

Outcome Observability: How Organizations Lose Value Quietly and How to See It Coming¹

Mehdi Kadaoui

1.0 Abstract

Most organizations declare success the moment a system goes live. But what happens next often tells a different story. Behavior shifts back to old patterns, workarounds creep in, and the outcomes everyone expected start to quietly unravel. The signals are there — they're just not being tracked.

This article introduces Outcome Observability as a discipline designed to help organizations detect and respond to that post-go-live drift. It's not about more dashboards or new KPIs. It's about staying connected to the original intent of the transformation, long after the delivery phase ends.

Drawing on recognizable patterns, leadership shifts, and real cases, this article shows what Outcome Observability is, how it works in practice, and why it matters — especially for leaders who care about more than just going live on time. It's for those who want to know: *Did we actually realize the outcome we set out to achieve?*

Because transformation doesn't fail loudly. It fades, if no one's watching.

2.0 Introduction

Most transformations don't fail on day one. They fail later...quietly, gradually, and often without a single moment of clear collapse.

The system goes live. The dashboard lights turn green. Executives applaud the delivery. Meanwhile, on the ground, something less visible is starting to happen. Behavior begins to shift back. Adoption hesitates. Teams create workarounds. The new processes don't quite land, or don't hold. At first, these signals are small, too small to trigger alarms. But they grow.

And because everyone has already declared success, no one is looking for them.

¹ How to cite this work: Kadaoui, M. (2025). Outcome Observability: How Organizations Lose Value Quietly and How to See It Coming, *PM World Journal*, Vol. XIV, Issue X, October.

Outcome Observability begins when the delivery teams step back, and someone needs to stay present. It's the discipline of noticing when outcomes are starting to bend away from intent. Not through metrics alone, but through behavior. Not just in results, but in how people actually work, decide, and adapt.

Where most models end (at go-live) this approach begins.

What you'll find in the pages ahead isn't a toolkit or checklist. It's a way of staying connected to what was promised, and a method for knowing whether it's still unfolding... or quietly unraveling.

2.1 Success Lives In The Gray Zone

Most organizations still treat transformation success as a binary outcome. It worked, or it didn't. It launched, or it failed. The project closed. The benefits were logged.

But that's not how transformation plays out in the real world.

The truth is: most outcomes don't arrive all at once. They **emerge** and sometimes, they drift. They begin to take shape, partially. Some behaviors change. Some stick. Others fade. The technical rollout completes, but the cultural shift remains incomplete. The dashboard tells one story. The day-to-day tells another.

This is the gray zone — where success isn't clear-cut, but conditional.

And unless you're watching closely, you won't see what's slipping away.

Leaders need a way to stay present in that zone, not by flooding it with metrics, but by listening for the small signals: resistance, reversions, workarounds, fatigue. These don't show up in the PMO report. But they're real. And if ignored, they'll define the actual outcome.

Transformation doesn't fail in one moment. It fails in increments and in silence.

As McKinsey noted in their global transformation study, over 70% of large-scale efforts fall short of achieving sustained impact. ([McKinsey](#))

3.0 What Outcome Observability Really Is

At its core, Outcome Observability is about keeping the transformation alive after delivery. It's a way to detect when the promised change is holding and when it's slipping.

It doesn't require a new tool. It requires attention.

Think of it as a light governance layer that stays connected to lived experience — to what's actually happening after the system goes live, the training is done, and the spotlight has moved on. It's not meant to control or interfere with operations. It exists to answer one question that most organizations quietly stop asking:

Did we actually realize what we said we would?

Too often, that question gets lost beneath layers of dashboards and status updates. The numbers take over — clean, confident, and automated. But numbers don't always tell you if the intent behind the change is still intact. And once behaviors begin to shift, it becomes harder to tell whether you're still on track... or just meeting targets that no longer mean what they used to.

