

# An Innovative Methodology For Software Testing Education In Higher Education Institutes Of Pakistan

Muzmil Memon; Rafia Naz Memon; Irfana Memon; Aijaz Ahmed Arain; Mukhtiar Memon

**Abstract:** Software testing is an important phase of development of high quality software. Despite its importance, students are not provided sufficient software testing education at university undergraduate and postgraduate level to start working as a software testing professional in companies. The aim of this study is to identify and analyze the different software testing education problems faced by students and teachers, and to propose a methodology to address the software testing education problems in higher education institutes of Pakistan. To validate the methodology, an online survey has been conducted from teachers of software testing course in higher education institutes of Pakistan, total 36 experts participated voluntarily in the study. From results, it was observed that software testing methodology can address many identified software testing education problems, its elements are fully relevant with software testing course contents, and are helpful and effective for students and teachers of software testing course. It can be concluded from the study that the software testing methodology should be implemented and followed in higher education institutes of Pakistan.

**Index Terms:** Software Testing Education, Software Testing Methodology, Software Testing Education Problems

## 1 INTRODUCTION

Software Testing is main phase for development of high quality software. Despite of vital role of software testing education, it is evident from literature that less attention has been paid to it and researchers still report a lot of problems in software testing education, such as the students taking software testing course have less/no industrial experience, students only focus on small projects rather than real projects, students mostly find software testing course hard to understand and boring due to occurrence of lot of bugs, maximum time consumption for little code, lack of low student practical skill, and lack of students grading to test their code [1, 3]. Therefore, there is need to develop and maintain software testing teaching/learning elements, tools and practices for better teaching and learning software testing education. To address the software testing education problems, this research provide a solution in the form of software testing methodology which is based on three main parts, that are teaching, learning and assessment with support of e-learning method, software testing education tools, software testing education games, software testing student grading tools for instructors, social interaction tools between a teacher and student.

## 2 LITERATURE REVIEW

### 2.1 Software Testing Education Current Trends

From literature, it is evident that software testing is considered an important course for the students of software engineering course [4]. The methods and contents of software testing keep changing. Teaching and learning of software testing course in higher education institute need constant improvement. The research studies related to software testing education have been done in different countries including Brazil, USA, Hong Kong, Ireland, China, Canada, Malaysia and Brazil and so on. Researchers observed that software testing education has been paid less attention as compared to other phases of software development. They also observed that the students of computer science and software engineering when taught software testing course, find it less important hence take less interest in it. Therefore there is a need to add some basic contents of software testing in preliminary courses of software engineering

discipline such as in introduction to computing course for better student software testing education concepts [1]. Researchers provided many solutions to address software testing education problems such as e-learning methods, case-based learning exercises for students, hybrid course based lecturer activity, CDIO based engineering framework for software testing education, software testing education tools, software testing education games, software testing student grading tools for instructors, social interaction and discussion tools between a teacher and student, but students still face many problems in learning the course.

### 2.2 Software Testing Education Problems

The researchers observed multiple problems during teaching software testing course such as lack of practical work, lack of sufficient time period to cover course, lack of software testing certification [3]. While researchers also reported many problems in learning the course such as mostly students taking software testing course have no industrial experience, they only focus on small projects rather than on real industry based projects, students found software testing course hard to understand and boring due to occurrence of lot of bugs, maximum time consumed for little code, lack of practical skills to test their own code [3] [2] [5]. For the teachers it is difficult to examine every student assessment, even to grade every student individually and to check student skills [5].

### 2.3 State of Art

Mauricio et. al. (2019) conducted a survey from students regarding software testing course difficulties, they proposed a software testing curriculum to improve the quality of software testing course and they found that students have improved their mistakes and learnt hard topics of software testing course through proposed software testing curriculum [20]. Gordon et. al. (2019) proposed a Code Defenders game for software testing course in undergraduate and graduate level of universities, they engaged regular course session on weekly basis with Code Defenders sessions. From results, they proved that Code Defenders sessions can address multiple student problems and can improve student progress. From students experience they observed that the Code Defenders was a positive learning activity and it improved

student performance as well [21]. Tiwari et al. (2018) examined a case based learning (CBL) exercise of software testing course as self-learning environment. They conducted a survey with testing methods and techniques; from evaluation they observed that 96% gave importance to CBL through statistical analysis however gender difference affected the results [6]. Natan et al. (2018) observed that teaching is a difficult task specifically teaching to undergraduate students of software Testing course, they conducted a survey through different Brazilian computing courses in different universities. From evaluation, they found out how topics of software testing are taught at universities, teaching practices and challenges during evaluation of software testing learners. They showed that in academics there is need of more software testing education elements and knowledge for software testing teachers and learners in universities of Brazilian [7]. Thiago et al. (2018) provided software testing education card game which improved student software testing skills, they observed that software testing is an important approach to update and maintain software quality, they examined that teaching in software testing is followed as more theoretical rather than to make it practical side. They provided software testing learning card game GreaTest with a single player and allow participant to play as test analyst. A game was evaluated from teachers, undergraduate and post graduate students through survey and they found it helpful and important for student skills even more students are interested to play software testing learning game GreaTest [8]. Lauvas et al. (2018) observed that in universities software testing course is paid less attention while teaching important topics of computer science and software engineering education. They suggested that software testing education game, real project tools and industry based tools are best motivated learning tools for students and shall help to address issues of software testing education [9]. Sun et al. (2012) presented a hybrid course which is based on an online lecturer activity and lab programming practice by using different assessment methods. They proved that web-based education can give more advantage and more easy as compared to traditional classes. They examined that students in the hybrid section and in traditional class have no difference, however students can get technology based advantage using web-based section such as hybrid section, they examined that student can also get more computer knowledge by using this tool. From evaluation they found out that hybrid design in software testing course is more accepted by students because hybrid web-based design software testing course contains all students' learning activities and learning material such as lecture videos and assessment methods [10]. Edwards (2003) proposed a method and tool for teaching software testing. According to him, teachers are always busy due to overburden work, therefore students should be taught using automated tools such as Test- Driven Development (TDD) and Web-Cat tools for software testing course [2]. Souza et al. (2011) proposed an automatic tool named ProgTest which is based on testing activities, they proved that ProgTest tool is beneficial for teaching software testing course and for evaluation of student programming and testing assignments. They proposed other tools named SAC web-based tool and a generic tool named OTO. They conducted a survey from students and teachers in academics to validate their tool. From survey results they observed that through the help of ProgTest tool teacher can observe student difficulties during

