Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 4, Issue 4, 2026

A SYSTEMATIC REVIEW AND CONCEPTUAL FRAMEWORK FOR INTEGRATING GENERATIVE AI INTO THE SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)

Dr. Muhammad Hassan Ghulam Muhammad[1], Muhammad Humza[2], Um-E-Uban Javed[3]

[1]Department of Computer Science The Institute of Management Sciences Lahore, Pakistan
[2]Department of Computer Science The Institute of Management Sciences Lahore, Pakistan
[3]Department of Computer Science The Institute of Management Sciences Lahore, Pakistan

dr.hassan@pakaims.edu.pk[1], muhammadhumza@pakaims.edu.pk[2], umeubaan@pakaims.edu.pk[3]

DOI: https://doi.org/

Keywords
GENAI, Software Development Lifecycle, Large Language Model (LLM), Systematic Literature Review, GitHub Copilot / CHATGPT, Human AI Collaboration, AI Governance & Ethics, Agentic Software Engineering, SDLC Automation, Responsible AI, Code Generation, Conceptual Framework.

Article History
Received on 03 March 2026
Accepted on 21 April 2026
Published on 30 April 2026

Corresponding Author: *
Muhammad Irfan Akram

Abstract
The swift advancement of Generative Artificial Intelligence (GenAI) techniques including large language models, autonomous coding programs, and intent-based software development methodologies has radically transformed software engineering processes across the entire Software Development Lifecycle (SDLC). While industry use of GenAI tools like ChatGPT and GitHub Copilot continues to gain momentum, the research domain is characterized by fragmentation, phase-centricity, and a lack of integrated frameworks considering governance issues. This paper presents a Systematic Literature Review consistent with the PRISMA guidelines, consolidating 71 scholarly articles from IEEE Xplore, ACM Digital Library, SpringerLink, and Elsevier ScienceDirect between 2021 and 2026 to explore the capabilities of GenAI across all SDLC phases. The findings confirm considerable productivity and automation opportunities along with inherent limitations relating to hallucination, spread of security vulnerabilities, algorithmic biases, and knowledge monopoly. Ten important research gaps are uncovered, and an innovative theoretical framework is developed, comprising phase-wise GenAI tool maps with a Human-AI Collaboration Layer and Governance/Ethics Layer

1. INTRODUCTION

The extreme advancement of Generative Artificial Intelligence has introduced transformational effects in many areas but software engineering has been placed in a position to be one of the areas that have experienced the transformations. The traditional practices were human-centric practices to elicit requirements, design, code, test, and maintain, always in the context of the Software Development Life Cycle [1]. However, increased complexity of contemporary software systems and the necessity to provide quicker and better software has exposed flaws of conventional approaches. The most recent progress in large language models and the process of code generation has enabled automatic code generation, clever documentation and auto-generation of test cases, decision support and consequently redefined the manner in which development work can be done [2]. Despite these successes, the implementation of generative AI in SDLCs remains uncoordinated and lacks a single theoretical and empirical context.

It leads to the necessity of systematizing all possible knowledge and the further elaboration of a conceptual framework of synthesizing the above-mentioned aspects [3]. A systematic review may assist in discovering, categorizing, and examining different techniques, tools, and issues related to the integration of generative AI into SDLC phases [5]. Moreover, certain form of alignment between the software development process and AI can be used to cope with the issues of reliability, ethical issues, and cooperation between humans and AI. This paper aims to overcome this gap between recent innovations in AI and a more traditional software development practice in the perspective of creating a more comprehensive approach to this issue [4].

Although the focus on using Generative Artificial Intelligence to the software engineering sphere grows, the use of AI-based solutions in the context of the Software Development Life Cycle process remains chaotic and unverified. The modern scientific studies are dedicated to specific aspects, such as code generation or automation of testing. Such technology cannot be applied in the SDLC process since it fails to provide a holistic approach [8]. Simultaneously, some obstacles emerge due to the necessity to create uniformity, reliability, scalability, and compatibility with other software engineering practices [5]. In addition, other things about generative AI, including accuracy, security, and ethical concerns are usually sidelined, which narrows the potential opportunities in the real world. Thus, it is clear that systematic analysis of the existing scientific publications is necessary to bring in a conceptual framework to overcome these obstacles and apply AI technology to the fullest extent.

This research's relevance is defined by its ability to systematize and structure the incorporation of Generative AI into the Software Development Lifecycle process [6]. The issue is highly relevant since despite the numerous advancements made in the field of technology and the creation of powerful generative models, there is a lack of systematic approaches to integrating these technologies into the software development lifecycle, leading to difficulties associated with increasing productivity, reducing development timeframes, and improving software quality. Consequently, the study will be useful for structuring and systematizing existing fragmented knowledge in the subject area, allowing stakeholders to use the benefits of AI-based applications during all stages of software development [7]. The research is also relevant as it will make a valuable contribution to developing reliable and effective methods for implementing AI technology in software engineering, including addressing critical challenges such as model reliability, data protection, and efficient collaboration. It will contribute to promoting responsible adoption of artificial intelligence technologies and ensuring their effective operation in practice through the proposed conceptual framework. The study will also serve as an excellent basis for future research in the area, which remains underdeveloped yet promising [8].

