

New DevX Roadmap: A Blueprint for Dynamic Wayfinding

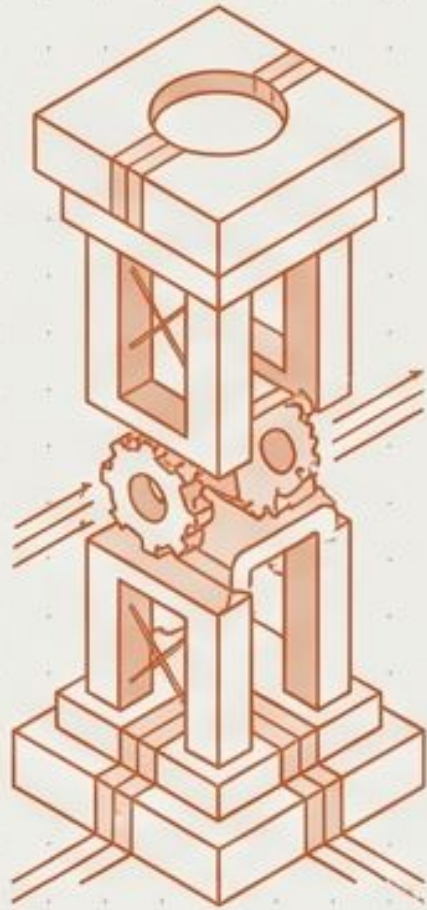
How engineering leaders navigate the shift to agentic development.

The technology frontier in agentic development moves faster than any plan; what was experimental six months ago is table stakes today.

The question is not "what is the plan?" but "how do we make good decisions as the landscape moves?"

The Three Decision Principles

Elevate the Next Constraint



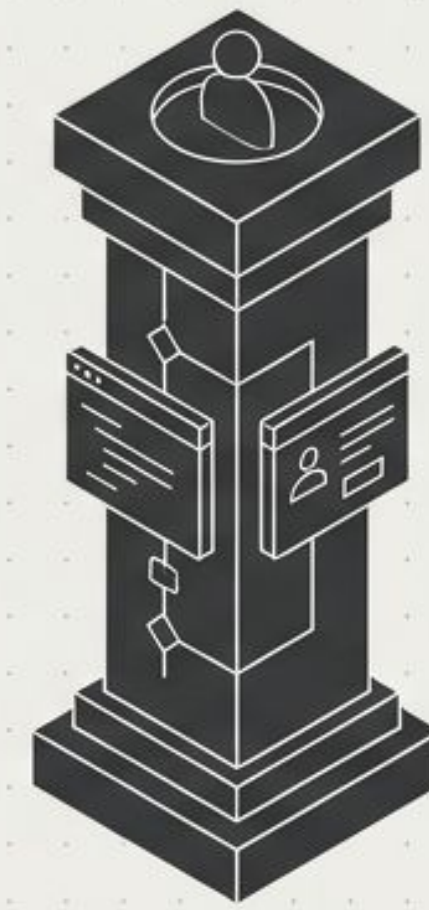
Identify the binding bottleneck, design the intervention, and move to the next. Do not attempt to optimise the entire system simultaneously.

Everything as Code



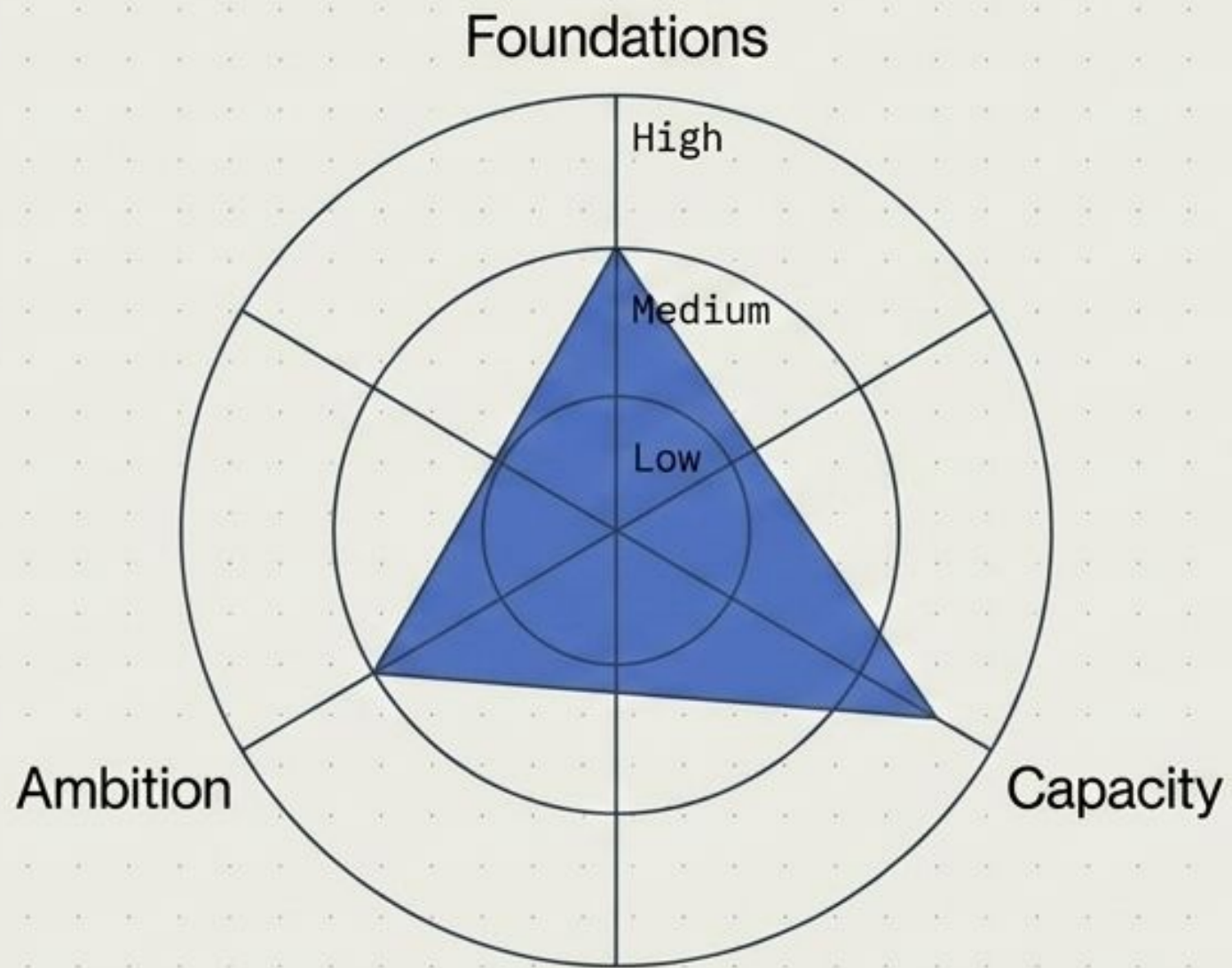
Codify what matters so it is versioned, testable, and governable. Agents require governed inputs; ad hoc knowledge does not scale.

