

# Doug Sparks

A man in a dark suit and tie stands against a blue sky background. Instead of a head, there is a large, white, fluffy cloud. The text 'Best Practices Are Lying to You' is overlaid on the right side of the image.

## Best Practices Are Lying to You

What 47 Years Across Wall Street  
and Healthcare Taught Me About  
Systems

# **Best Practices Are Lying to You**

## **What 47 Years Across Wall Street and Healthcare Taught Me About Systems**

Author's Note: A Confession Before We Start  
I should probably begin with a confession.

For a long time, I believed in best practices. I trusted them. I taught them. I defended them in meetings where everyone nodded seriously while silently hoping someone else would ask the uncomfortable question.

Best practices are reassuring. They suggest that somewhere—possibly in a very nice conference room—someone has already figured this out. That if we just follow the steps, preferably in the right order and with the right software, everything will work out fine.

Forty-seven years later, I've learned something less comforting, but far more useful: Best practices don't fail because people ignore them.

They fail because real systems don't behave the way best practices expect them to. I've spent time on Wall Street, where elegant models meet emotional markets, usually before lunch. I've spent decades in healthcare, where human biology, behavior, incentives, technology, and time all collide—often in the same afternoon. I've watched incredibly smart people design systems that worked beautifully... right up until real humans got involved.

This isn't a book about how everyone is doing everything wrong. It's about why smart systems produce confusing outcomes, even when everyone involved is competent, well-intentioned, and trying hard.

I'm not here to preach. Preaching assumes the problem is ignorance. Experience teaches you the problem is usually something else entirely. Think of this book as a guided tour of patterns I've seen repeat themselves across finance, healthcare, technology, and organizations of every shape and size. If there's humor here, it's mostly the kind that comes from recognition. If there's wisdom, it's earned the slow way.

If you're looking for a checklist, this probably isn't your book.  
If you've ever wondered why the "right" solution didn't work the way it was supposed to—welcome. You're in good company.  
Let's talk about systems.

### Table of Contents

Chapter 1: The Myth of the Perfect System

Chapter 2: Incentives — The Invisible Hand Nobody Talks About

Chapter 3: Networks Are Alive (Even When We Pretend They Aren't)

Chapter 4: Healthcare Didn't Get Complicated. It Always Was.

Chapter 5: Technology Is Never the Point

Chapter 6: Scale Changes Everything (Usually for the Worse)

Chapter 7: Designing for Humans, Not Heroes

Chapter 8: A Better Way to Think About Systems

About the Author

## **Chapter 1: The Myth of the Perfect System**

### **Every system looks brilliant on the whiteboard.**

I've never seen a bad system design during a presentation. Arrows flow smoothly. Boxes align perfectly. Dependencies are labeled. Risks are "mitigated." Someone inevitably says, "This should scale nicely," which is presentation-speak for we haven't tested this yet.

Then Monday happens.

Suddenly the system is exposed to a factor that didn't appear on the diagram: reality. People misunderstand steps. Incentives pull in opposite directions. Technology behaves exactly as designed—and completely differently than expected. Someone invents a workaround by Tuesday morning, and by Friday it's considered "the way we do things."

At that point, someone usually says, "The system is fine. The problem is adoption."

That sentence has ended more productive conversations than almost anything else I've heard.

### **Why We Love Perfect Systems**

The idea of a perfect system is deeply appealing. It promises control in a messy world. It suggests that with enough intelligence, planning, and discipline, we can design uncertainty out of existence.

Wall Street loves this idea. Healthcare depends on it. Technology sells it very convincingly.

And to be fair, systems matter. Design matters. Process matters. But here's the part we tend to skip over:

Most systems are designed for how we wish people behaved, not how they actually do.

We assume rational decisions, consistent behavior, and shared priorities. We assume information flows cleanly. We assume people will follow the process because the process is logical.

Anyone who has worked in a real organization is probably smiling right now.

