Why We Still Suck At Resilience

Organizational Dynamics

Adrian Hornsby

Why We Still Suck At Resilience

Organizational Dynamics

Adrian Hornsby

This book is available at https://leanpub.com/whywestillsuckatresilience

This version was published on 2026-02-13

Leanpub

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Preface

The cost of failure keeps rising. Industry survey data suggests that over 90% of mid-size and large enterprises face costs exceeding $300,000 per hour of downtime, with over 40% reporting costs between $1 million and $5 million per hour.[1] As more business runs on software, as customer expectations for reliability rise, as systems become more interconnected and failures cascade further, the price of getting resilience wrong compounds.

Organizations invest accordingly. Some implement chaos engineering programs, others focus on load testing or incident analysis, many run operational readiness reviews or conduct GameDays. They follow frameworks, adopt best practices, hire experts. The specific practices vary, but the commitment is real.

Yet when production fails in ways that matter, organizations often discover their practices didn't prepare them. The chaos experiments didn't anticipate the cascading failure that just happened, the load testing didn't reveal the capacity cliff they just hit, the incident analyses didn't surface the recurring pattern, and the GameDays didn't build the coordination capability the response required. The practices operated, the boxes got checked, but the learning that would build resilience didn't accumulate.

If you've watched a well-funded resilience program produce impressive artifacts while the same types of failures keep recurring, this book is for you. If you've sat through incident reviews that carefully avoid the actual problems, if you've seen chaos engineering become a compliance exercise, if you've wondered why all the investment isn't building the capability it should, you're not alone. These patterns are systematic, not accidental.

This book examines why resilience practices often fail to produce resilience, and what enables them to actually work. It explores the fundamental gap that resilience work must manage, the organizational forces that widen it, and the conditions that enable learning despite those forces.

The core argument is simple. Resilience is an ongoing practice of learning and navigation, not an engineering problem to solve through sufficient invest-

[1]Information Technology Intelligence Consulting (ITIC), "2024 Hourly Cost of Downtime Report," 2024, https://itic-corp.com/itic-2024-hourly-cost-of-downtime-report/.

ment and comprehensive planning. The gap between how we think our systems work and how they actually work will never close completely. The question is whether we're discovering and managing that gap continuously, or waiting for production failures to reveal it.

What This Book Offers

This book assumes you already have practices operating or are considering implementing them. Many excellent resources exist for step-by-step implementation guides. What this book offers is a way to understand why those practices aren't producing what they should, and what conditions enable them to actually work.

The approach is exploratory rather than prescriptive. I describe patterns I've observed across more than a decade of work, frameworks I've found useful for making sense of those patterns, and conditions I've seen enable learning. These ideas aren't new. They've been developed over decades in aviation, nuclear, and healthcare. What I'm offering is their application to software organizations, where the same patterns appear in different contexts. I'm offering perspective on dynamics that I believe are systematic and important.

The book is structured to build understanding progressively. Part 1 establishes the foundation: how complex systems fail, why learning is the only viable response, why practices designed for learning often fail to produce it, and what bedrock conditions enable practices to work. Part 2 examines five specific practices, showing how each reveals different dimensions of the gap and how each drifts toward theater under organizational pressures. Part 3 explores how to navigate these dynamics: making drift visible, designing for adaptability, treating resilience as ongoing practice rather than achievable destination. The final chapter examines how these dynamics manifest as AI becomes embedded in both the systems we build and the tools we use to build them.

Who This Book Is For

If you're involved in resilience work in any capacity, this book is for you. Engineers implementing practices, leaders sponsoring programs, consultants advising organizations, architects designing systems, SREs operating them.

The perspective applies across roles because the dynamics are organizational and cultural, not specific to particular job functions.

You might be frustrated that resilience practices feel like checkbox exercises rather than learning, confused why significant investment hasn't built the adaptive capacity you expected, or concerned that your organization talks about resilience but doesn't actually behave in ways that build it. You might simply sense that something is wrong beneath the surface of impressive-sounding programs but lack language to articulate what.

The book provides concepts and vocabulary for understanding what's happening and discussing it precisely. The tensions described here are real and irreducible. The goal is making them visible and navigable.

How to Read This Book

The chapters build on each other. The foundational concepts in Part 1 are referenced throughout Parts 2 and 3. The specific practices in Part 2 assume understanding of why practices fail from Part 1. The navigation guidance in Part 3 assumes both the foundation and the practice-specific observations.

That said, if you're dealing with a specific immediate challenge, the Part 2 chapters can stand relatively independently. If chaos engineering has become validation theater, Chapter 7 addresses that directly. If incident analysis stays shallow, Chapter 9 explores why and what enables depth. If you need to understand a particular practice urgently, start there, but expect references to concepts from Part 1 that might require backtracking.

The book uses a small set of core concepts repeatedly: Work-as-Imagined versus Work-as-Done, organizational tensions (five primary, five secondary), bedrock conditions (psychological safety, appropriate incentives, leadership support), learning cycles. These concepts appear across chapters because they're fundamental to understanding how resilience work actually functions. The repetition is deliberate, because understanding develops through seeing how the same concepts manifest in different contexts.

A Note on Examples

The examples and stories throughout this book come from more than a decade of building and operating software, running chaos experiments and load tests, facilitating incident analyses and GameDays, conducting operational readiness reviews, and observing how organizations actually practice resilience. All examples have been anonymized and often composited from multiple organizations to protect confidentiality. If you recognize patterns described here, it's because these dynamics emerge consistently across organizations. They're features of how complex sociotechnical systems and human organizations work, not fingerprints of specific companies.

