

Software Testing Techniques & Quality Models: A Comparative Study

M. Subhan Dar, Shahra Asif Haafza, Saleem Zubair Ahmad

Abstract— Quality indeed is the basic root in software innovation. Testing is a significant part of securing program reliability. Testing science is as ancient as contemporary computer technology. Testing applications is a critical technique for increasing the efficiency of software. Quality models are very useful in quality assurance, and each model acknowledges a crucial qualitative study of characteristics and interactions. In this paper we have progressed through so many vital parts for software quality software testing methodology. Our study focuses on determining, evaluating and concepts are related between the new testing techniques and reliability models based on various factors and qualities

Index Terms— Software Testing, Testing Methodologies, Testing Techniques, Software Quality, Quality Models, comparative study of testing techniques, comparative study of quality models.

1 INTRODUCTION

Software testing is a technology whose objective is to examine or enhance the characteristics of the system or its specifications and whether it meets the necessary qualities or not[1]. Software as well as browser testing for client and server architecture are the key methods used. The assessment of software is one of the primary tools to produce software of high quality. Software audits are carried out for the recognition of software fault defects[2]. Quality assurance is a methodology intended to assess a program or product's feature or potential and to evaluate its consistency. Software testing also evaluates the software for other qualitative software parameters such as usability, utilizability, credibility, protection, power, performance, portability, maintenance, usability, etc. [3].

This behavior thus creates a divergence between the real and predicted outcomes. Software testing refers, in the existing system or software, to the exploration of bug, malfunction or incomplete specifications. This research hence gives stakeholders dependable awareness of the product consistency[4]. The central component of the research is to identify failures in software to remedy defects. In important to maintain functionality and reliability, testing is an essential condition for successful management in software engineering[5]. In order to make sure that software meets the technological and business requirement, it can also be interpreted as a verification and validation method of software [6].

Verification is conducted in order to ensure that perhaps the program complies towards structural tests and is identical to the techniques work while validation is done with test software (SUT).

In contrast, convenient and structural measurements were included in the testing techniques. Functional verification is carried through depending on functional criteria, while structural evidence is collected on the code itself[2]. In order to assess applications for durability, compatibility, efficacy, reproducibility, etc. software verification is done as well. Testing is pricey but resisting automated testing proves much more expensive. Software Quality Assurance is an integral feature of it and different businesses invest up to 40-fifty percent of overall product testing production efforts[1]. The idea of Software Quality, called a Quality Model, is highly complex and can only be fairly portrayed by some organized function and attribute structures. Upon its specs and scope of the software product, testers and developers have conditions for quality models that allow quality measures of the structure to be formed [7].

For developers, consumers and software engineers it is also perceived as a very powerful predictor. Software engineering literature requires multiple quality models, all of which have a variety of quality characteristics or influences, as some models call them[8]. Software development is not a "miracle cure" that guarantees high quality device output. While a "correct" correctness that means technology will always behave in the same way [9].

2 LITERATURE REVIEW

Software is essential to give most organizations a competitive advantage. In business products, software and databases, software has now become a big part [10]. Software development

M Subhan Dar is currently pursuing masters degree program in software engineering in superior University, Lahore, Pakistan E-mail: msse-f20-008@superior.edu.pk

Shahra Asif is currently pursuing masters degree program in software engineering in superior University, Lahore, Pakistan E-mail: msse-f20-008@superior.edu.pk

Saleem Zubair Ahmad is currently teaching to masters degree programs in superior University, Lahore, Pakistan E-mail: saleem.zubair@superior.edu.pk

has always been a task, In the area of software production, software testing is an indispensable task, and it plays its basic role. Software testing is an activity aimed at identifying errors in a given code segment. [11].

When describing the importance of testing, Anju Bansal pointed out that testing is a crucial activity in the implementation phase (SDLC). To better evaluate its consistency, software testing is an essential mean. [12]. Gómez et al. [13] recommended Black-box testing or white-box statistical tools may be used by technical workers. Nidhra and Dondeti [14] brought up literature on all testing techniques the overall intention of this study was to determine the important scientific methodology and case scenarios and their entitlements clearly. Acharya and Pandya [15] implemented the three promotional tools, and how to apply this approach to inspect key user interfaces.