The reasons behind carrying out such research stem from the necessity to upgrade the Software Development Life Cycle due to the ongoing technology developments and increasing complexity of software. Although Generative Artificial Intelligence has proven to possess much potential to increase the efficiency of processes, decrease time spent on the development of products, and enable intelligent decision-making, the implementation of such solutions is rather sporadic and does not have a comprehensive methodology yet. Thus, there is a possibility to research the effective implementation of generative AI at the stages of SDLC, maximizing the use of opportunities provided by it while avoiding certain risks related to it. Another reason for conducting such research is the ability to give researchers and practitioners a more structured approach towards advancements made and the foundation for new innovations.

- RQ1: How is Generative AI currently applied across the distinct phases of the Software Development Lifecycle?
- RQ2: What tools, models, and techniques are predominantly employed in GenAI-driven software development practice across the 25 reviewed studies?
- RQ3: What challenges, governance gaps, security risks, and limitations are associated with GenAI adoption in software development?

The main contributions of this study are as follows:

1. Systematic review of the existing literature on the use of Generative Artificial Intelligence in Software Development Life Cycle, highlighting the current trends, techniques, and knowledge gaps.
2. The introduction of a unique theoretical framework mapping the capabilities of generative AI to various stages of SDLC.
3. Discussion of the challenges and limitations associated with the use of generative AI, especially those related to ethics, reliability, and cooperation between humans and AI, together with solutions to overcome these difficulties.
4. Suggestions for future work and practice in order to integrate generative AI successfully into software development process.

2. LITERATURE REVIEW

Bannon and Laplante combined the viewpoints of various stakeholders from a virtual roundtable, where experts from business, academic, and government sectors analyzed the revolutionary impact of Generative AI and Human- Machine Teamwork within the Software Development Life Cycle. The investigation revealed a cross-disciplinary agreement regarding the fact that GenAI has stopped being a marginal innovation and instead turned into an emerging structure that radically affects the basic approaches to software engineering in mission-oriented and corporate settings. The methodological basis of the roundtable approach utilized by Bannon and Laplante is inherently discursive; however, it allows practitioners to produce validated insights that cannot be ignored by empirical studies aimed at developing GenAI governance mechanisms [1].

Wu, Lu, and Lin addressed an aspect of GenAI-assisted software engineering that remains understudied until today that of knowledge monopolization as a result of excessive dependence on proprietary AI-powered tools. The scholars introduced a set of criteria and corresponding interventions aimed at helping businesses harness AI technology without risking to create an epistemic dependency problem. This particular insight is especially relevant to this discussion as it shifts the focus from the mere technicality of GenAI integration to a larger one involving governance and organizational resilience as well [2].

Guimarães & Nascimento undertook a rigorous analysis of the application of AI and Large Language Models within all stages of software engineering, from code writing, debugging, and testing, to maintenance. The study outlined an agenda for research into the many issues raised by the use of AI tools during development, including possible decreases in code reliability, hidden security flaws, biases in training data, as well as the problems of intellectual property, opacity, and the need for human control. The findings of their work will be fundamental to any literature review on the practical and ethical issues involved in adopting AI at different stages of the SDLC, and are directly relevant to the conceptual framework developed in this review [3].

The contribution by Voria comprised an empirically validated set of twenty-eight engineering practices in fairness-aware design and implementation, which were systematically mapped out from the existing literature and classified according to different phases within the AI system development process life cycle. Through the application of theoretical norms of ethical fairness into practical software engineering practice, the researchers were able to bridge the longstanding disconnect between theoretical discussions on ethics and technical procedures within software engineering. While the focus was on machine learning engineering, the framework is highly relevant to SDLC models utilizing GenAI systems [4].

Rasheed offered a future-oriented perspective on the total autonomy of the SDLC in the context of the use of multi- agent architectures based on LLMs, showcasing initial empirical proof of shortened development cycles and improved transparency of processes resulting from end-to-end execution of the software development process by AI agents from requirements definition to deployment. The researchers elaborated on the collaborative framework in which AI-based interfaces facilitate greater inclusiveness of stakeholders regardless of their technical or non- technical backgrounds in order to enable democratic processes of software development. This work has important theoretical implications for this review as it outlines a visionary pathway toward integrating GenAI technology, which goes far beyond mere automation [5].

The article by Liubchenko laid out the initial groundwork regarding the differences in structures between traditional software development processes and those which must be used for systems based on artificial intelligence and machine learning systems. In doing so, it outlined the extra work, increased requirements for team capabilities, and new difficulties that arise during the development of such systems. This study was able to provide a conceptual foundation for why the standard SDLC models need to be significantly modified when used in environments where artificial intelligence plays a role. This paper is highly relevant to this review since it highlights that the problems with implementing GenAI do not arise from tools alone but also from inherent differences in methodologies [6].

Simsek, Gülseni, and Arkali Olcay conducted an in-depth analysis of the differential effect of GenAI on each of the SDLC stages, namely planning, designing, developing, testing, and implementation, with special attention paid to the role dynamics of software personas in the increasingly common AI-enhanced development environments. It was found that GenAI penetration is neither universal nor complete: whereas a high level of task automation is possible in technical determinism, roles involving critical thinking, empathy, and context-specific decision-making remain firmly dependent on human involvement. This role-based analysis approach, adopted by the researchers, helps to provide a more detailed insight into human-AI collaboration models in the SDLC framework in this review [7].