evaluation of student code and instructor can easily improve their methods. It is observed through students survey that ProgTest tool is helpful for student code testing and programming even students can improve their skills as well [12]. Bin & Shiming (2014) introduced a model named CDIO that was proved a very good platform for teaching software testing course [13]. Dhiauddin et al. (2014) proposed a design of software testing program at postgraduate studies for those who want to specialize in software testing. They highlighted that in the world of software development, developers developed high quality softwares for human problems, therefore industry required skillful software tester and still there is method or tool required to improve software testing education at the universities especially at postgraduate studies [15]. Valle et al. (2015) identified student problems such as students having lower knowledge about testing their code, software testing professionals still facing salary issues in software industries and students have lowest interest due to more errors occurred during testing. They proposed a solution as software testing education games such as iTest learning, Jogo das 7 Falhas, JETS, U-TEST and ILearnTest [1]. Astigarraga et al. (2010) observed that there is no professional software testing certifications available at various academic studies. They proposed that student can use open source software testing tool and software bug tracking systems for their better practices, testing skills and finding of bugs during testing of code. They proposed a best teaching method for software testing education named bug reporting method [11].

### 3 RESEARCH METHODOLOGY

As In this research, software testing education problems have been identified through literature survey, then a solution has been proposed to address the identified software testing education problems. To validate the solution, an online questionnaire survey has been conducted which is divided into three parts, first part contain questions about importance of software testing elements, material and methods of software testing methodology, second part contain questions about effectiveness of the software testing methodology, also it contains questions about elements which are helpful to students and relevant material and methods of software testing course, third part contain questions about implementation ratio of software testing methodology, lecturer activity, student progress and suggestions for improving software testing methodology. The survey data has been collected online from the teachers of higher education institutes of Pakistan using Google docs. The targeted participants of this study were the professors, and lecturers who have experience in teaching software testing course as they can be considered as software testing experts because of their experience and knowledge in software testing course and it was expected that they can better respond to the questionnaire. During survey more than 60 teachers were invited for evaluation of this study through emails and personal requests, however only 36 teachers participated voluntarily in the research study in response to invitation.

#### 3.1 Identified Software Testing Problems

Identified software testing education problems have categorized into teaching, learning and assessment problems.

1) Teaching Problems: Teacher need more time to

evaluate student testing skills and code, however there is short time available for teacher to evaluate student skills and conduct classes [2]. Sometime teachers may want to share extra knowl- edge and material but due to limited time slots, teachers can cover only important course contents, also student's practical skills are not examined through theory exams in universities [13]. In academics there is no professional software testing trainings contents adopted in software testing course syllabus even there is no software testing certifications followed in soft- ware testing course syllabus in higher education institutes [15]. Figure 1 shows identified teaching problems of software testing education.

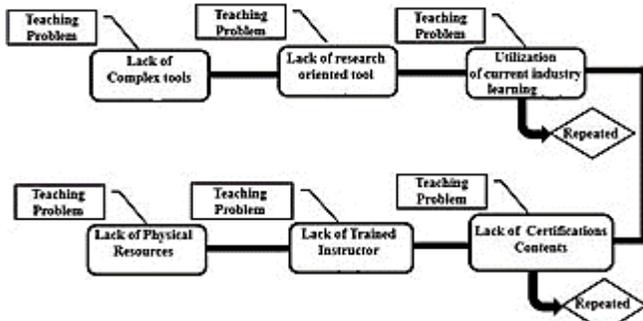


Fig. 1. Teaching problems of Software Testing education

2) Learning Problems: In higher education institutes students have low practical skills to test their own code even student have difficulties to check and test their own programming code by using any easy to use software testing tool [2]. There are many issues with software tester salaries because software engineers get hired directly from universities but software testers do not. After proper software testing training through software houses they can be hired as software tester and during training period they usually get low salary and benefits as compared to software engineers or developers, in result students normally take low interest in software testing education [1, 2]. When students of computer science and software engineering are taught software testing course, they find it less important and take less interest in it. Figure 2 shows identified learning problems of software testing education.

