Examining the Utilization of Artificial Intelligence Tools by Students in Software Engineering Projects

Amir Dirin[1][a] and Teemu H. Laine[2,*][b]
[1]*Metropolia University of Applied Sciences, Helsinki, Finland*
[2]*Department of Digital Media, Ajou University, Suwon, Republic of Korea*

Keywords: Software Engineering, Higher Education, Project, Artificial Intelligence, GitHub Copilot, ChatGPT, Agile Method.

Abstract: With the popularity of AI-based tools, the landscape of learning and teaching software engineering has shifted to a new era, which has left both educators and students confused regarding the extent to which these tools are reliable, secure, and, most importantly, result in efficient student competence development. In this study, we explored how the use of AI tools such as ChatGPT and GitHub Copilot affect the performance of 36 students in nine teams in a software engineering project course. We also explore the perceptions of the students regarding the use of AI tools in software engineering. We divided the project teams into three groups based on their use of AI tools: group 1 used AI tools freely, group 2 used AI tools in a restricted manner, and group 3 did not use any AI tools. The results indicated that while all groups successfully finished their projects, AI tools were of great help in user story creation and completing a high number of features and tasks. However, groups 1 and 2 also require time to learn the AI tools and the resulting software quality was lower than that of group 3. In conclusion, AI tools like Copilot and ChatGPT can become powerful companions to software engineering students in their educational activities.

1 INTRODUCTION

The field of computer science is constantly evolving with the advancement of new technologies. Even for professionals, it can be a challenge to keep up with the pace of all the developments. The emergence of ChatGPT has sparked discussions about its potential and the opportunities it may bring about (Taecharungroj, 2023). It has fostered a positive attitude toward both learning and teaching (Ali et al., 2023; Rospigliosi, 2023). However, over time its capabilities and performance have been evaluated by researchers. For example, Thorp (2023) explored the writing capabilities of ChatGPT and found that it produces amusing text. In another study, Gilson et al. (2023) demonstrated that ChatGPT provides logical and informational responses for a majority of questions. Barenkamp et al. (2020) showed that AI can be effectively applied across various phases of software engineering to accelerate development processes cost-effectively. Georgievski (2023) proposed an AI-based software development lifecycle for AI-based system architecture, aiming to replace traditional software engineering methodologies. Neumann et al. (2023) suggested practical guidelines for effectively utilizing ChatGPT in educational settings. Even though ChatGPT has been in existence for almost two years, uncertainty persists regarding where artificial intelligence (AI) development will have the most significant impact. Engineering fields have been grappling with a significant concern: whether teaching programming is still necessary, given that AI tools like ChatGPT and GitHub Copilot can write programs and devise diverse solutions to given problems. Even though software engineering encompasses more than just coding, the process involves multiple phases to ultimately deliver a finalized product.

AI tools have a significant impact on software engineering and other engineering fields,

[a] https://orcid.org/0000-0002-4851-5711
[b] https://orcid.org/0000-0001-5966-992X
[*]Corresponding author: teemu@ubilife.net

Dirin, A. and Laine, T.
Examining the Utilization of Artificial Intelligence Tools by Students in Software Engineering Projects.
DOI: 10.5220/0012729400003693

In *Proceedings of the 16th International Conference on Computer Supported Education (CSEDU 2024) - Volume 2*, pages 286-293
ISBN: 978-989-758-697-2; ISSN: 2184-5026

encompassing various aspects such as project management, requirements engineering (Dalpiaz & Niu, 2020), concept design (Verganti et al., 2020), and implementation. While there are challenges involved in different phases of software engineering, it is plausible to address them using AI tools. These tools have the potential to assist in project risk analysis, budget calculation, resource allocation, cost prediction, and change anticipation, among other tasks. Many commercial AI-based tools support the management of software projects, such as Asana, Notion, and Monday.

