*International Journal of Latest Research in Science and Technology* **ISSN (Online):2278-5299**
*Volume 6, Issue 1: Page No.1-12,January-February 2017*
*https://www.mnkpublication.com/journal/ijlrst/index.php*

# IMPROVING SOFTWARE QUALITY MANAGEMENT: TESTING, REVIEW, INSPECTION AND WALKTHROUGH

**Alhuseen O. Alsayed and Anwar L. Bilgrami[1]**
Deanship of Scientific Research, King Abdulaziz University,
Jeddah, Saudi Arabia
[1]Correspoinding author: E-mail: alegman@kau.edu.sa; bilgrami1956@hotmail.com

***Abstract*** *-During the software development, errors may occur even when software testing assurance tools are of the best quality. Avoiding such errors and artifacts from products at early stages is critically important in order to supply the needs of users. This paper presents methods of improving software quality through various phases of product testing such as validation, verification, testing review, inspection and walk through etc.*

*It is clear that techniques used to check the quality of software in product lifecycle will be testing and reviews. Testing and reviews are part of validation and verification (V&V) processes, that assist to improve the quality of software product. There are some researchers arguing that one technique is more reliable and efficient than others. However, it seems that there are testing techniques and reviews suitable for specific occasions. For example, an inspection technique is better in finding a design error while testing techniques are unable to find errors in requirements documents or in source code.*

***Keywords***-Software quality, testing, verification reviews, inspections, walkthroughs.

## 1. INTRODUCTION

Quality is never an accident; it is always the result of high intention, sincere effort, intelligent direction and skillful execution; it represents the wise choice of many alternatives (William A. Foster).

Quality refers to "all characteristics and significant features of a product or an activity which lead to satisfy the given requirements"[1]. A high-quality product is associated with a number of features. These factors can be described in the requirements of specifications. During the present time, in order to guarantee quality, most companies must conduct testing, reviews, inspections and walkthroughs in order to validate and verify quality of software products. Sule [2] stated that the efficiency and capability of any organization is measured by the quality of products and services. Therefore, most companies must focus on improving their software and assurance to their quality.

Software testing and reviews are conducted in each phase of life cycle of software development in order to ensure that the product is bug-free. Moreover, each phase of software development has to be verified and validated. The software verification process is conducted to ensure that each phase delivers correct product, while the validation process ensures that product meets requirements and specifications. Software testing has various types: unit testing, integration testing, system testing and acceptance testing. Each test begins after the previous one is successfully performed.

This study aims to explore software quality improvement through the process of software testing, inspections, reviews and walkthroughs. The report is divided into two sections; first, software testing is explained along with its techniques and types of testing and the second section deals with inspections, reviews and walkthroughs, explaining how the product is tested by various methods in order to ensure that the product meets intended requirements and specifications.

## 2. VAIDATION AND VERIFICATION

Validation and verification testing is the process of ensuring the outcome of each phase in software development life cycle meeting user's requirements and specifications. Software validation and verification processes are important processes in software development life cycle along with types of testing that guarantee quality of products. In this section, validation and verification testing processes are described in detail.

### 2.1 Validation

Validation testing is the process that answers the question whether the product that has been built is correct and right in order. It ensures that the product has been developed as intended based on requirements and specifications [3]. This type of testing takes place after the verification testing is carried out. Validation testing is conducted at the end of each phase in the software development cycle in order to validate quality of each complete product.

### 2.2 Verification

Verification addresses the question, "Are we building the product right?" [4]. Software verification ensures that every step in building software product delivers correct product by conducting reviews and inspection meetings. It examines output of each and every software development phases to produce correct product. Software verification has two major types, which are dynamic testing and statistic testing. These types of verification testing are briefly described below.

#### 2.2.1 Statistic Testing and Dynamic Testing

Statistic testing focuses on system code in order to determine software quality without actual execution [1]. Some techniques used to conduct statistic testing are code inspection, program analysis and model checking. In dynamic testing, tested code is executed to find any defect in the code and helps in rectifying it [1].

