ISSN 2047-3338

A Survey on Software Testing: Technique, Comparison and Analysis

Junaid Khan[1], Riaz Ullah Khan[2], Faheem khan[3] and Shafaat Raza[4]

[1,4]Gandahara University Peshawar, Pakistan
[3]University of Malakand, KPK, Pakistan
[2]University of Sunderland, UK

***Abstract*– In software testing it is adequate if testing methods are only evaluated on error identifying ability but they can also be compared upon to test which among them increases reliability. To ascertain a check for the constructive testing, we require calculating present narratives of testing methods not only for usefulness and effectiveness but also for their capability of increasing software testing reliability. Actually, software engineering is a main platform for many studies and can investigate a wide research in the selected subject and can be broadly implemented with various methods concerning a various responses all through the development cycle, which are proposed for many problems and aims at, as testing is a famous validation method in engineering and business.**

***Index Terms*– Software Testing, Techniques, Comparison and Issues**

I. INTRODUCTION

SOFTWARE testing consists the significant portion of software development cycle. The absence of testing has produced a lot of software issues in the past and as a result conveyed many societal issues and challenges and economic loses. Regardless of all the hard work still engineering and industry place in quality of software, success of testing as a result very degraded and useless as it was acquired. Because of these degradation and losses reported the industry wide deficiency in testing. As [1] testing is an extensive validation method in industry, yet it having no permanent infrastructure, costly and unexpected results. To solve such kind of issue, we have to overall test the system but will have a problem of the deficiency of time and assets and availability of resources that can degrade our efforts to successfully fulfill testing. Hence we will choose suitable testing methods that will enhance test efficiency.

To know which testing method is suitable and having the appropriate information concerning to efficiency and cost of testing and resources of testing in such are testing method. To get this kind of information, it depends on their operation, on the subject that applies for, the programming languages and the software under the test etc.

There are still many existing and advance that are made in efficiency and cost of testing and resources of testing yet we have to do more as still such outcomes are not very clear and result oriented. Therefore we will more and more assess, measure and evaluate software testing methods in such a technique that are result oriented .

On the other hand, one significant point that should be noted that the focal point of the majority experiments performed so far are only on evaluating error finding effectiveness and efficiency and not a much focus on to evaluate software testing methods on software Reliability improvement as to produce dependable software is as well a main focus on software testing. Therefore we require evaluating software testing methods not only for evaluating error finding but also for increasing the software Reliability [2].

Software testing has a very significant impact in software Engineering as it noticing the software system implementation to verify either that performs a planned and verified possible failure. Generally a testing implemented in Engineering for assurance of quality. If we look at these hundred of thousand of existing codes consecutively perform and to consider how it to describe it to the existing software Engineering research literature? Where the research resources conceding and reaching? When we write our research papers?

The main question of software Engineering researcher is that testing is associated with the functionality Testing. These sorts of problems can deal by:

i). how to generate and choose test cases

ii). how to deal it with expertly

iii). how to measure adequacy etc

These three are the very fundamentals issues in software engineering research. As a researcher we will not provide influential evidence that the methods they suggest in their research will be the adequate and be basically valuable to interact with the practitioners who are generally up to date, intelligent, and well educated. The main issue they will face is the inadequate resources and limited time bound for testing large, difficult systems, though it is commonly understandable that their viewpoint what they will know in the research literature as not focusing on their problems. As it is curious for research results to be practiced up with an industrial amount of empirical study that provide valid authentication of the planned method, practitioners usually face that execution is not supposed to be threat.

Finally, practitioners again and again condemn the deficit of vigorous appliances preserved for a well-planned testing research method. As we know that the practitioners are usually not willing to offer significant timeout of their formerly rigid arrangement to implement a newest method that they observe as unconfirmed.

The apparent candidness of observing a sample of runs, on the other hand, testing tie up a series of activities, techniques and factors originate several compound disputes.