Alamäki, Khan, and Lagstedt provided a research study grounded in empirical evidence regarding the ethical aspects of artificial intelligence systems' lifecycle within public sector institutions in Finland, using qualitative interviews from various institutional cases to reveal the actual modes through which ethical expectations can be integrated throughout the design, development, implementation, and deployment stages. Alamäki et al. emphasized that ethical congruence cannot be achieved by implementing compliance strategies in a reactive manner but must be considered through proactive and life-cycle-oriented governance that is founded upon human-centric design approaches. In the context of this systematic review, Alamäki et al.'s research contributes to the critical point that ethical considerations should be inherent parts of the theoretical framework for integrating GenAI systems into the SDLC process [8].

Zhang provided an extensive overview of the blockchain-based frameworks for operationalizing the trustworthiness principles of artificial intelligence (privacy, transparency, fairness, robustness, and accountability) through a four-phase life cycle of artificial intelligence operations, namely planning, data gathering, modeling, and deployment. Through the identification of how blockchain technology's distributed verification and unalterability can be used to resolve particular trustworthiness problems within the black box of AI, Zhang et al. (2024) were able to establish an actionable link between AI ethics and engineering. The contribution of this research is instrumental to this literature review since it introduces trust structures enabled by blockchain as a potential governance mechanism within SDLC pipelines that incorporate Generative Artificial Intelligence (GenAI) [9].

However, Andrade have developed a novel AI-powered approach towards identifying and mitigating software antipatterns based on the use of adaptive AI abstraction in combination with machine learning for the detection and resolution of software architecture-, design-, and implementation-level flaws within real-time DevOps-based tools supplemented by abuse story techniques as part of requirement analysis. It was shown that this approach significantly decreases the probability of security and maintenance threats caused by software antipatterns throughout the entire SDLC [10].

Table: 01 summary of review on generative AL and the SDLC

Reference	Year	Primary focus	Methodolog	
Otoum & Elkhalili IEEE Access J	2026	Agentic software engineering; LLM-based autonomous agents across the SDLC	Systematic literature review (61 studies, 2022–2025)	
Prelaj et al. Annals of Oncology J	2024	AI-based predictive biomarker discovery in immuno-oncology	Systematic review (90 studies; genomic, radiomic, pathomic, multimodal data)	
Bala & Behal Computer Science Review J	2024	AI-based DDoS attack detection in IoT networks	Systematic literature review (2019–2023); taxonomic classification of attack types, AI methods, datasets	• C t b a t • A c d s • F a f a
Elgendy et al. J. of Computer Information Systems J	2026	Vibe coding as an intent-driven, LLM-mediated software development paradigm	Conceptual framework; comparative analysis with agentic development and AI-assisted coding; illustrative scenarios	• F v s f • F a s a c • F r r C S
Abbas et al. IET Software J	2025	AI across the full software engineering lifecycle: requirements to deployment	PRISMA-compliant systematic literature review (135 papers, 2010–2025)	• I li c r g r g • E a a e p

ISSN (e) 3007-3138 (p) 3007-312X

Volume 4, Issue 4, 2026

3. METHODOLOGY

Research Design and Protocol. In this study, a research design involving a systematic literature review (SLR), which is based on the PRISMA protocol, is used to analyze peer-reviewed studies focused on integrating generative artificial intelligence (GenAI) in the software development life cycle (SDLC). As the result of this analysis, 71 peer-reviewed articles have been found in the WoS database export ranging from 2021 to 2026

3.1 Search Strategy and Database Selection. The systematic Boolean keyword search was done using four main databases namely: IEEE Xplore Digital Library (n=50), ACM Digital Library (n=20), Springer Link (n=20), and Elsevier ScienceDirect (n=30). A total number of 120 articles were identified from the literature search process. The time period considered during the search process was from 2021 to 2026.