Outcome Observability brings a different posture — not just measuring whether something is working, but observing how it's working, who it's working for, and whether it's working in the way it was originally meant to.

It's not about checking boxes or assigning blame. It's about staying close to the purpose that sparked the change in the first place — and making sure it's still alive in how people actually work.

3.1 What The Business Needs To Observe

Outcome observability doesn't mean watching everything.

It means being able to see the right things (at the right time) to know whether a transformation is working.

What must the business observe?

- **Value:**

Are the promised benefits (cost, speed, experience, compliance) actually materializing?

- **Adoption:**

Are users engaging with the new tools, processes, or structures or finding ways around them?

- **Behavior:**

Are decisions aligned with the designed state or are old mental models reasserting themselves?

- **Continuity:**

As new changes occur; releases, restructures, parallel programs, are we still holding the line?

3.2 IT Observability Is Well-Established. Outcome Observability Is Still Emerging

In IT, observability is no longer optional. Systems are built to be watched. Performance, error rates, latency, dependencies — everything has a trace. Logs, metrics, alerts — the infrastructure is designed to speak up when something's wrong.

We've built entire teams and platforms around this principle. SREs, NOCs, AIOps — all working to make sure that if the system starts to drift, we'll know before users do.

But in business operations? In post-transformation behavior?

There's no equivalent.

Once the project is delivered, most observability stops. We assume value will emerge as planned. We expect adoption will happen. We trust that alignment will hold.

And when it doesn't, we usually find out too late — not because the signs weren't there, but because no one was looking for them.

That's the gap. Technical observability is mature. Business observability — especially post-go-live — is still catching up. In many organizations, it hasn't even been defined.

Outcome Observability is a response to that gap. Not to replicate IT observability, but to create a parallel discipline — one focused on value, behavior, and realization.

The signals are different. The tools are lighter. But the intent is the same; don't wait until value is lost to realize it's slipping.

As Harvard Business Review points out, digital transformations often fail not because of technology, but because organizations lose sight of the behavioral and strategic shifts required to realize lasting impact. ([HBR](#))

3.3 How Outcome Observability compares to IT Observability

<u>Aspect</u>	<u>IT Observability</u>	<u>Outcome Observability</u>
Purpose	Monitor system behavior and performance in real time	Assess whether transformation outcomes are taking hold and lasting
Focus	System health, performance, and errors	Value, and behavioral alignment
Signals tracked	Metrics, logs, traces	Adoption patterns, decision logic, value shifts
Start point	Begins in development and continues through operations	Begins during delivery and persists across future change
Ownership	DevOps, SREs, engineering teams	Transformation owners, function leads, Process Owner
Detects	Failures, anomalies, latency	Logic drift, missed value, behavioral reversion
Outcome	Resilient, performant systems	Aligned, value-generating business behavior

Business doesn't need telemetry, it needs traceable meaning.

4.0 Embedding Observability During Delivery and Beyond

4.1 Outcome Observability Starts In Delivery, Not After Go-Live

Outcome observability must begin **before** go-live, during the formation of behaviors and outcomes.

It must begin during delivery, when behaviors are forming, when outcomes are being built, not just implemented.

In agile and iterative environments, outcome observability must become part of the rhythm, not a checkpoint after the fact.

Dimension	Key principles
------------------	-----------------------

When	<ul style="list-style-type: none">- Begin as soon as outcomes are defined- Embed observability logic during delivery, not after go-live
How	<ul style="list-style-type: none">- Define value and adoption signals early- Use realization checkpoints across delivery- Track usage patterns, behavioral drift, and workarounds continuously
Who	<ul style="list-style-type: none">- Business stakeholders must own observability- PMs and delivery teams can support but should not lead- Without business ownership, observability becomes hollow reporting

Ownership may live within the domain most impacted (e.g., Finance, Ops, HR) but must be formally assigned, with explicit continuity accountability.

4.2 A Strategic Layer, Not a Final Step

Outcome observability is not just a post-transformation function.

