World Journal of Educational Research
ISSN 2375-9771 (Print) ISSN 2333-5998 (Online)
Vol. 12, No. 6, 2025
www.scholink.org/ojs/index.php/wjer

Original Paper

Exploration and Practice of Teaching Reform of AI Empowerment

Xiaoyan Hou[1a,*], Nan Chen[1b], QinChuan Lei[1c] & Pengteng Huang[1d]

[1] Department of Computing, Guangdong University of Science and Technology, Dongguan, China

[a] 944360614@qq.com, [b] leiqinchuan@gdust.edu.cn, [c] chennan@gdust.edu.cn, [d] 2388161806@ qq.com

[*] Corresponding author

Received: August 22, 2025 Accepted: October 24, 2025 Online Published: November 13, 2025

doi:10.22158/wjer.v12n6p34 URL: http://dx.doi.org/10.22158/wjer.v12n6p34

Abstract

With the rapid development of artificial intelligence technology such as large language model, the professional development of software engineering is undergoing profound changes. There are some problems in the teaching of traditional software engineering course, such as boring theory and few practical links. This study uses AI tools to empower teaching, runs through real project cases in teaching, and reconstructs teaching content according to students' learning characteristics. By deeply integrating AI tools into the software life cycle of software projects, such as requirements analysis, system design, coding, testing and document writing, the curriculum team builds a mixed teaching mode of "theory leading, tool empowerment and project driving". The teaching practice of one semester shows that the reform can effectively stimulate students' interest in learning, improve their engineering practice ability, man-machine cooperation ability and critical thinking, and significantly improve the project completion and code quality, which provides a reference path for training software engineering talents to meet the needs of the intelligent era.

Keywords

Software Engineering, Teaching Reform, Artificial Intelligence, Project-Driven Learning

1. Introduction

The rise of generative artificial intelligence (AIGC), especially the popularity of large-scale language models (LLMs), has had a great impact on software engineering (Wang & Feng, 2023). AI tools have penetrated into all aspects of software development, from project requirements refinement to code generation, from test case design to document writing, all of which can be assisted by AI tools (Chen, Chen, & Lin, 2020). This requires that the talent training of software engineering major in our school

must keep pace with the times, integrate AI tools into the course teaching, and enhance students' employment competitiveness (Wang & Li, 2024).

Software Engineering, as a required course for software engineering majors, adopts the textbook Software Engineering Course compiled by Wu Di, which focuses on the knowledge of software life cycle of waterfall model and theoretical knowledge of UML modeling. Students can be familiar with concepts but lack specific project application practice, and the connection between theory and practice is not close enough, which leads to insufficient learning motivation and poor teaching effect. This contradiction is particularly prominent among students of this major. Students in this major are usually active in thinking and have a strong interest in cutting-edge technology and application practice, but there are some problems such as weak theoretical foundation, insufficient learning patience and easy frustration in pure theoretical study.

Based on this, this study carried out a teaching reform practice, aiming at exploring a new path to deeply integrate AI tools into curriculum teaching. In this teaching reform, the domestic excellent big model DeepSeek and bean bag tools are selected as the core auxiliary tools, and students are more skilled in using them. The reform takes the classical software engineering theory as the main teaching content, improves the teaching efficiency with AI, works in groups, and is driven by a complete software project, so as to make students learn by doing and learn from using, and finally realize students' deep understanding of software life cycle knowledge and comprehensive improvement of modern engineering ability (Zhang & Liu, 2021).

2. The Status Quo and problem analysis

2.1 The Traditional "Software Engineering" Teaching Pain Points

1) Theory is seriously out of touch with practice: students systematically learn a large number of data flow diagrams, flow diagrams, use case diagrams, sequence diagrams and activity diagrams in class, but their application ability in real projects is weak, their practical ability is poor, and the connection between theory and practice is weak.

2) Document writing is not standardized: the process of writing requirement specifications and designing documents is boring, the documents written by students are simple and lack of standardization, and the systematic analysis and design of the project are insufficient.

3) Weak coding ability: Some students can't turn good system design into operational code because of their weak programming ability, resulting in some theoretical knowledge not being applied to practical projects.

4) Low team cooperation efficiency: In the process of team development, the project progress is often slow due to poor communication and unreasonable task assignment, and the cooperation and management emphasized by software engineering cannot be effectively experienced.