A Note on Interpretation

My sample is biased. The organizations I've worked with are not representative of all software organizations. Most reached out because they were already investing in resilience, often operating in cloud environments at significant scale. The patterns I describe may manifest differently in smaller organizations, in different industries, or in contexts I haven't encountered.

Interpretation is always present when making sense of organizational patterns. Researchers with good intentions can look at the same data and derive completely different stories from it. The world is messy. I've tried to ground my interpretations in established research and extensive observation, but they remain interpretations. Where I've seen consistent patterns across many organizations, I have more confidence. Where I'm extrapolating from limited observation, I've tried to say so. The frameworks in this book are lenses for making sense of dynamics, not laws of organizational physics.

Acknowledgments

The intellectual foundations come from resilience engineering research. Erik Hollnagel developed the Work-as-Imagined versus Work-as-Done framework that anchors this book. His work with David D. Woods and others, particularly "Resilience Engineering in Practice: A Guidebook," profoundly shaped my understanding of how resilience actually works in operational settings. Richard Cook's "How Complex Systems Fail" provides foundational understanding of system behavior. Sidney Dekker's "Drift into Failure" describes how organizations unconsciously move toward brittleness through accumulated small decisions, a theme central to this book. Nancy Leveson's systems thinking approach to safety influences how I understand complex sociotechnical systems.

Chris Argyris and Donald Schön's work on organizational learning provides the framework for understanding single-loop, double-loop, and deutero-learning that appears in Chapter 2.

But academic resilience engineering would have remained largely inaccessible to software organizations without practitioners who translated these concepts into operational practice. John Allspaw deserves particular recognition for putting learning at the center of resilience work in software engineering. Through his work at Etsy and later at Adaptive Capacity Labs, and through talks, writings, and teaching, Allspaw introduced resilience engineering concepts to software organizations, emphasizing learning from incidents over purely preventive measures. His pioneering work on blameless postmortems and learning-oriented incident analysis showed that incidents are unplanned investments in learning about the gap between expected and actual system behavior. He demonstrated how to move from "what broke and who broke it" to "what can we learn about how our systems actually behave versus how we think they behave," drawing from fields like cognitive systems engineering. That reframing, from blame and prevention to learning and adaptation, fundamentally changed how software organizations approach resilience. The learning-centered approach throughout this book builds directly on the foundation Allspaw established.

Jesse Robbins pioneered GameDays at Amazon in the early 2000s, creating structured practice for incident response and coordination. Casey Rosenthal and Nora Jones advanced chaos engineering theory and practice, showing how deliberate experimentation reveals gaps between assumptions and reality.

Uwe Friedrichsen's writings on resilience engineering and complex systems, Fred Hebert's explorations of resilience and complex systems in software, and Lorin Hochstein's work on software incidents and complexity have deepened my understanding of how these concepts apply in software contexts. Many others whose work I've learned from deserve acknowledgment.

This book builds on that foundation. What I've tried to contribute is understanding of the organizational dynamics that prevent practices from functioning as intended and what enables them to work despite those dynamics. The observations are mine. The interpretations are mine. The frameworks for making sense of patterns are mine. But the work stands on shoulders of people who established the field.

What I Hope You Take Away

The organizational forces that undermine resilience are real, the tensions are inherent, and the drift toward brittleness disguised as safety is normal rather than exceptional. But these forces can be made visible, the tensions can be navigated consciously, the drift can be recognized and actively managed.

Resilience as ongoing practice rather than achievable destination; learning and navigation rather than control and elimination. That reframing is what enables resilience work that actually builds capability rather than creating expensive theater.

Once you can name what's happening, you can see it. Once you can see it, you can talk about it. Once you can talk about it, you can make different choices.

I hope this book helps you with the vocabulary. The choices are yours.

Thank You

Thank you to Anu, who gave me the time, the space, and the encouragement to actually finish this thing. And to Oreo (my dog) for making sure I got my daily walks in every kind of Finnish weather: rain, snow, minus thirty, sunny, it didn't matter. He took me for a walk to make sure I had time to think.

To my friends around the world, for being exactly who they are.

Special thanks to those who dared read early versions without judging: Jason, Ricardo, Paula, Benjamin, Russ, and Markus. Your comments shaped this book more than you know.

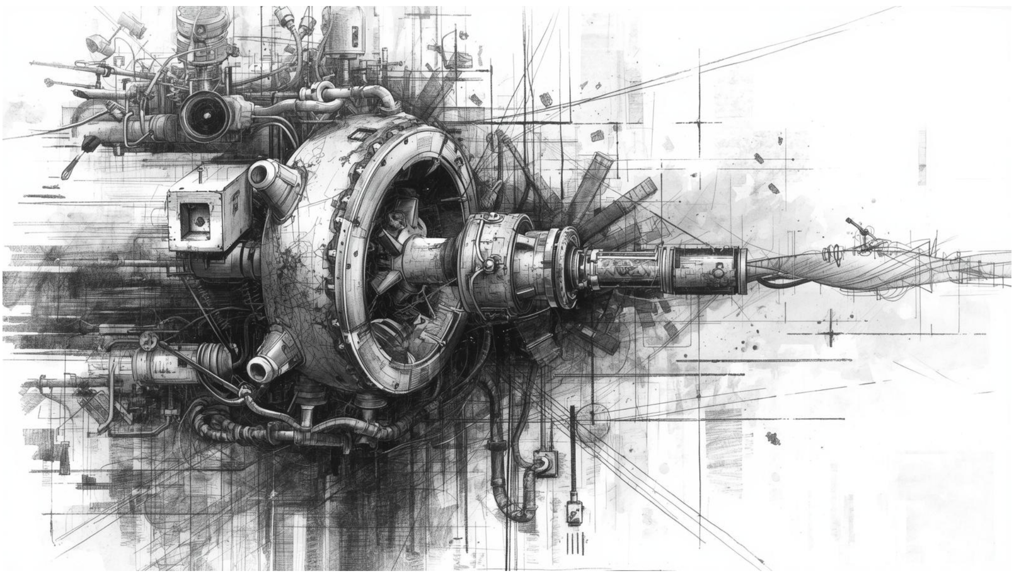
And very special thanks to John Allspaw for his critical feedback, support, and time. Everyone wants John to review their blog, book, or postmortem. I keep waiting for an email saying he meant to reply to a different Adrian.