Surely, with the complication, incidence and sensitiveness of software growing quickly, assuring that it perform according to the ideal point of quality and Reliability changes into more difficult, problematic and inexpensive. This paper assembles the several outstanding research challenges for software testing into a balanced, stable and secure Roadmap.

In section we will discuss and show that destinations are the combination of important and unachievable objectives are called as 'Dreams'. As researchers faces with several tests and encourages possible characteristics of the very complicated and inexplicable situations. In section 2 we will discuss and show the research part that shows the Achievements. In section 3 we will discuss and show the software testing and Reliability. In section 4 we will discuss and show Correctness testing vs. Reliability testing. In section 5 we will discuss and show is reliability is left out. In section 6 we will discuss and show the Software Testing. In section 7 we will discuss and show the conclusion and future guidelines.

II. ACHIEVEMENTS

Presently our point of view changes its aims and goals towards the inclusive and optimistic observation of avoidance. Presently testing is illustrated as a broad and steady act throughout the development technique [3], whose arrangement as beizer defines:

A) Testing Process

Certainly, a great deal of work in early time developed interested in methods and apparatus that assist to create like 'test design thinking' further organized and assist it within the method development. A numeral of test method model has been considered for industrial execution. Several substitutes allocated dissimilarity of a v model suggestion appropriately recognized test method which is doubtful with a number of ineffectual and needlessly industrial, on the contrary further quick methods is also supported. Relating to testing in thorough, a remarkable model achieving concentration on test driven development (TDD) [4]. The organization of an appropriate method was planned for testing in fose-2000 [1] between the basic research and a dynamic research these days.

B) Test Method

Extremely loaded values designed by the previous work to assist the methodical acknowledgment of test cases. Usually, eminent among black-box and white-box focusing on the source code whether it is influenced in encouraging the testing. A further sophisticated perceptive is resulting in [8], and the large details testing of current principle of numerous surveys and textbook are given in [8].

C) Comparison between Test Method

In consequent with assessment of method for satisfactoriness of the test and for test, many research and previous methods have integrated numerous systematic assessments amongst different methods [6], [7]. These papers and course book have permitted make a chain of command of relative concentration to specify among similar method, and to distinguish the features maneuvering the opportunity of finding the mistakes, highlight further its specific and particular on complementary division by unsystematic Testing.

D). Object-Oriented Testing

The 20 years later the focal point was on testing the object oriented software. Redundant the concept of reprocess communicates forwarding the Object-Oriented programming can so far circumvent the requirement for testing, researchers quickly appreciate the Object-Oriented development made new threats and complication, for that reason increasing the necessity and complication of testing [8]. For the most part, among the essential methods of Object-Oriented, Encapsulation could assist veil bugs and generate test complication; inheritance needs broad many testing of inherited code. However, suitable plans for efficient testing of incremental integration are essential to seize complicated spectrum of probable dynamic and static reliance's amongst categories.

E) Component-Based Testing

In the early research work, component based development yield quick software development with less resource. Testing in such away made fresh disputes that can be differentiate among industrial and educational. The benchmark and satisfactory for being set in different steps and structure, as a result the necessities of the user components component user to retest the component where it is planned. Nevertheless very important complicatedness is the countenance of the shortage of information for investigation and to test the component externally. Certainly, as component points are explicated according to particular component model, which do not suggest maximum information for useful testing As a result work of the research has supported. The testing of component-based testing was as well planned as a fundamental test in Fose-2000. See many new survey related to this topic in [9].