It's a **persistent governance capability**, not a constant stream like IT observability, but a layer the business must maintain to protect value and alignment over time.

While it may operate with less frequency and automation than IT's telemetry, it's no less essential. Why?

- Transformations cascade. One change triggers another.
- Drift is structural. It can emerge months or even years later.
- Dependencies grow. One release can destabilize another domain.
- Behavior is fragile. Even adopted change can revert if no one is watching.

IT observability is about uptime and performance.

Outcome observability is about coherence and continuity and without it, we're flying blind between transformations.

5.0 Outcome Observability in Practice: The Three Shifts

Outcome Observability isn't a tool. It's a posture. And like any posture, it shows up through habits — small behavioral shifts that change how teams see, interpret, and respond to what's unfolding after delivery.

You don't need to stand up a new function or deploy another platform. You need to shift how the organization listens to outcomes — and who's responsible for keeping that listening alive.

In practice, three shifts make the difference between passive reporting and true observability. They're not dramatic. But when they're absent, value fades unnoticed.

Spot Signals, Not Just Metrics

Outcome Observability doesn't mean adding more KPIs to the dashboard. It means recognizing weak signals that show whether transformation is truly embedding or quietly eroding.

For example, if users are logging into a new system but still doing their work in Excel, adoption metrics may look "green," but the signal tells a different story. This is where value slippage begins, not with failure, but with silent bypasses.

Common trap: Organizations confuse login activity or training completion with actual behavior change.

In practice: Teams should define 3 to 5 behavioral signals before go-live, not just usage metrics, but actions that reflect true adoption (e.g., how often manual steps are skipped, whether old forms are still in use, how frequently exceptions are raised).

Make Ownership Visible

Who is watching?

In many transformations, delivery teams ship the work, but no one owns the reality that follows. Outcome observability requires someone to take accountability for watching what matters, not just closing tickets.

That "someone" is often not a single role. It can be a business process owner, an operational leader, or a cross-functional steward.

What matters is that ownership is declared and doesn't disappear after go-live.

A business lead who routinely checks post-rollout support trends and challenges the team when uptake plateaus, is playing the true role of a value steward.

Common trap: Ownership is assigned to a team or function, not to an individual. Everyone assumes someone else is watching.

In practice: Assign a named business owner, not from IT, but from the domain being transformed. That person must know what to look for, when to escalate, and how to decide whether signals reflect friction, resistance, or the need to intervene.

Stay in the Loop, Lightly

Observability doesn't mean reporting. It means staying connected to reality as it evolves.

This connection must be light enough not to burden teams but strong enough to detect early drift. A regular checkpoint. A real usage conversation. A review of support trends. These loops keep transformation alive and adaptive, not frozen in launch-day assumptions.

A 15-minute monthly review between Ops and Transformation, focused only on usage signals and pain points, may do more to safeguard value than any dashboard.

Common trap: Success is declared (prematurely) at go-live, and governance has stopped, no one is following.

In practice: Schedule recurring checkpoints for a deep reflection about the situation, not just for reporting. And then ask questions about adoption, value realisation, workarounds, etc. For example: Are people using the system as intended? Are manual workarounds creeping back? What's the business sentiment around this change?

Creating value is just the start — keeping it alive is what really counts. If you're not staying close, you're falling behind.

5.1 This Discipline Only Lives If Leaders Want It To

You can't just throw a tool at observability and call it done. And no, you can't automate your way out of it either. This isn't something that bubbles up from the grassroots — it needs to start at the top.

Here's the real issue: if leadership treats go-live like the end of the road, observability never stands a chance. It just won't stick.

But when leaders keep asking, “*Are we still heading toward the outcomes we set out to achieve?*” — that changes the game. Suddenly, teams have a reason to tune in, to notice what’s happening, to speak up when things drift.

This isn’t about adding more reports. It’s about staying engaged. Staying awake. And only leadership can make that feel necessary — not optional.