Undoubtedly, the impact of software engineering on both industrialization and societal adaptation to digitalization has been significant (Borg et al., 2018). With the emergence of AI, this development has now entered a new phase. The implementation of weak AI solutions, such as chatbots, has already demonstrated the potential for transformational change in support teams and teaching (Chiu et al., 2023). More recently, the introduction of strong AI solutions, such as ChatGPT, heralds a new era of development that promises to take both industries and society to unprecedented levels (F.-Y. Wang et al., 2023).

Despite the remarkable advancements made in AI, there remain several unresolved questions that require further research and development, particularly regarding the cognitive (Guerrero et al., 2023) and ethical aspects of AI (Maciel, 2023), such as ownership of AI (Kim & Song, 2023), decision-making and trust (Kaplan et al., 2023), data security (Dirin, et al. 2023) and data integrity (Khan et al., 2023). While these issues are directly related to the software development aspect of AI, this research plan will focus primarily on the software engineering discipline perspective rather than the philosophical, security, or user experience perspectives. One of the primary rationales behind prioritizing software processes and solutions over AI is that while AI is seen as an enabler, it is the software engineering process that plays a pivotal role in tackling the challenges facing both industries and society.

This paper aims to evaluate the impacts of AI tools, specifically ChatGPT and GitHub Copilot, on outcomes, student motivation, and perceived potential in software engineering student projects.

2 RELATED RESEARCH

2.1 Artificial Intelligence

The field of AI was established by John McCarthy and other colleagues at the Dartmouth Conference in 1956 (Rajaraman, 2014). They defined AI as the simulation of human-like intelligence, including learning and related features, by machines. The related features are reasoning, problem-solving, perception, and understanding. Therefore, machines equipped with capabilities such as visual perception (Esteva et al., 2021), speech recognition (Jung et al., 2020), decision-making (Leyer & Schneider, 2021), and language transactions (Mieczkowski et al., 2021) are considered to be under the umbrella of AI.

2.2 AI-Based Tools in Education

Education institutes and educational strategists continuously pursue to improve the effectiveness of teaching and learning by embracing novel technologies and methodologies, such as AI-based applications and tools. Among these activities, the personalization of learning (Dirin & Laine, 2018; Tan et al., 2023) has received significant attention. Intelligent tutoring systems have been designed and developed to provide customized instructions, tutoring, and feedback to students to improve their learning experience. Recently, efforts have been made to facilitate learning and teaching with AI. For example, AI has been employed for automated grading of short answers (Süzen et al., 2020). Further attempts to personalize student learning through the adaptation of the learning environments have been ongoing for some time (How & Hung, 2019; Walkington, 2013). Moreover, AI has enabled the expansion of learning and teaching beyond educational environments, as seen in language learning platforms (Rebolledo Font De La Vall & González Araya, 2023). Furthermore, educational content creation has been improved through the advancement of generative AI tools, natural language processing, and machine learning algorithms (Du et al., 2023).

2.3 AI Tools in Software Engineering Education

AI has already made an impact in almost every domain of contemporary life, and the field of software engineering is no exception. Daun and Brings (2023) have recommended that it is essential to adapt teaching software engineering with the latest AI development, specifically by providing guidelines to students on the extent to which they need to utilize AI. ChatGPT has already been utilized for various purposes in software engineering courses, such as in the system analysis course (Albonico & Varela, 2023) where ChatGPT is used to answer student inquiries.

Furthermore, as articulated by Ozkaya (2023), AI is being applied to various tasks in software engineering, including code generation as demonstrated by tools like Copilot by GitHub. Puryear and Sprint (2022) demonstrated that Copilot generates unique code for introductory assignments with code accuracy scores ranging from 68% to 95% which means that the code Copilot generates is largely aligned with human expectations.

3 RESEARCH QUESTIONS AND METHODS

3.1 Research Questions

We pursue to answer the following research questions in this study:

1. How does using AI tools affect the implemented functions in software engineering student projects?
2. How does using AI tools affect the completed tasks per sprint in software engineering student projects?
3. How do software engineering students perceive using AI tools in school projects?