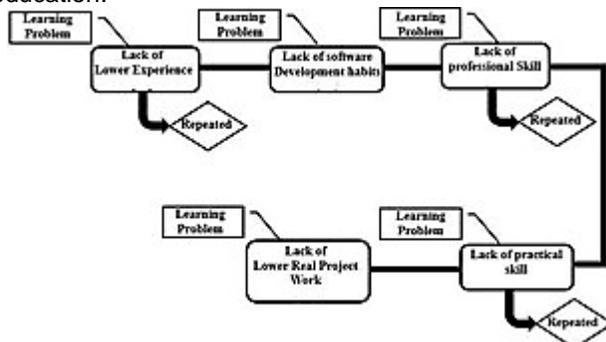


Fig. 2. Learning Problems of software testing education

3) Assessment and General Problems: Assessment is an important activity as it is very essential for teachers to evaluate students testing performance and skills of students. Due to more bugs and more code students usually get unmotivated to test their own code and take less interest in software testing course, they are left with less experience to find their errors [2]. Students are interested to see their detail error report with

suggestions; students need user- friendly tools to test their own code [18, 19]. Figure 3 shows identified software testing education assessment and general problems.

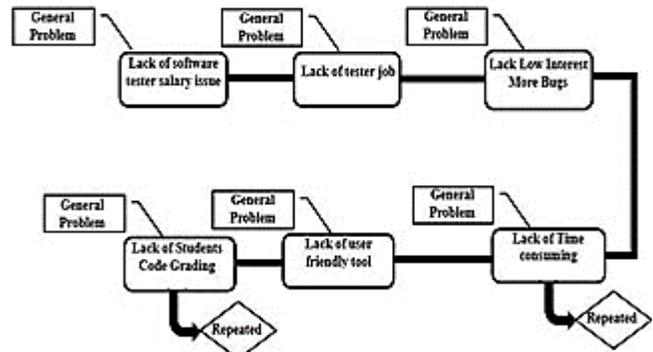


Fig. 3. Software Testing Assessment/General Problem

In above Figure 1, 2, 3 the word repeated is used because all these problems are mostly identified in different research studies and are faced by the academics during teaching and learning software testing education.

3.2 Proposed Software Testing Methodology

An innovative software testing methodology has been proposed in this research which provide teaching material, teaching method based on e-learning concepts, learning and assessment methods including software testing certification contents learning, online software testing assessment tools for students grading and online discussion course based blogs which a range of solutions to different software testing education problems [16, 17]. Figure 4 shows the proposed software testing methodology which is based on three main parts that are teaching, learning and assessment.

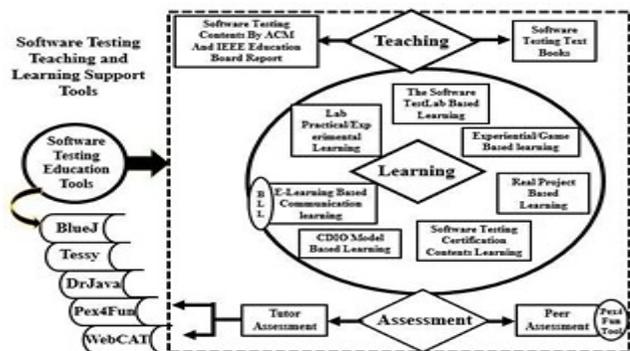


Fig. 4. Software Testing Methodology

a) Teaching: To address the teacher's problems in software testing course at higher education institutes, this part has been included in the methodology for instructors in relation with teaching and learning material, software testing education support tools, online assessment and social interaction tools. Teaching is based on following three subparts..

b) Software Testing Contents by ACM and IEEE Education Board: According to ACM and IEEE report, it is proposed that software testing course should be planed according to the software testing contents proposed by ACM and IEEE report and shown in Table 1 [23]. It is suggest that ACM and IEEE software testing contents should be

implemented in software testing course syllabus in higher education institutes.

**TABLE 1**  
**SOFTWARE TESTING CONTENTS BY ACM AND**  
**IEEE REPORT [23]**

S. No	Knowledge Unit
1	Basic definitions, concepts, and relationships among quality, quality assurance (product and process), and testing.
2	Introductory definitions and concepts of different testing techniques (for non-executables and executables), test process, and levels of testing
3	Basic set theory and propositional calculus for testing
4	General Concept of Black-Box (functional testing techniques) and Boundary-Value/Robustness testing
5	Equivalence class based testing technique
6	Decision-table based testing technique
7	Basic graph theory, paths, and adjacency matrix for testing
8	General Concept of White-box (structural testing technique) and various paths-based coverage testing techniques, including Basis testing and Cyclomatic complexity number
9	Dataflow testing
10	Slice-based testing
11	Evaluation of and metrics for relationship of gaps and redundancies among the different Structural Testing techniques
12	Test planning, test metrics, and test status tracking process and techniques
13	Test Execution Processes, Levels of Testing and Control, and Configuration Management
14	Integration testing techniques (top down, bottom-up, neighborhood, MM-path, etc.) and metrics
15	Systems and Regression testing techniques using threads and operational profile; relationship to customer "acceptance" test
16	Interaction testing and modelling techniques using petri-net, state machine, decision tables, object oriented classes, etc.

**c)** Software testing text books: Three text books are added in software testing methodology in teaching part, each book has different contents and it is suggested that they should be referred in software testing course. Table 2 shows the book title, author and content summary of three individual software testing education text books.