Gauri et al. [1] explained up to now, the successful methodologies are "white, black and grey box". Except for these methods, other methods have not been successfully implemented. Dr. Hussain and Dr. Singh [16] authors concluded White Box test technology would have better results for code quality. Sawant et al. [3] described testing criteria, software testing priorities and concepts. They outlined alternative approaches, such as credibility audits, performance indicators and safety standards [3], [17] a comparison was also made between debugging and testing, and it is pointed out that testing is more than just debugging [3].

Poonam [5] also discusses the basis and approaches of software testing for software testing case design. Planned and executed a series of test steps unit, verification, integration and system testing. The work of Sethi [18] also reflects on analyzing the levels, varieties and innovations of software development to identify the optimal program management techniques. They concluded that manual testing is very difficult and therefore cost-effective, while using automated testing tools can reduce testing costs.

Much literature has been reviewed on software quality models Researchers have often sought to achieve flexibility, usability, credibility and other areas of software over the past few decades to make it easier to use and boost customer loyalty [19]. The quality model plays a vital role in ensuring quality and has been developed for more than 40 years [20]. Hemayati and Rashid [21] said that the existing quality models are not comprehensive enough because they do not consider all aspects related to quality. To determine all quality related features of the software development process, they investigated 19 quality models and analyzed their quality attributes.

Miguel et al. [22] While reviewing the quality model; 14 main models, 6 basic models, 4 tailor-made models and 4 open source models, concluded that The core design methodology will be ISO 9126/ISO 25010, and encounter needs to go forward with the model as a quality factor. Gordieiev et al. [34] described the QM by a hierarchical structure. The elements of the hierarchical structure are the collection of features and the affiliation between them. AL-Badareen et al. [19] adopts well-known conceptual model for android devices (ISO, McCall, Boehm, FURPS and Dromey). Each model is investigated intensively, conveying the upsides and downsides. Finally, a complete comparison has been done of the selected styles.

In order to obtain precise and accurate gaps between software security models, a new comparative algorithm is presented. Summon and Rohtak [24] models were very helpful. When it comes to professional, they exemplify what people think is noticeable. Depending on their demands, various companies may use different testing procedures. They discussed a comparative analysis of various software quality models (McCall (1977), Boehm QM (1978), ISO 9126 standard QM (1986), FURPS (1987)/FURPS+ (2000), (CMM 1991) and 11 others) used by various organizations.

3 SOFTWARE TESTING PRINCIPLE

Premise is the rule or approach to be implemented in action. Below that are various research principles:

- Evaluate a program for regression
- Testing the procedure under which model is created in order to identify faults. In the case of software errors, we must demonstrate them.
- Start testing early
- This enables to address serious mistakes in the early stages, limiting rework in initial stages to analyze information.
- Testing is context dependent.
- For various points in time, research can be acceptable and different..
- Define Test Plan
- The Project Methodology commonly provides a clear description, test targets, test approach, test environments, test outcomes, risk and reduction deliverables, plans, test thresholds to be used, procedures, strategies and tools to be using it. The scope statement should serve the needs of a company and its customers reliably.
- Design Effective
- Test cases should be described so that the toxicology reports are uncontroversial and observable.
- Examining among relevant and inaccurate words

- In accordance to the appropriate access, we will now have to test process for obviously false and unresponsive components.
- Assessment of different participants must occur in various level
- Reasonable notice at research levels, even as different groups of people can undertake experiments differently using various sample approaches at different testing levels.
- End of Testing
- Somewhere testing must also be avoided. If the hazard is now within some level or capped, the testing may be stopped [3].

4 SOFTWARE TESTING METHODOLOGIES

4.1 White Box Testing

The process function and intrinsic summary have been seen in this study. It is also exceptionally suitable to evaluate and correct defects, as defects will always be detected before complications arise[25]. The analysis identifies inputs in order to apply the code and chooses the necessary outcomes. The willingness to program and to undertake learning is vital. White Box Work is intended to test the experience for users and the overall message [16]. White box validation is a compliance analysis method which could be used to verify the functionality of the implementation process, confirm the security functions introduced, and recognize malicious activity.