2.2 Analysis of Learning Situation

This course is carried out in the next semester of junior year of software engineering major. Students generally have strong hands-on ability, respond positively to learning methods that can directly bring a sense of accomplishment, and have high enthusiasm for learning programming courses such as Java, Python and SpringBoot, but have low interest in theoretical courses and insufficient attention. However, some students have some problems in the interview, such as strong coding ability, weak expression ability and lack of document writing ability. Therefore, the teaching of this course should be based on students' learning situation, guide students to learn in generalization ability, system analysis and design ability, and maintain students' learning enthusiasm through immediate feedback and learning achievement.

2.3 The Potential Risks of Introducing AI Tools

According to the talent cultivation plan of software engineering education and the requirements of information system project managers for candidates, the teacher team of this course has reorganized the knowledge content of "Software Process and Management", and determined the course objectives based on the content of each chapter and its supporting relationship with the course objectives. The specific objectives of the course "Software Process and Management" are as follows.

3. The Design of AI tool Integration Teaching Reform

3.1 The Overall Design Concept

The core idea of this reform is: AI empowerment teaching. AI is not a substitute for students' thinking, but an auxiliary tool to accelerate students' understanding of the theoretical knowledge of this course and lower the threshold of project practice. This course group has established the teaching policy of "theory leading, tool empowerment and project driving", and takes "critical use of AI" as the basic principle, requiring students to always maintain a cautious and critical attitude towards the output of AI.

3.2 Reconstruction of Course Content and Teaching Links

This course group takes the life cycle of software projects as the main line, systematically embeds AI applications in every stage of software development, and students carry out real project practice in the form of groups, and carry out the whole learning process of this course with their projects.

3.2.1 Demand Analysis Stage

Teaching pain point: it is difficult for students to obtain project requirements comprehensively and systematically.

AI empowerment: Teachers guide students to ask DeepSeek questions: "Please list 30 core functional requirements for the' campus second-hand trading platform' we developed and classify them according to' user roles'."

Teaching practice: DeepSeek quickly generates a list of functions including buyers, sellers and administrators, such as product release, search, online communication, order management, background

audit and so on. On this basis, students discuss in groups and optimize the requirements according to the actual situation to form the final demand list. This greatly reduces the difficulty of demand acquisition, and makes students intuitively understand the difference between demand acquisition and demand analysis.

3.2.2 System Design Stage

Teaching pain point: Students feel that drawing UML diagrams is tedious, and there are mistakes and omissions in modeling, and the standardization is insufficient.

AI empowerment: Students use DeepSeek to assist in database design and module design.

Hint Example 1: "According to the demand of' campus second-hand trading platform', design the ER diagram of its database, give the fields of the core data table, and present it in the form of Markdown table."

Hint Example 2: "Draw a sequence diagram for the' Commodity Publishing' function, involving users, front-end and back-end services and databases."

Teaching practice: The chart descriptions or codes generated by DeepSeek provide students with high-quality first draft of system design. Students need to understand, review and modify these modeling graphics, and complete the standardized modeling graphics on this basis. In the process, students deepen their understanding of design principles such as database paradigm and module coupling.

3.2.3 Coding Implementation Stage

Teaching pain point: some students' coding ability is not enough to realize the whole project.

AI empowerment: Students use DeepSeek as a powerful programming assistant.

Hint Example 1: "Using the Spring Boot framework, write a Java code of a '(Product' entity class, including id, name, description, price, seller ID and other fields."

Prompt word example 2: "Help me to write a function to realize the fuzzy search function of second-hand goods."

Hint Example 3: "There is a security hole in my user login code, please help me analyze and fix it."

Teaching practice: DeepSeek greatly alleviates students' coding anxiety caused by unfamiliar grammar and API, and enables them to devote more energy to business logic implementation and system architecture. Teachers can focus on teaching how to write accurate prompts, and emphasize understanding the logical design of codes rather than copying codes, and ask students to modify codes according to practical application scenarios.

3.2.4 Coding Implementation Stage

Teaching pain point: students ignore system testing, and the written test cases are too simple and have low coverage.

AI empowerment: Use DeepSeek to generate test cases and test codes.

Example of prompt: "Write JUnit unit test cases for the' commodity publishing' function I just generated, covering the boundary conditions such as normal publishing, empty price and negative price."

