AI Agent Governance Framework

v2.0.0

Author: John J. McCormick · Updated: February 28, 2026 · CC BY 4.0 · aiagentgovernance.org

SUGGESTED CITATION

McCormick, J. J. "AI Agent Governance Framework: A Governance Methodology for Autonomous Software Production Systems." v2.0.0, February 2026. aiagentgovernance.org. https://aiagentgovernance.org/framework/operator-governance-framework/

> *A Governance Methodology for Autonomous Software Production Systems*

> ***Authorship context:*** *This is a practitioner's methodology, not an academic paper. The author built and operated the governed agent system described here — writing governance controls in production as agent failures occurred, documenting incidents in real time, and extracting the methodology from operational experience. The theoretical frameworks cited throughout (Dekker, Reason, Vaughan, Boyd, Weick & Sutcliffe) were identified after the governance mechanisms were built; they provide vocabulary for patterns that production operations had already discovered empirically.*

Abstract

Governance ensures agents do the right work, the right way, with human oversight at every phase transition. The AI Agent Governance Framework is a governance methodology for supervising autonomous AI agents performing real work under delegated authority. It addresses a structural gap in the current AI landscape: the space between model-level compliance, security controls, and observability tooling — none of which govern how agents do their work. The framework provides a phase-gated lifecycle, a set of governing principles, a behavioral failure taxonomy, and a maturity model for organizations deploying AI agents at any scale. The governance methodology operates independently of any specific enterprise application and governs the software production system itself. This framework was derived from continuous production operation of a governed autonomous software production system in which agents performed real software engineering work under enforced authorization and audit controls.

---

1. Origin and Motivation

This framework emerged during the development and operation of a governed autonomous software production system designed to build regulated enterprise applications. The system operates under a governed control plane with full auditability. Governance controls were not designed in advance — they were introduced in response to observed operational failures within that system. The framework documents those controls and their evolution.

The Problem

Organizations deploying AI agents at scale face a governance gap that existing frameworks do not close:

- **Observability** shows what agents did. It does not determine whether agents should have done it.
- **Security** blocks what agents should not do. It does not ensure agents do the right work in the right way.
- **Compliance** checks that regulatory requirements are met at the model and policy level. It does not govern the operational lifecycle of agent work.
- **Governance** ensures agents do the right work, the right way, with human oversight at every phase transition.

Each of these layers is necessary. None is sufficient. The governance gap exists between them — a structural space that no existing tool category fills. The gap is not theoretical: production operations have documented recurring behavioral failure modes that fall entirely within it. Agents that pass every security check, produce clean observability traces, and operate within compliant organizations can still perform work from fabricated premises, skip governance phases, or assume authority they do not have.

Empirical Basis

The framework's empirical foundation consists of eleven documented incidents across five drift categories, producing eight behavioral failure pattern categories and eight governance directives. Each incident is formally documented with root cause analysis, remediation, and the governance mechanism it produced. Every mechanism in the framework traces to a specific operational failure.

The theoretical foundations were identified after the governance mechanisms were built — they provide vocabulary for patterns that production operations had already discovered empirically. These include: Two-Person Integrity (U.S. DoD, 1962), the

Swiss Cheese Model (Reason, 1990), Drift Into Failure (Dekker, 2011), Normalization of Deviance (Vaughan, 1996), the OODA Loop (Boyd, 1976), and High-Reliability Organizations (Weick & Sutcliffe, 2007).

---

2. Scope and Non-Scope

The framework defines governance principles and lifecycle structure. It does not prescribe implementation details or enforcement mechanisms. Organizations implementing this framework will determine the specific technical controls appropriate to their environment.

In Scope	Out of Scope
Governance principles for agent operations	Implementation architecture
Phase-gated lifecycle model	Enforcement mechanism design
Behavioral failure pattern categories	Detection heuristics or algorithms
Governance maturity assessment	Platform-specific tooling
Authority and trust models	Identity, session, or token mechanics
Incident-driven methodology evolution	Routing or orchestration mechanics
Regulatory alignment mapping	Sector-specific compliance programs

This boundary is deliberate. The framework defines governance methodology. It does not disclose specific control plane implementations, enforcement mechanisms, or internal architecture. Organizations adopt the methodology and implement it with controls appropriate to their context.