4.2 Black Box Testing

A black box is any system that would not comprehend confidential data and functions or is transparent to the user. It compares applications based upon specifications for functionality and production and it has no coding or internal program structure. The key function is to monitor how well the deliverables meet the system requirements[25]. The black-box review is often called practical screening as here other than in the nature of inputs and outputs can even be grasped the functionality of the black-box.

Its core purpose is to ensure that the input is approved and output correctly generated. Functionality inspections take effect and the needs[1]. The Black Box Tests provide very little if any discussion of the system's internal functional structure, they look at the important ingredient of the system. It assures the right recognition of input and the implementing those ideas of output [3].

4.3 Grey box testing

Vetting the Gray Box is a way of measuring software programs with tools. The procedure is independently of the platform and

expression. The current Gray box theory depends heightening as to how the functionality is delivered and tested using such a host system debugger[3]. The system is defined as checking software, and therefore is comfortable with the internal rationale and coding behind its implementation. It incorporates internal data methods and algorithms rather than anomaly based tests which is less than white box testing for testing test cases. This method is significant when trying to check the addition of multiple or more code modules written by various developers [25].

5 A COMPARISON OF TESTING TECHNIQUES

The table below highlights the research differentiation between Testing Methodologies

TABLE 1
COMPARISON OF DIFFERENT QUALITY ATTRIBUTES

Sr#	BLACK BOX TESTING	GREY BOX TESTING	WHITE BOX TESTING
1	Only analyze the basic aspects, do not understand the internal workings.	Have a certain understanding of internal work.	Fully understand the internal workings.
2	Also called Functional testing.	Also called Translucent Testing	Also called Structural testing.
3	End customers, analysts and developers behaved (user acceptance testing).	Initiated by end-users, review sites and designers (user acceptance testing).	Development teams enforce it.
4	Black box testing is based on requirements	It uses a straightforward form of black box assessment.	Required to check the simple linear regression.
5	This can only be done through hit and trial way	(if known) Data fields and internal boundaries can Tested	Data domain and internal Boundaries can be better tested
6	Inappropriate for algorithm testing.	Inappropriate for algorithm testing.	Appropriate for algorithm testing.
7	In this type of testing programmer and tester are independent of each other.	This testing method is platform and language independent.	Skilled testers are required to conduct this type of testing
8	It's really the least duration and yet the most extensive	Require some practice and adsorption capacities of test	The most rigorous and time-consuming test.
9	Defect detection rate is low in this type of testing	Defect detection rate is medium in this type of testing	Defect detection rate is high in this type of testing
10	Less costly	Cost effective	Highly costly
11	Granularity is less in this	Granularity is average in this	Granularity is high in this

6 SOFTWARE TESTING METHODS

The major interest in this section is on the various methods of software testing. Test methods for three software applications can also be divided:-

6.1 Manual testing (static testing)

The procedure can be performed uniformly in the early stage of its life cycle and slowly and meticulously. Static testing is often linked to. Analyst, developer and testing team [3] are essential. Analysis of computer software without ever executable files is the stiffer analysis [26]. Study is often pursued with some interpretation of the source code with some kind of object code in the other instances[27]. Different static or the schemes for manual testing are listed downwards

- Walk through
- Analysis casual
- Scientific Revaluation
- Evaluation

6.2 Automated Testing (dynamic testing)

Automation is always used by tester to develop the products by machine. Automation Software is used to run test cases or manually executed scenarios once [9]. It happens routinely with the program, feedback values and testing whether it would be possible to accomplish such test cases manually, or using artificial intelligence, as expected [27].

7 SOFTWARE QUALITY

Software testing techniques are the tools which are used to make sure the quality of the product for this purposes different quality models are also been used widely. Software quality is a structured and comprehensive number of software quality further across.

- SQA, quality management, estimation etc. was used. The consistency of the curriculum rated as: IEEE 610.12 (IEEE 1990)
- The overall satisfaction of the plan or system regarding clients' wishes or specs [28].