**Publication History**
*Manuscript Received : 4 January 2017*
*Manuscript Accepted : 11 February 2017*
*Revision Received : 25 February 2017*
*Manuscript Published : 28 February 2017*

### 2.3. Waterfall Model

The waterfall model is one of the most widely used methods in software development life cycle for developmental process [1]. The waterfall model is called as the traditional or classic model used in software development [5]. The waterfall model is well-known model in software development because it is easy to implement and it has five phases. Each phase starts after when the previous phase is finished, which means the waterfall model phases are interdependent. The five phases are software requirement phase, design phase, implementation phase, integration phase, operation phase and maintenance phase.

Verification and validation are activities done in software development to verify and validate that the product has met end user requirements and specifications. It is therefore, verification and validation activities must take place during every phase of the waterfall model to ensure that software product is developed as intended based on the requirements and specifications. These phases are discussed below in detail.

- **Software Requirement Phase**
  In this phase, all user requirements are gathered. Analysis of requirement is conducted to carefully investigate possibility and validity' of requirements [1]. This is the most crucial stage as any misunderstanding or misinterpretation may cause validation issues later. The developer or the programmer must verify which component of requirements will be incorporated into system and explain how it will be carried out [1]. The outcome of this phase is a document called "requirements specification document", which will be considered as the guidelines for the next stage of software development: Software design.
- **Software Designing Phase**
  Software design is one of the most important phases of the waterfall model. During software requirement phase, software design is conducted by using requirement specifications as the guidelines to fully investigate and understand components to be created [1]. The software developers divide customer requirements into logical modules for easy implementation. Moreover, developer or programmer must define how design should appear and perform. Once the design is defined, the system will be designed. This process helps software developers to define and determine the right hardware and software for use [5].
- **Implementation Phase**
  Implementation is the third phase of the waterfall model. During this phase, the actual software is developed. The system is divided into units or modules for ease of implementation. During implementation phase, software developers start writing codes for each module based on algorithm designed during design phase [1]. Each unit is developed and tested for its functionality [2]. This process is referred to as unit testing, which mainly verifies if the units meet their requirements. Therefore, testers use the system to test it and verify whether the system is working as intended based on user requirements and specifications [1]. After the system has been tested and verified, the integration phase starts.
- **Integration Phase**
  After all units have been developed and tested, the integration phase begins. During this phase, individual units are incorporated and each unit is tested to form a complete system [1]. After all units have been integrated, software developers conduct system's test, which is referred to as integration testing [5]. The purpose of integration testing is to identify problems that may be created after all units have been integrated [1]. Moreover, verification is carried out to ensure that the entire system is working effectively as intended based on the requirements and specifications [5]. Once, test results come back positive and system is working properly, the system will move to the next stage.
- **Operation and Maintenance Phase**
  After when the entire system is working properly based on user requirements and specifications, the system is ready for delivery [5]. The system is delivered to the end users after alpha and beta testing [2].This phase mainly focuses to address problems that may arise after the system has been delivered to the end users. It is therefore, the software development team has the responsibility to maintain the system, and resolves any problem that may arise after when users start using the system [1].

## 3. SOFTWARE TESTING

Software testing is considered to be an important factor in the product's lifecycle. When the product works effectively and properly according to the customer's requirements, it will have a greater life. Software testing is the process of finding defects or bugs in the system, which occurs due to errors in the application [1].

