

Introduction: The Origin of the Moron Test

I never met my grandfather.

He died at fifty-one. A heart attack — sudden and final, the kind of departure that leaves no room for goodbye. I was not yet born. I would never hear his voice, never watch him work, never sit across from him and absorb the thing my father spent a lifetime trying to describe.

But I know him.

I know him the way you know a river by its canyon — by the shape of what it carved through.

In Los Angeles, in the years before technology meant software, my grandfather maintained systems that most people couldn't comprehend and no one was allowed to get wrong. Entire floors of office buildings filled with early IBM and Western Electric equipment — machines that would eventually become the backbone of AT&T. These weren't devices you troubleshoot with a manual. They were organisms. Interconnected, temperamental, and unforgiving. They filled rooms. They carried institutional weight. One miscalculation didn't produce an error message. It shut down organizations.

What my father remembered — what he told me, carefully, across years of conversations that I didn't fully understand until much later — wasn't the scale of the work.

It was the *thinking*.

My grandfather was obsessed with sequence. With dependency. With the question that most people skip past on their way to the answer: *What has to be true before this can work?*

He believed that complexity was never an excuse for disorder. That a system should never demand heroism from the people inside it. That if something failed, the failure didn't start at the point of collapse — it started upstream, in a decision that someone made without thinking far enough ahead.

He believed in this so deeply that when he and my grandmother built their moving company, they didn't call it Westlake Moving. They didn't call it Westlake Transport. They called it Westlake Moving *Systems*.

That word — *systems* — wasn't marketing. It was identity. A declaration that the way things connected mattered more than the things themselves. That the invisible architecture underneath determined whether the visible work held together or came apart.

And it worked. Westlake Moving Systems survived the Great Depression. Not because the economy was kind to moving companies — it wasn't kind to anyone. Because the business was built on a way of thinking that didn't break under pressure.

My father talked about my grandfather with a reverence that bordered on physical. You could feel it in his body when he spoke about him — a straightening, a weight. But underneath that reverence lived something heavier.

A longing that never resolved. His father had died young, and the shadow of that loss — the things left unsaid, the lessons left half-delivered, the conversations that should have happened and never did — became something my father spent his entire life trying to complete.

He became the inheritor of a way of seeing the world. And he passed it to me — not in speeches, not in lectures, but in the spaces between his words. In the long pauses before he answered a question, as if he were checking his answer against a standard only he could see. In the way he refused shortcuts, even easy ones. In his quiet, relentless insistence that if something didn't make sense, you stopped and fixed the thinking before you touched the work.

My father was a math major at Pomona College. He earned his MBA at Northwestern. He was accepted to Stanford Law. His path was clean, linear, and prestigious — the kind of trajectory that looks inevitable in retrospect.

Then his father died.

And the trajectory ended.

He went home. No deliberation. No self-pity. No calculation about what he was giving up. He went home because the family business needed someone, and he was someone. He took over Westlake Moving Systems — not as a fallback, not as a consolation — as a duty.

And over time, with the same methodical thinking his father had taught him, he grew it into something formidable.

But the lesson he passed to me wasn't about growth.

It was about *judgment*.

He taught me to think in second and third order. Not "what happens next," but "what happens after what happens next." He taught me to anticipate how humans would behave inside systems — not how they *should* behave, but how they *would* behave when they were tired, distracted, rushed, or afraid. He taught me to design workflows that still worked on bad days, with imperfect people, under real pressure.

He taught me that the gap between a system that works in theory and a system that works in practice is almost always a failure of empathy — a designer who understood the machine but not the human standing in front of it.

I didn't learn any of this in a classroom.

My earliest memories are physical. The smell of cardboard and machine oil. The low groan of a loaded truck settling onto its suspension. The scrape of a dolly across concrete at five in the morning — work starting before anyone was awake and ending only when it was finished.

No speeches. No chest-thumping. Just outcomes.

I learned early that real competence is quiet. It doesn't posture. It doesn't need credit. It just moves forward — one correct decision at a time.

Football sharpened everything.

I played at Brigham Young University under LaVell Edwards — a man who understood something about winning that most people misinterpret when they hear it.

We were taught that losing was never an option. Not as bravado. Not as a slogan on a locker room wall. As *assumption*. Every practice, every film session, every game — we inhabited the belief that we would win. There was no alternate storyline. No contingency for failure. Winning wasn't something we hoped for. It was something we breathed.

But what made it powerful was how we carried it. We didn't boast. We didn't taunt. We shook hands before kickoff. We wished opponents well. We told them, honestly, that we hoped they played the best game of their lives.

