

Article Citation Format

Oveh, R.O, Preye P. & Atenaga M. (2018): Software Developers Approach to Software Testing: The Nigerian Context . Journal of Digital Innovations & Contemp Res. In Sc., Eng & Tech. Vol. 6, No. 2. Pp 29-34

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 15th May, 2018
Review Type: Blind
Final Acceptance: 24th June, 2018
DOI Prefix: 10.22624

Software Developers Approach to Software Testing: The Nigerian Context

¹Oveh, R.O, ¹Preye P. & ²Atenaga M.

¹Department of Mathematics and Computer Science, Western Delta University, Oghara, Delta State.

²Department of Physics with Electronics, Western Delta University, Oghara, Delta State

E-mail: omo_rich@yahoo.com

Phone: +2347036142579

ABSTRACT

Software testing should be seen as an art and not a science. Its relevance cannot be overemphasized. The impact of testing on the quality of software is undeniable .It produces software with a high quality that performs optimally in its life time. This research sought to know if Software companies (developers / managers) follow standard process of software testing, the importance that is placed on software testing and how the environment does affects their approach to software testing. It was discovered during this research that though the software developers are well trained to carry out software testing, the standard procedure are not usually followed. It was also noticed that pressure (internally and externally), coupled with environmental factors affect software testing process/ procedure. Appropriate recommendations were also proffered.

Keywords: Software testing, software, developers, software quality

1. BACKGROUND TO THE STUDY

Lack of proper software test have led to poor quality software and hence losses since year 2000 (Everett and McLeod, 2007). Software testing is not a phase, hence it should be treated as a cycle i.e before, during and after the software development. Myers and Sandler (2004) sees software testing in software development as an art due to its diversity. The objective of software test is to detect any defect, error, flaw or fault that could be inherent (Marchszak, 2012). The defect could be semantic or syntactic in nature. Semantic defect would cause a program to produce the wrong result, while syntactic defect is violation of the program defect present in the syntax that would prevent it from running. Meyer (2008) sees the objective of testing a program as an attempt to make it fail. However, it can be argued that the aim of testing is to increase the value of software.

Eliminating defects (and to be able to do so, by finding failures) is the instrument to reach higher quality and, therefore, value (Meyer, 2004). Software quality improvement should be the goal of software testing (Pezze and Young, 2007). A quality factor represents a behavioral characteristic of a system. Some examples of high-level quality factors are correctness, reliability, efficiency, testability, maintainability, and reusability. A quality criterion is an attribute of a quality factor that is related to software development. For example, modularity is an attribute of the architecture of a software system.

A highly modular software allows designers to put cohesive components in one module, thereby improving the maintainability of the system (Naik and Tripathy (2008). Software testing helps to check software quality. Software quality deals with concept of the software meeting the customer's specification. It shows that there are measureable values used to access the functionality of the software. Fitzpatrick (1996) defined software quality as the extent to which an industry-defined set of desirable features are incorporated into a product so as to enhance its lifetime performance. From his definition, it shows that there are industry defined set of desirable features which is used in accessing the software with the aim of enhancing the software performance in its life time. He also looks at the software performance for a long term (i.e. a life time). Highsmith (2002) that there are divergent opinions on the definition of software quality. Some base its definition on customer's value, defects level or performance. Hence the definition of software quality is either intrinsic (in the software) or extrinsic (customer's specification). There are different software quality factors which include: correctness, reliability, efficiency, integrity, usability, maintainability, flexibility, reusability, interoperability (McCall, 1977; Deutsch and Willis, 1988, Evans and Marciniak, 1987).

2. STATEMENT OF PROBLEM

Software has evolved significantly to be part of our daily lives. From the application in our mobile devices, to the computer in our offices and schools, we make use of software. Lyu (1993) emphasised the rising need for improved quality of software in terms of reliability, usability, dependability and maintainability. Software testing is an approach that is aimed at improving software quality and reducing failure as lack of proper testing can lead to a failed software or the software not performing optimally as intended. Obaro (2013) opined that software should be tested to improve the quality of software and to deliver a failed free system to the clients. Failure to test software properly is a major cause of software failure. Software testing is done with the intent of finding errors in the software. It ensures that the software is tested to see if does what is supposed to do or if it does what it is not supposed to do. (Jorgensen, 1995; Myers, 2004).