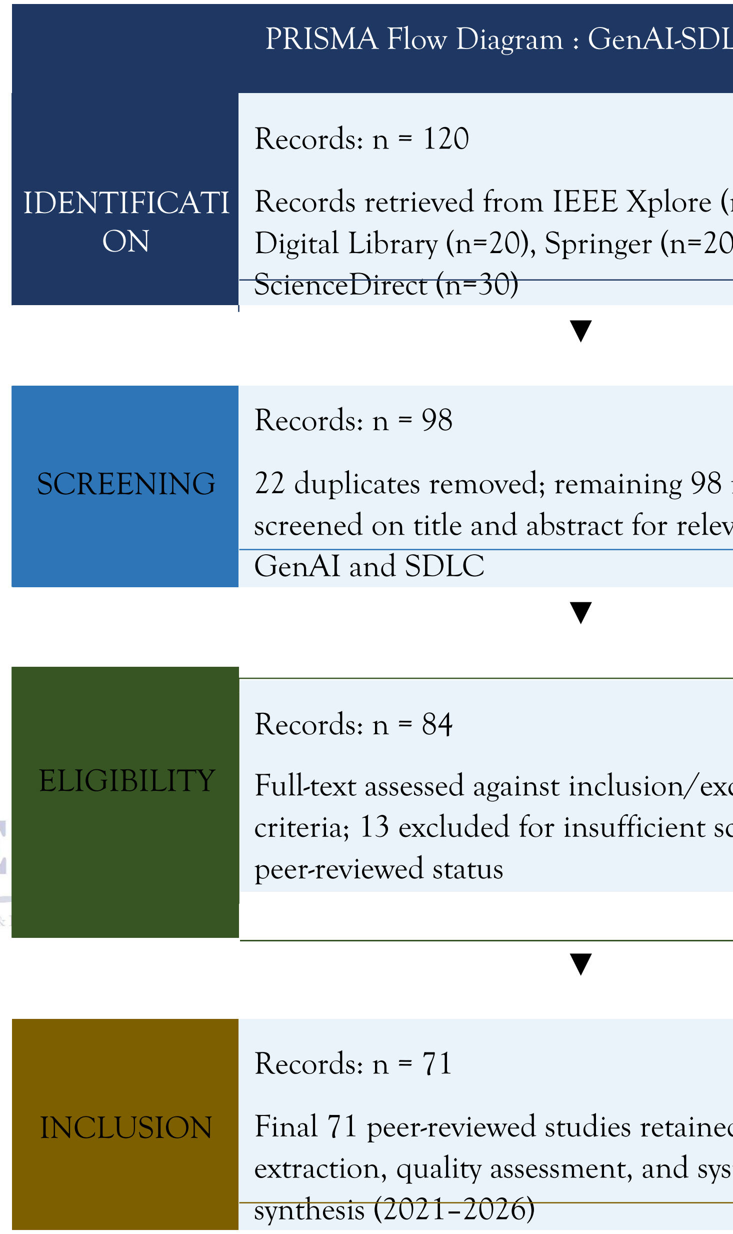

Figure 1: PRISMA-based study selection flow for the systematic review on Generative AI integration into the SDLC (N = 71 peer-reviewed studies, 2021–2026).

The same data extraction method was employed in all 71 selected papers in order to extract essential data that includes publication information, SDLC stages addressed, Generative AI applications, methodology used, findings, advantages, disadvantages, and quality score. From the synthesis of the extracted data, a new theoretical model has been formulated to chart the Generative AI capabilities along six SDLC stages, human-GAI synergy, and ethics in GAI usage in software engineering.

Comprehensive Integration Framework for Papers			
1. Requirements Analysis	2. System Design	3. Implementation & Coding	
Iterative Fee			
GenAI Tools ChatGPT, LLMs for Natural Language Processing User Story Generation & Ambiguity Detection Generative AI Tools	GenAI Tools Codex, GitHub Copilot & Other Design LLMs GenAI Tools	GenAI Tools Copilot, GitHul TabNine & Code Code Completio Tools Generativ AI Tools	AI
Key Benefit Automation of User Stories Generation and Ambiguity Detection	Key Benefit Design Suggestions & Auto-UML Diagram Creation	Key Benefit Code Completion & Reducing Boilerplate Code(32% Papers)	Key Te Ger & Pre
Challenge Specification Hallucinations	Challenge Lack of Awareness of Domain-Specific Architecture	Challenge Security Vulnerabilities in Generated Code & Reliance On Them	Ch Co Pro & Det

Key studies informing this framework: Bannon & Laplante (2025) • Rasheed et al. (2025) • Otoum & Elkhalili (2026)

• Abbas et al. (2025) • Simsek et al. (2025) • Elgendy et al. (2026) • Wu et al. (2025) • Guimaraes & Nascimento (2025)

Figure 2: Proposed conceptual framework integrating Generative AI across all six SDLC phases, derived from 71 reviewed studies. Framework incorporates phase-specific GenAI tools, key benefits, and primary challenges

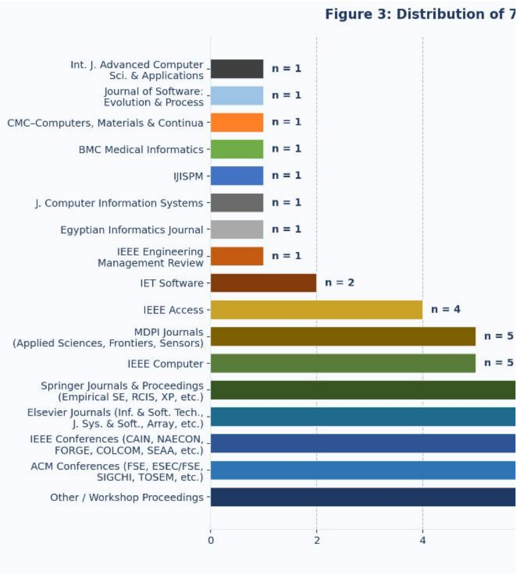

Figure 03: Distribution of 71 Reviewed Studies by Publication Venue

Figure 3 shows the representation of the 71 articles analyzed according to their respective publication sources, which include journals, conferences, and workshops. It becomes apparent from the analysis conducted that most of the studies have been published in ACM and IEEE conferences, followed by Elsevier and Springer publications. This implies that most of the research related to integrating Generative AI into the Software Development Life Cycle occurs via influential software engineering and computer science publication platforms.