Humans Above The Loop



Redesign verification rather than adding more reviewers. Humans belong governing verification systems and shaping intent, not reviewing every line of code.

Assessing Organisational Readiness



Factor 1: Engineering Foundations

Are Git workflows mature? Is CI/CD dependable? Are tests meaningful? Low foundations require fixing the baseline before widening agent use.

Factor 2: Capacity for Change

Does the organisation learn from experiments and codify what works, or do local experiments rarely become shared practice?

Factor 3: Transformation Ambition

Is leadership willing to tolerate unevenness and protect experimentation? Lower ambition requires steadier progress with clearer proof.

Warning: No amount of AI tooling compensates for fragile pipelines and missing tests.

Calibrating the Implementation Posture

A single team cannot oscillate between postures; standardise per team and reassess quarterly.

Present-Forward

Approach: **Build infrastructure ahead of the constraint.**

Risk Management: **Sequencing** (Build before need). **Lower risk, slower start, higher predictability.**

Best Fit: Legacy estates, regulated environments, core platform teams.

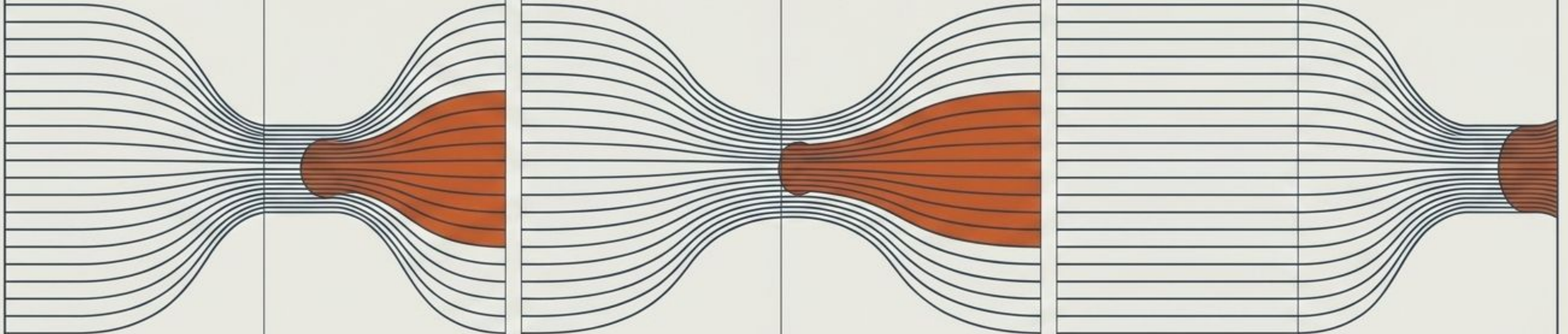
Future-Back

Approach: **Build infrastructure when the constraint demands it.**

Risk Management: **Scoping** (Start with a Minimum Viable Business to contain the blast radius). **Faster learning, rougher edges.**

Best Fit: **Startups**, greenfield teams, protected product areas.

The Migration of the Binding Constraint



Stage 1: Observed and Understood

Constraint: Supervision

AI tools are in use, but every agent output queues behind a manual human reviewer.

Stage 2: Engineered Validation

Constraint: Validation

Context and guard rails reduce routine drag, replacing manual supervision with engineered checks.

