Computing, Performance and Communication Systems (2023)
Clausius Scientific Press, Canada
DOI: 10.23977/cpcs.2023.070101
ISSN 2371-8889 Vol. 7 Num. 1

Software Testing Instructional Design and Practice Based on BOPPPS Model

Yang Yan[a,*], Li Rong, Wang Sai

Computer School, Central China Normal University, Wuhan, Hubei, China
[a]ms_yangyan@ccnu.edu.cn
**Corresponding author*

Keywords: Software Testing, BOPPPS, Instructional Design

Abstract: To address the problems in traditional teaching of Software Testing course, the BOPPPS model is integrated into its instructional design. And the boundary value analysis in black-box testing is used as an example. Some methods and specific operations are given for each instructional module, so as to improve students' learning autonomy and classroom efficiency. The feedback shows that the BOPPPS model is significantly helpful to improve the teaching effect.

1. Introduction

Software Testing is a compulsory course for Software Engineering majors. The course introduces the basic theories, methods and techniques of software testing, focusing on the practicality, standardization and systematization of the contents [1]. It aims to let students experience the importance of software testing in the software development cycle after the previous coding courses, and to change from the traditional understanding of development over testing to the awareness that development and testing are equally important [2]. Traditional Software Testing courses are often based on theoretical lectures, mainly on the basic theory of software testing, testing methods and testing process. The coherence of teaching contents is low, and students generally lack participation, feedback and thinking, which leads to unsatisfactory classroom teaching effect [3]. Therefore, it is necessary to design efficient teaching sessions to enhance the quality of Software Testing classroom teaching and improve the learning effectiveness of students.

The BOPPPS model is a model of teaching and learning process that emphasizes student participation and feedback, which is highly recommended by many prestigious schools in North America. It helps teachers break down and analyze the teaching and learning process, identify blind spots, and improve teaching effectiveness [4]. Its application shows that it is an effective, efficient and productive model that can promote students' active participation in classroom teaching [5]. This paper discusses the application of the BOPPPS model to the teaching of Software Testing to improve students' interest and classroom participation in order to achieve the purpose of effective teaching.

2. BOPPPS Model

BOPPPS is an effective instructional design model that originated in North America. It divides the teaching and learning process into six modules from the perspective of teaching organization and implementation. These modules follow the teaching cycle of "instructional objectives → instructional activities → instructional assessment → instructional objectives" in the double-loop learning theory, which means that the instructional objectives and design are revised under the guidance of teaching practices [6]. Compared with traditional teaching methods, the BOPPPS model emphasizes the importance of learning objectives and highlights the student as the main subject. Teaching activities are centered on students, stimulating students' participation and interaction in class, and focusing on students' feedback on learning, which is an effective way to carry out classroom teaching nowadays [7].

Each module of the BOPPPS model is described separately as following.

(1) Bridge-in: Teachers need to establish the connection between the introduction and the learning content based on the knowledge points of the lesson and the interest of the students. A variety of forms, methods, and tools can be used to attract students' attention and get them to join the lesson with interest and enthusiasm.

(2) Objective: Teachers set achievable and measurable learning objectives for class contents from the students' perspective so that they know exactly what they can do after learning the lesson.

(3) Pre-assessment: Pre-assessment is an essential stage before the start of the lesson. It is a way to understand the mastery of students' prior knowledge by testing them such as questions, choices, voting, etc., in order to prepare for subsequent teaching.

(4) Participatory Learning: Participatory learning is a prominent part of BOPPPS, highlighting the students' main role and making them actively participate in the teaching process.

(5) Post-assessment: After the class content is taught, students' learning is checked by post-assessment to see if the learning objectives presented in the class have been achieved.

(6) Summary: Finally, the knowledge points of the teaching content is summarized and reflected upon. This session allows the teacher and students to participate together to further deepen the learning content of the lesson.

To sum up, the BOPPPS model forms a good closed loop for the whole teaching content from before class (bridge-in, objective, pre-assessment), during class (participatory learning), and after class (post-assessment, summary), which is conducive to the teaching activities [8]. Students are deeply involved in the whole learning process, which has better teaching effect compared with the traditional one-way output teaching model of teachers.

