



Whitepaper

Managing up, managing down, and the innovation paradox

Why senior-manager failure modes kill the innovation their organizations depend on.

Innovation series.

A synthesis of the innovation-environment framework and the senior-manager failure-mode framework, with a representative case study. Names and sector-identifying details have been removed.

Prepared by

lateralworks
Innovation methodology

Date

April 2026
Synthesis paper

Online

lateralworks.com
Innovation paradox series

Table of contents

Managing up, managing down, and the innovation paradox

How good ideas die	04
Five conditions innovation environments need	05
The manager in the middle	07
The manager who over-invests in managing up	08
The manager who over-invests in managing down	09
How this plays out in performance	10
What good looks like	11
The leader's decision	11
Appendix A — A representative case study	13
References	22

Overview

Abstract

Innovation needs two things that pull against each other. Ideas need room to develop before they are evaluated, and they need rigor to determine whether they are worth pursuing. The paradox, documented by Amy Edmondson and others, is that organizations tuned for execution are structurally hostile to the early, fragile state in which good ideas exist. When rigor arrives too early, promising work dies. Over time, people stop bringing ideas forward.

The companies that solve this paradox across decades — Google, 3M, Amazon, Pixar, Toyota, Analog Devices — share a structural pattern. They **sequence encouragement before scrutiny**. They **separate the creative act from the evaluative one**. They make feedback predictable rather than an ambush. They protect the space for exploration from short-term pressure. They reward the act of contributing, not the outcome alone. These conditions are cultural, but they do not maintain themselves. Senior managers hold them up or tear them down every week.

This paper examines the two ways senior managers fail the innovation environment their companies depend on. One over-invests in managing up: optimizing the upward narrative, absorbing direct reports' work into first-person-plural framing that becomes singular at the executive layer, hoarding strategic context as personal currency. The other over-invests in managing down: deep in the operational weeds, protective of the team, absent from the rooms where direction gets shaped. Both failure modes break the condition the innovation-environment framework calls *protection from above*. They do it for opposite reasons, with consequences that unfold on different timelines.

The paper closes with a detailed case study (Appendix A) from a well-known technology company. A director-level leader solved a three-year-old commercial problem through self-directed weekend learning in agentic AI, delivered two production-grade applications and a live community platform, and in the process resolved an OPEX problem the CEO had been complaining about for months. What followed is the failure mode in compressed form. Senior managers killed the original promotion. The CEO's compromise was a workload doubling rather than a real role change. The CEO then killed the working community platform itself, mandating a year-long product engineering review for a system already deployed and in active internal use, and characterizing the director's production-grade work as "vibe coding."

The case is a textbook instance of the innovation paradox failing at every senior layer of the organization. It also surfaces a second paradox specific to the AI moment: a company that publicly commits to AI-driven efficiency but applies pre-AI engineering controls to AI-built work, eliminating the speed advantage that was the reason for using AI in the first place.

Core thesis. The innovation paradox is a problem of environment. Senior leadership *is* the environment. When the managing-up failure mode dominates the middle of an organization, and when the executive layer above declines to correct it or participates in the same evaluative reflexes itself, no innovation policy can outrun the signal it sends to the team: *wait to be told*.

01

Framework

How innovation environments work

Innovation rarely fails because people lack good ideas. It fails because the organizational environment is hostile to ideas in their early state. When constraints, skepticism, or institutional inertia arrive too soon, two things happen. The specific idea gets killed before it can prove itself. And, over time, people stop bringing ideas forward at all. They learn to wait to be told rather than propose something new.

The evidence for this pattern is no longer debated. Detert and Edmondson's research on workplace silence found that roughly 85 percent of employees have withheld important information from their manager at least once because they feared the consequences of speaking up [1]. Edmondson's broader work frames psychological safety as the single most decisive factor in whether teams surface problems early or late [2]. Morrison and Milliken characterize the systemic version of this behavior as organizational silence [3]. Google's Project Aristotle, a two-year internal study of 180 teams, found psychological safety to be the strongest predictor of team performance, above team composition, individual expertise, or access to resources [4]. A manager cannot produce innovation from a team whose members have quietly decided that innovation is a career risk.

