AI GOVERNANCE SERIES | PART 3

The Operating Model for AI Governance

Who Owns the System When Governance Is the Architecture

By Greg Aldrich | Global CIO & Strategic Advisor | April 2026

Most organizations think AI governance is just about forming the right committee.

That's not where the problem is.

The real challenge is designing an operating model where governance is embedded, not applied after the fact. In Part 1 of this series, I argued that organizations cannot wait for regulation to define responsible AI. In Part 2, I described how spec-driven development provides the technical architecture for the shift in governance at machine speed. Part 3 answers the question that those two articles inevitably raise: who owns it, and how should your organization be structured to make it real?

The honest answer is that most organizations are not structured for this at all. They have governance bodies and engineering teams operating in parallel, with different timelines, different incentives, and fundamentally different definitions of what "success" means.

> ***Committees produce alignment documents. Engineers ship systems. The gap between those two activities is where AI risk actually lives.***

Most organizations treat AI governance as only a committee problem. It is an operating model problem.

Why the Current Model Breaks

The organizational structure most enterprises inherited was built in an environment where governance and development operated on separate timelines. Governance — housed in Legal, Risk, and Compliance — evaluated systems after they were designed or deployed. Engineering — in product and development teams — moved at sprint velocity. The two functions met at defined gates: a compliance review here, a security sign-off there.

That model was already under pressure before AI. It breaks entirely under agent-driven development.

When AI systems can generate, iterate, and deploy code faster than any governance process can review it, the gate model fails. Not because the people in governance aren't capable, but because the architecture of oversight stays structurally incompatible with the architecture of production. You cannot govern at a quarterly cadence what is being built at daily velocity.

The deeper problem is incentive misalignment. Engineering teams are measured on delivery speed, feature completeness, and system performance. Governance functions are measured on risk reduction, policy conformance, and audit outcomes. These are not naturally opposing goals — but in most organizations, they are structurally treated as if they are. The result is a continuous tension that no committee meeting can resolve, because the tension is not a people problem. It is a design problem.

> ***"Committees can align decisions. They cannot control systems."***

The Core Organizational Tension

When I work with organizations on AI governance, the conversation always comes back to the same question: who owns it?

Specifically: who owns the specification? Who owns the validation? Who owns the risk when something goes wrong?

The answer matters enormously because each answer produces a different failure mode. If engineering owns governance, it weakens over time — not through negligence, but because engineering's primary accountability is to deliver, and governance requirements become friction to be minimized. If Legal or Compliance owns it, velocity collapses — not because lawyers don't understand technology, but because their accountability is to risk elimination, which is a different objective than responsible risk management. And if no one owns it clearly — the most common outcome — the organization has the appearance of governance without the reality of control.

I have seen all three failure modes up close. The engineering-owned version produces fast systems with invisible risk accumulation. The compliance-owned version produces safe documents and slow systems. The no-clear-owner version produces confident presentations to leadership and genuine surprises in production.

> ***"When ownership is unclear, control does not exist. It is only a matter of time before that becomes visible."***

The answer is not to find the right department to own governance. It is to redesign how ownership works.

The Shift: From Functional Ownership to System Ownership

In Part 2, I described how spec-driven development moves control upstream — from code review to specification definition. The organizational implication of that shift is equally significant: governance cannot be "owned" by a department the way a product or a process is owned. It must be expressed through systems.

Practically, this means governance is not located in a team. It is expressed through the specifications that define what systems must do, the validation pipelines that continuously enforce those definitions, and the deployment controls that prevent systems from operating outside their defined boundaries. The humans who govern are the ones who design, maintain, and evolve those systems — not the ones who review outputs after the fact.

This is a meaningful shift for most organizations to absorb, because it changes what governance people do. The job is no longer primarily evaluation and approval. It is architecture and enforcement. That requires different skills, different relationships with engineering, and different metrics for success.

To be clear: this does not eliminate the need for human assessment in governance. It repositions where that judgment has the most leverage — upstream, in the definition of constraints and acceptable boundaries, rather than downstream in the review of individual implementations.

