

Issues, Challenges and Best Practices of Software Testing Activity

ZULKEFLI MANSOR

Research Center for Software Technology and Management
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
43000 Bangi, Selangor Darul Ehsan
MALAYSIA
kefflee@ukm.edu.my <http://www.ftsm.ukm.my/kefflee>

ENEBELI E.NDUDI

Faculty of Computer Science and Information Technology
Universiti Selangor
Jln Timur Tambahan, 45700 Bestari Jaya, Selangor Darul Ehsan
MALAYSIA
vickhuston@yahoo.com

Abstract: - Software testing activity is a huge challenge in software development project today due to end user needs. The end users need the project to be completed in short time with zero defect and high quality of product. Therefore, the testing activity should be started as earlier as possible because it will help in fixing enormous errors in early stages of software development and reduces the rework of tracking bugs in the later stages. In addition, a clear understanding on the objective of testing and the planning on how, where, and what the system should be tested for is also challenges to the testing team. These issue makes software testing time consuming process coupled with various challenges erupting from inability of software testers to perform their task effectively. Hence, this paper investigated the issues, challenges and best practices of software testing activity. The document analysis was carried out in order to analyze the information. The findings reveals that 9 main issues and challenges in software testing activities. 9 best practices also discover from the survey. The findings will help software community especially testing team to aware the issues and challenges could be faced and how to perform testing activities in best way.

Key-Words: - Quality, Effort, Failure, Product, Bugs, Defect, Error

1 Introduction

Software products are fast growing and succeeding in easing end-user needs to complete various task using computers in the recent years. The end-user often wants to get software products to be implemented as soon as possible. They need a developed software product with new features to be created within short period of time. However users do not want to deal with defected software product. In order to maximize profits and keep happy customers, software development industries utilize various developments disciplined to create a product that meets end-user requirements. Failure to meet user requirement has cost several software product to exist even add extra problem to industries. Studies have shown that software failure is traceable to ineffective testing of the system under development. Researchers have recommended regular testing by competent individual to counter ineffective testing challenges. Software testing

constitutes a crucial part of software development process. Inefficient or lack of testing has caused much software project failure in the past, also have led to some social problems and financial losses due to its inability to meet stakeholder's expectations. Regardless of efforts made by software engineers to contain the challenges of software testing, software product have never been disassociated with bugs or defects. Prior research pointed the incompetence of a software tester to be the root of challenges facing software quality assurance. However, challenges of software testing did not just attached to incompetent tester but other factors such as poor documentation, poor planning, software domain and others. In order to solve testing challenges, it is important to perform testing over and over and quest for undercover future defect. This paper tends to address the issues and challenges facing software testing by reviewing prior challenges that were articulated by various researches. Solutions to the

challenges as suggested by prior studies are to be discussed and presented for awareness such as bug detection and finally testing best practices.

Although testing is cable detecting present bug in software product but cannot predict or uncover future errors. This make testing time consuming practice as testers scrutinize the entire system often to contain these challenges. Testing of software is a widespread validation method within industry, but it is still largely ad hoc, expensive, and uncertain effective. Therefore a solution to these challenges is to test a system completely. Nevertheless, software tester always faced with lack of time and resources, which is capable of limiting their ability to effectively complete software testing efforts, thus ruling out exhaustive testing [1]. This study investigates the challenges facing testing through an existing researches relating the said challenges. As fewer attentions were given to issues facing testing and countermeasures to it might constraint this study from gathering more information that really captures the problems in software testing. However, prior research review will virtually provide the necessary information and close as possible. As this study focuses on the challenges rocking testing, countermeasures will be investigated through a secondary research approach. The research approach of this study adopts a conceptual method where information are to be retrieved from a credible and reliable sources.

2 Methodology

The document analysis method was employed in order to analyses the issues and challenges of software testing activity. The selected journals and articles related to the issues were analyses which from 2006 onwards.

3 Current Challenges

Testing of software today is a huge challenging activity in software system projects. Katherine & Alagarsamy [2] and [3] defined testing as “one of the five main technical activity areas of the software engineering lifecycle that still poses substantial challenges. It’s generally believed that unidentified bug at the time of software testing is one of the biggest challenges soft-ware testing is facing currently. This is because a software system that passes it present test without defect does not guarantee future error free of the software.

The following sub sections will discuss few current challenges faced by testing team.