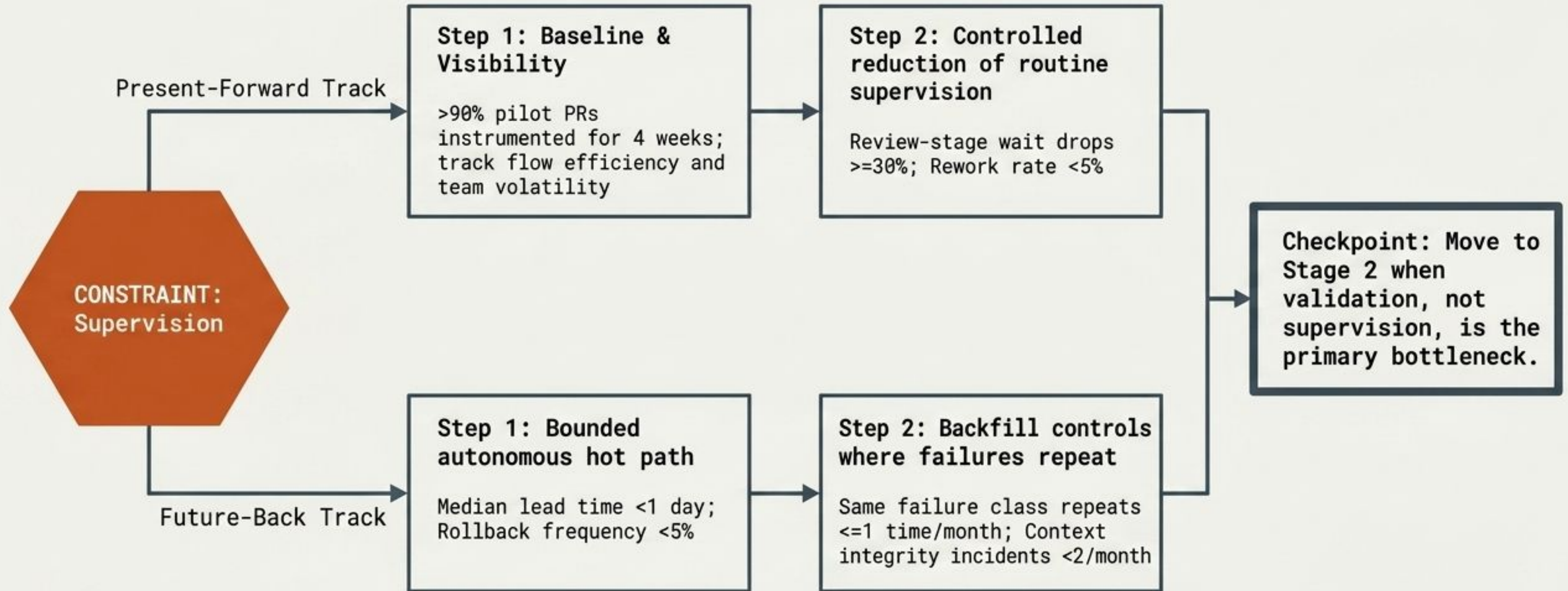
Stage 3: Risk-Governed Orchestration

Constraint: Intent Quality

Validation is automated; human focus shifts entirely to upstream risk-based oversight and intent specification.

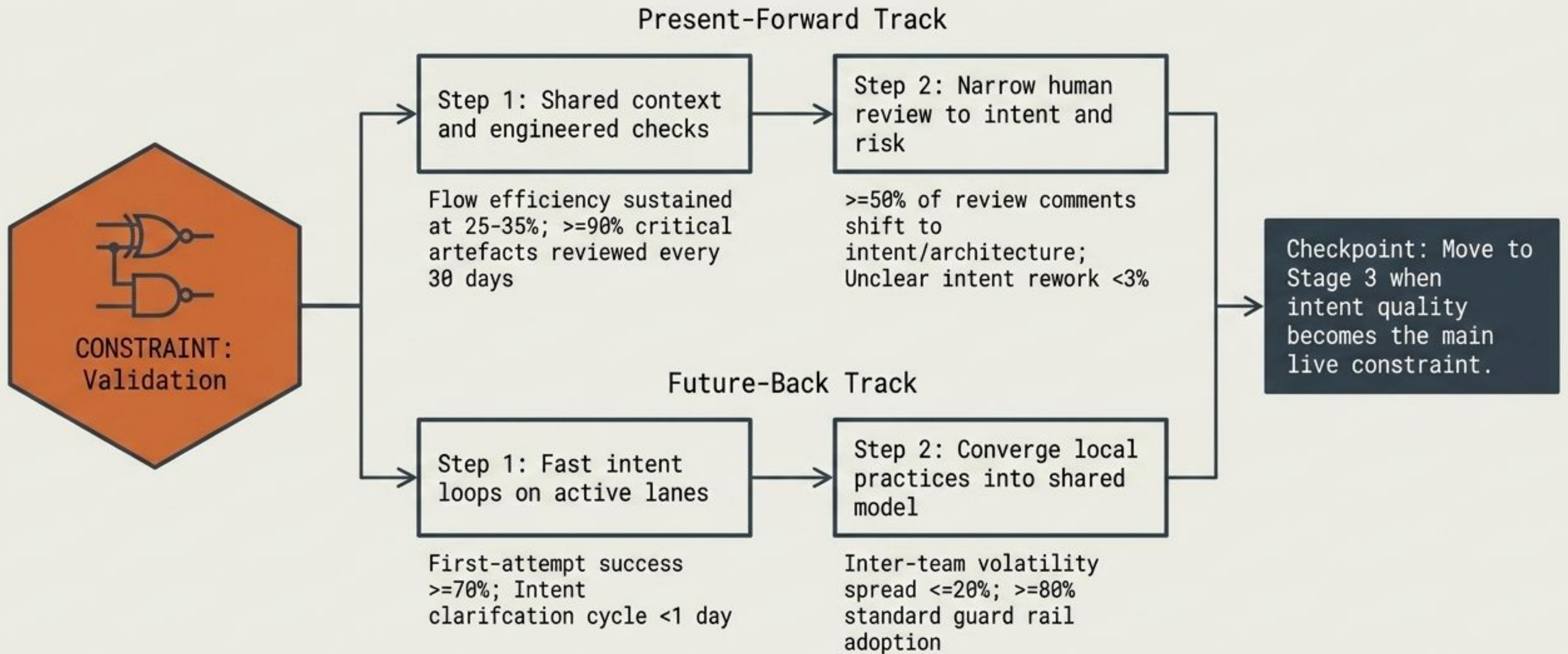
Stage 1: Overcoming the Supervision Bottleneck

