ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 3
www.ijisea.org

Generative Artificial Intelligence for Software Development and Automation: A Comprehensive Review

Khagga Dhanikonda[1], **Dr. A. Rama Rao**[2]

1. Assistant Professor, Computer Science and Engineering Department, Akkineni Nageswara Rao College of Engineering and Technology, Gudivada.

2. Professor, Mechanical Engineering Department, Akkineni Nageswara Rao College of Engineering and Technology, Gudivada.

Email-ID: kondadhani@gmail.com[1], ramaraomlec@gmail.com[2]

DOI: 10.5281/zenodo.20082441

Abstract

Generative Artificial Intelligence (GenAI) has emerged as a transformative paradigm in software engineering, fundamentally altering how software is designed, developed, tested, and maintained. Leveraging large language models (LLMs), deep learning, and transformer-based architectures, generative AI enables automation of coding tasks, enhances developer productivity, and introduces new workflows such as AI-assisted programming and autonomous agents. This paper presents a comprehensive review of generative AI in software development and automation, synthesizing recent literature on its applications, benefits, limitations, and future directions. The findings suggest that while GenAI significantly improves efficiency and accelerates development cycles, it also introduces challenges related to code quality, security, and reliability. The study emphasizes the importance of human-in-the-loop systems and governance frameworks for responsible adoption.

Keywords: Generative AI, Software Development, Automation, Large Language Models, Code Generation, DevOps

1. Introduction

Software development has evolved from manual coding practices to automated pipelines. The emergence of generative AI represents a paradigm shift where machines actively participate in code generation, testing, and maintenance.

Generative AI refers to models capable of producing new content such as text and code using deep learning techniques. In software engineering, these models are trained on large datasets of code and natural language, enabling them to generate syntactically correct and semantically meaningful outputs [1, 2].

Recent studies show that generative AI is transforming development workflows and enabling new paradigms such as AI-assisted programming and conversational coding [2, 3, 4]. This paper aims to analyze generative AI across the Software Development Life Cycle (SDLC), evaluate its benefits and challenges, and identify future research directions [22, 23, 24, 25].

2. Principles of Green Chemistry

Green chemistry is based on twelve fundamental principles that guide scientists and industries toward sustainable chemical practices [24].

2.1 Prevention of Waste

It is better to prevent waste generation than to treat or clean up waste after it has been created. Waste prevention reduces environmental pollution and production costs [25].

2.2 Atom Economy

Chemical reactions should maximize the incorporation of all materials into the final product. Higher atom economy minimizes byproduct formation [26].

2.3 Less Hazardous Chemical Synthesis

Chemical processes should use and generate substances with minimal toxicity to human health and the environment [27].

2.4 Designing Safer Chemicals

Chemical products should be designed to perform their intended function while minimizing toxicity [28].

2.5 Safer Solvents and Auxiliaries

The use of harmful solvents and auxiliary substances should be avoided whenever possible [29].

2.6 Energy Efficiency

Energy requirements should be minimized. Chemical processes should ideally operate at ambient temperature and pressure [30].

2.7 Use of Renewable Feedstocks

Renewable raw materials such as biomass, agricultural waste, and natural products should replace non-renewable fossil resources [31].

2.8 Reduce Derivatives

Unnecessary derivatization steps should be minimized because they require additional chemicals and generate waste [32].

2.9 Catalysis

Catalytic reagents are superior to stoichiometric reagents because they improve reaction efficiency and reduce waste generation [33].

2.10 Design for Degradation

Chemical products should be designed to degrade into harmless substances after use [34].

2.11 Real-Time Analysis for Pollution Prevention

Analytical methods should monitor chemical processes in real time to prevent hazardous substance formation [35].

2.12 Inherently Safer Chemistry for Accident Prevention

Chemical processes should minimize the potential for accidents such as explosions, fires, and toxic releases [36].

3. Background and Related Work

Generative AI is built upon transformer-based architectures, which have significantly improved contextual understanding in code generation tasks [26, 27, 28, 29, 30]. Several studies highlight the rapid growth of AI-assisted development tools and their increasing adoption in industry [5, 6].
However, research also indicates mixed results regarding productivity and code quality, suggesting the need for further evaluation [31, 32, 33, 34, 35].

4. Generative AI in SDLC

4.1 Overview of AI Integration in SDLC

Generative AI is integrated across all phases of the Software Development Lifecycle (SDLC), enhancing efficiency and automation [14].

4.2 Requirements Engineering

AI models can convert natural language into structured requirements, though their effectiveness is limited in complex domains [36, 37, 38, 39, 40].

