RIVER / The Canonical Statement

CANONICAL STATEMENT V0.6 / JUNE 2026

The canonical statement, *in full.*

This document is the canonical statement of RIVER. It is audience-agnostic, tool-neutral, and written to serve as the substrate from which tailored artifacts for specific readers, organizations, and operational contexts are derived.

Contents

I / Premise

What RIVER is.

RIVER (Realized Impact and Value Evolution Reporting) is a framework for measuring and operating the full value chain of software delivery, from the idea that prompted the work through the impact the work produces. It is built for organizations that practice, or are moving toward, progressive and reversible release, and it takes the separation of deploy from release as a foundational premise rather than an implementation detail.

The framework exists because the modern practice of release has made the segments of the value chain downstream of deploy separately observable and separately operable for the first time, and because no framework binds those segments to a per-release operating artifact. RIVER encompasses the four metrics of DORA (the DevOps Research and Assessment program) as the reference instrumentation for the commit-to-deploy segment, and extends measurement and discipline across release, adoption, and impact.

This document is the canonical statement of the framework. It is audience-agnostic, tool-neutral, and written to serve as the substrate from which tailored artifacts for specific readers, organizations, and operational contexts are derived.

II / The problem RIVER addresses

A question every organization asks and none can answer.

Every software organization is trying to answer a single question that, at the current state of practice, it cannot rigorously answer: **is the value we generate equivalent to, or greater than, the cost of our staff and platform?** CEOs ask it. Boards ask it. CFOs ask it. Engineering leaders are asked to defend their scope and headcount against it. Product leaders are asked to defend their roadmap against it. The question is central to how modern software businesses are operated, funded, and held accountable.

The reason it cannot be answered is that the value stream in most software organizations is fractured across functions. Product defines outcomes in terms of business metrics. Engineering ships code and measures itself in terms of delivery performance. Operations reports on uptime, incidents, and service-level objectives. Each function has its own metrics, its own success criteria, and its own vocabulary, and the objectives set at one end of the value chain routinely fail to translate to the measurements taken at the other. The question of whether the work produced value never lives inside a single artifact that all three functions share.

Measurement compounds the fracture. For most of the industry's history, measurement of engineering work effectively stopped at deploy. A commit becomes a deploy, and the deploy

reaches users; delivery performance and business impact could be conflated because there was no meaningful gap between them. In the current world, where feature flags, progressive rollouts, and targeted cohorts have cleanly separated deploy from release, that conflation has become incoherent. Deployed code may sit unreleased for weeks. A release may reach one percent of users and be reversed without any redeploy. Elite delivery performance and zero business impact now coexist routinely. The industry's research frameworks have begun to describe that gap; none of them closes it, because describing a gap after the fact and operating on it release by release are different practices.

RIVER is the framework that fills that silence.

III / Why the problem is newly tractable

Deploy and release, once separated.

The problem RIVER addresses is not new in principle; it is newly tractable. For as long as software has been measured, there have been gaps between what teams ship and what businesses gain from shipping. What has changed is the instrumentation. Over the past decade a set of practices has propagated through the industry: feature flags, progressive delivery, canary releases, cohort targeting, guarded automation, reversible experimentation. These practices have made the release-and-after segment of the value chain separately observable. That which can be separately observed can be separately measured. That which can be separately measured can be separately operated on.

Before these practices were common, a deploy was effectively a release. Measurement of delivery performance was, in practical terms, measurement of user impact, because there was no meaningful temporal or cohort-based separation between the two events. DORA's four metrics were defined in and for that world, and they measure it well. The world they were defined for no longer describes how most high-functioning software organizations ship today.

RIVER is possible now because the operational separation of deploy from release has become mature enough to carry a shared vocabulary. It was not possible ten years ago. It is not merely possible now; it is necessary, because the gap between what the four delivery metrics measure and what the business asks has become wide enough that it can no longer be bridged informally. And the gap is no longer widening at human speed.

IV / The acceleration

The faster the factory, the more load-bearing the delta.

AI is compressing the segment of the value chain that was already best measured. DORA's own research across 2024 and 2025 documents the shape of that compression: as AI adoption rises,

nearly every upstream process measure improves (documentation, code quality, review speed) while delivery performance, and stability in particular, degrades. The finding DORA draws from this is the delivery-layer case of RIVER's thesis, stated in DORA's own voice: gains in the development process do not translate automatically into gains in delivery.