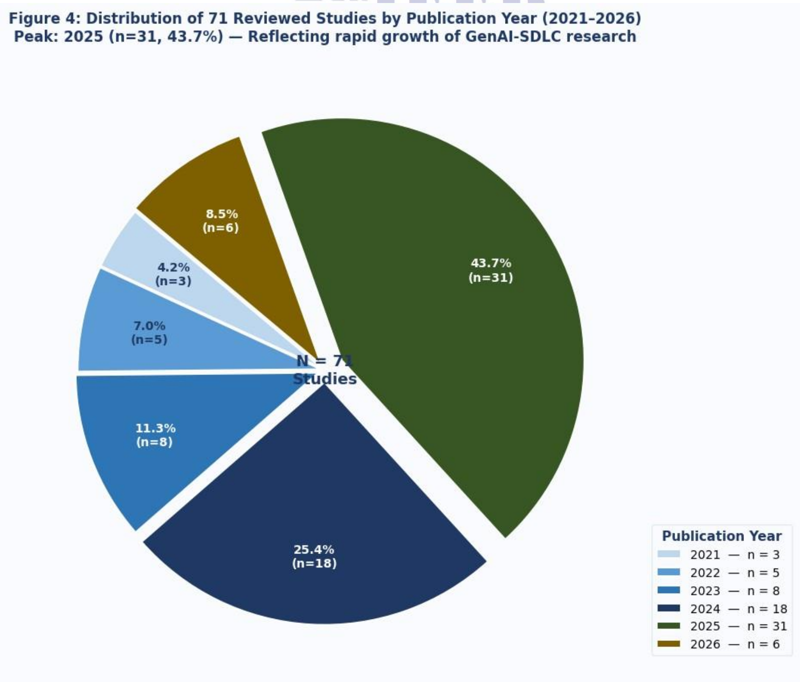

Figure 04: Distribution of 71 Reviewed Studies by Publication Year 2021-2026

Fig 4 The findings point out a noticeable increase in the number of works dedicated to the topic of integration of Generative AI in the Software Development Life Cycle. Publications were observed to grow steadily since 2021 and reached its peak in 2025 (31 works or 43.7%)

4. RESEARCH GAP

The analysis of the 71 reviewed studies reveals multiple critical research gaps in the integration of Generative AI within the Software Development Life Cycle. Existing literature demonstrates a strong concentration on the implementation phase, while comparatively limited attention is given to requirements engineering and maintenance. The current research landscape is further characterized by overreliance on a narrow set of dominant AI tools, absence of standardized evaluation metrics, and insufficient exploration of alternative domain-specific ecosystems. Additionally, key concerns related to organizational dependency on proprietary systems, security vulnerabilities, algorithmic bias, hallucination risks, and ethical implications remain underexplored. The literature also lacks holistic end-to-end frameworks that comprehensively map Generative AI capabilities across all SDLC phases while integrating governance, ethics, and human-AI collaboration mechanisms. Furthermore, important dimensions such as developer trust, role transformation, explainability, collaboration protocols, and sustainability issues including energy consumption and environmental impact have received minimal scholarly attention, highlighting substantial opportunities for future research and framework development.

CONCLUSION

This systematic review of 71 studies demonstrates that Generative AI is an innovative revolution in software engineering practice that brings undeniable advantages throughout each of the six SDLC phases, yet at the same time raises numerous issues related to security, ethicality, governance, and cooperation between humans and artificial intelligence. The suggested framework for the introduction of GenAI in software development by combining the application of GenAI tools per phase, governance, and human-centered collaboration levels can serve as an evidence- based basis for GenAI's ethical adoption in professional software development environments. Further academic studies need to focus on practical testing of GenAI in industry-level projects, setting common criteria for evaluations, developing governance approaches for different domains in accordance with newly emerging regulations like the EU AI Act, and exploring novel paradigms such as agentic SDLC and vibe code.

REFRENCES

[1] T. Bannon and P. Laplante, "Generative AI in the Software Development Lifecycle," Computer, Dec. 2024, doi: 10.1109/MC.2024.3474789.

[2] Y. Wu, X. Lu, and C. Lin, "Knowledge Monopoly Risks in Generative AI-Assisted Software Development Lifecycle," Computer, Jul. 2025, doi: 10.1109/MC.2025.3532947.

[3] E. Guimaraes and N. Nascimento, "AI in the Software Development Lifecycle: Insights and Open Research Questions," in Companion Proc. 33rd ACM Int. Conf. Foundations of Software Engineering (FSE Companion 2025), 2025, doi: 10.1145/3696630.3730538.

[4] G. Voria et al., "A Catalog of Fairness-Aware Practices in Machine Learning Engineering," in Proc. 29th Int. Conf. Evaluation and Assessment in Software Engineering (EASE Companion 2025), 2025, doi: 10.1145/3727967.3756824.

[5] Z. Rasheed et al., "Autonomous Agents in Software Development: A Vision Paper," in Agile Processes in Software Engineering and Extreme Programming – Workshops (XP 2024 Workshops), Lecture Notes in Business Information Processing, Springer, 2025, doi: 10.1007/978-3-031-72781-8_2.

