Software Testing Based on Software Product Quality Metrics (SPQM)

2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS) | 979-8-3503-9737-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICACCS57279.2023.10113076

Vijay SivaKrishna
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
mvijaysiva2000@gmail.com

Pasala Saketh Kumar
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
sakethkumar523@gmail.com

Jhansi Jhaveri
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
jhansijhaveri532003@gmail.com

Lakshmi Sai Akanksha
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
tanneeruakanksha@gmail.com

P.Venkateswara Rao
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
pvrao@kluniversity.in

N.Raghavendra Sai
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation,Vaddeswaram, AP, India
nallagatlaraghavendra@kluniversity.in

Abstract—Producinghigh-quality products has emerged-as a crucial component of corporate success in today's marketplace. In this regard, establishing and holding to software product to assess the present quality standardto environment and promote system to improvethe quality of the productit significance. Testing software is an essential step in the software development process. This technique is more accurate and effective when it is automated. Emerged new testing tools for automated testing. The selection of the proper tools has become a challenging and confusing activity as a result of the abundance and variety of testing instruments. The current study attempts to assess the current research areas and trends on this subject that have appeared in the literature over the previous twelve years. Based on their titles and abstracts, 50 conference and article papers on SPQM published in the interval 2010 and 2022 were subjected to Systematic Mapping (SM) research. The findings are presented using a combination of diagrams, written explanations, and mind-mapping techniques, which include a trend map for the period from 2010 to 2022, details about the field and measurement techniques, identified areas for improvement, and consistency across sources such as conference papers, journal articles, and internationally recognized quality models. Future research that aims to advance this important topic may build on the outcome of this study.This research may provide a starting point for further investigations meant to advance this vital area of study

Keywords—Software quality testing; SPQM; the life cycle of Software testing, product quality; test plan; test cases; test execution; test coverage; defects; bug reports; test automation.

I. INTRODUCTION

The process of enhancing product quality through the use of Software Quality Metrics (SQM) is defined with function that inputs software data and outputs a number indicating the extent to which a particular characteristic of the software has an effect on its quality [1]. Producing high-quality products has emerged as a crucial aspect in today's competitive business world, ensuring rivals' long-term success. Determining and implementing the use of to determine the current quality levels using Software Product Quality Metrics (SPQM).

Software testing entails a variety of procedures carried out with the specific purpose of identifying software flaws. It validates and confirms if the program or product is operating appropriately and free of any bugs or mistakes that could cause a defect. To ensure dependability and quality of the software the earlier phase errors will help [2].This study seeks to apply the SM approach to a specific set of papers(May be conference and articles) from the previous ten years in accordance with this principle. As noted in their names and abstract parts, these studies specifically focus on SPQM.

Metrics and assessments of measurement findings serve as reliable gauges of the effectiveness of organizational operations and/or product quality [3]. For example, an emergency plan for intervention will be immediately activated if the measurements detect a deviation below the threshold, indicated by a red signal. [4]. These indicators serve as a safety net for businesses in such circumstances. Measuring the existing condition is essential for being able to compete in the industry since it allows undesirable scenarios to be avoided and ongoing progress to be improved. This is so important that the order of map is constructed using the incorrect set of measurements, the users will almost certainly be misled [5].In other words, the pointer will likely display inaccurate data If the metrics are not defined properly to achieve the goal using calibrated and validated metrics. In situations where, for instance, the charts show that operations are going well, the business may have filed for bankruptcy.

According to the 2018-2019 World Quality Report, customer satisfaction was reported to be only approximately 40%[6], and less than 40% of software faults are discovered before being delivered to end users. Additionally, there has been a 40% improvement in product/software quality.[7] A crucial procedure for improving the quality of processes and products is the mapping of SPQM. As stated in the same paper, the worldwide standards and software process models treat the existing state of the SPQM with the highest care.

Project Management Body of Knowledge (PMBOK) and Capability Maturity Model Integration (CMMI) are employed to systematically plan, develop, and manage software engineering projects and systems.

