

INTEGRATING AI TOOLS IN UX AND SOFTWARE EDUCATION: INSIGHTS FROM TEACHING PRACTICE

Robert Pucher

Department of Computer Science, University of Applied Sciences Technikum Wien (Austria)

Abstract

Artificial intelligence (AI) is increasingly influencing software engineering education. This paper reports experiences from integrating an AI-supported learning environment based on Google NotebookLM into a usability testing course in the master's program Software Engineering at the University of Applied Sciences Technikum Wien. Students used curated materials and structured question sets during self-study and applied usability and accessibility testing methods in practical exercises. Assessment was conducted through face-to-face examinations. Results show a high level of methodological correctness in the student work. Of 47 assessed students, 78.7% received the highest grade and the pass rate was 97.9%. Course evaluations were also very positive. The experience suggests that AI-supported learning environments can effectively support structured preparation and learning outcomes when combined with clear assignments and controlled assessment formats.

Keywords: *AI in software engineering education, usability testing, AI-supported learning.*

1. Introduction

Artificial intelligence (AI) is increasingly influencing software development and the way software engineering is taught in higher education. Tools based on large language models can support learning activities such as information retrieval, structured analysis, and the preparation of written material (Alanazi, et al., 2025; Shi et al., 2025; Manorat, Rattanapoka, & Srisuphab, 2025; Khan, Akbar, & Kasurinen, 2025). This development raises an important question for educators: how can AI tools be integrated into teaching while maintaining methodological rigor and reliable assessment?

This question is particularly relevant in courses that teach usability testing, where students must learn to design and interpret evaluations of interactive systems with users. These activities require the formulation of research questions, the preparation of structured test scenarios, and the analysis of observations—tasks that many students initially find difficult to structure.

The course “Usability Testing” in the master's program Software Engineering at the University of Applied Sciences Technikum Wien therefore integrates an AI-supported learning environment based on Google NotebookLM. Students interact with curated course materials and predefined guiding questions that structure their exploration of the content.

While AI tools support the learning phase, assessment is conducted through a face-to-face examination to ensure independent understanding of usability testing methods. This paper reports insights from this teaching approach and examines how AI-supported learning influences student engagement and comprehension in usability testing education.

2. Educational context

The course “Usability Testing” is part of the master's program Software Engineering (MSE) at the University of Applied Sciences Technikum Wien, Austria. The four-semester program combines advanced software engineering with human-centered design and interaction design competencies. Within this curriculum, usability and human factors are treated as essential elements of modern software development.

The course belongs to the Human Factors module and introduces students to the principles and methods of usability evaluation. Students learn how to plan, conduct, and analyze usability tests, including test scenarios, task design, observation techniques, and the interpretation of qualitative and quantitative findings.

Embedded in a technically oriented software engineering curriculum, the course connects usability evaluation with broader development practices. Students apply usability testing methods to digital systems and prototypes, learning how empirical user feedback informs design decisions and iterative development.

3. Teaching approach and examination method

The course “Usability Testing” follows a blended learning format combining structured self-study, practice-oriented classroom sessions, and competence-based assessment. The teaching design follows a flipped classroom approach in which students first acquire theoretical knowledge independently and then apply it in practical classroom activities.

A central element of the self-study phase is Google NotebookLM as an AI-supported learning environment. The environment is based on curated course sources, including the textbook *User Experience and Usability: Basics and Methods* (Seibert-Giller & Pucher, 2023), UX and accessibility training syllabi, and structured question sets used for exam preparation. Students work with approximately 100 guiding questions per session, supported by concise answer outlines, which help structure their preparation and deepen their understanding of usability and accessibility topics.

The learning materials also include precise task descriptions for both the usability test and the accessibility test conducted during the course. As a result, students enter the classroom sessions with a clear understanding of the methods and objectives of the exercises.

During the in-person sessions, students work in small groups to conduct usability tests on websites or applications. They take on different roles, such as moderator, participant, or observer, and systematically document usability issues. Additional activities address accessibility evaluation and the integration of usability findings into agile development processes such as Scrum.

Assessment combines practical work and individual knowledge verification. Students present the usability or accessibility tests they conducted and explain their methodological approach and findings. The theoretical questions are assessed exclusively in face-to-face examinations without AI support to ensure independent understanding. This approach ensures that students are able to apply usability and accessibility methods in a structured and professionally relevant way within modern software development environments.

4. Results

Students applied the methods studied during the AI-supported self-study phase in practical usability and accessibility evaluations. Working in small groups, they designed and conducted usability tests on selected websites or applications. Test tasks were defined, sessions were moderated using the think-aloud method, and usability issues were systematically documented. The results were summarized in short test reports and presented in class together with recommendations for interface improvements.