3. BOPPPS-Based Instructional Design and Practice of Software Testing

Software Testing, as a mandatory course for Software Engineering majors, introduces testing methods and testing processes. Black-box and white-box testing methods are the main focus of this course. The textbook presentation of this content is mainly based on a large number of theoretical text statements combined with examples. In the teaching process, students are poorly engaged and lack interest in learning, which leads to poor learning results and little understanding of the nature of testing methods. To solve this problem, the BOPPPS teaching model is introduced into the teaching process, aiming to enhance the classroom teaching effect, make students actively participate in the learning of knowledge, and improve the learning efficiency. The main tasks of each module of the BOPPPS model and their implementation forms and delivery places are shown in Table 1.

Table 1: Main tasks of each module and their implementation form and delivery place in the BOPPPS model

No.	Module Name	Main Tasks	Form	Delivery Place
1	Bridge-in	Guide students to think and start teaching	Introduction with practical examples or inspiring questions	Offline class
2	Objective	Clarify teaching objectives (or learning objectives)	Specific, clear, measurable statements	Offline class / Cloud platform
3	Pre-assessment	Understand students' prior knowledge and abilities	Class questions or quizzes	Offline class / Cloud platform
4	Participatory learning	Adopt positive learning strategies to make students deeply participate in classroom teaching	Teacher questions, class exercises, knowledge competitions, case studies, group discussions, teacher-student discussions, and student mutual evaluations.	Offline class
5	Post-assessment	Check the completion of students' learning objectives and assess the effectiveness of class teaching	Objective-specific quizzes and homework	Offline class / Cloud platform
6	Summary	Summarize and conclude the teaching contents	Mind maps	Offline class

The following is an example of the boundary value analysis in the black box testing in Software Testing course to study the design of teaching process based on the BOPPPS model. The specific implementation process is as follows.

Bridge-in (B): Firstly, we introduce the "passenger ship overload incident" - a bug in the automatic ticketing system of a passenger terminal makes the actual number of tickets sold for a certain type of small vessel much larger than the number of tickets that should be sold. As a direct result, the actual number of passengers on board was much higher than the authorized number of passengers, which nearly caused a major accident. Through this bridge-in, students are asked to think about the seriousness of software bugs, and are guided to further think about how to effectively find the defects in the software and improve the quality of the software. The next step is to introduce students to a common black box testing method, the topic of the lesson - boundary value analysis. Regardless of the variety of rich functions implemented in a software, its internal implementation inevitably requires the definition and judgment of a variety of data boundaries in order to be able to carry out the required processing for different data boundaries. The judgment of data boundaries is where the software is more prone to errors, often making the software make the wrong processing, thus failing to meet the software requirements. For this situation, the boundary value method can be used to design test cases.

Objective (O): Students are introduced to the instructional objectives of the course. The teaching objectives are described in three dimensions: knowledge objective, ability objective, and affective objective. The knowledge objective is to understand and master the boundary value analysis method for designing test cases. The ability objective is to be able to flexibly use the boundary value analysis to design test cases for specific problems encountered in practice, and to detect possible defects in the system through boundary value test cases. The affective objective is to make students realize the importance of software testing in the software development process, to cultivate a scientific and rigorous working attitude, and to develop a careful and conscientious scientific research habit.

Pre-assessment (P): In the pre-assessment session the teacher can ask students questions such as whether they understand the components of a test case, whether they know what a boundary value is, and why there is a boundary value analysis. The discussion is first conducted in small groups,

then representatives convey the results of the discussion, then collectively review the concept and specific composition of test cases, and finally give an example to show that in the software development process, defects are often more likely to occur on the boundary of the input or output domain of the object under test rather than inside the input or output domain. For example, the programmer misspells $\leqslant$ as < and the counter happens to undercount once. The pre-assessment enables the instructor to understand how well students understand the concept of test cases and boundary value testing, and prepares them for the subsequent use of boundary value analysis to design test cases.

Participatory Learning (P): Participatory learning allows students to learn by doing and implicitly receive course knowledge. It is a very important teaching session in the BOPPPS model, where the teacher first asks how to choose the boundary value data in a software test and then gives an example. For example, if the year is specified between 1800 and 2050 in the next-day problem, how should the boundary data for the year be selected for the test. Students can discuss this spontaneously in groups. In fact, the test data can be taken from 7 points, 1799, 1801, 2049 and 2051 in addition to 1800, 1925 and 2050 as test data. Through this example, students are guided to think about how to determine the boundaries of the input or output domain to ensure that all possible boundaries of the object under test are covered, and how to determine the range of neighborhoods near the boundaries of the input or output domain to facilitate the timely detection of all defects lurking near the boundaries. The process and method of designing test cases using boundary value analysis is introduced around the learning objectives by selecting values exactly equal to, just greater than, or just less than the boundary as test data, rather than typical or arbitrary values in the equivalence class. By citing different project examples to introduce the boundary value test case design method, students can feel the importance of boundary value testing and really understand this test case design method. Participatory learning can take various forms, such as teacher questions, class exercises, knowledge competitions, case studies, group discussions, teacher-student discussions and student mutual evaluations, to liven up the classroom atmosphere and try to involve every student in the learning process.

