Risks and Challenges of using Agentic AI in Enterprise Software Systems

Rajeew Vishvakarma
Project Manager, Infosys
Sunrise, Florida
United States, 33323

Sooraj Jacob
Software Engineer, IntraEdge
Sunrise, Florida
United States, 33323

ABSTRACT
This study addresses the novel risks and governance challenges introduced by agentic AI systems in enterprise software environments. Unlike traditional AI or deterministic software, agentic AI autonomously executes complex workflows, interacts with tools, and accesses sensitive data, creating security, reliability, and accountability risks that are inadequately managed by existing software engineering and AI governance approaches. To mitigate these issues, this study proposes a Risk-Aware Human-in-the-Loop (RA-HIL) Governance Framework, which integrates layered controls across governance policy, agent planning, tool authorization, runtime monitoring, and audit-feedback management. This framework enables risk-based, fine-grained autonomy decisions that balance automation and the necessary human oversight.

A scenario-based evaluation using a defect triage workflow demonstrated significant improvements over baseline agentic AI operations in terms of security control, reliability, auditability, accountability, and operational efficiency. Key enhancements include restricted unauthorized tool use, evidence-based action recommendations, comprehensive audit trails, clear human approval for medium and high-risk actions, and preserved automation for low-risk tasks.

The RA-HIL framework reconceptualizes agentic AI as a controlled software workflow rather than a standalone model, emphasizing action-level governance tailored to impact, sensitivity, and reversibility. The limitations include the qualitative nature of the evaluation and the need for domain-specific adaptations. Future work involves middleware implementation, quantitative validation in production environments, and exploration of adversarial resilience, providing a practical governance model for the secure, reliable, and accountable deployment of agentic AI in enterprise systems.

General Terms
Artificial Intelligence, Software Engineering, Security, Governance, Reliability

Keywords
Agentic AI, AI Governance, Human-in-the-Loop, Software Quality, Enterprise AI, Secure Software Engineering, Auditability

1. INTRODUCTION
Enterprises are under tremendous pressure to improve their customer care services, software development costs, turnaround incident responses, compliance reviews, process decision-making, and workflow automation. Hence, enterprises are integrating AI to improve their services. Modern Agentic AI systems are much better than traditional AI assistants, which can perform basic tasks such as predictions and natural language generation[1], [2], [3]. Agentic AI systems are better designed and controlled than non-agentic AI systems. Advanced AI systems have great capabilities to perform complex tasks, use various tools to achieve desired results, access enterprise data without compromising privacy and security, interact with APIs, and complete workflows without manual intervention[4].

Enterprise decisions to use agentic AI systems over traditional AI assistants can significantly improve their current processes of ticketing, documentation, code repositories, approval workflows, and so on. However, this will impact the occurrence of software-related risks. New risks of using agentic AI include the possibility of executing processes incorrectly, resulting in errors in updating servicing ticket resolutions, accessing restricted documents, responding to customer queries, code generation, recording compliance information, and making operational decisions[5], [6], [7].

Existing methodologies and techniques in software engineering and cybersecurity are sometimes inadequate for agentic artificial intelligence systems. In existing practice, software quality assurance is concerned with deterministic functionality, defect prevention, regression testing, test coverage, and release management. The approach to the governance of conventional AI systems includes concerns regarding model performance, fairness, explainability, documentation, and monitoring. However, agentic AI involves the use of models, tools, memory management, workflow execution, external data retrieval, and human intervention. The risk of security issues may lie not only in the model outcome but also in the interactions between various components[8], [9].

AI risk management recommendations have highlighted principles such as governance, transparency, trustworthy behavior, and lifecycle-oriented control mechanisms. The NIST defines governing, analyzing, measuring, and managing AI risks, whereas the NIST Generative AI Profile applies this principle to generative AI systems. In application security, OWASP has identified some critical risks associated with LLM applications, such as prompt injection, improper handling of outputs, exposure of sensitive data, and granting excessive agency. However, these challenges can be exacerbated by the interconnections with other tools [10], [11].

The research gap covered by this study is the lack of an easy-to-implement governance structure that can help software development groups utilize agentic AI. Firms have been trying AI in the areas of service management, software engineering, compliance, operations, and dealing with customers. However, most of them do not have a way to manage the autonomy, accountability, access, monitoring, record-keeping, and need for human intervention in AI.

This paper presents the Risk-Aware Human-in-the-Loop

Governance Framework, a practical structure designed for agentic AI-enabled enterprise software systems. The framework delineates controls across five domains: governance policy, agent planning, tool authorization, runtime monitoring, and audit-feedback management. It facilitates organizations in determining which agentic actions should be observed, recommended, prepared for approval, executed, or blocked. The contributions of this study encompass a risk-centered approach to agentic software quality, a layered governance model, an action-level autonomy mechanism, and a scenario-based evaluation supported by comparative control analysis.

2. BACKGROUND AND RELATED WORK

2.1 Agentic AI in Enterprise Software Systems

Agent-based artificial intelligence involves AI systems that achieve objectives through reasoning, planning, instrument manipulation, perception, and action repetition. For example, agentic LLMs involve AI systems that can reason, act, and interact with users, instruments, and the environment. ReAct demonstrates that language models can integrate reasoning traces with relevant actions during interactions with other data sources.

The enterprise setting offers a good context for agentic AI, as most processes require information gathering, workflows, decision-making, and interaction between various applications. AI agents could be helpful in defect prioritization, help desk services, log analysis, code review, preparation of regulatory documents, customer support, searching through knowledge bases and business reporting. In this regard, the operation of agents is determined not only by the model itself but also by the prompts, retrieved documents, memory, available tools, roles of users, policies, and permissions. It is logical to view the concept of agentic AI as a sociotechnical software system[12], [13].

2.2 AI Governance and Risk Management

AI governance involves a set of processes, techniques, and practices employed to ensure that an AI application is trusted, secured, reliable, and accountable. The NIST AI Risk Management Framework offers a framework for assessing and managing the risks of AI through the use of governance, mapping, measurement, and management functions. The NIST Generative AI Profile is designed to perform the same functions for generative AI [14], [15].

