

WHITE PAPER OCTOBER 2026

The Adaptive Enterprise Loop

Connecting strategic direction with value delivery and system learning

Zayne Nair

www.zaynenair.com

Organisations become more adaptive when strategic choices, cross-team coordination, delivery, and system learning work together.

This paper explains how to connect those disciplines, expose constraints, and improve the operating system through bounded experiments. It is intended for leaders responsible for product, technology, delivery, and organizational change.

The practical recommendation

Start with one strategically meaningful flow of value. Make work and waiting visible, test one intervention, and use outcome evidence to decide what to change next.

A synthesis of Flight Levels | OKRs | Team Topologies | Systems thinking

Executive summary

Faster teams do not automatically create a more responsive business. Work can still wait for portfolio decisions, cross-team handoffs, shared services, and approval gates. Enterprise agility depends on how these parts of the organisation work together.

The Adaptive Enterprise Loop connects Strategic Intent, Flow Coordination, and Value Delivery through a discipline of System Learning. Adaptive Agility is the capability that emerges when those elements reinforce each other. Leaders set outcomes, priorities, guardrails, and review cadences; teams and coordinating groups use evidence to improve how value moves through the system.

The model draws on Flight Levels, Objectives and Key Results, Team Topologies, and systems thinking. Its contribution is an integrated operating logic that can coexist with existing delivery methods. It is a conceptual synthesis, rather than an independently validated framework or a claim that adoption alone produces better performance.

The model at a glance

Element	Core question	Contribution
Strategic Intent	Are we doing the right things?	Outcome hypotheses and investment choices
Flow Coordination	Can work move across boundaries?	Capacity visibility and fewer dependencies
Value Delivery	Are changes creating value?	Small increments and customer feedback
System Learning	Where is the system constrained?	Experiments and adjustments across all levels
Adaptive Agility	Can we adapt without chaos or waste?	The capability produced by the whole system

Begin with one value stream or customer journey that matters strategically and has visible delivery friction. Establish a baseline, identify the most important constraint, and run a bounded experiment. Extend the approach only when evidence supports doing so.

Reading guide

The problem and operating model establish the rationale. The next sections explain each element, followed by an illustrative example, a 90-day starting plan, measures, and source references.

The problem this model addresses

An organisation can adopt Scrum, Kanban, product discovery, DevOps, and continuous delivery while remaining slow to respond to customers. These practices may improve work inside teams, yet the wider journey from an opportunity to customer impact can remain fragmented.

A product team might complete a change quickly and then wait for an environment, a security decision, a data dependency, or capacity in another team. Each function can report healthy utilisation while the customer sees little progress. Local performance measures can hide these delays because they begin only when a team accepts the work and stop when that team finishes its part.

The implication is that enterprise responsiveness must be examined across the whole flow. Improving one team is useful when it relieves a meaningful constraint. When the constraint sits elsewhere, asking that team to work faster may simply increase the queue in front of the next bottleneck.

Connecting different perspectives

Flight Levels distinguishes strategy at Flight Level 3, end-to-end coordination at Flight Level 2, and operational delivery at Flight Level 1. These are perspectives on work, rather than a requirement to mirror the management hierarchy. Strategic choices shape coordinated work, while operational and coordination insights inform strategy. [1]

The Adaptive Enterprise Loop makes the learning relationship explicit. It asks whether decisions, interactions, team boundaries, technology, and measures are helping value reach customers. Evidence from delivery can change a strategic assumption; a recurring dependency can prompt a platform improvement or a clearer ownership boundary.