---

3. Definitions

Five core terms anchor the framework. These are defined precisely to distinguish them from colloquial or overlapping usage in adjacent fields.

Operator. The human authority responsible for governing agent operations. The operator approves phase transitions, routes work between agents, makes trust calibration decisions, and maintains the governance audit trail. The operator role may be filled by a single individual or distributed across a team. The governance principle: humans authorize, agents execute.

Agent. An autonomous software system that performs work under delegated authority. An agent receives assignments, executes multi-step tasks, makes intermediate decisions, and produces work product that enters production systems. Agents are distinct from tools (which execute single actions) and models (which generate outputs without operational autonomy).

Governance. The operational discipline of supervising autonomous AI agents performing real work under delegated authority. Governance addresses the full lifecycle of agent work — assignment, execution, review, remediation, and approval — with structured human oversight at defined control points. Governance is distinct from model governance, security governance, and compliance governance.

Authorization. The formal granting of permission for an agent to proceed with a defined scope of work. Authorization occurs at phase gates where a designated authority approves the transition from one phase to the next. Authorization is distinct from authentication (verifying identity) and from capability (having the technical ability to perform an action). An agent may be authenticated and capable without being authorized.

Execution. The performance of defined work by an agent within an approved scope. Execution occurs within the boundaries established by authorization and is subject to review before the work advances to the next governance phase. Execution without authorization — or execution that exceeds authorized scope — is a governance failure regardless of whether the work product is correct.

---

4. Core Principles

Nine design principles govern all governance mechanism design within this framework. These principles are derived from the theoretical foundations and validated against the incident history.

Principle 1: No Single Actor Holds Both Keys

No agent should be both the author and sole executor of a governance decision. An agent that writes a plan and then executes that plan without independent verification is a single actor holding both authorization keys. Phase gates enforce separation of authority: the proposing agent and the approving actor are always distinct.

Derived from separation of duty (SoD, NIST SP 800-53 AC-5) and the separation-of-authority principle in Two-Person Integrity (TPI, U.S. DoD, 1962). The MOIAS implementation enforces SoD: no single actor can both propose and authorize a critical action. TPI demonstrates that this separation-of-authority principle has 60

years of operational validation in the most demanding safety-critical context known — U.S. nuclear weapons operations. However, MOIAS implements SoD's role-based separation, not TPI's concurrent-presence or symmetric-incapacity requirements (where neither actor alone can complete the action, enforced by physical interlock design). SoD has its own extensive operational history in financial controls and access management at global scale.

Principle 2: Defense Layers Must Fail Independently

When adding a new governance mechanism, its failure mode must be independent of existing layers. Two layers that fail for the same reason are effectively one layer. Safety comes from stacking layers with different failure modes, not from making any single layer perfect.

Derived from: Reason's Swiss Cheese Model (1990). Note: In single-operator deployments, this principle provides partial independence only — a single human sitting at every phase gate means that one cognitive failure mode (automation bias, attention drift, fatigue) can compromise multiple layers simultaneously. Full independence of defense layers requires distributed or role-separated oversight, which the maturity model addresses at Level 2 and above through independent validation functions.

Principle 3: Detect Drift Before It Normalizes

Every deviation from expected behavior is documented, even when the outcome is good. A pattern of borderline outcomes is a drift signal, even if no individual outcome was incorrect. The antidote to drift is a structural commitment to never normalizing deviation.

Derived from: Dekker's Drift Into Failure (2011) and Vaughan's Normalization of Deviance (1996).

Principle 4: Orientation Determines What You Can See

Different actors oriented toward different questions catch different failures. A planning agent oriented toward throughput will miss compliance gaps. An operator oriented toward work quality will miss plan prerequisite gaps. Governance systems need actors with diverse orientations, not redundant actors with the same orientation.

Derived from: Boyd's OODA Loop (1976).

Principle 5: Authenticate Both the Person and the Action

Knowing who approved a decision is necessary but not sufficient. Knowing whether the action was compliant is also necessary. Both must pass independently.

Principle 6: Governance Is Empirical, Not Aspirational

Every mechanism should be traceable to a specific incident or measurable drift pattern. Mechanisms that cannot be traced to a real failure are questioned — governance complexity without safety value is itself a risk.