For enterprise software, governance should be implemented through control mechanisms such as ownership, approvals, roles and responsibilities, documentation, monitoring, logging, incidents, and reviews. With agentic AI systems, there must be a definition of the scope of activities that an agent can see, decide, advise, and act on.

2.3 Security Risks in LLM and Agentic Applications

Security threats in LLM applications are currently a significant issue affecting enterprise adoption. The OWASP recognizes prompt injection, security flaws regarding output, sensitive data exposure, supply chain security issues, and overagency as some of the most critical threats posed by LLMs. Such threats are heightened when agents use tools capable of taking action [16].

An example of prompt manipulation is an agent's ability to ignore commands, share information, or misuse resources. Overprivileging allows the system to perform tasks outside its scope. Information leakage can occur via prompts, outputs, logs, memories, and/or tool data payloads. Another risk is insecure output usage due to the use of unverified generated code or messages.

2.4 Human-in-the-Loop Control

In human-in-the-loop control, which is a vital guiding principle for the design of risky AI systems, the human user remains engaged during the process of evaluation, approval, disapproval, or correction of actions aided by AI. In classic AI, supervision is typically geared towards the validation of output results. With agentic AI, supervision must go much further.

A useful governance framework must differentiate between low-, medium-, and high-risk actions. Actions categorized as low risk may be performed automatically while logging them. Actions that fall under the medium risk category may require review before execution. Actions that are categorized as high risk should always require permission, higher authorization, or no access[17], [18].

2.5 Software Quality and Auditability

Software Quality encompasses correctness, reliability, maintainability, security, usability, and performance. With agentic AI, traceability, control, explainability, and auditing must be considered. For example, a trustworthy agentic software would have to provide proof of how the task has been interpreted, tools used, information that was accessed, actions that were suggested, and if a person authorized those actions[19].

In the case of any actions performed by the software that impact customers, employees, accounting, manufacturing, or compliance, auditable logging is extremely essential. If proper logging is not performed, there could be no way for any entity to trace back why the software took any action or why a certain recommendation was made.

2.6 Research Gap

Previous research in the areas of AI governance, LLM security, secure software engineering, and human supervision serves as a good starting point; however, these topics are usually considered independently of each other. The agentic AI approach necessitates an integrated perspective because it involves the model inference process, use of tools, workflow orchestration, and supervision by humans. The void filled in This study addresses the development of a concise and risk-aware governance structure that ties agent autonomy to software attributes[5], [20], [21].

3. PROBLEM STATEMENT

Agentic AI constitutes an entirely novel category of enterprise software risk because it involves reasoning based on natural language and autonomous or partially autonomous actions. In conventional software applications, behavior tends to be dictated by deterministic rules, access control mechanisms, and workflows. For conventional AI systems, the focus of risk management is typically limited to issues such as predictive accuracy, fairness, transparency, or model drift. In the case of agentic AI, the risk stems from the convergence of all these elements[22], [23].

The key issue is the lack of effective governance mechanisms that would allow control over agent behavior within business applications. In their absence, an agent could access unintended information, choose the wrong tool, perform a risky action, leak confidential information, or create information that will be used directly by another system[24], [25].

The first difficulty is that the agentive actions are dynamic. An identical agent might exhibit different behaviors based on prompts, phrasing, environment, document recall, tool permissions, memory conditions, user roles, and intermediate perceptions. This makes testing and monitoring more difficult.

The second difficulty is the fuzzy line between recommendation and execution. While the AI system begins as an advisor, it is eventually integrated into a workflow with the capacity to perform actions such as creating tickets, updating information, sending messages, executing APIs, and initiating workflows [26].

3.1 Unauthorized Tool Use

The AI agent can call an inappropriate set of tools, plugins, scripts, or APIs for a specific task or user type. Without restricted access, the AI agent can perform operations outside the scope of its business function, thereby disrupting workflows or incorrectly updating systems.

3.2 Sensitive Data Exposure

Agent-based systems may require access to documentation, customer databases, tickets, logs, source code, or operational data. Any sensitive information can be disclosed using prompts, responses, agent memories, logs, retrieval mechanisms, or even by invoking external tools.

3.3 Prompt Manipulation and Instruction Conflict

Prompt injection and instruction conflict can cause the agent to disregard system policy, expose hidden instructions, misapply tools, or act according to harmful inputs. In business processes, manipulation can occur via user messages, documents, emails, tickets, web content, or code comments[27], [28].

3.4 Hallucinated or Unsupported Execution

The reasoning generated by LLM-driven agents could be erroneous but confident nonetheless. If erroneous reasoning is linked to any kind of tool execution, this could lead to faulty workflow execution, fake evidence of compliance, false classification of a defect, or erroneous operation recommendations [29], [30].

3.5 Weak Auditability

However, for most AI systems, logging occurs only for the final results. In the case of agentic AI, the organization requires the goal of the task, the plan generated, the relevant context extracted, tools called, intermediate results, action approvals, actions not taken, and finally, the final result obtained.

3.6 Unclear Human Accountability

The agentic AI system raises issues regarding accountability because decisions can be made based on the intentions of the user, the logic employed by the machine itself, and the way it is programmed, designed, and used. When the limits of approval are unclear, identifying those responsible for making wrong decisions becomes problematic.

Therefore, enterprise organizations require a governance framework to guide the authorization, monitoring, auditing, and control of agency-based artificial intelligence throughout their entire action cycle. This would help integrate software quality, software security, privacy protection, human supervision, and accountability into a framework [25], [31].

4. PROPOSED RISK-AWARE HUMAN-IN-THE-LOOP GOVERNANCE FRAMEWORK

The following section introduces the RA-HIL Governance Framework, which enables the secure and dependable adoption of agentic AI. This framework governs the actions throughout the entire life cycle, from task understanding to execution, monitoring, and auditing of results. The guiding principle of this framework is that the governance of an AI agent cannot be limited to the model itself. Rather, governance must cover all aspects, including policy, prompts, tools, data access, human intervention points, execution logging, and feedback[32], [33].

The five layers of the architecture include the Governance Policy, agent planning, tool access and authorization, Runtime Monitoring, and Audit and Feedback layers. The layers represent the scope and limits of the actions that an agent can take from observation to execution.