A deliberate approach to change

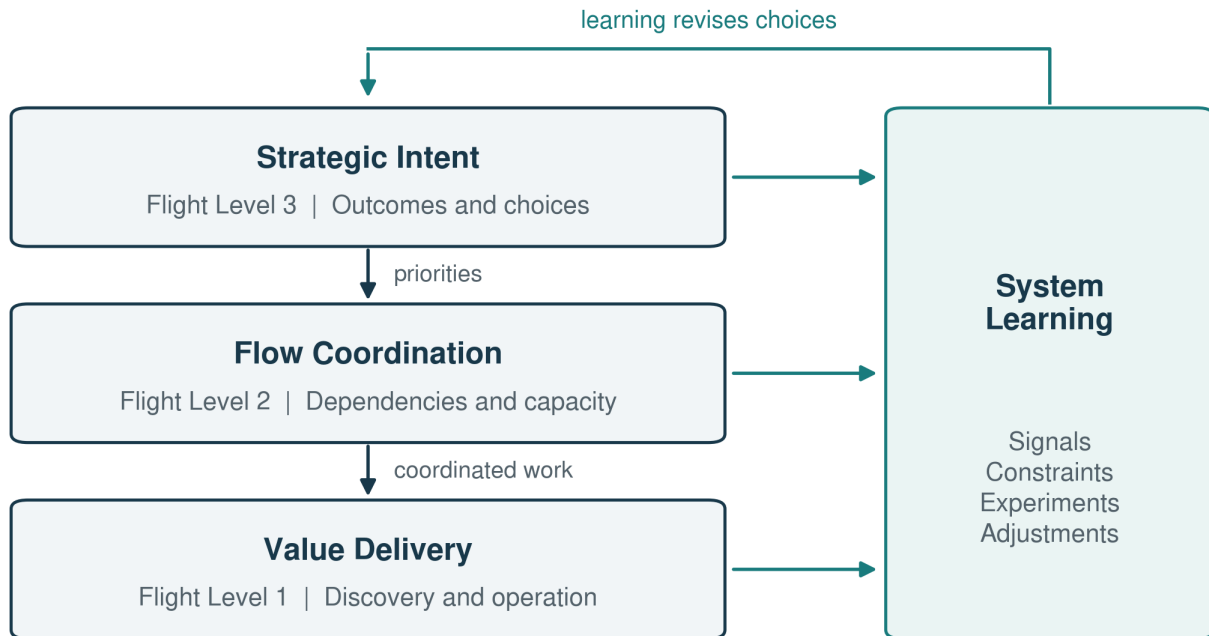
Leadership should establish purpose, constraints, and the evidence required to assess progress. The detailed design of the operating system then evolves through experience. This paper calls that approach structured emergence: direction is deliberate, while teams and coordinating groups test how best to achieve it.

This requires real decision authority. A coordination view will have limited effect if nobody can stop work, resolve conflicting priorities, or change a recurring approval rule. Leaders remain accountable for trade-offs that exceed team authority, including investment, risk, and ownership across the organisation.

The goal is an ongoing capacity to change while delivering value. A transformation program may support that goal, but completion of a program is not evidence that the capability exists.

How the operating model connects

Three operating perspectives connect direction to delivery. System Learning spans all three, examining both customer outcomes and the conditions that produced them. The diagram shows the main relationships; it does not prescribe a reporting structure or a fixed sequence of meetings.



Adaptive Agility is the capability produced by the connected system

Figure 1 The connected operating model. Evidence from each perspective informs system learning and revises strategic choices. Adaptive agility is the resulting capability.

Direction flows down and evidence flows back

Strategic Intent expresses the outcomes and trade-offs that matter. Flow Coordination translates those choices into work that can move across teams within available capacity. Value Delivery turns that work into changes that customers can use, and that teams can observe in operation.

System Learning examines results from every perspective. It can expose a weak outcome hypothesis, a recurring dependency, an overloaded team, or a rule that creates avoidable waiting. The response may be a delivery improvement, a change in coordination, or a revised investment decision.

Feedback should travel when it matters. A planned review cadence provides discipline, but significant customer, operational, or risk evidence should be able to trigger an earlier decision.

Strategic Intent

Are we doing the right things?

Strategic Intent makes explicit choices about where to invest, which outcomes matter, and what work should start, continue, or stop. It provides enough context for teams to make local decisions without turning strategy into a detailed project plan.

Flight Level 3 connects longer-term direction with nearer-term priorities and evaluates progress across the levels. OKRs can provide a useful language for that intent. Google’s guidance distinguishes measurable results from a shared task list, while Google’s OKR Playbook emphasises outcomes rather than activities. [1, 2, 3]

Express the intended change

Output commitment	Outcome to investigate
Launch a new customer portal	Increase successful self-service completion in priority journeys
Migrate five applications	Reduce recovery time and operational risk for critical services
Deliver the data platform	Reduce the time teams need to access trusted customer data

An outcome statement becomes useful when it specifies the affected customers or services, a baseline, a target, and a time horizon. The examples above describe the direction; they still need local evidence and agreed measures before becoming complete key results.