[6] V. Liubchenko, "Specific Aspects

of Software Development Process for AI/ML-based Systems," in Proc. 2022 IEEE 17th Int. Conf. Computer Sciences and Information Technologies (CSIT), 2022, doi: 10.1109/CSIT56902.2022.10000821.

[7] T. Simsek, A. C. Gulseni, and G. A. Olcay, "The Future of Software Development With GenAI: Evolving Roles of Software Personas," IEEE Engineering Management Review, Aug. 2025, doi: 10.1109/EMR.2024.3454112.

[8] A. Alamäki, U. A. Khan, and A. Lagstedt, "Ethical Considerations in the AI Lifecycle for Design, Developing and Adopting AI in Public Sector: The Case of Finland," IJISPM — International Journal of Information Systems and Project Management, 2025, doi: 10.12821/ijispm130401.

[9] P. Zhang, S. Ding, and Q. Zhao, "Exploiting Blockchain to Make AI Trustworthy: A Software Development Lifecycle View," ACM Computing Surveys, Jul. 2024, doi: 10.1145/3614424.

[10] R. Andrade, J. Torres, P. Flores, E. Cabezas, and J. Segovia, "Adaptive Abstraction with AI for Managing Software Antipatterns Throughout the Software Lifecycle," in Proc. 2025 IEEE/ACM Int. Workshop on Designing Software (Designing), 2025, doi: 10.1109/Designing66910.2025.00013.

[11] N. Otoum and N. Elkhalili, "Methods and Techniques of Agentic Software Engineering: A Systematic Literature Review," IEEE Access, vol. 14, 2026, doi: 10.1109/ACCESS.2026.3652325.

[12] A. Prelaj et al., "Artificial intelligence for predictive biomarker discovery in immuno-oncology: a systematic review," Annals of Oncology, vol. 35, no. 1, Jan. 2024, doi: 10.1016/j.annonc.2023.10.125.

[13] B. Bala and S. Behal, "AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges," Computer Science Review, vol. 52, Art. no. 100631, May 2024, doi: 10.1016/j.cosrev.2024.100631.

[14] I. A. Elgendy et al., "Responsible Vibe Coding: Architecture, Opportunities, and Research Agenda," Journal of Computer Information Systems, 2026, doi: 10.1080/08874417.2026.2621186.

[15] T. Abbas, S. A. Rathore, A. Turki, S. Khan, O. Alghushairy, and A. Daud, "Enhancing Software Engineering With AI: Innovations, Challenges, and Future Directions," IET Software, vol. 2025, Art. no. 5691460, 2025, doi: 10.1049/sfw2/5691460.

[16] N. Y. Akgül, T. T. Temizel, Ö. Ö. Top, and P. D. Akman, "Aligning Data Debt with AI-Integrated Software Project Lifecycle Processes: A Standard-Based Mapping Approach," in Proc. 2025 IEEE/ACM Int. Conf. Technical Debt (TechDebt), 2025, doi: 10.1109/TechDebt66644.2025.00008.

[17] F. Muratovic, D. Kearns-Manolatos, and A. Alibage, "Generative AI in Software Development: Challenges, Opportunities, and New Paradigms for Quality Assurance," Computer, Jul. 2025, doi: 10.1109/MC.2025.3556330.

[18] I. Udoidiok, H. Reza, and J. Zhang, "Exploring AI Integration in Software Development: Case Studies and Insights," in Proc. IEEE National Aerospace and Electronics Conf. (NAECON 2024), 2024, doi: 10.1109/NAECON61878.2024.10670347.

[19] C. Treude and M. A. Gerosa, "How Developers Interact with AI: A Taxonomy of Human-AI Collaboration in Software Engineering," in Proc. 2025 IEEE/ACM 2nd Int. Conf. AI Foundation Models and Software Engineering (FORGE), 2025, doi: 10.1109/Forge66646.2025.00033.

[20] S. Ayouni, R. A. Khan, M. Maddeh, H. S. Alwageed, I. Keshta, and A. O. Almagrabi, "Exploring the Synergistic Collaboration of Human Agentic-AI in Enhancing the Security of the Software Development Lifecycle," Egyptian Informatics Journal, Mar. 2026, doi: 10.1016/j.eij.2026.100915.

[21] S. Semenov, V. Tsukur, V. Molokanova, M. Muchacki, G. Litawa, M. Mozhaiev, and I. Petrovska, "Mathematical Model of the Software Development Process with Hybrid Management Elements," Applied Sciences, Oct. 2025, doi: 10.3390/app152111667.

[22] L. Cruz, X. Franch, and S. Martínez-Fernández, "Innovating for Tomorrow: The Convergence of Software Engineering and Green AI," ACM Transactions on Software Engineering and Methodology, Jun. 2025, doi: 10.1145/3712007.

[23] M. Haakman, L. Cruz, H.

Huijgens, and A. van Deursen, "AI Lifecycle Models Need to Be Revised: An Exploratory Study in Fintech," Empirical Software Engineering, Sep. 2021, doi: 10.1007/s10664-021-09993-1.

[24] A. M. Alberti, A. V. de A. Leal, A. G. Dalla-Costa, and C. B. Both, "AIPyCraft: AI-Assisted Software Development Lifecycle for 6G Blockchain Oracle Validation," in Proc. STAF Workshops 2025 (STAF-WS 2025), 2025.

