iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** V **Month of publication:** May 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.82076

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue V May 2026- Available at www.ijraset.com*

# Applying Agentic AI for Intelligent Software Testing and Vulnerability Management

Subodh Jain
*R.K.D.F. Institute of Science & Technology, Bhopal, India*

***Abstract:*** ***The increasing complexity of enterprise software systems has significantly expanded the challenges associated with software testing, security validation, and vulnerability management. Traditional automation approaches are primarily rule-based and require substantial manual intervention, resulting in slower response times, increased operational overhead, and delayed vulnerability remediation. Recent advancements in Artificial Intelligence (AI), particularly Agentic AI systems, offer new possibilities for autonomous decision-making and intelligent software lifecycle management.***

***This paper explores the application of Agentic AI for intelligent software testing and vulnerability management in enterprise environments. The proposed framework utilizes multiple intelligent agents capable of autonomously performing test case generation, vulnerability detection, code analysis, remediation recommendation, regression validation, and reporting activities. Unlike conventional automation systems, Agentic AI systems possess contextual awareness, adaptive learning capabilities, and collaborative decision-making mechanisms that improve operational efficiency and testing effectiveness.***

***The paper presents a conceptual multi-agent architecture for integrating Agentic AI into DevSecOps workflows and software quality assurance processes. The proposed approach aims to reduce manual testing effort, accelerate vulnerability identification, improve remediation cycles, and enhance overall software reliability. Additionally, the study discusses implementation challenges, governance considerations, security risks, and future opportunities associated with autonomous AI-driven software engineering systems.***

***Keywords: Agentic AI, Software Testing, Vulnerability Management, DevSecOps, Artificial Intelligence, Cybersecurity Automation, QA Automation, Intelligent Agents, Autonomous Systems.***

## I. INTRODUCTION

Modern enterprise software systems have become increasingly distributed, scalable, and complex due to the adoption of cloud computing, virtualization, microservices architectures, and continuous deployment models. As organizations accelerate digital transformation initiatives, software development and operational teams face growing challenges related to software quality assurance, security vulnerability management, and release cycle optimization.

Traditional software testing and vulnerability management processes rely heavily on predefined rules, manual analysis, static automation scripts, and reactive security monitoring mechanisms. These approaches often struggle to adapt to rapidly changing environments, resulting in delayed vulnerability detection, incomplete test coverage, increased operational overhead, and higher security risks. The emergence of Artificial Intelligence (AI) has introduced new possibilities for intelligent automation in software engineering practices. Recent advancements in Large Language Models (LLMs), autonomous agents, and multi-agent systems have enabled the development of Agentic AI frameworks capable of contextual reasoning, adaptive decision-making, and collaborative task execution.

Agentic AI refers to AI systems that can autonomously perform tasks, make decisions, coordinate with other agents, and dynamically adapt to changing operational conditions with minimal human intervention. Unlike conventional automation systems that follow static workflows, Agentic AI systems possess greater flexibility, reasoning capabilities, and autonomous orchestration features.

In software engineering and cybersecurity domains, Agentic AI systems can significantly improve testing automation, security analysis, vulnerability management, incident response, and operational efficiency. Intelligent agents can autonomously analyze source code, generate test cases, identify vulnerabilities, recommend remediation strategies, execute validation tests, and generate operational reports.

This paper proposes a conceptual framework for applying Agentic AI in intelligent software testing and vulnerability management. The proposed approach integrates multiple intelligent agents into DevSecOps and software quality assurance workflows to improve automation effectiveness, reduce manual effort, and enhance software reliability.

## II. LITERATURE REVIEW

Artificial Intelligence has become an important research area within software engineering, cybersecurity, and automation domains. Several studies have explored the application of AI-driven techniques for software testing, vulnerability detection, intelligent monitoring, and autonomous operational management.

Traditional automated software testing approaches primarily depend on static scripts, predefined workflows, and manual maintenance. While automation frameworks improve execution efficiency, they lack contextual understanding and adaptive behavior. Recent advancements in Machine Learning (ML) and Natural Language Processing (NLP) have enabled the development of intelligent testing systems capable of dynamic decision-making.

Large Language Models (LLMs) such as GPT-based architectures have demonstrated strong capabilities in code generation, code analysis, debugging assistance, and test case creation. AI-assisted development tools have significantly enhanced developer productivity and software engineering efficiency.

Several frameworks such as LangChain, AutoGPT, CrewAI, and autonomous orchestration systems have introduced the concept of AI agents capable of independently performing complex workflows. Multi-agent systems allow different AI agents to collaborate, exchange contextual information, and perform specialized operational tasks.