The same research observes that AI does not drain the value from software work; it "expedites its realization," compressing the time between starting valuable work and finishing it, and freeing capacity. What fills the freed capacity is left open. That open question is the framework question. Capacity aimed at volume produces more deployed, unreleased, unmeasured output, faster. Capacity aimed at impact requires a declared target to aim at, and nothing in the delivery toolchain supplies one.

The arrival of agents sharpens the problem from one of speed to one of context. An enormous amount of organizational knowledge passes through humans implicitly: the guardrails, the routing, the value-context that human workers carry without being asked to articulate it. Agents do not carry that context. When an agent takes over a piece of work, what gets lost is precisely the unwritten answer to what the work is for. DORA's 2025 research on AI names the pace problem agents create and answers it with the mechanisms it has: continuous integration, small batches, fast feedback. Those mechanisms govern how fast work moves through the factory. None of them records what the work is for, or binds the agent to it.

RIVER's answer is structural. The release delta, defined in Section VIII, makes the value-context explicit: hypothesis, success signal, target cohort, and horizon, declared before exposure. A human team and an agent operating against the same declared delta are bound by the same context, because the context is no longer implicit. Acceleration does not weaken the case for declaration; it is the case. The faster the factory runs, the more load-bearing each declared delta becomes, because the volume of undeclared work the freed capacity can produce grows with the speed. The evidence for the acceleration is DORA's own research, which is reason to treat that research fairly rather than to claim it as a trophy.

V / DORA, fairly treated

What DORA measures, and where the difference now lives.

DORA is the most successful measurement framework in the modern history of software engineering, and RIVER is built to encompass its metrics, not displace them. Treating DORA fairly is essential, both to intellectual honesty and to RIVER's adoption. The framework RIVER is most often confused with is also the framework it most depends on.

DORA measures the commit-to-deploy segment of the value chain through four metrics: deployment frequency, lead time for changes, change failure rate, and time to restore service. These four have earned their standing. They give teams a rigorous, comparable, and portable account of delivery performance. They made team-level and organization-level self-location possible where,

before them, conversations about engineering performance were largely anecdotal. They survived the scrutiny of longitudinal research, published annually, and refined over a decade. They are the standard against which any new measurement framework in this space must calibrate itself.

Treating DORA fairly in 2026 also means describing where its research has gone, because the common shorthand, that DORA stops at deploy, is no longer accurate. DORA's research has moved steadily downstream. Its 2024 report concedes the attribution problem directly: effects nearest the point of impact are easy to attribute, and attribution gets harder the farther downstream the effects travel. Its generative AI research engages value explicitly, at the layer of the developer's experienced value of their own work. Its 2025 report adopts the flow of work from idea to customer as a unit of analysis in its value stream management chapter. A framework that claims DORA cannot see past deploy is arguing with a version of DORA that no longer publishes.

What survives DORA's expansion is a distinction of posture. DORA reaches downstream as research and diagnosis: surveys, telemetry, factor scores, correlations, retrospective value-stream maps. It observes outcomes after the fact and correlates them with capabilities. It does not bind a declared artifact to each release before exposure and evaluate the release against it after. That is not a deficiency of DORA; it is the difference between an instrument and an operating discipline, and it is the axis on which RIVER is positioned. DORA's value stream analysis tells an organization where the constraint in its stream is. The release delta is how the organization acts on it, release by release.

DORA's 2025 report makes this division legible from its own side. Its chapter on metrics frameworks separates the framework from the metrics from the collection method, distinguishes frameworks by the goal they serve, and names product excellence and organizational effectiveness as goals that require different frameworks than delivery performance. In that taxonomy, RIVER is a framework whose goal is value realization, and the four DORA metrics are its measurement for the delivery segment.

RIVER therefore adopts the DORA metrics as-is for the segment they cover. There is no RIVER metric that replaces a DORA metric; for the commit-to-deploy segment, DORA is RIVER's measurement. What RIVER adds is a per-release operating artifact, and measurement scoped to that artifact, for the segments past deploy.

One vocabulary collision needs naming, because both frameworks now lean on the same word. In DORA's recent research, value most often means the developer's experienced value of their own work, alongside outcome factor scores. In RIVER, value means realized business and user impact, scoped per delta and evaluated against a declared success signal. The Glossary holds the formal disambiguation; this document uses the RIVER sense throughout.

VI / RIVER, defined

The framework, in definition.

