SINGULARICS

White Paper

What to Measure

Measure the model’s mechanics, not delivery volume. Volume metrics will look strange during transition and tempt people to game them.

Alexander S. Petty
SINGULARICS | singularics.ai

First published August 2026.

singularics.ai

Contents

Abstract

When delivery cost falls toward zero and the constraint moves to human judgment, the metrics an organization watches have to move with it. Volume metrics designed for scarce delivery capacity will look strange during transition and will tempt people to game them. This paper defines two classes of measurement for the AI-native operating model: leading indicators watched weekly to diagnose the model's mechanics, and outcome indicators reviewed monthly to confirm those mechanics are producing results. It names what not to measure and why, catalogs eleven failure modes with their observable tells and corrections, and explains how the Replace rate serves as a single diagnostic for slice quality. The governing principle throughout is that these metrics diagnose the model's health, not individual performance, and should never be used for individual evaluation.

Measurement in the AI Era

Every enterprise we have worked with enters this transition carrying a dashboard built for the old constraint. Velocity charts, story point burn-ups, utilization heat maps. Within six weeks of adopting AI-native delivery, every one of those charts becomes misleading. Velocity spikes because the fleet produces more output per day than the old team produced per sprint. Leadership sees the spike and declares success. Meanwhile, the rework rate has doubled because nobody is watching whether the output is right. The dashboard says the machine is running. It does not say the machine is running in the right direction.

The guiding principle of iterative delivery has not changed. Work in small increments, sequence the highest value first, and get feedback quickly. What changed is the cost of an increment. When agent fleets can produce working software in

a single day, the old metrics stop measuring the real work and start measuring what is easy to count.

Velocity measured how much a team could deliver in a sprint. That was useful when delivery was the constraint, because it told you whether the constraint was loosening or tightening. When delivery stops being the constraint, velocity measures how fast the machine runs, which is no longer the interesting question. The interesting question is whether the right person made the right call, whether the harness proved it, and whether the organization learned from it.

> *When execution was the constraint, we measured execution throughput. Now that judgment is the constraint, we measure judgment quality. The metrics have to follow the constraint.*

This is not an argument against measurement. It is an argument for measuring different things. An organization that keeps watching velocity during the transition will see numbers that look either impossibly good or confusingly bad, depending on how the work is sliced, and both readings will provoke the wrong response. Impossibly good numbers invite complacency. Confusingly bad numbers invite intervention that disrupts a model still finding its footing.

Three principles govern what belongs on the dashboard.

Measure mechanics, not volume. The daily cycle has moving parts: Decision Owner presence, spec quality, harness integrity, blocker resolution, capacity release. Each can be observed directly and each predicts whether the model is working. Volume follows when the mechanics are healthy and stalls when they are not, so watching volume is watching a trailing shadow rather than the thing casting it.

Separate leading from lagging. Leading indicators tell you this week whether something is drifting. Outcome indicators tell you this month whether the drift mattered. Looking at outcomes alone means learning too late. Looking at leading indicators alone means reacting to noise.

Never measure individuals. The moment these metrics are used for individual performance evaluation, people will optimize for the metric rather than for the model. A Decision Owner whose presence rate is tracked for their review will show up and not engage. A Fleet Lead whose rework ratio is tracked will stop recording rework. The metrics exist to diagnose the system, not to judge the people inside it.

Leading Indicators (Weekly)

Leading indicators are watched weekly because they diagnose the model's mechanics in near-real time. They tell you whether the daily cycle is closing properly before the consequences of a broken cycle show up in outcomes.