Goal: Establish real usage, visible baselines, and a credible next investment.



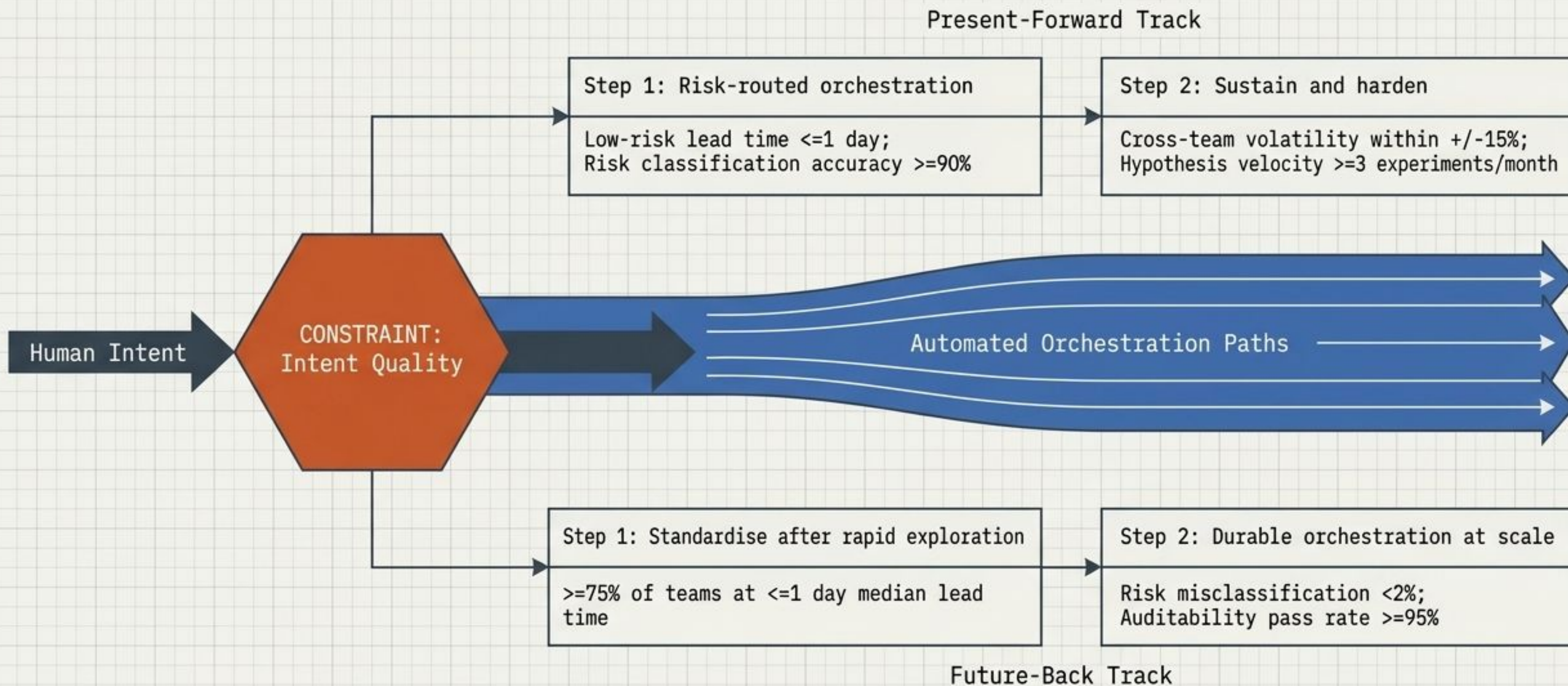
Stage 2: Engineered Validation

Goal: Make agent success repeatable by engineering the environment around the work.