Additionally, [8] Outbid et al examine the necessity for software quality and provide guidelines to assess the software quality of European initiatives. [9] by taking into account software quality models and standards, particularly ISO 25010.

The main objective of this paper, as supported by the goal, as evidenced by the aim of the literature review is to determine the current state of knowledge in Software Product Quality Management (SPQM) in order to guide future research in this field. An activity is a worthwhile experience that offers a thorough overview of a research topic as well as transferable research skills [10].Such a study also offers a methodical and impartial approach to determining the kind and volume of empirical study facts that are accessible to address a specific research examination [11].

Upon reviewing the literature on Software Product Quality Management (SPQM), only a limited number of systematic literature studies were found to be focused on object-oriented software metrics, leaving a gap in knowledge for practitioners and researchers in the area of Software Quality Assurance. Additionally, it has been noted that the majority of earlier SM research often only focus on a very limited area of SPQM.

Recently, [12] backed additional research in this area.

To this purpose, the primary research topic focuses on learning about and examining how the SPQM studies have evolved over the past ten years, as well as what the patterns, trends, and gaps are.

around here. Researchers, SQA specialists, and business divisions in charge of software processes and quality would all benefit greatly from this.

With the use of the SM approach,the goal is to work on to analyze SPQM data from 2009-2019 using the SM approach, in order to identify patterns, trends, and gaps in the field[13]. This approach provides a trend map that highlights the relationships between different terminologies within the software engineering industry is connected to all levels.Other innovative aspects of this study include the utilization of the Open AIRE digital library and the representation of the trends using the mind mapping technique.

The remaining article sections are organizedare given below. In Section III, the research methodology, research questions, criteria for selecting papers, an overview, standards for assessing the quality of studies, and data extraction are presented. Moreover, it comprises details concerning software quality models and metrics, along with relevant data for studying SPQM.RQ responses are accompanied by visual images in Section IV. Section V includes the mind mapping of the SPQM trend as well as a discussion of these trends by contrasting them with other publications, standards, and quality models. Four forms of validity are used in Section VI to discuss the risks to the results' validity. By highlighting the current research scenario, changes in the field over the past ten years, and improvement potential in SPQM, the conclusion, outlines the limitations and makes recommendations for additional research[14].

II. Background And Related Work

Since it has been a significant subject of study for so long, there are many different methods of software testing that have been put forth in the literature. Software product quality measurements are the foundation of one of the most popular techniques to software testing. The strength and weaknesses of software products can be learned via quality metrics, which are used to assess and measure the quality of software products. There has been a significant amount of research in the area of software testing based on software product quality metrics. Numerous quality criteria, such as functionality, dependability, efficiency, maintainability, and usability, have been presented by researchers for software products. The specific needs of the software product and the objectives of the software organization will determine the quality metrics that should be employed. Many different software testing techniques, including functional testing, performance testing, and security testing, have been proposed by researchers. The specific needs of the software product and the objectives of the software organization determine the testing methodology to be used. For assessing the data gathered on software product quality indicators, researchers have suggested a number of approaches, including statistical analysis, regression analysis, and machine learning techniques, among others. The specific needs of the software product and the objectives of the software organization determine the analytical approach to be used. Researchers have suggested a number of approaches, including graphs, tables, and other visual aids, for communicating the outcomes of software testing based on metrics for the quality of software products. The specific needs of the software product and the objectives of the software organization determine the reporting mechanism to be used.A class-level metric called DIT (Depth of Inheritance Tree) calculates how deep an inheritance hierarchy is for a given class. It is determined by calculating the number of inheritance levels that exist between the class and the inheritance tree's root. The class is seen to be more complex and has a higher DIT value when the inheritance tree is deeper. A class that inherits properties and methods from numerous other classes may have a high DIT value, which might be a sign that it is too complex and may be challenging to maintain. It can also be an indication of a bad design, where a class should have been split up into several classes with more distinct roles. Examples of class-level metrics include LCOM and DIT. Defect Density (DD) is an SPQM used to assess how well defects are found throughout a project's lifespan [4]. The number of known flaws is divided by the size of the product to get the Delivered Defect Density (DDD) statistic. The test-level metrics are DD and DDD[15].