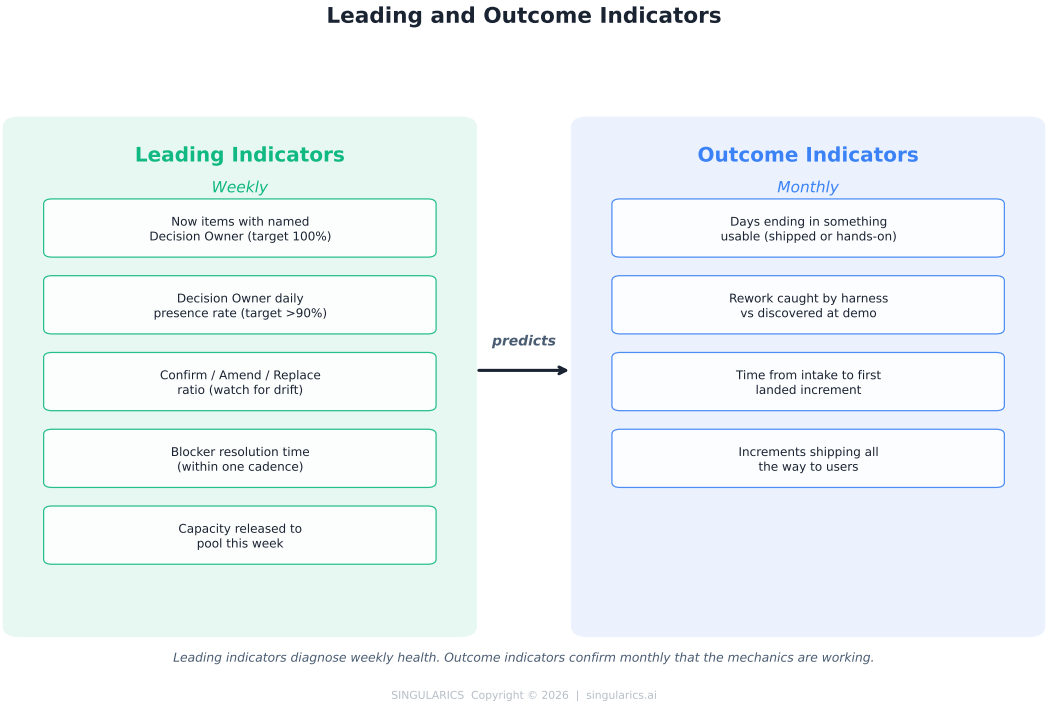

Figure 1: Leading indicators are watched weekly and predict outcome indicators reviewed monthly. The leading set diagnoses the model's mechanics. The outcome set confirms those mechanics are producing results.

Percentage of Now items with a named Decision Owner at the committed release level

Target: 100%. This is not aspirational. It is the gate.

The No Owner No Now rule says nothing enters the Now horizon without a named Decision Owner released to at least 60% of their working week. This metric simply counts compliance. If the number drops below 100%, some work is in flight without the person who can accept it, and the daily cycle for that work is not closing.

The correction is never to relax the gate. It is either to release the owner properly or to return the work to Next until an owner is available. Work without an owner accumulates decisions nobody made and nobody will defend.

The decision rule: if the percentage drops below 100% for more than one consecutive week, the Flow Council must act at that week's calibration. Either a Decision Owner is released, or the unowned work returns to Next. No third option. Tolerating unowned work in Now for even two weeks produces an average of eight unreviewed increments, each carrying accumulated risk that is invisible until the demo.

Decision Owner daily presence rate

Target: above 90%.

Daily presence is the mechanism. A Decision Owner who is present four days out of five is running at 80%, which means one day in five the team builds without judgment and either stalls or accumulates risk. At 60% presence the model is running on a deputy more often than not, which is a proxy by another route.

Presence means available to the team for questions, present at the demo, and present at daily planning. It does not mean sitting in the team's workspace all day. The measurement is simple: was the owner reachable when needed, did they attend the demo, and did they attend daily planning? Three yeses is a present day. Anything else is an absent day.

Confirm, Amend, and Replace ratio

No fixed target. Watch for drift.

At daily planning the team makes one of three calls about tomorrow's work. Confirm means the plan holds. Amend means the plan needs adjustment. Replace means the plan is wrong and a new spec is needed. A healthy team runs roughly 60–70% Confirm, 20–30% Amend, and 5–15% Replace over any rolling four-week window.

The ratio is a diagnostic, not a scorecard. Persistently high Replace means slices are too big. Persistently zero Replace means the work is so well understood that it may not need this model. A sudden spike in Replace suggests the outcome was under-specified at intake or the environment shifted. Each pattern has a different correction, which is why watching the ratio is worth more than targeting a number.