### **When Smart Design Meets Human Reality**

Early in my career, I watched teams build systems that were mathematically sound, procedurally airtight, and operationally doomed. Not because they were poorly designed—but because they were fragile. They worked perfectly as long as nothing unexpected happened.

### **Unexpected things, as it turns out, happen quite a lot.**

In finance, fragility shows up when markets behave irrationally longer than models assume they should. In healthcare, it appears when patients don't follow care plans, clinicians adapt workflows to survive the day, and administrators chase metrics that only vaguely resemble outcomes.

In technology, it shows up when users do things no one anticipated—often because the system made the "right" thing harder than the "easy" thing.

The problem isn't intelligence.

It's assumption.

Perfect systems assume stability.

Real systems live in motion.

Complexity Isn't the Enemy—Denial Is

We tend to treat complexity as a design flaw. If something is complicated, we assume it must be poorly thought out. If it's messy, we try to simplify it until it fits into our tools, dashboards, or governance models.

But complexity is often a signal, not a mistake.

Healthcare isn't complex because someone failed to organize it properly. It's complex because human bodies, emotions, economics, culture, and time interact in ways that refuse to stay in neat boxes.

Financial markets aren't unpredictable because traders lack discipline. They're unpredictable because millions of incentives collide every second.

When we pretend complexity shouldn't exist, we design systems that break under pressure—and then blame the people inside them.

How Systems Really Fail

Most systems don't fail dramatically.

They erode.

They get patched. Workarounds multiply. Exceptions become routine. Performance slips just enough that everyone feels the strain, but not enough to justify stopping everything and starting over.

Eventually, people stop trusting the system and start trusting each other instead.

Ironically, that's often when things start working again.

What This Book Is Actually About

This book isn't about rejecting best practices. It's about understanding their limits.

Best practices are snapshots taken under specific conditions. They're useful—until the conditions change. And the conditions always change.

What I've learned over 47 years is that resilient systems don't aim for perfection. They assume imperfection. They don't fight human behavior; they design around it. They don't punish deviation; they learn from it.

If that sounds less tidy than a best practice manual, that's because it is.

But it's also a lot more honest.

**Chapter 2: Incentives — The Invisible Hand Nobody Talks About**

If you want to understand why a system behaves the way it does, don't start with the mission statement.

Start with the incentives.



This is one of those lessons that sounds obvious once you've learned it—and remains surprisingly easy to ignore anyway. I've seen beautifully worded visions posted on walls, websites, and slide decks while the system beneath them quietly did something else entirely.

Not because anyone was dishonest.  
Because incentives always win.

### Why Good Intentions Lose So Reliably

Most systems are designed by people with good intentions. I've rarely met anyone who wakes up hoping to build something inefficient, unfair, or dysfunctional. Yet those outcomes appear with remarkable consistency.

The reason is simple: intentions are static. Incentives are dynamic. Intentions describe what we say we want. Incentives shape what people actually do, especially when they're busy, stressed, or under pressure—which is to say, most of the time.

On Wall Street, this is almost too obvious to mention. You can talk about long-term value all day, but if compensation rewards short-term performance, guess which one wins? In healthcare, the same pattern plays out more quietly but just as powerfully. We talk about patient-centered care, while reimbursement models reward volume, documentation, or compliance metrics that only loosely correlate with outcomes.

The system isn't broken.  
It's doing exactly what it's being paid to do.

### The Polite Fiction of Alignment

Organizations love the word alignment. Strategy decks are full of it. Leaders speak earnestly about getting everyone "on the same page."

What they usually mean is:

"We hope people will behave the way we designed the system, even if the incentives suggest otherwise."

That hope rarely survives first contact with reality.

Alignment doesn't happen because people agree with the mission. It happens when incentives—financial, professional, social, and emotional—pull in the same direction. When they don't, people make rational choices inside irrational systems and then get blamed for the results.