Furthermore, the following are examples of SPQM and outlined in ISO 25010 and ISO 25012: The reliability-metric that represents the average duration between system failures is referred to as the MTBF (Mean-Time Between Failures). MTBF is an illustration of the SPQ's Reliability quality characteristic and can be seen as a system-level

statistic. The Customer product perception and satisfactionrepresent serve as an example of the Satisfaction quality-attribute of Qiana[4].Inherent data quality is characterized by five factors are Accuracy, Completeness, Consistency, Credibility, and Correctness. On the other hand, system-dependent data quality traits are Availability, Portability, and Recoverability[16].One example of system-level metrics are measurements that reveal details about a computer system or network's general performance and behavior. Numerous pieces of data, including CPU, memory, disc, network, and system uptime consumption, might be included in these measures.

System-level measurements include, for instance: CPU utilization: The amount of time that is now being used by the CPU. Memory utilization: The proportion of total memory that is being used. I/O for discs: The rate of read and write operations. Bytes are sent and received over a network in a second. Uptime: The length of time the system has been operational following a restart. These indicators can be used to locate performance bottlenecks and resolve them, keep track of system health and resource utilization, and aid with capacity and scalability planning for the future[17]. For monitoring and gathering system-level metrics, a variety of tools are available, including the built-in utilities of the operating system, open-source monitoring tools, and paid monitoring software. Only research [18] in Table 1 compares to the current study, but it only looks at internal metrics for object-oriented software from 2004 to 2013.

Additionally, although unrelated to the current work, the paper [19] is similar. Software quality models are the subject of SM in [19], which focuses on the model's inclusive aspects and architectural quality supports. The same study takes into account requirements for software quality models and the architecture that supports quality models in terms of publishing trends, research genres, and common meta-model components. There is also no information regarding the SPQM as it is described in the current study. They, therefore, differ from our study in terms of scope, methodology, and RQs. The SPQMs' varied and extensive content led to the selection of the 2009–2019 time period for in-depth analysis[20].

The current study utilizes SPQM, a methodology chosen from the key area of knowledge within the software Engineering Body of Knowledge[21]. TheSPQMs are gaining significant attention according to recent trends in the field, as a result, these models can be continuously improved to align with international standards and best practices. [6]. From the perspective of SPQM, this study is complementary. For better understanding the reasoning presented here, it is therefore anticipated that they are at least somewhat familiar with the terminology of software quality and SPQM[22].

III. RESEARCHMETHOD

Here, is the study is based on the software methodology concepts and ideas presented in some articles and research papers. The research process includes developing a methodology, setting goals and criteria for quality assessment, conducting literature search and selecting papers, extracting and synthesizing data, evaluating software testing tools, discussing findings, assessing potential validity threats, drawing conclusions, identifying limitations, and making recommendations for future work.

IV. RESEARCH METHODOLOGY

The Quality metric definition to be utilized in software testingis the first stage in the approach. Functionality,dependability,efficiency, maintainability, and usability are a few examples of these. The selection of the software items to be tested is the next step[23]. This can be accomplished by choosing a software product sample that is representative or by choosing software products that are known to have quality problems. Testing the chosen software solutions is how the quality metrics data isgathered. Functional testing, performance testing, and other forms of testing may be included here as necessary[24]. The collection and analysis of data related to quality metrics are involved in software product quality evaluation. This can entail figuring out mean values, standard deviations, and other acceptable statistical measurements. The analysis's findings are presented in a thorough and insightful manner. Graphs, tables, and other visual aids may be used in this, if necessary. Based on measures for measuring the quality of software products, the technique described in this paper offers an organized approach to software testing[25]. The approach entails the following steps: choosing software products for testing, specifying the quality measures to be used, gathering data on the quality metrics, evaluating the data, and reporting the results. This methodology can assist software companies in efficiently assessing the software product qualityand implementing any necessary improvements[26].