Ownership does not disappear in this model. It becomes explicit, distributed, and enforced through systems rather than concentrated in a single function.

What the Organization Actually Looks Like

Based on what I've seen work — and what I've built in my own environments — here is a reference model for an AI governance operating structure that can actually function at scale. It has five layers, each with a distinct role and a distinct accountability.

This model replicates the system architecture itself: strategy defines boundaries, specifications translate them, engineering executes within them, and validation continuously enforces them.

Layer 1: AI Governance Council — *The Strategic Layer*

The Governance Council sets the organization's risk posture, policy direction, and acceptable boundaries for AI deployment. It answers the highest-level questions: what categories of AI use are permitted, what outcomes are unacceptable, and what the organization's obligations are to regulators, customers, and employees.

What it does not do is equally important. The Governance Council does not review individual AI implementations. It does not approve specific systems. It does not function as a quality gate in the development process. Organizations that ask their governance council to operate at that level of detail create a bottleneck that slows delivery without improving control — and they exhaust senior leaders' attention on implementation decisions that should be handled structurally.

The Governance Council's job is to set the rules of the game. Other layers enforce them.

Layer 2: Specification Owners — *The Control Layer*

This is the most critical and least defined role in most organizations today, and it is where the operating model either works or fails.

Specification Owners are the people responsible for translating governance intent into enforceable technical definitions. They take the risk posture established by the Governance Council, the compliance requirements imposed by regulators, and the business outcomes defined by leadership — and they convert those inputs into structured specifications that AI systems and coding agents must satisfy.

This role sits at the intersection of business, governance, and engineering. In practice, this often looks like a hybrid of product leadership, enterprise architecture, and risk translation—formalized into a single accountable role—with clear authority to define constraints that engineering must satisfy. It requires sufficient technical depth to write specifications that are actually enforceable, sufficient business context to ensure they conform to outcomes, and sufficient governance literacy to ensure they capture the right constraints. Most organizations do not have this role. They have business analysts who write requirements, lawyers who write policies, and engineers who write technical specifications — but no one whose job is to hold all three together in a single enforceable artifact.

Creating this role — naming it, funding it, and giving it legitimate authority — is the single highest-leverage organizational decision most enterprises can make in AI governance today.

> ***"The Specification Owner is the most important role in AI governance that most organizations have not yet created."***

Layer 3: Engineering — *The Execution Layer*

Engineering builds the systems, pipelines, and agent orchestration that deliver AI capabilities. In a clearly defined operating model, Engineering's relationship to governance changes significantly:

instead of being the final accountable party for governance outcomes, Engineering functions within a defined-constraint environment established by Specification Owners and enforced by validation infrastructure.

This is not a diminishment of Engineering's role — it is a clarification of it. Engineers are freed from the ambiguous responsibility of making governance judgment calls on the fly because those calls have already been made upstream. Their job is to build systems that satisfy specifications, and to raise the alarm when specifications are unclear, incomplete, or technically unworkable. That feedback loop between Engineering and Specification Owners is one of the most important governance mechanisms in the model.

Layer 4: Validation & Assurance — *The Enforcement Layer*

Validation and Assurance owns the verification pipelines, testing frameworks, and continuous compliance infrastructure that enforce specifications at machine speed. In a traditional model, this function is QA. In an agent-driven model, it becomes the enforcement layer of governance.

The distinction matters. QA asks whether the system works. Validation and Assurance assess whether the system satisfies its specifications, including functional correctness, risk boundary compliance, auditability, and behavioral constraints. These are different questions, and they require different infrastructure to answer continuously rather than episodically.

Organizations that treat this function as elevated QA will underinvest in it. Organizations that treat it as governance infrastructure will fund it accordingly — and will find that it pays for itself in reduced remediation costs and regulatory exposure.

Layer 5: AI Committee — *The Interface Layer*

I want to be direct about how I position this layer, because it is easy to misread as a demotion of AI Committees. It is not.