7.1 Software Quality Models

We characterize the quality performance measure as numerical simulations offering quantitative estimates for selected quality features or sub-characteristics known as software measurement data [29]. In the manner of analyzing different forms of electronic products [19]. Multiple software quality models are also being introduced. The Quality Framework has been developed for assessing those qualities that the developer needs and explains in the final method. Up to this point, we have implemented a selection of quality models. In 1978 they stipulated a quality software system and separated the two faction's characteristics, namely.

- Depth Analyses of the service
- Migration of the service.

Still one of the most popular models were McCall and Joseph. Under these terms, McCall and Joseph indicated those functionality. In response to most McCall influences, Boehm's second model was proposed; appears to contain hardware capabilities, records and other features. The third FURPS model disintegrated efficiency into two different instructions sections,

- Functional requirement
- Non Functional requirement

Input- and planned output-defined functional and non-functional needs such as usability, reliability, etc. The third Dromey's model presented new innovative services to consider a different model, i.e. reusability and maturity. The roles were listed in four categories.

- Fix ability
- Inner
- Background
- Informative.

But so far as several main elements were worried, ISO recommended ISO 9 126 (ISO, 2001) useful tips which delivers a standardized interpretation of the quality of software. Reliability, Performance, Usability, Portability Maintainability.

7.2 Need of Software Quality

In many stressful applications software is currently used and software defects have indeed been substantial, even physical, damaging. These applications could include aviation software or car drive software, satellite communications software, industrial plants or power plants.

7.3 Importance of Software Quality

For customers and developers alike, product experience is the key because the organization is unhappy to work with high quality applications and for the creator too.

Increasing Criticality of Software Unconsciously, the individual or user is nervous with the storytelling ability and efficiency of the software. Elevated interdependence on computers by organizations is really being adopted and software is increasingly used in areas that are vital to survival.

The Intangibility of Software This makes it almost impossible to know whenever a given project assignment has been implemented effectively. The results of that very function can be meaningful by asking designers to build "project requirements" for quality testing [24].

7.4 Quality Models Background

In possible to correlate diverse types of technologies services, many software quality models are presented. The most important quality management parameters were calculated in

numerous studies in this subsection [19].

McCall quality model

The quality model brought to the system development system and systems integration mechanism by Jim McCall, known as McCall quality model[30] Is one of the earliest productivity model that also relied also on Quality Triangular. Differences did cover that value template of McCall: I Service of the Product (ii) Conversion to the Service (iii). The McCall model was particularly interested in keeping in mind the ties about reliability and measurements. This model was used to develop other quality models[31].

FURPS Quality Model

Robert Grady and Hewlett-Packard Co. have implemented the FURPS concept. The capabilities is split into two groups by working and non-functional specifications, and as per the customer's needs[19]. FURPS stands for functionality, usability, reliability performance and supportability[24].

ISO 9126 Quality Model

Seeking new evaluation standard is ISO 9126. The standard is separated into four sections highlighting key areas: the quality model; outward metrics; monitoring programs; as well as metric quality[32]. This standard is utilized to identify a software cooperative work and a compilation of characteristics assessing directions [24].

Dromey's Quality Model

The Dromey model is based on the consistency standpoint. Furthermore the quality assessment is distinguishable for each product and an even more clinical manifestations is performed out[22]. Dromey appears to contain character traits of high quality: functionality, durability, achievement, accessibility, maintenance, portability, reusability and maturity of the process[32].

8 THE COMPARISON METHOD

Today, multiple user experience analyses were done on a basis of established as for McCall, Boehm, ISO and FURPS types appraisal of general and different items of software quality. In order to calibrate the closed template and per the expected range, the idea of producing software quality models is focused on distinctions between picked well-known models. The comparisons were carried out with the help of unique activities, understandings including affects. A dichotomy exists with either the concept of software quality variables.

This dissertation involves establishing a standardized

method of distinguishing and mathematically identifying between model software output. Which helps to prevent disagreements during creation. In comparison, it helps identify a particular standard for performing the simulation for software quality. Table 1 shows comparisons have been made, but with development indicators, among the same prototypes and the same influences. It demonstrates that associations have been formed depending on multiple perspectives even by researchers. Hamada, for example, indicates that the credibility of McCall, Boehm and ISO models is compatible.