One of the quiet tragedies of poorly designed incentives is how often they turn good people into scapegoats.

### You Get What You Measure (Whether You Like It or Not)

Measurement feels objective. Numbers feel neutral. That's part of their appeal. But measurement is never neutral—it's directional.

What you measure tells people what matters. What you reward tells them what really matters. And what you ignore tells them what's safe to ignore.

In finance, metrics drive behavior so directly that entire industries exist to optimize around them. In healthcare, metrics are often layered on top of already complex clinical realities, creating situations where clinicians are forced to choose between doing what's right for the patient and what's required by the system.

When that choice exists, the system has already failed—quietly, politely, and with excellent documentation.

### The Workarounds Are the Real Design

One of the most revealing moments in any organization is watching how people actually get things done.

Not the official process.

The unofficial one.

The shadow spreadsheets.

The side conversations.

The “here's how it really works” explanations given to new hires.

These workarounds aren't signs of laziness or resistance. They're signs that the formal system doesn't align with reality. People adapt because they have to. And over time, those adaptations become the true operating system.

Ironically, leaders often respond by adding more rules, more controls, and more oversight—further misaligning incentives and driving workarounds deeper underground.

The system becomes more complex, less transparent, and harder to change. Everyone feels the friction, but no one feels empowered to fix it.

### Incentives Don't Care About Your Values

This is uncomfortable to say, but it's important:

Incentives are indifferent to values.

You can value quality, safety, innovation, compassion, and collaboration—and still build systems that discourage all of them. Not because values don't matter, but because incentives speak louder, faster, and more consistently.

People don't wake up trying to undermine the system. They wake up trying to survive it.

When incentives reward speed over thought, documentation over understanding, compliance over outcomes, or appearances over substance, people respond intelligently within those constraints.

Calling that “resistance to change” misses the point entirely.

Designing With Incentives in Mind

The most effective systems I've seen share a common trait: they make the right behavior the easiest behavior.

Not the most heroic.

Not the most self-sacrificing.

Just the easiest.

They assume people are busy, imperfect, and occasionally distracted. They reduce the cost—emotional, cognitive, and operational—of doing the right thing. They align rewards with outcomes instead of proxies whenever possible.

And when alignment isn't perfect (because it never is), they build in feedback loops instead of blame.

### A Practical Test

If you want to evaluate any system quickly, try this:

Ask people what frustrates them.

Then ask what gets rewarded.

The gap between those answers will tell you almost everything you need to know.

### Where This Takes Us Next

Incentives explain why systems behave the way they do. But they don't explain how behavior spreads, adapts, and evolves over time.

For that, we need to talk about networks—not just technical ones, but human ones. Systems don't operate in isolation. They interact, influence, and respond like living things.

Which brings us to the next uncomfortable idea:

Systems aren't machines.

They're ecosystems.

## Chapter 3: Networks Are Alive (Even When We Pretend They Aren't)

One of the most persistent mistakes we make when designing systems is treating them like machines. Machines are predictable. If something breaks, you replace the part. If performance drops, you tune the settings. Cause and effect behave nicely, preferably in a straight line.

Networks do not behave that way.

Networks adapt. They reroute. They compensate. They remember. And sometimes, they resist.

The trouble is that we keep using machine logic to manage network behavior—and then act surprised when the results feel... alive.

### Why Linear Thinking Keeps Letting Us Down

Linear thinking is comforting. It says:

- If we do A, B will happen.
- If B doesn't happen, we didn't do A correctly.
- Try harder.

This works reasonably well for simple problems. It works terribly for interconnected ones. Healthcare is a network. Financial markets are networks.

Organizations are networks. Even technology platforms are less about code than about the behaviors that form around them.

Change one node, and others respond—sometimes immediately, sometimes months later, sometimes in ways that make no sense unless you zoom out.

I've watched small policy changes ripple through organizations like shockwaves, producing outcomes nobody anticipated. Not because anyone failed to plan, but because networks respond as wholes, not parts.