Make the hypothesis and trade-offs visible

Each initiative should explain why the proposed change is expected to influence the outcome. A portal may improve self-service only if customers can find it, complete the relevant task, and trust the result. Recording that hypothesis helps teams challenge the proposed solution without losing sight of the strategic purpose.

Priorities must also reflect capacity. Starting every worthwhile initiative creates competition for the same people, services, and decision-makers. Leaders should make the work being deferred or stopped visible alongside the work they choose to fund.

Use evidence to revise direction

Strategy needs enough stability to guide decisions and enough flexibility to respond to meaningful evidence. Review outcome movement, customer signals, operational risk, and flow constraints together. Revise an initiative when its assumptions fail, or a better opportunity becomes clear; avoid changing direction merely because a new request is louder.

The feedback loop connects market signals, strategic priorities, outcome evidence, and reprioritisation. Leadership owns the destination and guardrails, while the organisation learns which route is effective.

Flow Coordination

Are we coordinating to deliver outcomes?

Flow Coordination connects strategic priorities with the work of teams, products, functions, and services. Its purpose is to make initiatives, dependencies, decision points, capacity, and waiting visible. Flight Level 2 provides this end-to-end coordination perspective. [1]

The view should help people decide what can move next and what must be resolved. A board used only to collect status updates adds little. Repeated handoffs are a signal to examine ownership, interfaces, and internal services, rather than simply schedule more coordination.

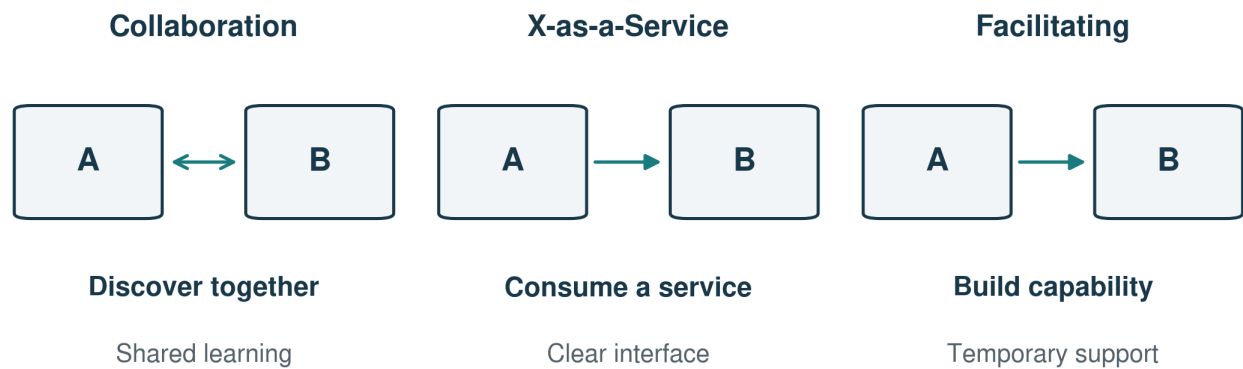


Figure 2 Three purposeful interaction modes. A and B denote teams. Relationships should evolve as learning objectives and capability needs change. Schematic based on Team Topologies concepts. [4, 5]

Mode	Use it for	Review condition
Collaboration	Joint discovery or a new boundary	End or change mode when learning goals are met
X-as-a-Service	A usable internal service	Check adoption and the effort needed to consume it
Facilitating	Capability transfer or a missing skill	Reduce support as the receiving team becomes capable

Team Topologies treats these modes as purposeful relationships. Collaboration needs a defined objective and timeframe; internal services should evolve from consumer needs; facilitation should make sustained dependence unnecessary. [4]

For a selected flow, assign ownership to the next decision and identify dependencies that repeatedly block progress. A recurring request may justify a clearer interface or a self-service capability. The design aim is to reduce the coordination work the system requires.

Value Delivery

Are we delivering changes that create value?