**TABLE 2**  
**SOFTWARE TESTING BOOK CONTENTS SUMMARY**  
**[24, 25]**

Book Name and Author	Book Contents
Book-1: S. Naik and P. Tripathy, Software Testing and Quality Assurance: Theory and Practice: John Wiley & Sons, 2008 [13]	Contain chapters about maturity models test
Book-2: A. P. Mathur, Foundations	Contains practical exercises &

of Software Testing: Pearson Education, 2008 [14]	preparing students for examination
Book-3: P. Ammann and J. Offutt, Introduction to Software Testing: Cambridge University Press, 2008 [15].	Contain chapters for testing tools

**d)** Software Testing Education Tools: Experience is most important when performing software testing and student at preliminary level have low practical skills and experience. When software tester completes their code and removes errors in correct manner, a new code error free code is produced [26]. Therefore, students need software testing education based learning tool for improving their practical skills, experience and need to remove errors easily through the tool. Following are considered useful software testing education tools and hence included in methodology. BlueJ: BlueJ is software testing education tool which is specifically developed for teaching software testing at an introductory level [27]. The back end code is automatically created by the tool. Through BlueJ tool students can create different objects directly without writing individual program. Tessa: In this tool student can perform all basic tasks of testing, tessa provides learning support during testing such as covering the test life cycle, test generation, test execution and observation of code [27]. Pex4fun: In this tool student can write their code from any browser with intellisense and code can be run and analyzed through cloud [28]. It is a teacher/student interactive tool where handouts, slide, demos and knowledge through detail discussion provided to students and assessment of students are performed as well [6]. It works on web platform and run on an android phone, it also contains editor like Microsoft visual studio code editor [29]. Web-Cat: Assessment is an important for teaching and learning part and assessment play an essential role for helping instructors to evaluate student progress. Web-Cat is an interface between teacher and student through web browser [6]. An interface for students where feedback is available which contains certain error details, suggestions with highlighted code line numbers [19, 34]. Web-Cat is a best tool for evaluation of student's assignment while it automatically recognize errors in student's assignment code and suggest corrections, Web-Cat shows current and past student results card and student assignment summary report with student name, partner name and total score. Summary card shows previous files, remarks and deductions of student testing assignment marks.

i) Learning: There are many identified software testing learning problems [32, 19, 27], therefore, learning activity has been included in software testing methodology for students which includes seven student learning methods.

ii) Lab practical and experimental learning: Researchers found 5 labs for better student practical of software testing education at initial level. Table 3 shows 5 labs of software testing education.

**TABLE 3**  
**FIVE PRACTICAL LABS OF SOFTWARE TESTING**  
**COURSE [34]**

S. No	Labs Name
Lab 1	Introduction to testing and defect tracking
Lab 2	Requirements-based test generation

Lab 3	Code coverage: adequacy criteria and test case correction
Lab 4	Functional GUI testing
Lab 5	Mutation testing and analysis

iii)E-Learning Based Communication Learning: In this learning method, students can discuss project work, course material, and course related issues and easily connect with other students and teachers [35]. In this method, Pex4Fun tool is used for e-learning based communication [14, 18].

#### iv)CDIO Model based Software Testing Learning:

Researcher proposed a CDIO philosophy, CDIO model or CDIO engineering educational framework for better teaching/learning software testing course. CDIO stands for Conceive, Design, Implement and Operate. According to this model the software testing course contents should be manipulated by following percentage

- 1) Theory knowledge 35%
- 2) Experiment 15%
- 3) Project 50%

According to CDIO philosophy, there is a need to develop occupation experience center laboratory for students of software testing which is different from normal laboratory because professional experience center is fully focused on student-centered thinking and user-friendly environment. Even occupation experience center should be established not only for students but for professional experience as well, this will improve student's team work [13].

vi)Software Testing Certification Contents Learning: Software testing certification programs are unknown to various academics [35]. Therefore, researchers suggested that a syllabus of software testing course should be designed with software testing professional certification contents. Following are software testing certifications for software tester.

- 1) Certification organization including international institute of software testing (IIST)
- 2) Certified tester foundation level syllabus

vii)Real Project Based Learning: Through real project based learning, student having opportunity to improve their in-depth skills into a real project based work and students can perform practice on real environment based projects. Through real project based work learning, students know their weak points and students can work on real decisions and students can deploy their practical as well as academics experience and practices [36]. Students can take their own personnel individual decisions to complete their real project tasks [29].

viii)Experimental and Game Based Learning: Experimental learning methods provide students with organizational experience in software testing course while students get opportunity to improve their mistakes through experimental learning [37]. Learning through games provide better course contents' concept with interest, fun and motivation environments. Through educational games students can improve their knowledge and skills. The software testing education games for students' learning are presented in Table 4.