F) Protocol Testing

To correspond among the component of a system which is distributed the set of laws of protocol that manage, and in a sort to assist interoperability they have to be precisely exacting. Protocol testing is intended at validating protocol implementations by their specs. The later on are free of charge by standard organizations. Pressed by the demands of permitting exchanging of information, protocol testing research has succeeding all divided and classified trail concerning software Testing. Actually, survival of definite state based specs of chosen behavior. As such outcomes were observed a partial specific software testing. On transmission identical and latest complexity of assuring suitable transmission amongst the component that is not very near and a starting job on a broad side for any current software; therefore a research work on software testing can effectively research the performance of acknowledge standard appropriate specs, that is the development in current functions. Simultaneously as protocol were simply and consistently good, in this developed time and technology the very important concern is towards the top level of transmission protocol , and as a result difficulties arises of additional and typically software testing. As a result, the theoretical partition among protocol testing and normal software testing challenges is decreasing with the passage of time.

G. Reliability Testing

The main concern of software testing is its reliability i.e., the possibility for software about its malfunctioning procedure for a precise stage of time in specific environments. Reliability testing shows that with any procedure we cannot identify the last failure, and as a result, with no way for the operational conclusion to induce testing, and it try its best to eliminate those failures that are very clear in regular processes: instinctively the testers reproduce how the customer would utilize the system. Software reliability is characteristically provisional depends on reliability model.

Various model can be utilized, depends on whether the given errors are eliminated, and want to find in which situation the Reliability increases, or when the simple clearance or certification will be given to Reliability. Research in software reliability has zigzag research in software testing in various useful process and operations. Model for software reliability have been ardently measures in years 80's and 90's [10]. Such models are at the instant developed and may be produced method presenting quantitative guideline for how and how greatly the method to be test. E.g., this would concluded by Musa in his software reliability engineered testing (SRET) method [10], and is as excellently guided, that practices statistical test methods to give up proficient reliability proceedings [11]. Unluckily, the performance of reliability testing has not improve at the related rapidity of theoretical progress in software reliability, perhaps as it is a difficult and costly association, on the other hand, for the integral problems of identifying the necessary and prepared sketch out [12]. On the other hand currently the statement for reliability and other reliability reactions is rising and consequently the desire starting for valuable methods to reasonably test useful and extra purposeful events of up to date software –intensive system.

H) Software Testing and Reliability

The software testing and software reliability have usually related to two split groups. Nevertheless at present there is a tough connection between software testing and software Reliability. A significant feature of testing is to formulate quality and its features noticeable to the Reliability of the software. The Reliability features are not straightforwardly computable and have to be deriving from other procedures like failure data composed through testing. Software testing is an efficient method for calculating the current Reliability and expecting future Reliability and also to develop it. Complications of the Reliability features are that it only has an importance if it is associated to a definite client of the system. Various clients understanding various Reliability, Reliability they use the system in various techniques. If we are to approximate, expect or endorse the Reliability, we must narrate this to the practice of the system. One way of relating the Reliability to the practice is to pertaining usage-based testing [13].

Therefore software reliability can be practice to calculate, what the progress has been prepared in system-level testing [14], [15]. The amount of a continuance effort can be resolute by the sum of system Reliability that can be forfeit for the time being [16]. The notion of Reliability as well permits us to enumerate the failure-related quality feature of a software system. Quantification of the quality portion of software systems allows developers and administrator an improved approaching into the procedure of software development. In future, it is significant to carry these two collections are much related, therefore, software testing could efficiently performed, and software reliability can be precisely calculated and enhanced.

III. CORRECTNESS TESTING VS. RELIABILITY TESTING

The software testing is a general expression which contains trying to confirm and progress every phase of software quality. Software testing methods provides numerous functions in software testing life cycle. It shows the test information flood showing various reasons of software testing and correctness is the lowest need of the software. Therefore purpose of correctness of concluding program regarding its needs is the necessary functions of testing.

