ATLANTIS PRESS

Advances in Social Science, Education and Humanities Research (ASSEHR), volume 182

2018 2nd International Conference on Education, Economics and Management Research (ICEEMR 2018)

Exploration and Practice of Collaborative Teaching Mode for Software Engineering Course Group

Haifeng Wang
College of Information Science and Engineering
Linyi University
Linyi, China
gadfly7@126.com

Zhansheng Yang
College of Information Science and Engineering
Linyi University
Linyi, China
yangzhansheng@lyu.edu.cn

Yunpeng Cao*
College of Information Science and Engineering
Linyi University
Linyi, China
caoyunpeng@lyu.edu.cn

***Abstract*—Aiming at the experimental course teaching in software engineering major course group, a collaborative teaching mode based on knowledge point sharing and management is designed. In the same teaching period, multiple course teaching are combined to achieve deep integrated study of students' many major courses, and to strengthen the collaboration and cooperation among teachers. A set of practical innovative training modes for practical courses is achieved to provide a feasible solution for improving the comprehensive engineering attainment of software engineering students.**

Keywords—software engineering; collaborative teaching; practical teaching course group; knowledge point management

I. INTRODUCTION

The practical ability of undergraduates of software engineering major is the core of personnel training in application-oriented universities, which is mainly reflected in two aspects: major ability and profession ability [1]. Major ability is the focus of student training. Students are required to have good system analysis and design abilities, software development abilities, software testing abilities and certain project management abilities. Professional qualification abilities require students to have the abilities to acquire and process information, communication abilities, abilities to use foreign language, international perspectives, abilities of relearning, professional ethics and qualities, abilities of team cooperation and thinking abilities. The practical teaching has a significant effect on the training of major and professional abilities. It is an important process for forming the engineering abilities of software engineering undergraduates. Therefore, the reform and innovation of software engineering practice teaching process mode is of great significance, affecting the training of students' comprehensive engineering abilities and the cultivating of vocational qualification abilities.

II. PROBLEMS IN CURRENT PRACTICAL TEACHING

In the software engineering practice teaching of domestic application-oriented colleges and universities, project-driven teaching method is implemented in computer room. However, the following problems commonly exist:

A. The teacher teaches in isolation when each course is taught

There is a lack of necessary communication between various courses in software engineering practice teaching. Each course has relatively independent teaching plans, tasks and objectives. In this case, a large number of knowledge points will be repeatedly explained and even be missed in the practical teaching of each course. For example, in the practical course of Software Project Management, teachers will ignore some of the conventional knowledge of database design, because the teacher thinks that the relevant teacher of database course has taught it, but the actual teaching effect shows that some problems are caused by forgetting database knowledge. The practical course content of Software Testing and Quality Management is to test Web information management program, but the Web Application Development and Design of the same teaching period only goes to the front-end development part. If the two courses are planned uniformly on teaching goal and tasks are scheduled collaboratively, the practice teaching will be more effective.

B. Students listen to lessons in isolation when each course is taught

During learning process, software engineering students seldom jointly study the knowledge of various major courses actively. In practice, they are only learning to deal with each course. Therefore, it is very difficult for students to establish overall software engineering attainment and professional quality in project practice, and they lack integration of various professional and technical courses. For example, in the same

Fund Projects: Shandong Provincial Excellent Course Project (No. 2013BK106), Linyi University School-level Teaching Reform Project (No.20142245).

study period, two courses Web Application Development and Software Testing and Quality Management are studied. The front-end development knowledge in Web Application Development is closely related to the black box interface testing in software testing. However, when students are learning, they either care about how to design a layout of a web page in front-end development, or care about how to determine the test function point in interface test. However, there is a lack of thinking about front-end design from the perspective of front-end quality management and a lack of ability to determine the scope of software testing from the perspective of design science.

In order to solve the above problems, the practical teaching of software engineering needs to implement collaborative teaching and learning within the scope of the entire course group. Establishing a complete collaborative teaching mode is the key to solving this problem.

III. The Design of Collaborative Practice Teaching Mode

Modern collaborative teaching is a form of planned cooperative teaching that began in 1930 with the leadership of John Dewey [2]. Team teaching refers to a teaching team composed of two or more teachers with different specialities. In one or more subject areas, the group proposes teaching plan together. Teachers use different teaching media, equipment and a variety of different collaborative methods to conduct large-scale teaching activities according to their expertise. Finally, the team members jointly evaluate the students' learning situation. According to modern collaborative teaching thinking, software engineering practical teaching activities are reformed, and teachers of multiple professional skill courses are coordinated to perform collaborative teaching in various ways. Compared with traditional collaborative teaching mode design, the following focuses are highlighted:

A. Collaborative Teaching Team

The core and cornerstone of collaborative teaching is an educational team with collaborative spirit. It is composed of teachers who teach a single major course and teachers with different experience abilities. The collaborative teaching team takes students as service objects, teachers with complementary skills and knowledge expertise and with good communication and cooperation abilities as the main body, software engineering practice courses and major construction as the platform and takes improving students' practical teaching quality as the goal to build a kind of innovative teaching basic form [3]. The collaborative teaching team can solve three basic problems: Firstly, software engineering practice teaching requires particularly high professional skills, and it is difficult for a teacher to master all the knowledge and skills of a professional group. There is a difference in knowledge structure and teaching methods in any teaching team. Collaborative teaching team can complement each other, promote the development of software engineering and increase the professional level of the entire community. Secondly, there is not only cooperation but also competition among teachers in collaborative teaching team, which inevitably stimulates teachers' teaching enthusiasm and promotes the improvement of teaching quality. Thirdly, collaborative teaching team is the main force to complete the task of software engineering practice course group knowledge point collaborative teaching system, and it performs collaborative evaluation and assessment for the same student groups from multiple angles, to promote the experimental learning effect of college students.

B. Collaborative practice teaching content based on knowledge point

In the collaborative practical teaching content design of software engineering, the project-led multi-course collaborative teaching is carried out. In the practical teaching of one course, a software design project is introduced. A team consisting of 2-3 teachers from different professional courses focuses on solving practical problems. The team mainly designs a single course and involves teaching content of two or more course related theories and techniques.

In teaching content design, project task is the main line, and specific teaching content is achieved through knowledge point management. The knowledge point conveys the basic unit of teaching information in teaching activities, including theoretical principles, concept definitions, formulae and techniques [4,5]. Each practical course in software engineering consists of a collection of knowledge points and logical relationships between knowledge points. However, collaborative teaching team must complete the logical relationship diagram of all knowledge points in the entire course group, and design teaching content on this basis to facilitate collaborative teaching process.

1) Division of Knowledge points

In the design of knowledge point’s pedigree, we must first solve the principle of knowledge points. In each course, knowledge points are divided into core knowledge points and extended knowledge points. The core knowledge points cover the basic concepts, technical principles, and basic theories of a single course. It belongs to the classic knowledge and skills that must be taught. Extended knowledge points mainly include new knowledge and techniques, as well as knowledge points related to other professional courses. The principle of division granularity of knowledge points is as follows: For the core knowledge points, chapters and sections are used as units to split the knowledge points so that the content is full when the teacher teaches knowledge. If the granularity of the core knowledge points is small, the knowledge points will be too messy to cover the mainline structure of project. On the other hand, extended knowledge points, especially of other professional courses, must be divided into small granularities. The fine-grained division of extensive knowledge points has two advantages: (1) It is convenient to carry out personalized teaching by using micro-class means for extended knowledge points. For example, students meet a database knowledge point database index in software test practice course. Individual students want to immediately recall the database index knowledge point, and can immediately open the micro-course of extended knowledge points. (2) In the project-oriented knowledge point management, fine-grained division of knowledge points facilitates the optimization of students’ learning activities.

2) Knowledge Point Pedigree Diagram Design

For software engineering professional course group, the organic integration of logical diagrams of knowledge points in each course will form a professional knowledge pedigree diagram [6]. In knowledge point pedigree diagram, each course uses a fine-grained chain edge to achieve knowledge fusion and collaboration. In order to realize the design of knowledge point pedigree diagram, the following logical relationships between extended knowledge points should be considered: (1) The knowledge points appear in partial order relationship in hereditary map to form parent-child relationship; (2) The time sequences of knowledge points is biased on partial order relationship. Before learning a knowledge point, another related knowledge point must be learned. There is a preceding or succeeding relationship between them; (3) Two knowledge points are at the same level, and there are no father-child or other relationships between knowledge points. Their learning order is of regardless, constituting a parallel relationship. Knowledge points are connected with each other through the above three basic relationships to form the knowledge pedigree diagram of software engineering practice teaching system. The pedigree diagram facilitates knowledge management and cooperation in collaborative teaching and facilitates students' positioning, expansion and learning of knowledge.

3) Linked List of project knowledge points

The knowledge point pedigree diagram formed by software engineering specialty course group reflects the complex relationships such as network structure, interaction, crossover, evolution and derivation among knowledge elements or knowledge groups. Our goal is to complete the knowledge point management of each professional course project based on knowledge pedigree diagram. Firstly, a single course is used as unit to create knowledge point bank. Then, a list of knowledge points involved in project tasks is established by using project tasks in each course as clues. The teachers of this professional course determine the core knowledge points and relationships. The collaborative teacher team identifies the extended knowledge points involving multiple courses together, and searches extended knowledge points according to keywords in the pedigree diagram of knowledge points. Finally, a knowledge point linked list is established for every experimental task project. Teaching management and student evaluation is carried out on the basis of knowledge point linked list of each project.

