East Asian Journal of Multidisciplinary Research (EAJMR)
Vol. 5, No. 3, 2026: 1039 - 1048

Next-Gen Developers: Rethinking Software Engineering Education in the Age of Generative AI

Von Gabayan[1]*, Bernadeth Liggayu[2], Carlita Segundo[3]
[1] Department of Information Technology, College of Information Technology Education, Nueva Vizcaya State University, Bayombong, Philippines
[2] Department of Computer Science, College of Information Technology Education, Nueva Vizcaya State University, Bayombong, Philippines
[3] Department of Information Systems, College of Information Technology Education, Nueva Vizcaya State University, Bayombong, Philippines
Corresponding Author: Von Gabayan vp_gabayanjr@nvsu.edu.ph

ARTICLEINFO

Keywords: Generative AI, Software Engineering, AI-assisted Programming

Received : 12, January
Revised : 14, February
Accepted: 27, March

ABSTRACT

The rapid rise of generative artificial intelligence (GenAI) is reshaping software engineering practices and redefining the skills required of modern developers. AI-powered tools now assist in code generation, debugging, documentation, and system design, affecting both industry workflows and educational approaches. This study conducts a systematic literature review of 18 scholarly publications using the PRISMA framework to explore the integration of GenAI in software development and software engineering education. Four key themes emerged: transformation of development practices, integration of AI in programming education, ethical and governance considerations, and workforce skill evolution. Findings highlight productivity gains and opportunities for outcome-based learning but reveal challenges in curriculum design, ethical AI use, and developing foundational programming and critical thinking skills, emphasizing the need for adaptive AI-integrated education.

DOI prefix: https://doi.org/10.55927/eajmr.v5i3.59
ISSN-E: 2828 – 1519 (Online)
https://journaleajmr.my.id/index.php/eajmr

INTRODUCTION

The emergence of generative artificial intelligence (GenAI) has introduced a transformative shift in the field of software engineering, influencing how software systems are designed, developed, and maintained. Recent advancements in AI-powered tools now allow developers to automatically generate source code, assist in debugging processes, produce technical documentation, and support system design decisions. These technologies are increasingly becoming part of modern software development environments and are reshaping traditional development practices. As a result, the competencies required of software developers are evolving beyond conventional programming skills toward a broader set of capabilities that include collaboration with intelligent systems and evaluation of AI-generated outputs. Scholars have noted that generative AI is gradually redefining software engineering workflows and development productivity, highlighting its growing significance in the software industry.

Despite these technological advancements, the integration of generative AI within higher education—particularly in software engineering and information technology programs—remains a complex challenge. Many academic institutions continue to rely on traditional approaches to programming education that emphasize manual coding practices and conventional software development methodologies. While these foundational skills remain essential, the emergence of AI-assisted development tools has altered the nature of tasks performed by software engineers. Developers are increasingly expected to work alongside AI systems, interpret machine-generated outputs, and ensure the quality, security, and reliability of generated code. Without adequate educational guidance, students may become overly dependent on automated systems or may lack the critical thinking skills necessary to assess AI-generated software artifacts.

Another important issue concerns the limited understanding of how generative AI should be effectively incorporated into software engineering education. Although an increasing number of studies have explored the use of generative AI in development processes, its pedagogical implications are still being examined. Educational institutions are currently faced with questions regarding how AI tools should be introduced in programming courses, how these tools can support outcome-based education, and how learners can be trained to use generative AI responsibly and effectively. Research also indicates that students demonstrate varying levels of trust, acceptance, and readiness in adopting generative AI technologies, which further complicates the integration of such tools in academic settings. These issues highlight the need for a more comprehensive understanding of existing studies that examine both the opportunities and challenges associated with generative AI in software engineering.

Addressing these concerns requires a critical reassessment of current pedagogical strategies in software engineering education. Rather than treating

generative AI as a replacement for traditional programming instruction, educators may consider leveraging these technologies as complementary tools that can enrich the learning experience. Integrating generative AI into programming education can foster creativity, enhance problem-solving skills, and expose students to modern development practices that are increasingly used in industry environments. Moreover, scholars emphasize the importance of guiding students in understanding the ethical, technical, and professional implications associated with AI-assisted software development, including issues related to AI safety, responsible use, and transparency.