4.1 Framework Overview

The proposed framework adopts the lifecycle of controlled actions. First, the organization establishes policies, risk, accountability, and approval thresholds. The agent then analyzes the user's goal and makes an organized plan. Prior to execution, the framework will assess whether the use of certain tools and actions in conjunction with the data by a particular individual at that risk level is sanctioned. Runtime action monitoring occurs during planning and execution, while the logging component tracks decisions, approvals, calls to tools, results, and feedback.

4.2 Governance Policy Layer

Acceptable use policies, levels of autonomy, user roles, prohibited activities, approval processes, data handling procedures, and accountability are included in the Governance Policy Layer. Enterprise activities need to be categorized in terms of risk levels. Low-risk activities could involve summarizing publicly available documents or classifying non-confidential issues. Medium-risk activities include preparing customer responses or coding for review. Activities categorized as high risk could involve accessing confidential information, updating production data, communicating externally, approving transactions, configuring systems, and running scripts[34], [35].

4.3 Agent Planning Layer

The Planning layer of the agent manages goal interpretation, task decomposition, and action selection. Planning is a key risk area because an improper plan will result in the misuse of tools and actions. A good plan would contain the objective of the task, data needed, suggested tools, desired output, risk level, and whether the action requires authorization. Medium-to high-risk actions should have a plan that is reviewed by either a human user or the policy engine prior to action execution. Instruction hierarchy control would ensure that user commands and data do not supersede corporate policy and security restrictions[36].

4.4 Tool Access and Authorization Layer

The Tool Access and Authorization Layer controls interactions with third-party tools, APIs, databases, scripts, workflow engines, and enterprise applications. All tools need to be defined with metadata, including their purpose, permission required, data sensitivity involved, side effects, and risk category[37], [38]. The document search tool can be of lower risk, the ticketing update tool of medium risk, and the

production database modification tool of higher risk. This layer must ensure that role-based access control, least privilege model, allowlisting of tools, and input and output validations are enforced.

4.5 Runtime Monitoring Layer

The Runtime Monitoring Layer monitors agent activity as it plans, selects tools, executes them, and generates responses. The monitoring process must be able to identify any unusual tool use, multiple failed attempts, attempts to access sensitive data, violations of policies, prompt injections, unsafe outputs, and divergence from the planned course. Unsafe activities can result in stopping the process, requesting human intervention, or ending the session altogether. Low-confidence suggestions, risky tool usage, and sensitive data access can trigger escalation[39].

4.6 Audit and Feedback Layer

The Audit and Feedback layer contains the evidence necessary for assessing, explaining, and improving the agent's actions. The auditing trail must contain the request of the user, the goal inferred, the plan generated, the context retrieved, the tools chosen, the inputs to these tools, their outputs, policies followed, human decisions on these actions, actions rejected by the agent, response, and result. Audits provide evidence for accountability, incident investigations, regulatory compliance audits, machine learning model improvement, and system debugging.

4.7 Risk-Based Autonomy Model

In the context of this structure, the approach to granting autonomy is based on a risk-based autonomous structure that sets out different levels of autonomy that an agent can have for a particular task. Rather than providing the agent with total autonomy or none at all, this approach defines four distinct levels of autonomy. These include Level 0, no autonomy; Level 1, recommendation only; Level 2, execution after authorization from a human operator; and Level 3, automatic execution with monitoring.

4.8 Implementation Flow

This process begins with the user assigning a task to the system. The next step is planning, wherein the agent creates a plan. The policy engine categorizes tasks and actions according to risk. The authorization engine determines whether there are any permissions for the software tools. Decisions are made regarding the execution of actions, requests for authorization, or denial of permission. Runtime monitoring tracks the actions. The evidence and output were collected. Finally, human feedback helps to enhance the controls.

5. RISK-CONTROL MAPPING FOR ENTERPRISE AGENTIC AI

Implementing this model will be much easier if the risks associated with agency in AI are matched with appropriate governance, software quality and security controls. The following is the matching table based on risk management in the AI lifecycle, security in generative AI, and secure software development guidelines. This model can help businesses understand what could happen, which controls can mitigate the risk, whether human intervention is necessary, what must be retained as evidence, and how to manage any unsafe behavior[40], [41], [42].

Inappropriate use of tools and excessive agency can be avoided by the use of tool allow lists, role-based access, least privilege principles, approvals, and autonomous decisions for actions. The runtime system must prevent the unauthorized invocation of tools, detect the reoccurrence of mistakes, and report irrevocable actions and those that could affect production.

Control related to data is required on a different control path. Exposure of sensitive data and improper use of memory must be managed by means of data classification, data masking, filtering for retrieving data, data retention restrictions, and data context. Runtime checks must be performed to identify sensitive information in prompts, responses, and tool data payloads. In the case of regulated data, the agent needs to stay in recommendation only or approval-required mode [43].

Manipulation of prompts, execution without adequate evidence, and insecure handling of outputs can be addressed by implementing an instruction hierarchy, source validation, response-based responses, output schema checks, sandboxing, and confidence escalation. In the absence of evidence or in the case of contradictory instructions, the agent must pause execution for human review. The audit evidence should contain source information, conflict information, evidence, validation status, reviewer decision, and output.

Finally, problems associated with weak auditability, vague accountability, and interrupted workflows can be addressed through the required logging of actions, ownership policies, idempotent actions, rollbacks, and exceptions. It becomes possible to track the rationale behind the decision made, the person who made the authorization, which systems were impacted, and how the company remedied or improved the issue.

5.1 Risk-Based Control Selection

However, not all activities require the same level of control as walking. For instance, summarizing a public document might simply involve logging and some degree of validation or verification. However, creating a response for a customer may require human approval. Any update on a manufacturing document or execution of a script should necessitate strong authorization, approval, and supporting audit evidence. The controls must be chosen based on the impact, sensitivity, and reversibility of the action[44].