In addition, each group performed an accessibility evaluation of a website. Students applied WCAG-based criteria, used automated accessibility testing tools, and conducted manual checks such as keyboard navigation and screen reader testing. Identified accessibility barriers were documented and prioritized according to severity.

All students closely followed the clearly defined methodological guidelines and task descriptions provided in the course materials. As a result, the submitted reports and presentations showed a consistently structured approach and a high level of methodological correctness.

The grading results reflect this outcome. Of the 47 assessed students, 37 (78.7%) received the grade “Excellent”, 9 (19.1%) received “Good”, and one student (2.1%) received a failing grade. The average course grade was 1.43, with a median of 1 (Excellent) and an overall pass rate of 97.9%. These results indicate that the students were able to apply usability and accessibility evaluation methods successfully and according to the defined methodological framework.

5. Discussion

The results of the course are strikingly positive and require careful interpretation. With nearly eighty percent of the students receiving the highest grade and an overall pass rate close to one hundred percent, the outcomes may initially appear unusually strong. One possible explanation lies in the highly structured learning environment. The combination of curated learning materials, clearly defined tasks, and extensive guiding questions created a transparent framework that enabled students to prepare systematically for both the practical assignments and the face-to-face examination.

Another important factor is the AI-supported learning environment. The use of Google NotebookLM allowed students to interact with curated course sources and structured question sets in a

flexible way. Instead of replacing learning, the AI environment functioned as a scaffold that helped students organize information, clarify concepts, and rehearse explanations before the in-person sessions. This appears to be particularly effective in software engineering education, where students are accustomed to working with technical tools and structured problem-solving approaches.

The professional background of the students also plays a role. All participants in the master's program work at least part-time in industry. As a result, they are familiar with the practical challenges of software development and recognize the relevance of usability and accessibility testing in real projects. This professional experience likely contributed to the high level of engagement observed during the practical exercises and presentations.

Student feedback from the official course evaluation was also very positive. Overall satisfaction received an average rating of 1.00 on the Austrian five-point scale (1 = very positive). Students rated the course structure, clarity of explanations, and practical exercises with an average of 1.00. Learning objectives and examination modalities were rated 1.20, and alignment with other modules 1.40. The perceived workload was moderate (2.40). Qualitative feedback highlighted that the accessibility testing exercise helped students recognize the gap between regulatory requirements and the accessibility of many real-world websites.

At the same time, the strong results highlight an important methodological aspect: although AI tools supported the learning process, the assessment itself was conducted in a controlled face-to-face setting without AI assistance. This design ensured that the high grades reflect actual understanding rather than AI-generated answers.

Overall, the experience suggests that AI-supported learning environments can be particularly effective in software engineering education when combined with structured materials, clear assignments, and controlled assessment formats. Rather than replacing traditional teaching methods, AI tools can enhance preparation and comprehension while maintaining academic rigor.

6. Conclusion

The results of the course indicate that students were able to successfully apply usability and accessibility testing methods in a structured and methodologically sound manner. The combination of AI-supported self-study, clearly defined assignments, and practice-oriented classroom activities enabled students to prepare effectively for the practical exercises and examinations.

The grading results were very strong, with most students receiving the highest grade and an overall pass rate of 97.9%. Course evaluations were also highly positive, with students rating the structure of the course, the clarity of explanations, and the practical exercises very highly.

Overall, the results suggest that the combination of AI-supported learning environments, structured teaching materials, and face-to-face assessment can support effective learning outcomes in usability testing education within software engineering programs.

References

- Alanazi, M., Soh, B., Samra, H., & Li, A. (2025). The influence of artificial intelligence tools on learning outcomes in computer programming: A systematic review and meta-analysis. *Computers, 14*(5).
- Khan, M., Akbar, M. A., & Kasurinen, J. (2025). Integrating large language models in software engineering education. *arXiv preprint arXiv:2509.04877*
- Manorat, P., Rattanapoka, S., & Srisuphab, K. (2025). Artificial intelligence in computer programming education: A systematic literature review. *Computers and Education: Artificial Intelligence, 6*.
- Seibert-Giller, V. & Pucher, R. (2023). *User Experience and Usability: Basics and Methods*. Vienna, Austria: UXQCC Publishing.
- Shi, Y., Wang, H., Zhao, L., & Chen, J. (2025). Large language models in education: A systematic review. *Computers and Education: Artificial Intelligence, 6*.