AI Committees — like the one I chair at Andrew Wommack Ministries and Charis Bible College — are genuinely valuable governance forums. They bring together leaders from across the organization — operations, finance, HR, legal, technology, and business units — to provide a cross-functional perspective on responsible AI use. That convening function is irreplaceable. No technical system can substitute for well-informed human decision-making applied across organizational boundaries.

The reposition I'm suggesting is not about reducing the AI Committee's importance. It is about clarifying its role. The AI Committee is most effective as a coordination layer — the connective tissue between the strategic intent of the Governance Council and the operational reality of engineering and validation. It surfaces emerging issues, translates organizational context into governance guidance, and assures that the humans closest to business operations have a structured voice in AI governance.

What it should not be asked to do is function as the primary control mechanism — reviewing individual systems, approving specific deployments, or serving as the governance gate in development workflows. That is what the other four layers are for. When an AI Committee is asked to carry that weight, it becomes overloaded, slows delivery, and paradoxically produces weaker governance because the real control work — specification, validation, enforcement — never gets built.

> ***"The AI Committee is not where governance breaks down — it is where governance connects. The problem is when organizations build the committee and stop there, assuming alignment equals control."***

Where Most Organizations Break Down

The failure patterns are consistent enough that I can predict them before I walk into a client engagement. The specifics vary. The structure is almost always the same.

- Committees reviewing implementations. The Governance Council or AI Committee is pulled into approving individual systems because there is no structural enforcement layer to do so. Senior leaders spend their governance time on implementation decisions rather than on policy direction. Everything slows, and the decisions that get made are no better for it.
- No clear specification of ownership. Requirements exist — in policy documents, legal memos, and engineering tickets — but no one has the accountability to synthesize them into a single enforceable artifact. The specification is implicit, so the validation criteria are implicit as well, making enforcement impossible.
- Governance is isolated from engineering. The functions that should be most tightly coupled — Specification Owners and Engineering — operate through handoffs rather than collaboration. By the time governance input reaches the engineering team, the architectural decisions have already been made.
- Validation is treated as QA. The verification function checks whether systems work, not whether they satisfy governance constraints. Behavioral boundaries, auditability requirements, and compliance conditions are evaluated episodically — at release gates — rather than continuously throughout development.
- Accountability diffused across functions. When something goes wrong, every function has a plausible explanation for why it was someone else's responsibility. Diffuse accountability is a structural failure, not a people problem.

The underlying pattern is the same in every case: the organization appears to be governed on paper, with the right bodies and documents in place. But control is not operational, because the architecture required to enforce governance at the speed and scale of AI development was never built.

The Role of Executive Leadership

This is not a legal problem. It is not purely a technical problem. It is an operating model design problem — and operating model design is leadership work.

The most effective executive sponsors of AI governance I've worked with share a specific set of behaviors. They explicitly and publicly define ownership, meaning the Specification Owner role exists on the org chart with a name next to it, not as a shared responsibility across three departments. They coordinate incentives structurally, which means governance performance is part of how engineering leaders are evaluated, not a separate compliance obligation that competes with their delivery metrics. They fund validation infrastructure as a governance investment, not as an IT cost to be minimized.

Most importantly, they treat governance architecture as a design decision that belongs on their agenda — not something that will emerge organically from a committee structure. I've seen organizations spend two years revising their AI governance policy documents while their development teams deployed agent-assisted systems with no specification discipline and no verification infrastructure. The policy was advanced. The control was nonexistent.

The design work that precedes every other AI governance investment is the organizational design: who owns what, how the layers connect, where responsibility lives, and how the system enforces itself when no one is watching. That work belongs to leadership. It cannot be delegated to a committee.