V. RESEARCH QUESTIONS

The software methodology study's research questions are separated into three categories:

A. What are the challenges facing software testing in 2023?

Software testing is a crucial component of the creation of software that guarantees the dependability and quality of software products. Software testing's goals include finding bugs and flaws in the program and making sure it complies with all criteria[27]. Software product quality metrics, which are used to assess the level of software output, must be employed throughout software testing.In this research study, In this context, we will discuss the use of software testing metrics for assessing software product quality, as well as the challenges associated with this processthat software testing will face between 2010 and 2022[28].

The complexity of software products is one of the biggest issues facing software testing in 2023 [50]. Software products have become more complicated as a result of the advancement of new technologies and the rising demand for them, making it more challenging to adequately test them. Another issue is the growing need for quicker software development cycles, which may result in less time being allocated for testing. The quality of software products may suffer as a result, and there may be more flaws overall[29].

B. How can software testing is based on software product quality metrics help to overcome these challenges?

By offering a systematic method to testing that guarantees all components of the software are properly evaluated, software testing based on metrics for the quality of software products can aid in overcoming the difficulties facing software testing in 2023[30]. The use of quality metrics of software product can also aid in reducing the time required for testing by allowing testers to focus their attention on the most vital parts of the software.Additionally, by identifying regions of the software that are vulnerable to errors, testers can concentrate their efforts there and raise the software's quality. This is made possible by the use of software product quality metrics[41].

C. What are the most commonly used quality metrics for software testing in software product?

Code coverage, cyclomatic complexity, reliability, and usability are the four quality criteria for software products that are most frequently employed in software testing. In order to detect sections of the code that are not covered by tests, code coverage analyses the amount of code that is run during testing[42]. Cyclomatic complexity analyses the code's complexity and identifies sections that are vulnerable to errors. Usability measures how easily a program can be used and how well it can accommodate the needs of its users, whereas reliability measures a program's capacity to carry out its intended functions without failing.[31-40]

Software testing based on measurements for the quality of software products is a crucial component of software development and can aid in overcoming the difficulties that software testing will face in 2023[43]. Software product quality measures can be used to guarantee that the final products are of a high caliber and adhere to the specifications [48]. Software testing can be carried out more successfully and software products can be created to a better standard by employing these measures [49].Quality Metrics for Software Products is shown in Fig. 1. Software Testing Methods is shown in Fig. 2. Software Quality indicator & Metrics for Testing is shown in Fig. 3.

Metric	Definition
Functionality	how well a piece of software complies with its specifications and performs as planned
Reliability	the capacity of a software programme to carry out its specified tasks flawlessly.
Efficiency	The amount of computational resources (e.g., time and memory) required by a software product to perform its intended functions.
Maintainability	the simplicity with which a piece of software can be updated or maintained over time.
Usability	the simplicity with which its target customers can use a piece of software.

Fig. 1. Quality Metrics for Software Products

Methods	Definition
Functional testing	Testing that verifies that a software product performs its intended functions as specified.
Performance testing	Testing that verifies that a software product performs its intended functions efficiently and within acceptable performance bounds.
Security testing	Testing that verifies that a software product is secure and protects against potential security threats.
Other testing methods	Other testing methods that may be appropriate depending on the specific requirements for the software product and the goals of the software organization.

Fig. 2. Software Testing Methods

Indicator/Metric	Problem Identified
Functionality	Inadequate specification of software requirements leading to incorrect functionality
Reliability	Lack of consideration for edge cases and unexpected inputs leading to system failures
Efficiency	Poor design and algorithms leading to slow performance and increased resource usage
Maintainability	Code complexity and poor documentation leading to difficulties in maintaining and updating the software
Usability	Lack of user-centered design leading to confusing and difficult to use interfaces

Fig. 3. Software Quality indicator & Metrics for Testing

Software testing tools results:

Fig. 4show the key Issues Facing Software Organizations in 2023 in Software Testing Based on Software Product Quality Metrics.