Most general methods of testing are centered on discovering errors to the extent that is achievable. Therefore, the major purpose is error detection i.e., correctness testing. Software testers test software with no reference to how software will function in the field, as frequently the surrounding cannot be completely showed in the labs [17]. As a result they expend more time demanding to split the software than performing standard operations and typically design test situations for outstanding and periphery circumstances only. Error detection does not essentially motivate self-reliance, [18] states one significant objective of software testing is to calculate the dependability of tested software as well as to enhance it. If the failures are more significant than errors, then the aim will follow it throughout the test phase that could also alter. On the other hand, an additional purpose of testing might to enhance our assurance in failure-free operation of the Software i.e., Reliability testing. In such a situation, we will not practice the finding of numerous errors as probable but will struggle for an elevated Reliability. To get dependence in the everyday operation of a software system, we have to imitate those circumstances. The testing of software should be tested according to the report which is operational in a way to permit precise reliability evaluation and calculation. In Reliability testing we are we are not keen in errors, but in their appearance. An error which regularly shows itself will in common basis further spoil than an error which rarely shows up. Reliability testing will logically expose previous failures that are further possible in real operation, thus direct their practices at setting up the further significant errors [19]. Our objective is to concentrate our testing efforts on both kinds, so as to increase software which will be accurate plus Reliable. The error-finding efficiency of a Correctness testing method pivot on whether the tester's hypothesis related to errors show actuality; for Reliability testing to carry on its assurance of improved utilization of resources, it is very essential for the testing report to be the actual part of operational use.

IV. IS RELIABILITY IS LEFT OUT?

It is very useful to determine it quantitatively in measuring the quality of software is its Reliability [20]. There search that has been so far evaluated shows us to assess testing methods are the focal point towards error finding capability of the testing methods. Most preceding assessments are measured all malfunctions to be equal to one another; in actuality this is not the situation. Error finding capability measure is helpful for assessing testing methods when the aim of testing is to get more reliance on the program is free from errors. It is not essential for a software testing method that can get more errors than others can attain elevated Reliability. Testing to discover errors may be more efficient, but if it exposes failures that emerge irrelevantly throughout real operation, test practices will be end up in irrelevantly making better the software which shows consumption of test and resources. If the aim of testing is to get better the Reliability of the program then the evaluation of test efficiency will have to be differentiating between those errors that make [21].

As testing methods will be employed for this reason also, it will be fascinating to measure efficacy of various testing methods for Reliability increase evaluation r testing methods should be tested for their efficiency in showing failures kinds, and as a result Reliability enhance. We can use software Reliability as a situation for measuring methods related to it efficiency to create Reliable software. For instance, think two testing methods T1 and T2 which are functional on software individually. As by observing the Reliability level of the software, we can examine which method is more effectual in generating software systems of elevated Reliability. So we become capable to contrast testing techniques in a manner that permits us to state that if a system has been tested by

Method T1, possible to have fewer threats (more Reliable) linked with its utilization than if it has been tested by methods T2.

Previously numerous times we evaluate testing methods this way. Frankl, Hamlet, Littlewood and Strigini initiated various methods for evaluating testing methods; the idea of *delivered Reliability* as a way of evaluating testing methods [22]. Delivered Reliability explains the plan that for a given program, condition, and operational division, test situations generated according to a specific testing method and several errors can or cannot identify and that modification of those errors that are identified by that testing method will enhance the Reliability of the program with the help of operational distribution. As several test situations generated according to

The particular testing methods will identify various sets of errors, various developments in Reliability can effect from concerning various testing method. A probabilistic notion of delivered Reliability is justified in [21]. As a result, instead of evaluating on the probability of finding one or more errors or failures, or the amount of errors identified, their objective is to support their estimation of a testing standard on the Reliability of the program beneath the test, once it has been tested by a given approach. While the perception on which this opinion is supported is exciting, as a lot of research is required in this instruction.

V. SOFTWARE TESTING

In software testing we have validation vs. Verification

Validation – assessing through, or when the development cycle terminate to evaluate if it assures particular needs.

Verification – assessing to establish if the generation of a particular development phase assures the circumstances forced at the beginning of that phase.

» Error, Error and failure

A substandard program implementation is a “failure”, pointing to an “error” in the program; the result itself is a “mistake” in the programmers' philosophy ([Meyer, 2008).