Blocker resolution time

Target: within one cadence.

A blocker raised at daily planning should be resolved by the Flow Council the same evening or, at the latest, carried to the Intent Council at the next check-in. A blocker that outlives one cadence is a blocker the governance structure failed to clear. Counting how often that happens, and how long blockers stay open, tells you whether the escalation path is functioning or decorative.

Persistent long-lived blockers usually mean one of two things. Either the Flow Council is not meeting its resolution commitment, or blockers are being raised that do not belong at that level and the team is skipping the Decision Owner's authority. Both are worth knowing.

Capacity released to the pool this week

No fixed target. Watch for zeroes.

When an outcome is accepted, the people and the fleet return to the pool and the Flow Council redeploys them. If the released capacity number is zero week after week, one of two things is happening. Either no outcomes are being completed, which is a separate problem. Or outcomes are being completed but the people are staying where they are, which means the organization is rebuilding standing teams under new names.

Release is the half most organizations skip, and skipping it is what kills fungible capacity. Making it visible on a weekly dashboard is the simplest way to hold the discipline.

Outcome Indicators (Monthly)

Outcome indicators are reviewed monthly because they need enough data to distinguish signal from noise. A single week's outcome numbers are volatile. A month's pattern is information.

Percentage of days ending in something usable

This is the most direct measure of whether the daily cycle is producing value. Each day should end with either a shipped increment reaching real users or a landed increment in a hands-on environment where stakeholders can use it themselves. Days that end with neither are days the cycle did not close.

The target is not 100%. Some days end in learning rather than shipping, and that is legitimate. But the percentage should be consistently above 70% over a month. Below that, the daily cycle is not closing often enough to justify its cost, and the cause is somewhere in the leading indicators.

70% minimum over a rolling month. Below 70%, look at Decision Owner presence first, then spec quality, then slice size. The cause is almost always one of those three, and the leading indicators will point to which one.

Rework caught by the harness versus discovered at demo

The harness is the Decision Owner's instrument. It proves each increment against the spec continuously in CI. Rework discovered by the harness before the demo is cheap. Rework discovered at or after the demo is expensive, because it means the harness did not catch something it should have.

Over time, this ratio should move sharply toward the harness. If rework keeps appearing at the demo, one of three things is wrong. The harness is testing the implementation rather than the spec. The spec criteria are not checkable. Or the Fleet Lead is not reviewing checks against criteria. Each has a different correction, and the ratio tells you to look.

Time from intake to first landed increment

This measures how quickly the model converts a request into something tangible. In a healthy model, the time from intake to first landed increment is measured in

days, not months. If the number is growing, look at the leading indicators. The most common causes are slow owner assignment, slow blocker resolution, and oversized first slices.

This is a system metric, not a team metric. It measures how long the intake pipeline takes to produce work, form a team, assign an owner, and land the first result. Blaming a team for a slow intake-to-increment time is like blaming a car for traffic.

Percentage of increments shipping all the way to users

This measures pipeline and clearance maturity rather than team performance. An increment that lands in a hands-on environment but never reaches production is a clearance gap, not a quality gap. Watching this percentage tells the Flow Council where to invest in clearing deployment paths.

A low number is not a failure of the model. It is a signal that the regulated path has not yet been established for the relevant change classes. Establishing it is high-value Flow Council work, because each class cleared converts every future increment in that class from a queued release into a same-day one.

A Tuesday Reading

The measures earn their keep in combination. Here is one composite reading and the decision it forced. Owner presence sits at 84%, just under the line. The Confirm rate has run above 90% for three weeks, Replace at zero. Rework caught at the demo rather than the harness is creeping up. Any one of those alone is a shrug. Together they tell one story. The owner is stretched, planning has stopped being a real decision, the specs are gliding through unchallenged, and the harness is quietly falling behind what the specs claim.

The correction was not a dashboard review. The council raised the owner's release with the sponsoring officer, and daily planning got its Replace question back. Three weeks later the Replace rate was 8% and demo-stage rework had turned. That is what a measure with a decision attached looks like in practice.