RIVER is a framework for measuring and operating the full value chain from idea to impact in organizations that practice progressive, controlled, and reversible release. It names, types, and measures the segments of the value chain past deploy: exposure, cohort progression, reversal, guardrail activation, experiment linkage, adoption, and impact. It organizes those measurements around a structured artifact, the release delta, that is declared before a release begins and evaluated after. It provides a maturity ladder that describes how organizations grow into the practice. It treats the whole system (framework, artifact, metrics, ladder) as a coherent operating discipline, not as a dashboard.

RIVER is philosophy-coupled to the separation of deploy and release. It does not make sense in organizations that ship in a single binary event from commit to user. It is tool-neutral in expression: any organization practicing progressive and reversible release can instrument RIVER, using any combination of platforms and internal systems that produce the required data. The framework's credibility depends on this portability.

VII / Worldview

The five commitments RIVER is built on.

RIVER rests on five commitments about how release should be practiced. These commitments are descriptive of where the industry's most capable organizations are already operating and prescriptive for organizations that are moving toward that practice. They are not negotiable within the framework; an organization that rejects them is not a RIVER candidate, and the framework will not help it.

01 / Release is progressive and controlled. A release is not a binary transition from unshipped to shipped. It is a staged exposure of functionality across cohorts over time, with explicit control at each stage. The question is not whether a release happened but where it has reached and what it has produced there.

02 / Targeting is the unit of control. Release is to a segment, a percentage, a tier, a geography, a customer class, an experimental arm. "Production" is a destination; it is not the unit at which release decisions are made. Targeting rules are explicit, versioned, and recoverable.

03 / Reversibility is native. A release can be pulled back in seconds without a redeploy. This changes the definition of release failure. In the DORA era, failure meant a bad deploy that had to be rolled back; in the RIVER era, failure means a release that degraded a metric and had to be reversed.

The cost of trying a release is approximately zero; the cost of being wrong is the time spent in the degraded state.

04 / Experimentation is intrinsic to release. The same mechanism that exposes a feature can measure whether it worked. Experimentation is not a separate discipline layered on top of release; it is a capability of release itself. This collapses the distance between shipping and learning, and it is the mechanism by which RIVER becomes an evolution reporting framework rather than a snapshot measurement framework.

05 / Release is a product decision, not just an engineering one. Targeting rules, success criteria, and cohort definitions are co-owned by product and engineering, with operations accountable for the guardrails. Release is where the three functions must agree on what the work was for. RIVER formalizes that agreement into a declared artifact.

An organization that holds these five commitments, even imperfectly, is operating in the environment RIVER is built for. An organization that rejects any of them is in a different operating regime, and the framework will not produce the outcomes it is designed to produce.

VIII / Release delta

The unit of analysis.

The central abstraction of RIVER is the release delta. A release delta is a declared, structured artifact created before a release begins. It answers five questions: what type of release this is; what hypothesis is being tested; what success signal, at what magnitude, over what window, and against what baseline, will tell us the hypothesis held; what cohort the release is for and how exposure will progress through it; and by when we expect to know. A release delta is not a wish or a roadmap line; it is a commitment, recorded before the work is exposed, to evaluate the work against criteria that cannot be revised after the fact.

The release delta is the unit of analysis in RIVER. Every metric is scoped to a delta. Team-level, product-line-level, and organization-level rollups aggregate over deltas, not over deploys, flags, tickets, or story points. This scoping is the structural feature that makes the framework coherent: without it, RIVER would be a pile of metrics about feature flags, and flags are not a meaningful unit of business analysis.

The act of declaring a delta before shipping is coercive, and the coercion is the point. Declaration forces three questions that most release ceremonies today let teams avoid: what specifically do we believe will happen; how will we know; by when. Most teams can answer the first question in soft language after a release; few can answer the second and third in any language before one. RIVER's measurement value and its organizational-discipline value come from the same act. Declaration is what produces the measurement; the discipline that declaration requires is what produces the operating change.

The baseline, declared.

A success signal is itself a structured component. It declares the metric, the direction and magnitude of expected movement, the time window, and the baseline the movement is judged against. The baseline is the component teams most often leave implicit, and it is the one a declaration cannot survive without: a signal that activation rises by six points means nothing until it names what the six points are measured from. Every delta declares its baseline at declaration time.