Post-assessment (P): The main function of the post-assessment is to test the teaching effect and understand the students' knowledge mastery. By using the quiz and homework functions of our cloud platform, students' knowledge is assessed by assigning corresponding test questions or after-class homework. The content of the post-assessment should reflect students' achievement of learning objectives, and consider how students use knowledge to solve practical problems after learning. Through the delicate design of test questions for different objectives, the learning effect can be tested from different aspects. After the test is completed, teachers can use the statistical function of the cloud platform to identify the knowledge points with high error rates and give targeted comments and explanations on the answer results to further consolidate the teaching content.

Summary (S): At the end of the class, the teacher reviews the whole lesson, summarizes the class content in the form of a mind map, and then introduces the learning theme of the next lesson. On the other hand, it is also possible to introduce civic education into it, so that students can understand that in addition to professional skills of software testing, they must also develop good qualities (such as carefulness, innovation, etc.) and play the spirit of teamwork in order to successfully complete each task in the future software testing work. The main role of the teacher's summary is to link up the various scattered knowledge points in the class so that students can develop a knowledge network in their minds, thus enhancing the teaching effectiveness.

The above is the instructional design of the boundary value analysis method in black box testing technique based on the BOPPPS model. The effect shows that the BOPPPS model focuses more on guiding students than the traditional lecturing, and the classroom atmosphere is prompted to be

more active through the interlocking teaching design and compact teaching pace. Students think with questions and try to explore the answers to their questions step by step. Their questioning and speaking rates increased significantly. There is a discussion forum for the course content on our cloud platform, where students can express their opinions and share their insights and the latest testing techniques. The survey found that most students thought the class content was well designed and were able to proactively give feedback and communicate with the instructor about problems encountered during classroom learning. In the anonymous teaching evaluation at the end of the period, students' satisfaction with the course has improved significantly compared with previous years.

4. Conclusions

The BOPPPS-based Software Testing instructional design advocates student-centered teaching, which makes students truly participate in it and exercise their independent thinking ability compared with traditional teaching methods. At the same time, the teaching activities are closely focused on the objectives to guide students' learning step by step to achieve the integration and efficient absorption of knowledge. Finally, through summary and feedback, gaps are identified and missing knowledge points are made up to consolidate learning achievements. Practice shows that this teaching method largely improves students' ability to analyze and solve problems, stimulates their interest in learning, and achieves good teaching effects.

References

[1] Guo W., Tan F. Design and Practice of Teaching Black-Box Testing Technique - Equivalence Class Classification Method Based on BOPPPS Model [J]. Computer Knowledge and Technology, 2020, 16(26): 138-139.

[2] Yan T. Teaching Design of "Software Testing Techniques" Based on BOPPPS Model—A Black-Box Testing Equivalence Classification Method as an Example [J]. Popular Standardization, 2020(13): 125-126.

[3] Wang J., Wu J., Zheng D., Fang Y. Research on Blended Teaching of the Software Testing Course Based on OBE and CDIO [J]. Software Engineering, 2019, 22(10): 54-56.

[4] Ma L., Gao J., Zhou G. Design and Implementation of Online Software Engineering Instruction Integrating OBE and BOPPPS [J]. Computer Knowledge and Technology, 2021, 17(6): 10-12.

[5] Zhou W., Li J., Bao W., Liu L. Knowledge Graph Analysis of BOPPPS Model Research in China [J]. Journal of Higher Education Research, 2019, 42(3): 44-52, 66.

[6] Cao D., Yin X. The BOPPPS Teaching Mode in Canada and Its Implications for Higher Education Reform [J]. Research and Exploration in Laboratory, 2016, 35(2): 196-200, 249.

[7] Wang Y. Instructional Design Based on Blended Teaching Model of "Internet + BOPPPS" [J]. Computer Knowledge and Technology, 2021, 17(3): 170-172.

[8] Zhang H. Teaching Design Practice of BOPPPS Model in Software Testing Technology Course [J]. Computer Era, 2022(07): 133-135.