The Anti-Metrics

Four metrics that most organizations carry into this transition actively distort the behavior the model depends on. Each should be retired, and each deserves an explanation of why.

Velocity

What it distorts: slice size. Velocity measures how many story points or units of work a team completes per sprint. When delivery was the constraint, velocity told you whether the constraint was loosening. When delivery is no longer the constraint, velocity measures machine throughput, which is not the interesting question.

Worse, velocity creates a ratchet. A team that delivers more this sprint is expected to deliver at least as much next sprint. In a model where daily slices vary in size based on what the outcome needs, that ratchet punishes teams for doing the right thing. A week spent on a hard specification problem that produces one small, correct increment will look worse on a velocity chart than a week of easy builds that produced five increments nobody needed.

The distortion is specific: teams will enlarge slices to inflate the count, or avoid spec-and-explore days because they produce zero velocity. Both behaviors undermine the one-day cycle. Spec-and-explore days are legitimate outcomes, and a metric that penalizes them penalizes learning.

The counterargument is that leadership needs a throughput number. They do, and the model provides one: percentage of days ending in something usable. That metric captures throughput without rewarding volume for its own sake. If leadership insists on a story-point equivalent, the conversation is about trust, not about data, and no metric will resolve it. Address the trust gap directly rather than inventing numbers to fill it.

Story points

What it distorts: estimation. Story points are an estimation tool for scarce delivery capacity. They answer the question "how much of our limited capacity will this consume?" When capacity is no longer the constraint, the question has no useful answer. An agent fleet does not have a velocity in story points. Assigning points to work that a fleet will complete in hours is an exercise in fiction.

The deeper problem is that story points invite negotiation about size rather than conversation about clarity. In this model, the only sizing question to check is whether the spec fits on one page and the criteria are checkable. If it does and

they are, the fleet can build it in a day. If it does not, the slice is too big. That is a simpler and more honest assessment than a planning poker session.

The distortion is that teams will spend time estimating rather than specifying. A 30-minute estimation session is 30 minutes not spent on criteria quality, and criteria quality is what determines whether the harness proves anything useful.

Utilization

What it distorts: availability. Utilization measures whether people are busy. In this model, the most valuable thing a Decision Owner does is wait. Not idle waiting, but responsive availability: being reachable within minutes so the fleet is never blocked on a judgment call. A Decision Owner at 100% utilization is a Decision Owner who cannot answer questions, which means the fleet stalls and the daily cycle does not close.

Measuring utilization in an AI-native model rewards the opposite of what the model needs. It rewards packed calendars over responsive availability. It rewards visible activity over judgment quality. It rewards being busy over being present.

The distortion is concrete: a Fleet Lead blocked for four hours waiting on a judgment call has burned a build day. That cost is invisible to a utilization dashboard that shows the Decision Owner at 100% productive. Retire utilization. Replace it with Decision Owner presence rate, which measures the real question: was the owner reachable when the team needed them?

Agent output volume

What it distorts: direction. It is tempting to measure how much output the agent fleet produces: lines of code, pull requests merged, tasks completed. All of these are pure volume metrics, and all of them reward the fleet for producing more rather than producing right.

An agent fleet that produces ten thousand lines of code against a vague spec has created ten thousand lines of rework. An agent fleet that produces two hundred lines against a precise spec has created value. Volume does not distinguish the two. The harness does. Measure what the harness proves, not what the fleet produces.

The distortion is that volume metrics reward the fleet for running fast in any direction, including the wrong one. Speed of production raises the cost of ambiguity: an agent fleet pointed at a vague spec produces more wrong output per hour than a human team ever could. The only metric that separates productive volume from destructive volume is the harness, which checks what was built against what was specified. Volume without the harness is noise with a number attached.

The flow metrics, read second

The DORA and flow measures deserve their own sentence, because they are good measures that answer a different question. Deployment frequency and change lead time measure the machine half of the system, and in this model the machine half is rarely the constraint. A portfolio can post excellent flow numbers while owners are absent and the Replace rate sits at zero, which is a system optimizing its healthy half while its judgment half decays. Keep the flow measures if they are already instrumented, and read them after the judgment measures, not before.

