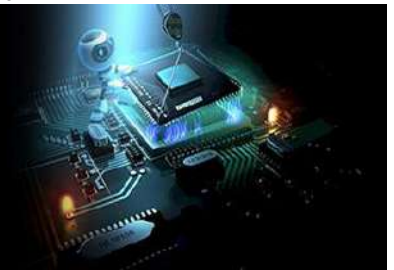


International Journal of Engineering in Computer Science



E-ISSN: 2663-3590
P-ISSN: 2663-3582
IJECS 2019; 1(1): 47-51
Received: 23-11-2018
Accepted: 25-12-2018

Mohammad Daud Haiderzai
Assistant Professor Software
Engineering Department
Nangarhar University,
Jalalabad, Afghanistan

Mohammad Ismail Khattab
Assistant Professor Software
Engineering Department
Nangarhar University,
Jalalabad, Afghanistan

How software testing impact the quality of software systems?

Mohammad Daud Haiderzai and Mohammad Ismail Khattab

DOI: <https://doi.org/10.33545/26633582.2019.v1.i1.a.14>

Abstract

Software testing course is one of the main and significant phenomenon of software development, whereas systematically testing the program is not the core concept of software engineering education. As a conclusion most of the software developers consider testing as liability, in addition to this one of the main reason that standard testing method seems to be boring and students do not show more interest to this field. This research offers the impact of software testing on quality of software systems which is a challenging demand for IT industries and software houses.

This paper focuses more on finding long term benefits of software systems with respect to different factors such as economic, social and personal dimensions, collaborative ability in developing and testing of engineering applications, practical approach of real-life application and software testing supplementing with new concepts, understand the testing and programming ability of students in the context of software testing and comparative study on tailor-made training program and generalized training program on software testing course for the development of software systems with high quality.

Keywords: software testing, phenomenon of software development, IT industries and software houses

1. Introduction

Software is a collection of programs, which runs on multiple platforms to perform different tasks and operations. as it is obvious fact that testing software is much more time consuming as well as budget consuming process, as per the studies done by various sources half of the overall cost of the software is constitute by testing process ^[15].

The term Software testing is a systematic approach attempting to find defects, bugs in a software system. The vigorous purpose of testing or software testing is to search for bugs in the software and report it ^[1]. Software testing is a significant block of the core software engineering main curriculum and it implicate students to have better hands-on implementation capability and strength. The concept of Software testing terminology needs a well-appointed technical expertise, competence, learning and novation susceptibility in order to develop high quality software systems ^[1].

As there is much needed demand for software testing experts in most of the software industries but on the other side death of professionals seen in this field, it seems that Universities and colleges offers very few courses related to this course (Software Testing) as this course needs more professional experienced tester to test the each and every step of software development life cycle process and to produce high quality applications and systems to the market ^[3].

Considering these situations we have conducted a literature review and research on the impact and teaching software testing course in undergraduate and post graduate level in order to come out with the solutions and challenges in the industries to develop software systems with high quality.

To achieve our goals we have formulated the following research questions as:

RQ1. What factor plays key role to produce software with high quality?

RQ2. What are the useful methodology in testing software systems?

Background

There is a worldwide awareness of realizing the importance of software testing. Many universities now offer a course in software testing for undergraduate and graduate studies in Computer Science and Engineering, Computer Applications, Information Technology,

Corresponding Author:
Mohammad Daud Haiderzai
Assistant Professor Software
Engineering Department
Nangarhar University,
Jalalabad, Afghanistan

Engineering ^[4]. Currently most of the industries experiencing huge shortages of software testing skills and experts, High education institutes should be fully aware of the situation and future trend, understand the requirements for software testing engineers ^[1].

2. Method

The protocol adopted in this review follows the guidelines for systematic literature reviews presented by Kitchenham and Charters ^[16] and Petticrew and Roberts ^[17].

A. Research Questions

The main objective of this research and review is to find out the answer for the impact of software testing on the development of good quality software systems, and the main focus will be on the following research questions related to testing impact on software systems.

RQ1. What factor plays key role to produce software with high quality?

RQ2. What are the useful methodology in testing software systems?