RIVER is design-aware and non-prescriptive about how the comparison is made. Evaluation methods span a spectrum, from randomized experimental arms through guarded progressive rollouts to purely observational attribution, and stronger designs produce stronger attribution. The framework names the spectrum; it does not mandate a position on it. Not every release can carry a randomized design, and a framework that demanded one would exclude most of the legitimate release activity it exists to measure. What the framework requires is honesty about where on the spectrum a given delta sits, and the declared baseline is what records it.

Four baseline types cover the spectrum, ordered by the strength of attribution they support. Like the named metrics, they are version-one candidates, subject to empirical refinement through the research program.

BASELINE TYPE	THE MOVEMENT IS JUDGED AGAINST	ATTRIBUTION
Concurrent Comparison	A group not exposed during the same window: a holdout, an experimental arm, a matched cohort.	Strongest. The comparison shares the delta's time window, isolating the release from concurrent change.
Forecast Comparison	A declared projection of the metric in the absence of the release. The forecast is recorded at declaration; a projection produced after the fact is storytelling, not a baseline.	Strong when the forecast is honest. The projection itself is the contestable element.
Historical Comparison	The metric's own past: a trailing average, a prior period, a pre/post split.	Weakest comparative attribution. Confounded by seasonality and by everything else that changed in the period; still legitimate RIVER practice, named as what it is.
Absolute Threshold	A fixed value with no comparison group: error rate stays below a ceiling, latency stays under a bound.	Not comparative. The native baseline for guardrail-driven releases and for most Risk Reduction and Platform/Enablement deltas.

The ordering carries the claim: attribution is only as strong as the baseline it is judged against. An organization's distribution across baseline types is a fact about its evidentiary practice, not a ranking of its teams, and it gives *Outcome Attribution Rate* a segmentation dimension the research program can calibrate. The same attachment rate means different things in an organization living at Concurrent Comparison and one living at Historical Comparison.

IX / Delta taxonomy

Six types, each measured against its own standard.

Not every release is supposed to move revenue, and a framework that collapses release performance into a single metric will distort the work it measures. RIVER types releases at declaration time into one of six categories, each with its own success criteria. Typing is not optional and is not decided after the fact. The type is part of the declared delta.

TYPE	WHAT IT PURSUES	SUCCESS LOOKS LIKE
Growth	Acquire, activate, or convert users.	Movement in funnel metrics within the exposed cohort: conversion rate, activation rate, lead velocity.
Retention / Engagement	Deepen usage among existing users.	Increased frequency, depth, or stickiness of use within the exposed cohort.
Monetization	Expand revenue per user, unlock new revenue, or convert free to paid.	Direct revenue movement within the exposed cohort.
Experience / Quality	Improve the existing experience without changing what the product does.	Satisfaction measures, support ticket reduction, or task completion rate improvements.
Platform / Enablement	Make future work faster, safer, or more reliable. The user is another engineer or team, not an end customer.	Downstream velocity gains, incident rate reductions, or adoption by internal teams.
Risk Reduction	Address compliance, security, resilience, or regulatory exposure.	Reduction in the specific risk being targeted, not growth or engagement.

The taxonomy does two important things. It makes room for legitimate engineering work that does not move customer-facing metrics, which is essential, because a framework that treats Platform/Enablement releases as failures because they do not move funnel metrics will distort the

work a serious engineering organization does. It also forces precision about what a release was for: the act of choosing a type is the act of naming what success would look like in a category commensurate with the work.

X / Adoption, layered

Three signals, not one.

Adoption in RIVER is three signals, not one, because the word “adoption” in common use collapses three distinct phenomena that have different shapes, different time horizons, and different levels of evidentiary value.

First-use *(fast, weak signal)*. Has an exposed user touched the feature at all? First-use is a fast signal, resolving in hours to days, and a weak one on its own. It is most useful as an early indicator of discoverability problems: if first-use does not occur, the feature is not findable, and no downstream measurement will be meaningful. First-use is not evidence of value; it is evidence of visibility.

Sustained-use *(workflow signal)*. Is the feature used repeatedly, over time, by exposed users who have already touched it once? Sustained-use is a medium-horizon signal, resolving in weeks. It tells us the feature has found a place in real workflows. Sustained-use is a stronger signal than first-use but still incomplete as evidence of value: a feature can be used repeatedly without producing the outcome it was built to produce.

Value-realization *(primary adoption metric)*. Has the user completed the action the feature was built to enable, as defined by the declared success signal for the delta? Value-realization is a slow, high-fidelity signal, resolving in weeks to months. It is the primary RIVER adoption metric. A feature with high first-use and sustained-use but low value-realization is being used without producing the outcome it was built for, and that pattern is information the framework needs to surface.

