A Case Study on the Application of Case-Based Learning in Software Testing

Saurabh Tiwari
DAIICT Gandhinagar, India
saurabh_t@daiict.ac.in

Veena Saini
NIT Jalandhar, India
veenas.cs.16@nitj.ac.in

Paramvir Singh
NIT Jalandhar, India
singhpv@nitj.ac.in

Ashish Sureka
Ashoka University, India
ashish.sureka@ashoka.edu.in

ABSTRACT

Software testing is a popular mean of examining the adequacy of a developed product. However, in academic institutions more emphasis is given to software development than ensuring its quality. In order to address the gaps between existing university-level software testing education and the training standards used in industry, we experiment with employing a popular teaching method Case-Based Learning (CBL) for the first time to facilitate the training of selected software testing concepts at tertiary-level. The CBL exercise is conducted for undergraduate students of DAIICT, Gandhinagar (India) to cultivate the decision making skills in a self-learning environment. After the CBL execution we collect students' responses through a short survey and perform an empirical analysis on the survey results. The outcome of this CBL practice is positive as a majority of students are able to achieve the five stated objectives of CBL. We examine that there is no statistically significant difference between students' responses to two different CBL cases that we use to practice some aspects of software testing. We also investigate the difference in students' feedback based on gender diversity. Moreover, we draw useful inferences from the opinions of TAs (Teaching Assistants) about the CBL sessions.

KEYWORDS

Software engineering education, case-based learning, software testing, teaching methodology

1 RESEARCH MOTIVATION AND AIM

Testing is one of the most crucial stage in Software Development Life Cycle (SDLC). However, Software Testing is not taught as a separate course in most of the university settings and there is a paucity of effective teaching and learning techniques in this domain. Seeing the requirements of today's era, the traditional lecture-based learning approach is not sufficient and there is a need of active learning methods in software testing education [13]. Though there is no pedagogy that fits well in every scenario, but it was observed that teaching methods that include real world examples followed by discussions, team work, decision making tasks, brainstorming, and presentations, engaged students more in classroom activities and proved very much effective for the overall growth of students [11]. One such interactive learning technique is the Case-Based Learning (CBL).

The previous literature reflects that CBL is used since long in the fields of Health Science education, Law education, and Business education [6], but the application of CBL in Software Testing course is unexplored. Looking at the importance of CBL and the issues of software testing education, we aim to introduce case-based learning for teaching Software Testing course. Our objective is to examine the effectiveness of CBL in teaching Software Testing discipline. We intend to develop software testing cases and share them publicly through Software Engineering Case-Based Learning Database (SEABED)[1] with the Software Engineering (SE) community.

A case in CBL is a unique, complex, and uncertain narrative structure of some contemporary interest arousing event or problem [9]. In order to answer the case questions, students need to investigate the problem thoroughly and apply the concepts that they learned during the lecture sessions [8]. Apart from CBL implementation, our goal is to examine the impact of gender diversity and change in case problem on students' responses.

2 RELATED WORK AND RESEARCH CONTRIBUTIONS

In this section, we present the works related to our study and highlight our novel and unique contributions.

2.1 Experience Reports on Teaching Software Testing

Fraser et al. made software testing education enjoyable by mapping core software testing concepts to the categories of puzzles in the framework of a game. With each level progress in the game, students grasped better testing skills [1]. Garousi et al. used industrial tools and projects for undergraduate software testing labs to demonstrate the real world testing scenario to the students [4].

2.2 Case-based Learning for SE Education

Saini et al. proposed an open source web-based software engineering case-based learning platform. They provided a case repository along with a case template, that provides guidelines for case writing

[1] http://seabed.in/

ISEC'18, Hyderabad, India

DOI: xx.xxx/xxx_x

[11]. Garg et al. created a case related to software architecture and introduced a Case-Oriented Learning Environment. This helped students to learn better software engineering skills [3].

2.3 Case-based Learning in other Domains

Kundra et al. used CBL for teaching the concepts of Compiler Design. They observed improvement in the learning, critical thinking, engagement, communication skills, and team work of students [5]. Peplow et al. compared the responses of female and male medical students towards a CBL program. They observed that the female students benefited more than the male students from initial discussions and group activities, and developed better communication skills [10].