**TABLE 4**  
**SOFTWARE TESTING EDUCATION GAMES [1]**

S. No	Software Testing Education Games
-------	----------------------------------

1	iTest Learning
2	Jogo das 7 F alhas
3	JETS
4	U-TEST
5	ILearnTest

iX)The Software Testlab Based Learning: TestLab is student learning platform for software testing course where research papers, tutorials, experience reports, projects and different publications and sponsors are available for students. It provides the knowledge and services shown in Table 5. Table 5

**TESTLAB COMPETENCY CATEGORIES [38]**

Test Practitioner	Perform a defined test within an established test environment, and document results.
Test Analyst	Handle front-end of testing process. Determine testing needs. Assure that the specification is an adequate basis for starting the testing process
Test Designer	Given a specification: design test cases using published and TestLab techniques
Test Builder	Construct machinery needed to run the test cases
Test Inspector	Verify that a testing task has been performed correctly according to TestLab procedures and standards
Test Environmentalist	Set up and support the test environment
Test Specialist	Perform a series of testing tasks spanning multiple phases of the test life cycle

X)Assessment: For the teachers it is difficult to examine every student assessment, even to grade every student individually and to check student skills [8]. Therefore, through assessment part in software testing methodology, a solution has been provided to teachers and students through online assessment tools and peer assessment method where teacher can examine and evaluate every student skills and progress easily while students can see their detail error mistakes report, code corrections, code suggestions and bugs through report generation task using online software testing assessment tools.

Xi)Tutor Assessment: A general form of assessment at higher education institutes which consists of following two parts.

Xii)Tutor Assessment based on Online Automatic Grading System Tool: Managing large assignments for assessment of students is a difficult and time-consuming task for instructor. Therefore, through tutor assessment based on online automated grading system tool, all tasks can be done with teacher feedback [43]. Web-Cat and Pex4Fun tool are best automated grading system tools for online grading of student assignments and assessment.

Xiii)Tutor Assessment based on Manual system: Tutor Assessment is supervised by lecturer who examine the student tasks and documentation group wise. According to

tutor assessment method there should be group discussion environment where student groups can discuss their problems with solutions and lecturer can observe the strengths and weaknesses of their solutions and in the end teacher can provide feedback which lies on performance of students.

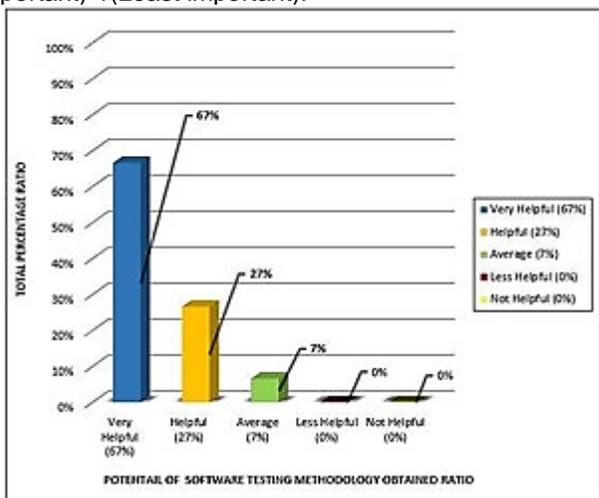
(iX)Peer Assessment: It is a process based on student groups where different grading performed by peers. In this method, different groups are created where students write their code which is compared with lecturer hidden code to assess the accuracy of peer assessment for grading [14].

**3.3 VALIDATION OF PROPOSED METHODOLOGY**

The proposed software testing methodology has been validated by teachers teaching software testing course in undergraduate and postgraduate studies through questionnaire survey. The validation results are presented in subsection below.

**A. Importance of software testing teaching, learning and assessment material and methods**

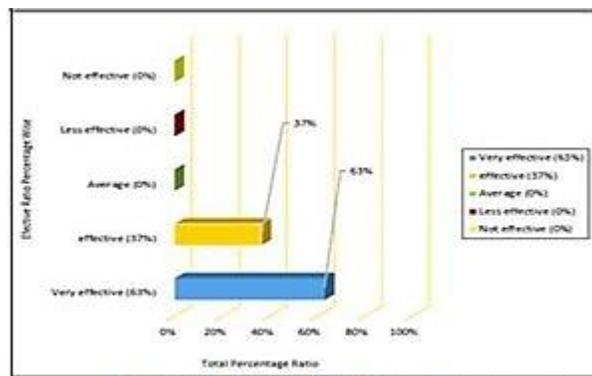
Figure 5 shows participant’s response about the importance of the elements related to software testing teaching, learning and assessment material and method used in the software testing methodology with the scale 5 (Most important)-1(Least important).



**Fig. 5.** The Ratings on Importance of Software Testing Teaching, Learning and Assessment Material and Methods

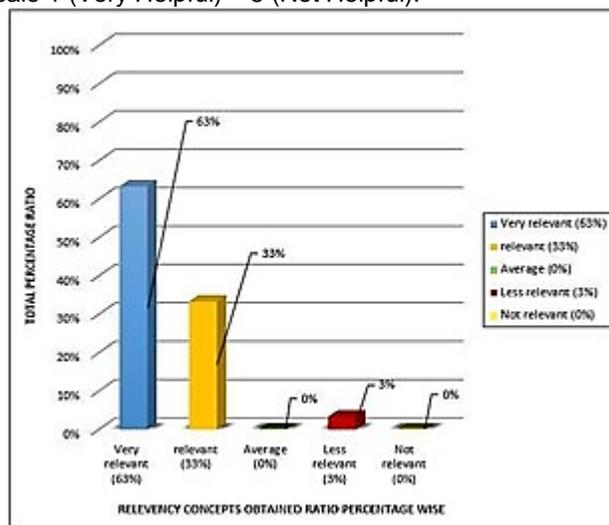
**B. Effectiveness, Potential and Relevancy of Concepts Used in Methodology**

Figure 6 shows participant’s responses to rate the effectiveness of the software testing methodology with the scale 1 (Very effective) – 5 (Not effective).