All three layers are measured within the exposed cohort, not the total user base; the total user base is not a meaningful baseline in a progressive-release world. The definition of “user” depends on the delta type: for a customer-facing release, the user is the end customer; for a Platform/Enablement release, the user is another engineer or another team. And the phrase “fully adopted” is not framework language. Most features never reach saturation of their target cohort, and that is often fine. The framework concept is **target adoption**: adoption sufficient to validate the declared delta. Precision here matters; the common-use phrase sets an expectation the framework does not hold.

XI / Metric families

Seven families, mapped to the value chain.

RIVER organizes its metrics into seven families that correspond to stages of the value chain. The families are intentionally asymmetric with DORA's four; forcing a four-to-four parallel would flatter DORA at the cost of honesty. The specific metrics within each family named here are version-one candidates, subject to empirical refinement through the research program.

FAMILY	WHAT IT MEASURES	REPRESENTATIVE METRIC
Exposure	How long deployed code sits unreleased, and how quickly it moves from deploy to first user exposure. Exposure metrics reveal the gap that delivery metrics cannot see.	*Feature Dark Time*
Cohort Progression	How exposure moves through its target cohort, and whether cohorts advance smoothly or are blocked by recurring guardrail triggers.	*Rollout Velocity*, *Graduation Smoothness*
Reversal	How often releases are reversed through kill-switches, flag-offs, or targeting rollbacks. RIVER's change-failure analog at the release layer rather than the deploy layer; it registers a failure mode the four delivery metrics do not.	*Release Reversal Rate*
Guardrail	How often automated guardrails paused or reversed a release in response to a monitored metric. Measures the organization's ability to respond to release-level signals without human intervention, which is a maturity marker.	*Guarded Release Activation Rate*
Experiment-Linked	How systematically releases are tied to declared hypotheses and outcome attribution. These metrics measure the adoption of RIVER itself.	*Hypothesis Attachment Rate*, *Outcome Attribution Rate*
Adoption	First-use, sustained-use, and value-realization within the exposed cohort. Adoption metrics are defined per-delta, because what adoption means differs by delta type.	*Value-Realization Rate*
Impact	Whether adopted features moved the business metrics they were built to move. Impact metrics are the framework's terminal measurement. Outcome Realization Rate is the	*Release-to-KPI Lead Time*, *Outcome Realization Rate*

FAMILY	WHAT IT MEASURES	REPRESENTATIVE METRIC
	metric most directly responsive to the central business question.	

XII / The maturity ladder

Five levels. Teams climb into them.

RIVER has a dual identity. It is a measurement framework, and it is a maturity practice. Teams do not adopt RIVER in a single act; they climb into it over quarters or years. This matters because DORA's most powerful rhetorical device was its maturity tiering: elite, high, medium, and low gave teams a self-locating language and an aspirational ladder. RIVER needs the same structure, with the same practical function.

The five levels are defined here in outline. Empirical calibration of the transitions between levels is part of the research program.

#	LEVEL	TAG	DESCRIPTION
1	Deploy	Delivery hygiene	The team ships reliably on a DORA foundation. Deployment frequency, lead time for changes, change failure rate, and time to restore are tracked and trending in the right direction. Deploy and release may still happen together. Level 1 is the foundation of a good release practice, not a deficient state to be escaped; teams that have not solved delivery hygiene cannot productively adopt release-level practice on top of it.
2	Control	Deploy ≠ release	Deploy and release are separate events in the team's practice. Rollouts are progressive. Cohort targeting is used. A release can be reversed in seconds without a redeploy. Release-layer metrics exist, but they are not yet consistent across teams or across release types within a team. Level 2 is where most organizations with feature-flag platforms land without deliberate work beyond the tool installation.
3	Declare	Delta, ahead of ship	Before a release ships, the team states what it expects to happen: hypothesis, success signal, target cohort, time horizon. Some releases are measured against

