace of a PM at Stripe. The PM with three product surfaces can keep up with all three. The new PM ships in their first month at a level that used to take a year to learn. The floor of the work just got higher.

The ceiling is still the same place it has always been. As far as you are willing to sit with a user and care about their problem.

Here is what AI does not change. The customer still wants someone who sees them. The engineering team still needs a clear *why*. The roadmap still has to make a choice. The launch still has to be owned. The user still wants to know there is a human on the other side of the product who took their mission seriously. AI cannot do that part. It can only set the stage.

So if you are running product with AI, three things are now your job and only yours. Three forms of giving a shit, in the boardroom.

Show up. In the room with the user, not the transcript. On the call, not in the summary. AI cannot replace your time. It can only multiply it.

Tell the truth. About what you actually know versus what you assumed. About what the data says versus what your gut says when they conflict. About what your team can ship versus what AI made look easy.

Own the outcome. You signed off on it, you own it. As I named in Chapter 12, the most dangerous line in any review meeting is *"the AI did it."* The agent did not write the spec alone. The model did not pick the customer to ship to. AI is a teammate, and like any teammate, it can let you down. The responsibility to finish does not transfer.

That is the trade AI offers in the boardroom. A faster team, in exchange for a PM who still gives a shit. It is the best deal product management has ever seen.

But only if you stay lovable enough to do it.

Two halves of one argument. Under the bar, AI raised the floor of training. In the boardroom, AI raised the floor of execution. In neither case did AI raise the ceiling of the work that matters. The lift is still yours. The user is still theirs. The mission is still the only thing worth running on top of any of it.

There is a metaphor I have used in my published work, and it is the right one to land here. AI is the Iron Man suit. The suit makes you faster, stronger, harder to kill. It does not tell you what is worth fighting for. It does not tell you who you are fighting for. It does not tell you when to take the suit off and look someone in the eye. The human inside the suit is still the only one who can answer those questions.

The suit changed everything about the work. The mission inside the suit did not move at all.

The second metaphor, also from my published work, names what's structurally happening: AI is the new operating system. The interface layer is shifting from applications to AI environments themselves. The conversation is becoming the workspace. Everything that used to be an app — your dashboard, your roadmap tool, your customer interview, your training log — is becoming an app inside an AI environment that stitches them together.

That is real. It is also incomplete.

The OS, on its own, runs nothing. It needs an app to run. And there is one app that AI cannot build, ship, debug, or maintain on its own. The mission. The decision about who you are building for, and why, and what you owe them. That decision belongs to a human. It always has. It still does.

AI is the new OS.

The mission is the only app that matters.

And the only thing that runs the app is a person who gives a shit.

That is the argument of this chapter, and the argument of the year I added it.

The Drift will show up here too. We named it three times in this book. Mission overridden by metric in Chapter 1. Sprinting without recovery in Chapter 6. Legacy-shaped Drift in Chapter 12. The AI-shaped Drift, the fourth one we have now lived long enough to see clearly, looks like a team that has stopped asking *why* because the *how* has become so frictionless. Why interview the user when AI summarized her last call? Why agonize over the brief when AI will refine it? Why slow down to feel the user's problem when AI can render the problem for you in plausible English?

That is the Drift in its 2026 shape — same shape as the others, same antidote, too.

The mission. The user. The motto.

Give a shit. About the work. About the people. About the outcome.

That is the only app that matters.

Now go run it.

▪ CONCLUSION

14

**GIVING A SHIT
WORKS**

At every stage of my life — whether wearing a uniform, defending networks, under a barbell, or building a product — I kept running into the same lesson: giving a shit works.

We've already talked about those nights advocating for air assets while serving as an intelligence analyst overseas. When the real challenge wasn't the data, but the conversations, the negotiations, and the weight of knowing that if I couldn't make the case clearly, people downrange might not get the support they needed. Giving a shit wasn't optional. It was the difference between letting things happen and fighting for what mattered most.

I saw the same pattern later as a SOC analyst and lead defending some of the largest government networks in the world. The best defenders weren't always the smartest or the fastest. They were the ones who gave a shit enough to search for the serendipitous connections between seemingly disparate events — the deep analyses that led to the discovery of attacks in time to stop damage and loss to critical systems. The analysts with empathy for teammates, respect for users, and pride in their work were the ones who set the tone.

When I moved into product management, giving a shit meant obsessing over the details of problems I'd lived myself. How slow alert triage wasted defenders' time, or how confusing interfaces kept critical data hidden. At Endgame, in a mature, crowded EDR market, our willingness to listen to defenders, to give a shit about their realities, helped us build a product recognized as a Visionary by Gartner. When Endgame was acquired by Elastic, giving a shit meant betting on the users again. Launching Elastic SIEM, building the Elastic Security division from the ground up, and in just a few years, earning recognition from Gartner and Forrester as one of the leaders in a fiercely competitive SIEM space. But more important than any analyst recognition was the validation from our users. Real stories of success, attacks mitigated, remediation accelerated, and defenders delighted. That was the true measure of whether we gave a shit enough.

Under the barbell, giving a shit took a different shape. It meant tracking tiny progress when nothing felt like it was changing, resting on days when ego screamed to push harder, and spotting a training partner with the same care you'd want from them. It meant seeing that strength isn't just what you can lift, but how you show up for others. Because real strength lifts others.

And then came the year I finished writing this book, and the year I had to add a chapter.

The chapter you just read names what AI changed and what it didn't. I won't relitigate the argument here. I'll just say it once more, in the place this book has always been heading. Real strength is the part AI cannot do. The mission is what only a human can run. The motto still works. Maybe more than ever.

That's the lesson I want you to carry with you from this book: giving a shit works. It will cost you something. Time, energy, vulnerability. But what you get in return is better than any shortcut: teams who trust you, skills that endure, a mission that matters, and a life of meaning.

So don't let cynicism, busyness, or ego keep you from caring deeply. And don't confuse kindness for weakness, or empathy and compromise for giving in. In a time when the world seems to be forgetting our shared humanity, be the bastion of integrity. Be the one who shows heart. Care enough to show up, to do the hard reps, to listen, to adapt, and to lift the people around you. Give a shit. About the work. About your mission. And most importantly, about others. Because when you do, you'll build something stronger than you ever thought possible.

Now go build it.

COLOPHON

Set in Oswald, Merriweather,
and JetBrains Mono.

Designed and built in 2026.

Read the live edition at missionbuilt.io.

Source openly on GitHub.

■ CC BY-NC-SA 4.0 ■

MISSION BUILT · END

**REAL STRENGTH IS
LIFTING OTHERS.**

