Int Journal of Engineering and Computer Science

Open access

*Review Article*

# Generative AI in Software Engineering: Applications, Challenges, and Future Directions-A Systematic Review

**Mohammed Basha***

*Adhara University, AP, India

***Correspondence:** Mohammed Basha, Adhara University, AP, India, E-mail: mbasha99@gmail,com; DOI: 10.1042/JCTCS.1.1.0001

## Abstract

Generative Artificial Intelligence (GenAI) has emerged as one of the most influential technological advancements in software engineering, fundamentally transforming how software systems are designed, developed, tested, deployed, and maintained. The rapid evolution of Large Language Models (LLMs), transformer-based architectures, and multimodal foundation models has enabled developers to automate complex programming tasks, improve software quality, and enhance productivity throughout the Software Development Life Cycle (SDLC). Modern AI-assisted development environments support intelligent code completion, automated documentation, software testing, bug detection, code refactoring, and requirements analysis, thereby reducing development time while improving software reliability.

This systematic review examines the current state of Generative AI in software engineering by analyzing recent research and industrial developments. The review discusses the evolution of Generative AI technologies, core architectural concepts, and their integration into different phases of software development. Major applications—including requirements engineering, software design, code generation, testing, debugging, DevOps, maintenance, and documentation—are explored in detail. Furthermore, the review compares widely adopted AI-assisted development platforms and evaluates their capabilities, strengths, and limitations.

The study also investigates critical challenges such as hallucinated code, software security vulnerabilities, privacy concerns, intellectual property issues, model bias, explainability, and ethical considerations. Finally, emerging research opportunities, including autonomous software engineering, AI-human collaboration, domain-specific language models, and trustworthy AI systems, are discussed. The review concludes that while Generative AI has enormous potential to reshape software engineering practices, responsible governance, robust validation, and human oversight remain essential for ensuring software quality, security, and reliability.

**Keywords**: Generative artificial intelligence, Software engineering, Large language models, Code generation, Software development life cycle

**Received date:** April 22, 2023; **Accepted date:** May 05, 2023; **Published date:** May 15, 2023

**Citation:** Mohammed Basha (2023) Generative AI in Software Engineering: Applications, Challenges, and Future Directions-A Systematic Review. Int J Eng Comp Sci. 1: 1.



## Introduction

Software engineering has continuously evolved in response to advances in computing technologies. Early software development relied heavily on manual coding, structured programming, and traditional software engineering methodologies. The introduction of object-oriented programming, agile development, cloud computing, and DevOps significantly improved software quality and development efficiency. Today, the emergence of Generative Artificial Intelligence represents another major milestone in this evolution.

Generative AI refers to artificial intelligence systems capable of producing new content—including text, source code, images, documentation, and design artifacts—by learning patterns from vast datasets. Unlike conventional machine learning models that primarily perform prediction or classification tasks, Generative AI creates original outputs based on contextual understanding. Recent advancements in transformer-based

architectures and Large Language Models (LLMs) have enabled AI systems to generate human-like programming code, explain algorithms, produce technical documentation, and assist developers throughout the software development lifecycle.

The software industry has rapidly adopted AI-powered development tools to enhance programmer productivity and reduce repetitive tasks. Intelligent programming assistants now support automatic code completion, function generation, debugging, code review, unit test generation, and software documentation. These capabilities enable developers to focus on high-level system design and problem-solving while automating routine programming activities. Consequently, software organizations are increasingly integrating AI-assisted development environments into daily engineering workflows.

Generative AI has applications across nearly every phase of the Software Development Life Cycle (SDLC). During requirements engineering, AI can summarize stakeholder requirements and assist in drafting software specifications. In software design, it can generate UML diagrams, suggest design patterns, and recommend architectural improvements. During implementation, AI-powered code generation accelerates software development by producing syntactically correct code snippets in multiple programming languages. Testing and quality assurance benefit from automated test-case generation, defect prediction, and intelligent debugging. Deployment and maintenance activities are supported through automated documentation, log analysis, infrastructure automation, and predictive maintenance.

Despite these advantages, the adoption of Generative AI introduces several technical and ethical challenges. AI-generated code may contain logical errors, security vulnerabilities, or outdated programming practices. Hallucinations—where AI confidently generates incorrect information—remain a significant concern in software development. Additional challenges include intellectual property disputes, privacy risks associated with proprietary code, model bias, explainability, regulatory compliance, and overreliance on AI-generated solutions. Addressing these issues requires robust validation frameworks, human oversight, and responsible AI governance.

The rapid pace of research in Generative AI has resulted in a growing body of literature spanning academic publications, industrial case studies, and open-source initiatives. However, existing studies often focus on specific applications such as code generation or automated testing rather than providing a comprehensive overview of the technology's role across the SDLC. This review aims to bridge that gap by synthesizing current knowledge, identifying trends, comparing leading approaches, and highlighting future research directions.

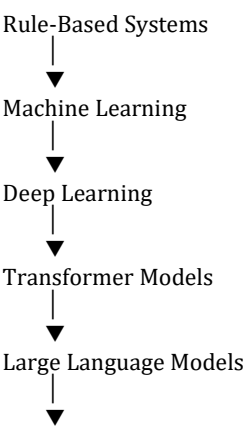
The objectives of this review are to:

- Examine the evolution of Generative AI technologies relevant to software engineering.
- Analyze applications of Generative AI across the software development lifecycle.
- Compare popular AI-assisted software development tools and platforms.
- Discuss technical, ethical, legal, and security challenges.
- Identify future research opportunities and emerging trends.

The remainder of this article is organized as follows. Section 2 describes the review methodology. Section 3 presents the evolution and foundations of Generative AI. Section 4 examines applications across the SDLC. Section 5 compares prominent AI-assisted development tools. Sections 6 and 7 discuss benefits, challenges, security, and ethical issues. Section 8 outlines future research directions, and Section 9 concludes the review.

| Year | Milestone | Significance |
|---|---|---|
| 1956 | Dartmouth AI Workshop | Birth of Artificial Intelligence research |
| 1980s | Expert Systems | Rule-based software assistance |
| 1997 | Statistical Machine Learning | Improved predictive software models |
| 2012 | Deep Learning Breakthrough | Significant gains in AI capabilities |
| 2017 | Transformer Architecture | Foundation for modern Large Language Models |
| 2020–Present | Large Language Models | AI-assisted software engineering, code generation, testing, and documentation |

**Table 1:** Major milestones in the evolution of generative AI for software engineering.



Generative AI for Software Engineering

**Figure 1**: Conceptual evolution of AI in software engineering