Value Delivery is where teams discover, build, release, operate, and improve products and services. Flight Level 1 is the operational perspective, compatible with the working practices that suit each team. [1]

Where possible, align a team to a meaningful flow of customer or business value. Team Topologies describes a stream-aligned team as aligned to a flow of work, usually within a business domain. Enabling, platform, and complicated subsystem teams can provide supporting capabilities where specialised knowledge or shared services are needed. [5]

Give ownership practical meaning

A team should be able to explain who it serves, what it owns, which outcome it is trying to influence, and what it can change directly. Ownership also includes the feedback needed to judge the result. A team that releases software but cannot observe customer use or operational behaviour has an incomplete learning loop.

Autonomy works within strategic context and agreed guardrails. Stable teams build shared context; changes to their boundaries should respond to a clear flow or capability need. Leaders should specify which decisions teams can make and which require escalation. Coordination should remove avoidable dependencies and resolve conflicts, so autonomy is supported by the wider system.

Deliver in small testable increments

Small batches reduce the amount of work committed before feedback arrives. They make it easier to test an assumption, detect a problem, and change course. DORA's 2024 report reinforces the importance of small batches and testing alongside emerging tools, while recommending an experimental approach to improvement. [6]

A useful increment must be safe to release and observable in use. For a service journey, that may mean improving one step for a defined customer group. For an internal platform, it may mean offering one recurring capability with documentation and a usable interface. The ambition can remain large while each commitment stays small enough to evaluate.

Close the feedback loop

The operational loop is deploy, observe, learn, and improve. A release is evidence that a change reached production; it is not, by itself, evidence that customers gained value. Review adoption, task success, quality, and service behaviour against the intended outcome.

When the expected result fails to appear, examine the explanation before adding scope. The problem may be discoverability, a wrong assumption about user needs, a dependency elsewhere in the journey, or a change that customers cannot use successfully. Bring that evidence back to coordination and strategic reviews.

System Learning

Where is the system constraining itself?

System Learning examines the whole journey from a signal or opportunity to a decision, discovery, delivery, customer use, and measured impact. It prevents the organisation from treating an improvement in one part as proof that the whole system improved.

Systems thinking draws attention to feedback, delays, rules, and the information available to decision makers. These can matter as much as changing a numerical target. Meadows' work on leverage points provides a useful foundation for examining where an intervention changes system behaviour. [7]

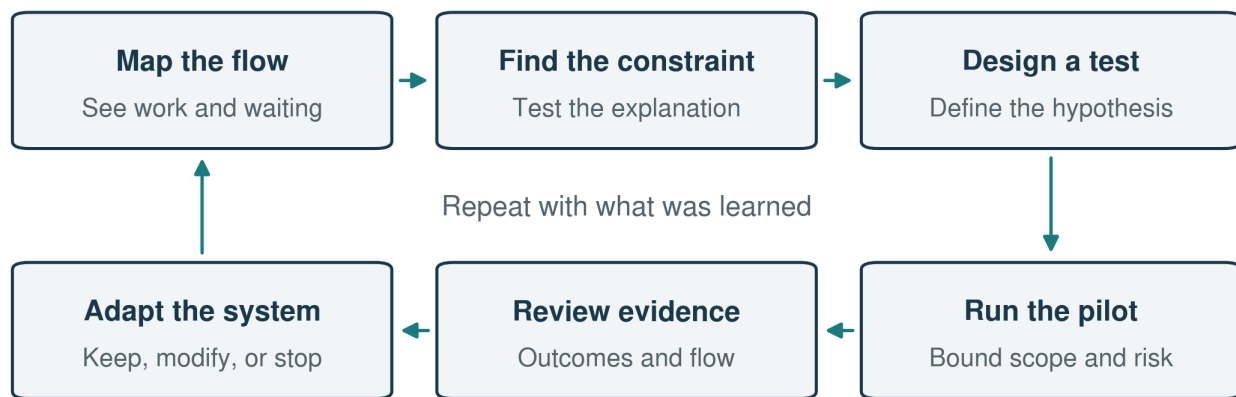


Figure 3 A bounded improvement cycle. Review the explanation and the intervention as hypotheses, then adapt using what the pilot reveals.

Investigate the constraint before the symptom

Slow delivery may reflect too much work in progress, an unclear priority, a shared environment, a fragile platform, an approval gate, or a recurring cross-team dependency. Trace where work waits and why. Test the explanation against evidence from the people and services involved.

An experiment should state the hypothesis, scope, owner, expected effect, measures, duration, and decision rule. It should also identify the conditions that would justify stopping or reversing the change. This makes learning reviewable and keeps the commitment proportionate to the evidence available.

