Int Journal of Engineering and Computer Science

Open access

Review Article

Artificial Intelligence Agents in Software Engineering: Architecture, Applications, Challenges, and Future Directions-A Comprehensive Review

John Gorge*

**Department of Biomedical Engineering, National Center for Health Innovation, London, UK*

***Correspondence:** John Gorge, Department of Biomedical Engineering, National Center for Health Innovation, London, UK, E-mail: jgeore@gmail,com; DOI: 10.1042/JCTCS.4.1.0008

Abstract

Artificial Intelligence (AI) agents have emerged as a transformative paradigm in software engineering by enabling autonomous, goal-oriented, and collaborative execution of software development tasks. Unlike conventional AI tools that provide isolated recommendations, AI agents can perceive their environment, reason over complex objectives, invoke external tools, maintain contextual memory, and execute multi-step workflows with minimal human intervention. Recent advances in Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), tool use, and multi-agent systems have accelerated the adoption of AI agents in software design, coding, testing, deployment, maintenance, and project management. These capabilities have the potential to improve developer productivity, reduce repetitive work, and support intelligent decision-making throughout the Software Development Life Cycle (SDLC).

This review presents a comprehensive analysis of AI agents in software engineering by examining their architectural components, operational principles, and applications across software development activities. The review discusses single-agent and multi-agent architectures, planning and reasoning mechanisms, memory management, tool integration, and human–AI collaboration. It further compares representative AI-agent frameworks, evaluates their strengths and limitations, and identifies challenges related to reliability, security, explainability, privacy, governance, and evaluation. Finally, future research directions including autonomous software engineering, trustworthy AI agents, domain-specific agents, and collaborative multi-agent ecosystems are presented.

Keywords: Artificial intelligence agents, Software engineering, Large language models, Multi-agent systems, Software development life Cycle, Autonomous software engineering, Intelligent agents, Human–AI Collaboration

Received date: Jul 01, 2026; **Accepted date:** Jul 13, 2026; **Published date:** Jul 30, 2026

Citation: John Gorge (2026) Artificial Intelligence Agents in Software Engineering: Architecture, Applications, Challenges, and Future Directions-A Comprehensive Review. Int J Eng Comp Sci. 4: 1.

Introduction

Software engineering has evolved continuously over the past several decades, transitioning from manual programming practices to highly automated development environments. The increasing complexity of modern software systems, together with the rapid pace of technological innovation, has created demand for intelligent tools capable of assisting developers throughout the software development lifecycle. Recent progress in artificial intelligence has given rise to AI agents that extend beyond simple code-completion systems by performing autonomous reasoning, planning, tool execution, and iterative problem solving.

An AI agent is an intelligent software entity that perceives information from its environment, interprets user goals, plans appropriate actions, interacts with external tools or services, and adapts its behavior based on feedback. Modern AI agents often incorporate Large Language Models (LLMs) for reasoning, retrieval systems for accessing external knowledge, memory modules for maintaining context, and execution frameworks that enable interaction with development tools such as integrated development environments (IDEs), version-control systems, testing frameworks, and deployment pipelines.

The emergence of AI agents is changing software engineering practices. During requirements engineering, agents can summarize stakeholder discussions, identify ambiguities, and generate structured specifications. During software design, they can recommend architectures, generate design documentation, and evaluate design alternatives. In implementation, AI agents assist with code generation, code explanation, refactoring, debugging, and automated testing. They also support DevOps activities such as deployment automation, monitoring, and incident analysis, thereby reducing manual effort and improving operational efficiency.

Despite these benefits, AI agents introduce important challenges. Generated outputs may contain logical errors or insecure code, autonomous actions require appropriate oversight, and the use of proprietary source code raises privacy and intellectual-property concerns. Additional issues include explainability, governance, evaluation standards, and resilience against adversarial manipulation. Consequently, responsible deployment requires technical safeguards, organizational policies, and continuous human supervision.

This review synthesizes current knowledge on AI agents in software engineering by examining their architectures, applications, limitations, and future research directions. The article aims to provide researchers and practitioners with a structured overview of this rapidly evolving field while identifying opportunities for future investigation.

Objectives of the review

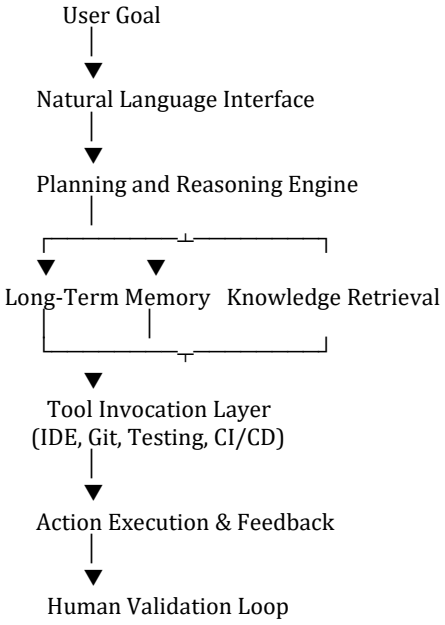
The objectives of this review are:

- To explain the architecture and operational principles of AI agents.
- To analyze applications of AI agents across the Software Development Life Cycle.
- To compare single-agent and multi-agent software engineering approaches.
- To discuss technical, ethical, security, and governance challenges.
- To identify future research opportunities for autonomous software engineering.

Table 1: Applications of AI agents in software engineering.

SDLC Phase	AI Agent Activities	Expected Benefits
Requirements Engineering	Requirement summarization, user-story generation	Improved requirement quality
Software Design	Architecture recommendations, design documentation	Better system design
Implementation	Code generation, refactoring, debugging	Higher developer productivity
Testing	Unit-test generation, defect analysis	Improved software quality
Deployment	CI/CD support, configuration assistance	Faster software delivery
Maintenance	Legacy code analysis, documentation updates	Reduced maintenance effort

Figure 1: Conceptual architecture of an AI agent for software engineering.



References

1) Feduto MA, Maksimovich N Ye, Bon EI, Zimatkin SM. Histological disorders of the neurons of the occipital cortex of the brain of rats with mechanical asphyxia in dynamics. News of Biomed Sci. 2022, 22:93-94.

2) Maksimovich NE, Bon E, Zimatkin SM. The rat brain and its response to ischemia: monograph. Grodno: GrGMU. 2020, 1:240.

3) Paxinos G. The rat brain in stereotaxis coordinates. Academic Press. 1998, 9: 242.

4) Maksimovich N Ye, Bon EI, Zimatkin SM. Central nervous system of the rat: textbook. GrSMU. 2022, 226.

5) Brown BM, Newcombe RG. Non-null semi-parametric inference for the Mann-Whitney measure. J Nonpar Stat. 2009, 743.