3. Systematic Literature Review Methodology

Systematic Review extracts the current research work in such way that can be assessed, analyzed and interpreted to produce a meaningful and vivid conclusion ^[14]. The SLR as compared to Surveys and literature reviews is more planned, better rigorous and thoroughly analyzed ^[14].

Kitchenham (2007) elaborated Systematic Review as: "A systematic literature review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question or topic area or phenomenon of interest".

4. Literature Review

Rasneet Kaur Chauhan and Iqbal Singh elaborated different testing techniques and tools and Some typical latest researches have been summarized such as unit testing, integration testing, system testing, acceptance testing, white box testing, black box testing and Grey box testing in addition to this some testing principles such as Start testing early, Test plan, test at different level and End of testing have been explore along with some testing tools such as: Ranorex, Rational Functional Tester (RFT) and Janova along with some latest research on software testing development such as: Test-driven development, Iterative and incremental testing, GUI automation test, Testability of component software, Embedded Software Simulation Test where all have direct impact in the development of software systems and application developments ^[5].

Wang *et al* stated in their paper huge competition and technology innovation have pushed software industries to find new ways to improve competitive advantages. They have designed Key Performance Indicator to test softwares and applications through web based learning system. However the employees do not know that Web-based learning system is so helpful to enhance their work performance. To come out with this problem they have offered a performance oriented concept where this approach utilizes performance measurement to clarify organizational goals and individual learning requirements as well as links them in applications of web-based learning ^[6].

Bhattacharjee, Neogi and Mahanti stated about the pair programming in teaching classes, where their studies were

implemented on both under graduate and post graduate students of different universities and colleges. Their findings are:

About 70% of the pair programmers can recognize the conditions of errors given to an unknown problem statement. Whereas the overall studies has confirmed that majority of the students were proud by the pair programming concepts and the average score was around 4.1 out of 5.0. Beside this another founding was that students were not well satisfied and equipped to test a program assignment and this lead toward a huge gap while they join software companies and industries ^[7].

Sneha and Gowda has presented different types of software testing techniques as well as software automation testing tools with the high impact on the development of high quality software systems, where it briefly discussed different types of software testing tools and techniques as well as the advantages of manual testing and automation testing, to wrap it up some deep studies has been carried out on various automated tools which used in different systems and platforms and finally the paper has been ended with the importance of automation testing over manual testing [8].

Chan FT, Tse WH and Chen TY have stated that most of the software companies developing software do not care about testing process sufficiently as a vital task. They have also mentioned that most of the universities curricula did not provide the graduates with the required knowledge in software testing. And it is found that half of the testing members have gained formal training in software testing, the training procedures varied from industry to industry, the surveyed companies in different sectors such as finance and insurance and banking provide special training in testing of their software to their staffs.

On the whole the mentioned research has also stated that the present software engineering curricula in universities and colleges should pay more attention on the coverage of software testing requirements and strengthen it further ^[9].

Towey D, Yueh Chen T & Chen TY they have elaborated more on teaching software testing course for undergraduate and post graduate students, where instructors have faced more uninterested students and unmotivated students to be a software tester or to select their profession as a software tester. To come out with this crucial problem they have come across with the solution and new approach called as Metamorphic testing where it deduct and ease the problems and specially Oracle problem as well as it is a suitable approach that provides diversity and creativity in the testers' approach. Need to mention that they have created a tutorial format of four parts, presentation of the fundamental terminology related to software testing where participant with a little or no knowledge of software engineering or testing become familiar so quickly. The second part is an interactive session to introduce the MT where instructors utilize examples and different algorithms to cooperate the participants apply and use MT. the third part is the engagement of participants in team work and group work using different programs and algorithms for implementing and applying MT skills to chunk of programs. Where the final stage is enables the participants to provide practical supervision on how to use MT in their work areas and companies to produce market required and quality software systems ^[10].

Joshi G & Desai P have clearly stated about a useful learning model called Spiral model, where the learners can

go through the previous and next topic in up and down through the curriculum to learn in a concrete manner.

To conclude, SLM approach is a fruitful teaching method that enables pupils to create and develop what is already familiar for them. Most important advantages of this approach is that it reduced the writing quality code to half ^[11].