The authenticity used in McCall and ISO reveals that McCall and Boehm include fairness. In all selected designs and the flexibility has been included, but Haiguang is the critical aspect for Boehm and a short and mid for maintainability in ISO. In terms of the concepts of software quality variables, correspondingly, theoretical perspectives of software quality are formulated to meet the same software quality's identification. Therefore, in this research we plan to use an outlook and discussion methodological approach of four well-known types of quality models.

TABLE 2
COMPARISON OF DIFFERENT QUALITY ATTRIBUTES

Sr #	Quality Attributes	ISO 9126 Quality Model [22]	FURPS Quality Model [24]	McCall Quality Model [28]	DROME Y'S Quality Model [8]
1	Functionality	✓	✓	✓	✓
2	Reliability	✓	✓	✓	✓
3	Usability	x	✓	✓	✓
4	Understandability	x	x	x	
5	Portability	✓	x	✓	✓
6	Efficiency	✓	✓	✓	✓
7	Maintainability	✓	✓	✓	✓
8	Correctness	x	x	✓	
9	Reusability	x	x	✓	✓
10	Testability	✓	x	✓	x
11	Integrity	✓	x	✓	x
12	Understandability	✓	✓	x	x
13	Accuracy	✓	x	x	x
14	Supportability		✓	✓	x
15	Interoperability	✓	x	✓	x

In addition, we evaluated and compared a number of similarities and identified the key variability between the models. This helps to establish a standardized method of

differentiated between model software outputs. It also serves to clarify a required target for performing the simulation of software quality