3.2 Research Method

A mixed-method approach was utilized, including the use of a questionnaire, a brief discussion with the students during project presentations, and an analysis of the project implementation and performance. The mixed-method approach was chosen to acquire diverse data from different viewpoints, which would provide a holistic understanding of the implications of using or not using AI tools in student projects. The brief discussion aimed to understand the students' perceptions of the use of AI in software development. This information complements the questionnaire data that gathers the students' insights about how AI helped or did not help them in their project implementation. This discussion was more like an exchange of ideas than a semi-structured interview. Therefore, the mixed-method approach enabled us to assess the project implementation from various perspectives, including the developer perspective, project outcomes, and project management. The students were required to assess their contributions to the project at each sprint, as well as reflect on how the project impacted their competency development.

A total of 36 students (27 males, 9 females, age range: 20-35) of a software engineering course were divided into nine teams of four students. Students were in their fourth semester and had completed full-stack development courses at the university. The students already knew each other and formed the teams based on their preferences. The only requirement enforced was the size of the team, which was four people.

We assigned the teams into three groups. In the first group (AITU), the teams were allowed to use ChatGPT and Copilot in development. The second group (PAIU) was allowed to use AI tools but they first had to receive approval from the first author (instructor). Lastly, the third group (NAIA) was tasked with developing their software solely based on their knowledge and skills without using any AI tools. During the sprint review meetings, each group shared their implementation status and applied tools with the first author. Additionally, in the sprint planning meeting, the first author, along with the group, planned the tasks and technology used for the upcoming sprint. This approach enabled the first author to closely monitor both the technology utilized and the progress of implementation.

All teams received the same high-level requirements for software to be developed over eight weeks consisting of four two-week sprints. The teams selected project topics, after which the first author presented the three AI usage groups (AITU, PAIU, and NAIA) to the teams. The teams then selected the AI usage group they wished to belong to.

The teams were required to adhere to the Scrum methodology, where the first author was the product owner, and use project management tools (e.g. Trello). Additionally, they learned about dependencies and continuous integration and development. As a result, the difficulties of the projects were primarily dependent on the implementation technologies they freely chose. All projects mandated students to apply three-tier architectural solutions.

The students in a team took turns acting as the scrum master, overseeing the project's progress. At the end of each sprint, the first author conducted a sprint review with the teams. Before the software project implementation, the product owner delineated the technical, functional, and non-functional requirements of the projects.

While the selection of implementation technology was free, the teams opted to utilize technologies with which they were most familiar. However, they were required to demonstrate proficiency in unit testing and incorporate various dependencies through Maven into their codebase. Code was expected to be systematically cleaned and refactored to ensure maintainability. Automation tools like Jenkins were employed for continuous integration. Additionally, GitHub was used for version control, while Trello was utilized for progress tracking. During the first week of Sprint 1, the focus was on conducting a feasibility study, architectural planning, and database design, utilizing UML and Entity Relationship diagrams. Additionally, Figma was employed for conceptual design.

In the assessment of the results, our main focus was on the number of task implementations, code cleanliness, and the applied refactoring approach, including the number of features implemented. However, we did not evaluate the technical performance of the resulting solutions or the robustness of the algorithms in this study.

4 RESULTS

Table 1 displays the project topics selected by the three groups.

Table 1: Topics the groups selected for their project.

Project Name	Teams	Requirement
Instant Course Feedback	R2, R6, R9 (AITU)	• A web application enabling students to submit anonymous feedback after a lecture. • It must include reporting, admin tool, and barcode generation functionality.
Health and Fitness Tracker	R1, R4, R7 (PAIU)	• A mobile application for tracking sports activities. • The application records daily sports activities, saves them, and allows users to make queries.
Language Learning App	R3, R5, R8 (NAIA)	• A game-based web application for Finnish language learning. • It enables peers to compete, tracks their progress, and generates reports on the results.

Our analysis of the project presentation and documentation outcomes indicate that NAIA tended to exert extensive effort to ensure a high-quality product. For example, they conducted unit tests for all functions, applied Jenkins, and more easily implemented localization. Our analysis of the resulting software revealed significant differences in the number of implemented features. The AITU and PAIU teams implemented almost all or nearly all of the planned features during the development sprints, but the NAIA teams postponed more than 30% of the features to the next sprints. However, all teams applied the test template and performed the required tasks.