In contrast to the existing work, this paper reports 3 novel research contributions:
(1) First implementation of CBL for teaching and practicing the concepts of Software Testing discipline for a large class of 164 students.
(2) Empirical analysis (including gender-based study) of CBL execution at Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT)[2], Gandhinagar, India.
(3) Writing two original software testing cases *StalwartX Case*[3] and *Browser Case*[4] and submitting them to existing software engineering case repository (SEABED).

3 EXPERIMENTAL SETUP

In this section, we present various elements of our experimental study.

3.1 Subjects

The subjects for this study were 164 third-year undergraduate students taking an elective course on Software Testing and Quality Analysis IT415 in Winter 2017 at DA-IICT, Gandhinagar, India. The CBL session was conducted as a graded exercise for the course. All the students have passed a course on Java programming, and an introductory course on Software Engineering as prerequisite to the IT415 course.

3.2 Cases

We created two software testing cases along with two sets of questions to assist planned CBL sessions related to software testing. The first case intends to facilitate the concepts of responsiveness testing and performance testing (in terms of speed or response time), whereas the second case facilitates the concepts of security testing and functionality testing. In order to fill the gap between industry and academia perspectives, the idea of the two cases was proposed by our industry author and then formulated by the authors from academia as per the course needs.

3.3 Experimental Design

We used a single-factor incomplete block design during the experiment [14]. In our experimental design (Table 1), every treatment (subject groups) has not worked on all the cases (means one group has worked only on one case), hence this is an incomplete block design. Also, in our study there is one blocking factor, i.e., students experience. Hence, the design is a single-factor incomplete block design. One can say the design to be a complete block design, If every treatment is used and replicated the same number of times in every block and if each treatment is used once in every block, it is a randomized complete block design. The class size of 164 students was divided into four groups having 41 students each. Out of 164 students, 113 were boys and 51 were girls. While randomly assigning students to each team, we ensure that the group must be balanced with respect to gender. A total of four sessions, each of two hours duration, were conducted. Eight Teams of five students each (with an exception of one six-member team) were randomly assigned in the session itself (shown in Table 1). Each case was linked to five questions, and each team member was given the responsibility of one specific question.

Table 1: Design of the Experiment

Case-based Learning				
Session	**Case**	**Subject**	**Group**	
Session I	Case I	S1-S41	G1-G7 (5 subjects)	G8 (6 Subjects)
Session II	Case II	S42- S82	G1-G7 (5 subjects)	G8 (6 Subjects)
Session III	Case I	S83-S123	G1-G7 (5 subjects)	G8 (6 Subjects)
Session IV	Case II	S124-S164	G1-G7 (5 subjects)	G8 (6 Subjects)

4 CBL EXECUTION AND DATA COLLECTION

4.1 Preparation and Subject Training

Four lecture sessions were conducted to teach the students about the topics related to the two cases, and one lecture was conducted to enable them with the basic idea of CBL. We chose to teach topics related to non-functional testing aspects. These aspects of software testing allow students to understand the problem and its solution based on the given constraint and environment. Testing non-functional aspects (e.g., performance, security, responsiveness, etc.) involve (1) deriving the test objectives for each one of them, and (2) based on those test objectives, identifying the ways by which the system under development can be tested. Similar to this, the real-world cases have many possibilities through which a system or part of it can be tested and analyzed. Keeping both in context, we have chosen to use these topics for the CBL sessions. The quantitative results and the experiences of the subjects and the TAs, highlighted the usefulness of CBL in learning the aforementioned topics.

4.2 Case Description

This section briefly describes the two cases along with few questions. Full cases can be downloaded from the footnote link 3,4.

Case A: *Suppose you are a Test Manager of Google Chrome. Everyone can associate with Google Chrome. Google Chrome is used by millions of people on various device types of different screen size — smartphone, desktop/laptop, tablets. Chrome is ubiquitous; a full Google account integration gives you a variety of features that are not limited to a single device. Testing Responsiveness (means the browser adjusting to the screen) is very critical for a browser. Emulating different screen sizes is not easy. Although there are various tools available that checks the*

[2]http://www.daiict.ac.in
[3]http://seabed.in/case-study/StalwartX_Case.pdf
[4]http://seabed.in/case-study/Google_Case.pdf

browser's response on different size viewports, but still manual testing is required to ensure whether all parameter are up to the mark. The browser should support various web languages efficiently. It should effectively support orientation of screens in tablets, mobile phone, etc. Response time for each application is also an important parameter that needs to be tested. The time taken by various widgets like dialogue box, check box, etc. to respond also need consideration. There are certain websites which include many APIs and libraries that may significantly slow down your browser.