The paradox sits in an uncomfortable spot. The same rigor that makes an organization effective at execution can make it lethal to innovation. Execution wants discipline and predictability. Innovation wants exploration and willingness to fund ideas before their value is clear. Most organizations optimize for the first and are accidentally hostile to the second. The leader's job is to sequence them so that rigor arrives at the right moment, after an idea has room to develop, rather than as the first response to anything new.

Five conditions

Innovation environments need

Companies that sustain innovation across decades share a structural pattern. 3M's 15 percent culture dates to the 1940s. Toyota's kaizen suggestion system launched in 1951. Analog Devices' EngineerZone community has been continuously operated since the late 2000s. The pattern separates these programs from the innovation initiatives that launch with fanfare and fade within a product cycle. It breaks into five conditions.

Sequence: encouragement before scrutiny

Google's 20 percent time gave engineers permission to explore before asking them to justify [5]. Amazon's Working Backwards process has teams write a press release describing the ideal customer experience before anyone asks about feasibility [6]. Pixar reviews films in their rough state rather than waiting for polish [7]. In each case the stress test follows idea development rather than preceding it.

Separation: the idea is not the evaluation

Amazon's PR/FAQ document lets teams iterate on *what could this be?* separately from *what would it take to build it?* Pixar's Braintrust has no authority. Feedback is advisory, not directive, which lets directors be candid about a film's weaknesses without feeling they have agreed to a specific fix. Keeping the creative work structurally distinct from the evaluative work prevents the evaluator from killing the idea simply by being the evaluator.

Ritual: feedback is predictable

Pixar's Braintrust meets on a regular schedule. Toyota's suggestion review is continuous with same-week implementation. Amazon's PR/FAQ reviews are a known step in the development cycle. Predictability builds psychological safety. Ambush does the opposite. The difference between a draft that receives honest feedback every other Tuesday and a draft that gets torn apart by a surprise review called yesterday by a skip-level is the difference between a team that keeps producing and a team that stops.

Protection from above

Every long-lived innovation environment has a senior leader who defends the space against short-term pressure. Google's 20 percent time decayed as product pressure grew and leadership stopped actively defending it; by the mid-2010s, internal reports suggested only about 10 percent of engineers consistently used the program. Analog Devices kept EngineerZone free of marketing interference because senior leaders actively refused to let the community become a sales channel. Pixar's Braintrust survived as a candid-feedback forum because Ed Catmull personally enforced its norms [7]. When the senior-leader protection goes away, the rest of the innovation environment collapses, even if the policy language remains on the intranet.

Process reward: the act of contributing, not the outcome alone

Toyota implements roughly 90 percent of employee suggestions [8] not because every idea is valuable, but because the company treats participation as the measured output. Most of the implemented suggestions are small. One widely cited example: a plant tour guide asked for a hook to hang her bag on during stops. The hook went in that week. The implementation was essentially free. The signal — *your idea gets taken seriously* — kept the system running for more than seven decades. Organizations that demand ROI justification on every suggestion cap themselves at the suggestions they can predict.

These five conditions are well documented. What the literature covers less carefully is that they do not maintain themselves. They depend on behavior by a specific layer of the organization.

02

Failure modes

How senior managers break the conditions

Every one of the five conditions depends on what senior managers do routinely. Sequence requires a manager who does not respond to new work with premature critique. Separation requires a manager who can offer advisory feedback without resolving it into a mandate. Ritual requires a manager who meets with the team on a rhythm that makes hard conversations normal. Protection requires a manager willing to absorb short-term pressure rather than pass it downward. Process reward requires a manager who signals, through attention and resources, that the act of contributing is valued.

When this layer behaves well, the innovation environment holds. When it fails, no organizational policy survives. A company can publish a 20-percent-time policy, install a Braintrust, adopt Working Backwards templates, and see every one of them converge to theater within a year if the senior managers immediately around the work are not maintaining the underlying conditions.

Two failure modes dominate. They are not equally common in every organization, and they are not equally visible. They damage the innovation environment in different ways.

Failure mode one

The manager who over-invests in managing up

This manager treats the relationship with senior leadership as the primary product.