**Fig. 6.** The Ratings on Effectiveness of Software Testing Methodology

Figure 7 shows participant responses if the teaching, learning and assessment material and methods used in software testing course can help in addressing software testing education problems and software testing concepts with the scale 1 (Very Helpful) – 5 (Not Helpful).



**Fig. 7.** The Ratings of Potential of Software Testing Methodology to Help Students in Software Testing Concepts and to Address Software Testing Education Problems

Figure 8 shows participant responses about the concepts used in the software testing methodology are relevant to software testing course with the scale 1 (Very relevant) – 5 (Not relevant).

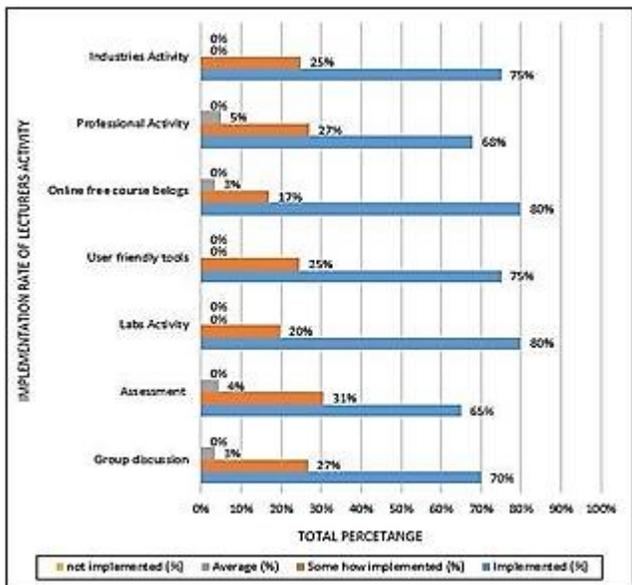


Fig. 8. The Ratings of Relevancy of Concepts Used in the Software Testing Methodology

C. Implementation ratio and ratings of software testing methodology and its elements

In Figure 9 shows participant responses to rate the implementation ratio (if software testing methodology implemented) with the scale 1 (Must be implemented – 5 (Not implemented)).

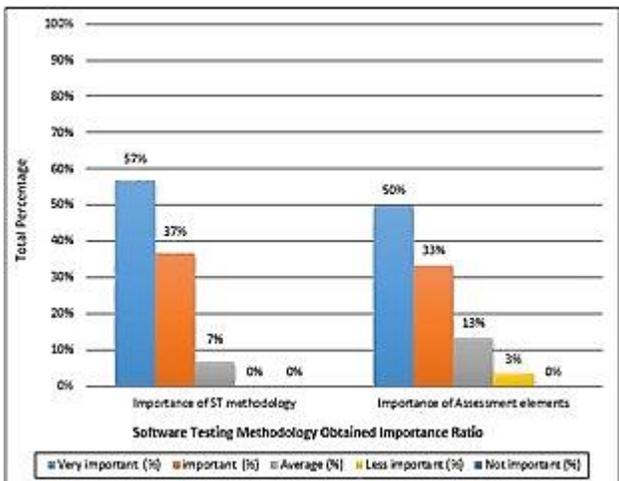


Fig. 9. The Ratings of Implementation of Software Testing Teaching, Learning and Assessment Elements

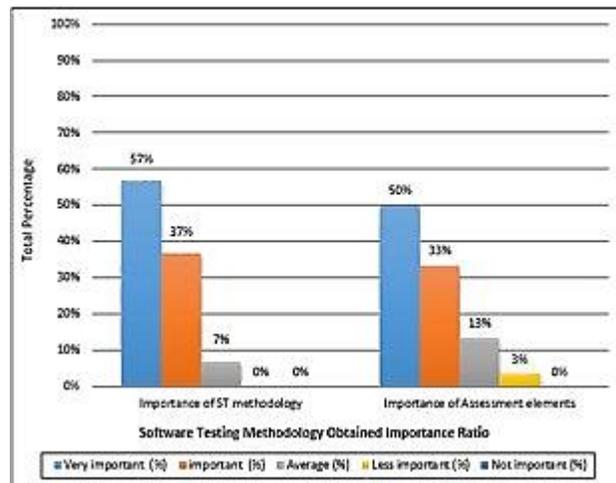


Fig. 10. The Ratings of Implementation of Software Testing Teachers Activity

Figure 11 shows participant responses to rate the importance for evaluation of students' progress with the scale 1 (Very important– 5 (Less important)).