At the end of eight weeks, all groups produced a functional product. However, it was clear that the groups' applications varied in performance, the number of tasks in the product backlog implemented in each sprint, and the features included. Based on the project's nature, each group distributed differently the tasks to the sprints. Sprint one focused on the feasibility study and technology setup for all groups. However, for those who used AI (AITU and PAIU), most of the effort in sprint 1 was dedicated to learning the AI tools. Students at AITU expressed a strong belief in the significant impact of AI on their project success. Only 25% of them believed to some extent that AI aided in accomplishing tasks. However, 13% of students held a belief that AI had no impact on the successful implementation of tasks. Moreover, 88% of the students at AITU indicated that there is still a deficiency in AI-based development tools for devising robust solutions. Conversely, 12% of the students in the AITU suggested that there are already sufficient AI-based tools to support complex software engineering projects.

Interestingly, both AITU and PAIU consulted ChatGPT in Sprint 1 to develop user stories after defining tasks in the product backlog.

The answers by PAIU were interesting as 70 % of the students in this group believed that AI to some extent helped them complete more tasks. However, 18% of them responded that AI had no impact at all on the implementation of more tasks in their project. Almost 6% of those group members who replied believed that AI helped to implement more tasks.

In NAIA, almost 19% believed that AI has no impact on implementing more tasks in software projects. Only 81% believed that AI may impact to implementation of more tasks to some extent.

Figure 1 presents the proportions of completed tasks in each sprint by the AITU teams. The proportions of completed tasks in each sprint are almost homogeneous. Despite using AI tools such as Copilot and ChatGPT, the teams in AITU completed a significant proportion of the tasks and user stories in sprint 1 (23%), as they learned how to use the AI tools. Furthermore, the teams in this group completed a similar proportion of tasks in sprint 3, which covered database development and merging different modules to work. The teams in this group postponed 18% of the tasks for the next product iterations.

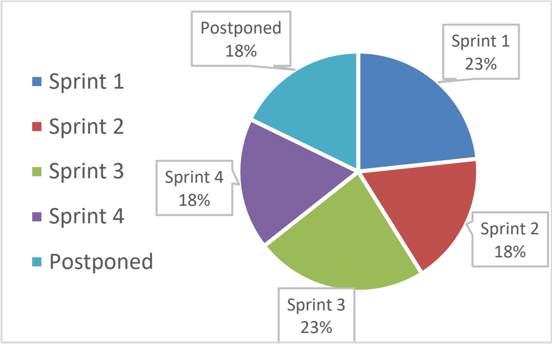

Figure 1: The task completion proportions for each sprint of AITU (free AI use).

Figure 2 depicts the outcomes of each sprint for the teams in the PAIU group. The proportions of completed tasks across the sprints indicate that the majority of task implementations occurred during sprints 2 and 3, a notable deviation from AITU. In sprint 1, tasks amounting to 18% were allocated to design and planning. Furthermore, only 6% of the tasks were deferred to subsequent iterations of product implementation.

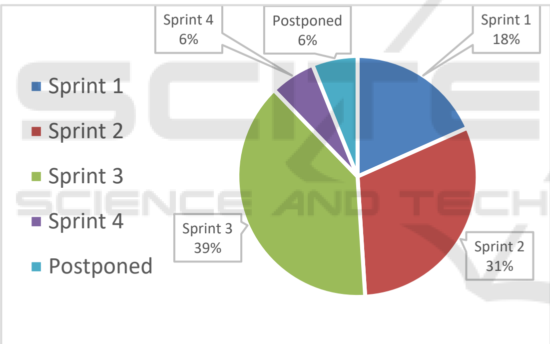

Figure 2: The task completion proportions for each sprint of PAIU (limited AI use).