Questions:
Q.1 Suppose the browser support various platforms. What is your planning strategy to test the responsiveness of the system?
Q.2 How will you ensure that the browser effectiveness will not get affected by the APIs and libraries used by the website? What test strategy will you apply to ensure the browser's performance?
Q.3 Write all the possible test cases for responsiveness testing with reference to the browser effectiveness when emulating with different screen sizes.

Case B: *StalwartX (driver-less vehicle) is an emerging company with breakthrough technology. They are building autopilots and driver-less vehicles. Suppose you are a test manager at StalwartX. It is not easy to achieve self-driving capability under real world conditions and requires a lot of perfection to create a car which can make decisions on its own. For acquisition of data from its surroundings, StalwartX is embedding the cars with eight cameras and one front facing radar. The car is equipped with a super computer, which uses a self -learning process (deep learning) so that it can tackle any situation. Self-driving serviceability varies with jurisdiction and relies heavily on comprehensive software validation and regulatory approval. These driver-less vehicles are capable of controlling themselves with no or very less human interference. The security of the car heavily depends upon the apps used to control this device. The car owners login to the Swt app (developed by StalwartX) and control the car. Autopilots are always connected to internet that makes them more vulnerable to cyber threats. The attack sometimes requires the car to be connected to a malicious Wifi hot-spot set up by the hacking team and then they can interfere with the controlling of the car. These attacks are capable of controlling the display and judgmental capabilities of the car like braking abruptly, slamming of the door while it is moving. Seeing the rapid growth of automakers, having higher standards of security become very crucial. Customers are keen to know the effectiveness of these autopilot software and how much they can rely upon them.*

Questions:
Q.1 What test cases will you apply in order to ensure functionality parameters are up to the mark? Enlist them in the form of user stories.
Q.2 What if somebody hacks the Swt app and gets the credentials of the car owner? One may track the car, enables keyless functionality, and even steals the car itself. What proactive measures will you check in order to ensure cyber security?
Q.3 How does the car detect whether radar or cameras are malfunctioning or not? What measures the car would take once it gets to know that there is some malfunctioning or weird behavior?

4.3 Execution

After the necessary preparation and training, the study was carried out in three phases, 'Case Understanding', 'Response Gathering', followed by 'Discussions'. In *Understanding Phase*, the case was allotted to each team and asked them to understand the case. Also, the instructor and one TA was there to help them to understand the case. In *Response Gathering Phase*, each student was asked to take the responsibility for one of the five case questions. However, the response to each question had to be finalized through team work. All responses to the case questions and survey questions were collected on paper sheets.

Each student team was asked to submit their responses on the response sheet for all the five questions, and then present their responses to the case questions. The students were encouraged to use different kind of resources for reference like internet, textbooks, etc. A targeted *discussion* was carried out after each team's presentation for their responses to the case questions.

The CBL exercise was assigned a weightage of 10% across all student assignments for the Software Testing course. The best team in a session was allotted 10 extra marks that would be added in their end-semester examination marks. The overall exercise was conducted by two groups each comprising one faculty member and one M.Tech research scholar from SE domain. Also, the students' responses were evaluated and graded both at the student-level and group-level in the class-session itself. When one team was presenting their work, other groups were asked to assign them marks on the scale of 1-5 based on their understanding of the problem and responses to each question. The overall performance was evaluated by the Instructor and TA. While evaluating a team performance, we also considered the marking scale given by other teams.

5 DATA ANALYSIS AND RESULTS

Table 2 shows the list of survey questions and students' opinions about the stated CBL learning outcomes, in terms of Strongly Agree (SA), Agree (A), Disagree (DA), and Strongly Disagree (SD). Kundra et al. [5] and Saini et al. [11] utilized the same questions for assessing the efficacy of CBL. From Table 2, it can be inferred that 99.4% of total number of students agreed (SA + A) that the case allowed for a deeper understanding of case concepts (Q2). Another observation is, 8.1% students don't think that the case discussion strengthened their communication skills to speak in front of the audience (Q9). A significant 98.8% students agreed that the cases were relevant to their course structure (Q1). Similarly 98.1% students felt that they were more engaged in the class during the CBL sessions (Q7). These strong agreements give a positive indication about the CBL exercises.