Their calendar is dominated by time spent with people above them: exec meetings, skip-level one-on-ones, strategy offsites, peer coalitions. One-on-ones with direct reports get rescheduled or delegated. They answer executive email fast and their own team slowly.

They invest heavily in how work is framed upward. Decks are polished, status reports optimistic, messaging in all-hands carefully managed. Problems get softened; wins get personally associated. Over time, the gap widens between what leadership believes about the team and what is actually happening.

Strategic context flows up and stops at them. They hoard organizational intelligence as personal currency with executives, and the team operates with a partial view of direction, priorities, and politics. When context does reach the team, it arrives filtered through whatever framing makes the manager look prescient. Herminia Ibarra's work on political skill treats upward-facing effort as a legitimate senior-leadership capability rather than an inherent defect [9]. The failure here is not political fluency. It is political fluency that has consumed everything else.

Work done by direct reports is presented in executive settings in a first-person plural that quietly becomes singular. When a direct report drives an outcome, the manager frames it as their own strategic call that the contributor executed on. When things go wrong, the explanation finds its way to someone beneath them.

They hire for narrative reinforcement. They tolerate weak performers who are loyal. They lose the people who want substantive leadership, often without understanding why, because the exit interviews get filtered too.

For innovation specifically, this manager is structurally incompatible with the protection condition. [A manager whose upward narrative depends on being the broker of their team's work cannot protect a space that produces work they cannot broker.](#) If a direct report develops an idea outside the manager's framing, the manager has two options: absorb the work into the framing, and deny credit to the originator; or suppress the work, and preserve the framing. Neither serves the organization. Both serve the manager.

Failure mode two

The manager who over-invests in managing down

The opposite archetype is less politically visible and just as damaging.

They are deep in the operational weeds: present in every design review, running the hard technical problem-solving themselves, protective of the team to the point of isolation. Their calendar is full of one-on-ones and team syncs and thin on executive exposure. They know their team's work in detail but cannot describe, in a single paragraph, how that work connects to the company's top three strategic priorities.

The failure modes feel virtuous from the inside. The team drifts from company direction because the manager is not in the rooms where direction gets shaped. Their strong work is invisible to leadership, so budget and headcount decisions go against them by default. The manager becomes a single point of failure on hard problems because they solve personally rather than scaling problem-solving through the team, which caps growth at their own throughput. Their career stalls, and the team's trajectory stalls with it.

For innovation, this manager fails differently. The team may be psychologically safe — this manager protects it — but the team's ideas never reach the rooms where resources flow. [A good idea that stays inside a manager's team is a good idea the company does not get.](#) Over a long horizon, this manager produces teams whose best work is, functionally, invisible.

How this plays out in performance

The consequences split along two axes: which metrics are being measured, and how much time is allowed to pass.

Short term, externally observed, the managing-up-heavy manager looks excellent. Executives rate them highly. Promotions come early. Visible projects flow to them. The managing-down-heavy manager looks adequate or underrated because their value shows up in team outcomes that are hard to attribute individually.

Over a longer horizon, the picture inverts. The managing-up-heavy manager's team decays: top-performer attrition, burnout among those who stay, missed commitments once the narrative can no longer outrun reality, a brittle knowledge base because no one invested in development. The managing-down-heavy manager's team is operationally strong but strategically stranded. When the reorg comes, the team takes disproportionate damage despite doing good work.

Aggregate effect

What the org actually loses

The organizational damage sits with the managing-up pattern. It degrades the quality of information the executive layer uses to make decisions. When many managers optimize for narrative over honesty, the C-suite ends up flying on instruments it cannot trust. Real risks surface late as crises rather than early as warnings. Recent research by Hagen and Zhao on middle-manager psychological safety finds that this layer reports the lowest safety scores in most organizations, below the executives above or the front line below [10]. The cascade is not accidental. The managers most incentivized to perform upward are also the ones most accountable in both directions, and they are the first to learn that honest reporting carries a personal cost.

For the manager personally, the reckoning arrives in one of three ways. They move before the narrative-reality gap collapses and carry their reputation to a new employer. They get caught by a visible failure and the executive trust that protected them evaporates. Or they plateau at a level where the next promotion requires demonstrable team-building, a capability they never developed, and their reputation turns out to be thinner than it looked, a pattern McCall and Lombardo documented as a leading cause of senior-executive derailment [11].