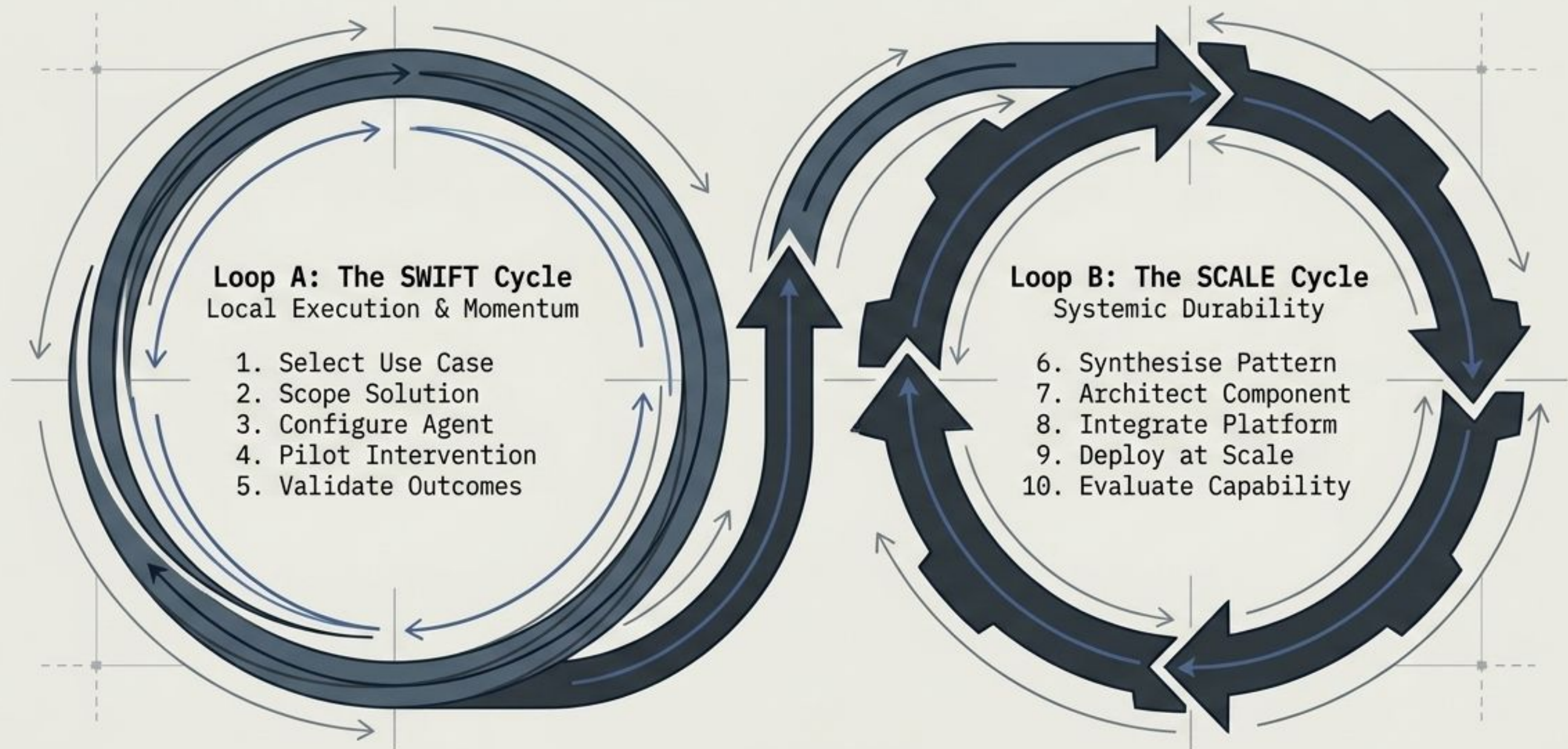
Stage 3: Risk-Governed Orchestration

Goal: Let suitable work move through orchestrated, risk-governed paths with humans above the loop.