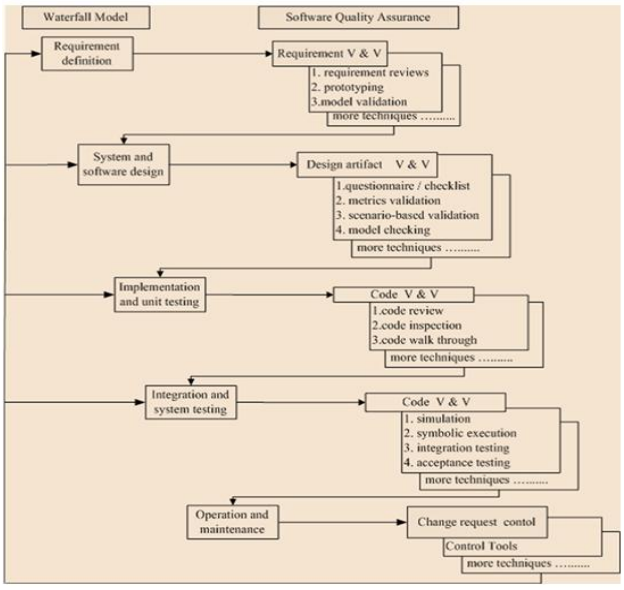


**Fig. 1. A waterfall model to Improve Software Quality**

The application error may lead to failure of the existing product and have high risks. In other words, software testing has different goals and objectives, which include revealing defects, obtaining level of confidence and preventing defects. The primary function of software testing, however, is to discover Software testing includes various techniques, such as white-box testing and black box testing. This section describes two testing techniques in detail to uncover and fix the bugs in them.

An effective test prior to product 's delivery can achieve product quality. Detecting defects before the product is released allows organization to produce high quality product. The major benefit of conducting early testing is reducing both business disruption from software errors and the cost of resolving issues. Therefore, several researchers have point out that it will be more cumbersome, troublesome and expensive if the bugs and defects are not detected at the early stages of testing [1].

## 4. SOFTWARE TESTING TECHNIQUES

### 4.1 White-box Testing

White-box testing mainly focuses on the internal data structure, code and algorithm of system application [1]. White-box testing is also called “structural, clear box and open box testing” [6]. It is therefore, testers must have specific knowledge about internal working parts of the system application. Unlike black box testing, knowledge of the internal data structure and code is required in white-box testing. To implement white-box testing, testers need to investigate internal parts of software application in order to get accurate test results based on requirements [1]. Testers could use white-box testing for units of software application, which will be discussed in later sections.

#### 4.1.1. Advantages and Disadvantages of White-Box Testing

There are various advantages of using white-box testing technique. First, the tester has to understand internal data structure, which makes it easy to find out which type of data can help in testing applications[3]. Another advantage is that white-box helps optimize the code. White-box testing helps in removing the extra lines of codes, aiding detection of hidden defects.

However, because white-box testing requires specific knowledge of internal working structure, skilled technical testers are needed to conduct this type of testing. Parekh [3] has indicated that it is impossible to investigate every part of the code to discover hidden defects, which may create problems that result in application failure.

#### 4.1.2 Black-box

The black-box testing technique focuses on functional part of software. This testing technique is based on input data and output results [1]. As the name "black-box" implies, tester does not need any knowledge about t internal software parts [7]. Black box focuses on output results [4]. In black box, testers must have specified requirements, which come from the end user or organization [6]. The tester implements test by inputting specific data and receiving intended output according to users' requirements. Black box testing is a useful testing technique, when it is combined with other testing types. Black box testing technique has various advantages and disadvantages for software tester.

In terms of advantages, one can first consider flexibility of use. The tester does not have to be a technical person or require to have advanced knowledge about internal working units. This technique also is less time-consuming [6]. Because testers have no concerns about internal working units, they do not spend huge amount of time testing all internal paths. Finally, simplicity is important because the black box can be used when there is a large and complex application or system, and because tester does not investigate the internal "mechanism" of the system [6].

With all these benefits, tester can still find some noticeable disadvantages with regards to black box testing. Firstly, black box requires identifying tricky inputs, which is difficult, if the test cases are not developed based on defined specifications [6]. Another disadvantage is that the black box may leave untested programs during the process. Despite these drawbacks, many testers find black box easy to use for testing a software product.

## 5- TYPES OF SOFTWARE TESTING

Software testing is of various types that includes unit testing, integrating testing, system testing and acceptance testing. Software testing is interdependent, since none of its various types depend on each other. Each type of testing starts only after when the previous is successfully completed.