In response to these emerging developments, this study proposes a systematic literature review to examine existing research on the role of generative AI in software development and its implications for software engineering education. By systematically analyzing scholarly publications related to generative AI, the study aims to identify key themes, emerging research directions, and current educational practices in the integration of AI-assisted development tools. The findings of this review are expected to contribute to a deeper understanding of how generative AI is transforming software engineering education and to provide insights that may guide the development of more adaptive, innovative, and future-oriented curricula for preparing the next generation of software developers in an AI-driven technological landscape.

LITERATURE REVIEW

The growing influence of generative artificial intelligence (GenAI) in software engineering has stimulated scholarly discussions on how development practices and educational models must evolve to prepare future software professionals. Researchers have explored both the technological transformation of software development workflows and the implications of these changes for computing education. This section presents two key themes that help frame the discussion of rethinking software engineering education in the age of generative AI.

Generative AI and the Transformation of Software Development Practices

Recent studies highlight how generative AI technologies are reshaping the processes involved in software development. According to Ebert and Louridas [5], generative AI tools can assist software practitioners in tasks such as automated code generation, software documentation, testing, and debugging. These capabilities allow developers to accelerate development cycles while improving productivity and collaboration within development teams. Similarly, Coutinho et al. conducted a pilot case study that demonstrated how generative AI can enhance software development productivity by reducing time spent on routine coding tasks and enabling developers to focus on higher-level design and problem-solving activities.

The integration of generative AI is also influencing project management and agile development practices. Bahi, Ghari, and Gahi explained that generative AI technologies can support agile software development by automating certain development tasks, facilitating rapid prototyping, and assisting project teams in managing complex development workflows. Furthermore, Rajbhoj et al. emphasized that AI-driven tools such as ChatGPT are increasingly being used by

developers to accelerate software development processes, assist in debugging, and provide code suggestions during programming tasks. These developments indicate that AI-assisted development environments are becoming an integral component of modern software engineering.

Beyond productivity improvements, generative AI is also influencing how developers collaborate and interact within development ecosystems. Ulfsnes, Moe, Stray, and Skarpen noted that the adoption of generative AI is changing team collaboration patterns by enabling developers to interact with AI agents during the development process. Additionally, Cui et al. found that generative AI tools can significantly affect high-skilled work among software developers by enhancing performance and altering traditional workflows. However, the increasing reliance on AI-generated outputs also raises concerns regarding reliability, ethical considerations, and responsible AI use, which must be addressed in both professional practice and academic training environments. These findings demonstrate that generative AI is not only transforming software development processes but also redefining the competencies required of modern software engineers.

Integration of Generative AI in Software Engineering Education

While generative AI is rapidly transforming software development practices, its integration into software engineering education is still an emerging area of research. Scholars have begun exploring how AI-driven tools can support teaching and learning in computing programs. Hernandez, Albina, and Caballero argued that generative AI can enhance programming education by supporting outcome-based teaching and learning approaches. Their study emphasized that AI-assisted programming tools can help students better understand coding concepts, generate programming solutions, and receive immediate feedback during the learning process.

Similarly, Petrovska, Clift, Moller, and Pearsall discussed the potential of incorporating generative AI into software development education to provide students with exposure to modern development tools used in industry. Integrating AI tools in classroom activities can help students develop practical skills in AI-assisted programming, enabling them to better adapt to evolving industry practices. However, the adoption of generative AI in academic settings also requires careful consideration of pedagogical strategies to ensure that students continue to develop fundamental programming knowledge and critical thinking skills.

In the context of higher education in the Philippines, several studies have explored the perceptions and acceptance of generative AI technologies among students and developers. Elinzano and Ching examined the acceptance and use of ChatGPT among software developers in the Philippines and found that factors such as perceived usefulness and ease of use significantly influence adoption. Similarly, Miranda et al. reported that tertiary students frequently use generative AI tools for academic tasks, although challenges related to trust, accuracy, and ethical use remain present. These findings suggest that while generative AI tools are increasingly being adopted by learners, institutions must establish clear

guidelines and structured learning frameworks to ensure responsible and effective use.

Another important dimension in the integration of generative AI in education involves governance, ethics, and responsible AI practices. Ligot emphasized the importance of developing layered frameworks to ensure safe and responsible AI deployment, while Piedad Jr. et al. proposed policy guidelines for regulating generative AI use in scholarly work within academic institutions. These initiatives highlight the growing recognition that educational institutions must balance innovation with responsible Technology governance.

METHODOLOGY

This study employed a quantitative systematic literature review using the PRISMA Framework to ensure a transparent and structured process for identifying, screening, and analyzing relevant scholarly studies related to generative artificial intelligence in software engineering and education. The review focused on 18 peer-reviewed scholarly publications obtained from major citation and academic databases including Scopus, Web of Science, Google Scholar, and IEEE Xplore. The literature search used key terms such as "generative AI in software development," "AI-assisted programming," "generative AI in software engineering education," and "AI-based programming tools." The PRISMA process involved four stages: identification, where relevant studies were retrieved from scholarly databases; screening, where duplicate and non-relevant studies were removed based on titles and abstracts; eligibility, where full-text articles were assessed according to inclusion criteria such as publication relevance, scholarly credibility, and focus on generative AI in software development or education; and inclusion, where 18 qualified studies were retained for final analysis. The selected literature was then quantitatively analyzed through thematic categorization and frequency analysis to identify prevailing research trends, contributions, and gaps in the integration of generative AI in software engineering practices and education. This methodological approach ensured a systematic and replicable review of current scholarly evidence supporting the study on rethinking software engineering education in the age of generative AI.

RESULTS AND DISCUSSION

Following the procedures of the PRISMA Framework, a systematic search of scholarly databases initially yielded a broad set of studies related to generative artificial intelligence in software development and education. During the identification stage, articles were retrieved from major academic indexing platforms such as Scopus, Web of Science, Google Scholar, and IEEE Xplore using keywords including generative AI, software engineering education, AI-assisted programming, and AI in software development. The initial search generated a large number of publications; however, after removing duplicate records and screening titles and abstracts, only studies directly related to generative AI in software engineering and education were retained.

During the screening stage, articles that did not focus on generative AI, were not peer-reviewed, or did not discuss implications for software development or education were excluded. The eligibility stage involved a detailed full-text review of the remaining articles to determine their relevance to the research objective. Studies were included if they discussed generative AI applications in software development, AI-assisted programming tools, software engineering productivity, AI integration in programming education, or governance and ethical issues related to AI use in academic environments. After applying the inclusion criteria, 18 scholarly publications were selected for the final review.

In the inclusion stage, the selected studies were analyzed using thematic categorization to identify major patterns and insights relevant to the transformation of software engineering education in the age of generative AI. The results show that existing literature emphasizes four major themes: (1) transformation of software development practices, (2) integration of generative AI in software engineering education, (3) ethical and governance challenges in AI-assisted development, and (4) evolving competencies and workforce transformation among next-generation developers. The findings demonstrate that generative AI is rapidly influencing both industry practices and academic learning environments, highlighting the need for educational institutions to adapt their curricula to meet emerging technological demands.

Ethical, Governance, and Responsible Use of Generative AI

Another important theme identified in the reviewed literature involves the ethical implications and governance of generative AI in both software engineering and academic environments. As AI-generated outputs become increasingly integrated into development workflows, concerns have emerged regarding transparency, accountability, intellectual property, and responsible use of AI systems. Ligot emphasized the importance of establishing structured frameworks to ensure the safe and ethical deployment of generative AI technologies. Such frameworks are necessary to guide developers and institutions in managing risks associated with algorithmic bias, misinformation, and unintended misuse of AI-generated content.

Similarly, Piedad Jr. et al. proposed policy guidelines for regulating the use of generative AI in academic and scholarly work within Philippine higher education institutions. Their work highlights the need for clear institutional policies to address issues such as academic integrity, authorship attribution, and responsible AI use. These concerns are particularly relevant in software engineering education, where students may rely heavily on AI-generated code during programming activities. Without proper regulation and ethical awareness, learners may fail to develop a deep understanding of programming concepts while simultaneously exposing themselves to potential academic misconduct risks. Therefore, integrating ethical AI literacy into computing education has become a crucial component of preparing responsible software developers in an AI-driven technological environment.

Workforce Transformation and Emerging Skills for Next-Generation Developers

Another emerging theme identified across the reviewed studies concerns the transformation of workforce skills required for future software developers. Generative AI tools are not merely changing development workflows but are also reshaping the competencies expected from professionals entering the software engineering industry. Sauvola et al. highlighted that AI-assisted development environments are shifting the role of developers from traditional coding tasks toward higher-level responsibilities such as system design, architecture planning, and critical evaluation of AI-generated outputs.

In addition, Shepherdson, Chew, and Purohit discussed how national initiatives related to AI prompt design and AI literacy are helping prepare future professionals to effectively collaborate with intelligent systems. These initiatives demonstrate that emerging roles in software development now include skills such as prompt engineering, AI-assisted problem solving, and evaluation of machine-generated outputs. Similarly, Cui et al. found that generative AI significantly influences productivity and performance among highly skilled developers, suggesting that AI collaboration is becoming an essential component of modern development practice. These findings indicate that educational institutions must reconsider traditional programming curricula and incorporate AI-related competencies to better prepare students for evolving industry demands.

Research Gaps

Despite the growing body of research on generative AI in software engineering and education, several important research gaps remain. First, many existing studies focus primarily on the technical capabilities of generative AI tools rather than their long-term impact on learning outcomes in software engineering education. Empirical research that evaluates how AI-assisted programming influences students' conceptual understanding, problem-solving abilities, and programming proficiency is still limited. Second, while several studies discuss the adoption and perception of generative AI among developers and students, there is still a lack of longitudinal studies examining how continuous exposure to AI-assisted development environments affects the development of programming expertise over time.

Another research gap concerns the integration of generative AI into structured software engineering curricula. Although researchers acknowledge the potential benefits of AI-assisted learning tools, few studies provide concrete instructional models or curriculum frameworks for incorporating generative AI into programming education. Additionally, research on ethical governance and responsible AI usage in computing education remains in its early stages, particularly within developing countries. These gaps highlight the need for further research exploring curriculum design, ethical guidelines, and pedagogical strategies that can effectively support AI-enhanced software engineering education.

CONCLUSIONS AND RECOMMENDATIONS

This systematic literature review examined 18 scholarly publications to explore the evolving role of generative artificial intelligence in software development and software engineering education. Using the PRISMA framework, the study identified major research themes that demonstrate how generative AI is transforming both professional development practices and educational environments. The findings reveal that generative AI technologies are increasingly being used to automate programming tasks, enhance software development productivity, and support collaborative workflows among developers. At the same time, the integration of AI tools into educational environments presents both opportunities and challenges for higher education institutions.

The review also highlights the growing importance of incorporating generative AI into software engineering curricula to prepare students for modern development environments. However, the findings indicate that the adoption of AI-assisted learning tools must be accompanied by ethical awareness, responsible AI practices, and critical evaluation skills. Educational institutions must therefore rethink traditional programming education by integrating AI-related competencies, promoting responsible AI use, and developing new teaching strategies that reflect industry practices. Ultimately, preparing next-generation developers requires a balanced approach that combines foundational programming knowledge with emerging skills in AI-assisted software development.

FURTHER STUDY

Based on the findings of this literature review, several potential research directions can be explored in future studies:

1. Development of a Generative AI-Integrated Software Engineering Curriculum Model for higher education institutions.
2. Empirical Evaluation of AI-Assisted Programming Tools in Improving Programming Skills among IT Students.
3. Design of an AI-Supported Programming Learning Environment Using Generative Code Assistants.
4. Ethical Framework for Responsible Use of Generative AI in Software Engineering Education.
5. Impact of Prompt Engineering Skills on Software Development Productivity among IT Students.
6. Comparative Study of Traditional Programming Instruction and AI-Assisted Programming Education.
7. Development of a Generative AI-Based Intelligent Programming Tutor for Computer Science Students.

These future studies can contribute to the development of more adaptive, innovative, and AI-integrated software engineering education models.

ACKNOWLEDGMENT

The researchers sincerely thank the scholars whose works informed this study, as well as colleagues, mentors, and friends for their guidance and support. Their contributions and encouragement greatly aided the completion of this

research and continue to inspire future studies in generative AI and software engineering education.

REFERENCES

Babaran Jr, C. L. Development of a Generative AI-Powered Application for Enhanced ICT Services Data Analysis and Interpretation.

Bahi, A., GHARI, J., & Gahi, Y. (2024). Integrating Generative AI for Advancing Agile Software Development and Mitigating Project Management Challenges. *International Journal of Advanced Computer Science & Applications*, *15*(3).

Coutinho, M., Marques, L., Santos, A., Dahia, M., França, C., & de Souza Santos, R. (2024, July). The role of generative ai in software development productivity: A pilot case study. In *Proceedings of the 1st ACM International Conference on AI-Powered Software* (pp. 131-138).

Cui, K. Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., & Salz, T. (2026). The effects of generative AI on high-skilled work: Evidence from three field experiments with software developers. *Management Science*.

Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. *IEEE software*, *40*(4), 30-38.

Elinzano, G. B. O., & Ching, M. R. (2024, November). ChatGPT Acceptance and Use Among Software Developers in the Philippines: A Structural Equation Modeling Approach Based on Extended UTAUT. In *International Conference on Multi-disciplinary Trends in Artificial Intelligence* (pp. 177-188). Singapore: Springer Nature Singapore.

Hernandez, A. A., Albina, E. M., & Caballero, A. R. (2025, June). Generative Artificial Intelligence for Programming Education: Enhancing Input to Outcome-Based Teaching and Learning Approaches. In *2025 Seventh International Symposium on Computer, Consumer and Control (IS3C)* (pp. 1-5). IEEE.

Ligot, D. V. (2024). Generative AI Safety: A Layered Framework for Ensuring Responsible AI Development and Deployment. *Available at SSRN 5008853*.

Majumdar, S., Paul, S., Paul, D., Bandyopadhyay, A., Chattopadhyay, S., Das, P. P., ... & Majumder, P. (2023). Generative ai for software metadata: Overview of the information retrieval in software engineering track at fire 2023. *arXiv preprint arXiv:2311.03374*.

Miranda, J. P., Bansil, J. A., Fernando, E., Gamboa, A., Hernandez, H., Cruz, M., ... & Penecilla, E. (2024, September). Prevalence, Devices Used, Reasons for Use, Trust, Barriers, and Challenges in Utilizing Generative AI among Tertiary Students. In *2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA)* (pp. 1-6). IEEE.

Petrovska, O., Clift, L., Moller, F., & Pearsall, R. (2024, January). Incorporating generative AI into software development education. In *Proceedings of the 8th Conference on Computing Education Practice* (pp. 37-40).

Piedad Jr, E., Tabud, S. L. C., Alina-Llabado, J. A., Danao, D. A., Gironella, M. C. A., Lim, J., ... & Yong, T. K. (2025). Regulating Generative AI in scholarly works: A policy brief for academic institutions in the Philippines.

Russo, D. (2024). Navigating the complexity of generative ai adoption in software engineering. *ACM Transactions on Software Engineering and Methodology*, *33*(5), 1-50.

Rajbhoj, A., Somase, A., Kulkarni, P., & Kulkarni, V. (2024, February). Accelerating software development using generative AI: ChatGPT case study. In *Proceedings of the 17th innovations in software engineering conference* (pp. 1-11).

Sauvola, J., Tarkoma, S., Klemettinen, M., Riekki, J., & Doermann, D. (2024). Future of software development with generative AI. *Automated Software Engineering*, *31*(1), 26.

Serdenia, J. R. C. (2025). Perceptions of Generative AI among Development Communication Students: Insights by Gender and Age from the Philippines. *EthAIca: Journal of Ethics, AI and Critical Analysis*, (4), 443.

Shepherdson, K., Chew, C., & Purohit, S. (2025). Practice-Led Case Study of Workforce Transformation through National AI Prompt Design Challenges in Singapore and the Philippines. *Journal of International DBA Studies-GGU*, *1*(002).

Ulfsnes, R., Moe, N. B., Stray, V., & Skarpen, M. (2024). Transforming software development with generative AI: Empirical insights on collaboration and workflow. In *Generative AI for effective software development* (pp. 219-234). Cham: Springer Nature Switzerland.