Using Student-generated Questions in Software Engineering Courses

Deepika Badampudi[1]

[1]Blekinge Institute of Technology, Sweden

Abstract

Using student-generated questions in assessments has several benefits, such as improving active learning and a deeper understanding of the subject content learned. Due to the recent shift in education modes (online education), tackling student participation and attendance has become more challenging. In this study, I report on the application of using student-generated questions in two different courses offered to master students in a software engineering program.

Keywords

Student-generated questions, Assessment, Student-centric education, Active learning

1. Introduction

The type of assessment can influence students' attitudes toward learning. Using student-generated questions for student-centered assessment allows students to reflect on their learning. It also allows students to demonstrate an understanding of the learning outcomes. Student-generated questions are a writing-to-learn approach where the students submit questions reflecting on the information they received [1].

Changes in the mode of education (remote/on-campus) due to the pandemic has emphasized existing challenges such as low attendance rate. In addition, active student participation is often challenging regardless of the education mode. Asking students to generate questions based on the lecture contents can improve active learning [2] and student attendance by involving students in the discussions about the questions.

Student-generated questions shifts the learning from *acquiring knowledge* (from teacher) to *learning knowledge* (student driven) [3]. Teacher-generated questions are used as assessment methods widely. However, student-generated questions are not as widely used even though teaching professionals considered shifting the responsibility to students in formulating questions as an effective assessment method [4, 5].

Researchers have investigated the use of student-generated questions for improving algorithmic thinking skills [6], and reading comprehension [7] in elementary school, and teaching physics [8] and pathology [9] in medical undergraduate education. In higher education, student participation can be challenging, particularly when international students are involved [10]. Student-generated questions can be effective in improving international student participation [10].

A study investigated student-generated questions in different subjects in the final academic years for courses in Computer Engineering [11]. Student-generated questions is used as an assessment method in other disciples. However, it is not extensively discussed in the area of software engineering in university education.

This study reports the application of using student-generated questions in two software engineering courses in the first and second-year master's program. The students are mainly international students. Canvas [1] is used as a learning platform in both courses.

2. Case 1: Seminars in software engineering course

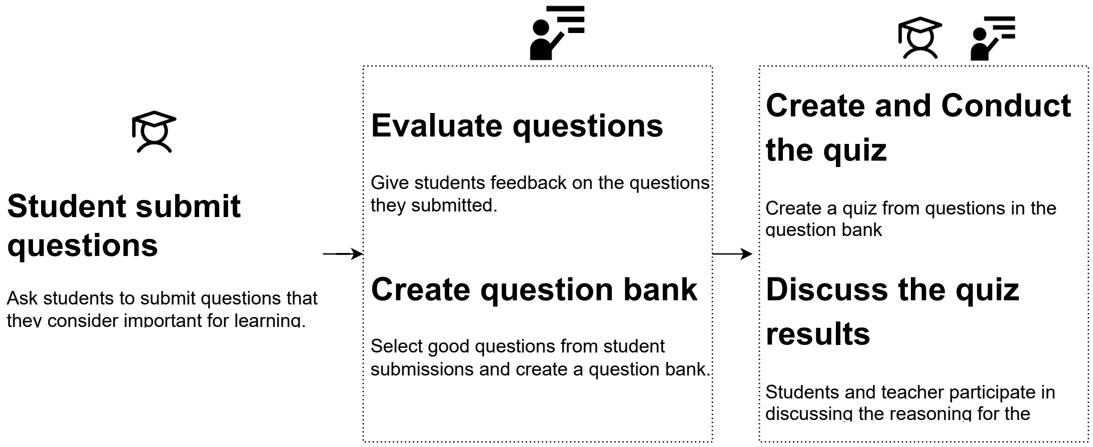
The seminars in a software engineering course are offered as the first course for first-year master students. The objective of the course is to give an introduction and overview of software engineering (SE) research and practice. The course consists of seminars on the different SE knowledge areas. Each seminar is given by an expert in the area. As a course responsible, I was interested to know what students understood from each lecture. After all the seminars on each SE area were concluded, I conducted an end seminar where the students could discuss what they had learned and clarify any misconceptions. The end seminar also provided a good opportunity for me to provide more explanations for topics the students needed help with. The student-generated questions facilitated understanding of students' learning in each SE knowledge area.

The application of student-generated questions in the

Proceedings of 6th Software Engineering Education Workshop (SEED 2022) co-located with APSEC 2022, 06-Dec, 2022, Japan (Virtual)

deepika.badampudi@bth.se (D. Badampudi)
0000-0002-6215-1774 (D. Badampudi)

[1] https://www.instructure.com/canvas


Student submit questions

Ask students to submit questions that they consider important for learning.