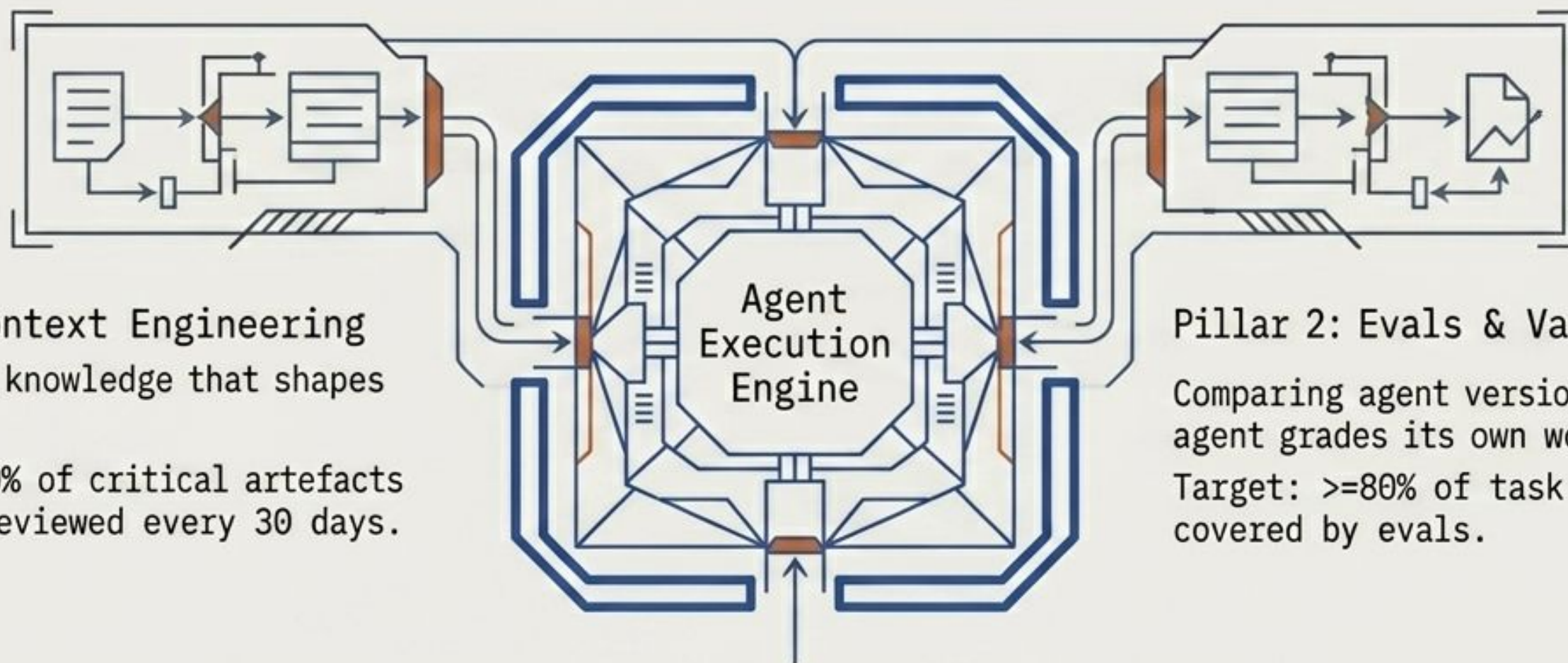
The Operating Loop: Elevating the Baseline

Key Insight: Extracting a pattern from the SWIFT cycle permanently elevates the baseline for the next cycle.



Building New Capabilities: AgentOps

Core Question: Can agents do good work? AgentOps extends the engineering tradition to make agents buildable, testable, and deployable.



Pillar 1: Context Engineering

Curating the knowledge that shapes behaviour.

Target: $\geq 90\%$ of critical artefacts reviewed every 30 days.

Pillar 2: Evals & Validation Gates

Comparing agent versions to ensure no agent grades its own work.

Target: $\geq 80\%$ of task types covered by evals.

Pillar 3: Observability & Failure Modes

Detecting metacognitive failures (context flooding, premature abandonment).

Target: Same failure class repeats ≤ 1 time/month; Recovery from degradation < 4 hours.

Building New Capabilities: Agentic Service Management (AGSM)

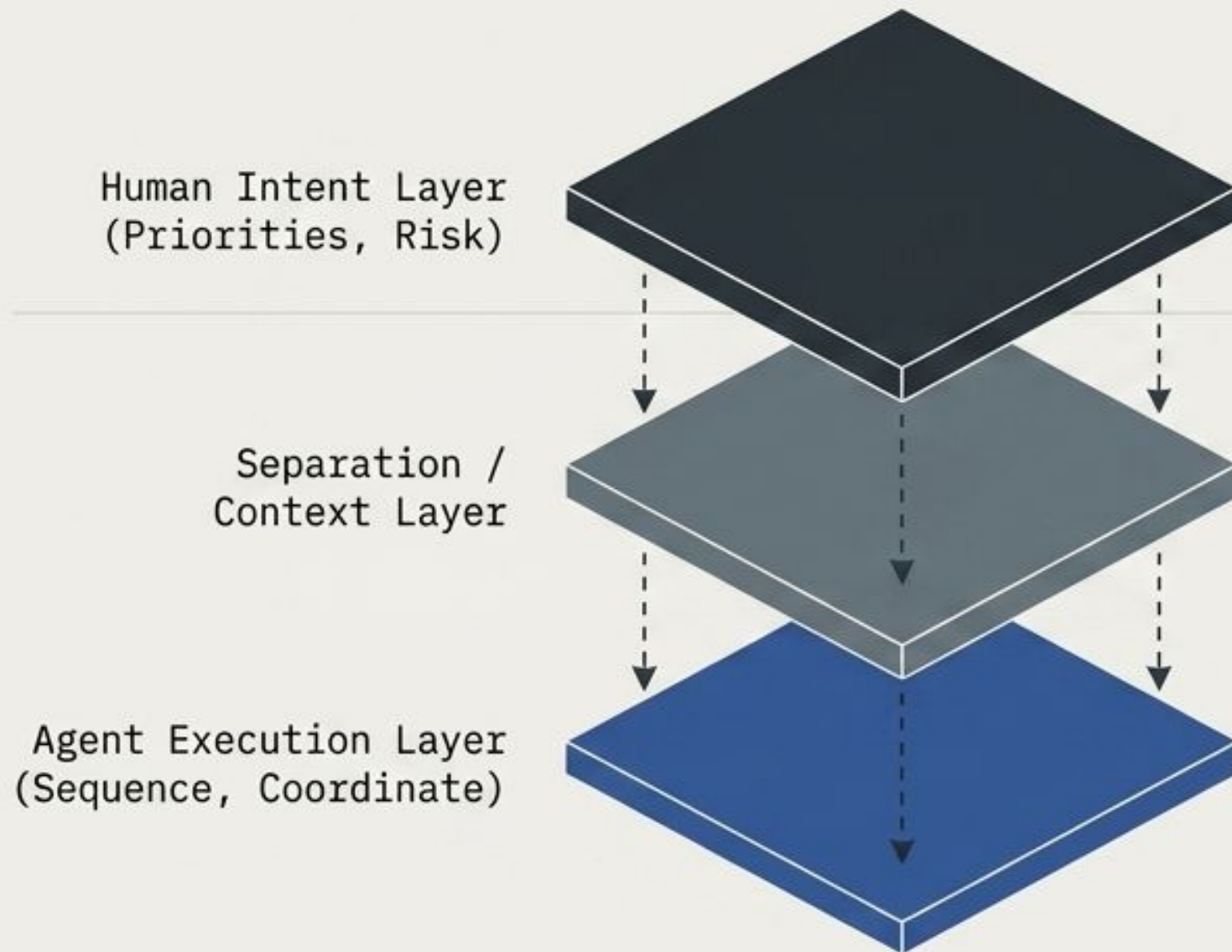
Core Question: Can humans defend the decisions? AGSM extends ITIL/service management to make agents governable and trustworthy.