5.2 Integration with Software Quality Practices

Risk control mapping can be incorporated into current quality and security efforts within software. In the requirements stage, organizations must identify autonomy thresholds and approvals. In the design phase, the use of tools, data flow, and the process for handling failures must be documented. In the testing phase, the injection attack resistance, tool authorization, safe outputs, logging capabilities, and approvals must be confirmed. During the deployment phase, actions must be monitored, feedback gathered, and incidents investigated[45].

5.3 Analytical Assessment of the RA-HIL Framework

The value of the RA-HIL framework lies in its ability to break agentic AI risk into action-level control decisions. Unlike AI governance, which focuses on the model, its outputs, or performance metrics, agentic AI systems spread risk across operational stages, such as user requests, generated plans, chosen tools, data accessed, execution contexts, and final actions. Therefore, this framework evaluates agentic AI as a

software workflow, rather than as a single model-level entity.

Risk is managed using five key control mechanisms. First, policy-based classification sets autonomy limits before execution to prevent an excessive agency. Second, planning reviews reduce unsupported actions by requiring the agent to clearly state its goals, data needs, tools, and expected actions. Third, tool authorization enforces the least privilege and allowlisting to prevent misuse. Fourth, runtime monitoring detects unauthorized tool calls, unsafe outputs, access to sensitive data, and deviations from the approved plan during execution. Finally, audit-feedback management improves traceability by maintaining evidence for decision-making and refining controls[46], [47].

This framework balances automation and control. A fully manual review of agent actions hampers efficiency, whereas fully automating actions risks security, privacy, and reliability. RA-HIL addresses this by assigning autonomy levels: low-risk, reversible tasks can be automated with logging, medium-risk tasks require approval, and high-risk or irreversible actions are blocked or escalated. This distinction is critical; for example, an agent can summarize a ticket but should not independently change a production configuration.

Beyond security, the framework improves reliability through evidence-based planning, increases accountability through approval points, enhances auditability through action documentation, and supports maintainability through feedback-updating policies, prompts, and tool permissions. Thus, RA-HIL is a governance and software quality mechanism, not merely a security control.

6. SCENARIO-BASED EVALUATION AND RESULTS

This section presents an evaluation of the proposed RA-HIL framework through a structured, scenario-based assessment of enterprise software defect triage and incident support workflows. The evaluation aims to determine whether the framework enhances control over agentic AI actions compared to an unguided workflow. As this study introduces a governance framework rather than a deployed production system, the assessment focuses on control aspects and scenario-based analysis. The goal was not to demonstrate production-scale performance improvements but to evaluate whether the framework delivers clearer benefits in terms of security, reliability, auditability, accountability, and operational control[5], [6], [48].

The evaluation contrasts the two conditions. The first is a baseline agentic AI workflow, where an AI agent performs tasks such as reading defect tickets, retrieving logs, searching knowledge base articles, generating recommendations, and preparing workflow updates without explicit risk-based autonomy controls. The second is the RA-HIL-governed workflow, where agentic tasks are managed through policy classification, structured planning, tool authorization, runtime monitoring, approval gates, and audit logging.

Five evaluation dimensions guided the analysis: security control, reliability, auditability, accountability, and operational efficiency. Security control assesses whether unauthorized tools, data, and actions are effectively restricted. Reliability examines whether the recommendations are evidence-based and subject to review when the risk is elevated. Auditability considers whether a system maintains sufficient evidence to reconstruct the agent's behavior. Accountability evaluates the clarity of human approval and responsibility assignments. Operational efficiency measures whether low-risk tasks can be automated without imposing unnecessary manual review.

6.1 Evaluation Design

The evaluation examines the impact of the RA-HIL framework on control strength within an enterprise AI workflow, contrasting it with a baseline that lacks explicit governance. This assessment employs a software defect triage scenario that includes low-, medium-, and high-risk agentic actions. Five dimensions—security control, reliability, auditability, accountability, and operational efficiency— were evaluated using a five-point scale ranging from weak to very strong control. The RA-HIL framework incorporates governance policies, approval gates, and audit management, thereby enhancing control without impeding the automation of low-risk tasks. The scores, derived from qualitative assessments, illustrated anticipated improvements and provided a foundation for future empirical validation.

6.2 Scenario Workflow and Framework Application

The process consists of the following: the user enters the defect ticket; the agent identifies symptoms; searches for relevant logs, incidents, and knowledge base articles; generates an investigation plan; policy determines risk level of action; low-risk summary permitted; medium-risk customer communication needs approval; high-risk actions require elevation or are blocked; runtime monitors tool invocation; and lastly, recommendations, tool invocation, approval, and output are logged.

Governance policies are used through RA-HIL for the agent to review tickets provided, summarize logs, and propose severity classifications without being able to edit configurations or change production, close incident tickets, communicate with customers, or initiate rollback without approval. Planning involves creating an investigation plan that must be followed. In the tool phase, there was limited access to ticketing information, logging, and approved sources of evidence. Monitoring involves the detection of unauthorized logs, tools that cannot be used, inappropriate severity classification, and production changes that lack proof.

Five key improvements were noted in this scenario-based assessment. First, there is an improvement in security controls because the agent will not be able to use external tools beyond what the user's role, task assignment, or risk classification allows. Second, there is a reliability improvement because the agent must base its classification and remediation recommendation on past data from retrieved logs and other knowledge repositories. Third, there is an improvement in auditability because of traceability and record keeping. Fourth, there is an increase in accountability in the sense that medium- and high-risk decisions must be owned and sanctioned before they can be taken. Fifth, the efficiency of operations is maintained because low-risk decisions, such as ticket summarization and authorized searches in the knowledge base, may be automated despite being monitored.

Within the defect triage flow, logging is appropriate for low-risk activities, such as summarizing the ticket and querying the knowledge base repository. Medium-risk activities include retrieving production logs, assessing the incident severity level, writing a response to the client, and updating the service

management repository; these operations must have controls around access, verification, and possibly human intervention. High-risk actions include performing rollbacks or making any changes to the production environment [49], [50].

This further proves that the issue of autonomy should be delegated to actions and not agents. It is necessary for an agent to be capable of automating the processes of observation and summary formation while being prevented from carrying out actions that affect any form of production.