Evaluate questions

Give students feedback on the questions they submitted.

Create question bank

Select good questions from student submissions and create a question bank.

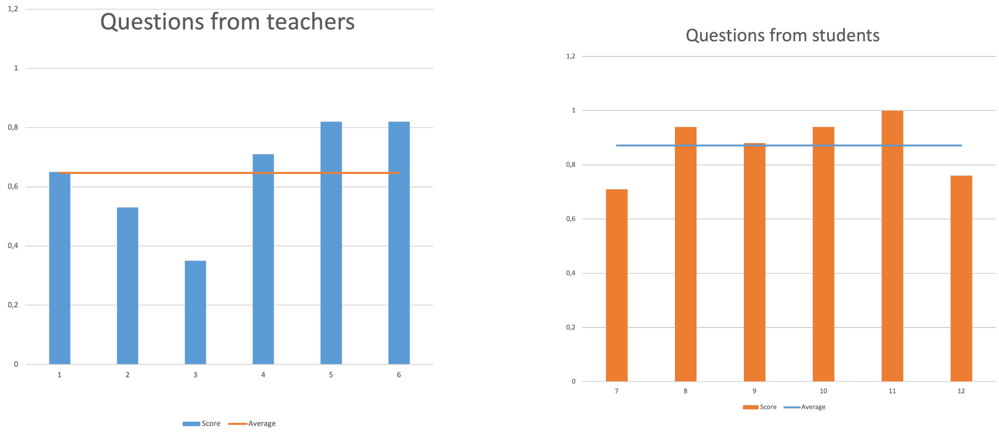
Create and Conduct the quiz

Create a quiz from questions in the question bank

Discuss the quiz results

Students and teacher participate in discussing the reasoning for the

Figure 1: Application of student-generated questions in Case 1.



(a) Teacher-generated questions

(b) Student-generated questions

Figure 2: Quiz results using teacher and student-generated questions

Seminars in SE course is depicted in Figure 1 and described as follows:

- **Submission of questions** - The students should submit one question for each seminar in the course. The questions should be based on the key takeaways of the seminars. The objective is not just to ask students to submit questions; instead, the students should motivate why their selected questions are important for learning. In addition, the students should also provide alternative answers (typically four) and a motivation for why each alternative is correct or wrong.
- **Evaluation of questions and alternatives** – Evaluation is based on how students select, formulate, and motivate the importance of the questions.

In addition, how students choose the alternative answers (tricky or straightforward) and justify the right and wrong alternatives. For example, students can include some common misconceptions in SE as alternatives. The students get either a pass or fail result for their submissions. If the questions, the alternatives, and the justifications for them are reasonable, they pass and earn credits for the submission. Sample of the feedback given to a submission - "*You do not argue why the selected question is essential for learning. For example, for Q2, you have described the importance of quality. However, you do not describe and elaborate on why it is important to know the different quality attributes.* "

- **Creation of the question bank** – The teacher selects good questions from the submission to create a question bank. Including questions with alternatives containing misconceptions can contribute to good discussions. The idea is to add more questions each year and expand the question bank.
- **Creation of the quiz** – The teacher creates the quiz by selecting the questions from the question bank. The objective is to select at least one question from each student. The questions can be a collection of questions from current and previous year students' submissions.
- **Discussion of the quiz results** - After the quiz, the teacher discusses the overall results (not individual). For example, why some students selected the wrong alternatives. The discussions are interactive. The students also participate in the reasoning for choosing the alternatives.

I included some teacher-generated questions the first time I used student-generated questions. I asked each expert to provide one question I could use in the final quiz. In the final quiz, half of the questions were teacher-generated. Figure 2 shows the quiz result using teacher and student-generated questions. On average, the questions from students had a mean score of 0.87, which indicates most of the class got the answers correct. The questions from the teachers received lower scores (mean score of 0.65), indicating that fewer students gave correct answers. The reason for the difference in mean scores is unknown. One possible reason is that the student-generated questions are easier. However, except for one question, the difference is not significant.

3. Case 2: Program comprehension

Program comprehension was one part of a course on software maintenance and evolution offered to second-year master students. The course was divided into three parts: program comprehension, code quality evaluation, refactoring, and finally, adding a new feature to the existing system. The students should first comprehend the source code before evaluating the code quality and refactoring the code smells. In the first instance of the code, we asked the students to comprehend the source code as part of the self-study assignment, which was not evaluated. However, to make the program comprehension part more structured and to ensure that students put effort into understanding the code, we introduced student-generated questions.