Finally, thank you to Claude (yes, the AI), for the editing help and for flagging every section that made the book longer rather than better.

Adrian Hornsby *Helsinki, Finland*, 2026

Part 1: The Problem

Chapter 1: How Complex Systems Fail



It works. No one fully understands how.

In 1998, Richard Cook published a short paper that would become foundational to understanding how complex systems actually behave. "How Complex Systems Fail" offers eighteen observations about the nature of failure in systems people depend on[1]. These observations emerged from Cook's work studying failures in medical systems, but they describe something universal about complexity itself.

The first observation is deceptively simple: "Complex systems are intrinsically hazardous systems." Not sometimes hazardous or hazardous only when poorly designed, but intrinsically hazardous. The systems we build to handle important work contain multiple points of potential failure. They always have. They always will. No amount of careful engineering eliminates this

[1] Richard I. Cook, "How Complex Systems Fail," Cognitive Technologies Laboratory, University of Chicago, 1998, https://how.complexsystems.fail/.

fundamental reality. You can reduce specific risks. You can add redundancy. You can improve monitoring and automate recovery. The underlying hazard remains because it emerges from the complexity itself, not from implementation choices.

Cook's fifth observation explains why these systems continue functioning despite being intrinsically hazardous: "Complex systems run in degraded mode." The system you think you have, the one described in architecture diagrams and capacity plans, exists only briefly if at all. Real systems operate continuously in states that deviate from design assumptions. A database runs with one replica offline. A service operates at capacity limits not anticipated when it was built. Monitoring has gaps where recent changes aren't fully instrumented. Configuration drifts across servers. Dependencies get updated without full regression testing. The system runs anyway, adapted by the people operating it to handle conditions that weren't part of the original design.

This degraded mode operation isn't a failure state to be corrected. It's normal. Cook's observation is that complex systems contain changing mixtures of failures latent within them. At any given moment, some components are operating outside their design parameters, some assumptions about the environment no longer hold, some protective mechanisms aren't quite working as intended. The system continues functioning because of ongoing adaptations by the people running it, not because all the components are operating within specification.

Werner Vogels, Amazon's CTO, captured this reality in a way that resonates with anyone operating systems at scale: "Everything fails, all the time."[2] Not occasionally when things go wrong, but continuously. At sufficient scale, every component will fail, every process will encounter unexpected conditions, and every assumption will eventually be violated. You cannot engineer your way out of this. No amount of redundancy, testing, or monitoring prevents failure. Failure is the normal operating condition, not an exception to be eliminated.

This might sound pessimistic, but it's actually liberating once you accept it. If failure is inevitable, the question isn't "how do we prevent all failures?" The question becomes "how do we build systems and organizations that can handle failure when it happens?" The shift from prevention to capability changes everything about how you approach resilience.

---

[2]Werner Vogels, "10 Lessons from 10 Years of Amazon Web Services," All Things Distributed (blog), March 2016, https://www.allthingsdistributed.com/2016/03/10-lessons-from-10-years-of-aws.html.

There's a deeper liberation here too. Once you accept that mistakes are part of the process, you stop being afraid of them. You learn from each specific failure, address what it reveals, and then move forward confidently to make new, different mistakes. New mistakes mean you're pushing into unexplored territory, building new capabilities, taking the risks that create value. The goal of learning from *this* failure is not to prevent all future failures. It's to free you to encounter new ones with confidence.

The Mechanism: Why Everything Fails

Understanding that complex systems are intrinsically hazardous and run in degraded mode raises an obvious question: what is the mechanism that makes failure inevitable? Why can't we just design systems that work as intended?

The answer lies in a gap that exists in every complex system, what resilience engineering calls the distinction between Work-as-Imagined (WAI) and Work-as-Done (WAD).[3] On one side of this gap is WAI. This is how we think systems operate. It's what's captured in architecture documents and design reviews. It's the system described in runbooks and operational procedures. It's what we teach new engineers during onboarding. It's the capacity model we used for planning. It's the failure modes we anticipated and designed protections for.

Work-as-Imagined is clean, logical, and rational. It's documentable. It's measurable. It's defensible to auditors and leadership. When we design systems, we work in the space of the imagined. We make assumptions about how components will interact, what load patterns will look like, how failures will manifest, what operators will do when things break. These assumptions feel reasonable at design time. They're based on the best information available. They account for known constraints and established patterns.