Figure 3 displays the task completion proportions in each sprint for NAIA. A majority of the tasks (47%) were completed in sprint 4, signalling a delay in the schedule, as the goal was to achieve an almost functional product by the conclusion of sprint 3. Furthermore, they completed more tasks in designing, defining the product visions, and articulating the user stories in Sprint 1 compared to the other two groups. During Sprint 1, they dedicated time to technology selection and design of the database. Unlike the other two groups, the product owner in the scrum meetings that this group had to allocate time and effort to investigate technologies and dependencies. NAIA had to allocate time to writing code, conducting unit tests, and implementing dependencies, resulting in the completion of only 3% of tasks in Sprint 2. Furthermore, NAIA had the highest proportion of tasks (23%) that were postponed to subsequent iterations of product development.

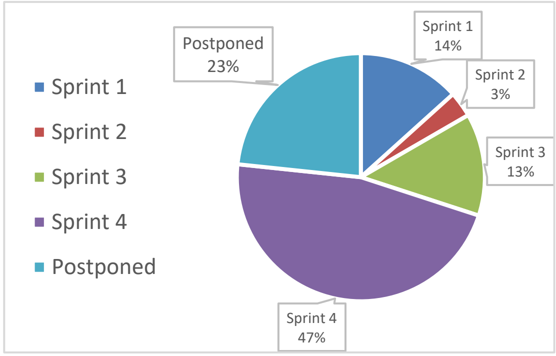

Figure 3: The task completion proportions for each sprint of NAIA (no AI use).

5 DISCUSSION

In this paper, we aimed to explore the effects of the utilization of AI tools by higher education students in software engineering projects. Specifically, we sought to evaluate the quality of the resulting software, the number of features implemented, and the impact of AI on task implementation and user stories in Agile methodology. We also explored the students' perceptions of using AI tools in software engineering.

5.1 Answering to Research Questions

By answering the three research questions, we aimed to uncover the impacts of AI tools on students' software projects and their insights on the use of those tools. AI methods and technologies help in general engineering assignments. Already in 1986, Goldberg (1986) demonstrated that AI can construct efficient software. More recently, Salehi (2018) showed that AI can assist in solving complex engineering problems. This is aligned with our study results as students in groups AITU and PAIU successfully created software with the help of AI tools. Our results demonstrate that AI facilitated the implementation of the tasks and features that were initially defined at the product backlog level. What we identified is that the resulting AI-facilitated software quality was not at the same level as that of a traditional software development approach. This fact was also highlighted by Cernau et al. (2022) (who recommended a pedagogical approach that checks the quality of the source code developed by students using AI techniques.

AI has also been applied to software requirements engineering for analysing requirements to create high-quality software as indicated by Dapliaz and Niu (2020). In our experiment, AI-facilitated teams implemented more tasks than those who did not use AI tools. On the other hand, Wang and Xu (2021), recommended that autonomous software requirement specifications need to be used for AI programming.

In general, AI-based tools like GitHub Copilot and ChatGPT can become powerful companions to software engineering students in their educational activities. Besides getting help for generating code, debugging, or generating test cases, ChatGPT was applied by the participants to come up with tasks and user stories for their product backlog in Agile development. Barke et al. (2022) categorized the uses of Copilot in two ways by students: (i) acceleration to complete the tasks, and (ii) exploring alternative options they have for the solution. Pearce et al. (2022) showed that the code generated by GitHub Copilot has almost 40 percent higher risk of having cybersecurity issues. Moreover, Ziegler et al. (2022) showed that those who apply for Copilot focus more on the immediate action rather than the long-term impact of the code. This was also visible in our study: those who applied AI focused on the completion of the tasks that had been planned for each sprint rather than paying attention to the quality of the code. Therefore, the AI-facilitated teams implemented more tasks than those who wrote the code by hand which is aligned with the findings of Imai (2022).

Furthermore, the use of ChatGPT to articulate user stories for product backlog is yet another way the students leveraged AI in their projects. From a product owner perspective, it was clear that AI-facilitated teams generated more innovative user stories compared to those who did not utilize AI tools. The impact of AI and non-AI software product development, as determined through comparisons of various development team setups, has yet to be investigated. This aspect warrants further analysis and exploration in future research.

