

Exploration of Case Teaching Mode in Software Engineering Course

Yan Wang *

Department of Computer, North China Electric Power University, Huadian Road, Baoding, China

* Corresponding author Email: wangyan1206@126.com

Abstract: Software engineering is a highly practical core course in computer related majors. The traditional teaching model focuses on teaching theoretical knowledge, while the case teaching model introduces practical cases and drives teaching activities with actual project cases, combining practice with theory to cultivate students' practical and problem-solving abilities. The paper analyzes the background and necessity of teaching reform in software engineering courses, and designs a software engineering case teaching mode, and provides practical suggestions. This teaching mode helps to enhance students' interest and understanding of the knowledge they have learned, cultivate their innovative thinking and comprehensive qualities.

Keywords: Software Engineering; Case Teaching Mode; Teaching Reform.

1. Introduction

Software engineering is an important course in computer science [1,2], which applies engineering methods to the software development process and guides software development, maintenance, and management through scientific and standardized methods. The software engineering course has the following characteristics:

1) Interdisciplinary integration: Software engineering courses involve knowledge from multiple fields, such as computer science, mathematics, engineering, etc. Therefore, it is also an interdisciplinary integrated course that requires students to comprehensively apply basic knowledge.

2) Emphasis on process management: The software engineering course emphasizes not only technical content, but also knowledge of software development process management and team collaboration, which is a

comprehensive application of technical content in lower grade majors.

3) Practical oriented: Software engineering courses focus on practical operations to enable students to better grasp the practical skills and experience of software development.

Traditional software engineering courses usually adopt a teaching method of theoretical explanation, but this method has some problems, such as the disconnection between theory and practice, and the lack of initiative among students. Therefore, the paper proposes a case study teaching model for software engineering courses, which is a case-based teaching method. By involving students in actual software development projects, students can better understand the theoretical knowledge and practical skills of software engineering.

2. Design of Case Teaching Mode

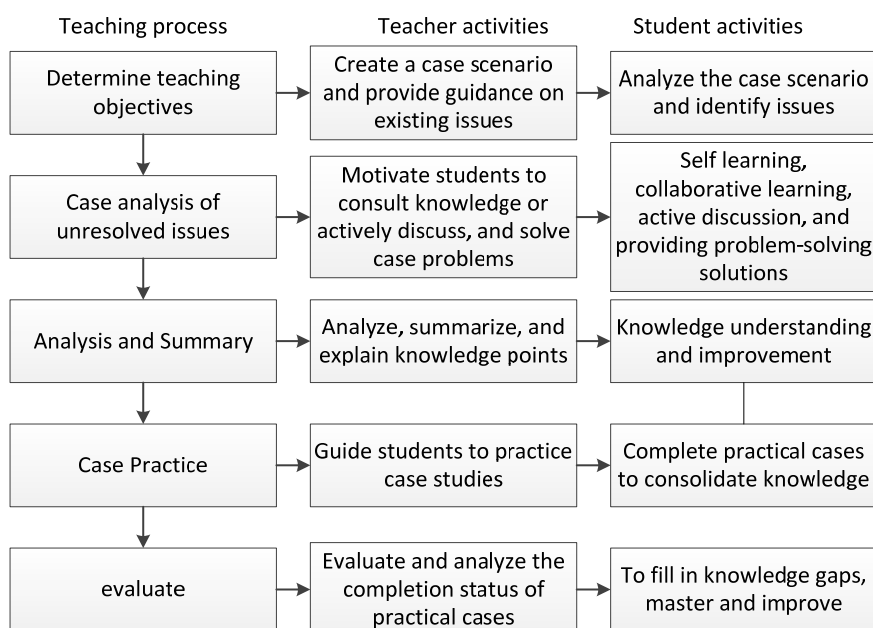


Figure 1. Case teaching mode for software engineering course

Case teaching [3] is a teaching method based on practical cases, which emphasizes extracting universal rules and

principles from practice, enabling students to understand and apply abstract theories through learning and thinking about specific cases. Case teaching has the characteristics of authenticity, inspiration, and comprehensiveness, which can help students better understand theoretical knowledge and improve their problem-solving abilities. As a highly practical discipline, software engineering emphasizes learning by doing through case studies, which can help students better understand and apply software engineering theory. In addition, through group division of labor and cooperation, it can also cultivate students' teamwork spirit and problem-solving ability, enabling them to better adapt to future work environments. Software engineering case teaching mode is designed as shown in the figure.

The design of software engineering case teaching mode needs to consider the following aspects:

1) Clarification of teaching objectives: Determining teaching objectives is the first step in designing a software engineering case teaching mode. The teaching objectives should include the knowledge, skills, and abilities that students need to master. For example, after the stage of software plan, the teaching objectives are to analyze the requirements of the software project. Here, the knowledge, skills, and abilities that students need to master include methods for obtaining requirements, extracting requirements, and evaluating requirements.