On the other side of the gap is WAD. This is how systems actually behave when they encounter real conditions with real constraints operated by real people under real pressures. It's the system on launch day, when real traffic immediately reveals interactions between components that behaved differently

[3]The distinction between prescribed work and actual work originates in French ergonomics from the 1950s, particularly Ombredane and Faverge's *L'analyse du travail* (1955), which contrasted *tâche prescrite* (prescribed task) with *activité réelle* (real activity). Erik Hollnagel popularized the English terms "Work-as-Imagined" and "Work-as-Done" within resilience engineering and Safety-II. See Erik Hollnagel, "Why is Work-as-Imagined Different from Work-as-Done?" in *Resilient Health Care, Volume 2: The Resilience of Everyday Clinical Work*, ed. Erik Hollnagel, Jeffrey Braithwaite, and Robert L. Wears (Farnham, UK: Ashgate, 2015), 249-264.

in testing. It's the system that has evolved through six months of production traffic with usage patterns that don't quite match the load testing assumptions. It's the configuration that has drifted because the documented change control process takes three days but the customer-impacting bug needed a fix in three hours. It's the monitoring that doesn't quite cover the interaction between the new feature and the legacy payment system. It's the runbook that gets bypassed because the person responding to the incident knows a faster path that isn't documented yet.

Work-as-Done includes all the adaptations, shortcuts, workarounds, and informal knowledge that accumulate as systems encounter reality. An engineer notices that deployements to the Asia Pacific region need an extra verification step not in the official process because of a subtle interaction with a local caching layer. A team learns that the database failover procedure in the runbook doesn't account for a recent schema change, so they've developed an informal pre-flight check they run first. An operator knows that when a particular alert fires, you should check three other things before following the documented response, because the alert sometimes triggers on conditions that resolve themselves.

This is tacit expertise. It's the knowledge that keeps systems running despite, and sometimes in spite of, documented procedures. It's hard to transfer to new team members. It's nearly impossible to audit. It often becomes visible only when the person who holds it leaves the team or isn't available during an incident.

The gap between WAI and WAD is not a sign that something has gone wrong. The gap is inevitable. It emerges naturally from the nature of complex systems.

The gap doesn't start small and grow. It exists from the beginning, because complex systems produce emergent behavior, properties that arise from the interaction of components, people, and context that cannot be predicted from understanding the parts individually. The system's actual behavior isn't fully designed; it emerges. You can understand every component and still not understand the system, because the system is the interactions, not the components.

When organizations systematically map how their operations actually work, even domain experts are surprised to discover that what they perceived as purposively designed and stable was in fact continuously produced through social interaction, far less centrally coordinated and far more emergent than anyone imagined.[4] The architecture diagram was already wrong before anyone used the system.

This isn't just a property of the system. It's also a property of the people trying to understand it. Human cognition has finite capacity. We cannot hold the full complexity of a system in our heads, so we simplify. We build mental models that compress thousands of interactions into manageable abstractions. Architecture diagrams, component boundaries, documented procedures. Herbert Simon called this bounded rationality, the recognition that humans face problems too complex to solve optimally, so they simplify and satisfice, finding representations that are good enough rather than complete.[5] Every mental model of a complex system is a lossy compression. The simplification is not a failure of discipline. It is a structural necessity.

The gap is therefore permanent. Systems produce emergent behavior that can't be fully predicted. Humans produce simplified models that can't fully capture it. Neither changes with better tools, better documentation, or better engineering. What changes is how adequate the simplified models are for the task at hand. Models can be improved, diversified, and stress-tested. The practices examined in this book do exactly that. They create structured ways to discover where your simplified model is wrong before production discovers it for you. The gap remains, but its consequences become more manageable.

And then it grows. Systems evolve continuously. New features change failure modes, dependency updates alter interaction patterns, scaling events reveal capacity assumptions that weren't quite right, and incident responses create new knowledge about system behavior under unanticipated conditions. The system in production six months after launch is fundamentally different from the system that was designed, even if the code is largely the same. The evolution happens through small changes that individually seem insignificant but cumulatively transform how the system actually works.

Real operating conditions never quite match design assumptions. Load

[4] Ruthanne Huising, "Moving off the Map: How Knowledge of Organizational Operations Empowers and Alienates," *Organization Science* 30, no. 5 (2019): 1054-1075. Huising studied employees assigned to map their organization's actual operations. Participants discovered that what they had perceived as purposively designed and relatively stable was continuously produced through social interaction. The finding extends beyond organizational processes to any complex sociotechnical system: the imagined version was never an accurate representation of the actual system.

[5] Herbert A. Simon, "A Behavioral Model of Rational Choice," *The Quarterly Journal of Economics* 69, no. 1 (1955): 99-118. Simon demonstrated that human decision-makers facing complex problems cannot process all available information, so they adopt simplified models of reality and seek satisfactory rather than optimal solutions.

patterns shift, user behavior changes, dependencies evolve, and network characteristics vary in ways that weren't anticipated at design time. For instance, the business launches in a new market with different usage patterns, a feature becomes unexpectedly popular and creates load on a service that was designed for different scale, or external events trigger cascading failures that weren't considered during planning. You cannot anticipate all of this during design. The gap between assumed conditions and actual conditions grows as the system encounters reality.

People adapt continuously to bridge these gaps. Engineers develop workarounds when processes are too slow. Operators learn which alerts need immediate response and which can wait. Teams discover that the documented procedure for a particular failure mode doesn't quite work, so they develop a modified approach. Knowledge accumulates about how to actually run the system effectively, distinct from how the system was designed to be run. This adaptation is necessary. It's what allows systems to continue functioning when reality diverges from assumptions.[6] But it also means that how work actually happens increasingly diverges from how work was imagined to happen.