6.3 Results and Analysis

The RA-HIL workflow enhances control across five key evaluation dimensions, with the most notable improvement in auditability. Unlike the baseline system, which may only retain final outputs or partial interaction histories, RA-HIL's audit layer of the RA-HIL comprehensively logs requests, inferred goals, plans, tool usage, evidence, policy checks, approvals, rejections, and outcomes. This detailed record significantly strengthens the incident investigation and compliance processes.

Security control is also improved by restricting tool access based on user roles, task context, tool metadata, and risk. For instance, low-risk actions, such as knowledge base searches, are automatically permitted, whereas high-risk operations, such as production configuration changes, are blocked or escalated, thereby reducing unauthorized tool use. Reliability benefits from requiring agents to produce structured plans supported by relevant evidence; a lack of sufficient confidence or evidence triggers human review instead of direct execution.

Accountability is clarified through explicit approval points for medium- and high-risk actions, ensuring that a designated human decision-maker oversees critical steps, making responsibility more manageable. Operational efficiency has improved moderately, maintaining automation for low-risk tasks while mandating human intervention for higher-risk activities. This balanced approach fosters governance that enhances control without sacrificing the necessary automation.

6.4 Limitations of Evaluation

This was a qualitative evaluation using scenarios. The evaluation showed the practicality of the framework; however, there was no assessment of its performance in an actual setting. This can be allowed for a preliminary framework for evaluating governance; however, it is recommended that future efforts be made to assess this framework using a real-life prototype platform based on indicators such as policy violation rate and others.

7. DISCUSSION, LIMITATIONS, AND FUTURE WORK

This suggested framework aims to address a pragmatic issue encountered by companies implementing agentic AI systems within their enterprise software environments. Agentic AI systems may engage with tools, fetch data, call APIs, update workflows, write code, and make business decisions. In terms of governance, it is far more complicated than traditional AI because of the nature of risk creation through all the aspects mentioned above [5], [25].

The most significant advantage of using such an approach is the treatment of agentic AI as a software product, and not solely as a deployed model. All these characteristics still have to be considered and addressed, although their significance is limited if we consider the ability of AI to act via tools and affect the work of organizations.

This framework may be valuable to software development teams, enterprise architects, security professionals, risk professionals, and business owners. For software engineers, the framework may help during both the requirements and design stages to define capabilities, tooling boundaries, approval gates, testing scenarios, and logging requirements. For security professionals, the framework can be used to determine prompt injection vulnerabilities, sensitive data handling practices, tool misuse, and unapproved execution [51], [52], [53].

This study has some limitations. First, the assessment process was performed using a scenario approach, which is not reflective of the actual deployment of the system in production. Second, the framework itself is generic and will require adjustments depending on the type of system being evaluated (such as financial, healthcare, insurance, telecommunication, or governmental). Third, while this study covers aspects of governance, security, reliability, and auditability, there are no specific details on the implementation of policy engines or prompt injection testing.

The framework can be implemented as governance middleware between the AI agent and enterprise tools in future research. Further analysis can focus on quantitative measures, including the number of tool calls that were prevented from being made because they violated an organization's policy, the ratio of violations, latency in the process of policy validation, completeness of the audit trail, and the time taken to complete a task. The framework must also undergo further experimentation with adversarial attacks that attempt to evade policies.

8. CONCLUSION

Agentic AI alters the risk landscape of enterprise software systems by transitioning from merely generating outputs to actively executing actions. This study introduces the RA-HIL Governance Framework to manage this shift through layered controls, including policy enforcement, planning, tool authorization, runtime monitoring, and audit-feedback management.

Unlike traditional approaches that treat agentic AI as a deployed model, this framework views it as a controlled software workflow. The updated scenario-based evaluation demonstrates that RA-HIL enhances security control, reliability, auditability, accountability, and operational efficiency compared to unguided agentic workflows. The most significant advancement is auditability, as the framework mandates capturing evidence throughout the entire action lifecycle. Security is bolstered by limiting tool usage, reliability is strengthened through evidence-based planning, and accountability is maintained by establishing human-approval checkpoints for medium- and high-risk actions.

The core contribution of this study is an action-level governance model tailored for enterprise agentic AI. Rather than assigning a single autonomy level to the entire agent, the RA-HIL assesses each proposed action individually, considering factors such as impact, data sensitivity, reversibility, user role, and policy constraints. This nuanced approach enables the automation of low-risk tasks while restricting or escalating the handling of higher-risk actions. Future efforts should focus on implementing this framework as

a governance middleware that operates between enterprise AI agents and software tools. Further validation should measure operational metrics, including policy violation rates, unauthorized tool-call prevention, approval latency, audit completeness, and task completion time. Expanding evaluations across diverse sectors, such as financial services, healthcare, cloud operations, customer service, and software engineering workflows, will help determine the broader applicability of the model [54] [55].

9. FIGURES/CAPTIONS/TABLES

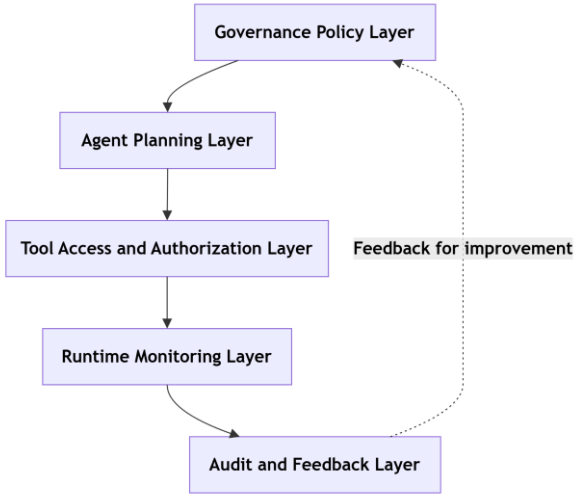

Figure 1. Risk-Aware Human-in-the-Loop Governance Framework for Enterprise Agentic AI Intelligence