4.3 Design Phase

Generative AI assists in suggesting architectures and design patterns, enabling rapid prototyping [41, 42, 43, 44].

4.4 Implementation

Code generation is the most advanced application, with AI tools significantly reducing development time [10, 45].

4.5 Testing

AI can generate test cases, detect bugs, and suggest fixes, improving software quality [5, 15, **?**].

4.6 Deployment and Maintenance

AI supports CI/CD pipelines, code refactoring, and system monitoring [14].

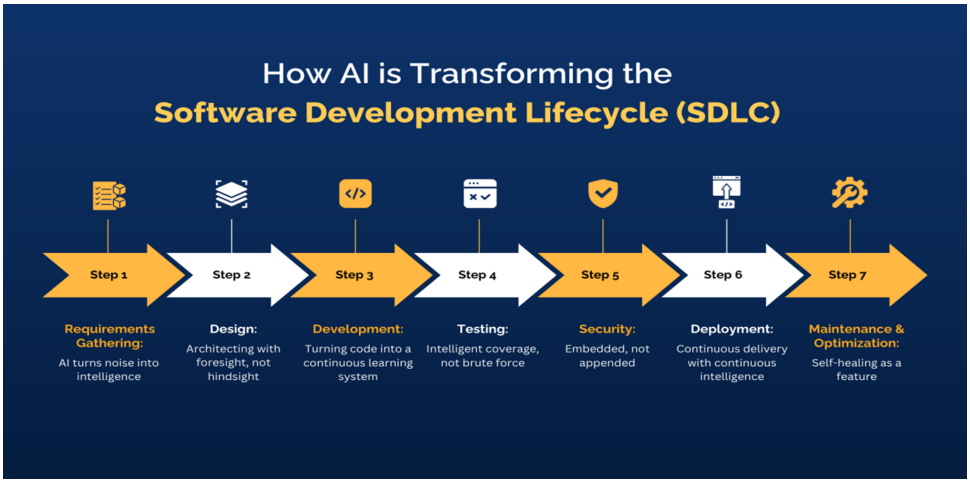

Figure 1: How AI Transforming SDLC

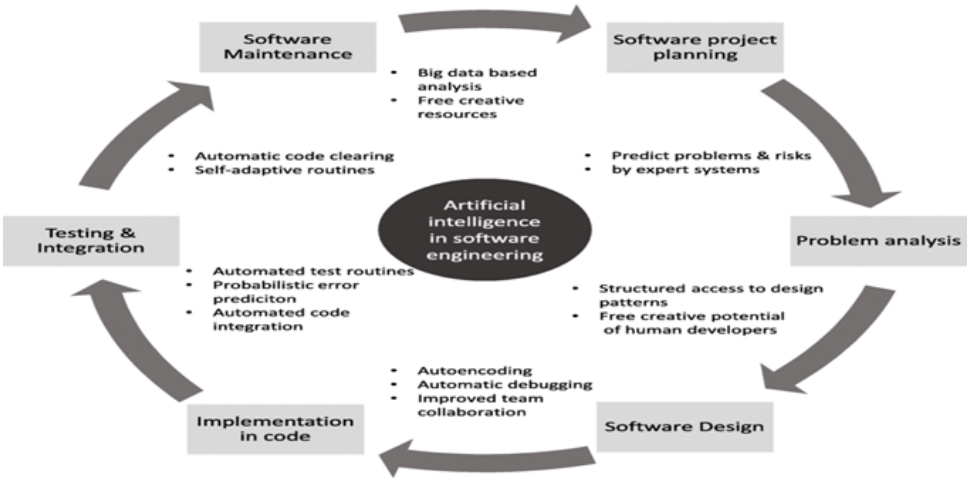

Figure 2: Mind Map of AI in Software Engineering

5. Applications of Generative AI

5.1 Key Applications

Automated Code Generation

Generative AI produces functions and modules from natural language prompts [8].

Code Completion

AI-powered tools provide real-time suggestions, improving coding efficiency [10].

Code Review

AI identifies bugs and enforces coding standards [8].

Documentation Generation

AI automates documentation, improving consistency and reducing effort [9].

DevOps Automation

AI supports infrastructure management and deployment automation [10].

Conversational Programming

Developers interact with AI systems using natural language interfaces [3].