Failure Modes and Their Tells

Every operating model has characteristic failure modes. The value of naming them in advance is that each has an observable tell, a behavior you can see in the metrics or in the room, which means the correction can start before the failure becomes structural. The table below catalogs eleven failure modes, the tell that reveals each, and the correction.

Failure Modes and Their Tells

Failure Mode	Tell	Severity
Decision Owner is a proxy	Acceptance takes >1 day; owner says "let me check with..."	Critical
Owner released on paper only	Misses daily planning more than once a week	Critical
Slices too big	Replace rate >30%; increments span more than one day	High
Spec as afterthought	Team starts building while spec is still open	High
Harness proves implementation	Coverage high, defects still reach the demo	High
Harness not in CI	Runs before demos rather than on every push	High
Demo-only acceptance	Accepted because it looked right	High
Capacity never released	Same people on same Effort beyond the outcome	Moderate
Trains reassemble	Flow Council members advocate for "their" teams	Moderate
Network becomes a help desk	Experts pulled in reactively after teams stall	Moderate
Escalation without resolution	Blockers raised but not resolved within one cadence	High

Severity: Critical: breaks the daily cycle · High: degrades quality silently · Moderate: weakens the model over time

Figure 2: Eleven failure modes with their observable tells and severity levels. Critical failures break the daily cycle. High-severity failures degrade quality silently. Moderate failures weaken the model over time.

Decision Owner is a proxy

Tell: Acceptance decisions consistently take more than a day, or the owner says “let me check with” before accepting.

This is the most damaging failure because it is silent. The role is filled, the person shows up, and the cycle appears to run. But every acceptance is a round trip to someone else, and the one-day cycle silently becomes a three-day cycle. The tell is in the elapsed time between demo and acceptance. If it is consistently more than a few hours, the person accepting is not the person deciding.

Correction: Replace the owner. Do not coach around this one. A proxy cannot be trained into an owner. The person who holds the domain knowledge must be identified and released.

Owner released on paper only

Tell: The owner misses daily planning more than once a week.

The sponsoring officer agreed to release this person at 60%, but their standing obligations were not actually reduced. They attend when they can, which is not daily, and the cycle degrades on the days they are absent.

Correction: The Flow Council raises the release level with the sponsoring officer, or the outcome returns to Next until the release is real. Paper commitments that are not honored are worse than no commitment, because they consume a team's capacity while starving it of judgment.

Slices too big

Tell: Replace rate above 30%, or increments regularly span more than one day.

When every demo overturns the plan, the organization is planning further ahead than its knowledge allows. The spec describes a week of work in one document, the demo invalidates most of it, and the team spends more time replanning than building.

Correction: Halve the slice. Re-run the checkability test on every criterion. If it does not fit on a page, it is two specs.

Spec as afterthought

Tell: The team starts building while the spec is still open.

The speed of the fleet makes this tempting. Why wait for the spec to be signed when the fleet could start now? Because the fleet will build whatever it is pointed at, including the wrong thing, and the cost of building the wrong thing at fleet speed is higher than the cost of waiting an hour for the spec.

Correction: Hard gate. No build day starts without a signed spec. The Fleet Lead enforces this, and the Flow Council holds the Fleet Lead accountable for it.

Harness proves the implementation

Tell: Coverage is high, but defects still reach the demo.

This happens when agents write tests against their own implementation rather than against the spec. The harness turns green because the tests confirm what the code does, not what the spec requires. Coverage numbers look healthy and the defects keep coming.

Correction: The Fleet Lead reviews checks against criteria, not against code. This is the single most important review the Fleet Lead performs, and it is why the Fleet Lead role cannot be eliminated.

Harness not in CI

Tell: The harness runs before demos rather than on every push.

A harness that runs sometimes is a suggestion. A harness that runs on every push is a gate. The difference is enforcement, because a suggestion can be overridden when the team is under pressure, and pressure is exactly when the gate is most needed.