Because without that tone from the top, observability doesn’t evolve. It just becomes another checkbox. And slowly, it disappears.

6.0 Where Outcome Observability Comes to Life

To make outcome observability tangible, here are three real-world-inspired mini-cases. Each one highlights a specific pillar in action, showing how early signals led to intervention, ownership, and real-time course correction.

These aren’t success stories. They are value-saving loops.

6.1 Mini-Case 1: Usage Shift Before Adoption (Pillar 1 – *Spot Signals, Not Just Metrics*)

Context

A large insurance company had deployed a new claims interface to speed up processing. After training, usage rates were near 100%. Adoption, by standard metrics, seemed successful.

Signal

Analytics revealed something strange: most users were still navigating via the legacy keyboard shortcuts, despite the new interface being mouse-optimized and designed to improve accuracy. No one had flagged this. But the pattern emerged in usage friction.

The Loop

An outcome observability checkpoint flagged this behavior. The product owner, operations lead, and a training rep ran a lightweight review with a sample of end-users. The feedback: the new flow added clicks without speeding things up, so users reverted to shortcuts.

What Changed

The team reworked the screen flow, aligning it better with how users actually process claims. Adoption metrics didn’t change, but *real* adoption now matched intent. Without signal-based observability, this friction would have remained invisible.

6.2 Mini-Case 2: Ownership Vacuum Post-Launch (Pillar 2 – *Make Ownership Visible*)

Context

In a regional health system, a new digital appointment scheduling platform was rolled out to reduce call-center load and missed appointments.

Signal

After three weeks, usage data showed digital scheduling was increasing. But the call center load hadn't dropped. The dashboard showed volume. But it didn't explain *why*.

The Loop

An outcome observability review asked: "Who owns this now?" It turned out that after go-live, no single business owner was assigned to watch the full process. A new cross-role steward was assigned. They surfaced the issue: patients were using the tool, but then calling to confirm their booking manually, unsure it had worked.

What Changed

A confirmation SMS was added, and a new post-booking FAQ clarified next steps. Within two weeks, call volume dropped. The metric moved *after* accountability was installed. Observability without ownership was just passive noise.

6.3 Mini-Case 3: Strategic Drift Detected in Usage (Pillar 3 – *Stay in the Loop, Lightly*)

Context

A national public-sector agency launched a compliance-tracking portal to help departments flag, resolve, and track recurring regulatory issues.

Signal

Everything was working. Issues were logged, and dashboards were populated. But a quarterly review (the observability loop) uncovered a deeper signal: most submissions were "closed" within one day, often with vague notes like "handled offline."

The Loop

The cross-functional team held a value loop review with business sponsors. They asked: *Are these workflows actually resolving root issues, or just cycling through the motions?*

What Changed

They added a light gate: issues could not be closed without identifying either a policy change, a user fix, or a process adjustment. Suddenly, closure took longer, but now it had meaning. Strategic observability was restored.

6.4 When No One Is Watching

Not every drift has a headline. Most don't. They surface quietly — in the corners of the organization no one's tracking. Here are three real-world patterns, seen in the aftermath of well-delivered projects that slowly lost their grip on intent.

- **The Phantom System**

A financial institution successfully deployed a new risk engine. It passed all tests, showed stable performance, and was technically in use. But after six months, analysts were still pulling data into spreadsheets to replicate the old model. It wasn't malicious — just familiar. The new logic never took hold.

Summary: Without behavioral visibility, systems run — but decisions don't change.

- **The Quiet Collapse**

A major healthcare provider launched a new scheduling system. Training was completed, usage was logged, and leadership moved on. But doctors found the interface clunky and reverted to WhatsApp. No one had defined adoption signals. No one noticed the workaround. Appointments dropped. Patients paid the price.

Summary: Drift begins where observation ends.