### Emergence: When the System Does Something New

There's a word for this phenomenon: emergence.

Emergence is what happens when a system produces behaviors that weren't explicitly designed. Traffic jams form without accidents. Informal leaders emerge without titles. Workarounds spread without permission.

From the outside, it looks chaotic. From the inside, it's perfectly rational.

People respond to constraints, incentives, and each other. Over time, patterns stabilize—not because anyone declared them official, but because they work well enough to survive.

The mistake we often make is trying to stamp out emergence instead of learning from it.

### Why Control Feels Good (and Fails Quietly)

When systems behave unpredictably, the instinctive response is control. More rules. More oversight. More reporting.

Control feels responsible. It signals leadership. It produces dashboards. Unfortunately, it also ignores how networks actually function.

Networks don't respond well to force. They respond to context. Push too hard in one place, and adaptation shows up somewhere else—usually where you're not looking.

In healthcare, excessive controls often shift burden to clinicians, increasing burnout without improving outcomes. In finance, tight constraints can push risk into less visible corners. In organizations, heavy governance slows decision-making while informal influence fills the gap.

The system doesn't become simpler. It becomes sneakier.

### Information Flow Is the Real Power

In networks, power doesn't sit where the org chart says it does. It sits where information flows fastest and most reliably.



Who knows what's really happening?  
Who hears problems early?  
Who connects people across silos?  
Those are the nodes that matter.

I've seen junior staff quietly hold systems together because they understood the informal network better than anyone with a title. I've also seen senior leaders struggle because they relied exclusively on formal channels that filtered reality into something more polite.

Healthy networks encourage honest signal. Unhealthy ones punish it—then wonder why problems arrive late and expensive.

### Resilience Beats Optimization

Optimization sounds sophisticated. It implies efficiency, precision, and maximum output. Resilience sounds boring by comparison. Until something goes wrong.

Optimized systems are fragile. They leave no room for variation, error, or surprise. Resilient systems expect disruption and absorb it without collapsing.

In living networks, resilience comes from:

- Redundancy
- Diversity
- Feedback loops
- Local decision-making

These qualities look inefficient on paper. They're invaluable in practice.

### Designing With Network Behavior in Mind

The most effective system designers I've known didn't try to control networks. They tried to understand them.

They watched how work actually moved. They paid attention to informal communication. They treated unexpected behavior as data, not defiance.

Most importantly, they asked better questions:

- Where does information slow down?
- Where do people improvise?
- What behaviors repeat even when discouraged?

Those answers reveal the system you actually have—not the one you hoped you built.

## The Human Thread

At the center of every network is a human thread. People interpret signals, make tradeoffs, and adapt in real time. They respond emotionally as well as rationally. They carry context no system can fully capture.

Ignoring that doesn't make systems cleaner. It makes them brittle.

Which leads us directly to healthcare—where complexity isn't a flaw, reductionism keeps failing us, and pretending otherwise has real human consequences.

## Chapter 4: Healthcare Didn't Get Complicated. It Always Was.

Every few years, someone declares that healthcare has become “too complicated.”

This is usually followed by a proposal to simplify it.

After nearly four decades working in and around healthcare, I can say this with some confidence: healthcare didn't become complicated. We just keep rediscovering that it always was. What's changed isn't the complexity. It's our tolerance for acknowledging it.

## The Seductive Appeal of Simplification

Simplification feels responsible. It suggests progress. It reassures boards, regulators, and budget committees that complexity can be tamed with the right framework, the right platform, or the right policy.

And sometimes simplification helps. Standardization saves lives. Protocols reduce variation where variation is dangerous. Checklists matter.

But simplification becomes a problem when it confuses clarity with correctness. Healthcare isn't a single problem to be solved. It's a constantly shifting interaction between biology, psychology, economics, culture, technology, and time. Every attempt to flatten that reality eventually runs into something stubbornly human.

