ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202502_8(2).0041

Research on the Teaching Reform of Software Engineering Course under the Background of Audit Evaluation

Chuancheng Ren, Guangchao Wang, Shuhai Liu and Cong Liu

School of Computer and Information Dezhou University, Dezhou 253023, China

Abstract

The quality assurance concept is an important characteristic of this round audit evaluation of undergraduate education and teaching. This paper takes the teaching reform of the software engineering course at Dezhou University as an example, analyzes some phenomena that exist in the course teaching process, and proposes course reform measures from four directions centered on the quality assurance concept: the implementation of course reform ideas, learning from others, the transformation of the teacher roles, classroom innovation, and experimental teaching theory. This paper provides a reference for other course reforms under the background of audit evaluation.

Keywords

Quality Assurance Concept; Audit Evaluation; Software Engineering Course; Learning-Centered.

1. Introduction

The emphasis on student-centered approaches, outcome-oriented strategies, and continuous improvement as quality assurance principles is a key characteristic of the current round of educational and teaching evaluations. Dezhou University, as a regional applied undergraduate institution, participated in the 2016 Ministry of Education's undergraduate teaching evaluation and has selected the second category, second type of evaluation for this round. From the key review indicators of the second category, it is evident that this round of evaluation focuses on the implementation of the quality assurance principles of student-centered, outcome-oriented, and continuous improvement in the course reform and construction process. Among them, professional courses are crucial for cultivating students' professional knowledge, abilities, and qualities, and the effectiveness of their reform largely reflects whether the quality assurance principles of the evaluation are effectively applied. Therefore, within the context of this round of evaluation, it is necessary to analyze the challenges faced in the reform and construction of professional courses, uphold and integrate the quality assurance principles of student-centered, outcome-oriented, and continuous improvement, and explore and research the reform and construction of professional courses.

Software engineering courses play an important role in preparing students for related careers such as software project requirements, software analysis and design, programming, software quality management, and software project management. As a core professional course in computer science, it is essential for enhancing the teaching quality of software engineering courses and meeting the evaluation requirements. This paper focuses on the software engineering course as a research object to discuss the exploration and research on the application of quality assurance principles in its reform.

2. Problems Encountered in Curriculum Reform

Analyze the key review focus in the second type of evaluation in this round and its relevance to the construction of professional courses. The key points can be summarized as whether the

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202502_8(2).0041

course construction helps improve students' learning attitudes, enhances the development of students' practical abilities, implements a "student-centered, teacher-led" classroom teaching model, builds a project case library, improves course assessment, and fosters a quality culture. We believe that aspects such as teaching management systems, course teams, student learning conditions, classroom teaching, and course resource development will present some challenges in the process of integrating quality assurance principles into the software engineering course.

2.1. There is a phenomenon of continuous improvement in the teaching management system.

After the previous round of teaching evaluations, the school has continuously improved and refined its teaching management system at the institutional level according to relevant requirements, and the individual colleges have also developed corresponding measures. Taking the course assessment management guidelines as an example to illustrate the improvements in teaching management:In the previous round of teaching evaluation's corrective phase, the course syllabus specified that student assessment scores would consist of two parts: regular performance and exam scores. The regular performance accounted for no less than 30%, and the objective questions in the exam accounted for no less than 30 points. During the online course phase, regular performance was updated to include attendance, with regular performance now accounting for no less than 40%, and the objective questions in the exam accounting for no less than 20 points. In preparation for the engineering education accreditation phase, the theory of student-centered, outcome-oriented, and continuous improvement was advocated, and the syllabus was revised. It specified that assessment methods would be divided into formative and summative assessments, and clearly stated that formative assessments would not include student attendance, with formative assessment scores accounting for no less than 50%. In this round of evaluation, the formative assessments were again specified to include midterm exams or unit tests, and the summative assessment no longer includes objective questions.

2.2. There is a situation where the course team operates inefficiently.

Issues such as limited team effectiveness, varying teaching abilities among members, fragmentation of team members, and inefficient team operation also exist within the software engineering course team. As a professional course, the course team for software engineering tends to have a relatively unstable leader, with one teacher taking charge of course development while other teachers share the outcomes of the course construction.