### 5.1 Unit Testing

A unit is the smallest testable part of software. It can have a few inputs and one output result. Unit testing is the most 'micro' scale of testing [8].It is also the first level of software testing where each unit or component of software or system application is tested to find bugs before moving on to integration testing [1]. Unit testing focuses on validation of performance of each unit as designed [9]. Unit testing is operated by isolating programs into number of units in programs and source codes in order to examine whether they are working properly [1]. Unit testing is the responsibility of software developers as they are familiar with designing and working of the product [8].Some studies have found that software testers could take part in unit testing. Once the units are tested and validated, it is time for integration testing.

### 5.2 Integration Testing

Integration testing is the second level of software testing, conducted when two units of software program are combined and tested together [1]. The focus of integration testing is to detect hidden bugs or problems after units have been integrated [1]. Due to combination of units, the likelihood of finding defects is higher. Therefore, units must be tested after integration. Integration testing also includes performance and load testing.

Performance testing is a type of testing used to examine and understand applications scalability, when the number of users or data is increased. In short, it determines “how fast some aspects of system perform under a particular workload” [8]. Performance testing is conducted by creating a test case in which the tester records rate of performance of a system at a

particular workload [1]. It is therefore, by using a performance testing, the tester will determine system's speed and its effectiveness.

Load testing is performed to examine maximum load that an application can handle [1]. As such, this testing enables the tester to know the highest level of load when the system is working properly, and the lowest level when the system is under stress [1].

**5.3 System Testing**

System testing is the third level of software testing, which comes after the system has been integrated and tested successfully. In a typical enterprise, programmers conduct unit testing to examine individual components or units of code [10]. Moreover, all individual components or units are integrated and tested. Once, components are integrated, the system as a whole is tested by examining functions of software based on overall requirements and specifications [1].This process ensures that the entire system is working properly and has no defects. It is therefore, the importance of "System Testing" for software quality cannot be overstated as it ensures functional and technical requirements. After the system testing is completed, user acceptance testing begins to ascertain whether the system satisfies requirements and needs of the end users.

**5.4. User Acceptance Testing**

Testing of user acceptance is the final stage of software testing. This level of testing is carried out after the product is tested. It is conducted by handing over the product to the end user to verify that the developed product meets specified requirements [3]. Acceptance testing falls under " black box methodology" where user is not concerned with internal codes of software product [4]. Moreover, end users are expected to use application and give feedback on product's functionality [1]. The focus of testing of user acceptance is to ensure that the requirements of the end users are met [11]. User acceptance testing is an essential step before the end user finally accepts software product. It is therefore, user acceptance testing has two types of testing: Alpha and Beta. The former is the type of user acceptance testing carried out at developer's site while the latter is done at user's site [4].

## 6. OTHER TYPES OF TESTINGS

There are other types of testing used to detect problems in software products, such as monkeys testing and regression testing.

**6.1. Monkeys Testing**

This type of testing is conducted randomly by using automated tools to test the units of software application. Monkey testing tests the unit by entering data into the software units to ensure proper handling of all possible user input that may cause software error. The focus of this type of testing is to ensure that the software's functionality is not affected when random input data enters the program. Monkey testing has two types of testing: 'smart monkeys' and 'dumb monkeys'. Smart monkeys are valuable for certain functions, such as load testing. Smart monkeys are efficient for detecting bugs but expensive to develop [12]. Dumb monkeys, on the other hand, are less efficient in detecting bugs but inexpensive to develop [12]. Dumb monkeys testing have to be developed with other types of testing due to the likelihood of bugs being hung and crashed.

**6.2. Regression Testing**

Regression testing is conducted when the software has been modified. This type of testing is done to ensure that the modified software meets its specified functional requirements and that the software works properly [10]. It is therefore, regression-testing attempts to verify that the software application works with no bugs even after any changes or modifications are made to the software. Moreover, it ensures that new changes or modification to the software application have not introduced any new bugs [1]. Because of any changes or modification made to the software application can cause bugs or software breakage, regression testing is an important type to be implemented. Regression testing will be widely performed on the entire application including the modified areas in the application [1].

