International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, Delaware, USA
Volume 13– No. 2, April 2026 – www.ijais.org

GenAI Copilot as an "Innovation Operating System": Controls, Learning Loops, and Integration Prerequisites

Basil Obute
PhD Student, Information Technology Department, University of the Cumberlands, Kentucky, USA
ORCID ID - 0009-0001-1348-4572

Kingsley C. Ugwu
PhD Student, Artificial Intelligence, University of the Cumberlands, Kentucky, USA
ORCID ID - 0009-0002-4061-5563

Nzeribe A. Okeh
PhD Student, Information Technology Department, University of the Cumberlands, Kentucky, USA
ORCID ID - 0009-0008-6358-1718

ABSTRACT
Enterprise GenAI copilot programs most commonly fail not because of poor model capabilities, but because businesses lack the necessary operating system for integrating and managing the key elements of a GenAI copilot, including: (i) data lineage and data retrieval provenance, (ii) tool integration and access control, (iii) governance-as-code (i.e. the ability to define and manage business rules through code), (iv) end-to-end traceability and approval processes, (v) learning loops (the ability to utilize and measure user activity and incidents as a means of improving the overall capability of GenAI). Drawing on sociotechnical systems, innovation systems, and Responsible AI research, we synthesize these into a 5-layer Innovation Operating System (IOS) and propose five falsifiable propositions (P1–P5) examining how IOS maturity, governance density, and learning loop maturity affect enterprise GenAI copilot performance. The study provides a reference implementation measured by: (a) IOS layer maturity, (b) a task-class governance density index, and (c) three performance proxies - Innovation Adoption Rate, Control Incident Frequency, and Retrieval Robustness Score. A replication package for this study includes a blueprint for all elements (schemas, queries, rubrics, notebooks, and a synthetic log generator).

General Terms
Artificial Intelligence, Innovation Management, Enterprise Systems, Knowledge Governance, Responsible AI

Keywords
Generative AI, Copilot, Information Systems, Governance, Data Lineage, Traceability, Evaluation, Design Science, Responsible AI

1. INTRODUCTION

Generative AI copilots can now serve as Enterprise Information Systems (EISs) for Mediating Work (McKinsey Global Institute, 2023) in many organizations. EISs provide organization-relevant information, enable tools and applications, assist in generating content drafts, and even initiate decisions. The problem of developing generative AI copilots as EIS is not related to the models themselves but to the way to connect them to the governable data, to enforce the policies, to keep track of the origin of the content (provenance), and to continue to develop and improve the system, without adding additional compliance or security risks.

Generative AI copilots must be regarded as Innovation Infrastructures, particularly as an Innovation Operating System (IOS), which is a sociotechnical stack of five interconnected layers: (L1) Data Lineage and Retrieval Provenance, (L2) Tool Integration and Identity-Aware Access Control, (L3) Governance-As-Code and Control Design, (L4) Traceability, Approvals, and Auditability, and (L5) Learning Loops Connecting Usage, Incidents, and Evaluation into Iterative Improvement.

This paper offers four practical contributions. First, it has developed a concise reference model for innovation systems that includes controls for responsible AI. Second, it has produced five testable propositions about maturity (P1), governance density (P2-P3), and learning loops (P4-P5). Third, it has described a measurement pipeline that will allow an organization to assess the success of its program by creating a computable index of governance density and a set of reproducible proxies for program performance. Fourth, it has presented a reference architecture for organizations to implement this work and a template for replication while protecting the organization's privacy.

There are two enterprise-specific constraints for adopting the Innovation System (IOS) framework. The first constraint is that, since copilots have access to multiple knowledge bases (wikis, tickets, contracts, etc.), the policies governing access to each knowledge base do not typically overlap. In other words, without a line of origin (lineage) or source, RAG could generate outdated or unauthorized content. The second constraint is that the risk surface includes potential data breaches, hallucinations of references/citations, and uncontrolled tool execution (Weidinger et al., 2023) - all of which are operational in nature and derive from how the organization integrates its workflow into the model rather than the model's behavior alone.

As a result, the IOS layers were designed to be quantifiable. Each layer of the IOS (registries, policies, logs, workflows, and assessment tools) requires that the features within each layer be evaluated against the compliance level of certain standards. Thus, organizations can evaluate their current state, identify where they should allocate their engineering resources (e.g., access control, auditing, and learning from events), and shift their focus to improving the quality of their responses rather than spending time on diminishing returns elsewhere.

Paper Roadmap: Section 2 reviews the relevant research. Section 3 discusses the IOS and hypotheses. Sections 4-5 discuss the methods used to develop the metrics. Sections 6-7 discuss the architecture and the evaluation methodologies, specifically, they provide a case study of governance density and a sensitivity analysis. Sections 8-10 discuss the implications, limitations, and conclusions.

This study does not propose another maturity model. Although IOS uses ordinal maturity scores for each layer (Absent/Partial/Operational/Instrumented), IOS differs from

conventional stage-gate maturity models (CMMI, P3M3) in three significant ways. First, IOS is a systems stack rather than a capability ladder: organizations may demonstrate high maturity in one layer (for example, L3-Governance-as-Code) and low maturity in another (for example, L1-Lineage), without implying any developmental sequence or prerequisite based solely on layer maturity. Second, IOS layer maturity is verified through artifacts rather than self-reporting, making it independently verifiable. Artifact presence is treated as necessary but not sufficient: a versioned lineage registry that is not actively queried during retrieval, or a policy engine whose rules are never enforced in production, both score as Partial rather than Operational. Third, IOS measures governance density and learning-loop maturity independently of layer maturity. Therefore, the five propositions can be tested and contradicted separately, a property that conventional maturity models generally do not support. The paper presents an operational model for data governance through: (i) a reference architecture using typical enterprise components; and (ii) a measurement framework that links operational governance decisions with measurable effects on adoption, incidents, and retrieval robustness.

2. RELATED WORK

According to Sociotechnical Systems Theory (Baxter & Sommerville, 2011; Trist, 1981), an organization's success depends on both Technical Subsystems (employees' tools and workflows) and Social Subsystems (employees' behaviors defined by roles, incentives, and workplace norms) operating at optimal levels. In this case, the technical subsystems of enterprise GenAI copilot are the retrieval pipeline, policy enforcement points, and audit logging. The social subsystems include the approval, incident response, and learning processes of the GenAI copilot. Until now, IOS has separated these sociotechnical systems through a governance design with a separate control plane rather than through a separate post-deployment policy document.

Research into Innovation Systems (Edquist, 2005; Lundvall, 1992; Nelson, 1993) demonstrates that the output of innovation arises from the interaction of three types of elements: institutions, infrastructure, and learning processes. This area of research has also led to multiple streams of research on knowledge governance (Kogut & Zander, 1992; Grant, 1996) absorptive capacity (Cohen & Levinthal, 1990; Flatten et al., 2011) and dynamic capabilities (Eisenhardt & Martin, 2000; Teece et al., 1997), demonstrating that firms vary in their ability to recognize opportunities and to use coordinated change to take advantage of those opportunities and to develop new patterns of behavior in their day-to-day operations. These streams of research support IOS Layer 5 (Learning Loops), which uses telemetry to systematically continually improve the firm's capabilities and avoid ad hoc adjustments to the language used to prompt the AI.

There have been many discussions regarding the operational governance of Enterprise AI within the literature, including policy-as-code (European Commission, 2024; Floridi et al., 2018; Lemley & Casey, 2021; NIST, 2023; Sag, 2023; UNESCO, 2022) and access propagation (Cihon et al., 2021; Mittelstadt et al., 2016) as well as on-demand audit evidence (Sag, 2023; UNESCO, 2022). Data Governance and Lineage research have also defined similar concepts and tools (Halevy et al., 2016); however, to date, these frameworks have largely been separated from the GenAI systems that generate final output from multiple input sources, leading to an unmet need for the IOS Layer's L1–L3.

Most empirical research on GenAI in the context of knowledge work focuses on the productivity and quality effects of GenAI systems for relatively narrow and well-defined tasks, including, but not limited to, drafting, coding, and customer support (Dell'Acqua et al., 2023; Brynjolfsson et al., 2023; Noy & Zhang, 2023). This body of research is useful and informative, however it assumes a rigidly defined set of tasks and conditions (e.g., "unlimited" access to data) and ignores many of the most common issues associated with large-scale GenAI deployments in enterprises, which include, but are not limited to, lack of transparency regarding the origin of retrieved information ("retrieval provenance"), lack of identity-aware access controls for the system and its tools ("identity-aware access"), and lack of control over what actions can be taken by the AI tools ("tool execution control") (Lewis et al., 2020; Schick et al., 2023; Yao et al., 2023; Perez & Ribeiro, 2022). These issues suggest that the governance and design of the system will be significant determinants of the success or failure of a GenAI deployment, and thus motivated the development of the IOS framework.

The literature therefore reveals a consistent problem: each research stream provides a partial account of enterprise AI deployment, but none provides a complete framework that integrates sociotechnical learning processes, engineered governance structures, and operational instrumentation into a single measurable architecture for scaling, auditing, and improving GenAI copilot programs.