References

- [1] J. Gaur, A. Goyal, T. Choudhury, and S. Sabitha, "A walk through of software testing techniques," 2016, pp. 103–108.
- [2] C. Sharma, S. Sabharwal, and R. Sibal, "A survey on software testing techniques using genetic algorithm," *ArXiv Prepr. ArXiv14111154*, 2014.
- [3] A. A. Sawant, P. H. Bari, and P. Chawan, "Software testing techniques and strategies," *Int. J. Eng. Res. Appl. IJERA*, vol. 2, no. 3, pp. 980–986, 2012.
- [4] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad, "Software testing techniques: A literature review," 2016, pp. 177–182.
- [5] P. D. Poonam, "Software Testing Strategies and Methodologies."
- [6] A. K. Arumugam, "Software Testing Techniques & New Trends."
- [7] C. Izurieta, I. Griffith, and C. Huvaere, "An industry perspective to comparing the sqale and quamo software quality models," 2017, pp. 287–296.
- [8] Y. Boukouchi, A. Marzak, H. Benlahmer, and H. Moutachauik, "Comparative study of software quality models," *IJCSI Int. J. Comput. Sci. Issues*, pp. 309–314, 2013.
- [9] K. Shikha and K. Bahl, "Software Testing Tools & Techniques for Web Applications," *Int. J. Eng. Tech. Res. IJETR*, vol. 3, no. 5, pp. 315–318, 2015.
- [10] K. Sheoran and O. P. Sangwan, "An Insight of software quality models applied in predicting software quality attributes: A comparative analysis," 2015, pp. 1–5.
- [11] S. Mehta and H. Bhambri, "Software Testing-A Brief Study," *IITM J. Inf. Technol.*, p. 27.
- [12] A. Bansal, "A Comparative Study of Software Testing Techniques," *Int J Comput Sci Mob Comput*, vol. 36, no. 6, pp. 579–584, 2014.
- [13] O. S. Gómez, K. Cortés-Verdín, and C. J. Pardo, "Efficiency of software testing techniques: A controlled experiment replication and network meta-analysis," *E-Inform. Softw. Eng. J.*, vol. 11, no. 1, 2017.
- [14] S. Nidhra and J. Dondeti, "Black box and white box testing techniques-a literature review," *Int. J. Embed. Syst. Appl. IJESA*, vol. 2, no. 2, pp. 29–50, 2012.
- [15] S. Acharya and V. Pandya, "Bridge between Black Box and White Box-Gray Box Testing Technique," *Int. J. Electron. Comput. Sci. Eng.*, vol. 2, no. 1, pp. 175–185, 2012.
- [16] T. Hussain and S. Singh, "A Comparative Study of Software Testing Techniques Viz. White Box Testing Black Box Testing and Grey Box Testing," *IJAPRR ISSN*, pp. 2350–1294, 2015.
- [17] S. R. Jan, S. T. U. Shah, Z. U. Johar, Y. Shah, and F. Khan, "An innovative approach to investigate various software testing techniques and strategies," *Int. J. Sci. Res. Sci. Eng. Technol. IJSRSET Print ISSN*, pp. 2395–1990, 2016.
- [18] A. Sethi, "A REVIEW PAPER ON LEVELS, TYPES& TECHNIQUES IN SOFTWARE TESTING," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 7, 2017.
- [19] A. B. AL-Badareen, M. H. Selamat, M. A. Jabar, J. Din, and S. Turaev, "Software Quality Models: A Comparative Study," in *Software Engineering and Computer Systems*, vol. 179, J. Mohamad Zain, W. M. bt Wan Mohd, and E. El-Qawasmeh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 46–55.
- [20] P. Nistala, K. V. Nori, and R. Reddy, "Software Quality Models: A Systematic Mapping Study," in *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, Montreal, QC, Canada, May 2019, pp. 125–134, doi: 10.1109/ICSSP.2019.00025.
- [21] H. Rashidi and M. Sadeghzadeh Hemayati, "Software Quality Models: A Comprehensive Review and Analysis," *J. Electr. Comput. Eng. Innov.*, vol. 6, no. 1, Jan. 2018, doi: 10.22061/jecei.2019.1076.
- [22] J. P. Miguel, D. Mauricio, and G. Rodríguez, "A Review of Software Quality Models for the Evaluation of Software Products," *Int. J. Softw. Eng. Appl.*, vol. 5, no. 6, pp. 31–53, Nov. 2014, doi: 10.5121/ijsea.2014.5603.
- [23] O. Gordieiev, V. Kharchenko, N. Fominykh, and V. Sklyar, "Evolution of Software Quality Models in Context of the Standard ISO 25010," in *Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. June 30 – July 4, 2014, Brunów, Poland*, vol. 286, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, Eds. Cham: Springer International Publishing, 2014, pp. 223–232.
- [24] M. W. Suman and M. Rohtak, "A comparative study of software quality models," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, pp. 5634–5638, 2014.
- [25] R. K. Chauhan and I. Singh, "Latest research and development on software testing techniques and tools," *Int. J. Curr. Eng. Technol.*, vol. 4, no. 4, pp. 2368–2372, 2014.
- [26] M. Atifi, A. Mamouni, and A. Marzak, "A comparative study of software testing techniques," 2017, pp. 373–390.
- [27] S. S. Ghuman, "Software Testing Techniques," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 10, pp. 988–993, 2014.
- [28] I. Singh, "Different Software Quality Model," *Int. J. Recent Innov. Trans Comput. Commun.*, vol. 1, no. 5, pp. 438–442, 2013.
- [29] D. Samadhiya, Su-Hua Wang, and Dengjie Chen, "Quality models: Role and value in software engineering," in *2010 2nd International Conference on Software Technology and Engineering*, San Juan, PR, USA, Oct. 2010, p. 5608852, doi: 10.1109/ICSTE.2010.5608852.
- [30] B. Singh and S. P. Kannoja, "A Review on Software Quality Models," in *2013 International Conference on Communication Systems and Network Technologies*, Gwalior, Apr. 2013, pp. 801–806, doi: 10.1109/CSNT.2013.171.
- [31] al-Farabi Kazakh national university, N. K. Uteliyeva, E. E. Ismail, Almaty University of Power Engineering and Telecommunications, D. F. Akhmetov, and Rakuno Gakuen University, Hokkaido, Japan, "Comparative analysis of models of quality of software tools," *J. Math. Mech. Comput. Sci.*, vol. 104, no. 4, pp. 85–94, 2019, doi: 10.26577/JMMCS-2019-4-m9.
- [32] A. B. Tomar and Vilas. M. Thakare, "A Systematic Study Of Software Quality Models," *Int. J. Softw. Eng. Appl.*, vol. 2, no. 4, pp. 61–70, Oct. 2011, doi: 10.5121/ijsea.2011.2406.