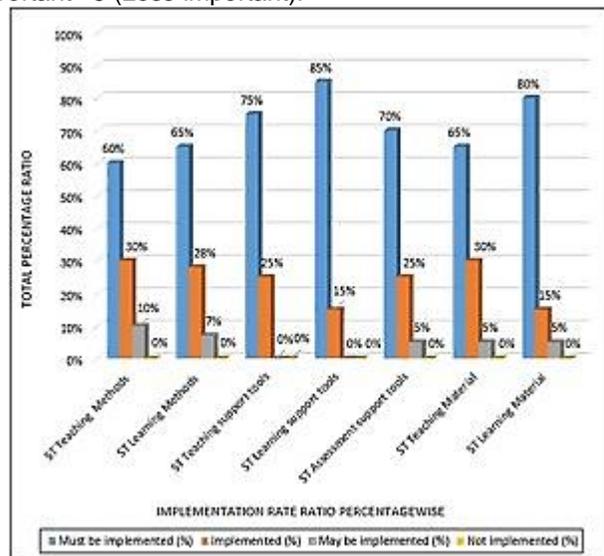
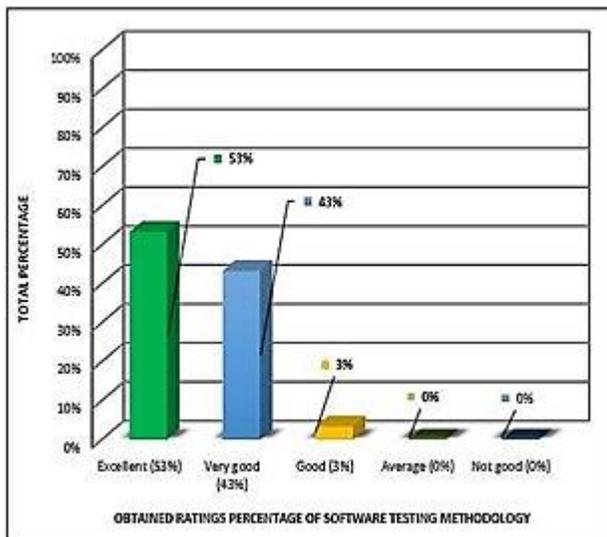


Fig. 11. Important Rate of Software Testing Methodology Learning and Assessment Evaluation Tools for Student Performance and Evaluation

Figure 10 shows participant responses to rate the implementation ratio of lecturers activity whether software testing methodology implements all lecturer activities with the scale1 (Must be implemented – 5 (Not implemented)).

Figure 12 shows participant responses to rate a software testing methodology and gave suggestions for improving the software testing methodology with the scale 1 (Excellent– 5 (Not good)).



**Fig. 12.** Rating of Software Testing Methodology

From evaluation from experts, it was observed that they found proposed software testing methodology helpful and effective for students and teachers. They validated that proposed software testing methodology should be implemented in software testing course in higher education institutes of Pakistan. They also validated that proposed software testing methodology can address the identified software testing education problems.

#### 4 RESEARCH LIMITATIONS

In this research study all elements of software testing methodology including teaching, learning and assessment are validated through online questionnaire survey from teachers however it can be further evaluated from students to validate the all learning elements of software testing methodology. Therefore this research is limited to only teacher's validation and evaluation of all elements of software testing methodology through survey responses.

#### 5 THREADS TO VALIDITY

To ensure results of this research, questionnaire survey conducted from experts included lecturers and researchers of various levels of experiences, qualifications and backgrounds of software testing education in higher education institutes of Pakistan. Clearly, an important limitation of the evaluation study involves the small sample size (only 36) and the relatively population (lecturers and researchers from Pakistan Country). This severely limits the external validity of this study. The goal of this evaluation is to validate the software testing education methodology for teaching, learning and assessment. Because of the software testing contents elements is being taught using syllabus topics and students and teacher face similar problems in teaching and learning software testing concepts. Therefore the results can be generalized.

#### 6 CONCLUSION

The aim of this study is to identify and analyze the different software testing education problems faced by students and teachers. Therefore this research provides a solution to address software testing education problems through software testing methodology which is based on three main parts that

are teaching, learning and assessment which provide a motivational platform to students such as use of tools for software testing education, software testing education games including gaming elements for teaching and learning software testing concepts, evaluation of student code in higher education institutes through an online grading software testing tools, open source learning projects, place students in team work, engage students in creative techniques, use of virtual environment, teachers may develop web-based online courses, assessments and practices for students by using web-based learning tool, e-learning method, the software TestLab based learning and software testing certifications contents learning. The proposed software testing methodology was evaluated from teachers of software testing education through an online questionnaire survey. From questionnaire results it was observed that software testing methodology can address identified software testing education problems, the elements of methodology are relevant with software testing course contents, software testing methodology elements are helpful and effective for students and teacher of software testing course in higher education institutes of Pakistan. The experts suggested that a software testing methodology should be implemented as a part of software testing course in higher education institutes of Pakistan.

#### 7 Future work

In this study, the significance of software testing Education has been explored in academics; and in this research results from teacher responses and feedback has been declared through online questionnaire survey which is conducted from only teachers of higher education institutes to validate the software testing methodology however in future same results can be conducted from students as well to ensure and confirm learning elements of software testing methodology from students. However software testing case study can also be added into software testing methodology.

#### REFERENCES

- [1] P. H. D. Valle, E. F. Barbosa, and J. C. Maldonado, CS curricula of the most relevant universities in Brazil and abroad: Perspective of software testing education., 2015.
- [2] S. H. Edwards, Teaching software testing: automatic grading meets test- first coding., 2003, vol. 26, no. 30.
- [3] F. T. Chan, T. H. Tse, W. H. Tang, and T. Y. Chen, Software testing education and training in Hong Kong., 2005.
- [4] K. Holl and V. Vieira, Focused quality assurance of mobile applications: Evaluation of a failure pattern classification., 2015.
- [5] F. Büchner, "Test Case Design." 2009.
- [6] S. Tiwari, V. Saini, P. Singh, and A. Sureka, "A case study on the application of case-based learning in software testing." ACM, 2018, p. 11.
- [7] L. N. Paschoal and S. D. R. S. Souza, "A," vol. 334. ACM, 2018, p. 343.
- [8] "Beppe, Thiago A., Ítalo Linhares de Araújo, Bruno Sabóia Aragão, Ismayle de Sousa Santos, Davi Ximenes, and Rossana M. Castro Andrade. "GreaTest: a card game to motivate the software testing learning." In Proceedings of the XXXII Brazilian Symposium on Software Engineering, pp. 298-307. ACM, 2018." to.