Principle 7: Cheap Redundancy Over Expensive Perfection

Multiple automated checks are more cost-effective than relying on a single human judgment. Human attention is the scarcest resource — mechanisms should reduce cognitive load on human decision-makers, not increase it.

Principle 8: Governance Must Reduce Operator Cognitive Burden

Governance systems must structure decisions into discrete, auditable units. Governance that increases operator cognitive load introduces systemic risk. Impact statements, structured phase transitions, and pre-validated decision surfaces exist to compress complex governance decisions into forms that operators can evaluate rapidly and reliably.

Principle 9: A Validator Who Participates in Planning Cannot Independently Assess the Plan

An actor who contributes to a plan’s design shares the plan’s assumptions. When that actor later reviews the plan’s execution, they assess it from within the same orientation that produced it — and cannot see failures that the orientation itself obscures. Genuine validator independence requires both orientation diversity (different analytical perspective) and, at enterprise scale, organizational independence (freedom from shared incentives).

This is derivable from Principle 1 (no single actor holds both keys) and Principle 4 (orientation determines what you can see). It adds a specific constraint: independence is not just about separating proposal from approval — it is about ensuring the approving actor brings a genuinely different analytical perspective. Two reviewers with the same training, the same context, and the same assumptions are effectively one reviewer, regardless of organizational separation.

Genuine validator independence requires both properties: orientation diversity ensures that the reviewing actor brings a different analytical perspective; organizational independence ensures that the reviewing actor is not subject to shared

incentives that compromise that perspective. In enterprise-scale deployments, organizational independence — as established in IV&V doctrine (MIL-STD-2168, IEEE 1012, NASA NPR 7150.2) — is the primary structural requirement, because shared budget, management chain, and stake in project outcomes can compromise orientation independence regardless of how different the analytical frameworks are. Orientation diversity is the quality multiplier that makes organizational independence effective — identically oriented but organizationally separated reviewers satisfy the structural requirement without providing genuine analytical diversity. In single-operator and small-team deployments where organizational independence is not yet achievable, orientation diversity provides the first meaningful layer of independence.

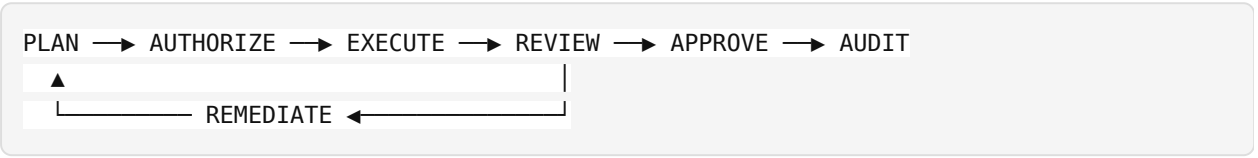
Derived from: Two-Person Integrity (U.S. DoD, 1962), Boyd's OODA Loop (1976), IV&V doctrine (MIL-STD-2168, IEEE 1012).

---

5. Lifecycle Model

The framework governs agent work through a sequence of phases, each separated by a gate that requires explicit approval before the agent may proceed. Gates are structural controls designed to prevent the executing agent from advancing its own work unilaterally.

Phases



Phase	Purpose	Gate Requirement
PLAN	Define work scope, acceptance criteria, and execution approach	Human approval of scope and approach
AUTHORIZE	Verify that plan requirements are met and execution is permitted	Authorization from designated authority
EXECUTE	Perform the defined work within the approved scope	Completion signal with evidence of work performed
REVIEW	Independent assessment of work quality and correctness	Structured findings report with severity classification
APPROVE	Human confirmation that the work meets acceptance criteria	Human sign-off
AUDIT	Verify governance lifecycle compliance and record governance decisions	Complete audit trail of lifecycle execution

The lifecycle is iterative — work may cycle through Review, Remediate, and Execute multiple times before reaching Approve. Specific implementations may define additional phases (e.g., separate engineering and architecture review phases) appropriate to their operational context.

Gate Properties

Every phase transition passes through a gate with four properties:

1. **Explicit approval required.** No agent may advance its own work to the next phase. A separate actor must approve the transition.
2. **Evidence-based.** Gate approval requires documented evidence that the current phase's requirements are met.
3. **Reversible.** If a gate check reveals deficiencies, work returns to the appropriate phase for remediation.
4. **Auditable.** Every gate decision is logged with: who approved, when, what evidence was presented, and what the approval authorizes.

