Realizing ISO/IEC 42001 through Decision Behavior Governance

From Administrative Claims to Engineering Facts

Author: Spark Tsai
Date: April 2026
Keywords: AI Governance, ISO/IEC 42001, Decision Behavior Governance (DBG), Decision Constraint Layer (DCL), AI Management System (AIMS), Auditability, Engineering Facts

---

Abstract

As ISO/IEC 42001 (Artificial Intelligence Management System, AIMS) emerges as the global reference standard for organizational AI governance, a structural gap has become increasingly evident between compliance intent and technical realization. Most implementations continue to rely on **administrative claims**—policies, codes of conduct, risk registers, and manual reviews—which are insufficient to govern the probabilistic and non-deterministic behavior of modern AI systems, particularly Large Language Models (LLMs).

This paper argues that effective compliance requires a transition from administrative claims to **engineering facts**. We propose **Decision Behavior Governance (DBG)** as an architectural realization path for ISO/IEC 42001. By situating governance within a non-bypassable **Decision Constraint Layer (DCL)** prior to execution, organizations can transform governance from ex-post surveillance into an ex-ante institutional condition.

Through this lens, the PDCA (Plan–Do–Check–Act) cycle is reconstructed as a verifiable engineering process, enabling accountability, auditability, and legal attribution in probabilistic AI environments.

---

1. Introduction: The Compliance Vacuum in ISO/IEC 42001

ISO/IEC 42001 establishes a comprehensive management framework for AI governance while intentionally remaining technologically agnostic. This flexibility, however, often results in a **compliance vacuum**: organizations possess extensive documentation yet lack technical mechanisms to ensure that AI agents adhere to defined governance constraints during black-box inference.

In probabilistic systems, stating a policy does not equate to controlling behavior. Runtime observations alone cannot prove that a decision was formed under governance. To satisfy the standard's requirement for *effective control*, governance must exist not merely as an administrative intention, but as an **architectural precondition** embedded into the decision lifecycle itself.

2. From Administrative Claims to Engineering Facts

We distinguish two fundamentally different categories of compliance evidence:

1. **Administrative Claims**
   Documents asserting what an AI system *should* do—such as ethics principles, internal policies, and risk assessments. These are necessary for governance intent but remain decoupled from the execution loop.

2. **Engineering Facts**
   Immutable, versioned, and non-bypassable artifacts that define what an AI system *was allowed* to do at the moment a decision was formed.

High-quality ISO/IEC 42001 compliance requires elevating governance evidence from the former to the latter.

---

3. Governance Existence as an Architectural Condition

To bridge policy and execution, an AI Management System must establish **Governance Existence**. In engineering terms, this is realized as a **Decision Constraint Layer (DCL)**—a conceptual architectural boundary positioned prior to execution.

The DCL ensures that every decision formed by an AI agent is structurally bounded before probabilistic sampling occurs. While ISO/IEC 42001 does not mandate a specific representation or syntax, governance existence must satisfy three core properties:

- **Non-bypassability**
  Constraints must be inherent to the decision path, not optional post-processing filters.

- **Determinism of Boundary**
  Constraints must provide a fixed structural boundary that contains stochastic model outputs.

- **Traceability**
  Every decision must be attributable to a specific, versioned set of constraints.

Governance exists once these conditions are met, regardless of whether runtime intervention is triggered.

---

4. Runtime Governance vs. Development Governance

Contemporary AI governance discourse frequently emphasizes runtime governance mechanisms, including output filtering, real-time monitoring, and execution-time

interception. These mechanisms are indispensable for managing operational risk, but their prominence has led to a conceptual conflation between governing execution and governing decision behavior.

4.1 Runtime Governance

Runtime governance operates after a decision has already been formed and entered the execution loop. Its functions include:

- Observing generated behavior
- Intercepting or suppressing policy violations
- Enforcing environmental restrictions
- Producing logs for audit and incident response

Runtime governance is reactive and event-driven. From a probabilistic standpoint, it functions as a stochastic filter applied to stochastic outputs. The absence of a triggered interception represents a statistical outcome—not evidence of structural control.

Runtime governance governs **outcomes and environments**, not the decision formation process itself. Crucially, it **cannot alter system-intrinsic decision logic**.

4.2 Development Governance

Development governance operates prior to execution, at the stage where the decision space is defined and institutionally authorized. Its functions include:

- Defining admissible decision premises
- Establishing explicit, versioned constraints
- Structuring priority, override, and exclusion rules
- Defining legitimacy conditions under which decisions may be formed

Development governance is ex-ante, structural, and deterministic. It governs the **decision structure**, not the execution environment.

In probabilistic AI systems, statistical uncertainty persists throughout the lifecycle. Governance therefore cannot be deferred to runtime alone. Development governance exists to bound the decision space before sampling occurs, reducing systemic risk that execution-stage controls—by their reactive nature—cannot reverse.

---

5. Decision Behavior Governance (DBG)

Decision Behavior Governance (DBG) targets the formation of decisions, rather than their outputs, environments, or side effects. It defines the institutional conditions under which decision-making is allowed to occur.

DBG does not optimize reasoning quality, nor does it replace runtime governance. Instead, it establishes governance existence prior to execution, enabling runtime mechanisms to operate within a meaningful structural context.

5.1 Human-in-the-Loop Compatibility

DBG does not eliminate human oversight. On the contrary, it defines the structural boundaries within which **human-in-the-loop intervention becomes meaningful, traceable, and accountable**. Human review, approval, or override can be explicitly incorporated as part of the decision constraint structure.

5.2 Risk Definition at the Decision Level

Under DBG, risk is not merely observed at runtime but can be **explicitly defined and bounded during decision formation**. Risk becomes a structural attribute of permissible decisions, rather than a post-hoc classification of outcomes.

---

6. Reconstructing the PDCA Cycle as an Engineering Process

DBG enables ISO/IEC 42001's PDCA cycle to be realized as a continuous engineering mechanism:

- **Plan**
  Translate governance objectives into structured decision constraints.
- **Do**
  Embed constraints into a non-bypassable architectural boundary. Governance is active by default.
- **Check**
  Verify compliance through automated reconciliation between runtime evidence and constraint artifacts.
- **Act**
  Update constraint versions directly, without retraining models or relying on manual enforcement.

This transformation converts governance from documentation maintenance into operational fact.

---

7. Due Diligence and Legal Attribution

In regulatory or legal inquiries, the distinction between **Invocation** and **Existence** becomes decisive.

- Runtime governance addresses *symptoms* at execution.

- DBG addresses *causes* within the decision structure.

By demonstrating that an AI agent operated within a predefined, versioned DCL, an organization establishes the strongest form of **due diligence**: proof that governance existed at the moment of decision formation, independent of probabilistic outcomes.

---

8. Conclusion: Governance as an Architectural Precondition

ISO/IEC 42001 should not be treated as a bureaucratic obligation, but as an opportunity for architectural maturity. Administrative claims express intent; **only engineering facts provide proof**.

Decision Behavior Governance ensures that governance is not a reactive afterthought, but a non-bypassable architectural precondition. In probabilistic AI systems, the only way to govern *how* decisions are made is to structurally constrain *where and when* they are formed.

We do not deny the necessity of runtime governance, nor do we argue that development governance alone is sufficient. Both are required, representing distinct stages of governance within the AI lifecycle. They do not replace one another; they complete different parts of the governance problem.

Effective AI governance demands both—clearly distinguished, properly situated, and institutionally aligned.