As the system evolves, as conditions change, as people develop new adaptations, the distance between WAI and WAD increases. Unless there is deliberate effort to understand how work actually happens and update the imagined version accordingly, the gap widens. Documentation becomes outdated as runbooks describe procedures that no longer match reality, architecture diagrams show systems that have since been modified, and capacity models reflect assumptions that no longer hold.

This is why everything fails. The gap between how we think systems work and how they actually work is where fragility lives. We design protections for imagined failure modes. Real failures emerge from the gap, from interactions and conditions we didn't anticipate, from the mismatch between assumed behavior and actual behavior. We create monitoring based on imagined system states. Real problems manifest in ways the monitoring doesn't capture because they involve the parts of WAD that aren't reflected in WAI.

---

[6] Woods and Branlat call this pattern *compensation*: the system finds new ways to perform despite degradation. Compensation keeps systems functioning but masks the underlying problem. When compensatory capacity is exhausted, the result is *decompensation*, a sudden collapse that appears to come from nowhere because the compensation was hiding the growing disturbance. See David D. Woods and Matthieu Branlat, "Basic Patterns in How Adaptive Systems Fail," in *Resilience Engineering in Practice: A Guidebook*, ed. Erik Hollnagel et al. (Farnham, UK: Ashgate, 2011). For an application of this pattern to software systems, see Fred Hebert, "Decompensation and Cascading Failures," Resilience in Software Foundation (blog), January 22, 2026, https://resilienceinsoftware.org/news/11454232.

Cook's insight that complex systems run in degraded mode is describing this gap. The degraded mode is the space between how the system was designed to operate and how it actually operates. The changing mixtures of failures he describes are the mismatches that exist at any moment between imagined behavior and actual behavior. Vogels' observation that everything fails all the time is describing the consequence. At scale, with sufficient complexity, the gap is large enough and dynamic enough that failures emerge continuously. They emerge not because of poor engineering but because the gap is fundamental to complex systems.

This might seem like a counsel of despair. If the gap is inevitable and growing, if failure emerges from this gap, what can be done?

Learning Is The Only Option

Eliminating the gap is impossible, and so is preventing all failures. The only productive response is learning: discovering where WAI and WAD diverge, understanding how people are already adapting to bridge mismatches, seeing both what's working and what's broken. The gap is permanent. What matters is whether you understand it, learn from it continuously, and develop the capacity to navigate it skillfully.

This is what resilience actually requires: developing organizational capacity to learn about and navigate the gap between how you think your systems work and how they actually work. Resilience isn't something you build from nothing. It already exists in every functioning system, in the workarounds engineers develop, the undocumented procedures operators follow, the tacit knowledge that keeps production running. The question is whether you're learning from that existing resilience, amplifying what works, and understanding what prevents effective adaptation.

A note on language. Throughout this book I describe practices that "expose" or "discover" the gap rather than "reduce" or "close" it. This framing is deliberate but requires nuance. Specific instances of the gap can absolutely be addressed. When chaos engineering reveals that a service doesn't handle dependency failure gracefully, you can fix that. When incident analysis shows that runbook procedures don't match operational reality, you can update the runbook. The practices expose gaps so you can address them before they cause problems.

What cannot be eliminated is the gap itself as a structural phenomenon. Systems evolve faster than documentation tracks. People adapt faster than procedures capture. Complete understanding of complex system behavior is unachievable. Address one gap and new ones emerge as contexts change. The gap is where both vulnerabilities and resilience live. Resilience comes from continuously learning about it.

What We Mean by Resilience

Before going further, we need clarity on what resilience actually means. The word gets used in many different ways, and those different meanings lead to very different approaches to improving it.

David Woods' foundational paper "Four Concepts for Resilience" provides the clearest framework I've found for distinguishing between these different meanings[7]. Woods identifies four distinct concepts that people refer to when they use the label "resilience."

Resilience as rebound focuses on recovery from disruptions and return to normal operation. This is reactive recovery. In software systems, rebound includes rolling back a bad deployment, restoring lost data from backups, rebooting a failed server, truncating log files that filled the disk and brought down a service. The emphasis is on getting back to a known good state after something has gone wrong. Rebound assumes that such a state exists and is still viable, that returning to it is both possible and desirable.

Resilience as robustness emphasizes the ability to absorb well-understood disturbances without human intervention. Robustness is what you get from redundancy: multiple replicas of a service, multi-region deployments, automated failover between database instances. It's what you get from protective patterns: circuit breakers that prevent cascading failures, rate limiting that protects against overload, timeouts that prevent indefinite waits, retry logic with exponential backoff. It's what you get from adaptive scaling that adds capacity automatically when demand increases.

Robustness works well for anticipated failure modes where you can model the disturbances in advance. You know that individual servers will fail, so you design for that. You know that networks will have transient issues, so you add

[7] David D. Woods, "Four Concepts for Resilience and the Implications for the Future of Resilience Engineering," *Reliability Engineering & System Safety* 141 (2015): 5-9.

retries. You know that dependencies will occasionally be slow or unavailable, so you implement timeouts and circuit breakers. You can build robust systems for these well-understood problems.

Resilience as graceful extensibility is about how systems stretch to handle surprises that fall outside their designed boundaries. This is what happens when something unexpected occurs and people respond by adjusting their approach, coordinating in novel ways, finding creative solutions under pressure. Graceful extensibility is your incident response capability. It's what allows teams to adapt in real time to conditions they've never seen before, to handle the failures you didn't anticipate and couldn't have robustness for.