The Operator-as-Router Pattern

A human operator sits between every phase transition. The operator is not a bottleneck — the operator is a control point. The operator-as-router pattern provides context preservation across agent sessions, priority management across workstreams, quality assessment at each gate, and an auditable routing trail.

This is a deliberate architectural choice. Full agent autonomy and full manual control are both failure modes. The lifecycle provides calibrated delegation: agents execute within scoped authority boundaries, and humans approve phase transitions.

Impact Statements

Every phase-transition request includes a structured impact statement that provides the approving human with a decision surface:

```
Risk:         LOW / MEDIUM / HIGH / CRITICAL
Action:       What this approval authorizes
If approved:  What happens next
If rejected:  What happens instead
Reversible:   Yes / No / Partial
Blast radius: Which systems or users are affected
```

Risk levels are determined by reversibility and blast radius, not by technical complexity:

Risk Level	Condition
LOW	Review-only gate, or all prior gates already passed
MEDIUM	First review of new work, or partial prior passes
HIGH	Production deployment, data migration, or schema change
CRITICAL	Irreversible action, multi-tenant impact, or security boundary change

Impact statements exist because human approvers face cognitive overload when reviewing dense technical content under time pressure. The structured format compresses complex governance decisions into forms that operators can evaluate rapidly and reliably.

Failure Containment

The phase-gated lifecycle is a containment architecture. Each gate limits the blast radius of a failure in the preceding phase:

Phase	What Could Go Wrong	What Limits the Damage	What Prevents Escalation
PLAN	Work scoped incorrectly; wrong problem addressed	AUTHORIZE gate checks plan against requirements before execution begins	No code written, no resources committed. Remediation cost: one planning cycle
EXECUTE	Agent deviates from plan, infers work from wrong source, expands scope	REVIEW gate catches deviations before work is accepted	Work product is isolated — not deployed, not integrated, not visible to end users
REVIEW	Reviewer misses a defect; review orientation too narrow	Separated authority (Principle 1) and orientation diversity (Principle 4) — multiple reviewers with different perspectives	Defect caught in APPROVE gate or in production monitoring. Incident documented and fed back into the learning loop
APPROVE	Operator approves work that contains an undetected defect	Structured impact statement forces explicit risk assessment. Override capability allows halt at any point	Deployment is the last gate, not the first. Every preceding gate has already filtered for known failure modes

Key containment properties:

1. **No code deploys without review approval.** The lifecycle structurally prevents agents from shipping unreviewed work (the failure mode documented in INC-009).
2. **Remediation occurs before production.** Defects found in REVIEW return to EXECUTE via REMEDIATE. The defect never reaches production.
3. **Human oversight prevented incorrect implementation.** In the author's operational context — a small sample not intended as a predictive performance metric — the human operator caught issues before deployment in 9 of 11 documented incidents. The governance lifecycle made the issues visible; the human made the decision. This observed outcome should not be extrapolated to other operational contexts.
4. **Every failure improves the system.** Incidents that escape containment produce new governance mechanisms (§8). The containment architecture is self-improving.

---

6. Human Oversight Model

The framework assumes that human operators provide the reliability layer in the governance lifecycle. Agents execute; humans authorize. Agents produce work; humans approve it. This structural separation is fundamental — and correct.

It is also incomplete.

6.1 Operator Authority Map

The operator exercises authority at five control points in the governance lifecycle:

Control Point	Authority	What the Operator Decides
PLAN gate	Scope approval	Whether the work should be done, and whether the proposed approach is sound
AUTHORIZE gate	Execution permission	Whether the agent may proceed with the approved plan
REVIEW routing	Reviewer assignment	Which independent actors review the work, and what orientation they bring
APPROVE gate	Acceptance decision	Whether the work meets acceptance criteria and may advance to production
Override	Process intervention	The operator may halt, redirect, or terminate any workstream at any point

The operator also exercises continuous authority over:

- **Agent trust calibration** — adjusting autonomy boundaries based on observed behavior
- **Workstream priority** — determining execution order across concurrent work
- **Incident classification** — determining root cause and governance implications
- **Methodology evolution** — approving changes to the governance framework itself