Correction: Move it into the pipeline. Until the harness runs on every push, it is not a harness. It is a testing script.

Demo-only acceptance

Tell: Acceptance happens because it looked right in the demo.

A demo shows one path through the system. The harness checks every path, every time. Acceptance based on a demo alone is acceptance based on incomplete evidence, and it accumulates risk with each increment that passes without full verification.

Correction: Return to evidence. The acceptance decision is made against the harness results and the spec criteria, not against a walkthrough. The demo is how humans build intuition about what was built. It is not the acceptance gate.

Capacity never released

Tell: The same people remain on the same work beyond the accepted outcome.

This is how standing teams rebuild themselves under new labels. The outcome is met, but nobody returns to the pool because the team has work to do and the people are comfortable. Within two quarters, the pool is empty and every team is permanent.

Correction: Make release a minuted Flow Council action with a date. If it is not recorded and tracked, it does not happen.

Trains reassemble

Tell: Flow Council members advocate for “their” teams rather than for the portfolio.

The Flow Council exists to hold a portfolio view. When its members start protecting capacity for work they personally care about, the portfolio function has been captured by its constituents and the pool cannot be re-planned.

Correction: Change how Flow Council members are evaluated. They are evaluated on the whole portfolio’s outcome, not on the performance of any single team or work stream.

Network becomes a help desk

Tell: Domain experts are pulled in reactively after teams stall, rather than injecting knowledge before the build starts.

The Domain Knowledge Network works on a push model. Experts read the day's specs and inject the constraints and context before anyone asks. When the model degrades to pull, experts are interrupted mid-day by teams that are already stuck, and their contribution is reactive troubleshooting rather than proactive governance.

Correction: Restore the push model driven by daily planning. Yesterday's daily planning names the expert tomorrow's spec needs. The network reads that signal and connects the expert before the day starts.

Escalation without resolution

Tell: Blockers are raised but not resolved within one cadence.

Escalation is only useful if it produces a decision. A blocker that is raised to the Flow Council and acknowledged but not resolved is a blocker that the governance structure has absorbed rather than cleared. The team remains stuck, and the escalation becomes a ritual rather than a mechanism.

Correction: The Flow Council lead who owns the blocker is named publicly, with a resolution deadline. If blockers routinely outlive their cadence, the Flow Council's composition or authority needs examination.

The Replace Rate as a Diagnostic

Of all the metrics described here, the Confirm/Amend/Replace ratio deserves special attention because it is the single best diagnostic for slice quality, and slice quality is what determines whether the daily cycle produces value or friction.

At daily planning, after the demo, the team makes one of three calls about tomorrow's work. Confirm means the queued spec is pulled as written. Amend means the spec needs adjustment but not replacement. Replace means the plan is wrong and a new spec is needed.

> *If every demo overturns the plan, your slices are too big. A demo that invalidates a week of planned work means you planned a week ahead of your knowledge. A demo that adjusts tomorrow means you sliced correctly.*

Persistently above 30%

A Replace rate persistently above 30% means the team is learning something expensive at every demo. That is not a failure of the team. It is a failure of slice size. The spec is describing more work than one day's demo can validate, so each demo invalidates more than it confirms.

The correction is always the same: halve the slice. If the outcome is genuinely exploratory and cannot be sliced smaller, change the cadence. Run the work as a series of spec-and-explore cycles rather than pretending it is incremental delivery. The model accommodates exploration. It does not accommodate pretending that exploration is delivery.

Persistently at 0%

A Replace rate of zero over more than four weeks means the team never learns anything surprising. That suggests the work is so well understood that the daily cycle is adding ceremony without adding learning. The team already knows what to build, and the spec-demo-accept loop is confirming what was already certain.

This is not a problem to solve. It is information about whether the work needs this model. Some work is well understood, and well-understood work does not benefit from daily judgment. It benefits from a clear plan and efficient execution. If Replace is at zero, ask whether the outcome would be better served by a simpler delivery approach.

The ratio tells you about slice quality

The Replace rate is not a performance metric. It is a quality signal about the specs, not about the people. A high Replace rate does not mean the Decision Owner is making bad calls. It means the slices are too large for the team's current learning rate. A zero Replace rate does not mean the Decision Owner is doing excellent work. It means the work is not teaching the team anything new.