For the organization, the reckoning is slower and more expensive. The innovations that would have come from empowered teams never appear. The capable people who would have led those teams leave for organizations whose senior managers amplify their work rather than compete with it. Attrition rises. Executive leadership frames the attrition as compensation, or culture, or onboarding, when the actual cause is structural: [the organization's revealed policy on initiative, communicated through the promotion and retention decisions its senior managers have been making for years, is that initiative is a risk.](#)

What good looks like

The senior managers who perform well over a long horizon do both, and they do them in service of each other. They manage up to secure cover, resources, strategic alignment, and honest air-time for the real state of the work. They manage down to develop people, produce outcomes, and surface the reality that makes their upward reporting trustworthy. Credibility upward is earned by delivering downward. The ability to deliver downward is enabled by the political cover secured upward.

For innovation, these managers maintain the five conditions actively. They sequence encouragement before scrutiny in their own interactions with their teams. They separate the idea from the evaluation in their own feedback. They make feedback a rhythm. They protect the space for exploration even when it produces work that falls outside their framing. They reward the act of contributing, not just the outcome.

The test question for any senior manager is uncomfortable. If my direct reports and my skip-level manager described my performance in the same room, would their accounts be recognizable as the same person? When the accounts diverge sharply, the manager is over-invested in one direction, and the direction of the divergence tells you which one.

Implication

The leader's decision

Every senior manager is making this call every week, whether they name it or not. Every staffing decision, every decision about which idea gets resources and which gets “needed in your current role,” every decision about whose work to broker upward and whose to leave at the local layer, is a choice about which failure mode to tolerate. The organization's innovation pipeline is the sum of these choices across the senior-management layer over time.

The appendix that follows is a specific, recent instance of these choices going the wrong way. The case is representative, not anomalous, and it is what the failure mode looks like when you can see the person it happened to.

Three signals make this case worth reading carefully. First, the work in question is not a marginal contribution — it is exactly the kind of AI-enabled, fast-cycle output the company has publicly committed to enabling. Second, the failure spans every senior layer that touched the work, from the immediate manager up to the CEO. Third, the failure is mechanical, not malicious: each individual decision is locally defensible, and the aggregate is the organization revealing what it actually values when initiative shows up unannounced.

[The case is what the aggregate attrition data in many companies looks like when you zoom in far enough to see a person.](#)

Failure mode

The revealed policy

“Innovation and personal initiative are not rewarded here. Stay in your sandbox. Wait to be told.”

Director-level leader, technology firm
Quoted in Appendix A

A

Appendix A

A representative case study

Names, products, and sector-identifying details have been removed. **The company is a well-known technology firm:** successful, profitable, with a dominant share of its vertical market. During the period described below it was experiencing unusual attrition and executive leadership was cycling through explanations. First compensation. Then culture. Then onboarding. None of those framings changed the trajectory.

The setup

The company had spent roughly three years trying to solve a persistent commercial problem. Customers were buying the platform but were not adopting it at the rate the renewal and expansion model required. Hardware worked. Contracts were signed. The operational chain from purchased capability to routine daily use had a gap that internal teams had repeatedly failed to close.

The conventional proposal, the one the organization had been circling for most of the three years, was a new engagement playbook executed by a substantially expanded customer-engagement organization. The headcount math assumed hiring a cohort of expensive senior engagement specialists distributed across the country's major markets. The OPEX increase would be significant. This had become a recurring issue at the executive level: the CEO had been complaining for months about cost growth in the customer organization, which was already running heavy. No alternative had been produced.

Into this, the company hired a director-level leader with a broad, loosely specified charter that amounted to *fix adoption*. He reported to a senior vice president running a major customer-facing function. That vice president reported, in turn, to the executive who ran the entire customer organization. Both were established, long-tenured operators. Neither had a background in the kind of work the new director was about to do.

Case study

What the director did

Within his first several weeks he performed the gap analysis the company had never performed in three years. The root cause was not training quality, product design, or account management discipline. The company was failing at a much earlier step: it did not reliably know *which external parties* should be receiving the platform's outputs in the first place. The coordination chain had a hole at step one, and everything downstream was compensating for it badly.