2) Case selection: Choosing the appropriate case is crucial for software engineering case teaching mode. A good case should have clear requirements, clear goals, complex business logic, and a complete development process. In teaching practice, the selected cases should be in the application field that students are familiar with, but also need to have a certain degree of complexity, such as library management systems, student course selection systems, etc.

3) The setting of learning methods: In the software engineering case teaching mode, learners not only need to master software development techniques and methods, but also need to improve their teamwork ability through communication and cooperation. Therefore, in teaching practice, setting up project teams, analyzing and solving problems, and practicing cases are all completed through the division of labor and collaboration among group members.

4) Design of evaluation methods: Evaluation is an essential part of education and teaching. In order to help students better understand in which areas they still need to improve, timely evaluate the completion of group projects. This can serve as both a process assessment part of the course and also play a role in identifying and improving knowledge.

3. Case Teaching Practice

3.1. Practical Steps of Case Teaching

Based on the characteristics of students and the course, the practical steps of software engineering case teaching are as follows:

1) Determine case: The teacher selects suitable cases as teaching materials based on the content of the software engineering course and the background of the students. Decompose the project case into several sub tasks according to the software lifecycle, each corresponding to a learning objective, including requirement analysis, system design, system implementation, system testing, etc. Students can also independently choose project cases according to their needs. By involving students in real-life software development

projects, it helps students better understand the concepts and processes of software engineering, and enhances their practical abilities.

2) Student grouping: Students form groups independently based on their interests and abilities, with 3-5 members in each group. Group members should complement each other's strengths as much as possible.

3) Question raising: The teacher guides students to propose unresolved problems based on case studies, and based on existing prerequisite knowledge and reference materials, engage in group discussions and research.

4) Problem solving: Group members work together and propose solutions based on their own expertise and interests. The teacher summarizes, analyzes, and teaches methods. After mastering the relevant knowledge, students adopt software engineering methods and tools for practice.

5) Display results: The results are presented in the form of a system and report, including requirement analysis specifications, design specifications, source code, and testing documents. The group showcases the process and results of problem-solving to the whole class, and accepts evaluations and feedback from other groups and teachers.

Through this teaching model, students can master software engineering knowledge in practice, improve their self-learning and teamwork abilities, cultivate innovative thinking and problem-solving abilities.

3.2. Auxiliary Teaching Measures

In order to better carry out case teaching, auxiliary teaching measures will be taken:

1) Introduced online teaching platform: By utilizing the teaching platform, functions such as online teaching, homework, and discussion are achieved, providing students with online learning resources, enhancing interaction between teachers and students, and enhancing their learning experience.

2) Introduced self-learning mechanism: By providing students with self-learning resources and mechanisms, it stimulates their interest and initiative in learning, and cultivates their self-learning ability.

3) Multiple evaluation mechanisms have been introduced: in addition to traditional written and experimental scores, various evaluation methods have been introduced, such as project evaluation, report evaluation, oral presentation evaluation, etc., to comprehensively assess students' knowledge level and abilities.

4. Conclusion

The software engineering case teaching mode is driven by cases, combining theory with practice, adhering to the idea of discovering problems, solving problems, project practice, evaluation and summary improvement, transforming passive learning into active learning, students can better understand software engineering theory knowledge, and thus improve learning effectiveness. Its advantages include:

1) Rich teaching content: The case teaching mode can cover various software engineering knowledge, such as requirements analysis, design, coding, testing, etc., enabling students to have a more comprehensive understanding of software engineering knowledge.

2) Practical ability enhancement: Enable students to apply theoretical knowledge to practical cases, thereby enhancing their practical and problem-solving abilities.

3)Improving learning interest: This model allows students to participate in practical projects, increasing the fun and challenge of learning, and stimulating their learning interest.

4)Improvement of teamwork ability: Students collaborate in teams to complete practical projects, which enhances their teamwork and communication skills.

5)Practical experience accumulation: The case teaching model can allow students to accumulate experience in practical projects, thereby better preparing for future career development.

In the practice of case teaching mode, teachers should pay attention to guiding the composition of student teams and the division of labor and cooperation in project practice, in order to fully leverage the practical role of project cases.

Acknowledgments

This work was supported by the Association of Fundamental Computing Education in Chinese Universities (No: 2023-AFCEC-270).

References

- [1] Daun M, Brings J. How chatgpt will change software engineering education[C]//Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1. 2023: 110-116.
- [2] Malhotra R, Massoudi M, Jindal R. Shifting from traditional engineering education towards competency-based approach: The most recommended approach-review[J]. Education and Information Technologies, 2023, 28(7): 9081-9111.
- [3] Mei Y. Application of Clinical Case Teaching Mode in Gynecological Nursing Teaching[J]. Computational and Mathematical Methods in Medicine, 2022, 2022.
- [4] Vargas A M, Bisi A, Chiappa A S, et al. Task-driven neural network models predict neural dynamics of proprioception[J]. Cell, 2024, 187(7): 1745-1761. e19.
- [5] Rasa H, Gjøtterud S, Selsaas T C, et al. Student-driven teaching and educational action research combined: an approach to teaching development and student empowerment[J]. Educational Action Research, 2024: 1-18.