## The Body Is Not a Machine (No Matter How Much We Wish It Were)

We like mechanical metaphors. They make the body feel understandable. Inputs, outputs, diagnostics, fixes.

The trouble is that bodies adapt. They compensate. They respond differently to the same intervention depending on context, stress, environment, and history. Add a human mind—beliefs, fear, motivation, trust—and suddenly the system becomes even less predictable.

Yet we keep designing healthcare systems as if predictability is the default and variation is the exception. Then we label patients “non-compliant” when reality doesn't cooperate. That word—non-compliant—has probably done more quiet damage than we realize.

## Clinicians Are Adapting, Not Resisting

One of the most persistent misunderstandings in healthcare system design is the assumption that clinicians resist change.

What they actually resist is friction that adds no value.

When workflows multiply clicks without improving care, clinicians adapt. When documentation grows while time with patients shrinks, they adapt. When systems prioritize billing logic over clinical logic, they adapt.

These adaptations are often framed as problems. In reality, they're survival strategies inside systems that underestimate complexity.

I've yet to meet a clinician who woke up hoping to undermine quality or safety. I've met many who quietly reengineered systems just to make it through the day.

## Technology Didn't Break Healthcare

It's fashionable to blame technology for healthcare's frustrations. Electronic records, interoperability challenges, alert fatigue—it's an easy target.

But technology didn't break healthcare. It exposed tensions that were already there.

Digital systems tend to formalize assumptions. When those assumptions don't match reality, the friction becomes visible. Paper systems hid a lot of improvisation. Software makes it explicit. The problem isn't that technology is rigid. It's that we often encode oversimplified models of care into tools that then scale faster than understanding.

## Metrics, Meaning, and Missed Signals

Healthcare is awash in metrics. Quality measures, performance indicators, compliance reports. Measurement is important. But when metrics become substitutes for meaning, trouble follows.

Some of the most important aspects of care—trust, understanding, continuity—are difficult to measure and easy to discount. Meanwhile, easily measured proxies take on outsized importance because they fit neatly into dashboards.

What gets lost is context.

A system optimized for measurement often becomes blind to nuance. And healthcare runs on nuance.

## Complexity Is Not a Moral Failure

There's an unspoken belief that complexity signals incompetence. That if things were designed better, they would be simpler.

In healthcare, that belief does real harm.

Complexity isn't a moral failure. It's a feature of systems that deal with humans, uncertainty, and life itself. The mistake isn't acknowledging complexity—it's pretending it shouldn't exist.

The most resilient healthcare environments I've seen don't fight complexity. They manage it. They create space for judgment. They respect local adaptation. They support humans instead of demanding heroics.

### Why This Matters Beyond Healthcare

Healthcare isn't unique—it's just honest.

The same forces show up in finance, technology, and organizations of every kind. Wherever humans intersect with systems under pressure, reductionism eventually runs out of runway.

Which brings us to a closely related misconception—the belief that technology itself drives transformation.

It doesn't.

### Chapter 5: Technology Is Never the Point

This may sound strange coming from someone who has spent a good portion of his career around technology, but it needs to be said plainly:

Technology is almost never the point.

It's a tool. Sometimes a powerful one. Occasionally a transformative one. But on its own, technology doesn't change systems nearly as much as we like to pretend it does.

What it does do—very reliably—is amplify whatever already exists.

### Why Technology Gets Blamed (and Praised Too Much)

When a system struggles, technology becomes the easiest target. If things are slow, fragmented, or frustrating, we assume the software is bad. If performance improves after a new platform launches, we credit the tool.

Both reactions miss the deeper story.

Technology doesn't create culture. It reveals it.

It doesn't invent workflows. It formalizes them.

It doesn't fix incentives. It scales them.

When technology "fails," it's usually because it made visible assumptions that no longer hold—or never did.

### The Myth of Transformation by Installation

Few phrases have caused more quiet disappointment than digital transformation.