The search for relevant literature employed an adaptation of the PRISMA protocol to perform a literature search in three categories of research: (1) Sociotechnical Systems and Innovation Systems, (2) Evaluation of GenAI Copilot Deployments, and (3) Information Systems Governance and Compliance. The search terms used to identify relevant literature were: Generative AI, Copilot, Retrieval-Augmented Generation (RAG), Data Lineage, Policy-as-Code, Audit Trail, MLOps, AI Governance, and Organizational Learning. Following completion of the literature searches, the literature was filtered to identify only those articles reporting on GenAI deployments at the enterprise level and providing sufficient detail to enable organizations to make operational decisions about their GenAI deployments.

The literature review also provided information regarding the layer definitions for the IOS framework and the selection of measures; each layer of the IOS framework is supported by at least one observable artifact and at least one documented failure mode in the literature (i.e., unauthorized retrieval of data; unverifiable citation of source material; uncontrolled actions taken by the AI tool).

Two significant systemic gaps were identified through this literature review. First, governance has always been theoretically decoupled from system design. While many theoretical frameworks provide guidance for designing governance systems in practice, they do not provide guidance for identifying enforceable control points within the deployed system or for collecting audit evidence to demonstrate that the control points are being enforced as intended. These two gaps motivated the development of IOS Layers L3 (Governance-as-Code) and L4 (Traceability and Auditability). The authors' prior work on co-designed fairness checklists (Madaio et al., 2022) illustrates the organizational barriers to translating governance intent into systematic practices and provides support for the use of enforceable, artifact-based approaches.

The majority of prior research assessing the effectiveness of copilot systems assesses the copilot systems as static tools and does not account for the continuous improvement process that

occurs subsequent to the deployment of a copilot system (i.e., changes to user prompts, changes to tools, changes to policy governing the copilot system, and changes to how the copilot system accesses data). The IOS model differs from prior frameworks in that it views both governance and learning as primary, measurable aspects of the architecture rather than secondary concerns applied after deployment.

3. IOS MODEL AND PROPOSITIONS

IOS identifies the minimal number of elements that an enterprise must have to deploy, administer, and continually improve Gen AI Copilot technology in an enterprise environment. Traditional maturity models measure the level of growth an organization has achieved, while IOS uses a system-based stack to identify all the components that comprise the deployment of Gen AI technology. Each layer of the stack will have established boundaries and interfaces, allowing each element to be quickly identified and assessed.

Layer 1 of the IOS architecture, is concerned with addressing the two primary issues faced by data scientists: 1) establishing a method for identifying all knowledge-based artifacts (document ID, version, owner, etc.) that can be audited for long periods of time after the fact, and 2) tracking the provenance (which sources were used, what versions of those sources, and why) for each piece of context that was used during an execution. L1 is primarily concerned with building a stable data foundation for future applications.

Layer 2 of the IOS architecture connects the Copilot (or other AI assistants) to enterprise systems such as search, ticketing, CRM, and code repositories. Layer 2 contains the various components needed to enable identity-aware authentication and authorization. As a result, a Copilot (or other AI assistant) will only be able to perform actions and/or obtain information that the user has access to, and Layer 2 will capture the identity of the tool invoked and the parameters passed to it for auditing purposes.

Layer 3 of the IOS architecture enables Governance-as-Code. Instead of implementing and managing controls through multiple methods (traditional ITIL processes, manual scripting, etc.), Layer 3 enables expressing organizational governance requirements as enforceable rules and checks. These rules and checks could include PII redaction, restrictions on retrievable topics, model routing based on risk level, citation requirements, and safe tool usage limitations, to name just a few. In addition to focusing on expressing governance requirements as enforceable rules and checks, Layer 3 also addresses how these rules and checks are versioned, tested, and deployed as software.

Layer 4 of the IOS Architecture, Traceability, Approvals, and Auditability, is responsible for creating immutable records of all interactions, retrievals, tool calls, and policy decisions, and for enforcing workflow gates when required (e.g., for high-risk outputs). Layer 4 supports compliance audits and provides accountability, eliminating the need for manual documentation.

The last layer of the IOS Architecture (Layer 5) addresses Learning Loops. A learning loop is the conversion of telemetry into actionable improvements across the IOS stack. Examples of learning loops include, but are not limited to, offline evaluation of performance metrics, online monitoring of performance metrics, incident triage, root cause analysis of incidents, and controlled changes to prompts, retrievals, tools, and policies. The ultimate goal of Layer 5 is to ensure that the IOS improves over time and does not remain static. One of the main reasons most prior enterprise AI deployments have failed is the inability to prevent "static governance" (controls that do not evolve) and "static capability" (systems that do not learn from failures).

Layer 5 of the IOS is based on two related bodies of theory. First, Argyris and Schon (1978) differentiate between single-loop learning (detection and correction of errors using pre-existing rules) and double-loop learning (revision of the underlying rules and policies that generated the error); IOS L5 is specifically designed to facilitate double-loop improvement by relating incident post-mortems to changes to policy and retrieval. Second, effective learning loops at scale require psychological safety (a shared understanding among team members that they will not experience interpersonal risk as a consequence of raising concerns, reporting incidents, or challenging governance policies) (Edmondson, 1999), which is why L5 makes incident reporting and participation in post-mortems a formalized and blame-free process rather than an informal one. Techniques such as reinforcement learning from human feedback (Christiano et al., 2017; Ouyang et al., 2022) are specialized, model-specific instances of this larger feedback principle; IOS L5 generalizes it across the entire program, including improvements to policy, retrieval, and tooling.

Table 1. IOS layer reference: function, artifacts, and failure mode addressed

Layer	Name	Primary Function	Key Artifacts	Failure Mode Addressed
L1	Data Lineage & Retrieval Provenance	Create versioned identifiers for each knowledge object and maintain an interactive log of every time information is retrieved information.	Registry for lineage; Logs for retrieval; Version ID.	The presence of stale or unauthorized content in an enterprise RAG (Retrieval-Augmented Generation) responses.
L2	Tool Integration & Identity-Aware Access	Connect the copilot to enterprise systems (search, ticketing, CRM, code repositories) with identity-aware authentication and authorization, ensuring the copilot can only access data and invoke	Adapters to connect to tools; Checks for entitlements; Event log of tool calls.	Unauthorized access to data; Unchecked execution of tools.

		tools the authenticated user is permitted to use, and capturing the identity and parameters of every tool call for auditing.		
L3	Governance-as-Code	Express governance rules as versioned, testable, CI/CD-deployed policy code evaluated on all inputs and outputs.	Engine for policy; Repository for rules; Test harness; Pipeline for CI/CD.	Policies that cannot be enforced; Policies that are not documented; PII (Personally Identifiable Information) is leaked.
L4	Traceability, Approvals & Auditability	Store immutable records of interactions that occurred within the system; Have users approve any high-risk actions taken by the system; Export evidence of audits.	Immutable record of interactions; Approval workflow; Audit export.	No accountability; Unchecked high-risk actions taken by the system.
L5	Learning Loops	Use telemetry and incident data to implement controlled changes to prompts, policies, retrieval mechanisms, and tools.	Eval harness; Dashboards for monitoring; Process for post-mortems; Log of changes.	Static governance; Drift in capabilities; Unlearned failures.

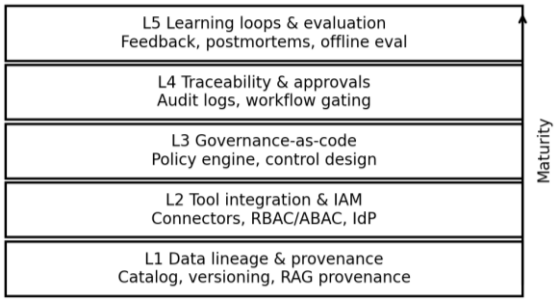

Figure 1. Innovation Operating System (IOS) layers and maturity direction

Three dimensions of IOS can be measured independently: (a) the maturity of each IOS layer, (b) governance density, and (c) learning-loop maturity. This distinction avoids conflating "more controls" with "a better operating system" and enables each proposition in this paper to be tested and falsified separately.