6.2 Monitoring Scope

The operator monitors for behavioral patterns during agent execution. Effective monitoring requires:

1. **Access to execution context** — not just outputs, but the agent's reasoning, source material access, and intermediate decisions
2. **Pattern recognition against the taxonomy** — knowing what the eight behavioral failure patterns look like in practice

3. **Baseline behavioral expectations** — understanding what normal execution looks like so that deviations are visible
4. **Cross-agent comparison** — recognizing when different agents exhibit similar drift patterns on different workstreams

What the operator cannot monitor without structural support:

- **Scale beyond human attention** — a single operator cannot watch every agent in real time as fleet size grows
- **Novel failure modes** — patterns not yet in the taxonomy require the operator to recognize something as wrong without a documented reference pattern
- **Slow drift** — gradual behavioral changes across many sessions that no single observation would flag (Dekker, 2011)
- **Own cognitive biases** — the operator's own susceptibility to the failure modes described below

6.3 The Human Drift Observation

The framework's incident history reveals a structural observation that AI agent governance frameworks have not yet explicitly integrated: **the behavioral patterns documented for AI agents are not unique to agents.** The underlying phenomena — automation bias, over-trust, and vigilance decrement — are well-established in the human factors literature (Parasuraman & Riley, 1997; Cummings, 2004). The framework's specific contribution is applying these findings to AI agent governance contexts, where the human failure modes have a direct structural mapping to the agent behavioral taxonomy. Humans operating as the governance authority exhibit structurally equivalent failure modes under operational conditions.

This is not a deficiency of any individual operator. It is a structural characteristic of human cognition under the conditions that agent governance requires — sustained attention, pattern recognition under time pressure, and authority exercised across many concurrent decisions. These are precisely the conditions under which Dekker (2011) and Vaughan (1996) document organizational drift, and under which the human factors literature documents automation bias — the tendency to over-rely on automated system outputs (Parasuraman & Manzey, 2010) and to calibrate trust inappropriately in automated systems (Lee & See, 2004). The Human Drift Observation applies these established findings to the specific context of AI agent governance, where the failure modes have a direct structural mapping to the agent behavioral taxonomy.

Observed human-side failure modes mapped to the behavioral pattern taxonomy:

Agent Pattern	Human Equivalent	Operational Example
BP-001: Inference Over Execution	Accepting AI output without verifying its evidence base	An operator receives an AI-generated assessment and treats it as validated analysis without checking whether the AI accessed the primary source material
BP-003: Governance Phase Skip	Approving phase transitions without reviewing the evidence	An operator approves a gate transition based on the impact statement summary without examining the underlying work product
BP-004: Scope Creep	Not flagging when agent output exceeds the authorized scope	An operator accepts expanded deliverables because they appear useful, without verifying they were within the approved work order
BP-005: Completion Without Verification	Treating agent completion signals as proof of correctness	An operator marks a workstream as complete because the agent reported completion, without independently verifying the deliverables
BP-007: Selective Reporting	Focusing on some agent outputs while skipping others	An operator reviews high-priority workstreams closely but approves routine work with less scrutiny, creating a blind spot for drift in routine operations
BP-008: Authority Assumption	Treating AI analysis as expert judgment	An operator defers to an AI's assessment of a domain where the AI has no demonstrated competence, because the output is well-structured and confident

6.4 Scalable Oversight Boundary

The human oversight model described in this framework assumes sufficient operator competence in the domain being evaluated. The mitigation mechanisms in §6.4 — structured decision surfaces, impact statements, evidence requirements — address cognitive burden and attention failures. They do not address a distinct problem: as AI agent capability in specialized domains exceeds operator competence in those domains, phase-gate oversight quality can degrade for epistemic rather than cognitive reasons. An operator evaluating architectural output in a domain where the agent is more capable faces an oversight failure mode that structured formatting cannot resolve.

This framework is validated for contexts where operators have relevant domain competence. Organizations planning to scale to domains or capability levels where competence asymmetry may arise should supplement this framework with additional oversight approaches (external expert validation, interpretability tooling, AI-assisted

review) before that scaling occurs. This is an acknowledged open problem in scalable oversight research (Christiano et al., 2018; Irving & Askell, 2019), not a framework deficiency.