In cybersecurity research, AI-driven vulnerability detection systems have shown promising results in identifying security weaknesses, malware patterns, anomalous behaviors, and configuration risks. Intelligent automation has also been integrated into Security Operations Centers (SOCs) to improve incident detection and response times.

DevSecOps methodologies emphasize integrating security practices into continuous integration and continuous deployment (CI/CD) pipelines. However, existing DevSecOps implementations still require substantial human intervention for vulnerability analysis, regression validation, and remediation coordination.

Although existing research highlights the benefits of AI-assisted software engineering, limited research has focused specifically on applying Agentic AI systems for integrated software testing and vulnerability management workflows. This paper attempts to bridge this gap by proposing a conceptual multi-agent framework capable of autonomous coordination and intelligent operational decision-making.

## III. PROBLEM STATEMENT

Enterprise software systems face several operational and security-related challenges due to increasing software complexity, rapid release cycles, and evolving cybersecurity threats.

Traditional testing and vulnerability management processes suffer from multiple limitations:

*1)* Heavy dependency on manual intervention
*2)* Limited contextual awareness in automation systems
*3)* Slow vulnerability remediation cycles
*4)* Incomplete regression testing coverage
*5)* High operational overhead
*6)* Difficulty handling dynamic enterprise environments
*7)* Reactive rather than proactive security management
*8)* Inconsistent test maintenance and scalability issues

Conventional automation frameworks operate using static rules and predefined workflows, making them less effective in adaptive decision-making scenarios. Security teams and QA teams often spend significant effort on repetitive tasks such as vulnerability analysis, test execution, remediation validation, and report generation.

Additionally, increasing software release frequency in CI/CD pipelines creates pressure on organizations to accelerate testing and vulnerability management processes without compromising quality and security.

Therefore, there is a need for intelligent autonomous systems capable of performing software testing and vulnerability management activities with improved contextual understanding, adaptive reasoning, and autonomous orchestration capabilities.

## IV. PROPOSED AGENTIC AI FRAMEWORK

This paper proposes a conceptual multi-agent Agentic AI framework for intelligent software testing and vulnerability management. The framework consists of multiple specialized AI agents collaborating autonomously to perform software quality assurance and security management activities.

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue V May 2026- Available at www.ijraset.com*

### *A. Framework Architecture*

The proposed architecture includes the following intelligent agents:

1. Test Case Generation Agent

This agent analyzes application requirements, source code, and previous defect patterns to automatically generate intelligent test cases.

Functions:

- Functional test generation
- Regression test creation
- Edge case identification
- Dynamic test prioritization

2. Vulnerability Detection Agent

This agent continuously scans source code, application logs, and deployment artifacts to identify potential security vulnerabilities.

Functions:

- Vulnerability scanning
- Pattern recognition
- Security risk assessment
- Threat classification

3. Code Analysis Agent

This agent performs intelligent code analysis to detect coding anomalies, insecure coding practices, and architectural weaknesses.

Functions:

- Static code analysis
- Dependency analysis
- Configuration validation
- Code quality assessment

4. Remediation Recommendation Agent

This agent recommends possible remediation strategies and security fixes based on identified vulnerabilities.

Functions:

- Patch recommendation
- Secure coding suggestions
- Dependency upgrade recommendations
- Configuration remediation

5. Validation Agent

This agent validates the effectiveness of remediation activities by executing regression tests and security validation workflows.

Functions:

- Regression execution
- Security validation
- Compatibility testing
- Risk verification

6. Reporting and Monitoring Agent

This agent generates operational reports, dashboards, and audit summaries for stakeholders.

Functions:

- Vulnerability reporting
- Test execution summaries
- Compliance dashboards
- Performance analytics

## V. WORKFLOW OF PROPOSED SYSTEM

The proposed Agentic AI workflow operates as follows:

*1)* Source code or deployment changes are detected.
*2)* The Code Analysis Agent reviews source code and dependencies.
*3)* The Vulnerability Detection Agent identifies potential security risks.
*4)* The Test Case Generation Agent creates intelligent test scenarios.
*5)* The Validation Agent executes regression and security tests.
*6)* The Remediation Agent suggests corrective actions.
*7)* The Reporting Agent generates operational insights and dashboards.
*8)* Continuous feedback is shared among agents for adaptive learning.

This collaborative architecture enables autonomous decision-making and continuous improvement across software testing and vulnerability management processes.

## VI. METHODOLOGY

This research follows a conceptual framework and comparative analysis approach.