## 7- RREVIW

In a review, the producer and other individuals examine the work product for defects. The work product is deliverable or the outcome of any phase in the software development lifecycle [13]. An example of work products would be requirement models. The objective of review is to detect any defect on the outcome of each software development lifecycle phase. Reviews can be formal or informal as briefly described below.

**7.1 Formal Reviews**

This type of review is conducted at the end of each phase in software development life cycle. Formal review is unlike informal review, since it has formal agenda and definite schedule. Moreover, the date and time for performing formal review is addressed in the project plan [13]. In the formal review, materials must be well organized and prepared, such as a software requirements review.

**7.2 Informal Reviews**

An informal review is a process in which the software tester asks a member of the software team to review the work products of each phase in software development [14]. Informal reviews can be conducted whenever needed. This type of review is done without a formal agenda or definite schedule. The date and the time of conducting any informal review process will not be addressed in project plan. Therefore, the software member can conduct informal review anytime as needed. The material of informal review may be as "informal as a computer listing or hand-written documentation."

## 8. WALKTHROUGH

Walkthrough is considered as the form of "software peer review" in which a work product is examined by the author of work product along with one or more colleagues to evaluate technical parts of software product [13]. In a walkthrough, the author presents a full description of software product to participants and asks them to write some comments regarding the technical quality or document content [15]. The walkthrough review is informally conducted to gain feedback about the technical content of the software product document.

## 9. INSPECTIONS

Inspection is one type of 'formal peer review' in the software development life cycle [13]. Formal inspection review is the most effective technique for software peer review [15]. This type of software review aims to find defects and problems mostly in documents, such as requirements, specifications, test plans, test cases, and coding. Another benefit of using inspection review is to find problems in each phase of software development and report these problems without fixing them.

Inspection teams are consisted of a moderator, author, reader, inspector and recorder
[15, 16].Each one has a different responsibility. The moderator is responsible for managing the inspection, reporting and scheduling meetings, collecting data and overseeing the follow-up. The author's responsibility is to define the artifacts to be inspected. The author also provides an overview of the product and may respond to questions raised about the product during the inspection [16]. While the reader is responsible for leading the team through the artifacts being inspected during the meeting, the inspector identifies defects in each work product [15, 16]. Finally, the recorder is responsible for documenting all defects in the list during the inspection meeting.

A formal inspection also consists of various activities [15].The inspection includes planning, overview, preparation, inspection meeting, rework and follow-up. In planning activities, the inspection team is selected and materials to be inspected are confirmed [16]. Moreover, since the inspection is conducted formally, the time and date for each material should be scheduled. The second step is an overview meeting wherein the author is given the opportunity to educate the inspection team about what is to be inspected [15].In the preparation step, the author allocates each participant's responsibility for examining the work artifacts prior to the inspection meeting. Identifying and classifying defects occurs during the inspection meeting. The entire inspection group should agree on these defects [15]. Once the defects are identified and classified, the rework process begins. In this process, the author is responsible for solving the defects discovered in the work product artifacts. The final step in the inspection process is a follow-up process wherein the moderator or the entire team verifies that all defects are removed, problems overcame and no more fixes remain [16]. After conducting the follow-up process, the inspection process is complete. Inspection is an important process in software product development as it guarantees high quality.

## 10. DISCUSSION

Many studies in the field of software development mention that the review technique can be more effective than the testing technique because it can find defects earlier than testing in the software development cycle. It is therefore the review techniques significantly affect the cost, quality, and development time of the software, since they can be done early in the development life cycle [15]. Inspection has another benefit of the reviews over testing; Inspection techniques can shorten delivery time by reducing the time spent in the integration and system test and debug phases, since a cleaner product is passed into those late-stage quality filters. Better quality in the completed product saves maintenance time. In addition, inspection techniques reduce efforts that the software tester has to exert on fixing bugs after delivery; this saves time and effort of software tester for new development work [15].