3.1 Inability of Testing to Detect Uncovers Defects

Some testing activity unable to detect uncover defect due to few constraints such as lack of experiences of testing team, no automated tool, lack of knowledge and others. Ahamed [4] mentioned that unidentified bugs could causes future software failure. The problem of performing a software testing is for defect detection which software can only suggest the presence of flaws, not their absence. The essence of software testing is to explore the whole system in search of bugs or for defect detection, and reliability estimation. Furthermore, [5] also supported the statement that testing can be used to show the presence of errors, but never to show their absence. Therefore, inability of software testing to track future error (defect) become a big challenge of software system testing.

3.2 Less Competent Testing Personnel

Testing process of software requires some sets of skills from the tester to perform testing effectively. When software tester or testing team is less competent, the finished product is likely to fail in term of meeting customer’s requirement or technical errors that can constraints organization profit maximization even running at loss. Ahamed [4] also believed that the issues of applying software testing to reliability estimation is that the input distribution used for selecting test cases may be flawed, that the key to software testing is trying to find the modes of software failure. Furthermore, the organization also difficult to find the suitable or competent testing team to be hired. Therefore, competent testing team to perform testing activity is really crucial in software development.

3.3 Lack of Testing Model

Another issues is lack of testing model or framework that can guide new software tester to refer as a guideline. Ramos, Esteban & Guzman [6] suggested that it is very important for software testing team members to develop other essential competences to perform testing activities efficiently in a global context in order solve current issues facing software testing. Unfortunately, there is no well-defined competence model [7] that allows organizations to assess step or procedures to perform testing activity in a software project. A testing model should consists details the process involved that required to be effective testing activity.

3.4 Time Consumption

Time consumption always become a barrier to the software development communities. In testing activity, the time allocation is very limited which normally more 60% of the time is allocated for development phases while time for testing activity is less than 40%. Trivedi [8] had quoted that modern software systems must be extremely reliable and correct. Automatic methods for ensuring software correctness range from static techniques, such as (software) model checking or static analysis, to dynamic techniques, such as testing. He also asserted that all the various testing techniques have strengths and weaknesses for example, model checking is automatic, exhaustive, but may also suffer from scalability issues.

3.5 Spurious Warning and Getting a Suitable Set of Cases

Static analysis method scales to very large programs but may give too many spurious warnings as well, while testing alone may miss important errors, since it is inherently incomplete [8]. He concluded that no testing approach of software is error free method. Trivedi claimed that “one of the major problems within software testing area is how to get a suitable set of cases to test a software system”. He recommended that the set should assure maximum effectiveness of software testing with the least possible number of test cases using one of the suitable testing techniques available for generating test cases.

3.6 Lack of Stakeholders Support

Lack of support from stakeholder especially client and top management is quite crucial. It has been identified as a serious problem in the software development activity. Quadri & Farroq [5] concludes that testing of software can only find out the known issues or errors. It provides no idea or help about uncovered defects. Testing doesn't provide help when stakeholders have to make decision to either release the product with errors for meeting the deadline; testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. They also view software testing as an activity that doesn't help to find the root causes that resulted in injection of defects in the first place during testing

3.7 Maintenance of Testing as a Service (TaaS)

Cloud computing idea which successfully saves the cost computer hardware and other values it added to information technology was view to cause some negative impact on software testing and maintenance. Testing as a Service (TaaS) is known as the major impact in cloud computing. TaaS cloud infrastructures is referred as a new business and service approach that a provider undertakes software testing practices of application system within a given cloud infrastructure for end-user as a service based on their demands.

3.8 Poor Documentation

Poor documentation always creates more challenges to software testing activity and the communities. The example of poor documentation such as lack of information provided, poor instructions, appendices and others in the documents. It is difficult to the users or to the software testing community to refer to the procedures or guidelines if problem with poor documentation is continuously occur. In addition, a failure to anticipate the reader's obstacles, questions, and environment adds more problems to software testing activity. In general, testing documentations are written for the authors and their environment not for the users. However, documents should be suitable for the users in order to cope with this challenge facing software testing to some an extent. It's very important to consider user documentation and system documentation as its necessary during the development of the software to avoid future challenges that might occur.

3.9 Poor Planning and Coordination

Poor planning and coordination were identified as one of the reason project failed. Planning and the coordination plan should be done clearly and accurately before project start. Software testing activity fails due to poor planning before executing testing cases. Planning for testing of software should be considered in prior phases of software project development [9] when testing is not given the appropriate time till the last stages of the project it leads to software project

4 Best Practices

In order to shutter the current challenges facing software testing, several actions needed to be taken. This section presents testers best practices that are capable of resolving the current issues in software

testing. Testing process consist of a test plan to a collections of test cases known as test suite. A test suite contains detailed guide-lines or objectives for each collection of test cases [10]. According to [11] “the goal of software testing is to design a set of minimal number of test cases such that it reveals as many faults as possible”. During software testing process user requirements should be well known before test case design during testing period; testing should be performed against those user requirements [5].