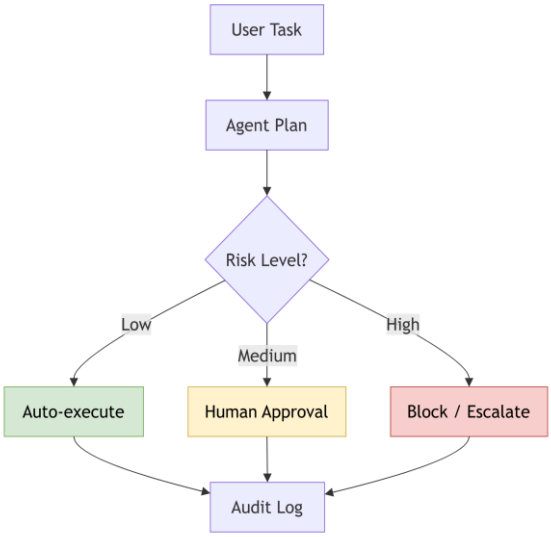

Figure 2. Risk-based autonomy decision flow for agentic AI action

Table 1. Evaluation criteria for RA-HIL scenario-based assessment

Evaluation Dimension	Measurement Basis	Baseline Condition	RA-HIL Condition
Security control	Ability to restrict unauthorized data, tools, and actions	Tool use depends mainly on prompt and available integration	Tool use is restricted by role, task, policy, and risk level
Reliability	Evidence support for classification and recommendation	Agent may recommend action without explicit evidence check	Agent must link recommendation to ticket, log, or knowledge source
Auditability	Ability to reconstruct agent behavior	Final answer or partial interaction may be logged	Request, plan, tool use, approval, evidence, and outcome are logged
Accountability	Clarity of human decision ownership	Responsibility may remain unclear after AI-supported action	Medium- and high-risk actions require defined approval ownership
Operational efficiency	Ability to automate low-risk work	Either over-automation or excessive manual review may occur	Low-risk tasks are automated; risky tasks are escalated

Table 2. Comparative evaluation of baseline and RA-HIL-governed defect-triage workflow

Evaluation Dimension	Baseline Score	RA-HIL Score	Observed Improvement
Security control	2	4	Unauthorized tool and data access are reduced through role-based authorization and tool allowlisting
Reliability	2	4	Evidence-grounded planning improves the quality of severity classification and remediation recommendation
Auditability	1	5	Complete action trail supports incident review and compliance evidence

Accountability	2	4	Human approval points clarify responsibility for medium- and high-risk actions
Operational efficiency	3	4	Low-risk tasks remain automated while high-risk actions are escalated

Scoring scale: 1 = weak control, 2 = limited control, 3 = moderate control, 4 = strong control, and 5 = very strong control.

The scores in Table 2 reflect an assessment using controlled scenarios rather than production telemetry. They compared the control efficacy of a baseline agentic workflow with the proposed RA-HIL-governed workflow in a defect-triage scenario. Future implementations may replace these qualitative scores with production metrics, including unauthorized tool-call prevention, approval latency, audit completeness, policy violation rates, and incident resolution time.

Figure 3 illustrates that the workflow governed by RA-HIL significantly enhances auditability. This improvement is attributed to the framework's explicit documentation of user requests, action plans, tool invocations, evidence, approval decisions, and final outcomes. Additionally, security control, reliability, and accountability are enhanced due to the agent's actions being restricted by risk classification, tool authorization, approval gates, and runtime monitoring. While operational efficiency experiences a moderate enhancement, this is because RA-HIL deliberately maintains human approval for medium and high-risk actions, permitting only low-risk activities to be automated.

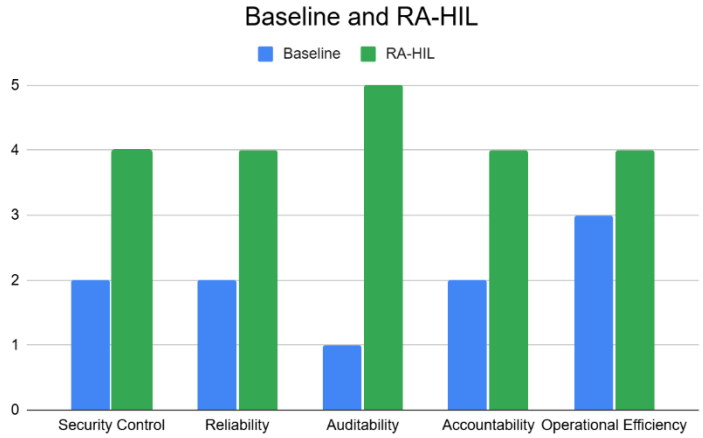

Figure 3. Comparative control-strength assessment of baseline and RA-HIL-governed defect triage workflows.

10. ACKNOWLEDGEMENTS

The authors thank the reviewers and editorial team of the International Journal of Computer Applications for their constructive feedback and suggestions. The authors also acknowledge the research and practitioner community in AI governance, secure software engineering, and responsible agentic AI for providing the broader foundation for this work.

11. REFERENCES

[1] D. Cahyono, H. E. Atmaja, and H. Zainarthur, “AI Agent Based Service Innovation to Enhance Efficiency and User Experience,” TMJ, vol. 10, no. 3, pp. 146–156, Jan. 2026, doi: 10.33050/tmj.v10i3.2585.

[2] I. Satpathy, A. Nayak, and V. Jain, “The Strategic Role of Artificial Intelligence (AI) in Service Delivery Systems,” Igi Global, 2024, pp. 291–310. doi: 10.4018/979-8-3693-7909-7.ch014.

[3] G. K. Sangam, “AI-Assisted CRM Agents for Sales, Service, and Support Operations,” IJAIDSML, vol. 7, pp. 245–249, Jan. 2026, doi: 10.63282/3050-9262.ijaidsml-v7i1p141.

[4] S. Ranjan, D. Chembachere, and L. Lobo, “Ethical and Security Considerations in Enterprise Agentic AI,” Apress, 2025, pp. 289–323. doi: 10.1007/979-8-8688-1542-3_7.

[5] S. K. Anuguthala, “Enterprise Agentic AI Lifecycle Governance: A Control-Driven Framework from Design to Decommissioning,” IJAIDSML, vol. 7, pp. 9–16, Jan. 2026, doi: 10.63282/3050-9262.ijaidsml-v7i2p103.