#	LEVEL	TAG	DESCRIPTION
			what was declared. The discipline is emerging but is not yet uniform across the team or the organization. The transition from Control to Declare is the hardest transition in the ladder, because it is where measurement stops being instrumentation and starts being ceremony.
4	Prove	Systematic attribution	Outcome attribution is systematic. Most releases carry a declared success signal and are evaluated against it. The organization builds a durable record of which deltas realized and which did not, and can report Outcome Realization Rate in planning reviews and skip-levels with the evidence to back it. Not every hypothesis will hold, and the fact that the record includes unrealized deltas is a feature, not a failure: the record's credibility depends on its honesty.
5	Learn	The compounding loop	Evidence from realized and unrealized deltas feeds the next planning cycle. Predictions about future releases sharpen over time, not just the measurements of past ones. This is where the Evolution in RIVER lives. Level 5 is rare in the current state of the industry; it is the destination the framework

#	LEVEL	TAG	DESCRIPTION
			describes, not a common condition.

A note on the ladder's function: it is not a ranking. It is a language. An organization that can say "we are at Control on most teams and Declare on two pilot teams" has a more tractable conversation about what to invest in next than an organization that can only say "we are working on our metrics." The ladder's credibility, like the metric families', depends on empirical calibration through research.

XIII / Portability

Philosophy-coupled. Tool-neutral.

RIVER is coupled to a specific philosophy of release (the five commitments in Section VII) and is deliberately neutral about the tools an organization uses to instrument it. Any platform, any combination of platforms, and any internal system that can produce the required data is a candidate for a RIVER implementation. The framework's credibility depends on this portability.

The reasoning is borrowed directly from DORA. DORA's four metrics did not depend on any specific CI system, deployment tool, or cloud platform. They were instrumentable on any stack that produced the relevant events. That tool-neutrality is what let DORA become industry vocabulary rather than a vendor's framework. A framework bound to a single tool ceases to be an industry framework and becomes a marketing asset; the loss of credibility is usually sudden and irreversible.

RIVER's philosophy-coupling, by contrast, is not negotiable. An organization that ships in single binary events from commit to all users cannot meaningfully implement RIVER, because the segments RIVER measures do not exist in that organization's practice. This is a feature. It defines the framework's scope honestly and prevents it from being claimed by organizations it cannot help.

The practical consequence is that some platforms make RIVER instrumentation materially easier than others: specifically, those that already sit at the deploy/release seam and generate exposure, cohort, and reversal data as a byproduct of normal operation. These platforms constitute reference substrate for RIVER. The framework's portability means it is not bound to any of them; its instrumentability means organizations using them will find adoption substantially less expensive.

XIV / What RIVER implies

The framework produces an operating change.

The most important claim RIVER makes is not about measurement. It is about operations. Adopting RIVER fully, through the Declare, Prove, and Learn levels of the ladder, forces a specific change in

how an organization works, and that change is the framework's primary benefit.

The change is specific and describable. Today, in most software organizations, product defines outcomes, engineering ships code, and operations reports on reliability, and each function measures itself in its own vocabulary. Objectives set at one end of the value chain are routinely lost in handoffs before the measurement at the other end. RIVER's release delta is a shared artifact that product, engineering, and operations co-own from declaration through evaluation. The hypothesis is shared. The success signal is shared. The cohort and horizon are shared. Because the artifact exists before the work begins and is evaluated after it completes, it survives the handoffs that currently lose the objective.

The framework is the vehicle. The operating change is the point.

The implication for organizational design is direct. RIVER-mature organizations do not operate as three adjacent functions that coordinate at boundaries; they operate as a single value-flow system whose component functions cohere around a shared artifact. This is not a restructuring. It is an operating discipline. The functional boundaries remain; what changes is what the functions agree they are accountable to.

The scope of this claim is large enough that it warrants explicit epistemic treatment, which follows in the next section.

XV / Epistemic status and the research program

What is asserted, what is observed, what is to be formalized.

Intellectual honesty requires naming the epistemic status of the framework's claims. RIVER makes three kinds of claim, and they are not all of the same type.

The descriptive claims about the separation of deploy and release, the boundary of the four delivery metrics, the posture of DORA's downstream research, the fractured state of the value stream in most organizations, and the absence of a shared per-release operating artifact are observable facts about the current state of practice. They do not require RIVER-specific research to validate; they are visible to any careful observer of the industry.

The structural claims about the framework itself (the delta taxonomy, the adoption layering, the metric families, the maturity ladder) are design decisions grounded in ten years of field observation across a large sample of software organizations. They are internally coherent and practitioner-grounded. They are not yet empirically formalized. Formalization is what the research program produces.