These errors are usually human errors from the programmers, analyst or client. Software testing could be done by the developer, tester, software manager, or even the end user/client. It could be done before developing the software, during the software development or after the software has been developed. Software testing is a well-known quality assurance method. Testing software completely for errors is not completely possible. Software testing validates and verifies. Verification answers the question: are you building it right? while validation answers the question: "are you building the right thing?". The broad classification of approaches to software testing is: (1) Black box testing (2) grey box testing (3) White box testing. In black box testing the tester has limited knowledge of the inner workings of the item being tested, while in white box testing the knowledge of the inner working of the software is known by the tester. In grey box testing the tester has a limited knowledge of the internal working of the software. (McConnell, 2004; Pressman, 2016; Tutorialspoint, 2017).

3. OBJECTIVE

This research seeks to answer the following questions:

1. **RQ1:** Do Software companies follow standard process of software testing?
2. **RQ2:** What importance is placed on software testing?
3. **RQ3:** How does the environment affects their approach to software testing?

4. METHODOLOGY

4.1 The Research Design

Questionnaire was the instrument used for this research. The respondents were selected based on their knowledge and experience in software development. They were mainly software developers/managers. The questionnaire was structured into two (2) parts: demographic information and developers perception. Sixty (60) questionnaire were administered across six (6) geographical areas in Nigeria (i.e. Lagos, Abuja, Kano, Edo, Delta, and Rivers State. Fifty (50) questionnaire were returned and used for the analysis.

5. DATA PRESENTATION

The demographic profile of the respondents is presented in Table 1. The respondents were software developers/ managers. Most of the respondents were male (76.0%) and the highest age range was between 21-30 (60%). They constitute the set of people that love using technology and would like to explore. Their experience in software usage was between 2-11 years.

Table 1: Demographic Information

Item	Frequency	Percentage
Gender		
Male	38	76.0
female	12	24.0
Age of the Respondents		
21-30 Years	30	60.0
31 -40 Years	10	20.0
41 - 50 Years`	5	10.0
50 and above	5	10.0
Total	50	100

Source: Fieldwork: 2017

The data was analyzed using frequency analysis to seek answer to the research questions. The Result of the analysis is presented in Table 2 and discussed subsequently.

Table 2: Software developers/ Managers Perception

Question	Frequency			
	Yes	(%)	No	(%)
Do you carry out software testing?	40	80.0	10	20.0
End users/stakeholders are usually satisfied with test software	17	34.0	33	66.0
End users/stakeholders are patient with software developers during the test phase of software development?	15	30.0	35	70.0
Software environment makes software testing difficult?	32	64.0	27	36.0
Were you formally trained on how to carry out software testing?	34	68.0	16	32.0
Much importance is placed on software testing to ensure development of quality software?	31	62.0	29	38.0
You follow standard procedures for software testing?	9	18.0	41	82.0
Total	50		100	

Source: Fieldwork: 2017

6. DISCUSSION OF FINDINGS

The response in table 2 shows that software testing is carried out by software developers. However it can be seen that the end-users/stakeholders are usually not satisfied with the software test. The cause of this is the next question that would come to mind, it can be noted from the response that end-users/ stakeholders are usually not patient with software developers during the test phase of software. This could be attributed to time (i.e. meeting deadline). It can be seen from their response that the software environment is a factor that makes software testing difficult. The response shows that the software developers were well trained for the software test, so there is no level of incompetence on the part of the software developers. The response also shows that importance is placed on software testing to ensure development of quality software. An important observation is that standard procedures for software testing is not usually followed. In answering the research question from the responses, we can see that:

1. RQ1: software developers / managers don't follow the standard process for software testing.
2. RQ2: High importance is placed on software testing, though it is not yielding the maximum result expected.
3. RQ3: the environment and other factors affects their approach to software testing.