The idea suggests that installing the right system produces transformation as a byproduct. That once the platform is live, behavior will follow.

In practice, the opposite is true.

Behavior changes first—or the technology becomes an expensive mirror reflecting old habits in high resolution.

I've seen organizations deploy sophisticated systems only to recreate existing processes inside them, inefficiencies and all. The interface changes. The friction doesn't. Sometimes it gets worse.

When that happens, technology becomes the scapegoat for decisions that were never addressed upstream.

### Users Are Not the Problem

Another familiar refrain:

"The system would work if people used it correctly."

This is rarely true, and when it is, it's rarely useful.

If a system requires perfect behavior to succeed, it's not robust—it's fragile. Humans are variable by nature. Fatigue, stress, interruptions, and competing priorities aren't edge cases. They're the operating environment.

Good technology design assumes imperfection. It anticipates shortcuts. It tolerates deviation. It helps people recover gracefully when things don't go as planned.

When systems demand heroics, burnout is not a failure of resilience—it's a predictable outcome.

### Automation Without Understanding

Automation is seductive. It promises speed, consistency, and efficiency.

But automating a poorly understood process doesn't fix it. It calcifies it.

In healthcare, automation without context can overwhelm clinicians with alerts that add noise instead of clarity. In finance, it can accelerate risk before anyone notices it accumulating. In organizations, it can lock in policies that no longer make sense but are now harder to change.

The question isn't "Can this be automated?"

It's "Should it be—and under what conditions?"

### The Real Work Happens Before the Tool

The most successful technology implementations I've seen had surprisingly little to do with the technology itself.

They invested heavily in:

- Understanding real workflows
- Clarifying decision points
- Aligning incentives
- Listening to the people who would live with the system every day

Only then did they choose tools.

The failures reversed that order. They selected platforms first, then tried to force reality to conform. Reality is stubborn.



## Technology as a Multiplier

Think of technology as a multiplier. It makes strong systems stronger—and weak ones louder.  
If communication is poor, technology accelerates misunderstanding.  
If incentives are misaligned, technology scales the misalignment.  
If trust is low, technology feels controlling instead of enabling.

This isn't a criticism of technology. It's a reminder to place it correctly in the hierarchy of change.

## What Actually Drives Transformation

Transformation happens when:

- People understand why change matters
- Incentives reinforce desired behavior
- Systems support—not punish—adaptation
- Technology follows purpose, not the other way around

Tools matter. But they matter after clarity, not before.

Which brings us to a lesson that shows up everywhere, from startups to global institutions:  
What works beautifully at small scale often fails spectacularly at large scale.

## Chapter 6: Scale Changes Everything (Usually for the Worse)

Almost every idea looks good at small scale.

That's not a criticism—it's a fact of how systems behave. When groups are small, communication is direct, context is shared, and people compensate for gaps instinctively. Problems get solved in the moment, often without anyone realizing a "system" was involved at all.

Then success happens. Growth follows. Scale arrives.

And suddenly, the same idea that worked beautifully at ten people starts breaking at ten thousand.

## Why Pilots Lie

Pilots are useful. They're also deeply misleading.

At pilot scale, everything is abnormal:

- Extra attention
- Extra resources
- Extra tolerance for workarounds
- Highly motivated participants

None of those conditions scale.

When leaders extrapolate pilot results directly into enterprise-wide expectations, they're often disappointed—not because the idea was bad, but because the conditions that made it work disappeared.

Scale removes intimacy. It exposes assumptions. It punishes fragility.

### Communication Is the First Casualty

At small scale, information flows informally. People overhear conversations. Clarifications happen in real time. Intent travels along with instruction.

At scale, communication becomes mediated—by tools, processes, layers, and policies. Context thins out. Signals lag. Misunderstandings multiply.

Organizations respond by adding more communication, which often means more volume, not more clarity. Important messages drown alongside irrelevant ones. People stop reading, then get blamed for missing updates.