C. Collaborative teaching mode

Firstly, a collaborative teaching team builds a linked list of knowledge points according to each experimental project in teaching plan and course design. The teaching content is designed with professional teacher as head and other professional teachers as assistance. Secondly, in the teaching process, the teachers create a project task list and assign tasks according to students' basic conditions, learning needs and learning abilities. At the same time, under the drive of project task, students use the methods of group cooperation and communication, problem discussion, difficulty analysis, task decomposition and so on to conduct independent learning by using MOOC, teacher s’ PPT, experimental training materials etc. During project training, when students encounter knowledge points of other professional courses, they choose supplementary learning methods such as micro-course, MOOC and so on according to their learning needs and status, and use an autonomous and personalized learning form to solve the knowledge points related to other courses. [7]. The professional teachers do not teach the content of project task any longer. Instead, they analyze learning situation according to students' task summary reports, collect common problems and common mistakes, use concentrated lectures, organize students to discuss problems in project design in groups, help students establish problem solving methods and strategies, guide students to analyze and solve problems, and form student-centered inversed practice classes. Finally, the project task is evaluated and summarized. Teachers should summarize and evaluate the learning process of project, pay attention to students' performance in learning process, and weaken the evaluation of project results. On the premise of analyzing the degree of internalization of each student's knowledge point, the achievement degree of project design ability and the accomplishment degree of engineering practice ability of the student's problem solving are evaluated, and the scores of this project training are obtained eventually.

IV. Implementation of Collaborative Practice Teaching Mode

The implementation of the practical teaching mode of software engineering course group based on collaborative teaching mode aims at the course system for cultivating students’ project design and program implementation capabilities of software engineering major. The mainly involved courses include C language programming, C# programming, Web program development and design, C++ programming and other programming language courses; data structures, database, algorithm design, computer networks, software engineering, software project management and software testing, graduation design and other practical and training courses.

A. Construction of Collaborative Teaching Team

The collaborative teaching team builds a heterogeneous teaching group with professional instructors as core. Firstly, a reasonable team goal is formulated. The team's goals must be measurable. For example, the average score of team members' teaching assessments is more than 90. Secondly, team roles and division of labor are planned reasonably and scientifically. According to the academic expertise and personality characteristics of each teacher, a reasonable division of roles is carried out to avoid dislocation and overloading of roles. Finally, an effective assessment mechanism is established. The assessment of team performance is strengthened and amplified so as to make the entire team's teaching goals and results clear and predictable, and the material rewards are directly linked to the performance goals of the entire team.

B. Construction of Collaborative Teaching System

The collaborative teaching team is responsible for completing the construction of collaborative teaching system. Linked lists of knowledge points are established for the content of experimental items for each professional course respectively.

Based on knowledge point linked list, the relevant course contents, micro-course, MOOC etc. are designed to guide students' learning. A knowledge point pedigree diagram is established for the entire software engineering major, and the construction of knowledge point database is completed after formalization processing. The evaluation of students' learning process and effects should also be carried out based on the linked list of knowledge points, to realize quantitative evaluation and information management system.

V. CONCLUSION

Through investigation and analysis of the practical teaching mode of domestic software engineering, it is found that the traditional experimental teaching mode tends to ignore the collaborative learning of multiple professional courses, resulting in that students miss the learned knowledge points or knowledge points are repeated. Collaborative teaching mode is a modern teaching mode, but the collaborative practical teaching mode with knowledge point management as core is not common. Students use project tasks to drive practical tasks, and perform personalized autonomous learning through knowledge point linked lists. Teachers gather difficult points of learning and obstacles of task completion by knowledge points, and finally use knowledge points to evaluate students' learning. This is a good experimental teaching mode, and it can be implemented in more application-oriented undergraduate high schools to make software engineering experiment teaching better and create better engineering practice and training environment for students.

REFERENCES

[1] Lv Pin, Wang Huaiting, Wang Xin. Software Engineering Professional personnel Training Scheme of Comparing with Foreign Universities and Deep Integration Certification [J]. Computer Education, 2017(1):38-43.(In Chinese)

[2] Liu Yingchun. Rational Analysis and Practical Prospect of Collaborative Teaching [J]. Journal of Zhejiang Normal University (Social Sciences), 2010, 35(3): 114-117. (In Chinese)

[3] Liu Xiaojian. On the Team Building of Collaborative Teaching[J]. University Education Science, 2011(1):41-45. (In Chinese)

[4] Xiao Kun, Li Xiaojian, Chen Shihong. A formal study of the operation of courseware knowledge points [J]. Computer Science, 2011, 35(1):181-184. (In Chinese)

[5] Shi Yueding, Zhang Shuyou, Xiang Chun. Research on Representation and Relevance Technology of Knowledge Point in Online Course[J]. Journal of Zhejiang University, 2003,37(5):508-511. (In Chinese)

[6] Hao Yuexing, Li Dongmei, Liu Yu. Pedigree-based Knowledge Retrieval Method for Specialized Fields[J]. Journal of Central South University, 2013, 44sup (2):313-316. (In Chinese)

[7] Zhao Xin. Knowledge Point Division and Micro-course Teaching Design of the Course Computer Network [J]. Journal of Dalian University, 2016,37(3):116-118. (In Chinese)