6. Benefits of Generative AI

6.1 Productivity and Efficiency

Table 1: Benefits of Generative AI

Benefit	Description
Increased Productivity	Automates repetitive coding tasks
Faster Development	Reduces development time
Accessibility	Enables non-programmers to code
Knowledge Support	Provides real-time explanations
Collaboration	Acts as AI pair programmer

Generative AI significantly improves productivity by reducing repetitive work and accelerating development cycles [5, 6].

6.2 Impact Summary

- Reduced workload
- Faster delivery
- Enhanced developer support

7. Challenges and Limitations

7.1 Technical Challenges

Table 2: Technical Challenges of Generative AI

Challenge	Description
Code Quality Issues	Incorrect or inefficient outputs
Security Risks	Vulnerable code generation
Context Limitations	Poor understanding of complex systems
Over-Reliance	Reduced developer expertise

AI-generated code often suffers from correctness and optimization issues, requiring human validation [8, 21].

7.2 Productivity Paradox

While AI improves efficiency for beginners, experienced developers may experience slowdowns due to verification overhead [13, 20].

8. Security and Ethical Considerations

8.1 Security Issues

Studies show that AI-generated code may introduce vulnerabilities such as injection flaws and insecure dependencies [11, 19].

8.2 Ethical Concerns

Key ethical issues include:

- Bias in training data
- Intellectual property concerns
- Privacy risks [12, 18]

8.3 Mitigation Strategies

- Human-in-the-loop validation
- Secure coding practices
- Governance frameworks

9. Impact on Developer Productivity

9.1 Positive Effects

- Faster development
- Reduced repetitive tasks

9.2 Negative Effects

- Increased need for code review
- Over-dependence on AI

9.3 Observations

The impact of generative AI depends on developer experience and task complexity [5, 17].

10. Tools and Technologies

Table 3: Popular Generative AI Tools

Tool	**Features**
GitHub Copilot	Code suggestions
OpenAI Codex	Code generation
Amazon CodeGuru	Code review
Tabnine	AI autocomplete
DeepCode	Static analysis

These tools demonstrate the growing integration of AI into development workflows [10, 16].

11. Future Research Directions

Future work should focus on:

- Improving reliability and correctness of generated code
- Enhancing human-AI collaboration
- Developing explainable AI models
- Establishing standardized evaluation metrics
- Addressing ethical and security concerns

12. Conclusion

Generative AI is revolutionizing software development by enabling intelligent automation, improving productivity, and transforming workflows. However, challenges related to security, quality, and ethics must be addressed.

Future systems should emphasize collaboration between humans and AI to ensure reliable and responsible software development.

References

[1] M. Alenezi, "AI-Driven Innovations in Software Engineering," *Applied Sciences*, 2025.

[2] V. Acharya, "Generative AI and Software Development," *arXiv*, 2025.

[3] L. Banh, "Copiloting the Future," *Information and Software Technology*, 2025.

[4] V. Gurgul et al., "State of Generative AI in Software Development," *arXiv*, 2026.

[5] A. Mohamed et al., "Impact of LLM Assistants on Productivity," *arXiv*, 2025.

[6] M. Coutinho et al., "Generative AI Productivity Study," *arXiv*, 2024.

[7] D. Hussein, "Usability of LLMs in Software Engineering," 2024.

[8] ResearchGate, "AI-Assisted Code Generation Review," 2025.

[9] ResearchGate, "Generative AI Study," 2025.

[10] IJFMR, "AI Coding Tools Overview," 2024.

[11] C. Negri-Ribalta, "AI and Software Security," 2024.

[12] SecureFlag, "Risks of Generative AI Coding," 2024.

[13] METR, "AI Productivity Study," 2025.

[14] ResearchGate, "AI in SDLC," 2025.

[15] ResearchGate, "Generative AI in Software Engineering," 2025.

[16] S. Koteswara Rao, "AMC Integrated CPW Fed Antennas for Bio-Communication: Design Trends and Performance," *AEU - International Journal of Electronics and Communications*.

[17] S. Koteswara Rao, "Metamaterial Absorber for L, S and C Band Applications," *Journal of Circuits Systems and Computers.*

[18] S. Koteswara Rao et al., "Development of CPW Fed Slot Antenna with CSRR for Biomedical Applications," *Journal of Circuits, Systems, and Computers.*

[19] S. Koteswara Rao et al., "A Novel SegNet Segmentation with MobileNet Brain Tumor Classification Using MRI Images," *SN Computer Science, Springer.*

[20] Dr. Koteswararao Seelam et al., "Medical Image Registration with Object Deviation Estimation through Motion Vectors Using Octave and Level Sampling," *Automatika (Taylor & Francis).*