2.3. There are some issues related to students and classroom teaching.

Some students, during their learning of the software engineering course, not only face issues such as weak knowledge structures, poor self-learning abilities, and smartphone addiction, but also show individual differences in self-discipline, learning attitudes, and the ability to apply knowledge practically. The problem of weak knowledge structure can lead to redundant work in the teaching of the software engineering course, and this phenomenon must be addressed in classroom teaching. After the online course phase, some students do not consciously put their phones in the phone bags before class, and even when the teacher approaches them, they blatantly continue using their phones.

We have attempted to implement different stages in the classroom teaching process, such as pre-class, in-class, and post-class activities to meet various requirements. Additionally, as per the college's instructions, we have explored different teaching models, transitioning from a "content-centered" teaching philosophy to a "student-centered" approach. However, we have encountered some unusual situations in the teaching of the software engineering course, such as some students not preparing for the course in advance, more than a third of the students not purchasing the textbook, and even refusing to buy second-hand books. In class, some students

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202502_8(2).0041

bring only their phones, and not other learning tools. They do not actively engage with the teacher, stand without answering questions when asked, and after being asked to hand in their phones, some students lie down on their desks, sleeping or mentally drifting away from the classroom.

2.4. There are issues in the goals of course construction.

The author has led and participated in the application process for school-level high-quality courses, excellent courses, and online courses in the software engineering program. Although a large amount of course resources, such as videos, lecture slides, question banks, exam banks, and case libraries, has been accumulated during the development of the software engineering course, the actual goal of the course construction was to meet the requirements for project applications, rather than being built around the quality assurance principles of student-centered, outcome-oriented, and continuous improvement.

3. The Overall Approach to Curriculum Reform

In line with the key points of the current round of audit evaluation, the reform of the software engineering course needs to follow the approach of "one engine, two wings, and three integrations."

"One engine" refers to the cultivation of virtue and the core task of education, which is to implement the fundamental mission of cultivating virtue and fostering talent. The reform of the software engineering course is driven by this engine, adhering to a student-centered approach, guided by societal needs, and following the talent cultivation plan for the computer science program. It gradually encourages students to develop patriotism, love for society, love for their class, love for their profession, love for their peers, and a passion for learning.

"Two wings" consist of: one wing being the quality assurance principles of student-centered, outcome-oriented, and continuous improvement, and the other wing being the ability to apply software engineering knowledge to solve complex problems in the field of software engineering.

"Three integrations" include three aspects:

1. The integration of various teaching modes in the teaching process to better serve classroom teaching;

2. The integration and construction of teaching cases to form a project library with local characteristics;

3. The integration of teacher roles, where teachers not only need to be competent in teaching the software engineering course but also continuously improve their teaching skills and the ability to combine theory with practice.

4. Specific Strategies for Curriculum Teaching Reform

4.1. Drawing on Insights from Other Successful Experiences in Curriculum Reform

In the context of the current round of audit evaluation, the reform of the software engineering course can benefit from the viewpoints and suggestions proposed by scholars. Zhiyi Li[1] analyze the quality assurance concepts and their connotations in this round of educational and teaching evaluations, providing substantive strategies on areas such as the functions of quality subjects, the configuration of quality elements, and methods of quality assurance. These insights can inspire how to implement the quality assurance principles (one wing) in the construction of the software engineering course. Regarding the adaptation to the higher authorities' requirements for improvements in teaching management systems, Zhang Anfu et al. [2] provide the composition, standards, and ideas for building a quality assurance system,

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202502_8(2).0041

which can guide the software engineering course in continuously adapting to changes in teaching management systems and avoiding detours. Gong Jianmin et al. [3] offer suggestions for writing course syllabi based on a student-centered approach, which can better assist the author in reading, understanding, and applying the already written software engineering course syllabi. Li Zhiyi et al. [4] analyze the logical starting point of teaching design and propose strategies for outcome-oriented course teaching objectives, content, and evaluation methods, providing certain ideas for the author to implement student-centered classroom teaching reform.