4.1 Test Planning and User Requirement

A well-defined test plan and user requirements are very important to the software development community. Wrongly defined and develop the test plan and the requirement causes huge impact to the organization.

Trivedi [8] suggested that before performing any soft-ware testing, it is better to have test plan first, and that user requirements should be well known before test case design. Software system under testing should be well study first and a carefully test plan should be carried out. Understanding user requirement is very important for any software tester to possess since knowledge about the users plays an important role in designing the soft-ware system test cases.

Zaineb & Manarvi [12] presumed that software tester and development members should be a part of requirement gathering team for better understanding of application requirements. Person must be well versed in a breadth of skills ranging from requirements analysis to systems design and programming. Gillenson, Racer & Richardson [13] found that person must be well versed in a breadth of skills ranging from requirements analysis to systems design and programming to be competent software tester; tester should possess a broadly based systems analysis skill and background that extends to requirements on the one end and to high-level systems design on the other.

Ahamed [4] added test plan as the general document for the entire project that defines the scope, approach to be taken, and the schedule of testing as well as identifies the test items for the entire testing process.

4.2 Understanding Software System Domain

Batra and Rishi [14] showed that understanding of the software system domain is one the characteristic of software tester. They believed that a software tester is individual who clearly understand the purpose of the software product. Ability to

understand the system under testing goes further than just a statement of all purpose of the design, and user documentation must be clearly written so that it will be easy to understand. Software tester should be able to understand the user requirement to be able to perform testing effectively. “Obviously, the user context must be taken into account,” in order to assure quality of software product.

Understanding of the software product under testing is a very crucial trait a qualified software tester should have. In order to analyze domain knowledge [15] divided the software tester domain knowledge into two main perspectives which include the users’ perspective and the application domain perspective. They claimed that users’ perspective includes knowledge of the practical process of real use and real users of the software system “together with a good understanding of the real operational context of the sys-tem. This also included knowledge of the higher level needs and goals of the users, i.e., for what purposes the system serves as part of the users’ own work”. They assumed that the knowledge category was further divided into three sub part which include “episodic knowledge of usage procedures and context, conceptual knowledge of the information con-tent and presentation in the usage context, and knowledge of problems in customer cases”.

4.3 Analyzing and Synthesizing Information

Software tester should be able to analyze and synthesize information. To be considered competent in area of communication, the tester should also be able to transmit information, ideas problems and solutions to an audience both skilled and unskilled and he is able to perform these tasks in English [6]. Software testers must be able to listen, assimilate and communicate warmly with others in a correct manner [16] expressing his or her feelings but in a respective order that please others as well. They believed that a tester should be able to manage information efficiently, able to take decisions that helps to deal with the risks involved in a software project.

4.4 Personal Skills

Critical thinking capacity and motivation capacity should be considered as parts of the most trait of a soft-ware defect tracker. Software tester should be able to motivate, support and get commitment to other team members [17], problem solving capacity [18] ability to solve problems using the information that he has available, his knowledge and expertise [8] and they also view ability to adapt to new situation is very necessary for a software

tester. However that with positive attitude a tester would be able to face up issues which might appear in the project execution and make most difficult situations works perfectly.

4.5 Software Working Mechanism

Itkonen et al. [15] noted task software testers to have knowledge of the software system's working mechanism, interactions and logic. Their case studies suggest that testers know how the features work together and the fundamental working logic of the system; tester understands how the system is supposed to react to certain kinds of changes in input data or con-figuration and can recognize failures based on that understanding.

4.6 Generic Software Knowledge

Itkonen et al. [15] referred generic software engineering knowledge to be applied mainly by visual inspection, evaluation of results or outputs, or as part of other software testing activities when unexpected failures occurred. The generic knowledge is applied without the testers paying much attention to describing the findings or how they recognized software defects.

4.7 Interaction between Developer and Tester

It is very interesting to know that a software tester should liaise with a system developer to achieve a reliable soft-ware product. The point that a tester should work in hand with a developer is just being revealed here, and its right to think that way since developers created the system, therefore the team to get vital information's when performing testing like unit testing should be the developer themselves. Although no author has share this view or ever refer a high performing software tester as a person must work in or communicated warmly with system developers. According to [19] and [20] irrespective of the methodology used for software development, there is value to ensuring that testers know particular developers with whom they are working..