Issues	**Description**
Lack of standardization in quality metrics	Lack of Different software organizations use different quality metrics, leading to confusion and difficulty in comparing the quality of different software products. Standardization in quality metrics
Inconsistent application of quality metrics	Software organizations may use quality metrics in an inconsistent manner, leading to differences in results.
Difficulty in incorporating customer feedback	Software organizations may have difficulty incorporating customer feedback into their software product quality metrics.
Inadequate resources	Software organizations may not have the necessary resources, including personnel, time, and budget, to implement software product quality metrics in a comprehensive and effective manner.

Inadequate training for software testing professionals	Software testing professionals may not have the necessary training to effectively use software product quality metrics in their work.

Fig.4. the key Issues Facing Software Organizations

VI. DISCUSSION

The present study targeted to enhance the quality of software testing tools in relation to their usability in an existing software product (RQ1) through evaluation using commonly accepted criteria[44].The three most criteria for assessing the excellence of security, reliability, andMaintainability are among the key quality attributes of software products.Furthermore, the most commonly employed quality attributes for evaluating software include Freedom from Effectiveness,Risk, and Satisfaction[45].An analysis of conference and article papers published in between 2000 and 2014 revealed that the three most significant quality factors were Understandability, Reliability, and Reusability.However, in a more recent analysis, the top three factors were found to be Maintainability, Reliability, and Security[46]. It is worth noting that Reliability is consistently considered a crucial aspect in determining the overall quality of a software product.

VII. RECOMMENDATIONS

Standardize quality metrics: Software organizations should work together to develop and adopt a standard set of quality metrics to ensure consistency and comparability in software testing[47].

Ensure consistent application of quality metrics: Software organizations should develop clear guidelines for the consistent application of quality metrics in software testing.Provide adequate resources

VIII. CONCLUSION

As part of our observations, we conclude that Software testing is an essential step in the process of developing software with help of software testing tools. Using software product quality metrics methods can help enhance thequality of a software product by helping to identify and fix defects and find a solution. In this process of observations, we have done literature work and noted all the pros and cons of the process of improving product quality. And also we worked on some software testing tools to figure out the problems how to identify the solve the problem in the product. we got to know that Some of the researchers can explain the techniques and predict the outcome and some of them show the pathway to solve the problem in the product to improve the quality and performance of the software product. we have to identify various steps and methods and gain extensive knowledge. This product quality improvement can extend in the future with the help of software testing tools and techniques.

REFERENCES

[1] Luis F. de Lima, Matheus C. Horstmann, David N. Neto,” On the Challenges of Automated Testing of Web Vulnerabilities, 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises 2020 IEEE.

[2] "ISO/IEC/IEEE 29119-2:2013[S]. Switzerland: ISO", "ISO/IEC/IEEE", Software and systems engineering-Software testing-Part 2: Test processes:, 2013.

[3] A.W. William and B. Venkataraman, "Some observations on software quality", School of computer and Information Sciences University of south Alabama mobile a136688.

[4] Software Quality Metrics for Object Oriented System Environments, [online] Available: http://satc.gstc.nasa.gov/SATC/PAPERS/OO_TECH/oo_tech.html.

[5] M. A. Jeusfeld, C. Quix and M. Jarke, "Design and Analysis of Quality Information for Data Warehouses", Conceptual Modeling – ER '98, vol. 1507, pp. 349-362, 1998.

[6] "International Standard Organization . (2001). ISO/IEC 9126–1: Software Engineering - Product Quality", vol. 1.

[7] R.W. Lissitz and S.B. Green, "Effects of the Number of Scale Points on Reliability: A Monte Carlo Approach", J. Applied Psychology, vol. 60, no. 1, pp. 10-13, 1975.

[8] V. Zeithaml, L. Berry and A. Parasuraman, "The Nature and Determinants of Customer Expectations of Service Quality", Journal of the Academy of Marketing Science, vol. 21, no. 1, pp. 1-12, 1993.

[9] Fei Li, Yumin He and Chaoying Wu, "Research on project quality management of software company based on CMMI[J]", Journal of Beihang University (SOCIAL SCIENCE EDITION), pp. 10, 2012.