Table 2. Construct a dictionary and operationalization (summary)

Construct	Definition	Primary evidence artifacts	Computation/scale	Interpretation notes
IOS Layer Maturity (Layers L1 – L5)	The degree to which each layer of the IOS stack has been implemented, working, and is being enforced in your production environment.	Registry/catalog; access control flow-down; policy/rule tests; immutable trace history; evaluation runs / post mortem runs.	The program maturity score is a vector of these scores (one for each layer) plus a summary index.	Scores will be determined based upon artifacts (not self-reported), and layers have dependencies.
Governance Density (by Task Class)	Control Intensity applied to a recurring task class (Gates + Enforced Rules + Automation + Latency + Exceptions).	Approval Workflow Events; Policy Decision Logs; Control Configs; Latency Metrics; Exception Registers.	A normalized index in [0, 1], calculated from observable components (Algorithm 1), with justification for weights, and sensitivity testing	For many tasks, higher governance density does not mean better performance; an inverse-U shape or diminishing returns with respect to throughp

			performed.	ut is common.
Learning Loop Maturity	Cadence and Reliability with which Telemetry and Incidents are driving Controlled Improvements.	Change Logs; Monitoring Dashboards; Eval Harness; Release Notes; Post Mortems for Incidents/Prompts/Policies/Retrieval.	0-3 ordinal Scale (Ad Hoc to Continuous) and may be used as a Moderator in Analysis.	Learning Loop Maturity should focus on Closed Loop Improvement, not just on whether monitoring tools are in place.
Innovation Adoption Rate (IAR) Proxy	Throughput/Adoption Proxy: The Extent Tasks are moving through Copilot-Assisted Stages.	Workflow State Transitions, Interaction Records, Task Metadata.	Progression Counts are normalized by the Eligible Population over the Time Window.	Proxy for Adoption /Throughput; Not Direct Business Value.
Control Incident Frequency (CIF/Qtr) Proxy	Number of Control-Relevant Incidents per Quarter (E.G., Policy Violation, Leakage, Unsafe Tool Action).	Audit Traces, Severity Taxonomy, Root Cause Labels, Incident Tickets.	Incidents per Quarter, Per Task Class, Optionally Severity Weighted.	Consistent Taxonomy Required; Changes in Reporting May Confuse Trends.
Retrieval Robustness Score (RRS) Proxy	Quality/ Robustness Proxy for Retrieval Grounding Under Perturbation and Freshness Constraints.	Grading Rubric; Inter-Rater Reliability Report; Offline Eval Set; Retrieval Traces.	Standardized Rubric Score; IRR (Alpha/ICC) Report; Sampling Method Description.	Designed to Detect Provenance/Freshness Failures; Complements IAR/CIF.

3.1 Governance density and task classes

Not all forms of governance are equivalent. Different task risks require different governance approaches. This paper identifies "task classes" as recurring work types with distinct risk profiles, including Internal Q&A, Customer Messaging, Regulatory Drafting, and Code Changes. Governance density is calculated for each task class based on observable elements such as gates, rules, automation level, role separation, latency, and exceptions. The resulting governance-density index enables comparison across task classes and supports analysis of control trade-offs.

3.2 Propositions

P1 (IOS Maturity): When considering the same task mix, there will be a positive correlation between IOS maturity and various proxies for performance (i.e., throughput/adoption and/or robustness). [See P3 for the moderation effect of L5 maturity on this relationship.]

P2 (Governance Density Trade Off): There is an inverse U-shaped relationship between governance density and throughput/adoption for most task classes; the exception is that incident frequency decreases with increases in governance density. In other words, the optimal level of governance density will depend on the risk inherent in each task class.

P3 (Learning Loop Moderation): L5 learning loop maturity moderates the relationship between IOS maturity and performance (P1), such that the positive effect of IOS maturity on IAR and RRS is stronger when L5 is active than when it is absent. Additionally, L5 maturity will strengthen the IOS-performance relationship by reducing governance costs through automation and increasing capabilities through evaluations that lead to updated processes.

P4 (Necessary conditions for regulated scaling): For both regulated and high-stakes domains, L1 lineage/provenance and L2 Identity-Aware Access are necessary preconditions for continued scalability; the absence of these two factors should correspond to either decreased adoption and/or increased incidents.

P5 (Traceability/Approvals): L4 Traceability and Approval Workflows reduce incident frequency and improve auditability; however, they can also increase cycle completion time, especially for medium-risk work.

3.3 Falsification and boundary conditions

Each proposition states conditions under which it could be contradicted or refined. For example, if high governance density always correlates with high throughput for low-risk tasks and no automation is present, this pattern would challenge both P2 and P5. If systems lacking strong L1/L2 characteristics continue to scale safely within regulated domains, this pattern would challenge P4. The hypotheses apply to enterprise copilots that include both retrieval and tooling functions; they do not apply to systems limited to pure generative creativity. The propositions also address program-level performance proxies rather than direct business value.

Table 3. Propositions summary: statement, key variables, and falsification conditions.

Proposition	Statement	Key Variables	Falsification Condition
P1 (IOS Maturity)	When learning loops are active, there is a positive relationship between an organization's IOS maturity and its performance metrics.	The relationship between IOS maturity and performance metrics arises from the interactions among IOS maturity, IAR (i.e., learning loops), RRS (i.e., learning loops), and L5 presence	High IOS maturity, with no L5 results, yields high performance metrics.

		(i.e., learning loops).	
P2 (Governance Density)	There is an inverse U-shaped relationship between governance density and an organization's throughput; however, the error rate decreases at a constant rate as governance density increases.	GD per task class; IAR; CIF/Qtr.	Regardless of the task's risk level, increasing an organization's governance density will always improve its throughput.
P3 (Learning Loop Moderation)	Learning loop maturity will enhance the relationship between IOS maturity and performance metrics by creating a mechanism for governance cost reductions using automation.	L5 maturity; GD; IAR; CIF/Qtr.	The maturity of L5 will not affect the relationship between GD and IAR.
P4 (Necessary Conditions)	To successfully scale regulated tasks without increasing error incidents, L1 and L2 must be operationalized.	IAR; CIF/Qtr in regulated task classes; L1/L2 presence.	Without operationalizing L1 and L2, regulated tasks can be scaled without adverse effects.
P5 (Traceability /Approvals)	Adding L4 to an organization will reduce error incidents and improve auditability; however, adding L4 to an organization with medium-risk tasks will also increase cycle-time latency.	CIF/Qtr; cycle-time latency; L4 presence.	Adding L4 to an organization does not reduce error incidents.

4. METHOD

The authors used a comparative case-study methodology, emphasizing theory-building through a design-oriented, exploratory approach (Yin, 2018; Eisenhardt, 1989; Miles et al., 2020). This approach is appropriate when developing new theoretical frameworks from emerging practices where controlled experiments are not yet feasible. Rather than studying a single deployment, the authors treated each copilot program as the unit of analysis - encompassing the deployed assistants, their system integrations, their governance configurations, and their evolution across multiple quarters.

This research study utilizes the Design Science Research methodology, within the context of the IS research paradigm to create artifact sets consisting of four artifacts: (1) a reference architecture for enterprise GenAI copilots, (2) an event schema that can capture and integrate events from multiple copilot interaction streams into a single format, (3) a governance-density algorithm which identifies areas of high density in terms of both governance and technological complexity, and (4) a measurement pipeline which aggregates the output of the measures and provides a means of monitoring performance over time; and assesses the utility of these artifact sets using two evaluation strategies: comparative pattern matching among deployment sites and a reproducible synthetic benchmark designed to test the computational robustness of each measure to be used in assessing copilot deployments. The four artifacts and two evaluation strategies described above are operationalized through the case sampling, data collection, and coding procedures described below.

4.1 Case sampling

Cases were not randomly selected for this study. The authors looked for companies with varying infrastructure maturity and differing levels of control over AI governance, and ensured that both regulated and non-regulated work was included. Each company is responsible for handling multiple task types; therefore, the authors can compare governance settings within a single company, avoiding the need to constantly compare across companies.

4.2 Data source

Four artifact types were collected and triangulated across cases: (1) operational telemetry - interaction logs, retrieval traces, tool call records, and policy decision logs; (2) governance artifacts - policy-as-code repositories, approval workflows, and role/entitlement matrices; (3) incident and audit records - trouble tickets, post-mortems, and audit exports; and (4) evaluation artifacts - rubric scorecards and offline benchmark results. No single artifact type was treated as definitive; findings were accepted only when consistent across multiple artifact types within the same case.

4.3 Coding and evaluation

Layer maturity is measured through quantifiable artifacts rather than self-reported measures. For L1, raters assessed whether a versioned lineage registry existed and whether the provenance of all retrievals had been recorded for each interaction. For L3, raters determined whether a policy engine enforced policy-based decisions (i.e., in real time) or whether the policy rules had only been documented in policy format. For L4, raters determined whether logs were set up as an immutable, append-only store. For L5, raters evaluated whether evaluations occurred on a defined cadence, whether feedback from incidents was systematically translated into changes, and whether those changes were managed using version-controlled change management processes. Governance density was evaluated per task type across all normalized component types. Layer maturity was assessed using closed-loop evidence, including regular evaluation cadences, incident-to-change linkage, and controlled release procedures.

A cross-case pattern-matching methodology was employed to determine whether the observed data supported or refuted each proposition. Where longitudinal artifacts were available, the study completed within-case temporal analysis to document how each case developed over time. Where no patterns were found, that absence was documented to contextualize the limitations of the propositions. Because the objective was theory development with practical application, the study prioritized measurement reliability and artifact transparency above statistical generalization (Yin, 2018).

Telemetry use is governed by enterprise confidentiality requirements. When raw logs cannot be disclosed because of

confidentiality risk, aggregate metrics can be provided instead of individual log entries. The replication package also provides synthetic data generators that create structurally comparable data without confidential elements.

4.4 Reliability of scoring

Each analyst applied the layer maturity rubric independently and scored each case individually. Discrepancies were resolved by reviewing evidence supporting each rating, not by negotiating a score. Across the four cases, pairwise analyst agreement on layer maturity tier (Absent/Partial/Operational/Instrumented) reached 89% before adjudication and 100% after evidence review, indicating that the rubric criteria were sufficiently specified to support consistent scoring. When feasible, governance density component score values were calculated based upon artifact data (e.g., the timestamp for approvals was used to compute latency; the repository of enforceable policy rules was used to determine the number of enforceable policy rules). Before finalizing the scoring tasks for each case, all RRS raters underwent calibration using a single "gold" set of artifacts representing the full range of retrieval quality observed across cases; during calibration, discrepancies between raters and the gold set were used to refine the rater's guide. Calibration continued until all raters achieved within-one-point agreement on 90% of calibration items before scoring live cases. All scoring materials, calibration sets, and adjudication records are included within the replication package.