The proposed Agentic AI architecture is evaluated against traditional software testing and vulnerability management models based on the following parameters:

*1)* Automation capability
*2)* Contextual understanding
*3)* Scalability
*4)* Vulnerability response time
*5)* Regression testing efficiency
*6)* Operational overhead
*7)* Adaptability

The framework assumes a simulated enterprise DevSecOps environment involving:

*1)* Continuous Integration/Continuous Deployment (CI/CD)
*2)* Automated build pipelines
*3)* Vulnerability scanning systems
*4)* Regression testing frameworks
*5)* Security validation workflows

Comparative analysis is performed to evaluate how Agentic AI systems can improve software quality assurance and security operations.

## VII. BENEFITS OF AGENTIC AI IN SOFTWARE TESTING AND VULNERABILITY MANAGEMENT

The proposed framework provides several advantages over traditional automation systems.

*1)* Reduced Manual Effort: Agentic AI systems automate repetitive operational tasks such as test generation, vulnerability analysis, and remediation validation, reducing dependency on manual intervention.
*2)* Faster Vulnerability Detection: Intelligent agents continuously monitor systems and analyze security risks in real time, enabling faster vulnerability identification.
*3)* Improved Test Coverage: AI-generated intelligent test cases improve functional and regression test coverage by dynamically identifying edge cases and risk areas.
*4)* Enhanced DevSecOps Integration: The proposed framework integrates security validation into CI/CD pipelines, improving continuous security assurance.
*5)* Autonomous Decision-Making: Agentic AI systems possess contextual awareness and adaptive reasoning capabilities, allowing autonomous orchestration of testing and remediation workflows.
*6)* Improved Operational Efficiency: The framework reduces operational delays, accelerates release cycles, and improves software reliability.

## VIII. CHALLENGES AND LIMITATIONS

Despite significant benefits, Agentic AI systems also introduce several technical and operational challenges.

1) Hallucination Risks: AI systems may generate incorrect recommendations or false-positive vulnerability assessments.
2) Security Concerns: AI agents handling sensitive source code and operational data may introduce privacy and security risks.
3) Governance and Compliance: Organizations require governance frameworks to ensure accountability, auditability, and compliance with regulatory standards.
4) Human Oversight Requirements: Critical security decisions still require human validation and expert supervision.
5) Infrastructure Complexity: Implementing multi-agent AI systems may increase infrastructure complexity and operational management requirements.

## IX. FUTURE SCOPE

Agentic AI systems are expected to play a major role in future software engineering and cybersecurity operations.

Future research directions include:

1) Self-healing enterprise systems
2) Autonomous security operations centers (SOCs)
3) AI-driven incident response
4) Intelligent root cause analysis
5) Predictive vulnerability management
6) Autonomous infrastructure optimization
7) AI-assisted compliance management

Advancements in Large Language Models, reinforcement learning, and autonomous orchestration systems are expected to further enhance the capabilities of Agentic AI frameworks.

## X. CONCLUSION

The increasing complexity of enterprise software systems requires more intelligent, adaptive, and autonomous approaches for software testing and vulnerability management.

This paper presented a conceptual multi-agent Agentic AI framework for intelligent software testing and vulnerability management. The proposed architecture integrates autonomous AI agents into DevSecOps workflows to improve testing automation, vulnerability detection, remediation validation, and operational reporting.

Unlike traditional rule-based automation systems, Agentic AI systems provide contextual reasoning, adaptive learning, and collaborative decision-making capabilities. The proposed framework has the potential to reduce manual effort, accelerate vulnerability response cycles, improve software quality, and enhance operational efficiency.

Although challenges related to governance, security, hallucinations, and human oversight remain important considerations, Agentic AI represents a promising direction for the future of intelligent software engineering and cybersecurity automation.

## REFERENCES

[1] Russell, S., & Norvig, P. Artificial Intelligence: A Modern Approach.
[2] Kim, G., Humble, J., Debois, P., & Willis, J. The DevOps Handbook.
[3] OWASP Foundation. OWASP Top 10 Security Risks.
[4] OpenAI Research Publications on Large Language Models.
[5] LangChain Documentation and Multi-Agent Framework Concepts.
[6] Research articles on AI-driven software testing and cybersecurity automation.
[7] Studies on autonomous agents and intelligent DevSecOps systems.
[8] Research on Machine Learning applications in vulnerability detection.
[9] NIST Cybersecurity Framework Documentation.
[10] Recent publications on Agentic AI and autonomous orchestration systems.