<p>Agent Lifecycle Management</p>	<p>Managing the Comprehension Gap</p>
<p>Registries tracking identity, ownership, and version.</p> <p>Target: 100% of production agents registered.</p>	<p>Monitoring the risk that agent throughput outpaces human understanding.</p> <p>Track very large PRs, shallow reviews, and reversions.</p>
<p>Operationalising Trust</p>	<p>Auditable Compliance</p>
<p>Governance operates on risk classification rules, not individual per-change approvals.</p> <p>Target: $\geq 90\%$ risk classification accuracy.</p>	<p>Ensuring the evidence chain is auditable without manual reconstruction.</p> <p>Target: $\geq 95\%$ compliance pass rate.</p>



Replacing Agile Ceremonies with Flow

The Shift: Traditional ceremonies assumed planning and execution lived in the same heads. Agentic development demands separating the intent layer from the execution layer.



Stage 1: Lighten

Time in ceremony drops $\geq 25\%$.
Use stand-ups to learn about agent failure modes, not just status updates.

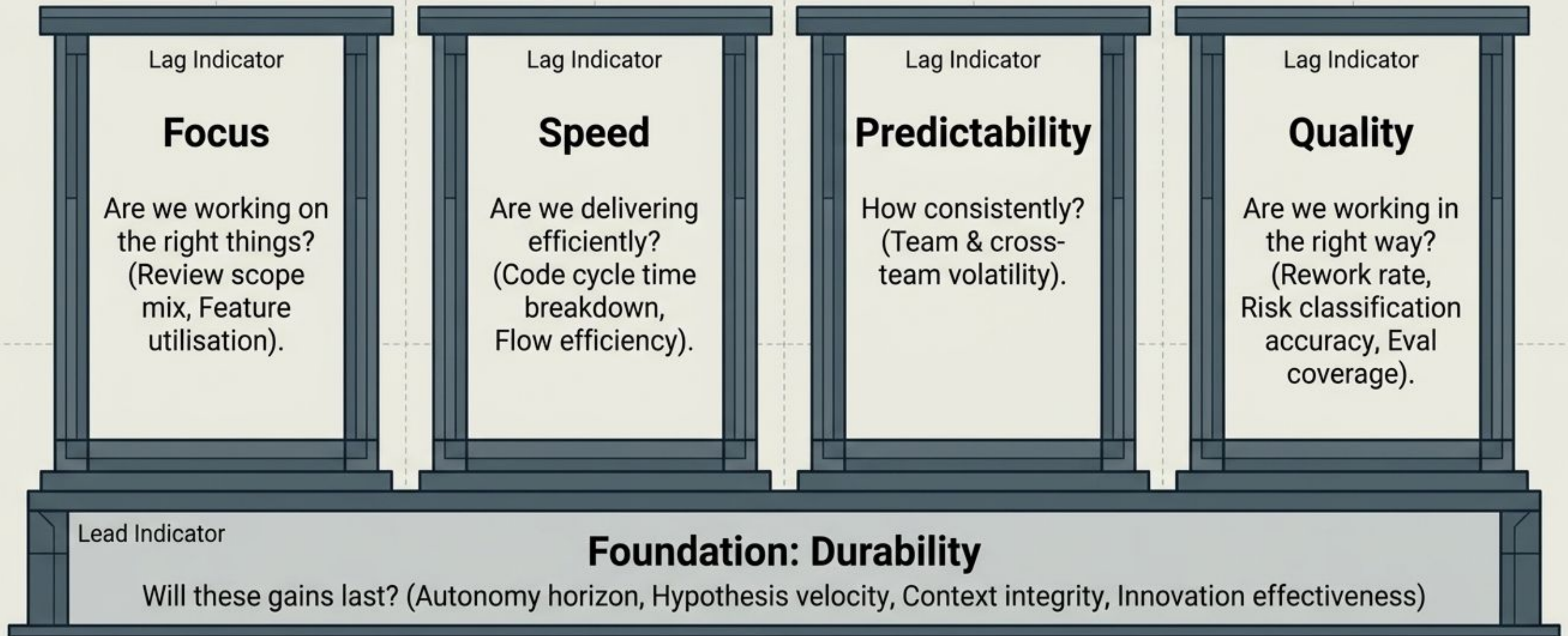
Stage 2: Separate

Humans define intent; agents self-organise execution from context.
Target: Intent-to-execution latency < 1 business day.