The IOS maturity model is formative; each layer contributes unique information and does not replace any other layer. The governance-density proxy is task-specific, and the same principle applies to the three performance proxies: Innovation Adoption Rate, Control Incident Frequency, and Retrieval Robustness Score. Therefore, the study explicitly maps constructs to proxies and does not treat any single score as representing the entirety of innovation outcomes.

Many variations across cases, including task mix and organizational incentives, may confound baseline process maturity and case characteristics. The analysis attempted to minimize this risk by stratifying results by task type and emphasizing within-case comparisons, especially when natural breaks in longitudinal data resulted from a policy or architectural upgrade.

5. MEASURES

The measures that have been developed allow for calculation using Enterprise Artifacts and will also serve as a means of providing learning and auditing support.

5.1 IOS maturity

Each layer (L1 through L5) of IOS has defined maturity levels based upon the availability and operational use of all required artifacts. For example: L1: Versioned knowledge objects + Provenance for retrieval; L3: Test Harness for Enforceable Policy Rules; L5: Routine Eval Cadence + Linkage of Change to Incident. The maturity levels were used as a comparison tool among the cases and as a prerequisite check before scaling.

5.2 Program performance proxy indicators

The Innovation Adoption Rate (IAR) - This is the progression from an initial pilot phase to scalable use and continued activity for each task classification, and is measured through metrics such as number of active users, number of recurring sessions, and task completion indicators; (II) The Control Incident Frequency per Quarter (CIF/Quarter) - This is the count of policy-related incidents on a quarterly basis which are then normalized against the volume of user-interactions (for example 10K interactions per quarter). The CIF/Quarter can also be segmented by incident severity. (III) The Retrieval Robustness Score (RRS) - A rubric-based evaluation of the accuracy of the sources retrieved, and the accuracy of the citations provided, using a stratified sample of interactions. Dimensions of the RRS include the relevance of the source(s) retrieved, the accuracy of the citations, and the system's resilience in responding to Adversarial or Ambiguous prompts.

RRS quality can be evaluated using reliability testing and sampling. Trained raters assessed RRS using a stratified random sample by task class and risk level. Inter-rater reliability (IRR) was assessed using Krippendorff's alpha (Krippendorff, 2004) and intraclass correlation coefficients (ICC; Koo & Li, 2016). Across the four cases, Krippendorff's alpha for RRS holistic scores ranged from 0.71 to 0.84, indicating substantial agreement; ICC (two-way mixed, absolute agreement) ranged from 0.73 to 0.87. Rater disagreements exceeding one scale point were adjudicated by reviewing retrieval traces, with final scores recorded in the calibration log included in the replication package. The package also provides rater guidance based on content-analysis literature (Hsieh & Shannon, 2005) and a full calibration protocol.

The Governance Density (GD) for each task type was calculated as the weighted average of normalized governance metrics. These normalized metrics include the number of formally defined gateways, the degree of role separation, the number of formally defined rules, the number of automated checks, the median time to approve a request, and the exception rate. Each metric was normalized to enable comparison across organizations by setting a ceiling on the metric (for example, no more than 3 formal gateways).

Expert-weighted and equally weighted results are provided through a stakeholder elicitation process using a 100-point allocation scale. Alternative weight combinations varying by +/- 20% are also included to limit arbitrary weight assignment. Spearman's rho (rho) was used to assess the stability of governance-density rankings and the consistency of proposition-related conclusions across weighting schemes.

An example of computing the Governance Density (see Section 7) demonstrates how it is computed for a regulated task class and how automation reduces the effective density by reducing approval times and costs associated with exception handling.

6. REFERENCE ARCHITECTURE

The Reference Architecture Design uses IOS as a modular collection of services that can be used alongside the typical Enterprise Stack elements (Identity Provider, Data Catalog, Workflow System, Logging/Analytics Pipeline) to serve as a foundation for IOS service creation. The Reference Architecture Design was intended to be technology neutral -- the contribution provides the decomposition, interface definitions, and the Audit Data Model.

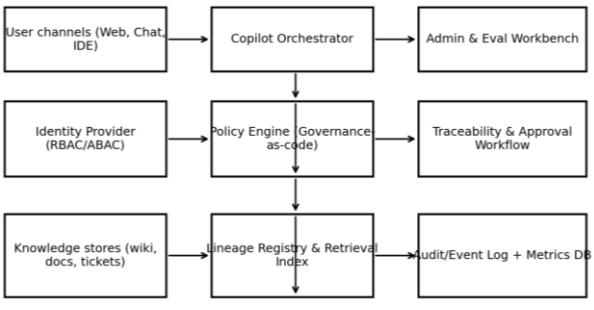

Figure 2. IOS reference architecture (control plane + data plane) and key services.

The Lineage & Provenance Registry (L1) will store a single, canonical identifier for each knowledge object, along with its version number, owner, retention class, and sensitivity tag. The LPR provides APIs to the Retrieval Systems to track which versions of objects they used to create their answers. Additionally, the LPR will support "Provenance Recall" during audits and incident investigations.

The Connector & Orchestration Layer (L2) will develop adapters to connect to multiple tools (search engines, CRM systems, ticketing systems, etc.) and ensure that identity data is passed through these tools while providing least privilege access through the use of entitlements evaluated at call time, domain- or tenant-based context filtering, and sending tool-call events and results.

The Policy Engine (L3) will assess the effectiveness of governance-as-code policies for all input and output elements: (1) allow/deny determinations, (2) redaction actions, (3) citations, (4) routing models, and (5) restrictions on tool usage. Governance-as-code policies are versioned, testable, and deployed through Continuous Integration/Continuous Deployment (CI/CD) to provide an auditable control plane.

The Traceability and Approval Service (L4) stores an immutable history for every interaction within the system, including user inputs, retrieval traces, tool calls, and policy decisions. This service provides workflow gateways for high-risk actions requiring human approval, dual control, or peer review and exports audit evidence in common formats.

The IOS deployment strategy is phased (L1 through L5). In the initial phase, Layers One and Two instruments will be deployed, and identity-aware access and provenance logging will be created. In the second phase, Layer Three policy enforcement instruments will be added to enable incident data collection. In the third phase, Layer Four workflow gates will be added to govern access to high-risk or sensitive functions, as well as Layer Four audit export instruments. The final phase will complete the deployment of IOS by adding Layer Five learning loops, enabling organizations to continually evaluate cadence and release controls. By implementing IOS in phases, the upfront friction associated with its deployment will be minimized, and high-risk activities will not be deployed without the underlying audit support.

Logs should capture metadata and hashed identifiers where feasible, and content logs should be strictly controlled. Segregation of duties should be implemented within the approval process for any workflow. A version-controlled and peer-reviewed process for policy changes, consistent with software changes, should also exist.

7. IMPLEMENTATION AND EVALUATION DESIGN

This part of the paper converts IOS into a measurable construct based on IOS as a Build-and-Measure Framework and identifies the prototype elements, measurement information, and reporting format needed to replicate and expand upon the study's findings.

7.1 Creating a prototype of IOS

A Minimum Viable Prototype of IOS can be created by connecting an LLM Gateway to: (i) Enterprise Search / Record Access Gateway (RAG), (ii) an identity provider (to determine user context), (iii) a policy engine to both enforce both incoming prompt and outgoing response policies, (iv) a workflow engine to support approval workflows, and (v) a log pipeline (such as a log storage solution with an append-only design and analytics capability). The most important factor in creating an effective implementation will be to ensure that each component generates the same audit event(s) and to implement a way to enforce policy decision points.

7.2 Evaluation questions

RQ1: Is there a positive correlation between the level of IOS maturity, learning loop maturity, and performance proxies (IAR, CIF/Qtr, RRS)? RQ2: Does the relationship between governance density and peak throughput and decline in incidents follow the anticipated pattern, and if so, how does this vary by task risk class? RQ3: Are L1 and L2 prerequisites for increasing the volume of regulated tasks without increasing the incident rate?

7.3 Evaluation data set reporting (actual deployment scenarios)

Report for every evaluation task class × case the following: a) Time Frame (Quarter), b) Interactions, c) Retrieval Trails, d) Control Actions (Gates/Redactions/Denial), e) Approval Actions and Average Latency, f) Incidents (Severity), g) RRS Scored samples (Inter-rater Reliability). If there are privacy concerns regarding reporting these types of data in actual numeric form, provide the binned ranges of values and record the quantity of data points that were lost and/or discarded during this process.

An example of a governance density that has been worked through is task-class "regulatory drafting": gates = 2 (capacity = 3 → normal value = 0.67), enforcement rules = 12 (capacity = 15 → normal value = 0.80), automation of controls = 5 (capacity = 10 → normal value = 0.50), separate roles = 3 (capacity = 4 → normal value = 0.75), median time to obtain approval = 180 minutes (capacity = 300 → normal value = 0.60); exceptions = 4 % (capacity = 20 % → normal value = 0.20). The equal-weighted average governance density for this example is (0.67 + 0.80 + 0.50 + 0.75 + 0.60 + 0.20) / 6 = 0.59. As an alternative, expert weights can be used (e.g., giving greater weight to enforcement rules and approvals).