He then did something almost no one at a company of this profile does. On his own initiative, without asking and without being asked, he spent his weekends learning agentic AI development tools. No manager assigned the learning. No program funded it. He used the time to build two production-grade applications that replaced a manual research process, roughly ten hours of analyst time per customer location, with an automated system that returned the same answer in seconds, at national scale. [The two applications together delivered, in weeks of nights-and-weekends effort, what the company's own internal estimates had scoped as a team-of-ten project requiring several months and hundreds of thousands of dollars of engineering investment.](#)

He went further. Having studied a twenty-year-old online community built by a large public technology company for its own technical end-user base, a community still active today and widely cited as a model of community-driven customer engagement, he designed, built, and deployed a full-stack branded community platform for his own company's end-user constituency. Role-based onboarding, moderated forums, a searchable knowledge base, gamification, staff dashboards. Working prototype. Live. Seeded with content. Same nights-and-weekends model.

“No one told me to do this. I did it on my own, on weekends, because the problem needed to be solved.”

The combined approach replaced the hiring-heavy plan the organization had been assuming. Instead of a distributed field team of senior engagement specialists spread across the country, the model required a handful of community managers running forums and generating platform content at the center. The cost structure was an order of magnitude lower. The capacity to scale nationally was higher. [The director solved the adoption problem he had been hired to solve, and quietly solved the OPEX problem the CEO had been complaining about since the project began.](#) The second problem had never been his to solve. He solved it anyway.

Case study

The reaction from above

The director's immediate manager and that manager's manager did not understand the technology he was using. Neither had asked for agentic AI. Neither had asked for a community platform. The work was visible, it was working, and it was not theirs to broker upward. It had outrun their narrative.

They did not stop him. Stopping him was no longer safe: the CEO had seen the output, was enthusiastic, and was using it in her own conversations. The two managers did something more revealing. They tolerated the work while maintaining their organizational framing that the director's *real* job was the narrower customer-adoption role they had scoped for him. They praised the innovation in the room with the CEO. They made clear, in rooms without her, that this was not what they needed him to focus on going forward.

Re-read the first failure mode in this paper. The calendar dominated by upward time. The framing polished for executives. The work of direct reports absorbed into first-person-plural phrasing that resolves, at the executive layer, as the manager's own strategic call. [This is that failure mode in a concrete instance, acting on a specific innovative employee in real time.](#)

What the two managers did, mechanically

The two intervening managers tolerated the work because they could not safely stop it. The CEO had seen the output and was using it. But the tolerance was strictly above the line. Below the line, in their own rooms, both managers continued to define the director's job as the narrow customer-adoption role they had originally scoped, and continued to position his AI work as a personal interest rather than a strategic contribution. This is the textbook signal: *the work is fine while the CEO is watching, and not what we need going forward when she is not.* Employees calibrate to the second framing, not the first.

The structural consequence is invisible until it shows up in the aggregate. The director's individual case is a single data point. The company has many directors and senior managers. Each instance of this pattern — work tolerated upward, suppressed laterally — sends the same signal to the rest of the organization. Over a quarter or two it accumulates into a culture where bringing forward unsanctioned work is understood to be a personal cost the contributor absorbs alone, and not a path to recognition.

Compromise

The promotion that was, and then was not, and then was something else

The CEO made a decision. The company needed an AI-native leader to carry this work across the organization, and the director had just demonstrated that he could be that leader. She contacted him directly, without routing through the two intervening managers, to tell him he was being considered for a new senior role. A new function. Reporting outside the customer organization. A significant promotion.

The director was, by his own account, genuinely excited. The conversation lasted roughly twenty-five minutes. He notified his immediate manager shortly afterward as a professional courtesy and received what seemed like cautious support.

The next day, the head of the customer organization killed the promotion. His stated reason was that the director was “needed in his current role.” The CEO, whose original instinct had been correct, did not let the kill stand cleanly. She intervened a second time and proposed what she framed as a compromise. The director would take on the new AI work at twenty percent of his time while continuing his existing customer-organization role at the other eighty. A title adjustment for the twenty percent portion was floated: a director-of-product slot under the product VP, contingent on cross-functional negotiation. A backfill requisition would open for his existing role. The director would lead the search for his own replacement. Once a successor was hired and ramped, he could move to the new role full-time.