In order to investigate the impact of dependent variable "students' response" on independent variable "case" we re-write the information of Table 2 in the form of Table 3. We state the null hypothesis as "there is no significant difference between the responses for Case A and Case B". Here after performing the t-test for agree%, we get a t-value of 0.2863 and p-value of 0.388959. The result is not significant at $p<0.05$. Therefore we accept the null hypothesis. Based on this result, we conclude that the two cases were equally effective and served toward the learning objectives.

Table 2: Survey questions grouped by the respective learning principles [SA: Strongly Agree, A: Agree, DA: Disagree, SD: Strongly Disagree, IE: Invalid/Blank Entry]

Teaching Objectives	Q.N	Questions	SA	A	DA	SD	IE
Learning	Q1	I feel the use of case was relevant in learning about course concepts.	107 (66.5%)	52 (32.3%)	1 (0.6%)	1 (0.6%)	0
Learning	Q2	The case allowed for a deeper understanding of course concepts.	81 (50.3%)	79 (49.1%)	0	1 (0.6%)	0
Learning	Q3	The case will help me to retain the different aspects of Software Engineering better.	81 (50.9%)	76 (47.8%)	1 (0.6%)	1 (0.6%)	2
Critical Thinking	Q4	The case allowed me to view an issue from multiple perspectives.	81 (50.6%)	74 (46.3%)	3 (1.9%)	2 (1.2%)	1
Critical Thinking	Q5	The case was helpful in synthesizing ideas and information presented in course.	71 (44.4%)	85 (52.8%)	4 (2.5%)	1 (0.6%)	0
Critical Thinking	Q6	The case added a lot of realism to class.	89 (55.3%)	67 (41.6%)	3 (1.9%)	2 (1.2%)	0
Engagement	Q7	I was more engaged in class when using the case.	84 (53.2%)	71 (44.9%)	2 (1.3%)	1 (0.6%)	3
Engagement	Q8	The case discussion increased my interests in learning about Software Engineering.	73 (45.6%)	83 (51.9%)	3 (1.9%)	1 (0.6%)	1
Communication Skills	Q9	The case discussion strengthened my communication skills to speak in front of the audience.	65 (40.6%)	82 (51.2%)	12 (7.5%)	1 (0.6%)	1
Team Work	Q10	The case discussion increased my confidence to work in a team.	82 (51.2%)	69 (41.3%)	7 (4.4%)	2 (1.2%)	1

Table 3: Percentage of SA, A, D, SD for the 10 questions sliced by Case (to investigate if case influences the satisfaction level)

	Case A (Google)					Case B (StalwartX)				
	SA%	A%	DA%	SD%	IE%	SA%	A%	DA%	SD%	IE%
Q1	76.5	22.2	0.0	1.2	0	42.5	56.3	1.3	0.0	0
Q2	51.9	46.9	0.0	1.2	0	46.3	53.8	0.0	0.0	0
Q3	54.3	44.3	0.0	1.2	0	51.3	47.4	1.3	0.0	2
Q4	51.3	43.8	3.8	1.3	1	41.3	57.5	0.0	1.3	0
Q5	58.0	39.5	1.2	1.2	0	48.8	47.5	3.8	0.0	0
Q6	53.1	43.2	1.2	2.5	0	57.5	40.0	2.5	0.0	0
Q7	55.0	42.5	1.3	1.3	1	47.4	51.3	1.3	0.0	2
Q8	53.8	42.5	2.5	1.3	1	50.0	48.8	1.3	0.0	0
Q9	55.0	40.0	3.8	1.3	1	41.3	47.5	11.3	0.0	0
Q10	37.5	60.0	1.3	1.3	1	48.8	42.5	7.5	1.3	0

To understand whether there is a significant difference in responses from male and female students, we categorized the information of Table 2 into Table 4, and performed a t-test analysis. We state the null hypothesis as "there is no significant difference between the responses from male and female students". The t-value for agree% is 2.49775 and p-value is 0.011206. Hence result is significant at $p<0.05$ and we reject the null hypothesis. That means there is a significant difference in responses from male and female students. Since women in computing have been in minority and it is important for the society to have a more balanced and equal representation across gender (and even across ethnicity, race, nationality, etc.), we did this gender specific analysis to provide some useful insights in this regard [7][2].