6.5 Implications for Framework Design

The human drift observation has a design-level consequence: **governance systems must protect against drift on both sides of the human-agent boundary.** A framework that assumes the human is always the reliable side has a structural gap equivalent to assuming any single safety layer is infallible — a violation of Principle 2 (Defense Layers Must Fail Independently).

The phase-gated lifecycle addresses this through five structural mechanisms:

1. **Structured decision surfaces.** Impact statements compress complex decisions into standardized formats. The operator evaluates a structured risk assessment, not raw technical output. This reduces the cognitive conditions under which human drift occurs (Principle 8).
2. **Evidence requirements at every gate.** Gate approval requires documented evidence, not trust. The operator cannot approve a phase transition without reviewing what the gate check requires — the structure prevents the skip pattern (BP-003).
3. **Separated authority.** No single actor — human or agent — holds both keys (Principle 1). The agent proposes; a different actor approves. This structural separation means human drift in one role is caught by a different actor in a different role.
4. **Orientation diversity.** Different reviewers bring different orientations (Principle 4). An architecture reviewer and an engineering reviewer catch different failures. If the operator's attention drifts in one dimension, a differently-oriented reviewer provides coverage in that dimension.
5. **The incident-driven learning loop.** When human drift produces a governance failure, the failure is documented, analyzed, and used to improve the governance system (§8). The framework treats human governance failures with the same rigor as agent governance failures — both are inputs to methodology evolution.

The human drift observation does not undermine the operator's authority. It strengthens the case for structural governance. An operator who knows that human attention is a finite, fallible resource builds governance systems that protect against

their own cognitive limitations — the same way a safety engineer designs systems that protect against human error. This is the standard practice in every safety-critical discipline (Reason, 1990). AI agent governance should be no different.

---

7. Governance Maturity Model

The maturity model defines five levels of organizational readiness for AI agent governance and three authority levels that progress orthogonally to maturity.

Maturity Levels

Level 0: Ungoverned

Agents deployed without formal governance. No phase gates. No behavioral monitoring. No structured audit trail. Agent autonomy is implicit — not explicitly scoped or calibrated. Failures are discovered after impact, if at all. The organization cannot demonstrate governance to regulators, auditors, or customers.

Where most organizations are today.

Level 1: Reactive

Basic governance structures in place. Agent work follows defined phases with explicit transitions. A human operator approves phase transitions. Known behavioral failure patterns are recognized manually. Deviations are documented. Failures are detected after impact but analyzed and used to improve governance.

Key gap: Detection is reactive. The operator is the sole detector.

Level 2: Structural

Automated mechanisms detect known behavioral patterns during agent execution, before impact. Phase transitions are validated against plan requirements. Agent autonomy is informed by behavioral history. Structured impact statements accompany every phase transition. Independent validation complements operator oversight.

Key gap: Detection is limited to known patterns. Novel failure modes are not caught until they become incidents.

Level 3: Predictive

Trend analysis across validation logs, incident patterns, and behavioral metrics identifies emerging risk before specific incidents occur. The governance system anticipates failure modes rather than reacting to them.

Key gap: Governance parameters are static. Adaptation requires manual policy updates.

Level 4: Adaptive / Standard-Setting

Governance mechanisms adapt based on detected drift patterns and empirical performance data within operator-approved policies. The organization's governance program is independently assessable against published standards. The organization contributes to governance standard development — contributing patterns, methodologies, and operational data to the broader community. The governance methodology itself is a versioned, improving artifact.

This is the target state. Cross-environment validation is an open area for further research.

Authority Levels

Orthogonal to maturity, governance systems progress through three authority structures:

Authority Level	Description
Advisory	Governance policies exist. Agent compliance is expected but not structurally enforced.
Enforced	Execution requires structural authorization. Agents cannot advance work without passing defined gates.
Separated Authority	Multiple independent actors required for authorization. No single actor — human or agent — can both propose and approve a governance action.

The progression from Advisory to Enforced introduces structural controls. The progression from Enforced to Separated Authority introduces independence of authorization — the governance principle derived from Two-Person Integrity.