In the abstract, this was a reasonable bridge. In practice, the structure of the offer, stripped of its language, was simpler than its presentation: *do two jobs for the pay of one, and we will revisit whether you can have just one of them later*. No salary adjustment. No formal title at the senior level that had originally been described. [The compensation for taking on a corporate-wide AI portfolio was the implicit promise of a real role to come, contingent on a backfill process the director himself was being asked to drive.](#) The original promotion had been a recognition. The compromise was a workload increase wrapped in transition language.

The director accepted. Declining was not safer than accepting. The same senior managers who had just killed the original offer were the people he would have to face if he turned down the second one. The CEO had personally championed the path. The risk of being seen to refuse opportunity outweighed the visible cost of the workload, and the AI work was where the company actually needed leadership. He took the deal.

Platform

The platform gets killed

The director's community platform — the second of his two AI builds, the one that had replaced the hiring-heavy plan and resolved the CEO's standing OPEX concern as a byproduct — was at this point a working prototype. Live. Industry-compliant. SOC2-aware authentication. Already loaded with company content: training videos, customer case studies, white papers, written material from across the organization. Internal teams were using the platform's gamification mechanics to compete on who could publish the most content. The platform was, by any operational definition, working.

The CEO killed it.

Her stated reason was that the company could not have a customer-facing product without first running it through the standard internal product engineering process. Specification, design, development, validation, decision gates. The full lifecycle, which inside this company was a year-and-then-some affair. The platform — already built, already deployed, already populated with content, already in active internal use — was to be paused, rewritten through the process, and reapproved at each gate. [In nearly every organization, that is how a working product is killed.](#)

The replacement plan was for the company to procure a commercial learning management system and graft community functionality onto it later. There was a problem with the plan, which the director tried to surface. The platform he had built was not just an LMS. It was an LMS-and-community hybrid, customized to the specific customer constituency the company served, integrated with the agency identification system he had also built. No commercial product on the market combined the two functions. The off-the-shelf LMS the company was now expected to purchase did one of the two functions and did it generically. Replacing a working, customer-customized hybrid system with a generic single-function tool was not a process improvement. It was a downgrade, and a costly one, paid for in license fees that would persist annually.

“Vibe coding”

In the conversations leading up to the platform decision, the CEO characterized the director's work as “vibe coding.” The term has emerged in the past year for casually-prompted, prototype-grade output from AI tools. It is the technical equivalent of calling a finished house a sketch.

The director's own response to that characterization, in a separate written exchange with a senior colleague, is worth quoting close to verbatim:

“As juxtaposition to ‘vibe coding’: my builds are full-stack, SOC2-compliant, auth-hardened, production-ready systems. I incorporate elements like the relevant legal frameworks into my build rules and edge-case testing. Current degradation points are at over 50 concurrent user submissions per minute, but most of that is governed by the AWS hosting tier I’m running on and can be modified on expansion. I’d put my builds up against a vendor. I have E&O and Cyber liability insurance, and at this point, two thousand hours in the last twelve months in learning and build time alone.”

Second paradox

The bureaucracy paradox

There is a deeper failure underneath the platform decision. The standard product engineering process the CEO invoked — specification, design, development, validation, decision gates across a year-and-then-some timeline — was built for an era of product development that agentic AI coding is making obsolete. **It is the control architecture of slow development.** It assumes building the system will take months. It assumes the cost of a wrong commitment is high. It assumes that gates exist to catch errors before they propagate, and that the sequence of expert review at each gate is the company's defense against expensive rework.

Agentic AI coding inverts those assumptions. Building takes weeks. The cost of a wrong commitment is low because rebuilding is cheap. Errors propagate at the speed of iteration, which makes iteration itself the defense against them. The director's two thousand hours of learning and build time across twelve months were not the production of a single artifact waiting for evaluation. They were the rapid iteration cycle, applied as a working method.

Agile development, in its original framing twenty years ago, promised this kind of rapid iteration. In most organizations, including this one, the promise was never fully realized. Control systems were too entrenched. The gates were too consequential. The reviewing layers had too much accountability for outcomes they could not move quickly enough to influence. Agile became, in practice, a vocabulary applied to a slower process rather than a structurally different way of building.