The system becomes noisier but less informed.

### Standardization vs. Judgment

Scale demands standardization. Without it, chaos follows.

But standardization has limits—especially when judgment matters.

In healthcare, overly rigid protocols can collide with individual patient needs. In finance, standardized models can miss localized risk. In organizations, uniform policies can ignore cultural and operational realities on the ground.

The tension isn't between structure and freedom. It's between rigidity and adaptability.

The systems that scale best standardize where it's safe and allow discretion where it's necessary.

Getting that balance wrong is one of the fastest ways to degrade performance.

### Complexity Grows Faster Than Control

Double the size of a system and you more than double the interactions within it. Leadership bandwidth, however, grows linearly at best.

This creates a familiar pattern:

- Leaders feel less informed
- Controls increase
- Decision-making slows
- Local initiative erodes
- Workarounds expand

The system doesn't collapse. It stiffens.

Over time, that stiffness becomes the defining characteristic—and innovation quietly exits.

### Why Bigger Systems Blame People

When scaled systems struggle, the explanation often points downward:

- Training wasn't sufficient
- Compliance wasn't enforced
- Adoption wasn't strong enough

These explanations are comforting because they preserve the idea that the system itself is sound. In reality, people are usually doing the best they can inside constraints they didn't create.

Scale magnifies design flaws that were invisible when human compensation filled the gaps. Calling that "user error" misses the lesson entirely.

### Designing for Scale Without Losing Humanity

The most resilient large systems I've encountered share a few traits:

- Clear principles, not endless rules
- Local authority paired with accountability
- Feedback that travels upward quickly
- Space for learning, not just enforcement

They accept that perfection doesn't scale—but adaptability can.

Which brings us to the heart of the matter: systems don't fail because humans are flawed. They fail because we design as if humans aren't.

### Chapter 7: Designing for Humans, Not Heroes

One of the quiet assumptions baked into many systems is that people will rise to the occasion.

That when things get hard, they'll work longer hours, pay closer attention, remember more details, and make fewer mistakes—all while navigating increasing complexity. In other words, we design systems that require heroes.

Heroes do exist. I've worked with many of them. They're dedicated, skilled, and deeply committed. They also burn out, leave, or eventually make a mistake—not because they stopped caring, but because heroism is not a sustainable operating model.

### The Cost of Relying on Heroics

Systems that depend on exceptional effort eventually exhaust the people holding them together. In healthcare, this shows up as clinicians compensating for workflow gaps, staffing shortages, and administrative overload.

In finance, it appears as individuals quietly managing risk that the system doesn't see. In organizations, it's the "go-to" people who solve problems no one officially owns. At first, this looks like strength. Over time, it becomes fragility.

When heroes leave, systems collapse—not because they were irreplaceable, but because the system was never designed to function without them.

### Humans Are the Constant, Not the Variable

We often talk about "human error" as if humans are the unpredictable element in otherwise stable systems.

In reality, humans are the most consistent part of the equation.

People get tired.

They multitask.

They respond emotionally.

They adapt to constraints.

None of this is surprising. What's surprising is how often systems are designed as if none of it will happen.

The most effective systems I've seen don't try to eliminate human behavior. They build around it.  
Designing for Reality

Designing for humans means accepting a few uncomfortable truths:

- People will take shortcuts
- People will misunderstand instructions
- People will prioritize survival over optimization
- People will adapt systems to fit their day, not the other way around

Good design doesn't punish this. It channels it.

It reduces cognitive load.

It makes the right action obvious.

It makes recovery from mistakes easy and safe.

Most importantly, it treats feedback as information, not insubordination.

### Psychological Safety Is a System Feature

One of the most overlooked design elements in any system is psychological safety.

When people feel safe speaking up, systems learn. When they don't, systems drift.

Errors get hidden. Near-misses go unreported. Small problems grow quietly until they're large enough to demand attention—usually at great cost.

