Journal of Contemporary Educational Research, 2026, Volume 10, Issue 7
http://ojs.bbwpublisher.com/index.php/JCER
Online ISSN: 2208-8474
Print ISSN: 2208-8466

A Multi-Role AI Agent for Project-Driven Teaching Practice in Software Engineering

Yan Yan, Runxia Ding, Hui Zhang, Wanjun Wang

School of Computing and Intelligent Technology, Qingdao City University, Qingdao 266106, Shandong, China

Abstract: To address the pain points in traditional software engineering teaching, such as disconnected learning scenarios and limited effectiveness of project-driven teaching, this study designs and implements a multi-role AI Agent named SE_Coach. It simulates roles in software development, such as product managers, and guides students through heuristic questioning and situational simulation. A 16-week controlled experiment was conducted with 80 junior computer science students. The experimental group adopted a model of teacher guidance + AI agent + group collaboration, while the control group used traditional group collaboration. The results show that the experimental group had significantly better self-efficacy and project outcome quality. SE_Coach effectively improves students' practical abilities and collaborative thinking, providing a feasible paradigm for AI-empowered software engineering teaching.

Keywords: Software engineering; AI agent; Project-driven teaching; Constructivist learning theory; Practical capability

Online publication: August 6, 2026

1. Introduction

To meet the new requirements for practical software engineering talents brought by industrial transformation and address the pain points in traditional software engineering teaching. This study, based on the constructivist learning theory, develops a multi-role AI Agent named SE_Coach. As shown in **Figure 1**, a three-element structure is created. SE_Coach simulates roles in software development projects (e.g., product manager, architect, test engineer); it guides students in in-depth participation through Socratic questioning and situational simulation, promoting knowledge internalization and the systematic construction of engineering capabilities. Furthermore, we conduct project-driven teaching practice based on this multi-role AI Agent to verify its effectiveness in cultivating students' software engineering capabilities and collaborative thinking.

Existing research has formed two main approaches. First, optimizing project-driven teaching to align with industrial practices [1]. Secondly, exploring the teaching application of AI agents as collaborators [2]. The former has shifted toward simulating real development management and team collaboration, such as the in-depth integration of the Scrum agile framework and Problem-Based

Learning (PBL) [3]. In the latter, multi-agent collaboration systems can simulate the collaborative work of roles in development teams, providing technical prototypes for building highly realistic project environments in teaching [4]. Despite such explorations, research on constructing highly realistic project practice environments with multi-role AI agents and empirically verifying their teaching effects is still in its infancy, which is the core research focus of this paper.

The core theoretical foundation of this paper is constructivist learning theory, whose central proposition is that knowledge is a meaning system actively constructed by learners through interaction with the external environment based on their existing experience [5]. This theory directly guides the functional design and interaction mechanism of the SE_Coach agent, which is specifically reflected in three orientations: active construction, social interaction, and situational embedding.

The active construction orientation involves teachers configuring the agent and building its knowledge base. The agent guides students to think independently through heuristic questioning, transforming students from passive knowledge recipients into active meaning constructors.

The social interaction orientation means the agent simulates multiple stakeholders to conduct dialogues and interactions with students, enabling students to improve their ability to solve problems in complex scenarios through communication from diverse perspectives.

The situational embedding orientation requires teachers to create project scenarios close to real software development, integrate the agent into the entire process of students' project development, break down the barrier between knowledge and practice, and cultivate transferable capabilities—this is consistent with the essence of situational learning and community of practice theories.

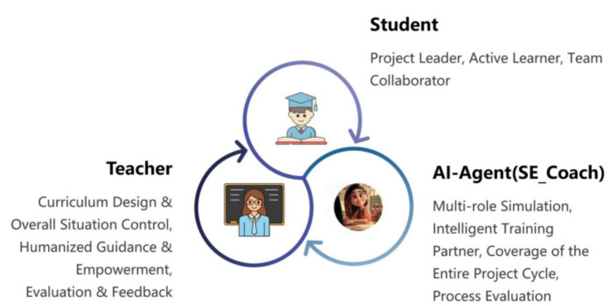


Figure 1. Teacher-student-AI three-element structure

2. Methods

Based on constructivism and aiming at the pain points in software engineering teaching, this study designed and implemented the AI Agent SE_Coach and integrated it into project-based teaching activities. A controlled experiment was conducted over 16 weeks, 48 class hours, with the independent variable being the participation of the AI agent in classroom activities.