- **The Drift That Became the Norm**

A government agency digitized a paper-heavy process. The solution was well designed and well received. But two critical steps still required manual handoffs between teams. No one caught it during design. No one revisited it after launch. Over time, those gaps created delays, friction, and a growing perception that “the new system doesn't work.”

Summary: In the absence of light-touch follow-through, gaps become truths.

7.0 From Signals to Sensemaking: Reading What Matters

Outcome Observability isn't just about surfacing signals, it's about interpreting them with context. That requires ownership, discussion, and a shared lens across business and delivery. Otherwise, organizations risk acting on noise.

<u>Signal Type</u>	<u>Possible Drift</u>	<u>Response Pattern</u>
Users using old workarounds	Behavior hasn't shifted	Re-check training, incentives, or actual process fit
Tickets rising again	Friction is re-emerging post go-live	Investigate root cause; update supporting artifacts
KPIs stable, but feedback drops	Silent disengagement	Initiate qualitative check-ins or pulse reviews

7.1 From Signal to Response

Outcome Observability only works if signals are taken seriously and met with structured attention.

That might mean:

- Escalating a drift in adoption behavior back to the transformation sponsor.
- Validating with operational leads whether process shortcuts are becoming norms.
- Reviewing whether original trade-offs are still valid under current conditions.

Observability isn't just watching. It's governance with eyes open.

For example:

if the signal is *"The new tool is being used, but old processes persist,"* this might indicate that:

- Process alignment never happened, just tool deployment.
- Incentives still reward old behaviors.
- Training focused on features, not purpose.

This is exactly what outcome observability reveals: not just usage, but meaning. And when those meanings surface early, leadership can step in, not to control, but to steer. Before intent fades. Before outcomes drift. Before value erodes.

Here’s what truly shifts when observability is embedded.

7.2 Where Outcome Observability Changes the Game

<u>Without Outcome Observability</u>	<u>With Outcome Observability</u>
Outcomes are assumed, not tracked	Outcomes are traced and verified
Adoption is declared, not observed	Adoption is evidenced with signals
No visibility after go-live	Realization is monitored over time
Business walks away post-delivery	Business owns and stewards the value
Metrics are generic or misaligned	Metrics are tied to value and behavior
Dashboards show usage, not impact	Dashboards signal realization and drift
Changes break things without detection	Signals surface erosion before failure

7.3 What Outcome Observability Looks Like

Let’s ground this concept in concrete shifts not tools, but **disciplines** embedded in the way we deliver and sustain change:

<u>Practice Area</u>	<u>Without Observability</u>	<u>With Outcome Observability</u>
Post-Go-Live Review	One-time report. Often delayed.	Ongoing light-touch check-ins.
Behavior Tracking	Assumed through training completion.	Tracked through usage and pattern changes.

Adoption Visibility	Based on self-reporting or perception.	Based on embedded signals and user behaviors.
Process Drift	Not visible until complaints emerge.	Detected through monitored weak spots.
Ownership	Assumed by delivery or PMO.	Declared and traceable to the business.
Operational Handovers	One-directional and often brittle.	Governed through shared continuity practices.

8.0 How to Start Without Permission

Most organizations don't build Outcome Observability into their transformation model. That shouldn't stop you from starting.

You don't need a new role or an official process. You need one person who's still paying attention after the delivery team moves on. One person asking: *Is the outcome actually happening — or did we stop watching too soon?*

Here's how to begin, without permission, politics, or new frameworks.

8.1 Pick One Transformation to Watch

Choose a transformation that its recent or at high stakes, where drift could have consequences. That would be your starting point.

What you need is a clear and precise pilot, there's no need to cover the entire portfolio.

Then ask yourself:

- Are we still realizing the outcomes we committed to?
- Is anyone actually responsible for monitoring the outcome?

8.2 Define 3 to 5 Outcome Signals

Focus on practical signals (not KPIs), especially the subtle ones:

- Do people continue to use the new process, or do they quietly return to it?