There are few fundamental concepts relating to bugs, and that the error and error failure are because of these bugs which are as under:

1. Communication problems
2. Altering of necessities
3. Errors in programming
4. Time limitations
5. Complication of software
6. Defectively written code
7. Self-esteem of the public

» Debugging vs. Testing

Debugging: The method of finding a error given a failure. Debugging is finding out what issues making a problem from which you are aware.

Testing: To find software by observing its implementation. Testing is demanding to locate problems from which you are not aware.

The less the number of debugging will decrease if the number of testing increases and it will help you for the improved awareness of your software is. You can put bugs if you have an improved awareness of your software.

»Static Analysis

Static analysis is not the code that can be implemented efficiently and can be validated against a synthetic specification that describe and explains the configuration of the artifact. Static analysis having the Verification behaviors and 60% faults can be with Static analysis and cannot validate the dynamic activities. Gilb and Graham change unit test by inspections (sommervillie, 2005).

» Dynamic Analysis

In Dynamic analysis the code is executable and the output is predictable if it provides input values and if for checking and there are validation activities in and can be employ to shows the presence of bugs but not their absence (Meyer, 2008) [23].

VI. SOFTWARE TESTING RESEARCH ROADMAP

A Roadmap gives guidelines to attain your destination are as under:

- The further eminent achievements from past research consisted of "you are here" red dot.
- There are relatively well-known phase, and with additional or few opportunities of achievement and yet we are in between of these challenges in front of the current and future testing research. Such testing represents the procedure to follow in the drive in the direction of Dreams and they are in the middle, the very crucial section of the roadmap.
- The ideal destination is explained in type of four set of Dreams. That utilizes this word to designate that it is asymptotic objective at the conclusion of four documented paths for research development. As they are inaccessible through the insinuation and its importance specifically uninterrupted acting as the ends of interest for cooperative, insightful research.

VII. CONCLUSION

To consider that the software testing is electrifying, attractive, intricate and articulated research field, and much anticipation from this research that has offered a useful conclusion of current and future challenges. The representation which shows to be in use as an effort in the development of that the assembly of populace may wish to familiar and increase.

Unfortunately, development may be slow by dissolution of software researcher into many dislocated population. For example, several actions have documented by group of people to bond and to give the latest outcomes, offering a small number of overlie amongst PC friends, involvement, combined information and proceedings. A necessary concluding remark alerts numerous useful relations amongst software testing and other research field. By focusing on the particular issue of software testing, we actually disregarded a lot of stimulating likelihood opening at the perimeter amongst testing and other areas. A small number of have been instantaneously met upon in this paper, as model checking methods ,or the use of search base methods, for the generation of test input, or application of methods to review acutance characteristics. Therefore to think that in fact a lot of the chances that will occur from a very holistic method to software testing research, user that understand the research can absolutely find out and appreciate numerous and original exciting synergies spanning crossways the research fields of software engineering.

Before assessing testing methods for error finding capacity simply, we will also consider other reasons into concern and as well error severity, price, effectiveness etc. We could also asses' methods for Reliability measurement. As we are aware of the fact that it is very hard to become make a useful comparison of testing methods and we cannot actually depend on imagining one testing method to surpass every other methods. There is always a particular significance or reason of assessing a specific test method, actually on the supposition that the method will be more effectual. In spite of our method, it will be useful to attempt to recognize what kinds of failures and errors a specific method can be supposed to get and at what price. We will also check whether testing method success and effectiveness rely on program to which it is functional, matter is who will apply it, the number of errors in the program or the kind of errors in the program.

REFERENCES

[1]. Bertolino, A. (2007). , "Software testing research: Achievements, challenges, dreams.", In FOSE '07: 2007Future of Software Engineering, pages 85–103, Washington, DC, USA. IEEE Computer Society.