The application of student-generated questions in the program comprehension is depicted in Figure 3 and described as follows:

1. **Submission of comprehension tasks** - The students should submit ten program comprehension tasks. The questions should be specific to the program and not generic. We identified five categories of comprehension tasks and asked the students to submit tasks covering the different categories as shown in Table 1. We provided the categories to ensure that the students did not submit similar tasks. For example, there is a risk that students submit all ten tasks explaining a piece of code. The students also should submit the answers to the comprehension tasks.
2. **Evaluation of the comprehension tasks** - The students get feedback on their submissions. If the students submit questions that do not represent categories or provide incorrect answers, they are asked to complement the assignment. Similar to Case 1, the students earn credits for submitting comprehension tasks.
3. **Selecting comprehension tasks of the workshop and conducting the workshop**- The teacher selects some of the comprehension tasks submitted by the students for a workshop. The students are allowed to work on the tasks in groups where they can discuss with their group members.
4. **Discussion on the comprehension tasks** - Once the students perform the tasks, they should discuss and reflect on the comprehensibility. The students should reflect on the difficulty level and the time taken to complete the task. For example, how much time it took to understand a method's functionality? Did the method implementation and naming make it easier to understand its functionality?

4. Lessons learned

In this section, I provide some reflections on applying S-generated questions in two software engineering courses offered in a master's program.

Instructing students on how to submit questions/tasks: The students may not understand the difficulty level of questions/tasks to submit. It is important to give instructions on the level of questions expected. In Case 1, the students were asked to submit questions corresponding to higher levels of Bloom's taxonomy (see Figure 4). And in Case 2, the students were provided with examples of different categories of tasks that they should cover in their submission. Providing detailed instruction is important to help students understand the assignment requirement. The instruction to submit the questions/-tasks should be well aligned with the course objective.

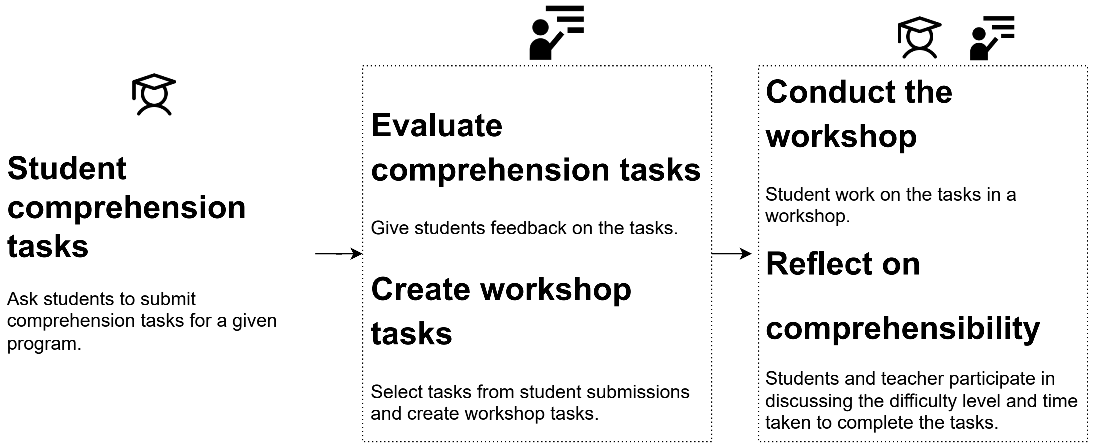

Student comprehension tasks
Ask students to submit comprehension tasks for a given program.
Evaluate comprehension tasks
Give students feedback on the tasks.
Create workshop tasks
Select tasks from student submissions and create workshop tasks.
Conduct the workshop
Student work on the tasks in a workshop.
Reflect on comprehensibility
Students and teacher participate in discussing the difficulty level and time taken to complete the tasks.

Figure 3: Application of student-generated questions in Case 2.

Table 1
Categories of comprehension tasks

Category: Explaining the code
Given a piece of code, explain the purpose of the code in plain English.
Given a piece of code, explain the method for realizing a specific purpose.
Given a variable, constant or method, explain its function.
Category: Finding location
Identify the code that defines the text of button X.
Identify code or component that is relevant for a particular domain concept.
Category: Analyzing impact
What is the effect of changing lines x in y?
If button X is pressed, what parts of the code are executed?
Category: Understanding complexity
Given the comment in file X and line Y, what is the reason for this comment? Could this comment be removed by improving the code's comprehensibility?
Where in component X lies the most complex code? What is the reason for the complexity?
Category: Exception handling
What actions the software system takes in response to exceptions, such as timeout?

Student-generated questions are not intended to minimize teachers effort: Student-generated questions aim to devise a student-centered assessment rather than reducing teacher's effort. It is important to provide feedback on the questions/tasks submitted by the students.