This report has investigated research in the improvement of software quality via the process of software testing, inspections, reviews and walkthroughs. It is clear that techniques used to check the quality of software in product lifecycle will be testing and reviews. Testing and reviews are part of validation and verification (V&V) processes that assist to improve the quality of software product [17]. There are some researchers arguing that one technique is more reliable and efficiency than others. However, it seems that there are testing techniques and reviews suitable for specific occasions. For example, an inspection technique is better in finding a design error while testing techniques are unable to find errors in requirements documents or in source code.

Mihnea & Constantenescu [18] stated that efficiency and quality are best served by approaching testing activities in a structured and scientific way, instead of, unfortunately, usual 'monkey-testing'. They have further suggested that effectiveness of testing effort can be maximized by selection of appropriate testing strategy, good management of testing process, and appropriate use of tools to support testing process. The net result would be an increase in the produced software quality and a decrease in costs, both of which can only be beneficial to a software development organization.

## REFERENCES


[1] B. Mills. Software testing: the sea monkey project, Bachelor's thesis, University of Applied Science, Turku, 80, 2010.

[2] K. A. Sule. Waterfall model phases, http://www.buzzle.com/ articles/waterfall-model-phases.html, viewed 29th August 2016, 2010.

[3] N. Parekh. Software – white box-testing strategy, http://www. buzzle.com/editorials/4-10-2005-68350.asp, viewed 2nd September 2016, 2010

[4] N. Parekh. Software testing - black box testing strategy: an introduction to black box testing strategy and types of black box testing, http://www.buzzle.com/editorials/4-10-2005-68349.asp, viewed 6th September 2016, 2011.

[5] B. Satalkar. Waterfall model life cycle, http://www.buzzle. com/ articles/waterfall-model-life-cycle.html, viewed on 21st October, 2016, 2010.

[6] TestPlant Ltd .Black-box vs. white-box testing: choosing the right approach to deliver quality applications, 1-4, 2011.

[7] Exforsys Inc. Unit Testing: Why? What? & How? , http://www.exforsys.com/tutorials/ testing/unit-testing.html, viewed on 21stSeptember, 2016, 2006.

[8] R. K. Bal. N.D Software testing, 2007.

[9] Exforsys Inc. System testing: why? what? & how?',http:// www.exforsys.com/tutorials/testing/system-testing-whywhathow.html, viewed 14th September 2016, 2006.

[10] Exforsys Inc. User acceptance testing: why? what? & how?, http://www.exforsys.com/tutorials/testing/what-is-user-acceptance-testing.html,viewed 16thOctober 2016, 2006.

[11] Exforsys Inc. What is regression testing ?, http://www.exforsys. com/tutorials/testing/what-is-regression-testing.html,viewed 15th September 2016, 2006.

[12] R. Davis. What is monkey testing?,http://www. robdavispe.com/free2/software-qa-testing-test-tester-2192.html, viewed on 21st September, 2016 (2011).

[13] E. Gottesdiener.Review, inspection, and walkthrough, 2011.

[14] Neodec. Informal reviews - the inexpensive way to get benefits, http://testcontext.blogspot.com/2008/12/informal-reviews-inexpensive-way-to-get.html, viewed 20thSeptember, 2016, 2008.

[15] E. K. Wieger, Improving quality through software inspections1, 1-15, 1995.

[16] O. Laitenberger.&M. G. De Baud.An encompassing life-cycle centric surveyof software inspection ISERN-98-32', 1-43, 2002.

[17] T. R. Devi. Improving Quality of Software through Formal Inspection International Journal of Engineering Research and Applications. 2: 552-557, 2012.

[18] I. Mihnea&R. Constantinescu. Testing: First step towards software quality. Journal of Applied Quantitative Methods. 3:241-243, 2008