When a database starts exhibiting bizarre performance characteristics no one has seen before, graceful extensibility is what allows engineers to investigate, form hypotheses, test them under pressure, and devise solutions even though this specific problem isn't in any runbook. When a new feature interacts with a legacy system in ways that weren't anticipated and starts causing cascading failures, graceful extensibility is what allows teams to understand the interaction, coordinate across organizational boundaries, and stabilize the system even though the failure mode wasn't designed for.

Resilience as sustained adaptability is the ability to maintain and evolve adaptive capacity over time as conditions change. All systems face continuous evolution. New features create new failure modes. Components that work well attract demand beyond their original design. Dependencies shift. Usage patterns change. Markets evolve. Sustained adaptability means building organizational capabilities that help you keep learning and adjusting as these changes occur.

This includes practices like learning from incidents in ways that deepen understanding across teams, not just fixing the immediate problem. It includes operational reviews that detect shifting patterns before they become crises. It includes distributing knowledge rather than concentrating it in individuals who become single points of failure. It includes creating organizational systems that continuously discover where WAI and WAD are diverging, and that update assumptions based on actual behavior.

Woods argues, and I agree, that while rebound and robustness are important, they have limited value for complex adaptive systems operating at scale. Rebound assumes you can return to a previous state, but modern systems evolve so continuously that the "previous state" may no longer be viable or even achievable. Robustness only works for well-understood failures. Most interesting problems are surprises, the failures you didn't anticipate and couldn't design protections for.

The real progress in resilience comes from graceful extensibility and sustained adaptability. These capabilities focus on handling the unexpected and building long-term adaptive capacity. They acknowledge that you cannot prevent all failures, cannot anticipate all conditions, cannot design for all interactions. What you can do is build the organizational capability to learn when reality diverges from assumptions, to adapt when surprised, and to maintain these capabilities as systems and conditions evolve.

The challenge is that most organizations focus heavily on rebound and robustness. They invest in automated recovery systems, comprehensive runbooks, tested disaster recovery procedures, redundant infrastructure, monitoring and alerting. These are valuable and necessary, but they're insufficient.

What's typically missing is the adaptive capacity to handle surprises: the ability to learn about the WAI-WAD gap before it produces customer-impacting failures, and the organizational systems that maintain learning capability over time as people change, systems evolve, and conditions shift.

The WAI-WAD gap is why rebound and robustness are insufficient. You design robustness for imagined failure modes. Real failures emerge from the gap, from the space where your assumptions don't match reality. You create runbooks for imagined recovery procedures. Real incidents require graceful extensibility because they involve conditions the runbooks don't cover.

What This Book Examines

This book focuses primarily on graceful extensibility and sustained adaptability. It examines five practices that serve as mechanisms for sustained adaptability, practices that build and maintain the capacity for graceful extensibility. In practical terms, we're exploring organizational systems for continuously developing and maintaining the adaptive capacity to handle whatever surprises emerge as your systems and organization evolve. That said, when discussing specific incidents or system behaviors, I may use the term "resilience" more broadly to include rebound and robustness as well, since all four capabilities matter in practice.

Managing the gap requires learning: discovering where WAI and WAD diverge, understanding how systems actually behave rather than how you think they behave, knowing how people respond to problems rather than what procedures say they should do, and detecting when assumptions no longer match reality, when adaptation has created new risks, when evolution has introduced new failure modes.

This learning cannot be a one-time effort because the gap is dynamic: systems evolve, conditions change, new features introduce new interactions, and dependencies shift. The learning must be continuous, systematic, and organizational. It must happen faster than the gap grows. It must inform how you design, operate, and evolve systems. It must become part of how the organization functions, not an occasional activity when something goes wrong.

This is what graceful extensibility and sustained adaptability actually mean in practice: building organizational capacity to continuously learn about and manage the WAI-WAD gap. The practices examined in this book are mechanisms for this learning. Each practice, when implemented well, discovers different dimensions of the gap.

Operational Readiness Reviews discover gaps between how teams think they're prepared and how prepared they actually are. They examine architecture, failure modes, monitoring, deployment procedures, disaster recovery, and organizational learning.

Load Testing discovers gaps in assumptions about capacity and performance. You have a capacity model that predicts how the system will perform under various load levels. Load testing reveals whether the model matches reality. Often it doesn't.

Chaos Engineering discovers gaps in assumptions about how systems behave under failure conditions. You think your service degrades gracefully when a dependency becomes unavailable. Chaos engineering tests that assumption. It reveals what actually happens when the dependency fails, which often differs from imagined behavior.

GameDays discover gaps in coordination and knowledge. Making these gaps visible when customer impact isn't at stake is far preferable to discovering them during a real incident.

Incident Analysis learns from gaps that reached production. An incident occurred because reality didn't match assumptions in some way. Every incident is an expensive lesson about the WAI-WAD gap.

Together, these practices enable systematic discovery of the gap rather than waiting for production failures to reveal it. They surface both what's broken and what's working. They build the organizational muscle for continuous learning about how your systems actually work, which is the foundation of sustained adaptability.

But as we'll see in the chapters ahead, having these practices in place doesn't guarantee that learning happens. Organizations can run all five practices while still failing to build adaptive capacity. The practices can become demonstrations of capability rather than learning mechanisms.

Understanding why that happens requires one more foundational concept: how organizations drift toward failure through accumulated reasonable decisions.

How Organizations Drift

Diane Vaughan's research on the Challenger disaster revealed a mechanism she called the normalization of deviance[8]. Small deviations from design specifications or established procedures get rationalized as acceptable. A component shows slightly more wear than expected, but nothing fails. The deviation becomes data suggesting the system tolerates more variation than originally specified. Further wear in subsequent operations gets evaluated against this new baseline rather than the original design standard. Over time, what would have been unacceptable deviation from the original specification becomes normal because the reference point has shifted.