The operational claim that adopting RIVER produces an organizational operating change beyond measurement improvement is a thesis advanced from pattern recognition in practice. The author's confidence in the claim is high, based on the consistency of the pattern across the sample. The

claim has not yet been empirically tested through structured research. Testing it, and formalizing what the operating change looks like in organizations at different scales and maturity levels, is a central objective of the research program.

The research program is staged deliberately. Each phase corresponds to a different level of evidentiary claim the framework can defensibly make.

Phase 1. Qualitative grounding through structured interviews with practitioners at twenty to thirty organizations across size and industry. Primary subjects are engineering managers, product leaders, and SRE/operations leaders, with director-level and C-suite conversations where they sharpen the organizational picture. Produces the framework's first calibration.

Phase 2. A pilot quantitative survey that tests whether teams can self-assess against the framework and produces a first data snapshot.

Phase 3. An annual longitudinal report that establishes empirically calibrated maturity tiers and year-over-year trends, on the DORA model.

A note on evidentiary honesty: a framework that overstates its evidentiary basis is a framework that collapses under its first published benchmark. RIVER is positioned to earn the evidentiary basis it needs, in stages, on the timeline that earns it.

XVI / Purpose

What RIVER is for.

RIVER exists to give software organizations a rigorous, shared, and portable way to answer the central business question their current measurement practices cannot answer: **is the value we generate equivalent to, or greater than, the cost of our staff and platform?**

The framework is designed to make that answer possible at three distinct levels of use. At the team level, it gives engineering managers a defensible way to connect their team's work to outcomes they can speak to in skip-levels and planning reviews. At the organization level, it gives product, engineering, and operations leaders a shared vocabulary for describing what the organization is accountable for and how it is performing against that accountability. At the industry level, it gives the practice of software engineering a shared operating discipline for the segments of the value chain past deploy.

Those three levels are the framework's obligations. If RIVER succeeds at all three, it will have earned a place alongside DORA in the permanent vocabulary of the industry. If it succeeds at only the first two, it will have been a useful organizational tool for the teams that adopt it. If it fails at all three, it will have produced a measurement taxonomy that was intellectually honest and operationally inert, and the attempt will not have been wasted, because the question it tried to answer is not going to stop being asked.

The silence RIVER fills was getting louder before this framework was written, and it will get louder still.

Colophon and changelog

Colophon. This document is the canonical statement of RIVER. It is the substrate from which tailored artifacts (proposals, case studies, slide decks, interview instruments, survey designs) are derived. Tailored artifacts may compress, reorder, or omit sections to suit their audience; they should not contradict this document. When they do, this document is the reference.

Version 0.6 corresponds to the pre-research state of the framework. Subsequent versions will be published as the research program produces formalization. Substantive changes between versions are tracked in the changelog below. Terminological changes will not be silent.

Changelog

V0.6 / June 2026. Expanded the success signal: a declared success signal now names its metric, direction, magnitude, time window, and baseline. The five questions of the release delta are unchanged; the baseline lives inside the success signal, not as a sixth question. Added the baseline taxonomy to Section VIII: four baseline types (Concurrent Comparison, Forecast Comparison, Historical Comparison, Absolute Threshold), ordered by attribution strength, as version-one candidates subject to empirical refinement. Stated the framework's posture on experimental design: design-aware and non-prescriptive. The framework names the spectrum of evaluation methods and asserts that stronger designs produce stronger attribution; it does not require any particular design. "Holdout" is ratified as an instance of Concurrent Comparison, with the formal definition deferred to the Glossary.

V0.5 / June 2026. Renamed the central artifact: release intent is now release delta, the intent taxonomy is now the delta taxonomy, and the Level 3 tag is now Delta, ahead of ship. Retired the one-liner "DORA measures the cost. RIVER measures the value" without replacement; the definition now stands on its own terms, and the relationship to DORA is stated once, in Section V, on the posture axis. Added Section IV, The acceleration, promoting the AI and agent-context claim from authorial motivation on the Standing page to a framework-level claim. Rewrote Section V to account for DORA's downstream evolution across its 2024 and 2025 publications; "encompasses" is now scoped to the four DORA metrics rather than the program. Updated Sections I, II, VI, and XVI to retire the claim that no framework describes the post-deploy segments; the surviving claim is that no framework operates them per release. Added the value vocabulary disambiguation, with formal definitions deferred to the Glossary. Editorial pass for house punctuation throughout.

V0.4 / April 2026. Prior canonical statement. Baseline entry, recorded retroactively.