Table 5 shows the questionnaire and respective answers from the two M.Tech TAs. When we asked about the overall experience with CBL the TA1 said "This is my first experience with CBL and I found it very interesting. I get a different perspective of the problem and enhance my thinking to explore new possibilities/solutions to the problem questions. I have clearly seen students' enthusiasm and interest in solving the problem using CBL approach. At the time of discussion, they are coming up with various alternatives and possibilities for a specific problem question".

Table 4: Percentage of SA, A, D, SD for the 10 questions sliced by Gender. NOTE: Overall class size is: 164 (51 Female, 113 Male) students. No. of students participated in the study: 161 (No. of female: 50, No. of male: 111)

	Male					Female				
	SA%	A%	DA%	SD%	IE%	SA%	A%	DA%	SD%	IE%
Q1	66.0	32.4	1.9	0	0	66.0	32.0	0.0	2.0	0
Q2	50.5	49.5	0.0	0	0	50.0	48.0	0.0	2.0	0
Q3	55.0	45.0	0.0	0	2	42.0	54.0	2.0	2.0	0
Q4	56.8	42.3	0.9	0	0	36.7	55.1	4.1	4.1	1
Q5	47.7	51.4	0.9	0	0	36.0	56.0	6.0	2.0	0
Q6	60.4	37.8	1.8	0	0	44.0	50.0	2.0	4.0	0
Q7	55.6	43.5	0.9	0	3	48.0	48.0	2.0	2.0	0
Q8	46.4	50.9	2.7	0	1	44.0	54.0	0.0	2.0	0
Q9	38.2	54.5	7.3	0	1	46.0	44.0	8.0	2.0	0
Q10	52.7	44.5	2.7	0	1	48.0	40.0	8.0	4.0	0

6 CHALLENGES AND RECOMMENDATIONS

Organizing CBL sessions regularly requires a lot of plannings and efforts. The cases need to be designed carefully and should not deviate from the scope of course structure. The students have only been gathering knowledge using the traditional teaching methodology. Hence, initially it was difficult to explain them the advantages of CBL approach. Moreover, selecting appropriate software testing topics to be covered by CBL sessions was a challenge. We tried to

Table 5: TAs Questionnaire and their responses

Questions	TA Response #1	TA Response #2
Q1.What prior experience did you have with case-based learning as a student and as a TA?	No, I don't have any experience with CBL as a student and as a TA.	No experience as a TA and some experience as a student
Q2. How much and what kind of preparation did you do before the case-based learning sessions?	Before the sessions, the case has been discussed with the instructor to understand the motivation of this exercise. I have also gone through the videos for conducting CBL.	I have gone through many case studies for the session.
Q3. Did you find imbalanced participation or a balanced participation among members of a group? On a scale of 1 to 5, where 1 is imbalance (not everyone equally participating) and 5 is balance.	5	3
Q4.Did you spend equal time on each group? On a scale of 1 to 5 in which 1 is unequal and 5 is equal.	4	2
Q5. Did you find facilitating or managing one group easier than other groups (some group dynamics may be poor while some may be good)? On a scale of 1 to 5 where 1 is hard and 5 is easy.	5	5
Q6. In your opinion, was the group size of 5-6 students appropriate or not? What should be the group size?	More. In my opinion, the group size should be 3-4.	Appropriate
Q7. Did you find the time allocated to each group for solving the case is sufficient? On a scale of 1 to 5 in which 1 is less and 5 is sufficient.	4 (If more time was given then it could have benefited more.)	5

divide the class into 4 equal halves and then conducted the exercise in 4 sessions. There is a possibility that the learning potential of students of these 4 halves differ, which might have affected the results. Managing large groups of students having different capabilities with the concepts was a big challenge. Since Session I and Session III (Table 1) were not conducted in parallel, therefore there is a possibility that students may have discussed the case problem with other groups prior to their own session. Hence it is required that for all sessions, we should use different cases, but of same difficulty-level for fair evaluation.