Masuda S pointed out in his paper regarding the collaboration and cooperation between academia and IT firms from their perspective in Japan. It is also been mentioned that as Japan is an aging country and the number of workers are decreasing and the solution for this issue is offshoring like China. It is also been discussed regarding the comparison of Japanese Software firms with US IT firms. Where around 70% of Japanese software companies are entertained by outsourced software firms as compared 34% of the US software firms are grabbed by them, in regard to challenges of software quality and Testing huge number of incidents such as financial systems, air-line ticketing systems and some other important software systems effected by software failures and problems, keep this in mind the required software systems needs more and high quality than any other time, in the long run the a solution with three angles has been given as:

- a) Using of innovated and up to date technologies such as cloud services, social network service, smartphone service
- b) Applying the agile development process and the ISO/IEC/IEEE 29119 Software Testing Standards
- c) Educating the human resources for both the academia as well as industries ^[13].

According to Tangtang *et al.* (2011). They have gone through a design model called UML which is mostly used in the industries now a days, a formal approach has been used to analyze UML design models to achieve the formal test models, in addition to this they have also stated that the comparison of the model needs to reduced based on the unambiguous language as the test models which are described by the natural language cannot be considered it is therefore language for the exposition of the test model required to be solemn, in their paper they have offered a systematic and more particular method for the implementation of software testing based on this design model ^[18].

5. How Much Testing is enough?

We cannot complete a software before it is tested. In addition to this, it does not matter how many specifications we are adding in it or what kind of latest technology we are using is. If a particular software has not been tested properly and thoroughly, it may lead to failure of the software. The importance of software testing has increased in the past few years with the increase in complexity in the nature of the software and the introduction of new technologies ^[20].

The following six topics where each can affect the amount of testing required for software development:

- a. Methodology types
- b. Defining Requirements
- c. Methodology Maturity
- d. Project Staff Competency
- e. Project staff experience
- f. Configuration management controls

6. Software Development Methodologies

The software development methodology facilitates the way to build software systems. In the early days of computing, software project managers had two responsibilities:

1. to develop a process for building software
2. To follow that process to project completion. Because all project leaders need a software development methodology, standardized processes were developed. Some organizations create their own software development methodology, whereas others acquire the methodology for building software from suppliers ^[21].

7. Testing

There are different types of testing techniques are used in the development of software systems and we briefly go through some of the most used ones in software industries.

1. Functional testing. Scores highly on its ability to demonstrate that a code module does deliver intended functionality, though does rely heavily on interpolation from the actual test data used to 'all inputs'. It thus falls short on demonstrating the absence of unwanted side-effects. Testing can be highly automated/repeatable, and should generate evidence.
2. Structural testing. This technique aims to (partly) compensate for the shortfall just mentioned above. By demonstrating that 100% of stated structural elements in the code have been executed at least once it can be argued that there is a reduced possibility of unwanted side-effects. Reduced, yes, but eliminated, no.
3. Statistical testing. This technique endeavors to simulate the operational profile of the software as it would be subjected to in real life. Rather than a carefully selected set of test cases a representative set is used. This can of course mean a lot of data has to be prepared, input, and then the corresponding outputs checked. It is therefore not a technique for the faint-hearted, but can give a high subjective confidence in the reliability of that code, as well as providing the basis for a mathematical calculation of its reliability ^[23, 24].

8. Reuse in Testing

Software testing has basically five steps which include: Test Case Generation, Test case Selection, test Execution, Result Generation and Result Analysis. Reuse can be explored at any of the different stages of software testing

- a. Reuse in Test Case generation
- b. Reuse in Test Case Selection
- c. Reuse in Execution Environment
- d. Reuse in Report generation System
- e. Reuse in Report Analysis

Read the detailed description related to each step in ^[22].

9. Answer to Research Questions

Our research questions main objective is to find out the factors and useful methodology in testing and developing high quality software systems have been answered through the different reviews. The factors which plays key role in order to develop and produce high quality software are:

10. RQ1. What factor plays key role to produce software with high quality?

- Accessibility and Usability

- Adaptability and Extensibility
- Correctness and Functionality
- Readability and Maintainability
- Testability
- Portability
- Interoperability
- Security

To develop and produce high quality software systems the above factors must be considered in the life cycle of systems [25].