Use the ratio to calibrate the size of specs, the depth of the spec queue, and the ambition of each day's increment. It is a thermostat, not a scorecard.

How to Use These Metrics

Watch for drift, not targets

Most of these metrics do not have a target that is right for every organization. What they have is a healthy range, and the signal is whether the numbers are moving toward that range or away from it. A Decision Owner presence rate of 85% that was 70% last month is better news than a presence rate of 92% that was 98% last month. The direction tells you more than the position.

Set the dashboard to show trends over rolling four-week windows. A single bad week is noise. A four-week slide is signal.

Diagnose the model, not individuals

These metrics diagnose the operating model's health. They tell the Flow Council whether the mechanics are working, where they are drifting, and what to correct. They are system metrics.

The moment they are used for individual performance evaluation, they stop working. A Decision Owner whose presence rate appears on their review will find ways to appear present without being available. A Fleet Lead whose rework ratio is tracked will stop recording rework discovered at demo. A Flow Council member whose capacity release numbers are watched will release people on paper and reassign them the next day.

> *The metrics exist to diagnose the system, not to judge the people inside it. The moment they are used for individual evaluation, people will optimize for the metric rather than for the model.*

If the organization needs individual performance metrics, those metrics should come from the outcomes the person was accountable for, not from the operating model's instrumentation. Did the Decision Owner's outcome ship? Did it produce the business result? Those are performance questions. Whether their presence rate was 92% is a system health question, and the two should never be conflated.

Review cadence

Leading indicators are reviewed weekly at Flow Council calibration. The review takes ten minutes and asks one question for each metric: is it drifting, and if so, what is the correction?

Outcome indicators are reviewed monthly, typically at the Intent Council's biweekly working session that falls nearest the month's end. The review asks whether the leading indicators predicted the outcomes correctly. If they did, the instrumentation is working. If they did not, either the leading indicators are measuring the wrong things or the connection between mechanics and outcomes is weaker than assumed. Both are worth knowing.

Handling anomalies

An anomaly in a leading indicator is a prompt for a conversation, not an automatic corrective action. The Flow Council sees that blocker resolution time spiked this week. Before acting, they ask why. Perhaps a single complex blocker drove the average up and has since been resolved. Perhaps the escalation path has degraded. The metric tells you to look. It does not tell you what you will find.

The worst response to a metric anomaly is a policy change announced the same week. The second worst is ignoring it. The right response is a conversation with the people closest to the work, held within days, that surfaces the cause and agrees on a correction if one is needed.

The decision framework for anomalies: if a leading indicator moves outside its healthy range for one week, note it. If it persists for two consecutive weeks, investigate at calibration. If it persists for three weeks, the Flow Council owns a named correction with a deadline. One week is noise. Two is a question. Three is a problem with an owner.

Velocity and Its Cousins

This measurement framework replaces three things that most organizations carry into a transition.

The velocity dashboard. Velocity, story points, and utilization are removed from the primary dashboard. They may still be collected if existing reporting requires them, but they are not used to diagnose the operating model's health and they are never used for team evaluation.

The status report. When the harness runs on every push and the evidence ledger records every acceptance decision, status is readable directly. Nobody compiles

a status deck. The governance forum opens by reading the ledger, not by hearing it summarized.

Individual performance metrics derived from system instrumentation. These metrics diagnose the system. The moment they appear on an individual performance review, people optimize for the metric rather than for the model, and the metrics stop measuring what they were designed to measure.

The Real Object of Measurement

Decision ownership. Decision Owner assignment and presence rate are the two most critical leading indicators. The Decision The role's qualification test, time commitment, and bounded authority are what these metrics monitor.

Harness integrity. The rework ratio (harness versus demo) directly measures harness integrity. The four layers of coverage and why QA as a phase disappears when the harness is working.

Cycle closure. The percentage of days ending in something usable is the outcome indicator that measures whether the daily cycle is closing. The hourly sequence explains why the cycle must close every day.