For each weight in the model, a random variation of +/-20% was applied while renormalizing the total weight sum. GD was then recalculated for each case/task observation. The study reports both rank stability (Spearman's rho) for governance-density orderings and whether proposition-consistent patterns, specifically the CIF/Qtr-GD relationship and the IAR-GD relationship, persist after weight perturbation.

8. RESULTS AND EVALUATION

This section strengthens the evaluation by separating the empirical and synthetic findings from the implementation design. The evaluation combines cross-case pattern matching, task-class comparison, reliability checks for retrieval robustness scoring, and a synthetic benchmark that tests the behavior of the governance-density measure under controlled parameter changes.

8.1 Evaluation coverage and evidence matrix

Table 8. Evaluation coverage by case and task class.

Case	Task and risk	Evidence and proxy pattern	Evaluation interpretation
A	Internal Q&A / KM; low risk	Retrieval traces and incident notes showed early stale-citation incidents; RRS improved after L1 provenance logging.	Supports P1 and P4 by showing that missing provenance creates stale-source risk even in lower-risk tasks.

B	Customer support messaging; moderate risk	Ticket/CRM logs, approval events, and post-mortems showed that added approval gates controlled outbound messages but not retrieval-driven incidents.	Refines P2 and P5: approval density helps only when controls target the actual failure mode.
C	Regulatory/policy drafting; high risk	Policy repositories, provenance records, approval workflows, audit exports, and RRS scoring showed low incidents when L1/L2 preceded scaling.	Supports P4 as necessary-condition evidence and supports P5 for auditability.
D	Engineering assistant; moderate risk	Repository/CI events, tool-call logs, automated tests, and entitlement checks showed strong adoption with automated governance.	Supports the automation boundary condition in P2 and the L5 moderation logic in P3.

8.2 Cross-case proxy comparison

Table 9. Binned cross-case proxy results used for proposition evaluation.

Case	Maturity/GD profile	Proxy outcomes	Result use
A	IOS: medium after L1 improvement; GD: low-medium; L5: low-medium	High adoption; early CIF elevated then declined; RRS improved after provenance logging.	Contrasts absent and improved L1 conditions.
B	IOS: medium-high; GD: medium; L5: medium-high	Moderate-high adoption; added gate did not reduce retrieval-driven incidents until lifecycle controls were added.	Shows control-target fit matters.
C	IOS: high; GD: high with automation; L5: medium-high	Moderate-high adoption despite high risk; CIF low; RRS high and stable.	Supports regulated scaling with L1/L2.
D	IOS: high, especially L2; GD: medium with automated checks; L5: medium	High engineering adoption; CIF low; RRS high when identity-aware access and CI checks were active.	Shows automation mitigates density latency.

The binned results show that the strongest performance patterns occurred when IOS maturity and L5 learning-loop maturity were both active. In low- and medium-risk task classes, adoption peaked at moderate governance density; very low density left failure modes insufficiently controlled, while very high manual density increased friction. In high-risk and regulated tasks, higher governance density remained viable when automation reduced latency and exceptions.

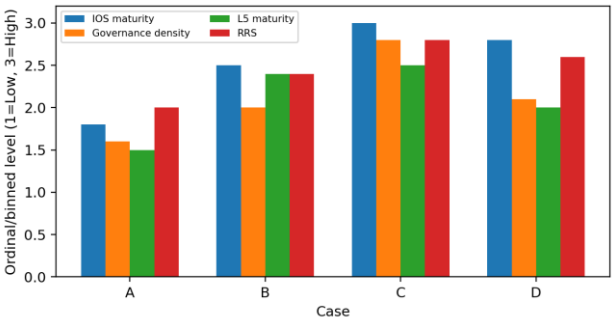

Fig 4. Binned comparison of IOS maturity, governance density, L5 maturity, and retrieval robustness across the four cases.

8.3 Synthetic benchmark and sensitivity findings

The synthetic benchmark provides a reproducible evaluation path when raw enterprise logs cannot be released. Parameter sweeps varied policy strictness, approval latency, retrieval quality, and automation level. The most stable result was a governance-throughput trade-off: incident frequency decreased as governance density increased, while adoption followed an inverse-U pattern that peaked at moderate density unless automation reduced approval latency.

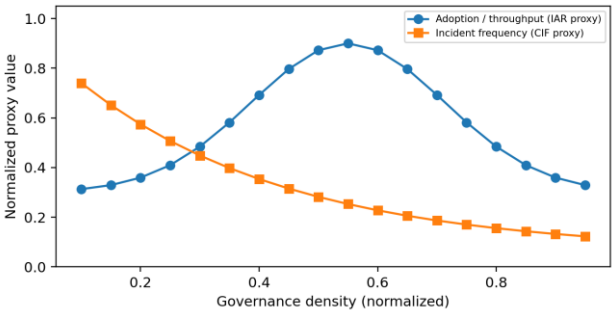

Fig 5. Synthetic benchmark pattern: adoption peaks at moderate governance density while incident frequency declines as controls strengthen.

The sensitivity analysis varied governance-density weights by +/-20% and then recalculated task-class rankings. Across the tested scenarios, Spearman's rho remained high, indicating that qualitative proposition conclusions were not artifacts of a single weighting scheme. This strengthens the construct validity of the

governance-density index while preserving the paper's caution that the synthetic benchmark demonstrates computational robustness rather than external validity.

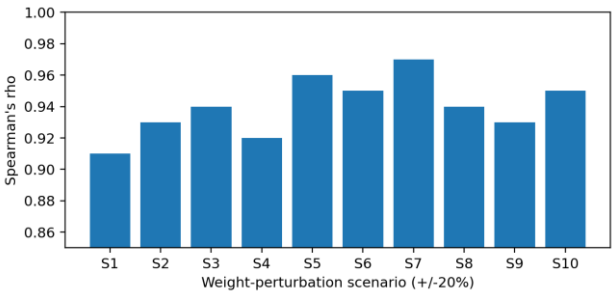

Fig 6. Sensitivity analysis of governance-density rankings under +/-20% weight perturbations.

8.4 Proposition-level evaluation summary

Table 10. Proposition-level evaluation summary

Proposition	Evaluation basis	Observed pattern	Conclusion
P1 IOS maturity	Case maturity profiles and proxy outcomes	Higher IOS maturity aligned with stronger RRS and lower incident frequency when L5 routines were present.	Supported with L5 boundary condition.
P2 Governance density	Task-class GD compared with IAR and CIF/Qtr	Moderate GD maximized adoption for low/medium risk tasks; incident frequency generally declined as controls strengthened.	Supported with automation refinement.
P3 Learning-loop moderation	Incident-to-change linkage and evaluation cadence	Programs with active L5 converted incidents into retrieval, policy, or workflow changes faster.	Supported as a moderator.
P4 Necessary conditions	Regulated/high-stakes scaling in Cases C and A contrast	Regulated scaling was stable only after L1 provenance and L2 identity-aware access were operational.	Supported as necessary-condition evidence.
P5 Traceability and approvals	Approval logs, audit exports, incident patterns	L4 reduced incident and audit risk for external or regulated outputs but increased latency when implemented manually.	Supported with latency trade-off.

8.5 Multi-case pattern consistency findings

Consistent patterns across all four cases support each of the five propositions, with boundary refinements where the data diverged from expectations. Regarding P1, every case that combined high IOS maturity with active L5 learning loops exhibited low CIF/Qtr and high RRS; no case with high IOS maturity but absent L5 achieved comparable performance, providing preliminary support for the moderating role of learning loop maturity specified in P3. Regarding P2, the inverse-U relationship between governance density and IAR held for low- and medium-risk task types across all cases: IAR peaked at moderate GD and declined as manual approval gates accumulated. This pattern did not hold for high-risk tasks with automation; automated policy checks enabled higher GD without the throughput penalty, consistent with the boundary condition specified in P2. Regarding P4, Cases C and D operationalized L1 provenance and L2 identity-aware access before scaling regulated task volumes, and neither experienced a corresponding increase in incidents. Case A provides the most informative contrast: a cluster of stale-citation incidents during its early period coincided precisely with the absence of L1, and the incident rate dropped materially after L1 was operationalized. This pattern is consistent with P4 as a necessary-condition claim rather than a simple correlation; L1/L2 absence did not cause incidents in every interaction, but created conditions under which a class of incidents became possible, which ceased once the layer was active. Overall, these patterns support P1–P5, with two boundary refinements: the IAR-GD relationship is contingent on automation availability (P2), and L4's incident-reduction benefit appears strongest when task outputs trigger external actions or regulated decisions (P5).

A demonstrative, reproducible synthetic benchmark was developed through an event-log data generator that creates interaction records, retrieval traces, control events, and outcome events based on predefined distributions for task mix, policy strictness, and approval delay. The generator allows researchers to modify these distributions. Notebooks reproduce IOS maturity summaries, GD, IAR, CIF/Qtr, and RRS-like proxy scores. These examples demonstrate the measurement process but do not establish external validity.