4.2. The Role of Teachers Shifts Toward a Student-Centered Approach

The transformation of the role of teachers in the software engineering course is an essential prerequisite for implementing a student-centered approach. Before the course began, during the holiday, the author familiarized themselves with the societal demands for knowledge, skills, and abilities required for jobs in software requirements, design and development, software testing, and maintenance through various channels. This helped drive the author to expand their software engineering knowledge and continuously master related software development tools, such as UML modeling tools and Git version control platforms. The author also revisited relevant literature on quality assurance principles, absorbing the course reform experiences of other scholars, and reviewed and understood the relationship between teaching and learning in a student-centered context. The author then aligned the course objectives with the software engineering course syllabus, reconstructed the teaching content, reorganized relevant teaching materials, expanded the project cases, and updated the software engineering course teaching platform. The initial teaching methods corresponding to the teaching content were determined.

During the teaching process, the author actively observed and analyzed students' self-learning abilities, their understanding of prerequisite knowledge, and their self-regulation skills through various means and channels. Based on these findings, the teaching content was appropriately adjusted, teaching methods were improved, and teaching skills were enhanced. The author believes that teaching the software engineering course is not merely a process of delivering knowledge but also about caring whether students can apply what they have learned to solve problems after mastering the course content. To achieve this, teachers need to transition into roles that combine both "strict" and "caring" characteristics. Teachers should hold high standards for students' attitudes, learning processes, and outcomes, while showing care and encouragement to students who are falling behind or making progress in their studies.

4.3. The process of classroom reform is a gradual one.

One of the key focus areas in this round of evaluation is to assess whether classroom teaching is "student-centered and teacher-led." It is essential for the instructor to clearly understand its connotation. The goal of classroom teaching reform is to ensure that every student achieves both their course objectives and corresponding graduation goals. It is important to maintain the original materials supporting classroom teaching, such as teaching calendars, student attendance sheets, lesson plans, textbooks, and process evaluation forms, in paper format for random inspections by higher authorities. This ensures that the software engineering course reform can be carried out smoothly throughout the semester. Additionally, all materials generated during the teaching process should be digitized to facilitate review by evaluation experts.

Furthermore, the instructor needs to analyze the students' learning situation and adopt appropriate classroom management and teaching strategies. For example, students in the 2022 cohort studying the software engineering course have relatively weak self-regulation abilities. In the early stages of the semester, the teacher must strictly check whether students put their phones in the designated phone bags before class, and this check should be done at the start of

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202502_8(2).0041

every class. Over time, this can help students develop the habit of putting away their phones and focusing on self-study.

Another example is that students are unwilling to sit in the first three rows of the classroom. One solution the author implemented is that students who wish to resubmit assignments or improve their grades must meet the condition of sitting in the front three rows for at least four classes. If they fulfill this condition, they are given the opportunity to resubmit their assignments.

The software engineering classroom serves as an important space for implementing both student autonomous learning and teacher guidance. Teachers need to respect students' individuality and influence them through their words and actions, gradually helping students develop good study habits, master learning methods, and apply their knowledge. By analyzing the results of each assessment, the teacher can track whether there is any change in students' active learning abilities. Attention should also be given to the learning outcomes of struggling students. Timely communication with these students can help increase their sense of presence and learning potential. The teacher should abandon the mindset that the goal is achieved simply by meeting the evaluation thresholds set by the college, and instead focus on continuous improvement in students' learning progress.

4.4. Implementing the "Student-Centered" Concept in Experimental Teaching

The "student-centered" teaching concept requires that, after learning the course, students should be able to clearly identify the expected learning outcomes. These outcomes include: the knowledge achievements related to mastering software engineering theoretical knowledge, the technical achievements of gathering and organizing materials and mastering experimental tools, and the ability to apply both knowledge and technical achievements to solve problems in the field of software engineering. Based on the course objectives outlined in the software engineering syllabus, the software engineering experimental courses are designed to include verification experiments, design experiments, and comprehensive experiments. Verification experiments are used to test students' understanding of basic software engineering knowledge; design experiments assess students' mastery of basic experimental skills; and comprehensive experiments evaluate students' ability to apply software engineering knowledge, tools, and methods to solve problems.