The authors suggest that the training of TAs before the CBL sessions is very important as this builds up a relationship of trust between TAs and instructors [12]. Moreover, TAs work at more individual student-level in comparison to the instructor, therefore their pre-preparation plays a vital role in the success of CBL execution.

7 CONCLUSION AND FUTURE WORK

We proposed case-based learning for software testing education with the aim to impart in-depth and practical knowledge through a self-learning environment. After conducting the CBL exercise, the students identified the advantages and limitations of various testing methods along with their applications in realistic problems. They approved the importance of CBL with an overall agreement of 96.78%. The statistical analysis revealed that there is no significant effect observed on students' responses with the change in case under execution. However, gender difference affected the results.

In future we plan to target additional topics related to Software Testing curriculum using CBL pedagogy. Moreover, it will be interesting to identify the topics of Software Testing course that are difficult to be exercised using CBL methodology. We also aim to conduct similar exercises for the students of other universities with more effective cases.

REFERENCES

[1] Benjamin S. Clegg, José Miguel Rojas, and Gordon Fraser. 2017. Teaching Software Testing Concepts Using a Mutation Testing Game *(ICSE-SEET '17)*. https://doi.org/10.1109/ICSE-SEET.2017.1

[2] W. DuBow, A. Kaminsky, and J. Weidler-Lewis. 2017. Multiple Factors Converge to Influence Women's Persistence in Computing: A Qualitative Analysis. *Computing in Science Engineering* (2017). https://doi.org/10.1109/MCSE.2017.42

[3] Kirti Garg, Ashish Sureka, and Vasudeva Varma. 2015. A Case Study on Teaching Software Engineering Concepts using a Case-Based Learning Environment.

[4] V. Garousi. 2011. Incorporating real-world industrial testing projects in software testing courses: Opportunities, challenges, and lessons learned. 396–400. https://doi.org/10.1109/CSEET.2011.5876112

[5] Divya Kundra and Ashish Sureka. 2016. An Experience Report on Teaching Compiler Design Concepts using Case-Based and Project-Based Learning Approaches. In *T4E*.

[6] Salma Mahgoub and Farrukh Nadeem. 2012. Interactive Case Based Learning in Teaching Decision Support Systems and Business Intelligence. ACM, Article 34. https://doi.org/10.1145/2379636.2379669

[7] J. B. Main and C. Schimpf. 2017. The Underrepresentation of Women in Computing Fields: A Synthesis of Literature Using a Life Course Perspective. *IEEE Transactions on Education* (2017).

[8] M. Otsuki and M. Samejima. 2013. An Intelligent Tutoring System for Case-Based E-Learning on Project Management. https://doi.org/10.1109/SMC.2013.592

[9] Pantelis M. Papadopoulos, Stavros N. Demetriadis, and Ioannis G. Stamelos. 2007. Case-based Instruction on the Web for Teaching Software Project Management. *SIGCSE Bull.* (2007). https://doi.org/10.1145/1269900.1268826

[10] P. V. Peplow. 1997. *Responses of Female and Male Medical Students to Case-Based Learning Programme in Anatomy.* Springer Netherlands, 118–120. https://doi.org/10.1007/978-94-011-4886-3_34

[11] Veena Saini, Paramvir Singh, and Ashish Sureka. 2017. SEABED: An Open-Source Software Engineering Case-Based Learning Database. In *Computer Software and Applications Conference(COMPSAC 2017).*

[12] G. Saunders-Smits, M. van den Bogaard, and Y. C. Chiang. 2009. Designing training sessions for TAs: experiences in Aerospace Engineering at Delft University of Technology. 1–6. https://doi.org/10.1109/FIE.2009.5350797

[13] A. Soska, J. Mottok, and C. Wolff. 2016. An experimental card game for software testing: Development, design and evaluation of a physical card game to deepen the knowledge of students in academic software testing education. https://doi.org/10.1109/EDUCON.2016.7474609

[14] Claes Wohlin, Per Runeson, Martin Hst, Magnus C. Ohlsson, Bjrn Regnell, and Anders Wessln. 2012. *Experimentation in Software Engineering.* Springer Publishing Company, Incorporated.