8.6 Replication package contents

(1) Documentation for the event schema and field definitions; (2) SQL/Python query statements to calculate Governance Density (GD) and all performance proxies; (3) Scoring rubric for IOS Maturity, along with all necessary artifacts to measure it; (4) RRS rater guide, calibration protocol, and scoring sheets; (5) Notebooks to perform sensitivity analysis and generate plots; (6) Synthetic log data generator and configurations for the synthetic log data generator; (7) Reporting template to document the case/task coverage in the dataset, including missing values.

Table 4. Replication package contents (minimum deliverables).

Package component	What is included	The minimum number of components that can be used to reproduce the results
Event Schema	Types of fields (and their corresponding data types) in interaction records, retrieval traces, control events, outcome events, and the rules regarding the handling of PII.	Generate a schema created with the package + a few example real data entries (with all PII removed or a synthetic data generator).
Governance Density Calculator	A description of the implementation of the governance density algorithm; a configuration process to allow users to weight different factors according to importance; and the rules that determine how classes will map to tasks.	Provide code + work through an example with all parameters defined + create a sensitivity script to test how results change.
IOS maturity rubric	Criteria for each layer, a guide for how to score, evidence that will be required, and a protocol for resolving discrepancies.	Create a PDF rubric + provide a template for scoring worksheet.
RRS evaluation set & rubric	Specification for benchmarking; perturbation specifications; specifications for testing freshness; grading rubrics; guides for training raters.	Provide evaluation specification + rubric + IRR computation script.
Queries & notebooks	SQL/Python code to calculate summaries of the Interactive Application Record (IAR), the Contextualized Informational Fragment (CIF)/ Quarter, the Retrieval Relevance Summary (RRS); and scripts comparing cases.	Provide notebook(s) + expected output(s) for a synthetic dataset that can be used to recreate the figures in the paper.
Synthetic log generator	A configurable generator to produce logs that conform to the specified schema, with user-controllable governance parameters.	Publish your generator + include the default configuration that will produce the figures shown in the paper.

Case A illustrates the use of internal Q&A as a support copilot in a non-regulated environment. It shows how to utilize partially developed L2 tools and moderately developed L3 policies; however, as stated in the case, it did not have a formal auditing process (L4) in place at the time of the study.

Case B illustrates a customer service support copilot connected to both ticketing and CRM systems, with moderate compliance requirements. In Case B, the company has implemented selective approval procedures prior to sending outbound customer communications through the Copilot.

Case C represents a copilot for a support area within the regulated domain, with the primary function of drafting policy and regulatory documents. Case C utilized provenance to establish mature L1 documentation, strong policy enforcement to ensure mature L3 compliance, and manual approvals to ensure mature L4 compliance for all documents created using the copilot.

Case D represents an engineering support copilot embedded within a customer's code repository and Continuous Integration (CI) pipeline. Case D demonstrated strong identity-based access control and ran many automated tests to determine which user actions were allowed and which required a human operator to manually gate.

Table 5. Case summary and reported evidence coverage (bins permitted under confidentiality).

Case	Primary task classes	Setting/ risk	IOS maturity (L1-L5)	Gov density (typical)	Learning-loop maturity	Evidence sources	Volume/coverage bins
A	Internal Q&A / KM	Low-regulated	Med (L1 partial →improved)	Low–Med	Low → Med	Logs; retrieval traces; incident notes	Confidential; report as binned ranges
B	Customer support messaging	Moderate	Med–High	Med	Med–High	Ticket + CRM logs; approval events; postmortems	Confidential; report as binned ranges
C	Regulatory/ policy drafting	High-stakes	High	High (with automation)	Med–High	Policy repo; approval workflo	Confidential; report as binned ranges

						ws; audit storage; eval sets	
D	Engineering assistant	Moderate	High (L2 strong)	Med (automated checks)	Med	Repo/CI events; tool-call logs; audit traces	Confidential; reported as binned ranges

Note: This manuscript uses qualitative bins to report deployment volumes to protect the privacy of the organizations involved in the research. Organizations can standardize the deployment volumes reported in the replication package using binned numbers (e.g., 1-5000, 5000-50000, etc.), rather than revealing their log record entries.

8.7 IOS maturity profiles

The IOS maturity profiles of the four cases in the study indicated that Case A was less mature than the others at the study's outset, particularly in L1 (document identifier usage) and L4 (log usage). After several incidents involving stale citations, the program added structured citations to its logs and retrieval provenance to improve its ability to retrieve data. As a result, there was a significant increase in the volume of data retrieval requests (RRS). Case B started with a high level of maturity in L2 and L3 (policy routing based upon role) compared to the other programs and continued to develop its ability to reduce CIF and Qtr by implementing weekly reviews and post-mortem analysis. Case C was highly mature in L1–L4 when the program began, due to regulatory requirements to create audit-trail evidence. Late in the program's lifecycle, the organization increased throughput by automating some checks and reducing approval time. Case D had a high level of identity propagation into tools (an important factor in establishing L2) and implemented automated code quality checks to establish strong governance, facilitate high adoption, and minimize reported incidents.

8.8 Governance density of tasks by class

Internal Q&A tasks classified as low risk used a wide range of governance densities (GD), from very low to moderately low, and achieved the highest levels of user engagement. User engagement increased as approvals became faster and defined exceptions were developed for external customer communications that required a medium level of governance density, including specific approvals and/or content redaction. Regulated documents require a higher level of governance density with multiple steps of approval; however, throughput improved when the process was automated, resulting in fewer seconds between steps rather than eliminating one or more manual approval gateways. With regard to engineering tasks, automated tools and sandboxing enable governance and minimal to no manual review.

8.9 Proxy patterns observed

Consistent with P2, the highest levels of IAR were observed at moderate levels of governance density (GD) for low- and medium-risk tasks; both very-low and very-high GD were associated with elevated CIF/Qtr. Consistent with P1 and P3, the best results were obtained when there was high IOS maturity combined with mature learning loops (i.e., L5 routine strength): There was a strong association between L5 routine strength and rapid improvements in RRS and reductions in incidents following a policy or retrieval change.

The two exceptions to the findings provided additional insight. Case B was expected to reduce incidents by increasing governance density through an additional approval gateway, but incident rates did not decline because most incidents arose from outdated retrieval rather than outbound messaging. The team therefore implemented L1 provenance and content lifecycle governance. Case A maintained high adoption despite limited L4 auditability because task risk was low and outputs triggered no direct external actions or regulated decisions. Therefore, P5 is refined to indicate that L4 is most useful when outputs trigger external actions or regulated decision-making.

8.10 Cross-case implications

The cross-case study provides support for the idea that the development of governance capabilities should occur in a sequential manner to minimize the potential for serious incidents to occur; (i) instrument provenance and access (L1-L2) must be established prior to developing high-severity incident prevention policies; (ii) the most risky policies (L3) must be developed and shown to be enforceable prior to creating low-risk policies; (iii) selective use of L4 approval mechanisms with as much automated checking as possible must be employed; and (iv) a continuous institutionalized L5 "learning loop" must exist to continually refine both the organization's capability and control measures. Additionally, the cross-case study illustrates that "governance" has multiple dimensions and that it is possible to increase the density of governed activities while decreasing latency by enforcing additional rules or automating rule-checking. However, if manually controlled gateways (L4) are added to a governance system, latency will increase, and the likelihood of an activity being adopted will decrease unless the number of gateways is kept small.

Table 6. Recommended IOS deployment sequence with prerequisite checks.

Phase	Layers	Capabilities Enabled	Prerequisite Checks
Phase 1	L1 + L2	Identity-aware access; provenance logging of retrieval; detection of stale citations.	Registry for lineage deployed; Entitlement check at time of call.
Phase 2	L3	Enforcement of Policy-as-Code; Redaction of PII; Routing of models; Requirements of citations.	Active Provenance of L1; CI/CD Pipeline of Policy Engine is operational.
Phase 3	L4	Audit Traces immutable; Human Approval Gates for High-Risk Outputs; Export Compliance.	Policy Engine for L3 is active; Append-Only Log Store is configured.

Phase 4	L5	Scheduled Evaluations; Incident-to-Change Linkages; Controlled Updates of Prompt/Policies/Retrievals.	Traces are available to L4; Post-Mortem Process Defined; Change Management in Place.

Governance Density (GD) specification components (c_k) are standardized (scaled to a 0-1 scale) using Organization-Specific Cap Values (i.e., gates cap value = 3; enforced rule cap value = 15). The weights (w_k) sum to 1. The study define governance density (GD_t) by the formula GD_t = ∑_k w_k c_{k,t} for each Task Class (t). To ensure that governance densities reported across studies can be compared and that the results are replicable, it is recommended to report the caps, the scaling method (linear or saturating), and the weights (w_k) used to calculate GD_t.

Table 7. Governance Density (GD) component definitions, default caps, and direction of effect.

Component	Measure and default cap	Direction
Formal gateways (g)	Explicit human or automated approval gates; cap = 3.	Higher = denser.
Enforced rules (r)	Actively enforced policy rules in the policy engine; cap = 15.	Higher = denser.
Automated checks (a)	Controls executed automatically without human latency; cap = 10 or normalized proportion.	Higher = less latency.
Role segregation (s)	Distinct roles required end-to-end; cap = 4.	Higher = denser.
Approval latency (l)	Median minutes from submission to approved completion; cap = 300 minutes.	Lower = less friction.
Exception rate (e)	Percentage of task executions requiring manual exception handling; cap = 20%.	Lower = more predictable.