[2]. Journal of Emerging Trends in Computing and Information Sciences ©2009-2011 CIS Journal. Vol. 2, No. 12, December 2011 ISSN 2079-8407

[3]. W. Hetzel. The Complete Guide to Software Testing, 2nd Edition. QED Inf. Sc., Inc., 1988.

[4]. D. Janzen, H. Saiedian, and L. Simex. Test-driven development concepts, taxonomy, and future direction. Computer, 38(9):43–50, Sept. 2005.

[5]. H. Zhu, P. A. V. Hall, and J. H. R. May. Software unit test coverage and adequacy. ACM Comput. Surv., 29(4):366– 427, 1997.

[6]. P. Frankl and E.Weyuker. Provable improvements on branch testing. IEEE Trans. Softw. Eng., 19(10):962–975, 1993.

[7]. H. Zhu. A formal analysis of the subsume relation between software test adequacy criteria. IEEE Trans. Softw. Eng., 22(4):248–255, 1996.

[8]. R. V. Binder.Testing Object-Oriented Systems Models, Patterns, and Tools. Addison Wesley Longman, Inc., Reading, MA, 2000.

[9]. M. J. Rehman, F. Jabeen, A. Bertolino, and A. Polini. Testing software components for integration: a survey of issues and techniques. Software Testing, Verification and Reliability, to appear.

[10]. M. Lyu., Handbook of Software Reliability Engineering. McGraw-Hill, New York, and IEEE CS Press, Lo Alamitos, 1996.

[11]. J. Poore, H. Mills, and D. Mutchler. Planning and certifying software system reliability. IEEE Software, pages 87–99, Jan. 1993.

[12]. D. Hamlet and R. Taylor. Partition testing does not inspire confidence. IEEE Trans. Softw. Eng., 16(12):1402–1411, 1990.

[13]. C. Wohlin, M. Höst, P. Runeson and A. Wesslén, "Software Reliability", in Encyclopedia of Physical Sciences and Technology (third edition), Vol. 15, Academic Press, 2001

[14]. S. Dalal and C. Mallows, "When Should One Stop Testing Software", Journal of the American Statistical Associations, Vol. 81, 1988, pp. 872–879.

[15]. M. C. K. Yang and A. Chao, "Reliability-Estimation and Stopping-Rules for Software Testing, Based on Repeated Appearances of Bugs", IEEE Transactions on Reliability, June 1995, pp. 315–321.

[16]. Kshirasagar Naik, Priyadarshi Tripathy, "Software testing and quality assurance: theory and practice", John Wiley & Sons, 2008.

[17]. M. R. Lyu, "Software Reliability Eng: A Roadmap," Intl.Conf. on Software Eng. (FOSE '07), May 2007, pp. 153-170.

[18]. Sheikh Umar Farooq and S.M.K. Quadri, ”Effectiveness of Software Testing Techniques on a Measurement Scale”, Oriental Journal of Computer Science & Technology, Vol. 3(1), 109-113 (2010).

[19]. P. Frankl, R. Hamlet, B. Littlewood, and L. Strigini. Choosing a Testing Method to Deliver Reliability. In International Conference on Software Engineering, pages 68–78, 1997.

[20]. M.R. Lyu (ed.), Handbook of Software Reliability Engineering, IEEE Computer Society Press and McGraw- Hill, 1996.

[21]. P.G. Frankl and Y. Deng, “Comparison of Delivered Reliability of Branch, Data Flow and Operational Testing: A Case Study,” Proc. ACM Int’l Symp. Software Testing and Analysis, ACM Press, 2000, pp. 124–131.

[22]. P. Frankl, D. Hamlet, B. Littlewood and L. Strigini, "Evaluating testing methods by delivered reliability", IEEE Transactions on Software Engineering, 24 (8), pp. 586-601, 1998.

[23]. Testing Software Product Lines: From Basics to Research Trends Paulo Anselmo M. S. Neto.