5.2 Validity of the Results

The results are valid within the context of our study. They are supported by the setup of our research, which incorporates the competence level of the students (fourth semester) and the capabilities and features of ChatGPT and Copilot utilized during the study. However, for broader applicability, further investigation and evaluation employing a more systematic approach would be beneficial. Quality assurance and technical performance measurements could have enhanced the study, but due to time constraints, they were not implemented. Nevertheless, we maintain that our results hold validity within the scope of our study, as we employed multiple approaches to assess task performance, application features, and developer discussions.

5.3 Implications and Future Work

This study revealed that students view AI tools like ChatGPT and Copilot as enhanced learning support mediums. As educators, we emphasize the importance of quality assurance and stress the need for thorough consideration of the software's quality resulting from the use of AI tools. The development of software applications with the help of AI tools requires a redefinition of software quality assurance, standards, and methods. We aim to extend the study to gather more systematic data, with a particular focus on the quality assurance of resultant applications.

ACKNOWLEDGEMENTS

The contribution of the second author was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2023S1A5C2A02095195).

REFERENCES

Albonico, M., & Varela, P. J. (2023). A Report on the Use of ChatGPT in Software Engineering and Systems Analysis Courses. Proceedings of the XXXVII Brazilian Symposium on Software Engineering, 303–311. https://doi.org/10.1145/3613372.3614189

Ali, J. K. M., Shamsan, M. A. A., Hezam, T. A., & Mohammed, A. A. Q. (2023). Impact of ChatGPT on Learning Motivation: Teachers and Students' Voices. Journal of English Studies in Arabia Felix, 2(1), 41–49. https://doi.org/10.56540/jesaf.v2i1.51

Barenkamp, M., Rebstadt, J., & Thomas, O. (2020). Applications of AI in classical software engineering. AI Perspectives, 2(1), 1. https://doi.org/10.1186/s42467-020-00005-4

Barke, S., James, M. B., & Polikarpova, N. (2022). Grounded Copilot: How Programmers Interact with Code-Generating Models. https://doi.org/10.48550/ARXIV.2206.15000

Borg, M., Olsson, T., Franke, U., & Assar, S. (2018). Digitalization of swedish government agencies: A perspective through the lens of a software development census. Proceedings of the 40th International Conference on Software Engineering: Software

Engineering in Society, 37–46. https://doi.org/10.1145/3183428.3183434

Cernau, L. D., Dioşan, L. S., & Şerban, C. (2022). A pedagogical approach in interleaving software quality concerns at an artificial intelligence course. Proceedings of the 4th International Workshop on Education through Advanced Software Engineering and Artificial Intelligence, 18–24. https://doi.org/10.1145/3548660.3561332

Chiu, T. K. F., Moorhouse, B. L., Chai, C. S., & Ismailov, M. (2023). Teacher support and student motivation to learn with Artificial Intelligence (AI) based chatbot. Interactive Learning Environments, 1–17. https://doi.org/10.1080/10494820.2023.2172044

Dalpiaz, F., & Niu, N. (2020). Requirements Engineering in the Days of Artificial Intelligence. IEEE Software, 37(4), 7–10. https://doi.org/10.1109/MS.2020.2986047

Daun, M., & Brings, J. (2023). How ChatGPT Will Change Software Engineering Education. Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1, 110–116. https://doi.org/10.1145/3587102.3588815

Dirin, A., & Laine, T. H. (2018). Towards an Adaptive Study Management Platform: Freedom Through Personalization: Proceedings of the 10th International Conference on Computer Supported Education, 432–439. https://doi.org/10.5220/0006788104a0439

Dirin, A., Oliver, I., & Laine, T. H. (2023). A Security Framework for Increasing Data and Device Integrity in Internet of Things Systems. Sensors, 23(17), 7532. https://doi.org/10.3390/s23177532

Dixon-Román, E., Nichols, T. P., & Nyame-Mensah, A. (2020). The racializing forces of/in AI educational technologies. Learning, Media and Technology, 45(3), 236–250. https://doi.org/10.1080/17439884.2020.1667825