[6] B. Sateli, F. D. Castillo, and R. Moshtagi, “Towards determining the criticality of AI applications: A model risk management perspective,” Proceedings of the Canadian Conference on Artificial Intelligence, June 2023, doi: 10.21428/594757db.74665221.

[7] J. Sayles, “The Current State of AI Governance and Model Risk Management,” Apress, 2024, pp. 1–18. doi: 10.1007/979-8-8688-0983-5_1.

[8] A. Chhabra, S. Datta, S. Nahin, and P. Mohapatra, “Agentic AI Security: Threats, Defenses, Evaluation, and Open Challenges,” Oct. 27, 2025, European Society for Socially Embedded Technologies. doi: 10.48550/arxiv.2510.23883.

[9] A. Abuadbba, N. Sultan, S. Nepal, and S. Jha, “Human Society-Inspired Approaches to Agentic AI Security: The 4C Framework,” Feb. 02, 2026, European Society for Socially Embedded Technologies. doi: 10.48550/arxiv.2602.01942.

[10] A. K. Sood, Combating Threats and Attacks Targeting The AI Ecosystem. De Gruyter, 2024. doi:

10.1515/9781501520549.

[11] A. Srivastava and S. Panda, "A Formal Framework for Assessing and Mitigating Emergent Security Risks in Generative AI Models: Bridging Theory and Dynamic Risk Mitigation," Oct. 15, 2024, European Society for Socially Embedded Technologies. doi: 10.48550/arxiv.2410.13897.

[12] R. Rajubhai Golani, "The Rise of AI Agents: Transforming Enterprise Automation," IJAEM, vol. 7, no. 4, pp. 132–139, Apr. 2025, doi: 10.35629/5252-0704132139.

[13] S. Sarferaz, "Implementing Agentic AI Into ERP Software," IEEE Access, vol. 13, pp. 178945–178960, Jan. 2025, doi: 10.1109/access.2025.3621887.

[14] R. Tong, M. Cortês, J. Defalco, M. Underwood, and J. Zalewski, "A First-Principles Based Risk Assessment Framework and the IEEE P3396 Standard," Mar. 31, 2025, European Society for Socially Embedded Technologies. doi: 10.48550/arxiv.2504.00091.

[15] R. J. Tong, M. Cortês, J. A. Defalco, M. Underwood, and J. Zalewski, "A First-Principles Based Risk Assessment Framework and the IEEE P3396 Standard," Institute Of Electrical Electronics Engineers, May 2025, pp. 1588–1595. doi: 10.1109/cai64502.2025.00237.

[16] M. Fasha, F. A. Rub, N. Matar, B. Sowan, M. Al Khaldy, and H. Barham, "Mitigating the OWASP Top 10 For Large Language Models Applications using Intelligent Agents," Institute Of Electrical Electronics Engineers, Feb. 2024, pp. 1–9. doi: 10.1109/iccr61006.2024.10532874.

[17] S. Sadiq, M. Zur Muehlen, and M. Indulska, "Governance, risk and compliance: Applications in information systems," Inf Syst Front, vol. 14, no. 2, pp. 123–124, Aug. 2011, doi: 10.1007/s10796-011-9320-2.

[18] R. Cirabisi and K. V. Handal, "Governance of Risk and Compliance," Apress, 2010, pp. 73–86. doi: 10.1007/978-1-4302-1593-6_6.

[19] A. Schmitz, M. Akila, D. Hecker, M. Poretschkin, and S. Wrobel, "The why and how of trustworthy AI," at - Automatisierungstechnik, vol. 70, no. 9, pp. 793–804, Sept. 2022, doi: 10.1515/auto-2022-0012.

[20] R. Khan, D. Joyce, and M. Habiba, "AGENTSAFE: A Unified Framework for Ethical Assurance and Governance in Agentic AI," Dec. 02, 2025, American Chemical Society. doi: 10.48550/arxiv.2512.03180.

[21] K. Huang et al., "AAGATE: A NIST AI RMF-Aligned Governance Platform for Agentic AI," Oct. 29, 2025, European Society for Socially Embedded Technologies. doi: 10.48550/arxiv.2510.25863.

[22] L. Chiris and A. Mishra, "AURA: An Agent Autonomy Risk Assessment Framework," Oct. 17, 2025, American Chemical Society. doi: 10.48550/arxiv.2510.15739.

[23] A. Xu et al., "Agentic AI for Enterprise: Emerging Applications and Real-world Challenges," Association for Computing Machinery, Aug. 2025, pp. 6300–6301. doi: 10.1145/3711896.3737871.

[24] K. Tallam, "A Five-Plane Reference Architecture for Runtime Governance of Production AI Agents," June 10, 2026. doi: 10.48550/arxiv.2606.12320.

[25] N. Kshetri, "Governing Agentic AI: Security, Identity, and Oversight in the Age of Autonomous Intelligent Systems," Computer, vol. 58, no. 8, pp. 123–129, Aug. 2025, doi: 10.1109/mc.2025.3572173.

[26] R. Khanna et al., "Finding AI's Faults with AAR/AI: An Empirical Study," ACM Trans. Interact. Intell. Syst., vol. 12, no. 1, pp. 1–33, Mar. 2022, doi: 10.1145/3487065.

[27] J. Mchugh, K. Šekrst, and J. Cefalu, "Prompt Injection 2.0: Hybrid AI Threats," July 17, 2025, American Chemical Society. doi: 10.48550/arxiv.2507.13169.

[28] C. Xiang et al., "Architecting Secure AI Agents: Perspectives on System-Level Defenses Against Indirect Prompt Injection Attacks," Mar. 31, 2026. doi: 10.48550/arxiv.2603.30016.

[29] Y. Sung, H. Kim, and D. Zhang, "VeriLA: A Human-Centered Evaluation Framework for Interpretable Verification of LLM Agent Failures," Mar. 16, 2025, European Society for Socially Embedded Technologies. doi: 10.48550/arxiv.2503.12651.