**SUGGESTED DIAGRAMS TO ADD**

Diagram 1: Proposed Agentic AI Architecture
Include:

1) Source Code Repository
2) CI/CD Pipeline
3) Test Case Generation Agent
4) Vulnerability Detection Agent
5) Validation Agent
6) Reporting Agent
7) Feedback Loop

Diagram 2: Traditional vs Agentic AI Workflow
Comparison between:

1) Manual testing workflow
2) Autonomous AI-driven workflow

Diagram 3: Vulnerability Management Lifecycle
Flow: Detection → Analysis → Remediation → Validation → Reporting

---

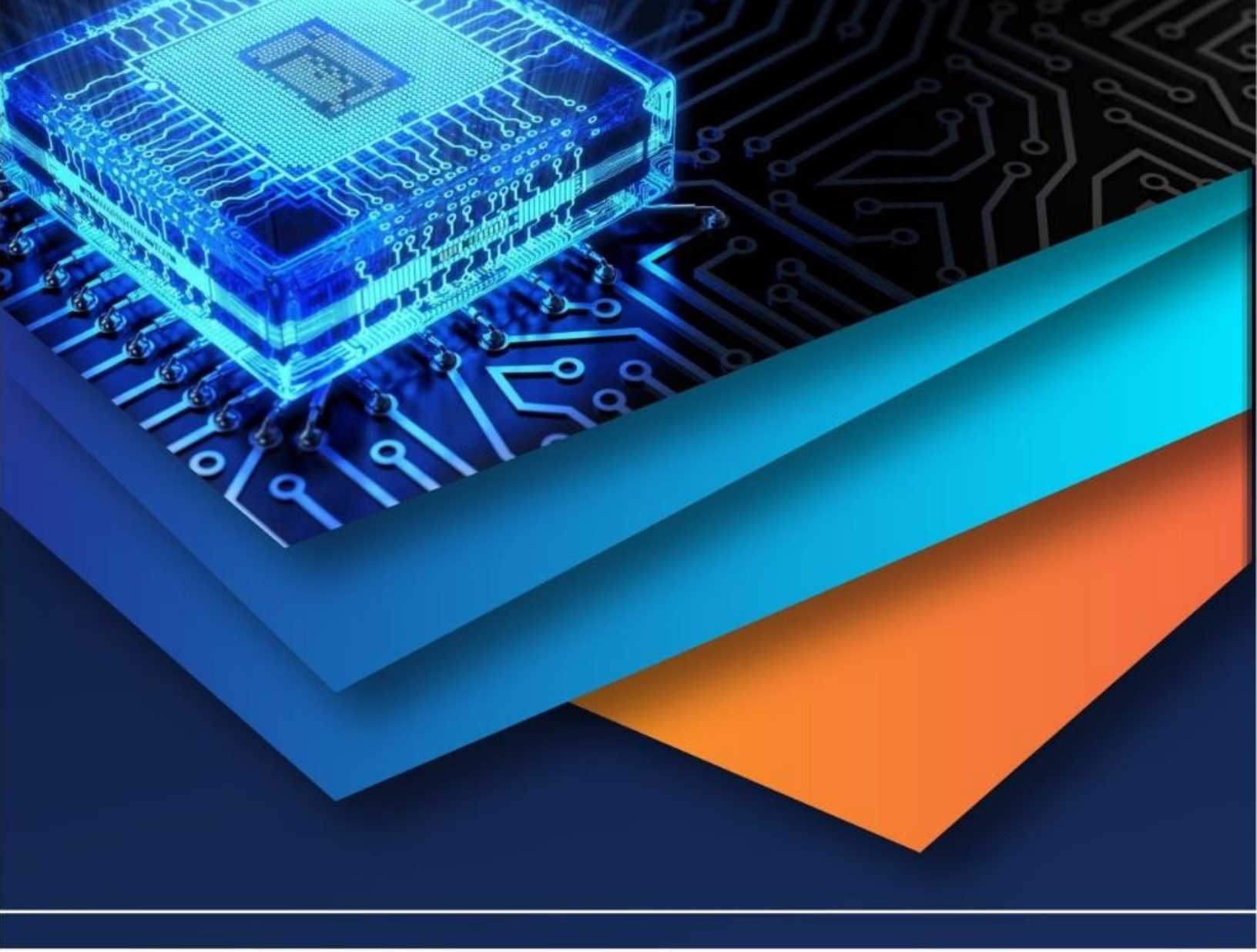
Suggested Publication Targets

1) IRJET
2) IJRASET
3) IEEE Regional Conferences
4) Springer Mid-Tier Conferences
5) UGC CARE Journals

---

Suggested Future Enhancements for Final Submission

1) Add architecture diagrams
2) Include workflow charts
3) Add sample case study
4) Convert into IEEE double-column format
5) Run plagiarism validation
6) Add 2–3 recent research citations from 2024–2026



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY