A preliminary Study on the Use of Generative AI in Software Engineering Education

Pablo Hidalgo, Adrián Domínguez, Daniel Rodriguez
Department of Computer Science
University of Alcala
Alcalá de Henares, 28805, Madrid, Spain
{adrian.dominguez, daniel.rodriguezg}@uah.es
pablo.hidalgo@ufv.es

Marko Mijač
University of Zagreb
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia
mmijac@foi.hr

***Abstract**. This study investigates the use of generative AI within a university-level software engineering (SE) course by analysing survey responses from second-year students. The survey explored multiple dimensions of students' interaction with AI while working on their course projects, including frequency of use, task types, prompting strategies, and perceived challenges. The results reveal widespread use of AI, especially for coding and debugging, though students also apply it across other phases of the software development lifecycle (SDLC). Despite this broad engagement, and even some use of advanced prompting, understanding of these techniques remains limited. All students reported verifying AI outputs, indicating low trust, which is further reinforced by the fact that many cited the inaccuracy of AI-generated results as their biggest challenge. Students also expressed a clear interest in improving their skills, particularly in prompt design. These findings underscore the need for structured support in AI literacy and prompting skills, as well as adapting course projects for an AI-enhanced learning context. This study provides a foundation for future research and instructional design in SE education.*

Keywords. Gen AI, software engineering, education

1 Introduction

Generative AI tools such as DeepSeek, Claude, ChatGPT, and GitHub Copilot are rapidly becoming part of the everyday toolkit for software developers, including students. In higher education, these tools offer opportunities to support coding, debugging, writing documentation, and other essential tasks in Software Engineering (SE) activities. The growing presence of these tools in educational contexts raises urgent questions about how students are engaging with them, what challenges students face, and what kinds of support students need use these tools effectively and responsibly.

As a result, the educational community has seen a surge of empirical studies investigating how generative AI is being used by students. This growing body of work is not only welcome but necessary. Given the diverse educational contexts, varying levels of AI literacy, and the rapid evolution of models, tools, and prompting strategies, no single study can offer generalizable conclusions. Continuous empirical research is needed to track how students adopt, adapt to, and are affected by these technologies, especially as institutional policies, pedagogical approaches, and the tools themselves evolve.

This paper contributes to the ongoing effort by examining how second-year undergraduate students used generative AI tools within a SE course. By analysing survey data on their usage patterns, prompting behaviours, challenges, and skill needs, we aim to provide timely insight into student practices and perceptions. These findings offer valuable implications for the design of instructional strategies, the development of AI literacy, and the adaptation of course projects to align with an AI-enhanced learning environment.

The rest of the paper is organized as follows: Section 2 provides background on generative AI in education and SE. Section 3 outlines the study's methodology, followed by survey results in Section 4. Section 5 presents a discussion of the findings, and Section 6 briefly covers the Threats to Validity. Finally, Section 7 concludes the paper with implications and directions for future research.

2 Background

In general, generative AI has transformed higher education by enabling personalized learning, automating repetitive tasks, and fostering creative problem-solving. Generative AI tools support diverse applications, from writing assistance to code generation. However, their adoption raises concerns about academic integrity, over-reliance, and equitable access.

For instance, Ali et al. (2024) conducted a systematic review of AI applications in education, identifying key challenges such as tool reliability and ethical concerns, which align with the struggles reported by students in our study. Similarly, Fu and Weng (2024) emphasized the importance of framing responsible, human-centered AI practices, reinforcing our call for ethical integration in curricula. Lepp and Kaimre (2025) explored student perceptions of generative AI in programming education, finding that while students appreciated AI's assistance, its impact on actual learning outcomes was mixed—supporting our observation that students may rely on AI without fully understanding its mechanisms. Jin et al. (2025) provided a global overview of institutional AI adoption policies, highlighting the need for clear guidelines and structured support, which our findings also suggest. Yusuf et al. (2025) proposed a conceptual framework for pedagogical AI agents, noting their potential to enhance engagement and learning outcomes. This is particularly relevant given our students' limited awareness of AI agents and their capabilities. Finally, Terragni et al. (2025) discussed the future of AI-driven SE, emphasizing the growing role of autonomous agents and the importance of preparing students for this shift—an area where our respondents showed interest but lacked exposure.

In SE, AI tools provide now a huge support helping with code generation, debugging, and testing. There are reports that GitHub Copilot, for example, vastly reduces coding time and improves task completion rates.[1]. On the other hand, there are also issues in the adoption of AI (Giannakos, 2024) as they can hinder students conceptual understanding by prioritizing task completion over a proper learning of the actual concepts. Furthermore, prompt engineering, i.e., the way precise inputs to optimize AI responses—is a critical skill for leveraging LLMs effectively, prompt patterns like few-shot prompting (providing examples to guide AI), chain-of-thought reasoning (breaking tasks into logical steps), and role-based prompting (assigning AI a specific role, e.g., "act as a senior developer") as essential for high-quality outputs. Not only that, but structured prompts can also improve code quality highlighting their importance in SE education.

Knot et al (2024) also states that students often rely on trial-and-error due to a lack of formal training, limiting their ability to tackle complex tasks like system design. Recent frameworks propose teaching prompt engineering as a core competency, similar to programming or debugging, see for example Lee and Palmer (2025). Hou et al (2024) review the use of Large Language Models in SE, highlighting their applications, optimization techniques, and challenges. They also include the number of studies using different prompt patterns.

AI agents, i.e., autonomous systems that perform multi-step tasks with minimal human intervention, are also transforming SE. Unlike LLMs, which respond to individual prompts, AI agents can autonomously write, test, and deploy code, integrating seamlessly with development environments. These agents support all SDLC phases by automating repetitive tasks. However, student awareness of AI agents is limited hindering students' ability to use agent in development projects.

Finally, there a need to for ethical integration of AI in education, including hallucination (AI generating incorrect outputs), bias in training data, and unequal access to tools. In SE, these are important, as biased AI outputs can introduce security vulnerabilities or perpetuate suboptimal design patterns. For example, an AI-generated sorting algorithm might prioritize speed over correctness if not properly verified, leading to errors in production systems. Nam and Bai (2024) emphasise the need for including ethics in the curricula to identify and mitigate AI biases, ensuring responsible use in technical disciplines.

3 Methodology

This study investigates the use of Generative AI in SE education through a survey-based approach combining quantitative and qualitative data collection. The research was guided by two questions:

- RQ1: How do students use Generative AI when performing SE tasks?
- RQ2: What challenges do students face when using Generative AI in their coursework?

To address these questions, we administered a structured survey to which 26 students enrolled in a second-year undergraduate SE course responded. The questions focused on students' use of Generative AI during their work on course-related software projects. While working on their projects, students were engaged in typical software development lifecycle (SDLC) activities, including requirements specification, software design, implementation (coding), and testing. Throughout the course, students were encouraged to explore and incorporate Generative AI tools in completing these activities and were provided with teaching materials that would help them do so. The questionnaire that was given to students comprised 10 questions, designed to capture both quantitative and qualitative data. Multiple-choice items were included to assess usage patterns and perceptions in a structured format, while open-ended

[1] https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-in-the-enterprise-with-accenture/

questions provided opportunities for participants to elaborate on their experiences and perspectives in greater depth.

The survey was administered in person during a scheduled class session that was compulsive at the end of the term. Therefore, the survey was answered by almost all students taking the course, however, the participation was voluntary, and all responses were anonymous. In the future, we will explore the possibility of linking surveys to academic scores, for further statistical analysis. Quantitative responses were analysed using descriptive statistics (e.g., frequency distributions and means) to identify patterns in students' use of AI. Qualitative responses were analysed using open coding to extract themes and patterns related to student experiences and expectations.

4. Results

In this section, we provide the summarised results of the survey.

When asked how often they use AI tools for SE tasks, students responded as follows (see *Fig. 1*): "Always" – 3 students, "Frequently" – 13 students, "Sometimes" – 9 students, 'Rarely' – 1 student, and 'Never' – 0 students.

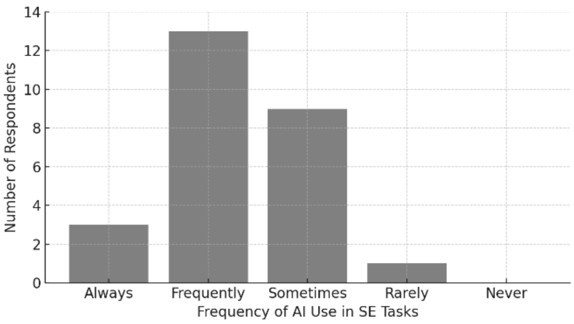

Figure 1. Frequency of AI use for SE tasks

Students were then asked to specify which SE tasks do they use AI for by choosing from multiple applicable options (see Fig. 2). The most frequently selected task was "Debugging," reported by 20 respondents. Both "Coding" and "Testing" were each selected by 16 respondents. "Requirements" was chosen by 13 respondents, while "Design" and "Documentation" were selected by 11 and 10 respondents, respectively.

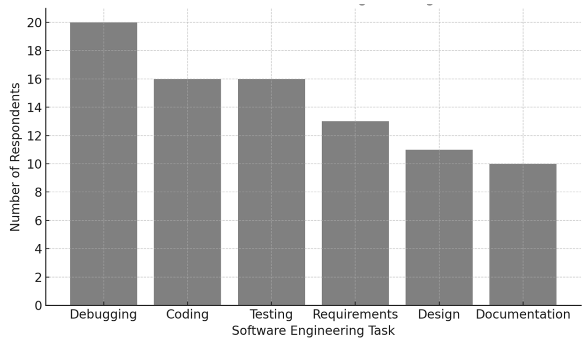

Figure 2. AI use across SE tasks

Regarding students' self-assessed competence with AI tools in the context of coding, a majority (19 respondents) considered themselves "Proficient," while 7 respondents rated their skills as "Basic."

In relation to debugging, students were asked how they typically ask AI for help, with multiple selections allowed (see Fig. 3). The most frequently reported approach was to "Paste errors," selected by 19 respondents. This was followed by "Ask for explanations" (15 respondents) and "Use step-by-step prompts" (13 respondents).

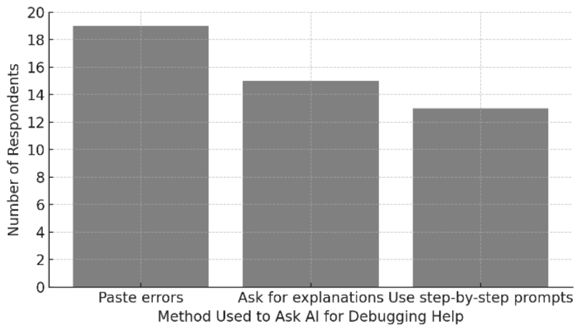

Figure 3. How students ask AI for debugging help

Students were asked whether they had experimented with advanced prompting techniques when interacting with AI tools. A total of 20 respondents indicated (see *Fig. 4*) that they had used at least one advanced strategy, while 6 stated that they relied solely on simple queries. Among those who had used advanced prompting, the most selected type was "Few-Shot" (13 respondents), followed by "Role-Based" (11), "Chain-of-Thought" (10), and "Constraints" (9).

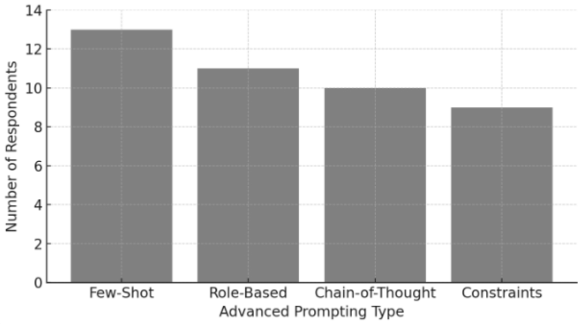

Figure 4. Use of advanced prompting techniques

In response to the question about verifying AI-generated outputs (see Fig. 5), 22 students indicated that they "always" check the results themselves. Four students reported that they do so "sometimes, for important things," while none selected the option "No, I trust it".

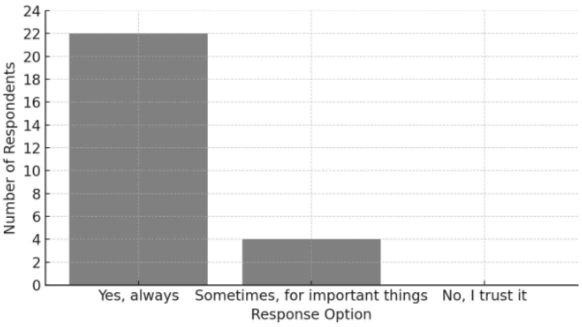

Figure 5. Verifying AI's results

When asked whether they understand what a prompt pattern is and how to use it, 14 students responded "Yes," while 10 indicated "No." It is worth noting that only 24 out of the 26 participants responded to this question.

The most commonly reported challenge students face when using AI is that it "makes mistakes," selected by 20 respondents. Other struggles were noted less frequently: 4 students pointed to ethical concerns, 3 admitted over-reliance on AI, and 2 cited other issues.

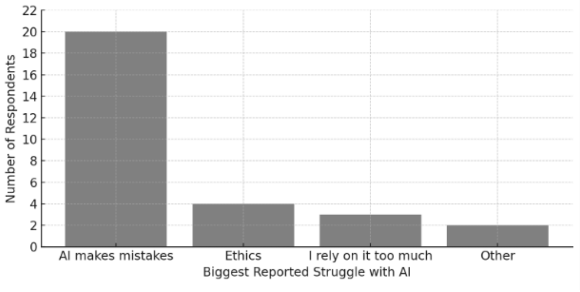

Figure 6. Students' struggles with AI

Out of 26 students, 22 provided a response to the open-ended question asking for an example where AI either helped or failed them. Eight students described positive experiences with AI assistance. Among these, 4 mentioned AI helped them with fixing errors, 2 with coding tasks, 1 with design and diagramming, and 1 with summarizing materials. In contrast, 14 students shared instances where AI failed to meet expectations. The most cited issue was incorrect code generation (7 respondents), followed by failures in error fixing (5 respondents) and diagram creation (2 respondents).

The survey results reveal several key outcomes regarding students' engagement with generative AI in SE tasks. First, AI use has become a regular part of students' workflow, with the majority reporting frequent or constant use which suggests a shift in how students approach software development, integrating AI tools as standard practice. Students used AI across a wide range of SE tasks, not just coding and debugging. However, the predominance of debugging and coding tasks also could reflect a tendency to rely on AI technical problem-solving rather than conceptual or design-oriented work.

The self-assessed competence levels show that most students feel confident in using AI for coding, which may reflect growing familiarity and confidence with these tools. Yet, it seems that fewer understand the underlying principles of prompting. This gap between practice and understanding highlights the need for formal training in prompt engineering for SE (this may not so necessary for general use, as LLMs are getting better at understanding user's use intend by the users). Finally, verification of AI outputs emerged as a challenge identified as the most common struggle. Open-ended responses showed that while some students benefited from AI assistance in coding and debugging, others encountered significant failures, especially in diagram creation. These mixed outcomes underscore the importance of teaching students how to critically evaluate and effectively prompt AI tools, in particular generating diagrams seems to be harder than generating code or test cases. Finally, students expressed a desire to improve their skills by applying AI to the full SE life cycle, indicating the necessity for educational support.

5 Discussion

In this section, we discuss our findings related to the two research questions stated previously in methodology section.

To begin with, we consider how students are currently using generative AI in the context of SE tasks (RQ1). The survey data indicate all students reported some level of AI use, with over 60% reporting that they use AI "frequently" or "always" for SE tasks. This suggests that AI has not only achieved a wide adoption among SE students but has become a regular part of their workflow rather than an occasional aid. Given this widespread adoption, SE educators may need to rethink the type, scope, and complexity of student projects to ensure they remain appropriately challenging and educational in an AI-augmented learning environment. Regarding the use of AI in different SE tasks, while code-centric tasks dominate, students do not stop there, they are also aware of AI's potential in earlier and later phases of development process. This awareness is important as the SDLC is both increasingly and deeply infused with AI at every stage.

Survey results imply that all students verify AI-generated results, with 85% of students report doing that "always". Even if this does not reflect actual practice, it at least implies a shared awareness of the necessity of result verification. This cautious approach is especially relevant in SE tasks, where precision and correctness are critical and unchecked errors can lead to significant issues and maintenance problems. The fact that verification is universally reported also indicates that students may have moved beyond the initial naivety and inflated expectations that often accompanied the early hype around generative AI.

Students expressed strong confidence in their AI skills when coding, with over 70% identifying as proficient. They reported a variety of techniques for debugging with AI—ranging from simply pasting error messages into a chat window to more structured, step-by-step interactions. When shown prompt examples, more than 75% indicated that they had used some form of advanced prompting, with Few-shot prompting, Chain-of-thought and Role-based prompt types being the most recognized ones. However, only 58% reported knowing what prompt patterns are, highlighting a clear gap between practice and understanding—and underscoring the need for more formal, structured education in this area.

Turning to the second research question, we examine the challenges students face when using generative AI in their coursework (RQ2). A clear majority of students identified AI's tendency to produce incorrect or misleading outputs as their biggest struggle. This concern was reported far more frequently than other issues such as ethical dilemmas or over-reliance on AI. The emphasis on AI-generated mistakes aligns with previously mentioned students' cautious behaviour, such as consistently verifying outputs. While technical reliability clearly emerged as the central challenge—far outweighing other concerns—some students did report issues such as ethical dilemmas or overreliance on AI tools. Although only a small number explicitly mentioned becoming too dependent on AI, this may serve as an early warning sign, particularly given that the respondents are second-year students still developing foundational SE skills. It is also possible that the actual number is higher, as some students may not yet recognize their overreliance or may be hesitant to admit it. Similarly, the low frequency of ethical concerns in the responses may point not to their irrelevance, but rather to a lack of awareness—highlighting the need for more structured education around responsible AI use in academic and professional contexts.

Open-ended responses provided interesting insights into AI's performance and students' satisfaction with it. While some students shared positive examples of AI assistance, particularly in error correction, coding, and summarization tasks, nearly twice as many described instances where AI failed to deliver accurate or usable results. Common failure points included incorrect code, ineffective debugging assistance, and flawed diagram generation. The higher number of failure reports may suggest that such scenarios are more frequent, or at least more memorable. This raises important questions about whether these failures stem from limitations in the tools themselves or from students using them ineffectively—again pointing to the need for formal education on how to prompt and evaluate AI effectively. It may be also possible that some of these frustrations reflect unrealistic expectations or low tolerance for imperfection. As second-year students who began their academic journey in the age of generative AI, they may take the technology for granted in ways that more mature students or professionals—who worked without such tools—do not. This generational shift in expectations may shape how students perceive both the value and the limitations of AI.

When asked what AI-related skills they would like to develop, students overwhelmingly expressed a desire to improve their prompting abilities. This aligns with earlier findings indicating widespread use of prompting techniques, but only partial understanding of underlying patterns. Some students also noted a desire to apply AI more holistically across the SE process, rather than limiting its use to isolated tasks like debugging or code generation. Others expressed interest in learning how to design software with AI support or even build AI systems themselves. These responses suggest that students are not only aware of their current skill gaps, but are also willing to learn beyond surface-level usage. The strong emphasis on prompting shows the need for instructional support in teaching students how to engage with AI tools more deliberately, strategically and effectively.

To translate these findings into actionable teaching strategies, educators should consider integrating structured AI literacy modules into SE curricula. These modules should cover not only the technical use of generative AI tools, but also prompt engineering, agents, ethical considerations, and critical evaluation of AI outputs. For example, SE educators need to design assignments that require students to use different prompting techniques and reflect on the outcomes, thereby deepening their understanding of prompt patterns and their impact on AI outcomes. Moreover, educators need to adapt project-based learning assignments to include AI-supported workflows. This could involve tasks where students must use AI tools for requirements gathering, design, coding, and testing, in CD/CI (Continuous Development/Continuous Development) followed by a critical review of the AI-generated output. Such activities would help students develop quality software with the help of AI across the whole SDLC.

6 Threats to Validity

This study presents several threats to the validity (TTV) that may affect the interpretation and generalizability of its findings. These are categorized below according to common validity frameworks in empirical SE research (Wohlin et al, 2024).

External Validity. The small sample size—26 students from a single second-year software engineering course—limits generalizability. Student behaviours may differ across institutions, academic levels, or course designs.

Internal Validity. Survey timing at the end of the course may have influenced responses due to recent experiences or exam stress. Uncontrolled factors like prior AI exposure or peer influence could also affect reported behaviours.

Construct Validity. The custom-designed, non-validated questionnaire may not reliably measure key constructs such as AI proficiency or ethical awareness. Additionally, self-reported data introduces bias, as students may misjudge their skills or respond in socially desirable ways.

Despite these TTV, this preliminary study offers some initial insights into student interactions with generative AI. These findings serve as a foundation for further research that incorporates validated instruments, broader survey samples (students and professionals), and mixed-research approaches to deepen our understanding of AI integration in SE engineering education.

7 Conclusions and future works

This preliminary study explored how second-year SE students use generative AI tools in their coursework and what challenges they face. The findings indicate that AI has already become a routine part of students' development practices, especially for tasks like debugging and coding. While students report confidence in their skills and actively experiment with advanced prompting strategies, gaps in understanding remain—a point also acknowledged by students themselves, who expressed a desire to learn more about prompting and AI use. Technical reliability remains the primary challenge to students, but ethical awareness, responsible use, and the risk of overreliance also appear to require further attention.

The results point to a clear need for structured AI literacy within SE education—especially in the areas of prompting, critical evaluation of outputs, ethical use, and awareness of potential overreliance. In parallel, student tasks and project designs may also need to evolve to remain challenging and meaningful in an environment where AI support is widespread.

As a preliminary study, this work lays the groundwork for more extensive and validated research involving larger and more diverse student cohorts and professionals. Future work will focus on designing and evaluating instructional modules that integrate generative AI into SE education. Additionally, future surveys could incorporate student performance on tasks involving AI tools as a proxy for understanding prompting patterns, thereby reducing reliance on self-reported data.

Acknowledgements

Daniel Rodriguez acknowledges "ACT-e: AI for enhanced classroom teaching" within the EUGLOH Consortium.

References

Ali, O., Murray, P.A., Momin, M., Dwivedi, Y.K. & Malik, T. (2024) The effects of artificial intelligence applications in educational settings: Challenges and strategies, *Technological Forecasting and Social Change*, 199, p. 123076. https://doi.org/10.1016/j.techfore.2023.123076

Braun, V. & Clarke, V. (2006) Using thematic analysis in psychology, *Qualitative Research in Psychology*, 3(2), pp. 77-101.

Cigdem, S., Rumyana, N. & Giuseppe, D. (2024) Software engineering education in the era of conversational AI: current trends and future directions, *Frontiers in Artificial Intelligence*, 7. https://doi.org/10.3389/frai.2024.1436350

Fu, Y. & Weng, Z. (2024) Navigating the ethical terrain of AI in education: A systematic review on framing responsible human-centered AI practices, *Computers and Education: Artificial Intelligence*, 7, p. 100306. https://doi.org/10.1016/j.caeai.2024.100306

Giannakos, M., Azevedo, R., Brusilovsky, P., Cukurova, M., Dimitriadis, Y., Hernandez-Leo, D. & Rienties, B. (2024) The promise and challenges of generative AI in education, *Behaviour & Information Technology*, pp. 1-27. https://doi.org/10.1080/0144929X.2024.2394886

Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J. & Wang, H. (2024) Large Language Models for Software Engineering: A Systematic Literature Review, *ACM Transactions on Software Engineering and*

Methodology, 33(8), Article 220, pp. 1-79. https://doi.org/10.1145/3695988

Jin, Y., Yan, L., Echeverria, V., Gašević, D. & Martinez-Maldonado, R. (2025) Generative AI in higher education: A global perspective of institutional adoption policies and guidelines, Computers and Education: Artificial Intelligence, 8, p. 100348. https://doi.org/10.1016/j.caeai.2024.100348

Knoth, N., Tolzin, A., Janson, A., & Leimeister, J. M. (2024). AI literacy and its implications for prompt engineering strategies. *Computers and Education: Artificial Intelligence, 6*, 100225. https://doi.org/10.1016/j.caeai.2024.100225

Lee, D. & Palmer, E. (2025) Prompt engineering in higher education: a systematic review to help inform curricula, *International Journal of Educational Technology in Higher Education*, 22, p. 7. https://doi.org/10.1186/s41239-025-00503-7

Lepp, M. & Kaimre, J. (2025) Does generative AI help in learning programming: Students perceptions, reported use and relation to performance, Computers in Human Behavior Reports, 18, p. 100642. https://doi.org/10.1016/j.chbr.2025.100642

Nam, K., & Bai, Y. (2024). Unpacking the role of AI ethics online education for science and engineering students. *International Journal of STEM Education*, 11, Article 35. https://doi.org/10.1186/s40594-024-00493-4

Terragni, V., Vella, A., Roop, P. & Blincoe, K. (2025) The Future of AI-Driven Software Engineering, *ACM Transactions on Software Engineering and Methodology*, 34(5), Article 120, pp. 1-20. https://doi.org/10.1145/3715003

Wang, S., Wang, F., Zhu, Z., Wang, J., Tran, T. & Du, Z. (2024) Artificial intelligence in education: A systematic literature review, *Expert Systems with Applications*, 252(Part A), p. 124167. https://doi.org/10.1016/j.eswa.2024.124167

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2024). Experimentation in Software Engineering (2nd ed.). Springer Berlin/Heidelberg, https://doi.org/10.1007/978-3-662-69306-3

Yusuf, H., Money, A. & Daylamani-Zad, D. (2025) Pedagogical AI conversational agents in higher education: a conceptual framework and survey of the state of the art, *Education Tech Research Dev*, 73, pp. 815-874. https://doi.org/10.1007/s11423-025-10447-4

Questionnaire

Section 1: How Do You Use AI?

1. How often do you use AI for coding/SE tasks? (Check one box)
   - Never (I don't use AI)
   - Rarely (only a few times)
   - Sometimes (for some tasks)
   - Frequently (almost every project)
   - Always (AI is part of my workflow)
2. Which SE tasks do you use AI for? (Check all that apply)
   - Requirements (e.g., "Generate user stories for a fitness app")
   - Design (e.g., "Create a UML diagram for a banking system")
   - Coding (e.g., "Write Python code for a REST API")
   - Debugging (e.g., "Explain why this Java error occurs")
   - Testing (e.g., "Generate unit tests for this function")
   - Documentation (e.g., "Summarize this code for a README file")
   - Other: ______

Section 2: How Skilled Are You with AI?

3. For CODING, which best describes you? (Check one)
   - Basic: I ask simple things (e.g., "How do I write a for-loop in Java?")
   - Proficient: I use smart prompts (e.g., "Optimize this SQL query for speed" or "Explain this algorithm, then rewrite it in C++.")
4. For DEBUGGING, how do you ask AI for help? (Check all that apply)
   - Paste errors (e.g., "Fix this error: NullPointerException")
   - Ask for explanations (e.g., "Why does this Python code give a TypeError?")
   - Use step-by-step prompts (e.g., "Analyze this stack trace and suggest fixes")
   - Other: ______
5. Have you tried ADVANCED PROMPTING? (Check all that apply)
   - Few-Shot (e.g., "Here's my code. Suggest improvements like these examples.")
   - Chain-of-Thought (e.g., "First explain the bug, then suggest fixes.")
   - Role-Based (e.g., "Act as a senior dev and review my code.")
   - Constraints (e.g., "Rewrite this function with O(1) space complexity.")
   - None (I only use simple questions)

Section 3: Challenges & Feedback

6. Do you CHECK AI's work? (Check one)
   - No, I trust it
   - Sometimes, for important things
   - Yes, always! (I test code/docs myself)

7. What's your BIGGEST struggle with AI? (Check one)
   - o AI makes mistakes
   - o I rely on it too much
   - o Ethics (e.g., plagiarism, cheating?)
   - o Other: ______
8. Give ONE example where AI helped (or failed) you:

(Example: "AI wrote a SQL query for me, but it had a syntax error.")
Your Answer:
____

9. What AI skills do you WANT TO LEARN?

(Example: "Better debugging prompts" or "How to design with AI")
Your Answer:
____

10. In general, do you think you know what a prompt pattern is and how to use them?
    - o Yes, I know them and use them
    - o No, I don't know about prompt patterns