Agentic AI coding is the first development methodology that delivers what Agile described. Production-grade code, written quickly, by individual contributors with the right tools, validated continuously rather than at gates. The director was using this methodology and producing in weeks what his company's traditional process would have produced in eighteen months at a hundredfold the cost. The platform was the proof of the method. It was working.

The CEO's response was to send the working proof of the new method through the gates of the old one.

A company that wants AI-driven efficiency but applies pre-AI controls to AI-built work is asking the technology to deliver outcomes the company's process is designed to prevent. The two cannot coexist. The control system was not designed to govern work of this kind. It was designed to govern multi-year engineering programs run by the company's existing staff. Forcing AI-enabled output through it does not validate the work. It eliminates the speed advantage that was the entire reason for using AI to do the work.

Predicament

What the director was actually asked to accept

Stack the three decisions together. The director is now being asked to continue performing his existing customer-organization role at full intensity, simultaneously build out a new corporate-wide AI function with no compensation change and no senior-level title, and watch the first serious AI product he built — the one that solved the problem he was hired to solve and incidentally resolved the CEO's standing OPEX concern — be paused, sent through a year-long review process likely to kill it, and replaced by a less capable commercial alternative.

The implicit deal is: take on the additional responsibility, prove yourself in the AI domain, and you will eventually have a real role with real authority. The reality of the deal is: do two jobs for one salary, and watch your existing AI work be discarded by the same leadership chain that is now asking you to commit to more of it. [The data the director has on whether the implicit promise will be honored is the organization's actual treatment of the work he already finished.](#)

The message the director received

The director's own summary of the experience is worth quoting, because he stated it plainly and because it generalizes beyond his own case:

"Innovation and personal initiative are not rewarded here. Stay in your sandbox. Wait to be told."

[That sentence is the company's actual policy on innovation.](#) Not the policy written in the culture deck. The policy communicated through action at the moment it was tested. Employees calibrate to the revealed policy, not the stated one. The company's attrition problem, previously characterized as compensation or onboarding or culture, is most accurately characterized as this: *capable people are receiving a consistent message that initiative is a career risk, and they are acting on it.*

Synthesis

Why this case matters

Three elements of this case map to specific claims in the body of this paper. A fourth surfaces a related but distinct paradox. Together they no longer implicate only the senior-manager layer. The case implicates the executive above it as well, and the broader engineering organization that supports her.

The director was punished for doing what he was hired to do, twice. First by the suppression of recognition: two production AI systems, an OPEX resolution that was not in his scope, and the suppression came from the exact layer whose structural function is to enable people like him. Then again, more visibly, by the platform being sent through a lifecycle gate process designed for ideas at a much earlier stage of maturity than this one had reached.

The “promotion” was not a promotion. It was a workload increase wrapped in transition language. The compensation for taking on a corporate-wide AI portfolio was the implicit promise of a real role to come, contingent on a backfill process the director was himself being asked to drive. The company secured the additional output without paying for it. Whether or not that was the intent, it is the structure.

The executive layer’s failure was not retreat. It was active participation. The CEO did not just decline to override her senior managers on the original offer. She killed the director’s working platform herself, characterizing his production-grade work as “vibe coding” in a way that revealed she did not understand what had been built.

The company is structurally unprepared for the technology it has committed to. The CEO has publicly committed to deploying AI across operations. The director showed what that looks like in concrete form: an AI-built system that produced a year-plus result in weeks. The leadership chain’s response was to send the result through the gate process designed for multi-year programs run by the existing engineering organization. [The control system cannot govern the development methodology AI enables, because the control system was designed for the methodology AI replaces.](#)

The test question, applied

If the director’s direct description of his performance and his manager’s description of his performance were read in the same room, would they be recognizable as the same person? They would not. The director would describe an employee who identified and solved a three-year-old commercial problem through self-directed learning and nights-and-weekends execution, resolved a standing executive cost concern as a byproduct, and shipped a production-grade hybrid platform that no commercial alternative replaces. His manager would describe a valuable individual contributor who is needed in his current role and whose side projects, while interesting, should not distract him from his primary responsibilities.