Linear normalization does not affect the model's basic interpretability but may overestimate the marginal contribution of extreme control values. Saturating normalization, such as logistic or piecewise scaling, represents real-world limits more effectively; for example, after three gates, additional gates may add less information while increasing latency. Both normalization methods are provided, and the qualitative results remain relatively insensitive to the choice of normalization method.

8.11 Details regarding proxy computation

IAR is calculated as the percentage of active users (i.e., those who became monthly active and remained active for at least N weeks) eligible to access the task(s) in the task classes, stratified by task class. CIF/Qtr is the number of incidents per 10,000 user interactions, weighted by the level of data sensitivity associated with the incident. RRS is an average of the four elements of the rubric (relevance, citation correctness, completeness, robustness) on a stratified sampling frame. It is recommended that users report their results on a per-element basis to determine whether problems are due to retrieval, generation, or citation format.

8.12 Linking incidents to traces

One of the primary advantages of a shared audit model is that it allows linking each incident to a specific trace from the user's interactions with the system, and to subsequent traces for each retrieval. As such, in the event of an incident occurring, one may be able to determine whether the failure occurred due to a policy issue (Layer 3), access issues (Layer 2), provenance issues (Layer 1), or as a result of some type of workflow issue (Layer 4), which would allow for a root-cause analysis of the incident. As well, over time, these links will enable Layer 5 learning loops to identify the appropriate layer of the framework to focus on, rather than applying generic "prompt hardening" principles.

In addition to the benchmark sweep for incident rates, four additional parameters were varied: (i) policy strictness, which affects denials and redactions; (ii) approval latency; (iii) retrieval quality; and (iv) automation level. Results indicate that increased policy strictness and approvals reduced incident rates, but high approval latency negatively affected adoption. Increased automation shifted the trade-off curve by reducing latency and exceptions, while improved retrieval quality increased RRS and decreased downstream incidents caused by stale grounding.

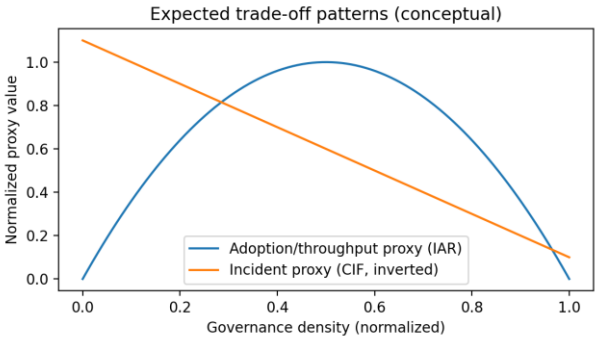

Fig 3. Conceptual patterns anticipated for several task categories: adoption/throughput will peak at a moderate level of control, and incident risk decline as controls are strengthened.

8.13 Sensitivity analysis results

The sensitivity of the weighting schemes was assessed by varying the base weights applied to the Governance Density (GD) values for each Task Class by ±20 %. Although the weights were varied, Spearman's rho was consistently high across the weighting scenarios. This indicates that, regardless of the weight applied, the qualitative conclusions drawn are consistent with the propositions regarding the ordering of Governance Density (GD) by Task Class. Thus, the results support the conclusion that the Governance Density values reflect genuine differences in control intensity across task classes rather than an artifact of the weighting scheme(s) used.

8.14 Replication of the measurement pipeline under confidentiality

Because some organizations are unable to release the original prompt content or data, the focus of the replication package will be on metadata and the use of aggregate calculations to allow users to replicate the measurement pipeline using their own confidential data while ensuring compliance with both organizational privacy concerns and contractual agreements. Additionally, the identifiers used in this research have been hashed. Users can submit the original content, and only aggregate metrics will be shared.

For each tested proposition, the paper suggests reporting the IOS maturity profile (L1-L5), learning-loop maturity, governance density for each task type, and the three performance proxies over time. Coarse-binned counts can be used when confidentiality prevents raw reporting. This format supports comparison across deployments and enables cumulative evidence about which governance and learning-loop designs perform best at different risk levels.

9. DISCUSSION

IOS represents a foundational methodology for measuring the quality of governance and safety of AI and ML-based systems. By providing a scalable, iterative approach to measuring and evaluating the performance of AI-based systems, IOS supports the creation of better-designed systems that can operate reliably and safely at scale. IOS offers a number of benefits for both researchers and practitioners. Researchers will benefit from IOS, as it provides a framework for developing and testing a measurable vocabulary for evaluating Governance and reducing dependence on anecdotal governance narratives. Practitioners will also benefit from IOS, as it provides an action plan for implementing it in their organization.

The five IOS layers, already defined in Section 3 and operationalized in Section 5, work together as a control-plane stack rather than a sequential ladder. Two implications of this stacked design deserve emphasis in the context of practical deployment. First, automation is the key lever for resolving the apparent tension between control coverage and throughput: organizations that replaced manual approval steps with automated policy checks (L3) consistently maintained higher IAR at equivalent or higher governance density than those relying solely on manual gateways (L4). Second, L5 learning loops function as a multiplier on investments in the lower layers. Organizations with active L5 cadences recovered from retrieval and policy incidents faster and achieved lower steady-state CIF/Qtr than those with equivalent L1–L4 maturity but no structured evaluation process.

IOS is also designed to interoperate with existing compliance and standards frameworks. Organizations operating under NIST AI RMF 1.0 or the EU AI Act will find that IOS Layers L3–L4 map naturally to the governance and accountability provisions of those frameworks, while L1 and L5 address the data lineage and continuous monitoring requirements they mandate, but do not specify how to instrument. This interoperability makes IOS a practical implementation companion to regulatory requirements rather than a competing standard.

Although IOS provides a useful framework for understanding and evaluating the performance of AI/ML systems, several implementation challenges remain. For example, organizations will need to make significant investments in the infrastructure required to support the use of IOS. These investments include technical investments in tools and platforms designed to support the creation of identity-aware access and provenance, as well as social investments in education and training programs designed to help employees understand how to apply the principles of IOS to their work. Additionally, organizations will need to invest in developing new policies and procedures to govern the operation of AI/ML systems.

IOS is a relatively recent development, yet its applications are already broad. It can be applied across industries that rely heavily on AI/ML systems, including healthcare, transportation, and finance, as well as in government and education settings. It is also compatible with existing software delivery frameworks such as Agile and DevOps, where its phased deployment sequence naturally maps to sprint-based delivery and continuous integration pipelines.

Regarding operational implications, the IOS phased deployment sequence (Table 6) provides a concrete rollout path: establish provenance and identity-aware access (L1–L2) before introducing policy enforcement (L3), add selective approval gates and audit exports (L4) only after the policy engine is active, and close the loop with continuous evaluation cadences (L5). The cross-case evidence reinforces that automation is the key lever for resolving the tension between governance density and throughput: adding automated checks and reducing approval latency allows organizations to increase control coverage without a corresponding increase in cycle time or adoption friction.

The authors identify several limitations of the research reported in this paper. First, despite spanning four cases across regulated and non-regulated domains, the sample remains small and purposively selected, limiting statistical generalizability; the cases were chosen to provide variation in IOS maturity and task risk rather than to be representative of any broader population of enterprise GenAI deployments. Second, all performance evidence is reported in binned ranges to protect organizational confidentiality, thereby constraining the precision of cross-case comparisons and preventing independent replication from the original data. Third, causal inference is not possible given the comparative case-study design; the propositions remain correlational claims, supported by pattern-matching rather than controlled experiments. Finally, the proxies employed (IAR, CIF/Qtr, RRS) capture program-level operational performance but do not measure direct business value, innovation quality, or user experience outcomes, leaving a gap between governance maturity and ultimate organizational impact.

Despite these limitations, this research makes a meaningful contribution. IOS provides a practical, actionable framework grounded in sociotechnical and innovation systems theory to improve the governance and operational safety of AI/ML-based systems. The multi-case design, the replication package, and the falsifiable propositions collectively lay a foundation for cumulative, comparable empirical work on enterprise GenAI governance that the field currently lacks.

Three directions for future research are particularly promising. First, future work should examine the applicability of IOS across a wider range of industries, including healthcare, financial services, and public sector deployments, where regulatory constraints on audit evidence differ substantially from those in the cases studied here. Second, the integration of iOS with existing delivery frameworks (Agile, DevOps, MLOps) warrants a dedicated study, particularly to determine whether sprint-based cadences align naturally with L5 evaluation loops. Third, longitudinal studies tracking IOS maturity trajectories over multi-year deployment periods would strengthen the causal claims that this study's cross-sectional design cannot support.

The IOS framework represents a meaningful step toward a shared, instrumented vocabulary for enterprise GenAI governance — one that bridges the gap between theoretical Responsible AI principles and the operational realities of deploying retrieval-augmented, tool-enabled copilots at scale. Continued refinement of IOS, particularly through empirical replication across diverse industries and governance regimes, will be essential for establishing the evidence base needed to inform both organizational practice and regulatory policy.