Du, H., Li, Z., Niyato, D., Kang, J., Xiong, Z., Xuemin, Shen, & Kim, D. I. (2023). Enabling AI-Generated Content (AIGC) Services in Wireless Edge Networks. https://doi.org/10.48550/ARXIV.2301.03220

Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., Liu, Y., Topol, E., Dean, J., & Socher, R. (2021). Deep learning-enabled medical computer vision. Npj Digital Medicine, 4(1), 5. https://doi.org/10.1038/s41746-020-00376-2

Georgievski, I. (2023). Conceptualising Software Development Lifecycle for Engineering AI Planning Systems. 2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN), 88–89. https://doi.org/10.1109/CAIN58948.2023.00019

Gilson, A., Safranek, C. W., Huang, T., Socrates, V., Chi, L., Taylor, R. A., & Chartash, D. (2023). How Does ChatGPT Perform on the United States Medical Licensing Examination? The Implications of Large Language Models for Medical Education and Knowledge Assessment. JMIR Medical Education, 9, e45312. https://doi.org/10.2196/45312

Goldberg, A. T. (1986). Knowledge-based programming: A survey of program design and construction techniques. IEEE Transactions on Software Engineering, SE-12(7), 752–768. https://doi.org/10.1109/TSE.1986.6312977

Guerrero, L. E., Castillo, L. F., Arango-López, J., & Moreira, F. (2023). A systematic review of integrated information theory: A perspective from artificial intelligence and the cognitive sciences. Neural Computing and Applications. https://doi.org/10.1007/s00521-023-08328-z

How, M.-L., & Hung, W. L. D. (2019). Educational Stakeholders' Independent Evaluation of an Artificial Intelligence-Enabled Adaptive Learning System Using Bayesian Network Predictive Simulations. Education Sciences, 9(2), 110. https://doi.org/10.3390/educsci9020110

Imai, S. (2022). Is GitHub copilot a substitute for human pair-programming?: An empirical study. Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings, 319–321. https://doi.org/10.1145/3510454.3522684

Jung, Y. H., Hong, S. K., Wang, H. S., Han, J. H., Pham, T. X., Park, H., Kim, J., Kang, S., Yoo, C. D., & Lee, K. J. (2020). Flexible Piezoelectric Acoustic Sensors and Machine Learning for Speech Processing. Advanced Materials, 32(35), 1904020. https://doi.org/10.1002/adma.201904020

Kaplan, A. D., Kessler, T. T., Brill, J. C., & Hancock, P. A. (2023). Trust in Artificial Intelligence: Meta-Analytic Findings. Human Factors: The Journal of the Human Factors and Ergonomics Society, 65(2), 337–359. https://doi.org/10.1177/00187208211013988

Khan, A. A., Laghari, A. A., Li, P., Dootio, M. A., & Karim, S. (2023). The collaborative role of blockchain, artificial intelligence, and industrial internet of things in digitalization of small and medium-size enterprises. Scientific Reports, 13(1), 1656. https://doi.org/10.1038/s41598-023-28707-9

Kim, T., & Song, H. (2023). Communicating the Limitations of AI: The Effect of Message Framing and Ownership on Trust in Artificial Intelligence. International Journal of Human–Computer Interaction, 39(4), 790–800. https://doi.org/10.1080/10447318.2022.2049134

Leyer, M., & Schneider, S. (2021). Decision augmentation and automation with artificial intelligence: Threat or opportunity for managers? Business Horizons, 64(5), 711–724. https://doi.org/10.1016/j.bushor.2021.02.026

Maciel, L. (2023). Editorial: ChatGPT and the ethical aspects of artificial intelligence. Revista de Gestão, 30(2), 110–112. https://doi.org/10.1108/REGE-04-2023-207

Mieczkowski, H., Hancock, J. T., Naaman, M., Jung, M., & Hohenstein, J. (2021). AI-Mediated Communication: Language Use and Interpersonal Effects in a Referential Communication Task. Proceedings of the ACM on Human-Computer Interaction, 5(CSCW1), 1–14. https://doi.org/10.1145/3449091