Both accounts are internally coherent. [Only one is honest about what actually happened.](#) The direction of the divergence — from *transformational* downward to *useful contributor* upward — tells you which failure mode is operating here. The pattern is the policy.

Reader questions

What the case raises

This case is presented without a resolution. The director, at the time of writing, has not yet made his decision. The questions below are the ones the case raises for any leader, manager, or contributor who recognizes the pattern in their own organization.

What should the director do?

Accept the workload-doubling and bet on the implicit promise? Decline and accept the cost to his standing? Stay and continue building, knowing the next AI product he ships may meet the same fate as the last one? The honest answer depends on factors outside the scope of this paper: his financial position, his professional reputation, his appetite for risk, the strength of his outside options. What the case foregrounds is that none of those personal factors should be load-bearing here. A company that has just demonstrated it cannot reliably distinguish a working production system from a casual prototype is asking him to bet his next two years on the assumption that the next decision will go differently.

What did this company do wrong?

The failure is not at one layer. The senior managers killed the original offer. The CEO retreated, then negotiated a compromise that doubled the workload without changing the title or compensation. The CEO then killed the platform itself. Each decision is defensible in isolation. Each is consistent with how the company has historically managed risk. [The pattern is what the company did wrong](#): the consistent signal, sent through actions taken at moments that mattered, that initiative will be tolerated but not enabled.

Why is attrition a problem at this company?

Executive leadership has framed it, in turn, as compensation, culture, and onboarding. None of those framings have changed the trajectory because none of them is the actual cause. The case suggests a different frame. The company is losing people because its revealed policy — the policy communicated through the promotion and platform decisions reviewed in this case — is that exceptional contribution is met with workload increase rather than compensation, and that exceptional work is met with process delay rather than deployment. People who can produce exceptional work are also people with options. They calibrate quickly, and they leave.

What did the CEO miss?

The opportunity in the director's work was not the AI tools, and not even the OPEX resolution that came with them. The opportunity was the demonstration that the company already had, on its current payroll, a leader capable of moving faster on AI than anyone she could plausibly hire externally. [The cost of the misunderstanding she signaled with "vibe coding" was not the platform, which can be rebuilt elsewhere by someone else with more patience](#). The cost was the message every other capable employee at the company received about what kind of work this company is actually willing to value when the work shows up. That message will continue to compound. The director's case is one data point. The aggregate is the attrition number the company has been trying to explain.

Sources

References

- [1] Detert, J. R., and Edmondson, A. C. "Implicit Voice Theories: Taken-for-Granted Rules of Self-Censorship at Work." *Academy of Management Journal*, Vol. 54, No. 3, June 2011, pp. 461–488.
- [2] Edmondson, A. C. *The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth*. Wiley, 2019.
- [3] Morrison, E. W., and Milliken, F. J. "Organizational Silence: A Barrier to Change and Development in a Pluralistic World." *Academy of Management Review*, Vol. 25, No. 4, 2000, pp. 706–725.
- [4] Google re:Work. "Guide: Understand Team Effectiveness" (Project Aristotle). rework.withgoogle.com.
- [5] Page, L., and Brin, S. "Letter from the Founders: An Owner's Manual for Google's Shareholders." Google IPO filing, 2004.
- [6] Bryar, C., and Carr, B. *Working Backwards: Insights, Stories, and Secrets from Inside Amazon*. St. Martin's Press, 2021.
- [7] Catmull, E., and Wallace, A. *Creativity, Inc.: Overcoming the Unseen Forces That Stand in the Way of True Inspiration*. Random House, 2014.
- [8] Liker, J. *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. McGraw-Hill, 2004.
- [9] Ibarra, H. *Act Like a Leader, Think Like a Leader*. Harvard Business Review Press, 2015.
- [10] Hagen, J. U., and Zhao, B. "Middle Managers Feel the Least Psychological Safety at Work." *Harvard Business Review*, October 2025. <https://hbr.org/2025/10/middle-managers-feel-the-least-psychological-safety-at-work>
- [11] McCall, M. W., Jr., and Lombardo, M. M. "Off the Track: Why and How Successful Executives Get Derailed." Center for Creative Leadership, Technical Report No. 21, 1983.