2.1. Experimental subjects

The subjects were 80 junior students majoring in Computer Science from two classes, with 40 students in the experimental class (Class 1) and 40 in the control class (Class 2). There were no significant differences between the two groups' prior knowledge reserves and practical foundation in software engineering, ensuring experimental fairness.

2.2. Experimental procedure

Week 1: The course requirements and project tasks were explained uniformly for both groups, and a pre-test was administered at the beginning of the semester. The experimental class was introduced to the functions, usage methods, and interaction rules of the SE_Coach agent, while the control class had no access to the AI Agent. Student grouping and project topic selection were completed for both groups.

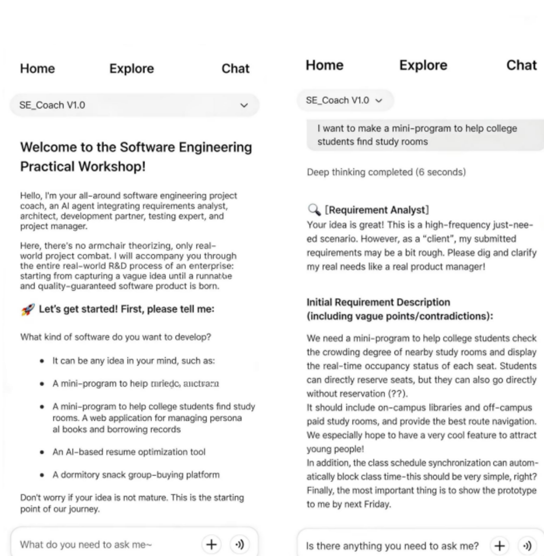
Week 2: Project-driven teaching was carried out. Teachers were responsible for knowledge explanation, as well as progress and quality control.

The experimental class adopted a model of teacher guidance + AI Agent situational simulation + group collaboration. Students interacted deeply with the agent at each project stage to obtain personalized guidance.

The control class adopted a model of teacher guidance + group collaboration. Students solved problems through group discussion or teacher consultation, with the same frequency of teacher guidance as the experimental class.

Final week: A post-test was administered at the end of the semester. The project outcomes of both groups were collected and scored uniformly. The dialogue logs between the experimental class and the agent were collected, semi-structured interviews were conducted with some students from both groups, and data collation and analysis were completed.

Figure 2(a) shows the introduction to the agent welcome page, and **Figure 2(b)** shows the interaction between students and the agent during simulated requirement analysis.



(a) (b)

Figure 2. Multi-role AI agent teaching assistance

3. Results and discussion

To verify the teaching effectiveness of the integrated model of the SE_Coach multi-role AI agent and project-driven teaching, this section conducts a systematic analysis from quantitative and qualitative dimensions, combining students' self-efficacy, project outcome quality, and interview feedback. The specific results and discussion are as follows.

3.1. Analysis of students' self-efficacy

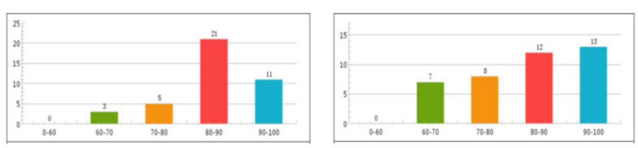
A 1-5-point Likert self-efficacy scale [6] was used, covering five core competency areas. The statistical results of the paired pre-test and post-test data are shown in **Table 1**. The post-test mean values in all areas were significantly higher than the pre-test values, with extremely significant differences ($P < 0.001$). This indicates that the SE_Coach multi-role situational simulation can effectively enhance students' confidence in their practical abilities in software development projects.