[10] ISO/IEC 15288 – Software and System Engineering – Life Cycle Management – System Life Cycle Processes, 2002.

[11] J. A. McCall, P. K. Richards, and G. F. Walters, "Factors in Software Quality. Volume I. Concepts and Definitions of Software Quality:," Fort Belvoir, VA: Defense Technical Information Center, 1977.

[12] S. T. Acuña, M. Gómez and N. Juristo, "How do personality team processes and task characteristics relate to job satisfaction and software quality?", Information and Software Technology, vol. 51, no. 3, pp. 627-639, 2009.

[13] Kan S., "Metrics and Models in Software Quality Engineering", Addison-Wesley, Reading, Massachusetts, 1996

[14] Wang Dong, “Artificial Intelligence for Web-based Educational Systems”, Advances in Intelligent Systems and Technologies, pp. 055-065. 2022. doi:10.53759/aist/978-9914-9946-0-5_7

[15] Ceren Ergenc and Yifei LI, “A Review of Art and Real World Applications of Intelligent Perception Systems”, Advances in Intelligent Systems and Technologies, pp. 076-086. 2022. doi:10.53759/aist/978-9914-9946-0-5_9

[16] Bystrický, M., Vranić, V.: Use case driven modularization as a basis for test drivenmodularization. In: Federated Conference on Computer Science and Information Systems(2017).

[17] V. Rastogi, “Software Development Life Cycle Models-Comparison , Consequences,” Int. J. Comput. Sci. Inf. Technol., 2015

[18] Mustafa, K. M., Al-Qutaish, R. E., &Muhairat, M. I. (2009). Classification of software testing tools based on the software testing methods, in 2009 International Conference on computer and Electrical Engineering, ICCEE 2009, 1, 229–233.

[19] Kyngäs, H.; Mikkonen, K.; Kääriäinen, M. The Application of Content Analysis in Nursing Science Research; Springer Nature: Berlin/Heidelberg, Germany, 2020.

[20] Jing Wang, Dayong Ren,” Research on Software Testing Technology under the Background of Big Data”, 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference IMCEC 2018.

[21] Nils Wild, Horst Lichter, Peter Kehren,” Test Automation Challenges for Application Landscape Frameworks” ,2020 IEEE

[22] K. Solanki, Y. Singh, S. Dalal, and P. R. Srivastava, ‘‘Test case prioritization: An approach based on modified ant colony optimization,’’ in Emerging Research in Computing, Information, Communication and Applications. 2016.

[23] R. Liu et al., "Region-convolutional neural network for detecting capsule surface defects", BoletinTecnico, vol. 55, no. 3, pp. 92-100, 2017.

[24] "A Unique Technique to Handle the Complexity and Improve the Effectiveness of Test Cases in Software Testing", Journal of Innovative Technology and Education, vol. 2, 2015.
[25] WSQB17: Waseda Software Quality Benchmark, [online] Available: http://www.washi.cs.waseda.ac.jp/wsqb/.
[26] Prowell, S.J., Poore, J.H. Sequence-based software specification of deterministic systems. Software: Practice and Experience 1998; 28(3):329-344.
[27] Standard Glossary of Terms used in Software Testing" (PDF). Version 3.1. International Software Testing Qualifications Board. Retrieved January 9, 2018.
[28] Saja Khalid Alferidah, Shakeel Ahmed," Automated Software Testing Tools", International Conference on Computing and Information Technology, University o f Tabuk, Kingdom o f Saudi Arabia. Volume: 01, Issue: ICCIT- 1441, Page No.: 183 – 186 Sep. 2020.
[29] T. J. Naidu, N. A. Basri, and S. Nagenthram, "SAHI vs. Selenium: A comparative analysis," 2014 International Conference on Contemporary Computing and Informatics (IC3I), pp. 967-970, 2014
[30] ASMA J. ABDULWARETH AND ASMA A. AL-SHARGABI, "Toward a Multi-Criteria Framework for Selecting Software Testing Tools" Qassim University, Buraydah, Saudi Arabia 2021 IEEE.