[25] S. N. Hart, P. L. Day, and C. A. Garcia, "Streamlining Medical Software Development with CARE Lifecycle and CARE Agent: An AI-Driven Technology Readiness Level Assessment Tool," BMC Medical Informatics and Decision Making, Jul. 2025, doi: 10.1186/s12911-025-03099-0.

[26] T. Ramos, A. Dean, and D. Mcgregor, "AI-Augmented Software Engineering: Revolutionizing or Challenging Software Quality and Testing?" Journal of Software: Evolution and Process, Feb. 2025, doi: 10.1002/smr.2741.

[27] J. S. Bhalla and M. K. Jodhka, "Leveraging Large Language Models in the Software Development Lifecycle: Opportunities and Challenges," International Journal of Advanced Computer Science and Applications, 2025.

[28] D. Robinson, C. Cabrera, A. D. Gordon, N. D. Lawrence, and L. Mennen, "Requirements Are All You Need: The Final Frontier for End-User Software Engineering," ACM Transactions on Software Engineering and Methodology, Jun. 2025, doi: 10.1145/3708524.

[29] L. Freeman, J. Robert, and H. Wojton, "The Impact of Generative AI on Test & Evaluation: Challenges and Opportunities," in Companion Proc. 33rd ACM Int. Conf. Foundations of Software Engineering (FSE Companion 2025), 2025, doi: 10.1145/3696630.3728723.

[30] E. Yahav, "Towards AI-Driven Software Development: Challenges and Lessons from the Field," in Proc. 31st ACM Joint Meeting European Software Engineering Conf. and Symp. Foundations of Software Engineering (ESEC/FSE 2023), 2023, doi: 10.1145/3611643.3633451.

[31] K. Qiu, N. Puccinelli, M. Ciniselli, and L. Di Grazia, "From Today's Code to Tomorrow's Symphony: The AI Transformation of Developer's Routine by 2030," ACM Transactions on Software Engineering and Methodology, Jun.

2025, doi: 10.1145/3709353.

[32] V. Dhanush, T. S. Chandra, U. Divakarla, and K. Chandrasekaran, "Secure Intelligence Development Lifecycle (SIDL) Model for Vulnerability Detection," in Proc. Advanced Network Technologies and Intelligent Computing (ANTIC 2024), Springer, 2025, doi: 10.1007/978-3-031-83790-6_6.

[33] M. Slupczynski, R. Reiners, and S. Decker, "CAPTURE: A Stakeholder-Centered Iterative MLOps Lifecycle," Applied Sciences, Jan. 2026, doi: 10.3390/app16031264.

[34] E. Kocaguneli, A. Tosun, and A. Bener, "AI-Based Models for Software Effort Estimation," in Proc. 36th Euromicro Conf. Software Engineering and Advanced Applications (SEAA), 2010, doi: 10.1109/SEAA.2010.19.

[35] E. De La Cruz, H. Le, K. Meduri, G. S. Nadella, and H. Gonaygunta, "Redefining the Programmer: Human-AI Collaboration, LLMs, and Security in Modern Software Engineering," CMC — Computers, Materials & Continua, 2025, doi: 10.32604/cmc.2025.068137.

[36] I. A. Elgendy et al., "Responsible Vibe Coding: Architecture, Opportunities, and Research Agenda," Journal of Computer Information Systems, Jan. 2026, doi: 10.1080/08874417.2026.2621186.

[37] Q. Lu, L. Zhu, X. Xu, J. Whittle, D. Zowghi, and A. Jacquet, "Responsible AI Pattern Catalogue: A Collection of Best Practices for AI Governance and Engineering," ACM Computing Surveys, Jul. 2024, doi: 10.1145/3626234.

[38] P. Hidalgo, A. Domínguez, D. Rodriguez, and M. Mijac, "A Preliminary Study on the Use of Generative AI in Software Engineering Education," in Proc. Central European Conf. Information and Intelligent Systems (CECIIS 2025), 2025.

[39] A. Schmidt, "Speeding Up the Engineering of Interactive Systems with Generative AI," in Companion Proc. 2023 ACM SIGCHI Symp. Engineering Interactive Computing Systems (EICS 2023), 2023, doi: 10.1145/3596454.3597176.

[40] C. Molina, A. Navarro, P. Madriñán, and B. Muñoz, "AI Programmer Cooperation in Health Software Development," in Proc. 2025 IEEE Colombian Conf. Communications and Computing (COLCOM), 2025, doi: 10.1109/COLCOM66267.2025.

11185780.

[41] T. Viszkok and P. Hegedus, "Unified SAST Benchmark: Compare AI-Driven and Traditional Static Analyzers," SoftwareX, Sep. 2025, doi: 10.1016/j.softx.2025.102336

[42] H. A. Al-Hashimi, R. A. Khan, H. S. Alwageed, A. M. Algarni, S. Ayouni, and A. O. Almagrabi, "Exploring the Role of Generative AI in Enhancing Cybersecurity in Software Development Life Cycle," Array, Dec. 2025, doi: 10.1016/j.array.2025.100509.

[43] M. Baqar and R. Khanda, "The Future of Software Testing: AI-Powered Test Case Generation and Validation," in Intelligent Computing, Vol. 2, Springer, 2025, doi: 10.1007/978-3-031-92605-1_18.