Self-Assessment Dimensions

Organizations assess their current maturity across five dimensions:

1. **Lifecycle Management** — From no formal lifecycle (L0) to adaptive lifecycle with continuous methodology evolution (L4).
2. **Behavioral Monitoring** — From no monitoring (L0) to adaptive detection with cross-environment pattern validation (L4).
3. **Trust Calibration** — From no calibration (L0) to empirically optimized autonomy boundaries within policy constraints (L4).
4. **Incident Management** — From no tracking (L0) to cross-organization incident pattern sharing and standard contribution (L4).

5. **Audit and Compliance** — From no audit trail (L0) to continuous independent attestation with governance certification (L4).

Regulatory Alignment

> ***Alignment guidance only — not a compliance certification.*** *The table below maps framework capabilities to regulatory requirements. These mappings are a general alignment guide and do not constitute compliance determinations. Applicability of EU AI Act requirements depends on Annex III high-risk classification for each specific deployment. This framework addresses deployer-side operational governance obligations; it does not address provider-side obligations under EU AI Act Arts. 13, 16, and 17 or Annex IV. Independent legal review is required before relying on these mappings for compliance purposes.*

Requirement	L0	L1	L2	L3	L4
Human oversight (EU AI Act Art. 14)	None	Basic	Structured	Adaptive	Adaptive with human drift controls
Risk management (EU AI Act Art. 9)	None	Incident-based	Pattern-based	Predictive	Adaptive risk boundaries
Record-keeping (EU AI Act Art. 12)	None	Basic audit trail	Structured audit	Independently attestable	Continuous independent attestation
NIST AI RMF GOVERN	Not addressed	Basic structures	Documented practices	Standard contribution	Standard-setting authority
ISO 42001 PDCA	Not addressed	Plan-Do	Plan-Do-Check	Full continuous PDCA	PDCA with cross-org validation

Note: This regulatory mapping provides a general alignment guide. Organizations should conduct independent legal review before relying on these mappings for compliance purposes, particularly for EU AI Act sub-article requirements. Capabilities described at L3 and L4 are extrapolated from the framework's operational trajectory and have not been demonstrated in the framework's primary operational context.

---

8. Failure-Driven Evolution

The framework improves continuously through a structured learning loop:

```
Incident observed
    → Root cause identified
    → Control introduced
    → Behavior changed
```

This loop has four important properties:

1. **Governance is empirical, not aspirational.** Every mechanism traces to a real failure. If a mechanism exists without a corresponding incident, it should be questioned.
2. **The incident log is the design document.** New governance mechanisms are justified by specific incidents, not by "best practice" claims.
3. **The methodology learns from production.** Each governance iteration is driven by a specific identified gap, not by theoretical analysis.
4. **Behavioral patterns accumulate.** As more organizations operate governed agent fleets, the library of documented behavioral failure modes grows. Each new pattern improves governance for all practitioners.

The Eight Behavioral Failure Categories

The framework documents eight categories of behavioral failure modes observed in production multi-agent operations. Each pattern represents a class of agent behavior that produces work appearing correct on the surface but violating governance principles in ways invisible to traditional observability.

Category	Name	Default Severity
BP-001	Inference Over Execution	Critical
BP-002	False Blocker Reporting	High
BP-003	Governance Phase Skip	High
BP-004	Scope Creep	Medium
BP-005	Completion Without Verification	High
BP-006	Work Order Contamination	Critical
BP-007	Selective Reporting	Medium
BP-008	Authority Assumption	Medium

These patterns differ from simple errors in important ways: their output appears correct, they require governance-aware monitoring to detect, their root cause is structural agent behavior rather than technical defect, and they recur systematically across agents and contexts. The patterns were extracted from eleven documented incidents with full forensic analysis. The taxonomy is not exhaustive — additional patterns should be expected as agent capabilities and deployment contexts evolve.

Detailed descriptions and provenance for each pattern are published in the Behavioral Pattern Taxonomy (framework/agent-failure-patterns.md).