The response from this research agrees with Everett and McLeod (2007) that lack of proper software test have led to poor quality software and hence losses. From the analysis derived from the tables above and other data collected from the survey, we can draw the following conclusions:

- i. The major challenges facing Nigeria's software development industry with regard to software testing include;
 - a. There are very few software developers in Nigeria that are formally trained to carry out software testing.
 - b. End-users/stakeholders are not patient with software developers during the test phase of software development.
 - c. End-users/stakeholders don't usually agree with the developer's budget for sufficient software testing in most software projects in Nigeria.
 - d. Software developers' knowledge of testing is limited to very few programming language IDEs and as such it is difficult to carry out testing in other development environments.
 - e. The testing methods adopted by most software developers in Nigeria are not in line with best practices for software testing.
- ii. Up to 73% of respondents did not indicate a specific method and tool for testing, the methods used by the other 27% of respondents for software testing include:
 - a. Agile testing
 - b. Black, Grey and White box testing
 - c. Random
 - d. Ad hoc
- iii. Software environment makes testing difficult therefore it is important for software developers in Nigeria to be properly trained on how to carry out testing in various integrated development environments so they are not limited to using only few IDEs for development even though they may not be very suitable for a given project.
- iv. Random and Ad hoc method of testing can result in poor quality software as they are methods of testing without any planning and documentation.

- v. Impatient end-users/stakeholders make software developers rush through the test phase of development without carrying out enough testing and that also results in the production of poor quality software

7. CONCLUDING REMARKS

The relevance of software testing cannot be overemphasized and the impact of testing on the quality of software is undeniable. It produces software with a high quality that performs optimally in its life time. In a growing software industry such as what we have in Nigeria, developers are aware of the importance of testing especially on software quality and should be very unrelenting in carrying out testing but unfortunately, the reverse is the case. It was discovered during this research that though the software developers are well trained to carry out software testing, the standard procedure are not usually followed. It was also noticed that pressure (internally and externally), coupled with environmental factors affect software testing process/procedure.

8. CONTRIBUTIONS TO KNOWLEDGE

This paper identified that software developers / managers don't follow the standard process for software testing, High importance is placed on software testing, though it is not yielding the maximum result expected and that the environment and other factors affects their approach to software testing. Based on our findings in this research we recommend the following:

1. Sufficient time for software testing should be negotiated in scope of software development with stake holders.
2. Software testing should be taken as a cycle (i.e. before software development, during software development and software development).
3. Standard software testing procedures should be followed at all times
4. Software testing training and retraining should be conducted for developers and stakeholders from time to time.

REFERENCES

1. Deutsch, M., and Wills, R. (1988). Software Quality engineering: A total technical and management approach, (Taylor, J) Prentice- Hall, Inc., New Jersey, USA.
2. Everett G. D. and McLeod, R. (2007). Software Testing Across the Entire Software Development Life Cycle. Wiley-Interscience, A John WILEY & SONS, INC, PUBLICATION. IEEE PRESS.
3. Fitzpatrick R. (1996) Software Quality: Definitions and Strategic Issues. MSc Computing Science (ITSM) Advanced Research Module. Staffordshire University, School of Computing Report.
4. Highsmith, J. (2002). *Agile Software Development Ecosystems*, Addison-Wesley Professional.
5. Jorgensen. P.C. (1995). Software Testing: A Craftsmans approach. USA: CRC Press
6. Lyu M.R. (1993) "Software Reliability Engineering: A Roadmap" Computer Science and Engineering Department, Livingstone Tower, the Chinese University of Hong Kong.
7. Marchszak T.A (2012) Improving Software Testing. Testing and organisational developments. Springer. <http://www.springer.com/978-3-642-27463-3>
8. McCall, J., Richards, P. and Walters, G. (1977). *Factors in software quality*, General Electric, National Technical Information Service., New York, USA.
9. McConnell S. (2004) Code Complete. 2nd edition. Wiley India Pvt Limited.
10. Meyer, B. (2008) Seven principles of software testing. Computer 41(8), 99–101
11. Myers, G. J. (2004). The Art of software testing Second Edition. USA: Wiley

12. Myers, G.J., and Sandler, C. (2004) *The Art of Software Testing*, 2nd edition. Wiley, Chichester.
13. Naik K.and Tripathy P. (2008) *Software Testing and Quality Assurance: Theory and Practice*. John Wiley & Sons, Inc. New Jersey.
14. Obaro J. (2013) "Software can generate \$10b annually if harnessed," [Online]. Available:<http://www.nigeriacommunicationsweek.com.ng/e-guest/software-can-generate-10b-annually-if-harnessed-obaro>
15. Pezze, M., and Young, M. (2007): *Software Testing and Analysis: Process, Principles and Techniques*. Wiley, New York.
16. Pressman R.S. (2016) *Software engineering - A practitioner approach*. 6th edition. McGraw-Hill.
17. Tutorialspoint (2017) software testing- software system evaluation. Available at www.tutorialspoint.com