- Do workarounds reappear without anyone recording them?
- Do decisions reflect the new operating model or the old one?

What matters is visibility, don't look for precision. Start small and make sure what you are watching is clear.

8.3 Assign Temporary Stewardship

Find someone close to the field: a sales manager, an operations manager, it has to be someone who has skin in the game. Give them temporary Stewardship for 30 days, not as a new role but just a clear request:

- Follow the signals
- Report if something isn't right
- Call a results loop if necessary

Taking charge — even temporarily — creates and enables Focus.

8.4 Run One Outcome Check-In

Run one short session (15 min is enough) to check the pulse and ask *are we still achieving what this transformation was meant to deliver?* And if the answer is unclear then you have to pause and check; Where is the drift? What changed?

Don't try to solve everything, just surface it.

8.5 Make One Drift Visible

Any real sign of divergence, a reversion or a contradiction must be documented, shared and named. That simple act will trigger the shift from unnoticed erosion to shared awareness.

Outcome Observability doesn't need permission to start, it needs a presence. Choose one outcome, track what matters and stay with it. That's how transformation stays real.

9.0 What Outcome Observability Is Not

Outcome observability could be seen as just another post-project hygiene which is a complete misunderstanding.

Outcome observability is — definitely — not:

- A KPI report shared quarterly
- A checklist for audit compliance
- A dashboard labeled “value tracking”
- A renamed PMO follow-up
- A tech-led monitoring system

Outcome observability is not a thing you measure. It's a capability you build. One that helps an organization notice misalignment before it becomes normal.

It's how you make sure your intent survives beyond PowerPoint slides and that value doesn't disappear in silence.

10.0 Conclusion

Outcome Observability is not a dashboard. It is a discipline. A way for the business to remain present (during delivery and beyond) not just as a sponsor, but as a steward of meaning.

If strategy matters, so must traceability. If value matters, so must vigilance.

And it starts with one simple act: pick one transformation, identify three signals that matter, and assign one real owner who won't disappear after go-live.

That's not an oversight. That's the difference between symbolic success and real, sustained transformation.

About the Author



Mehdi Kadaoui

Brussels, Belgium



Mehdi Kadaoui is a senior IT and transformation leader with over 17 years of international experience delivering complex enterprise programs across Europe, the Middle East, and North America. His career spans global logistics, SaaS, telecom, and public sector environments, where he has consistently bridged the gap between strategy and execution to drive sustainable digital outcomes.

As a former director and transformation strategist for industry leaders such as DHL, Atlas Copco, Efficacy, and Saudi Post Logistics, Mehdi has led ERP modernization, AI/analytics enablement, cloud-native migrations, and IT operating model redesigns. He is known for unifying cross-functional teams (IT, Data, Finance, CX) under shared governance models that enable real-time intelligence, compliance-by-design, and post-go-live value continuity.

Mehdi holds an Engineer's Degree in Computer Science and a Master's in Big Data & Systems Integration. He is PMP-certified, a Certified Scrum Product Owner (CSPO), and holds cybersecurity and digital transformation credentials from ISC2 and Stanford University. He has also completed advanced SAP coursework and delivered operational excellence across SAP S/4HANA landscapes in the Americas and Europe.

A contributing writer for **CIO.com**, Mehdi publishes widely on structural drift, transformation governance, and post-go-live failure modes. He is the author of the **"From Intent to Outcome"** series—an independent body of work exploring semantic governance, business observability, and transformation integrity frameworks. His published frameworks (Project Semantix, Outcome Observability, Structural Agile) offer new ways to govern meaning, behavior, and value beyond traditional delivery models.

Mehdi is trilingual (French, English, Arabic), based in Brussels, and actively advises global enterprises on how to deliver not just systems, but strategic intent with integrity.

Contact at em.kadaoui@techevolve.be or [From Intent to Outcome Canon](#)