5 Conclusion

As information technologies thrives to assist people to complete several task that would have been done manually on human effort based in present day. Software program has been the only mediator between human and computer hardware. This keeps the makes the demand for quality software product very high as people needs computer. Quality software is one that

meets stakeholders requirements such as user friendly and other task that were set for the program to complete effectively and efficiently. Testing has been the means of assuring the quality of the software product under development or developed. However, testing has been down with several challenges that undermine the effort of creating a quality software product. This paper has successfully outlined the current challenges and reviewed countermeasures to the arising issues incompetent software testers which have strong root in poor software product as a result of ineffective testing. The reviewed can be basis to develop software testing framework in order to enhance the effectiveness and efficiency of testing activity in software projects.

References:

- [1] Farooq, S. U., & Quadri, S. M. K, A novel approach for evaluating software testing techniques for reliability, *ARPJ Journal of Systems and Software*, 2(3), 2012, pp: 84-96
- [2] Katherine, A. V., & Alagarsamy, K. Software testing in cloud platform: A survey, *International Journal of Computer Applications*, 46(6), 2012, pp: 1-
- [3] Bertolino, A., Software Testing Research: Achievements, Challenges, Dreams and Future, *Software Engineering Journal*, 1(2), 2007, pp: 85–103
- [4] Ahamed, R., Studying the feasibility and importance of software testing: An analysis. *International Journal of Engineering Science and Technology*, 1(3), 2009, pp: 119-128
- [5] Quadri, S. M. K., & Farooq, S. U. Software testing – goals, principles, and limitations. *International Journal of Computer Applications*, 6(9), 2010, pp: 7-10
- [6] Ramos, J. S., Esteban , A. S., Guzmán , J. G., & Amescua, A., Design of a

- competence model for testing teams. *IET Software Journal*, 6 (5), 2012, pp.405-415.
- [7] Richa, R & Shallu, Performance Evaluation and Comparison of Software Testing Tools. *International Journal of Computer Science & Information Technology*, 3(7), 2013, pp: 711-716
- [8] Trivedi, S. H., Software testing techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(10), 2012, pp: 443-438
- [9] Haugset, B., & Hanssen, G. K., Automated Acceptance Testing using FIT, *42nd Hawaii International Conference on System Sciences*, 2009, Pp. 1-9
- [10] Isabella & Retna, E., Study Paper on Test Case Generation for GUI Based Testing. *International Journal of Software Engineering & Application*, 3 (1), 2012, pp.139-147.
- [11] Srivastava, P.R & Kim, T.H, Application of Genetic Algorithm in Software Testing, *International Journal of Software Engineering and Its Applications*, 3(4), 2009, pp: 87-96
- [12] Zaineb, G & Manarvi, I. A., Identification and analysis of causes for software bug rejection with impact over testing efficiency. *International Journal of Software Engineering & Application*, 2(4), 2011, pp:71-82
- [13] Gillenson, M. L., Racer, M. J., Richardson, S. M., & Zhang, X., Engaging testers early and through-out the software development process: Six models and a simulation study. *Journal of Information Technology Management*, 22 (1), 2011, pp.6-10.
- [14] Batra, S., & Rishi, R., Improving quality using testing strategies. *Journal of Global Research in Computer Science*, 2(6), 2011, pp:114.
- [15] Itkonen, J., Mantyla, M. V., & Lassenius, G., The role of the tester's knowledge in exploratory soft-ware testing. *IEEE Transactions on Software Engineering*, 39(5), 2013, pp: 1-14.
- [16] Tewari, A., & Misra, A. K., Testing Manners and Technologies: An Analysis. *International Journal of Computer Applications*, 3 (1), 2010, pp: 17-22
- [17] Iivonen, J., Identifying and characterizing highly performing testers – a case study in three software product companies, *Journal of Systems Integration*, 1(2), 2009, pp: 29.
- [18] Glass,R.L., Collard,R., Bertolino,A., Bach,J. and Kaner,C., Software testing and industry needs," *Software*, IEEE, 23 (4), 2006, pp: 55-57
- [19] Harish, R., Madhu, B. K., & Lokesha, V., A sophisticated study on best practices of agile software testing. *International Journal of Electronics Communication and Computer Engineering*, 3(1), 2012, pp: 26-30.
- [20] Devi, T. R., Importance of Testing in Software Development. *International Journal of Scientific & Engineering Research*, 3(5), 2012, pp: 1-4.