Teaching practice: Through AI, students can quickly establish the importance of testing and learn a comprehensive testing thinking.

3.2.5 Project Management and Document Writing

Teaching pain point: developing document writing is time-consuming and laborious, and students' enthusiasm is low.

AI empowerment: use DeepSeek as a document assistant.

Example of the prompt: "Please write an outline of the Requirements Specification for Campus Second-hand Trading Platform according to the requirements and design we discussed before, including introduction, general description, functional requirements and non-functional requirements."

Teaching practice: Students take the documents generated by AI as the first draft, and modify, refine and supplement them. This greatly saves students' time in document format and basic requirements, and students can directly focus on document review and optimization, which is more in line with industrial practice and improves document quality.

4. Teaching Methods and Implementation Mode

1) Mixed teaching mode: online learning of AI usage skills and software engineering video courses, while offline classes are transformed into man-machine collaboration workshops and project discussions.

2) Guided flip classroom: students use DeepSeek to complete basic tasks before class, and class time is used for in-depth discussion, scheme optimization and critical thinking.

3) Project-driven learning: Students develop the real practical projects of the team in groups of 4-6 people. Course assessment is closely linked to each milestone of the project.

4) AI usage log system: students are required to record the prompts of using DeepSeek at each stage, output the results and their own verification and modification process. This process cultivates students' AI critical thinking and academic integrity.

5. Evaluation and Reflection on Teaching Practice Effect

After a semester's teaching practice, the effect of the reform was evaluated by means of student interviews and comparison of project results.

5.1 Analysis of Teaching Effect

1) Students' interest and participation in learning have been significantly improved: more than half of the students said in the interview that the introduction of AI tools has made the theoretical study of Software Engineering easier. The activity of class discussion and project promotion has been obviously improved.

2) The standardization of software modeling has been significantly improved: The software system modeling graphics completed by students this semester, including use case diagram, E-R diagram, activity diagram, sequence diagram and data table design, have significantly improved the standardization and accuracy of modeling compared with previous sessions.

3) The ability to write documents has been improved: As the AI tool undertakes the initial draft design, students are more willing to participate in the review and optimization of documents and have a deeper understanding of high-quality software document standards.

4) Students' AI literacy has been improved: from the initial blind belief in AI output, students have gradually changed to cross-validation, access to official documents, review and optimization, and their AI literacy has been significantly improved.

5.1 Problems and Reflection

1) The engineering ability of cue words is uneven: the cue words written by some students are too general, which leads to unsatisfactory output results of DeepSeek, and teachers need to train cue words and provide cue words templates.

2) Critical thinking still needs to be strengthened: at the beginning of the semester, there are still students who directly copy and paste the system modeling graphics generated by AI, such as E-R diagrams and use case diagrams. Teachers distinguish the content generated by the above AI in class, and optimize it according to the actual application scenarios, and repeatedly emphasize the importance of verification.

3) Put forward new challenges to the role of teachers: the role of teachers has changed from imparting knowledge to designing learning tasks and guiding human-computer cooperation. This requires teachers to put more energy into designing teaching scenes and learning current AI tools.

6. Conclusion

This study successfully integrated AI tools into the course teaching of Software Engineering. Practice shows that the teaching reform, which is driven by specific project practice and takes AI as the whole process empowerment tool, can effectively meet the characteristics of students' learning situation, resolve various pain points in teaching, and help students establish a modern software engineering thinking of AI collaborative problem solving.

Acknowledgements

Funding

This work was funded by the Higher Education Teaching Reform Project of Guangdong University of Science and Technology under Grant GKZLGC2025057.

References

Chen, L., Chen, P., & Lin, Z. (2020). Artificial Intelligence in Education: A Review. *IEEE Access*, *8*, 752.

Wang, L. L., & Li, G. (2024). A summary of the research on programming teaching assistants based on large language model. *Journal of Software*, *35*(1), 120.

Wang, Q. Y., & Feng, A. (2023). Higher education reform driven by generative artificial intelligence: opportunities, challenges and paths. *Modern Distance Education Research*, *35*(4), 1322.

Zhang, L., & Liu, Q. (2021). Software Engineering Teaching Reform Research and Practice. *Computer Education*, *2021*(10), 120124.