This normalization happens through locally rational decisions: each small deviation makes sense given current pressures and recent experience. The system tolerated the last deviation, so this slightly larger one seems acceptable. Nothing failed last time, so the current practice appears validated. The organization's understanding of what's safe drifts along with its practices, so the growing gap remains invisible from inside.

The same mechanism operates in software systems. Say a deployment process skips a verification step because of time pressure. Nothing breaks, so the abbreviated process becomes acceptable for urgent deployments. Over months, the definition of "urgent" expands and the verification step gets

[8] Diane Vaughan, *The Challenger Launch Decision: Risky Technology, Culture, and Deviance at* NASA (Chicago: University of Chicago Press, 1996).

skipped more frequently. Eventually the abbreviated process becomes standard practice, and new team members never learn there was a verification step to begin with. The organization's baseline has shifted without anyone deciding to change it.

Success is the misleading signal that enables this drift. When deviations don't produce immediate failures, they feel validated. The team that skipped the verification step didn't experience problems, so the skip seems justified, and the organization that simplified its testing didn't have an outage, so the simplification appears safe. Each successful deviation reinforces the new baseline and makes further deviation feel acceptable.

This creates a cruel dynamic: the absence of failure becomes evidence that current practice is adequate, even as current practice drifts further from what would actually be safe under stress. The organization accumulates latent risk that remains hidden until conditions change or the system encounters stress it hasn't seen before.

Sidney Dekker studied this phenomenon across industries and described it as drift into failure[9]. Where Vaughan focused on how specific deviations become normalized, Dekker emphasized the systemic nature of the drift. Organizations don't fail because they ignore safety or make reckless decisions. They drift toward failure through accumulated small adaptations, each one a reasonable response to local pressures. Add one more approval step to prevent the last incident. Streamline this process to move faster. Cut that redundancy to reduce costs. Each decision makes sense locally. Collectively, over months and years, they move the organization toward brittleness while everyone involved believes they're making the system safer or more efficient.

This organizational drift is distinct from the WAI-WAD gap, though related. The gap is a structural mismatch between models and reality that exists even in well-run organizations. Drift is about how practices, procedures, and standards erode through accumulated reasonable decisions. Both contribute to fragility. Both grow invisibly. And both require deliberate effort to counteract.

Understanding organizational drift matters because it affects the resilience practices themselves. The practices examined in this book can drift just like any other organizational process. Chaos experiments become safer over time. Incident reviews become shallower. Load tests become less realistic. The

[9] Sidney Dekker, *Drift into Failure: From Hunting Broken Components to Understanding Complex Systems* (Farnham, UK: Ashgate Publishing, 2011).

very practices meant to discover gaps can normalize into ineffective routines. Recognizing this dynamic is the first step toward preventing it.

If the gap is inevitable, if drift is normal, if practices meant to build resilience can themselves drift toward theater, what actually works? The answer is learning. Not occasional learning when something goes wrong, but continuous, systematic learning that discovers the gap faster than it grows. That's where we turn next.

Key Concepts

- **Complex systems are intrinsically hazardous.** They contain multiple points of potential failure that cannot be eliminated through careful engineering. They always have, they always will.
- **The gap between WAI and WAD is inevitable.** Complex systems produce emergent behavior that can't be fully predicted. Human cognition requires simplified models that can't fully capture it. The gap exists from day one and widens as systems evolve, conditions change, and people adapt. Reality was already different from the documentation before anyone used the system.
- **Drift happens through normalization of deviance.** Small deviations get rationalized as acceptable. The baseline shifts. What was once unacceptable becomes normal because nothing failed. The gap grows invisibly.
- **Four concepts of resilience.** Rebound (recovery), robustness (absorbing known disturbances), graceful extensibility (handling surprises), and sustained adaptability (maintaining adaptive capacity over time). This book focuses on the latter two.
- **Learning is the only productive response.** You cannot eliminate the gap. You can learn about it, discover where assumptions diverge from reality, and update your understanding before production forces that discovery.

Actionable Checklist

- [ ] Identify where your mental models of system behavior come from. Design documents? Previous experience? Monitoring data? Ask whether these sources reflect current reality.
- [ ] Look for signs of normalization of deviance. What shortcuts have become standard practice? What deviations from documented procedures feel routine?
- [ ] Assess your resilience investments. Are you primarily focused on rebound and robustness (known failures) or on graceful extensibility and sustained adaptability (handling surprises)?
- [ ] Examine where tacit knowledge lives. Who knows how systems actually work versus how they were designed? What happens when those people are unavailable?

Chapter 2: What Learning Actually Means

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Organizational Learning and Learning Organizations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Three Types of Learning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Learning Cycle

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Five Practices and the Learning Cycle

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Visibility Prerequisite

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Vocabulary Barrier

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 3: Why Practices Fail to Build Learning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Definition Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Five Primary Tensions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Cascade Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

How This Manifests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 4: The Bedrock

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Psychological Safety

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Reciprocity, Not Just Provision

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Appropriate Incentives

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Leadership Support

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Ownership, Not Just Sponsorship

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Maintaining the Bedrock Under Pressure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Why the Bedrock Matters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Part 2: The Practices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 5: Operational Readiness Reviews

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What Readiness Reviews Actually Reveal

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Readiness Reviews as Theater

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Why Checklists Fail Silently

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Designing for Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Who Should Conduct Reviews

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What Readiness Reviews Require from the Bedrock

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Continuous Operational Review

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 6: Load Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What Load Tests Actually Reveal