[21] Dr. Koteswararao Seelam et al., "Cluster Based Energy Efficient Optimal Relay Selection Strategy for Multi Hop Reliable Cooperative Communication in Vehicular Communication," *International Journal of Intelligent Engineering and Systems.*

[22] Dr. Koteswararao Seelam et al., "Energy Efficient Design and Implementation of Approximate Adder for Image Processing Applications," *Serbian Journal of Electrical Engineering.*

[23] Dr. Koteswararao Seelam et al., "An Improved BAT-Optimized Cluster-Based Routing for Wireless Sensor Networks," *Intelligent Computing and Applications.*

[24] Dr. Koteswararao Seelam et al., "Implementation of Intelligent Smart Heart Health Monitoring System using IoT," *International Journal on Recent and Innovation Trends in Computing and Communication.*

[25] Dr. Koteswararao Seelam et al., "Performance Evaluation of Deep Learning Autoencoder in Single and Multi-Carrier Systems," *International Journal on Recent and Innovation Trends in Computing and Communication.*

[26] Dr. Koteswararao Seelam et al., "Design of Turbo Trellis Coding Modulation Scheme of Rate 4/9 for Rician Fading Channel," *International Journal on Recent and Innovation Trends in Computing and Communication.*

[27] S. Koteswara Rao et al., "Data Analysis Framework with an Associated Classification Model for Analyzing Cybercrime Underground Economy," *Journal of Engineering Sciences.*

[28] Manjunath B E et al., "Detection of Social Network Mental Disorders Through Mining of Online Social Media," *Journal of Engineering Sciences.*

[29] S. Koteswara Rao and P. Ramesh, "IoT based Smart Stove Safety System," *International Journal of Analytical and Experimental Modal Analysis.*

[30] Koteswararao Seelam et al., "An Efficient Hybrid BAT-Optimized Clustering for Wireless Sensor Networks," *International Journal of Electronic and Communication Technology.*

[31] Koteswararao Seelam et al., "Performance Analysis of LEACH, COTS and MST Algorithms in Cluster Formation," *IJCSN International Journal of Computer Science and Network.*

[32] Koteswararao Seelam et al., "Implementation of Multi-hop Cluster Base Routing Protocol for Wireless Sensor Networks," *International Journal of Computer Applications.*

[33] Koteswara Rao Seelam et al., "An Adaptive CSMA / TDMA Hybrid–MAC for Wireless Sensor Networks," *CIIT International Journal of Networking and Communication Engineering.*

[34] Koteswara Rao Seelam et al., "An Adaptive Power Control and Energy Efficient MAC Protocol for Wireless Sensor Networks," *International Journal of Computer Science and Application.*

[35] Koteswara Rao Seelam et al., "Energy Aware TDMA MAC for Wireless Sensor Networks," *International Journal of Distributed and Parallel Systems.*

[36] Koteswara Rao Seelam et al., "Performance Evaluation of Wireless Sensor Network Routing Protocols for Real Time Application Support," *Global Journal of Computer Science and Technology.*

[37] Koteswara Rao Seelam et al., "Prevention of Shared Root Node Attack in MAODV," *International Journal of Electronic and Communication Technology.*

[38] Koteswara Rao Seelam et al., "Sensor Networks Simulation in NS2.26," *International Journal of Electronic and Communication Technology.*

[39] Koteswara Rao Seelam et al., "An Efficient Distance-Energy-based Minimum Spanning Tree (DE-MST) for Wireless Sensor Networks," *International Journal of Computer Applications.*

[40] Koteswararao Seelam and B.S.L Gayathri, "Implementation and Verification of Low Latency and Low Power MAC Protocol for Wireless Sensor Networks," *IJMER.*

[41] Koteswararao Seelam and Ch. Mounica, "Evaluation of Reactive Routing Protocols for Wireless Sensor Networks," *IJEIT.*

[42] Koteswararao Seelam et al., "Optimized Super Resolution Reconstruction Framework For Cardiac MRI Images Perception," *IJCAT.*

[43] Unnava Divya and Koteswarararao Seelam, "Reduction of Effect of Timing Jitter on High Speed OFDM System Using Oversampling Technique," *IJRAT.*

[44] Ch. Ravikiran and S. Koteswararao, "Automatic Wavelet Based Nonlinear Image Enhancement Using WDRC for Aerial Imagery," *IJRAT.*

[45] Mr. Ashok Reddy and S. Koteswararao, "Improved CSMA/TDMA Hybrid-MAC for Wireless Sensor Networks," *IJRAT.*