[30] V. Bamil and R. K. K. Lingamgunta, "A Unified Evaluation and Governance Framework for Trustworthy LLM Agents," Jan. 09, 2026, Institute Of Electrical Electronics Engineers. doi: 10.36227/techrxiv.176799772.28164151/v1.

[31] J. Sayles, "Human Oversight of AI Systems," Apress, 2024, pp. 267–276. doi: 10.1007/979-8-8688-0983-5_12.

[32] J. Kraprayoon, Z. Williams, and R. Fayyaz, "AI Agent Governance: A Field Guide," May 27, 2025, American Chemical Society. doi: 10.48550/arxiv.2505.21808.

[33] M. Kumurdjieva, L. Doukovska, N. Ghouaiel, and A. Dhanani, "Governance of AI and Agentic Systems: Challenges and Methodologies," Institute Of Electrical Electronics Engineers, Nov. 2025, pp. 1–5. doi: 10.1109/bdkcse67969.2025.11300520.

[34] D. Dickstein and R. Flast, "Role of Corporate Governance." Wiley, pp. 245–271, Jan. 02, 2012. doi: 10.1002/9781119200987.ch13.

[35] J. Edwards and G. Weaver, "Governance, Risk Management, and Compliance." Wiley, pp. 1–18, Mar. 29, 2024. doi: 10.1002/9781394250226.ch1.

[36] L. Chrpa, J. Gemrot, and M. Pilat, "Towards a Safer Planning and Execution Concept," Institute Of Electrical Electronics Engineers, Nov. 2017, pp. 972–976. doi: 10.1109/ictai.2017.00149.

[37] J. H. Chae and N. Shiri, "Description Logic Framework for Access Control and Security in Object-Oriented Systems," Springer Science Business Media, 2007, pp. 565–573. doi: 10.1007/978-3-540-72530-5_68.

[38] S. Chong, "Authorization Contracts," Association for Computing Machinery, Oct. 2017, p. 75. doi: 10.1145/3139337.3139348.

[39] H. Errico, "Autonomous Action Runtime Management(AARM):A System Specification for Securing AI-Driven Actions at Runtime," Feb. 10, 2026, European Society for Socially Embedded Technologies. doi: 10.48550/arxiv.2602.09433.

[40] A. Wasil, J. Clymer, D. Krueger, E. Dardaman, S. Campos, and E. Murphy, "Affirmative safety: An approach to risk management for high-risk AI," Apr. 14,

2024, European Society for Socially Embedded Technologies. doi: 10.48550/arxiv.2406.15371.

[41] D. Bogdanov, P. Etti, L. Kamm, and F. Stomakhin, "Artificial Intelligence System Risk Management Methodology Based on Generalized Blueprints," Institute Of Electrical Electronics Engineers, May 2024, pp. 123–140. doi: 10.23919/cycon62501.2024.10685644.

[42] P. K. Dutta, S. Das, and D. Ganguly, "Safeguarding Business in the Age of AI for Organizational Resilience and Risk Management," Igi Global, 2024, pp. 78–101. doi: 10.4018/979-8-3693-1198-1.ch005.

[43] L. Sion, D. V. Landuyt, and W. Joosen, "An Overview of Runtime Data Protection Enforcement Approaches," Institute Of Electrical Electronics Engineers, Sept. 2021, pp. 351–358. doi: 10.1109/eurospw54576.2021.00044.

[44] C. Bostock, "The Practising Manager's Control Requirements," Measurement and Control, vol. 3, no. 2, pp. T30–T32, Feb. 1970, doi: 10.1177/002029407000300205.

[45] S. Türpe, A. Poller, J. Trukenmüller, J. Repp, and C. Bornmann, "Supporting Security Testers in Discovering Injection Flaws," Institute Of Electrical Electronics Engineers, Aug. 2008, pp. 64–68. doi: 10.1109/taic-part.2008.7.

[46] I. B. Utne, T. A. Johansen, and A. J. Sørensen, "Towards Risk-Based Rationality in Autonomous Systems and Operations," Institute Of Electrical Electronics Engineers, Jan. 2025, pp. 1–6. doi: 10.1109/rams48127.2025.10935192.

[47] M. Luckcuck and M. Farrell, "Regulating Safety and Security in Autonomous Robotic Systems.," July 2020, doi: 10.5281/zenodo.3937545.

[48] T. Opalana, "Integrating AI safety, security, and compliance controls to reduce systemic risk in enterprise AI deployments," Int. J. Comput. Artif. Intell., vol. 4, no. 2, pp. 71–82, July 2023, doi: 10.33545/27076571.2023.v4.i2a.263.

[49] A. Pecchia, G. Carrozza, M. Cinque, and D. Cotroneo, "Industry practices and event logging: assessment of a critical software development process," May 2015, pp. 169–178. doi: 10.5555/2819009.2819035.

[50] A. Pecchia, M. Cinque, G. Carrozza, and D. Cotroneo, "Industry Practices and Event Logging: Assessment of a Critical Software Development Process," Institute Of Electrical Electronics Engineers, May 2015, pp. 169–178. doi: 10.1109/icse.2015.145.

[51] E. Bihari, "Frameworks: How to Build One?," EDPACS, vol. 57, no. 4, pp. 7–14, Apr. 2018, doi: 10.1080/07366981.2018.1444009.

[52] D. Dodson, M. Souppaya, and K. Scarfone, "Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework (SSDF)," National Institute Of Standards Technology, Apr. 2020. doi: 10.6028/nist.cswp.04232020.

[53] N. Z. Jhanjhi, I. A. Shah, and S. N. Brohi, "Evaluating Cybersecurity Frameworks Safeguarding the Software Industry Against Evolving Threats," Igi Global, 2025, pp. 111–128. doi: 10.4018/979-8-3693-6250-1.ch006.

[54] K. Raja, "Responsible LLM Systems: Governance, Safety, and Evaluation Frameworks for Enterprise AI Agents," cfs, pp. 659–669, Feb. 2026, doi: 10.52710/cfs.951.

[55] P. K. Natta, "Scalable Governance Frameworks for AI-Driven Enterprise Automation and Decision-Making," JRPETM, vol. 8, no. 6, Nov. 2025, doi: 10.15662/ijrpetm.2025.0806022.