RQ2. What are the useful methodology in testing software systems?

The useful methodologies in testing software systems can be found in [26] and some well know are:

- Functional vs. Non-functional Testing
- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing
- Performance Testing
- Security Testing
- Usability Testing
- Compatibility Testing
- Testing With Test Complete

Process Quality leads to Product Quality

The quality of the development process significantly affects the value received by customers and development teams [27, 28]. The flow from specifications to process quality and then

to product quality can be understand through Figure 1. Hence, for high quality product it's important to concentrate more on development process as development of good quality software is usually an organizational effort, "something of higher quality has more value than of low quality". Without a process, software development is not possible.

Developers need some structure to follow, providing good understanding of overall requirements, high coupling between developers which leads to better, cheaper and faster delivery of quality products. Titansoft, one of the small scale software organization started adopting agile from 2015. In its Titan TPE branch all the projects were developed using the agile scrum method. Not only for software development, has the organization adopted the approach for doing its general purpose work also [29, 30]. This shows that not even small scale organizations are adopting agile methods for software development with limited resources available. But whether the quality has been maintained or not is a big question?

With the changing trends in software development, it still not reached with the level of producing software that is correct, easily usable, works reliably, maintainable, cost effective and on time [2]. Agile Software Development Process grown dramatically as a popular alternative for the development of software providing developers to streamline all the above factors in an efficient manner. Though the quality of the agile approach has been proved theoretically but not in a quantifiable manner [31, 32, 33, 34].

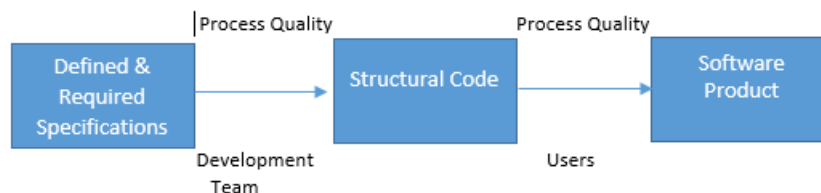


Fig 2: Process Quality leads to Product Quality

Conclusion

Developing a quality software product is an essential need for the software industry. Focusing on product's quality allows software users to adapt the product more easily and efficiently. Quality plays a vital role for the software users. It is a confirmation of all the requirements according to customer satisfaction. In our research paper we have gone through different reviews also discussed about the testing requirements, methodologies and its impact on the development of software systems.

In addition to this, the factors which directly impact the quality of software systems have been elaborated. Thus, testing concepts and types of testing has been discussed and finally the process quality that leads to product quality has been briefly described and discussed.

The paper presents an overall impact of testing on software development process for a quality product.

References

1. Yinnan Z, Xiaochi W. Implementation of Software Testing Course Based on CDIO. IEEE International Conference on Computer Science & Education. (ISSN 978-1-4244-9718-8). 2011; 11:107-110.
2. Lacerda LL, Felipe Furtado. Factors that help in the implantation of agile methods: A systematic mapping of the literature, In Proceedings of 13th Iberian Conference on Information Systems and Technologies (CISTI), 2018, 1-6.
3. Wang, *et al.* A Web-Based Learning System for Software Test Professionals. IEEE Transactions on Education. 2011; 54:263.
4. Singh Y. Software Testing. (1st Ed.). INDIA: Cambridge University Press, 2012.
5. Chauhan R, Singh I. Latest Research and Development on Software Testing Techniques and Tools, INPRESSCO, International Journal of Current Engineering and Technology, 2014.
6. Wang *et al.* A Web-Based Learning System for Software Test Professionals. IEEE Transactions on Education. 2011; 54:263-271.
7. Bhattacharjee V, Neogi Madhumita S, Mahanti R. Are our students prepared for testing based software development. 22nd Conference on Software Engineering Education and Training. 2009; 57(09):210-211.
8. Sneha K, Gowda M. Research on Software Testing Techniques and Software Automation Testing Tools. International Conference on Energy,