[44] M. Esposito, S. Sarbazvatan, T. Tse, and G. Silva-Atencio, "The Use of Artificial Intelligence for Automatic Analysis and Reporting of Software Defects," Frontiers in Artificial Intelligence, Dec. 2024, doi: 10.3389/frai.2024.1443956.

[45] M. Steidl, M. Felderer, and R. Ramler, "The Pipeline for the Continuous Development of Artificial Intelligence Models Current State of Research and Practice," Journal of Systems and Software, May 2023, doi: 10.1016/j.jss.2023.111615.

[46] S. Westerstrand, "Fairness in AI Systems Development: Beyond EU AI Act Compliance," in Proc. Int. Conf. Software Business (ICSOB 2024), Springer, 2025, doi: 10.1007/978-3-031-85849-9_9.

[47] T. Abbas, S. A. Rathore, A. Turki, S. Khan, O. Alghushairy, and A. Daud, "Enhancing Software Engineering With AI: Innovations, Challenges, and Future Directions," IET Software, 2025, doi: 10.1049/sfw2/5691460.

[48] R. De Souza, F. Fronchetti, S. Freire, and R. Spinola, "Software Fairness Debt: Building a Research Agenda for Addressing Bias in AI Systems," ACM Transactions on Software Engineering and Methodology, Jun. 2025, doi: 10.1145/3709357.

[49] S. Westerstrand, "Fairness in AI Systems Development: EU AI Act Compliance and Beyond," Information and Software Technology, Nov. 2025, doi: 10.1016/j.infsof.2025.107864.

[50] A. Sergeyuk, Y. Golubev, T. Bryksin, and I. Ahmed, "Using AI-Based Coding Assistants in Practice:

State of Affairs, Perceptions, and Ways Forward," Information and Software Technology, Feb. 2025, doi: 10.1016/j.infsof.2024.107610.

[51] S. Adepoju and M. Adepoju, "Cognitive Biases: Understanding and Designing Fair AI Systems for Software Development," International Journal of Advanced Computer Science and Applications, 2025.

[60] N. Cheimarios, "Scientific Software Development in the AI Era: Reproducibility, MLOps, and Applications in Soft Matter Physics," Frontiers in Physics, Nov. 2025, doi: 10.3389/fphy.2025.1711356.

[61] S. Shah, P. Thakor, A. Shah, and O. V. Singh, "Software as Medical Devices: Requirements and Regulatory Landscape in the United States," Expert Review of Medical Devices, Nov. 2025, doi: 10.1080/17434440.2025.2561918.

[62] D. Amalfitano, S. Faralli, J. C. R. Hauck, S. Matalonga, and D. Distante, "Artificial Intelligence Applied to Software Testing: A Tertiary Study," ACM Computing Surveys, Mar. 2024, doi: 10.1145/3616372.

[63] S. Martínez-Fernández, X. Franch, A. Jedlitschka, M. Oriol, and A. Trendowicz, "Developing and Operating Artificial Intelligence Models in Trustworthy Autonomous Systems," in Proc. Research Challenges in Information Science (RCIS 2021), Springer, 2021, doi: 10.1007/978-3-030-75018-3_14.

[64] A. Nazmeeva, "Bug in My Software," in Proc. SIGGRAPH Asia 2024 Art Papers, 2024, doi: 10.1145/3680530.3695452.

[65] S. Rajapaksha, J. Senanayake, H. Kalutarage, and M. O. Al-Kadri, "Enhancing Security Assurance in Software Development: AI-Based Vulnerable Code Detection with Static Analysis," in Proc. Computer Security — ESORICS 2023 Int. Workshops (CPS4CIP, Pt. II), Springer, 2024, doi: 10.1007/978-3-031-54129-2_20.

[66] K. Sidaoui, D. Mahr, and G. Odekerken-Schröder, "Generative AI in Responsible Conversational Agent Integration: Guidelines for Service Managers," Organizational Dynamics, Apr.–Jun. 2024, doi: 10.1016/j.orgdyn.2024.101045.

[67] G. Bogdanoski, F. Lucas, W. Kern, and K. Czechowska, "Translating the Regulatory Landscape of Medical Devices to Create Fit-for-Purpose Artificial Intelligence (AI) Cytometry Solutions," Cytometry Part B: Clinical Cytometry, Jul. 2024, doi: 10.1002/cyto.b.22167.

[68] Y. Song, X. Xie, and B. Xu, "When Debugging Encounters Artificial Intelligence: State of the Art and Open Challenges," Science China Information Sciences, Apr. 2024, doi: 10.1007/s11432-022-3803-9.

[69] M. Simic, M. Jolovic, T. Naumovic, P. Lukovac, and M. Despotovic-Zrakic, "Integrating GenAI Tools Into Software Engineering Education," Computer Applications in Engineering Education, Mar. 2026, doi: 10.1002/cae.70177.

[70] W. C. Lee, C. Blanchette, S. Pokras, J. Shaikh, and J. Miller, "The Evolution and Future of Integrated Evidence Planning," Expert Review of Pharmacoeconomics & Outcomes Research, Jul. 2025, doi: 10.1080/14737167.2025.2497876.