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Why Capacity Assumptions Are Systematically Wrong

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Load Testing as Theater

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Designing for Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Cliff

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Interpreting Results for Learning

What Load Testing Requires from the Bedrock

Chapter 7: Chaos Engineering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What Chaos Engineering Actually Reveals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chaos Engineering as Theater

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Designing for Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chaos Engineering Under Load

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What Chaos Engineering Requires from the Bedrock

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Double-Loop Learning Through Chaos Engineering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chaos Engineering and the WAI-WAD Gap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 8: GameDays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What GameDays Actually Reveal

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

GameDays as Theater

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Designing for Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Debrief Is Where Learning Happens

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Unscripted Versus Scripted Scenarios

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What GameDays Require from the Bedrock

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

GameDays and Knowledge Distribution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

GameDays and the WAI-WAD Gap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 9: Incident Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What Incidents Actually Reveal

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Incident Analysis as Theater

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Designing for Learning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Blameless Analysis in Practice

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What Incident Analysis Requires from the Bedrock

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Learning Loops from Incidents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Incident Analysis and the WAI-WAD Gap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Part 3: Navigating the Gap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 10: Designing for Adaptability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Learning from Successful Resilience

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Who Designs for Adaptability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Communication Patterns That Enable Learning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Hiring and Cultural Transmission

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Knowledge Distribution Systems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Bedrock at Organizational Scale

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Reward Systems That Reward What You Actually Want

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Psychological Safety Beyond the Team

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Resilience That Isn't

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Limits of Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 11: Making Drift Visible

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Recognizing the Pattern of Drift

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Making Trade-offs Explicit

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What Conscious Navigation Looks Like

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Role of Leadership in Making Drift Visible

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Limits of Visibility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 12: The Prevention Paradox

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Asymmetry

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

What Gets Cut, and What Gets Lost

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Tensions Resurface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Delay Obscures Causation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Why Leaders Can't See It

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Pattern Repeats Across Domains

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Navigating the Paradox

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 13: Organizing Resilience Work

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Why Organizational Structure Matters for Learning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Three Models That Struggle

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The SRE Promise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The SRE Reality

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Enabling Team Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Unified Ownership Enables Connections

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

The Enabling Team's Job

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Staffing the Enabling Team

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Measuring What Matters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Chapter 14: The Future of Resilience

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Automation's Learning Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

When Machines Do the Learning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

AI as Learning Mechanism

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Tooling Has Gotten It Wrong

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Starting Where You Are

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Part 4: Resources

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Appendix: Operational Readiness Review Template

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

How to Use This Template

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Can You Have the Right Answers to All Questions?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Who Should Conduct an ORR?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

When Should You Conduct an ORR?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

How Does an ORR Differ from Architecture Reviews?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Template Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

1 - Service Definition and Goals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

2 - Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

3 - Failures, Impact & Adaptive Capacity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

4 - Risk Assessment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

5 - Learning & Adaptation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

6 - Monitoring, Metrics & Alarms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

7 - Testing & Experimentation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

8 - Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

9 - Operations & Adaptive Capacity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

10 - Disaster Recovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

11 - Organizational Learning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Appendix: Incident Analysis Template

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

How to Use This Template

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Moving Beyond Root Cause Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Language That Promotes Learning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Developing Precise Vocabulary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Interview Approach

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Template Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

1 - Title

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

2 - Incident Details

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

3 - Owner & Review Committee

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

4 - Classification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

5 - Executive Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

6 - Supporting Data

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

7 - Customer Impact

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

8 - Incident Response Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

9 - Post-Incident Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

10 - Timeline

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

11 - Contributing Factors Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Discovery & Context

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Decision-Making Under Uncertainty (The Second Story)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Organizational Context & Pressures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Human Factors & Working Conditions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Communication & Coordination

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Technical Systems & Environment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Propagation & Cascades

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Warning Signs & Missed Signals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Knowledge & Preparedness

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

12 - Surprises & Learning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

13 - Action Items

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

14 - Learning Loops & Knowledge Sharing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Quality Checklist

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Works Cited

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Further Reading

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Foundational Works in Resilience Engineering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Organizational Learning and Culture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Practitioner Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Cognitive Science and Decision Making

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Glossary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

A

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

B

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

C

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

D

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

E

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

F

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

G

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

H

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

I

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

L

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

M

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

N

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

O

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

P

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

R

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

S

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

T

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

U

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

V

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

W

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

Y

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/whywestillsuckatresilience.

About the Author

Adrian Hornsby is the Founder and CEO of Resilium Labs, a consultancy that helps engineering organizations improve resilience and develop operational excellence.

Adrian's work centers on the question this book examines: why engineering organizations keep having the same types of incidents despite investing in incident analysis, GameDays, operational readiness reviews, and chaos engineering. He embeds with teams to observe how work actually happens, identifies gaps between stated practice and operational reality, and diagnoses the organizational dynamics that prevent teams from learning and adapting. Rather than implementing prescriptive solutions, he helps organizations see the forces undermining resilience and develop their own capacity to navigate them.

This perspective draws on more than two decades of building and operating software, including nearly a decade at Amazon Web Services. At AWS, Adrian developed open source chaos engineering tools, joined the Fault Injection Service as Principal Engineer, and helped scale the practices discussed in the book across Amazon and AWS.

Outside of his consulting work, Adrian delivers keynotes and workshops on resilience engineering, operational excellence, and chaos engineering, advises VC portfolio companies on engineering practices, and serves on advisory boards for several organizations. He is based in Helsinki, Finland.

Learn more at https://resiliumlabs.com.