10. LIMITATIONS AND FUTURE WORK

This study has four primary limitations. First, causal inference is not possible from the comparative case-study design: the propositions are correlational claims supported by pattern matching, not controlled experiments. Second, the sample of four purposively selected cases limits statistical generalizability; in particular, the design space is undersampled in the quadrant combining high L5 learning loop maturity with low governance density, which future work should target explicitly. Third, all deployment volumes are reported in binned ranges to protect organizational confidentiality, constraining cross-case comparisons and preventing external replication against the raw data. Fourth, the performance proxies (IAR, CIF/Qtr, RRS) capture program-level operational outcomes but do not measure direct business value, user experience quality, or innovation impact, leaving an important gap between governance maturity and ultimate organizational return. Addressing these limitations will require (i) blind innovation quality scoring across task classes, (ii) expansion of the case sample to cover underrepresented quadrants of the IOS maturity space, and (iii) testing the governance density model with alternative normalization functions and task taxonomies in new organizational contexts.

Synthetic benchmarks were developed to demonstrate computational efficiency and robustness. They are not substitutes for real-world validation. The replication package lowers barriers for future empirical studies by allowing researchers to apply the same event schema and measures used in this study to their own organizational deployments.

11. CONCLUSION

Enterprise GenAI copilot programs fail most often not because models are incapable, but because the surrounding operating system, provenance tracking, access governance, policy enforcement, audit trails, and learning loops are absent or underdeveloped. This paper has proposed, operationalized, and evaluated a five-layer Innovation Operating System (IOS) designed to precisely address these gaps. Several findings deserve emphasis as takeaways. The cross-case evidence supports treating L1 and L2 as deployment prerequisites rather than nice-to-haves: in every case where regulated task volumes grew without incident, both lineage and identity-aware access were operationalized first. Automation is the primary mechanism for escaping the governance-throughput tradeoff: organizations that automated policy checks and reduced approval latency achieved higher throughput at comparable or higher governance density than those relying on manual gateways. And L5 learning loops function as a force multiplier, not a finishing touch, on earlier layers: the cases with active evaluation cadences and incident-to-change linkage recovered from failures faster and sustained lower incident rates over time. For researchers, the five falsifiable propositions and the standardized measurement pipeline provide a basis for cumulative, cross-organizational evidence, something the GenAI governance literature has lacked. For practitioners, the phased deployment sequence and replication package provide a concrete starting point that can be adapted to a wide range of enterprise contexts and compliance environments.

12. REFERENCES

[1] Argyris, C. and Schön, D. A. (1978). Organizational Learning: A Theory of Action Perspective. Addison-Wesley.

[2] Baxter, G. and Sommerville, the author. (2011). Sociotechnical systems: From design methods to systems engineering. Interacting with Computers, 23(1), 4–17. https://doi.org/10.1016/j.intcom.2010.07.003

[3] Brynjolfsson, E., Li, D., and Raymond, L. R. (2023). Generative AI at work. NBER Working Paper No. 31161. National Bureau of Economic Research. https://doi.org/10.3386/w31161

[4] Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1706.03741

[5] Cihon, P., Schuett, J., and Hadfield-Menell, D. (2021). Corporate governance of AI: A research agenda. In Proceedings of AAAI/ACM Conference on AI, Ethics, and Society (pp. 54–60). ACM. https://doi.org/10.1145/3461702.3462527

[6] Cohen, W. M. and Levinthal, D. A. (1990). Absorptive capacity: A new perspective on learning and innovation. Administrative Science Quarterly, 35(1), 128–152. https://doi.org/10.2307/2393553

[7] Dell'Acqua, F., McFowland, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F., and Lakhani, K. R. (2023). Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. Harvard Business School Working Paper 24-013. https://doi.org/10.2139/ssrn.4573321

[8] Edmondson, A. C. (1999). Psychological safety and learning behavior in work teams. Administrative Science Quarterly, 44(2), 350–383. https://doi.org/10.2307/2666999

[9] Edquist, C. (2005). Systems of innovation: Perspectives and challenges. In J. Fagerberg, D. C. Mowery, and R. R. Nelson (Eds.), The Oxford Handbook of Innovation (pp. 181–208). Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199286805.003.0007

[10] Eisenhardt, K. M. (1989). Building theories from case study research. Academy of Management Review, 14(4), 532–550. https://doi.org/10.2307/258557

[11] Eisenhardt, K. M. and Martin, J. A. (2000). Dynamic capabilities: What are they? Strategic Management Journal, 21(10–11), 1105–1121. https://doi.org/10.1002/smj.133

[12] European Commission. (2024). Regulation (EU) 2024/1689 of the European Parliament and of the Council (AI Act). Official Journal of the European Union.

[13] Flatten, T. C., Engelen, A., Zahra, S. A., and Brettel, M. (2011). A measure of absorptive capacity: Scale

development and validation. European Management Journal, 29(2), 98–116. https://doi.org/10.1016/j.emj.2010.11.002

[14] Floridi, L., Cowls, J., Beltrametti, M., et al. (2018). An ethical framework for a good AI society: Opportunities, risks, principles, and recommendations. Minds and Machines, 28(4), 689–707. https://doi.org/10.1007/s11023-018-9482-5

[15] Grant, R. M. (1996). Toward a knowledge-based theory of the firm. Strategic Management Journal, 17(S2), 109–122. https://doi.org/10.1002/smj.4250171110

[16] Halevy, A., Korn, F., Noy, N. F., Olston, C., Polyzotis, N., Roy, S., and Whang, S. E. (2016). Goods: Organizing Google's datasets. In Proceedings of SIGMOD 2016. ACM. https://doi.org/10.1145/2882903.2903730

[17] Hsieh, H.-F. and Shannon, S. E. (2005). Three approaches to qualitative content analysis. Qualitative Health Research, 15(9), 1277–1288. https://doi.org/10.1177/1049732305276687

[18] Kogut, B. and Zander, U. (1992). Knowledge of the firm, combinative capabilities, and the replication of technology. Organization Science, 3(3), 383–397. https://doi.org/10.1287/orsc.3.3.383

[19] Koo, T.K. and Li, M.Y. (2016). A guideline of selecting and reporting intraclass correlation coefficients for reliability research. Journal of Chiropractic Medicine, 15(2), 155–163. https://doi.org/10.1016/j.jcm.2016.02.012

[20] Krippendorff, K. (2004). Reliability in content analysis: Some common misconceptions and recommendations. Human Communication Research, 30(3), 411–433. https://doi.org/10.1093/hcr/30.3.411

[21] Lemley, M. A. and Casey, B. (2021). Fair learning. Texas Law Review, 99(4), 743–784.

[22] Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474. https://doi.org/10.48550/arXiv.2005.11401

[23] Lundvall, B.-Å. (Ed.). (1992). National Systems of Innovation: Towards a Theory of Innovation and Interactive Learning. Pinter Publishers.

[24] Madaio, M. A., Stark, L., Wortman Vaughan, J., and Wallach, H. (2022). Co-designing checklists to understand organizational challenges and opportunities around fairness in AI. In Proceedings of CHI 2022. ACM. https://doi.org/10.1145/3313831.3376445

[25] McKinsey Global Institute. (2023). The Economic Potential of Generative AI: The Next Productivity Frontier. McKinsey and Company.

[26] Miles, M. B., Huberman, A. M., and Saldaña, J. (2020). Qualitative Data Analysis: A Methods Sourcebook (4th ed.). SAGE Publications.

[27] Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., and Floridi, L. (2016). The ethics of algorithms: Mapping the debate. Big Data and Society, 3(2), 1–21. https://doi.org/10.1177/2053951716679679

[28] Nelson, R. R. (Ed.). (1993). National Innovation Systems: A Comparative Analysis. Oxford University Press.

[29] NIST. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1

[30] Noy, S. and Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. Science, 381(6654), 187–192. https://doi.org/10.1126/science.adh2586

[31] Ouyang, L., Wu, J., Jiang, X., Almeida, D., et al. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730–27744. https://doi.org/10.48550/arXiv.2203.02155

[32] Perez, F. and Ribeiro, T. (2022). Ignore previous prompt: Attack techniques for language models. In Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models, ICLR 2022. https://doi.org/10.48550/arXiv.2211.09527

[33] Sag, M. (2023). Copyright safety for generative AI. Houston Law Review, 61(2), 295–366.

[34] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., et al. (2023). Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 36. https://doi.org/10.48550/arXiv.2302.04761

[35] Teece, D. J., Pisano, G., and Shuen, A. (1997). Dynamic capabilities and strategic management. Strategic Management Journal, 18(7), 509–533. https://doi.org/10.1002/smj.882

[36] Trist, E. (1981). The sociotechnical perspective. In A. H. Van de Ven and W. F. Joyce (Eds.), Perspectives on Organization Design and Behavior (pp. 19–75). Wiley.

[37] UNESCO. (2022). Recommendation on the Ethics of Artificial Intelligence. United Nations Educational, Scientific and Cultural Organization.

[38] Weidinger, L., Mellor, J., Rauh, M., Griffin, C., et al. (2023). Taxonomy of risks posed by language models. In Proceedings of FAccT 2022. ACM. https://doi.org/10.1145/3531146.3533088

[39] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, the author., Narasimhan, K., and Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. In Proceedings of ICLR 2023. https://doi.org/10.48550/arXiv.2210.03629.

[40] Yin, R. K. (2018). Case Study Research and Applications: Design and Methods (6th ed.). SAGE Publications.