Designing for humans means designing for honesty. And honesty requires safety.

### From Blame to Learning

Blame feels decisive. Learning feels slow.

But blame shuts down signal. Learning amplifies it.

Systems that improve over time create mechanisms to surface friction early and respond without punishment. They assume mistakes will happen and focus on reducing their impact instead of pretending they won't occur.

This shift—from blame to learning—isn't cultural fluff. It's operational strategy.

## What This Looks Like in Practice

Human-centered systems:

- Expect variation
- Reward transparency
- Simplify wherever possible
- Support judgment instead of replacing it
- Treat humans as partners, not liabilities

They don't demand perfection. They design for resilience.

And that brings us to the final question—not how to fix systems, but how to think about them differently in the first place.

## Chapter 8: A Better Way to Think About Systems

By this point, you may be wondering if there's a unifying principle behind all of this.

There is—but it's not a framework, a checklist, or a methodology.

It's a mindset.

## From Control to Curiosity

The most important shift I've seen over 47 years is moving from control to curiosity.

Instead of asking:

- "Why didn't people follow the process?"

Ask:

- "What made this the easiest option?"

Instead of:

- "How do we enforce compliance?"

Ask:

- "What signal is the system giving us?"

Curiosity keeps systems alive. Control eventually suffocates them.

## Seeing Systems as Conversations

Healthy systems behave less like machines and more like conversations.

They listen.

They respond.

They adjust.

When feedback flows freely, systems self-correct. When feedback is filtered, delayed, or punished, they drift.

This applies to technology, organizations, healthcare, and leadership itself.

## Principles Over Prescriptions

Best practices promise certainty. Principles offer guidance.



Principles scale because they adapt. They provide direction without pretending to know every answer in advance. They allow local judgment while maintaining coherence.

Systems built on principles are harder to measure—but far easier to sustain.

### Why This Matters Now

The pace of change isn't slowing. Complexity isn't receding. Systems are becoming more interconnected, not less.

In that environment, rigid thinking becomes a liability.

The leaders, organizations, and systems that thrive won't be the ones with the most rules. They'll be the ones that learn the fastest.

### An Invitation, Not a Conclusion

This isn't the end of an argument. It's an invitation to see systems differently.

To notice incentives before blaming behavior.

To respect complexity instead of denying it.

To design for humans as they are, not as we wish they'd be.

If this way of thinking resonates, it's because you've probably seen these patterns yourself. You may not have named them—but you've felt them.

That recognition is where better systems begin.

### A Final Note

If you're responsible for building, leading, or improving systems—and most of us are, whether we admit it or not—you don't need perfect answers.

You need better questions.

And the willingness to listen to what the system is already telling you.

### About the Author

Doug Sparks has spent 47 years working at the intersection of complex systems, including eight years on Wall Street and nearly four decades across healthcare, technology, and organizational leadership. Rather than approaching problems from theory alone, Doug's work focuses on how systems actually behave once real people, incentives, and constraints are involved. His perspective blends financial rigor, clinical reality, and systems thinking—without the jargon or the preaching.

Doug works with organizations that are tired of surface-level fixes and ready to address the deeper patterns shaping performance, outcomes, and resilience. His speaking and consulting focus on designing systems that work with human behavior instead of fighting it.

If this book resonated, it's because you've likely seen these patterns yourself. The work now is learning how to respond to them more intelligently.

Every generation believes it has finally figured out how systems should work. Then reality politely disagrees.

If systems behaved the way leaders hoped they would, incentives wouldn't matter. Unfortunately, they do.

The moment you treat a network like a machine, it starts behaving like something alive—and annoyed.

- Healthcare doesn't resist simplification. It survives it.
- Technology rarely causes problems. It just exposes the ones we were already ignoring.
- Scaling a system is easy—until it works exactly as designed.
- If your system depends on heroes, it's already in trouble.

The hardest part of fixing systems isn't effort. It's learning how to see them differently.