The software engineering experimental courses primarily consist of five major experiments: software development documentation standardization, software requirements engineering, software technical solutions, software analysis and design, and software testing. Each major experiment contains several sub-experiments. For instance, the software requirements engineering experiment includes sub-experiments such as software requirements gathering and software requirements modeling. The entire software engineering experimental course follows the main theme of analyzing and designing an actual software project. For example, the experiment topic for the 2022 cohort is titled "Analysis and Design of the Le Ling Jujube Popular Science Website" (this topic is based on our actual horizontal project). The teacher requires students to form experimental (study) groups by themselves, with each group consisting of 3 to 5 members. The writing of the experimental plan, experimental report, and the presentation of the results are all completed by the group as a unit, and the team score is based on the individual performance of each member in the class.

Taking the software development documentation standardization experiment as an example, the objective is for students to become proficient in reviewing relevant literature and project cases, selecting a development process model based on the experimental topic, and summarizing the necessary documentation required in the software development process to facilitate communication at each stage of development. The focus and difficulty of the experiment lie in how to standardize the development documents according to national or

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202502_8(2).0041

industry standards for software development documentation. The intended competency outcome is to help students develop the ability to identify and address documentation issues in the software development process by effectively utilizing literature and organizing standardized development documents. The experiment is implemented and assessed by randomly selecting two groups, who report and discuss their experimental results in class, with other students providing scores based on their team performance.

Additionally, to enhance students' interest and enthusiasm for experiments, we incorporate smaller "micro-experiments" into the larger experiments. For example, in the software requirements engineering experiment, students are asked to design the main interface of the Le Ling Jujube Popular Science Website. The results vary across groups, with some groups drawing the interface by hand, while others use software tools to create it. During the presentations and discussions, students increasingly raise questions and express opinions. For instance, in the verification experiment for class diagrams, we add a micro-experiment to demonstrate the conversion of single and multiple classes into program code. The experimental groups refer to relevant resources and demonstrate how they use modeling tools to convert single or multiple classes into the specified program code during their presentations.

As shown, by adding micro-experiments to the original experimental framework, students are encouraged to apply course knowledge and literature to solve encountered problems, which in turn increases their enthusiasm for the experiment.

5. Conclusion

In the teaching reform of the software engineering course, phenomena such as changing teaching management systems, loose course teams, diverse learning conditions, and unclear course construction goals exist. In order to better implement the quality assurance principles of this round of audit evaluation, the teaching reform of the software engineering course follows the approach of "one engine, two wings, and three integrations." By drawing on insights from other successful experiences, the reform aims to transform the role of instructors around quality assurance principles, confirm that "student-centered, teacher-led" classroom teaching is a continuous improvement process, and implement the "student-centered" concept in experimental teaching. This round of undergraduate education audit evaluation serves as an opportunity to actively explore software engineering course reform based on quality assurance principles, aligning with the evaluation's key focus areas, and more effectively serving the cultivation of application-oriented talents and the achievement of course objectives.

Acknowledgements

Dezhou University Institutional Teaching Reform Project (DZXY202008)

References

[1] Li Zhiyi. Key Points of Design and Implementation for the New Round of Audit Evaluation. Higher Engineering Education Research, 2021, Vol.3, p.9-15.

[2] Zhang Anfu, Xu Wu. Characteristics of the New Round of Undergraduate Education Teaching Audit Evaluation Scheme. Higher Education Development and Evaluation, 2021, Vol.37(No.06), p 1-13+119.

[3] Gong Jianmin, Xiao Beilei. Who is the Course Syllabus Written For? - On the Implementation of the Student-Centered Concept. Higher Engineering Education Research, 2020, (No.03), p 143-150.

[4] Li Zhiyi, Wang Zewu. Outcome-Oriented Course Teaching Design. Higher Education Development and Evaluation, 2021, Vol.37(No.03), p 91-98+113.