Evaluate customer impact alongside waiting time, quality, risk, adoption, and cognitive load. A faster queue is not an improvement if it transfers effort to another team or increases failures. Keep, modify, extend, or stop the intervention based on the combined evidence.

Adaptive Agility

Can we adapt without creating chaos or waste?

Adaptive Agility is the capability produced by the connected system. An organisation can sense meaningful change, make and revise trade-offs, coordinate across boundaries, deliver safely in small increments, and learn from the result.

Responsiveness must be judged together with reliability, customer value, and sustainable workloads. Constant reprioritisation can undermine the ability to finish work and learn from it. DORA’s 2024 findings associate unstable priorities with lower productivity and greater burnout, reinforcing the need for a considered approach to change. [6]

What the model integrates

Established discipline	Contribution within the loop
Flight Levels	Connects strategic choices, end-to-end coordination, and operations
OKRs	Expresses intended outcomes and makes progress reviewable
Team Topologies	Guides team boundaries, interaction modes, and supporting capabilities
Systems thinking	Examines constraints, feedback, and unintended effects
Business agility	Describes the capability that emerges when the elements work together

These disciplines remain distinct. A Flight Level is a perspective on work, a team topology describes a team’s role, and an OKR expresses intended progress. The model connects their use without treating them as interchangeable structures. [1, 2, 5, 7]

The vehicle metaphor

Strategic Intent is the steering and navigation; Flow Coordination is the transmission; Value Delivery is the engine; System Learning is the diagnostics and telemetry. Adaptive Agility is the ability to move with traction and change direction effectively. More engine power helps only when the rest of the vehicle can use it.

The boundary of the claim

The model is a way to investigate and improve an operating system. Its effectiveness depends on context, decision authority, capability, and the interventions chosen. The established disciplines inform the proposal; they do not independently validate the combined model. A pilot should assess whether it improves the selected flow and the outcomes that matter.

An illustrative application

The following scenario is hypothetical. It shows how the operating logic could be applied; it is not a reported case study or a claim of measured improvement.

A customer onboarding journey

A company wants more customers to complete onboarding successfully. The product team can release interface changes quickly, but work frequently waits for identity integration, policy clarification, and access to test data. Customers continue to abandon the journey even as the delivery backlog is completed.

Set the intended outcome

Leadership defines the customer segment and the intended improvement in successful onboarding, using an agreed baseline and review period. It keeps error rates, service incidents, and customer effort visible alongside completion. A new interface is an initiative to test, rather than the definition of success.

Make the flow and dependencies visible

The involved teams map the journey from the first decision to customer use. A coordination view identifies active initiatives, available capacity, and work waiting for another team or decision maker. The group discovers that access to realistic test data is a recurring dependency across several initiatives.

Choose a bounded intervention

The product and platform teams collaborate for a defined period to understand the need. They test a small internal service that provides approved test data through a documented interface. An enabling team facilitates the skills required to consume the service, with a clear point at which the product team can proceed independently.

Deliver, observe and review

The product team releases one onboarding improvement to a defined group and observes completion, errors, and customer feedback. The coordination group reviews waiting time and service adoption. System learning examines whether the test data service removed the dependency or merely created a new support queue.

If flow improves while onboarding completion remains unchanged, the team should investigate the customer hypothesis. If completion improves but incidents rise, the intervention needs adjustment. If the new service adds effort, the teams should revise or stop it. A positive result supports further testing in adjacent flows; it does not justify an immediate enterprise rollout.

The example demonstrates the connection: a strategic outcome guides the work, coordination exposes friction, teams deliver an increment, and evidence changes the next decision.

A practical 90-day starting plan

Start with one strategically meaningful value stream, customer journey, product area, or initiative with visible delivery friction. Choose a scope where the people involved can observe the full flow and act on at least one constraint.

First 30 days

Define the intended outcome and the evidence that would indicate progress. Agree on the customer or service scope, a baseline, and the operational conditions that must remain acceptable. Identify a sponsor who can resolve decisions beyond the authority of the participating teams.

Map the actual journey from strategic request to customer impact. Include teams, handoffs, decisions, queues, dependencies, and work in progress. Use a simple Flight Level 2 coordination view to expose where work is waiting. Record the measurement boundaries so later comparisons refer to the same flow.