Stage 3: Replace

Operating Loop cadence replaces sprints.
Constraint reviews and strategic pivots replace retrospectives.
Target: ≥ 3 decision-grade experiments/month.




The Practical Metric Stack

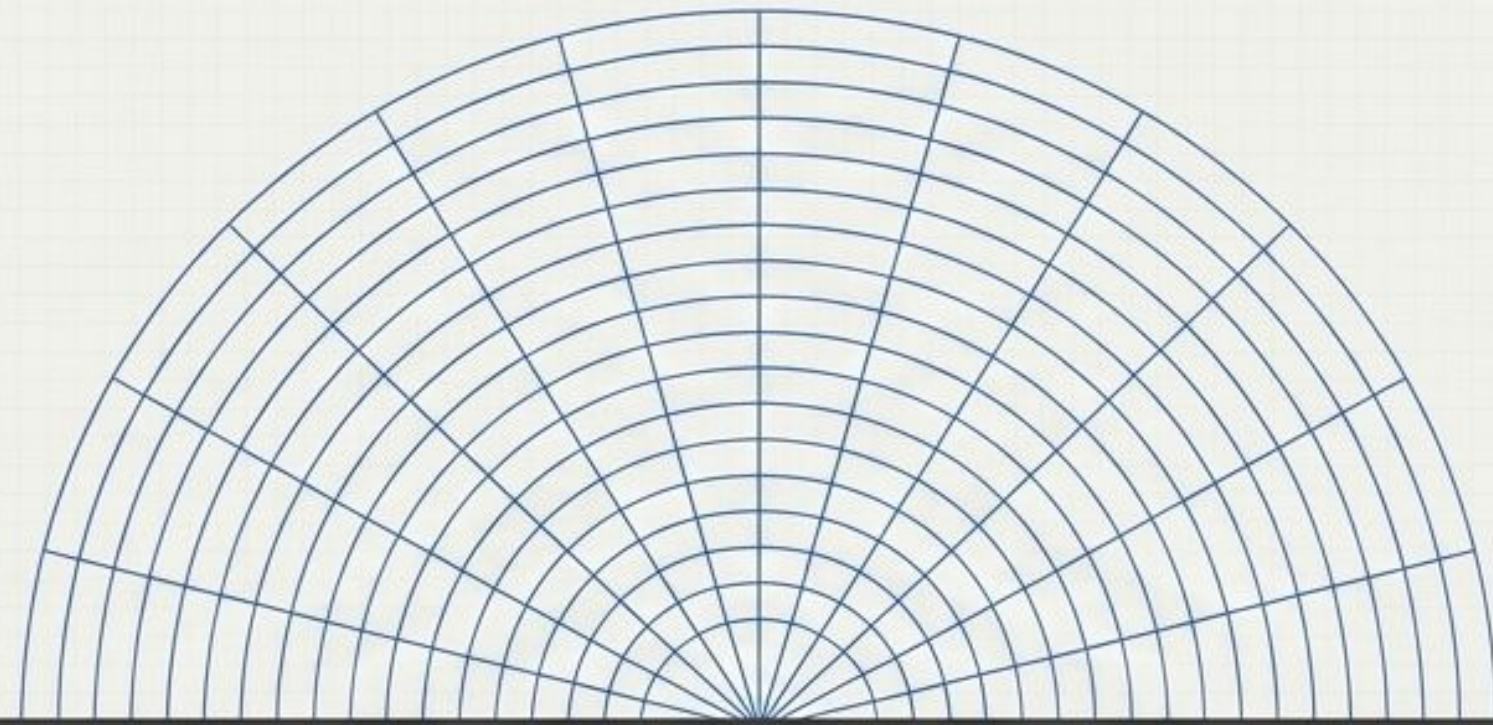


Critical Warning: Productivity metrics can look healthy while durability metrics quietly deteriorate. Measure both, or mistake a short-term spike for a sustainable trend.

Avoiding Measurement Traps & System Warnings

The Traps: Avoid Activity as Success (Commit/PR counts), Lag-Only Steering (Quarterly lead time without stage breakdowns), and Ceremony Dogma (Story-point velocity).

	Symptoms	Responses
	Symptom: Individual productivity rises, but organisational throughput stays flat. →	Response: The bottleneck migrated. Find the new downstream queue; do not celebrate local speed in isolation.
	Symptom: PR backlogs grow despite faster code generation. →	Response: Review is the constraint. Strengthen agent review and reduce PR size.
	Symptom: The Comprehension Gap widens (shallow reviews, high reversions). →	Response: Agent throughput is outpacing human understanding. Narrow autonomy scope temporarily and improve traceability.



NEWLY ELEVATED BASELINE

The Next Constraint

The organisations that move best through this journey will not be the ones that generate the most plans. They will be the ones that learn fastest where work is waiting, fix the next bottleneck before the previous fix wears off, and build the engineering and governance capabilities that make the operating model durable as autonomy rises.

Continually recalibrate as the constraints migrate. So what do you do next? **What is your next most pressing constraint?**