An Organizational Maturity Model

Organizations evolve through four recognizable stages in their governance operating model:

- **Stage 1 — Committee and Policy-Based:** Governance is meetings and documents. AI Committees convene, policies are written, and decisions are made by consensus. Control is a function of who is in the room, which means it is inconsistent, slow, and difficult to scale. This is where most organizations start — and where many remain longer than they should.
- **Stage 2 — Coordinated:** Functions are aligned but not integrated. Legal, Risk, Engineering, and Product share awareness of governance requirements and generally operate in good faith. But the handoffs between them are manual, the specifications are informal, and the validation is episodic. Governance improves, but it cannot keep pace with AI velocity.
- **Stage 3 — Embedded:** Governance is integrated into development workflows. Specification Owners exist as a defined function. Validation pipelines enforce constraints continuously. The AI Committee operates as a coordination layer rather than a control gate. Development and governance operate in sync rather than in sequence. This is where the real transformation begins.
- **Stage 4 — Architected:** Governance is expressed through systems. Specifications are living artifacts tied to business outcomes. Validation is automated and auditable. Engineer supervision is focused upstream — on definition and constraints — rather than downstream on review and remediation. The organization can scale AI deployment without proportionally increasing governance overhead.

Most organizations are between stages 1 and 2. The move to stage 3 is primarily an organizational design and leadership decision. The technology to support it exists. The will to restructure around it is the harder requirement.

What Good Looks Like

- The organizations I've seen execute this well don't look dramatically different from the outside. They have AI Committees. They have engineering teams. They have legal and compliance functions. What's different is how those factors connect, and what sits between them.

- Specification ownership is explicit. There is a named function — whether a team, a role, or a center of excellence — whose job is to translate governance intent into enforceable technical definitions. It has authority, budget, and a seat at the table in both governance and engineering conversations.
- Validation pipelines are treated as critical infrastructure. They are funded, maintained, and evolved with the same seriousness as production systems. When a validation pipeline fails, it is treated as a governance incident, not a QA backlog item.
- Governance and engineering operate in sync. Specification Owners are involved in system design from the beginning, not consulted after architecture decisions have been made. The feedback loop between engineering constraints and governance requirements runs continuously, not at release gates.
- The AI Committee coordinates rather than controls. It surfaces emerging issues, provides a cross-functional perspective, and ensures organizational context is reflected in governance decisions. It does not approve individual systems or serve as a development gate. Its value is in the quality of its input to the strategic and control layers, not in its position in the approval chain.
- Accountability is explicit and structural. Every governance function has a clear owner, a clear accountability, and a clear metric for success. When something goes wrong, the organizational design makes it possible to identify what failed and why — and to fix the structure, not just the outcome.

Final Thought

The progression of this series mirrors that of the problem itself. Part 1 established why AI governance must change — the structural gap between regulatory timelines and AI velocity, and the obligation to build governance capability before it is required. Part 2 described how spec-driven development provides the technical architecture for governance at scale. Part 3 answers the question that follows: who does the work, and how must the organization be structured to support it?

The answer, in each case, is the same.

> ***"Move control upstream. Make it structural. Build the systems that enforce it rather than the processes that review it."***

For AI Committees specifically: their value is not diminished by this model — it is clarified. A well-functioning AI Committee, surrounded by the governance infrastructure described here, becomes significantly more effective. It stops spending time on implementation approvals and starts doing the work only a cross-functional team can do: providing judgment, organizational context, and strategic direction that no automated system can replicate.

The question is no longer who approves AI. It is who defines the system — and how that definition is enforced.

Organizations that answer this well won't just govern AI more effectively. They will operate it at scale, with control, consistency, and confidence that reactive governance models cannot match.

This is Part 3 of an ongoing series on AI governance and organizational readiness.
Part 1: The AI Governance Tightrope | *Part 2: The New Control Layer*

About the Author

Greg Aldrich is a Global CIO and Strategic Advisor with 30+ years of experience helping boards, executive teams, and C-suite stakeholders navigate IT strategy, AI governance, and digital transformation. He serves as Senior Strategy Advisor at Blue Tree Technology Group and Senior Transition Architect at SDS Consulting, and currently serves as CIO at Andrew Wommack Ministries & Charis Bible College, where he chairs both the AI Committee and IT Steering Committee. He has advised organizations across financial services, gaming, healthcare, higher education, logistics, and nonprofit sectors.

Connect: linkedin.com/in/galdrich