- Communication, Data Analytics and Soft Computing. 2017; 17(978):77-81.
9. Chan FT, Tse WH, Chen TY. Software Testing Education and Training in Hong Kong. Fifth International Conference on Quality Software. 2005; 5:77-81.
10. Towey D, Yueh Chen T, Chen TY. Teaching Software Testing Skills. IEEE International Conference on Teaching, Assessment and Learning for Engineering. 2015; 15:161-162.
11. Joshi G, Desai P. Building Software Testing Skills in undergraduate students using Spiral Model Approach. 8th IEEE International Conference on Technology for Education. 2016; 16:244-245.
12. Benjamin Clegg S, Rojas Miguel J, Fraser G. Teaching Software Testing Concepts Using a Mutation Testing Game. IEEE 39th International Conference on Software Engineering Education. (ISSN 978-1-5386-2671-9). 2017; 17:33-36.
13. Masuda S. Software Testing in Industry and Academia: A View of Both Sides in Japan. 10th IEEE International Conference on Software Testing, Verification and Validation Workshops, 17:40-41.
14. Ruchika M. Empirical Research in Software Engineering. (1st Ed.). US: CRC Press, 2016.
15. Wasif, *et al.* Software test process improvement approaches: A systematic literature review and an industrial case study. The Journal of Systems and Software, (ISSN 0164-1212). 2015, 1-2.
16. Kitchenham B, Charters S, Guidelines for performing systematic literature reviews in software engineering, version 2.3, School Comput. Sci. Math., Keele Univ., Keele, U.K., Rep. EBSE-2007-01, 2007.
17. Petticrew M, Roberts H. Systematic Reviews in the Social Sciences: A Practical Guide. Hoboken.
18. Tangtang, *et al.* A Study on the methods of Software testing based on the design models. The 6th International Conference on Computer Science and Education, (ISSN-1-4244-9718-8). 2011; 11:111-113.
19. Danial H, Paul S, Peter W. Teaching and Testing. IEEE, (ISSN-0-8186-7249-8), 1996, 248-258.
20. Naresh C. Software Testing Principles and Practices. (2nd Ed.) India: Oxford University Press, 2017.
21. William EP. Effective Methods for Software Testing. (3rd Ed.). India: Wiley India Edition, 2012.
22. Maryam A, Rosziati I. Improving Effectiveness of Testing Using Reusability Factor. IEEE International Conference Software Engineering. 2014; 110(9):2-4.
23. May Hughes, Lunn. Reliability estimation from appropriate testing of plant protection software, Software Engineering Journal. ISSN 0268-6961. 1995; 10(6):206-218.
24. Ian G. Justifying Software Testing in the 21st Century. Proceedings of the Testing: Academic & Industrial Conference-Practice and Research Techniques, IEEE. 2006; 6(12):3-4.
25. Testorigencom. Test Origen, 2018. Retrieved 9 October, 2019, from <https://www.testorigen.com/top-factors-that-influence-software-quality-management/>.
26. Smartbearcom. Smartbearcom, 2019. Retrieved 9 October, 2019, from <https://smartbear.com/learn/automated-testing/software-testing-methodologies/>
27. Isaias P, Issa T. High level models and methodologies for information systems, Springer, 2015.
28. Sujatha P, Sankar GV, Rao AS, Satyanarayana T. The Role of Software Verification and Validation in Software Development Process, IETE Technical Review. 2001; 18(1):23-26.
29. Lin Y, Titansoft. From chaos to control, from control to freedom-5 lessons learned on our agile journey, <https://www.slideshare.net/yveslin888/fromchaos-to-controlfrom-control-to-freedom-5-lessonslearned-on-our-agile-journey>, 2016.
30. Titansoft. Our history and timeline, <http://www.titansoft.com/en/About-Us/Our-History-Timeline>, 2016.
31. Bhasin S. Quality Assurance in Agile-A study towards achieving excellence, IEEE, 2012, 64-67.
32. Jain P, Ahuja L, Sharma A. Current State of the Research in Agile Quality Development, in Proceedings of the International Conference on Computing for Sustainable Global Development, 2016, 1877-1879.
33. Hsu HJ, Lin Y. How Agile Impacts a Software Corporation: An Empirical Study, In Proceedings of 42nd Annual Computer Software and Applications Conference (COMPSAC), 2018, 20-25.
34. Parita J, Arun SH, Laxmi A. The Impact of Agile Software Development Process on the Quality of Software Product. IEEE. 2018; 12(8):3-5.