Days 31 to 90

Limit active initiatives within the selected flow and make deferred work explicit. Clarify ownership and the intended interaction modes between teams. Investigate one recurring dependency that could be reduced through a better boundary, capability facilitation, or an internal service.

Run at least one bounded improvement experiment. Define its hypothesis, scope, measures, review date, and decision rule before changing the system. Review customer, operational, and flow evidence together; include the experience of the people doing and consuming the work.

An outcome review should lead to a decision. Continue, modify, stop, or extend the intervention and document the explanation. New evidence can also justify revising the strategic hypothesis rather than improving the delivery of an initiative that no longer makes sense.

Beyond 90 days

Connect the coordination view to Flight Level 3 priorities and outcome reviews. Extend learning to adjacent flows where the evidence supports it. Evolve platform capabilities from recurring needs of stream-aligned teams, rather than an assumption that every team needs the same solution. CNCF's platform guidance emphasises internal user needs, self-service, documentation, and reduced cognitive load. [8]

Review team interactions, dependencies, and strategic assumptions regularly. Organisational changes should be proportionate to the problem observed. A clearer decision rule or interface may be sufficient; a broader change in ownership may be appropriate when smaller interventions cannot remove the constraint.

The 90-day horizon is a practical starting cadence, not a promise that enterprise agility can be established within a quarter.

Measures and leadership decisions

Use a small set of measures that connects the intended outcome with how work moves and how safely it is delivered. The measures below are examples for a pilot, not a universal scorecard. Choose definitions and data sources suited to the selected flow.

Perspective	Example evidence	Decision it informs
Customer outcome	Task success, adoption, customer effort	Is the initiative solving the intended problem?
Flow	Elapsed time, waiting time, work in progress	Where is progress constrained?
Quality and risk	Failure, rework, service incidents	Is faster flow creating unacceptable costs?
Team capability	Recurring dependencies, effort to use a service	Are teams becoming more self-sufficient?
Strategic learning	Hypotheses tested, decisions revised	Should we stop, change, or extend the work?

Keep measurement boundaries consistent. Team cycle time and elapsed time from strategic decision to customer impact answer different questions. Compare like periods and account for changes in workload, customer mix, and scope before attributing a result to the intervention.

Use reviews to make decisions

Leaders should examine whether the intended outcome moved, which constraint remains most important, and whether the intervention transferred effort or risk elsewhere. Teams need a clear response to the evidence: a changed priority, a resolved dependency, an adjusted guardrail, or an explicit decision to continue.

Measures should support investigation, rather than reward local optimisation. A rise in team throughput may be useful, but it is incomplete evidence if customer value still waits downstream. Use customer, operational, and flow evidence together.

Conclusion

The Adaptive Enterprise Loop connects strategy, coordination, delivery, and learning into an operating logic. It gives leaders and teams a way to investigate why value moves slowly, test a proportionate change, and revise their choices based on evidence.

The first step is concrete: select one important flow, establish what success would look like, and test an intervention against the actual constraint. The organisation becomes more adaptive as it learns to change its operating system while delivering value.

References

The sources below describe the established disciplines used in this paper. The Adaptive Enterprise Loop is the synthesis presented here. The illustrative scenario and starting plan are proposed applications, not empirical findings. Web sources were checked on 5 October 2026.

- [1] Klaus Leopold. [What gets done at which Flight Level?](#) LEANability, 15 November 2019.
- [2] Google re:Work Editorial Team. [Set goals with OKRs](#). Originally published 2016.
- [3] Google. [Google's OKR Playbook](#). Reprinted with Google's permission by What Matters.
- [4] Team Topologies. [Team Topologies Interaction Modes: Breaking Through Common Misconceptions](#). 21 February 2025, curated by Eduardo da Silva.
- [5] Team Topologies. [Key Concepts](#). Team types, interaction modes, and the flow of value.
- [6] DORA. [Accelerate State of DevOps Report 2024](#). Google Cloud, 2024, report overview and download.
- [7] Donella Meadows. [Leverage Points: Places to Intervene in a System](#). The Donella Meadows Project.
- [8] CNCF TAG App Delivery. [CNCf Platforms White Paper](#). Platform value, implementation, and measurement.