---

9. Limitations

This framework was developed in the context of a governed software production platform. The following limitations should be considered:

- **Single-context empirical base.** The behavioral patterns and governance mechanisms documented here were derived from one operational environment. Cross-environment and cross-architecture validation is an open area for further research.
- **Maturity model generality.** The maturity levels describe a general progression. Specific capability requirements at each level may vary by operational context, deployment scale, and regulatory domain.
- **Trust calibration specificity.** Trust calibration inputs and outputs are described as principles. Specific calibration mechanisms are implementation-dependent.
- **Taxonomy completeness.** The behavioral pattern taxonomy documents eight categories observed to date. It is not exhaustive. Additional patterns should be expected as agent architectures and deployment contexts evolve.
- **Risk reduction, not elimination.** This framework reduces the risk of governance failures in AI agent operations. It does not prevent all failures. Governance is a continuous discipline, not a one-time certification.
- **Privacy and data protection.** AI agents operating under this methodology will frequently access, process, and generate personal data. GDPR, CCPA, and sector-specific data protection obligations apply to such processing independently of this operational governance methodology and are outside its scope. The AUDIT phase and evidence trail can support data protection compliance documentation, but organizations must address data protection requirements through a parallel compliance program. Where agents influence decisions with legal or significant effects on individuals, GDPR Article 22 (automated decision-making) may apply; the operator-as-router model's evidence requirements and phase gate approvals

provide a structural basis for meaningful human involvement, but context-specific legal assessment is required.

---

10. Conclusion

AI agent governance is an operational discipline — distinct from model governance, security, and compliance — that addresses how autonomous agents perform real work under delegated authority. This framework provides the principles, lifecycle structure, and maturity model for that discipline. It was derived from production operations and improves through continued operational experience.

---

11. Further Reading

- MOIAS Methodology (framework/governance-lifecycle.md) — The complete phase-gated governance lifecycle
- Behavioral Pattern Taxonomy (framework/agent-failure-patterns.md) — Eight documented failure categories with full provenance
- Governance Maturity Model (framework/maturity-model.md) — Organizational readiness assessment with regulatory mapping
- Glossary (framework/glossary.md) — Canonical definitions for AI agent governance terms
- The Governance Gap (insights/governance-gap.md) — Why observability, security, and compliance are necessary but insufficient
- 8 Ways AI Agents Fail (insights/8-ways-agents-fail.md) — Behavioral failure modes invisible to standard monitoring
- DeepMind Delegation Paper Analysis (insights/deepmind-delegation.md) — Independent convergence from Google DeepMind research

---

12. References

1. Reason, J. (1990). *Human Error*. Cambridge University Press.
2. Reason, J. (1997). *Managing the Risks of Organizational Accidents*. Ashgate.
3. Dekker, S. (2011). *Drift into Failure: From Hunting Broken Components to Understanding Complex Systems*. Ashgate.
4. Vaughan, D. (1996). *The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA*. University of Chicago Press.

5. Rasmussen, J. (1997). “Risk management in a dynamic society: a modelling problem.” *Safety Science*, 27(2-3), 183-213.
6. Boyd, J. (1976). “Destruction and Creation.” Unpublished paper.
7. Weick, K.E. & Sutcliffe, K.M. (2007). *Managing the Unexpected: Resilient Performance in an Age of Uncertainty*. 2nd ed. Jossey-Bass.
8. U.S. Department of Defense. (1962). Two-Person Integrity (TPI). AFI 91-104.
9. Tomašev, N., Franklin, M. & Osindero, S. (2026). “Intelligent AI Delegation.” arXiv:2602.11865.
10. Parasuraman, R. & Manzey, D.H. (2010). “Complacency and Bias in Human Use of Automation: An Attentional Integration.” *Human Factors*, 52(3), 381-410.
11. Lee, J.D. & See, K.A. (2004). “Trust in Automation: Designing for Appropriate Reliance.” *Human Factors*, 46(1), 50-80.
12. Parasuraman, R. & Riley, V. (1997). “Humans and Automation: Use, Misuse, Disuse, Abuse.” *Human Factors*, 39(2), 230-253.
13. Cummings, M.L. (2004). “Automation Bias in Intelligent Time Critical Decision Support Systems.” AIAA 1st Intelligent Systems Technical Conference, AIAA 2004-6313.

---

13. Version History

Version	Date	Author	Description
1.0.0	2026-02-26	John J. McCormick	Initial canonical framework document
2.0.0	2026-02-28	John J. McCormick	Metadata standardization; citation, version, and PDF fields moved to frontmatter

---