Table 1. Results of students' self-efficacy

Competencyarea	Pre-testmean(SD)	Post-testmean(SD)	*t*-value	*P*-value
Requirements analysis	3. 1 (0.6)	3.7 (0.6)	-4.35	< 0.001
Software design	2.9 (0.5)	3.8 (0.7)	-7.1	< 0.001
Coding and debugging	2.8 (0.6)	3.7 (0.6)	-8.51	< 0.001
Testing and quality assurance	2.8 (0.6)	3.7 (0.6)	-6.73	< 0.001
Project management	2.9 (0.5)	3.8 (0.6)	-7.7	< 0.001

3.2. Comparison of project outcome scores

Two software engineering teachers conducted a blind evaluation of the project outcomes of both groups using a unified scoring standard. It covers five dimensions: requirements analysis document, software design scheme, coding quality, test report, and project management records. As shown in **Figure 3**, the average score of the experimental class (**Figure 3(a)**) is 84.68, which is significantly higher than the average score of 81.75 for the control class (**Figure 3(b)**). This indicates that the proposed teaching model can effectively improve the quality of students' project practice, further verifying the model's effectiveness.



(a) (b)

Figure 3. Comparison between the experimental class and the control class

3.3. Qualitative analysis

Coding analysis was conducted on the dialogue logs of the experimental class with the agent, and the results of semi-structured interviews with both groups. The findings are as follows.

Students in the experimental class recognized the efficiency and professionalism of the agent, stating that it could respond quickly and output information in a structured manner, significantly improving the efficiency of requirements acquisition.

Furthermore, the agent's guided questioning helped students focus on functional points and identify implicit requirements.

However, some students reported that the agent still had limitations in understanding ambiguous expressions and connecting contextual information.

Students in the control class generally reflected a lack of personalized guidance and 18 long waiting times for teacher responses when encountering problems.

4. Conclusion

This paper deeply integrates the multi-role AI Agent with project-driven teaching and proposes a new software engineering teaching method. This method not only achieves the in-depth integration of technology and teaching and responds to the trend of artificial intelligence empowering education, but also accurately addresses the pain points in traditional teaching, such as the disconnect between theory and practice, insufficient teaching resources, and limited personalized guidance.

Experimental results show that the SE_Coach, as a dynamic interactive learning companion tool, can effectively improve students' software engineering practical capabilities, self-efficacy, and the quality of project outcomes. Its design and application are consistent with the core of constructivist theory, providing a feasible and scalable practical paradigm for AI-empowered software engineering teaching.

In the future, the SE_Coach can be integrated into online course platforms to solve the problem of personalized guidance in large-scale online learning. At the same time, we will optimize the agent's role adaptability and contextual understanding ability, enrich its application scenarios in software engineering, and further promote the in-depth integration of AI-empowered education and software engineering teaching. This research is expected to provide more references for the implementation of smart education and the cultivation of practical software engineering talents.

Funding

2025 Annual Project of the Special Committee on Educational Digitalization, Research Society of Information and Data for Chinese Colleges and Universities (HED-2025-36)

Disclosure statement

The authors declare no conflict of interest.

References

[1] Veras NL, Rocha LS, Viana W, 2021, An Authoring Tool to Support Flipped Classroom in Software Engineering Teaching. Proceedings of the 17th ACM Conference on International Computing Education Research (ICER 2021), 403–404.

[2] Almeida C, França C, 2023, Improving the PBL Method with Experiential Learning Theory in Software Engineering Teaching. Proceedings of the 4th International Workshop on Software Engineering Education for the Next Generation (SEENG '22), 28–35.

[3] Dou J, Zheng B, 2025, Construction and Innovative Application of Intelligent Agents in College EFL Teaching. Proceedings of the 2025 6th International Conference on Computer Information and Big Data Applications (CIBDA

‘25), 1324–1329.

[4] Lyu B, Li C, Li H, et al., 2025, Exploring the Role of Teachable AI Agents’ Personality Traits in Shaping Student Interaction and Learning in Mathematics Education. In Proceedings of the 15th International Learning Analytics and Knowledge Conference (LAK ‘25). Association for Computing Machinery, New York, NY, USA, 887–894.

[5] Lu Y, Li G, 2025, Enhancing Electric Machine Education: An Integrated Approach Using 3D-Printed Models, Multiple Intelligences, and Gamification. In Proceedings of the 2025 6th International Conference on Education, Knowledge and Information Management (ICEKIM ‘25). Association for Computing Machinery, New York, NY, USA, 1–5.

[6] Meng TS, Po YW, Yun CC, et al., 2026, Rationalizer: Leveraging LLM to Support User Providing the Rationales Behind the Rating of Likert Scale Questionnaires. In Proceedings of the 31st International Conference on Intelligent User Interfaces (IUI ‘26). Association for Computing Machinery, New York, NY, USA, 85–105.