Improved attendance rate: When the students are asked to formulate questions based on the content taught in the lectures, they are more motivated to attend the lectures. Participating in the workshop was mandatory in Case 2 therefore, the effect on the attendance rate could not be measured.

Reflection on students' learning: In Case 1, reviewing the questions submitted by the students helped evaluate if the students were able to identify the key takeaways from the seminars. In addition, it also provided input to the teachers on what topics the students perceived to be important. In Case 2, the tasks submitted by the students and the student reflections in the workshop provided an understanding of students' learning.

Active participation: The use of student-generated questions resulted in a gamification effect. The students were interested to know which of their questions/tasks made it to the final quiz. In addition, the students could see what type of questions their peers submitted. If most of the class cannot answer a question correctly, then the student who submitted the question participates in the discussion actively.

Consider backup questions/tasks for assessment: Relying on students for quiz questions can be risky, par-

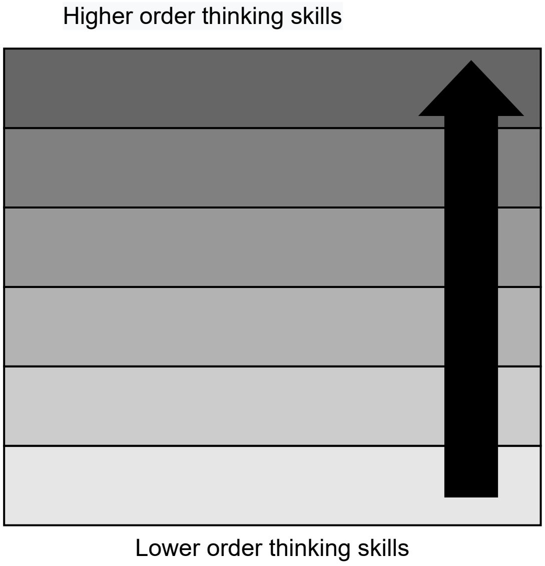


Figure 4: Blooms taxonomy [12].

ticularly if the students do not submit good questions. The teachers can create some questions as a backup to mitigate this threat. In addition, the teachers can create a question bank containing good questions from every cohort. However, creating a question bank is not possible in Case 2 when the software repository is changed every year.

5. Conclusions

Student-generated questions are effective in improving active learning and attendance rate. However, there are many other factors that influence active learning and attendance rate, which also should be considered for any effective application of assessment methods. Asking students to generate questions based on lectures' content rather than on a generic course level is effective in improving attendance. As part of future work, I intend to investigate student experiences of using student-generated questions.

Acknowledgments

Thanks to the teachers involved in courses - Seminars in software engineering and Evolution and maintenance.

References

[1] R. L. Bangert-Drowns, M. M. Hurley, B. Wilkinson, The effects of school-based writing-to-learn interventions on academic achievement: A meta-analysis, Review of educational research 74 (2004) 29–58.

[2] C. Chin, Student-generated questions: Encouraging inquisitive minds in learning science (2002).

[3] F.-Y. Yu, Scaffolding student-generated questions: Design and development of a customizable online learning system, Computers in Human Behavior 25 (2009) 1129–1138.

[4] S. I. Brown, M. I. Walter, The art of problem posing, Psychology Press, 2005.

[5] L. D. English, Promoting a problem-posing classroom, Teaching children mathematics 4 (1997) 172–179.

[6] C.-C. Hsu, T.-I. Wang, Applying game mechanics and student-generated questions to an online puzzle-based game learning system to promote algorithmic thinking skills, Computers & Education 121 (2018) 73–88.

[7] B. R. Mays, H.-C. Yeh, N.-S. Chen, The effects of using audience response systems incorporating student-generated questions on efl students' reading comprehension, The Asia-Pacific Education Researcher 29 (2020) 553–566.

[8] A. E. Kay, J. Hardy, R. K. Galloway, Learning from peer feedback on student-generated multiple choice questions: Views of introductory physics students, Physical Review Physics Education Research 14 (2018) 010119.

[9] R. Lakhtakia, F. Otaki, L. Alsuwaidi, N. Zary, et al., Assessment as learning in medical education: Feasibility and perceived impact of student-generated formative assessments, JMIR Medical Education 8 (2022) e35820.

[10] S. Lomer, L. Anthony-Okeke, Ethically engaging international students: student generated material in an active blended learning model, Teaching in Higher Education 24 (2019) 613–632.

[11] R. Arruabarrena, A. Sánchez, J. M. Blanco, J. A. Vadillo, I. Usandizaga, Integration of good practices of active methodologies with the reuse of student-generated content, International Journal of Educational Technology in Higher Education 16 (2019) 1–20.

[12] D. R. Krathwohl, A revision of bloom's taxonomy: An overview, Theory into practice 41 (2002) 212–218.