- [9] P. L. Jr and A. Arcuri, Recent Trends in Software Testing Education: A Systematic Literature Review., 2018.
- [10] L. Sun and M. Kindy, "Caroline Cecile Marcelle Liron, Christopher Grant, and Shirley Waterhouse. "Hybrid course design: Leading a new direction in learning programming languages."", 2012.
- [11] T. Astigarraga, E. M. Dow, C. Lara, R. Prewitt, and M. R. Ward, The emerging role of software testing in curricula., 2010.
- [12] Souza, D. M., B. H. Oliveira, J. C. Maldonado, S. R. Souza, and E. F. Barbosa, "Towards the use of an automatic assessment system in the teaching of software testing." 2014, pp. 1–8.
- [13] Z. Bin and Z. Shiming, "Experiment teaching reform for software testing course based on CDIO." Computer Science & Education, pp. 488–491, 2014.
- [14] O. A. L. Lemos, F. C. Ferrari, F. F. Silveira, and A. Garcia, Experience report: Can software testing education lead to more reliable code?., 2015.
- [15] Suffian, Muhammad Dhiauddin Mohamed, Suhaimi Ibrahim, and Mo- hamed Redzuan Abdullah. "A proposal of postgraduate programme for software testing specialization.", 2014.
- [16] Shah, Anuj Ramesh. "Web-cat: A web-based center for automated testing." diss., Virginia: Tech, 2003.
- [17] S. Gupta and A. Gupta, "E-Assessment Tools for Programming Lan- guages: A Review." vol. 14, 2018, pp. 65–70.
- [18] D. E. Krutz, S. A. Malachowsky, and T. Reichlmayr, "Using a real world project in a software testing course." 2014, pp. 49–54.
- [19] S. Sheth, J. S. Bell, and G. E. Kaiser, "A gameful approach to teaching software design and software testing-assignments and quests." 2013.
- [20] M. Aniche, F. Hermans, and Deursen, "Pragmatic Software Testing Education." 2019.
- [21] "Gamifying a Software Testing Course with Code Defenders." 2019.
- [22] V. Borsotti, "Barriers to gender diversity in software development education: actionable insights from a danish case study." 2018, pp. 146– 152.
- [23] M. Ardis, D. Budgen, G. W. Hislop, J. Offutt, M. Sebern, and W. Visser, "SE 2014: "curriculum guidelines for undergraduate degree programs in software engineering", Computer, vol. 48, no. 11, pp. 106–109, 2015.
- [24] S. Krusche and A. Seitz, "ArTEMiS: An Automatic Assessment Man- agement System for Interactive Learning." Education. ACM, 2 2018.
- [25] J. C. Dunlap, ""Problem-based learning and self-efficacy: How a cap- stone course prepares students for a profession," Educational Technology Research and Development", 2005.
- [26] S. H. Edwards, "Using software testing to move students from trial-and- error to reflection-in-action," ACM SIGCSE Bulletin, vol. 36, no. 1, pp. 26–30, 2004.
- [27] —, Teaching software testing: automatic grading meets test-first coding, 2003.
- [28] J. Wegener, R. Pitschinetz, K. Grimm, and M. Grochtmann, "TESSY– Yet Another Computer Aided Software Testing Tool?" 1994.
- [29] N. Tillmann, J. D. Halleux, T. Xie, and J. Bishop, Pex4Fun: A web-based environment for educational gaming via automated test generation, 11 2013.
- [30] N. Tillmann, J. Halleux, and T. Xie, "Pex for fun: Engineering an automated testing tool for serious games in computer science," 2011.
- [31] A. R. Shah, "Web-cat: A web-based center for automated testing," 2003.
- [32] P. H. D. Valle, A. M. Toda, E. F. Barbosa, and J. C. Maldonado, Educational games: "A contribution to software testing education", 10 2017.
- [33] R. Boelens, M. Voet, and B. D. Wever, "The design of blended learning in response to student diversity in higher education: Instructors' views and use of differentiated instruction in blended learning," Computers & Education, 2018.
- [34] R. Boelens, S. V. Laer, B. D. Wever, and J. Elen, "Blended learning in adult education: towards a definition of blended learning," 2015.
- [35] V. Garousi and A. Mathur, Current state of the software testing education in North American academia and some recommendations for the new educators, 3 2010.
- [36] D. A. Kolb, Experiential learning: Experience as the source of learning and development. FT press, 2014.
- [37] J. Ravitz, N. Hixson, M. English, and J. Mergendoller, Using project based learning to teach 21st century skills: Findings from a statewide initiative, Vancouver, Canada (Vol, 4 2012.
- [38] E. L. Jones, Grading student programs-a software testing approach, 2001.
- [39] —, An experiential approach to incorporating software testing into the computer science curriculum. IEEE, 2001, vol. 2.
- [40] N. Falchikov, Peer feedback marking: Developing peer assessment, 1995, vol. 32, no. 2.