And then we beat them.

Quietly.

All the points. None of the noise.

That idea — that confidence lives in clarity, not volume — became something I carried into every room I entered for the rest of my life.

Over the course of my career, I kept being drawn to the same kind of problem. Not the glamorous ones. The broken ones. The ones everyone had accepted because "that's just how it works."

One of the first times I saw the pattern at full scale was with a global electronics manufacturer. They didn't need advice. They needed capability — systems, processes, logistics infrastructure that no one else could bring to bear the way we could. We weren't theorizing about supply chain optimization. We were moving real product, at real volume, under real constraints.

By redesigning how goods were staged, sequenced, handled, and delivered, we cut their costs dramatically and compressed their time to market across the United States. We built a delivery experience seamless enough to support the kind of growth that compounds.

We won because we could see the whole system at once. And because we could execute what we saw.

Years later, I worked with the world's largest retailer on a problem that everyone else had already surrendered to: store remodels. Twenty million dollars per project. Damage-prone. Chaotic in ways that had been normalized through sheer repetition — year after year, the same waste, the same breakage, the same resigned acceptance that this was simply the cost of doing business.

It wasn't.

By tearing the process apart — sequence, accountability, material flow — and rebuilding it from first principles, we transformed the outcome. Quality improved. Damage plummeted. We saved them sixty cents on every dollar of a twenty-million-dollar remodel.

That engagement didn't produce a case study. It produced a decade-long relationship.

And again, the breakthrough wasn't genius. It was the refusal to accept something that didn't make sense just because it had always been that way.

Somewhere along the way, I noticed something about myself that I hadn't expected.

I had a gift for compression. Not summarizing — *compressing*. Taking something sprawling and tangled and reducing it to a single image, a single phrase, a single frame that made the truth land in a way no amount of explanation could.

Years after first working with that electronics manufacturer, I returned to resell them. New deal. New stakeholders. A room full of people who needed to understand, quickly, why we were the right partner at their scale.

Instead of leading with charts and capability decks, I opened with an image: an eight-hundred-pound gorilla. Teeth bared. Unapologetic.

The message was simple. *You are the dominant force in your market. We are the only logistics partner built to match that force. Anyone else — your weight would crush them.*

Before the meeting, I had sent my deck to a vice president I worked under — a woman named Dan. She reviewed it and called me.

"You cannot present this. Take it out."

I told her no. This was my client. My relationship. My sale. And this was the image that would make the truth impossible to ignore. I told her flat out I was going to do it.

She paused. Then: *"Okay. I guess we'll see where this one lands."*

I presented it. The gorilla filled the screen. The room went quiet — and then it broke. You could see recognition move across their faces. Not recognition of the gorilla. Recognition of themselves.

Weeks later, I walked into the implementation meeting. Two hundred people were already seated. Every single one of them was wearing a t-shirt with that gorilla printed on it.

They had made them as a surprise.

Two hundred people, wearing my idea on their chests. The idea a vice president told me to kill.

People don't remember explanations. They don't remember slide decks or capability summaries or competitive matrices.

They remember clarity.

They remember the moment something they already knew became impossible to deny.

I stepped away from the moving industry for nearly seven years.

The simple truth is this: when you move people's belongings — their grandmother's china, their children's first drawings, the furniture that sat in a marriage that no longer exists — people cry. Not often. But enough. And if you have even the beginning of a human bone in your body, hearing that crying is hard to take. Moving is not for the faint of heart. The weight of it — not the physical weight, the emotional weight — accumulates.

My opportunities in the broader logistics space were too good to pass up, and I needed the distance. So I left.

When I came back, it wasn't nostalgia that pulled me. It was obligation. The sense that an industry I understood deeply was being failed by the technology that was supposed to serve it.

Not long after my return, a private equity firm asked me to take over a moving-industry technology startup called NexMuv.

On paper, the software was ambitious. Impressive, even. It had vision. It had investment. It had the kind of feature set that looks transformative in a pitch deck.

In practice, I spent more than two years apologizing.

Apologizing for what the system couldn't do. Apologizing for limitations that users didn't understand because no one had been honest with them. Apologizing for a product built with intelligence but without alignment — without a clear grasp of how real people, under real pressure, would actually try to use it.

The software wasn't broken in the way that broken things are obvious. It was *misaligned*. It did things. Just not the right things, in the right order, for the right reasons. And the people on the other end — the moving company operators, the dispatchers, the salespeople trying to close jobs — they didn't experience "misalignment." They experienced frustration. Confusion. Failure.