Neumann, M., Rauschenberger, M., & Schön, E.-M. (2023). "We Need To Talk About ChatGPT": The Future of AI and Higher Education [Application/pdf]. 163 KB. https://doi.org/10.25968/OPUS-2467

Ozkaya, I. (2023). Application of Large Language Models to Software Engineering Tasks: Opportunities, Risks, and Implications. IEEE Software, 40(3), 4–8. https://doi.org/10.1109/MS.2023.3248401

Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2022). Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions. 2022 IEEE Symposium on Security and Privacy (SP), 754–768. https://doi.org/10.1109/SP46214.2022.9833571

Puryear, Ben; Sprint, Gina. (2022). Github copilot in the classroom: Learning to code with AI assistance. Journal of Computing Sciences in College, 38(1), 27–47.

Rajaraman, V. (2014). John McCarthy—Father of artificial intelligence. Resonance, 19(3), 198–207. https://doi.org/10.1007/s12045-014-0027-9

Rebolledo Font De La Vall, R., & González Araya, F. (2023). Exploring the Benefits and Challenges of AI-Language Learning Tools. International Journal of Social Sciences and Humanities Invention, 10(01), 7569–7576. https://doi.org/10.18535/ijsshi/v10i01.02

Rospigliosi, P. 'Asher.' (2023). Artificial intelligence in teaching and learning: What questions should we ask of ChatGPT? Interactive Learning Environments, 31(1), 1–3. https://doi.org/10.1080/10494820.2023.2180191

Salehi, H., & Burgueño, R. (2018). Emerging artificial intelligence methods in structural engineering. Engineering Structures, 171, 170–189. https://doi.org/10.1016/j.engstruct.2018.05.084

Süzen, N., Gorban, A. N., Levesley, J., & Mirkes, E. M. (2020). Automatic short answer grading and feedback using text mining methods. Procedia Computer Science, 169, 726–743. https://doi.org/10.1016/j.procs.2020.02.171

Taecharungroj, V. (2023). "What Can ChatGPT Do?" Analyzing Early Reactions to the Innovative AI Chatbot on Twitter. Big Data and Cognitive Computing, 7(1), 35. https://doi.org/10.3390/bdcc7010035

Tan, A. Z., Yu, H., Cui, L., & Yang, Q. (2023). Towards Personalized Federated Learning. IEEE Transactions on Neural Networks and Learning Systems, 34(12), 9587–9603. https://doi.org/10.1109/TNNLS.2022.3160699

Thorp, H. H. (2023). ChatGPT is fun, but not an author. Science, 379(6630), 313–313. https://doi.org/10.1126/science.adg7879

Verganti, R., Vendraminelli, L., & Iansiti, M. (2020). Innovation and Design in the Age of Artificial Intelligence. Journal of Product Innovation Management, 37(3), 212–227. https://doi.org/10.1111/jpim.12523

Walkington, C. A. (2013). Using adaptive learning technologies to personalize instruction to student interests: The impact of relevant contexts on performance and learning outcomes. Journal of Educational Psychology, 105(4), 932–945. https://doi.org/10.1037/a0031882

Wang, F.-Y., Yang, J., Wang, X., Li, J., & Han, Q.-L. (2023). Chat with ChatGPT on Industry 5.0: Learning and Decision-Making for Intelligent Industries. IEEE/CAA Journal of Automatica Sinica, 10(4), 831–834. https://doi.org/10.1109/JAS.2023.123552

Wang, Y., & Xu, J. Y. (2021). Autonomous Software Requirement Specifications towards AI Programming. 2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), 123–130. https://doi.org/10.1109/ICCICC53683.2021.9811311

Ziegler, A., Kalliamvakou, E., Li, X. A., Rice, A., Rifkin, D., Simister, S., Sittampalam, G., & Aftandilian, E. (2022). Productivity assessment of neural code completion. Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming, 21–29. https://doi.org/10.1145/3520312.3534864