And I was the one on the phone explaining why.

I can tolerate almost anything in business. Bad timing. Bad luck. Difficult markets. Long odds. What I cannot tolerate — what is nonnegotiable, what I will not abide under any circumstances — is having my ethics called into question.

And then it happened.

After wiring funds to an organization to fulfill a commitment, that organization's leader accused me of misrepresenting the software. Of overpromising what it could do. Of being, in his words, unethical in how I had sold it.

He wasn't questioning the product. He was questioning my character.

I sat in silence after that conversation ended. Not moving. The word hung in the air — *unethical* — and I could feel it settling into my chest like something physical. Then the shock burned off and the fury arrived. If there is one thing I am not — one thing I have never been, one thing I will fight to my last breath to protect — it is that. My integrity is not negotiable. It is not up for discussion. It is not a line item that gets weighed against revenue or convenience or someone else's disappointment.

But the fury wasn't the worst part.

The worst part was the week that followed.

I called dozens of people. Colleagues. Partners. People I had worked with over years, over decades. And I couldn't stop myself from asking them — each one, in different words, circling the same wound — *Am I unethical? Is that who I am? Is that what you've seen?*

Every single one of them said no. Emphatically. Some of them laughed at the question. Some of them got angry on my behalf.

It didn't matter. The fact that I had to ask — the fact that a system I was responsible for had put me in a position where someone reasonable could look at me and see a man who couldn't be trusted — that was the fracture. Not in my reputation. In my certainty.

I knew, in the silence after those phone calls, that something had to change. Not incrementally. Not eventually. Not after the next sprint cycle or the next funding round.

Now.

I had spent years studying artificial intelligence — not chasing the hype cycle, but understanding where it created genuine leverage. Where it simplified systems honestly. Where it reduced friction rather than hiding it. And where, quietly, it made things worse by adding complexity disguised as sophistication.

I took the NexMuv platform — the same software that had consumed more than two years and roughly a million dollars to build — and I rebuilt it.

Not version two. Not an upgrade. From the ground up.

In less than a month. At a fraction of the cost.

And for the first time, it worked the way it was always supposed to.

Not marginally better. Not "improved." *Ten times better.* Faster. Cleaner. Aligned with how the people using it actually thought, actually worked, actually needed it to behave.

Sitting in the quiet after that rebuild — watching a system do effortlessly what the previous version had spent two years struggling to approximate — I realized something I should have seen long before.

I had been applying the same filter my entire life.

Every logistics redesign. Every sales presentation. Every system I had ever built, evaluated, or torn apart. The same questions, asked in the same order — inherited from a grandfather I never met, refined by a father who never stopped teaching, pressure-tested under stadium lights and inside loading docks and across conference tables in rooms where the wrong answer cost real money.

Does this actually make sense?

Does it align with how humans behave — not how we wish they'd behave, but how they actually do when they're tired and overwhelmed and out of patience?

Does it reduce friction, or just relocate it?

Does it work without heroics?

And if it fails — was the failure predictable? Was it upstream, in a decision someone made without thinking far enough ahead?

I had carried that filter for decades. Used it a thousand times. Never gave it a name.

Sitting in that room, watching something finally work the way it should have worked from the beginning, I gave it one.

The Moron Test.

Not as an insult. The people who fail it are rarely stupid. They're usually brilliant — brilliant people who built something complex when something clear would have worked. Brilliant people who confused sophistication with usefulness. Brilliant people who designed for the system instead of the human standing in front of it.

Not as a joke. The consequences of failing it are too expensive, too human, and too preventable to be funny.

As a discipline.

A way of asking — before the code is written, before the interface is designed, before the system is shipped — whether this thing you are about to build will make the person using it feel capable. Or whether it will make them feel like they need help. Whether it will meet them where they are. Or whether it will demand they become someone they're not.

Most failures aren't caused by stupidity. They're caused by things that seemed obvious at the time — decisions that felt smart in the meeting and proved cruel in the field.

And once you learn to see that — once you learn to test the thinking before it becomes the system — you can't unsee it.

You see it in the software that requires a training budget to operate. In the app that needs a tutorial to explain itself. In the dashboard built for the person who designed it instead of the person who has to use it. In the hundred-thousand-dollar vehicle that makes a grown man swear because no one asked what would happen when he sat down without a guided tour.

You see it everywhere.

And once you do, you can't look away. You can only build differently.

This book is that test. Applied to everything.

Let's begin.