NUML International Journal of Engineering and Computing

Volume: 3 Issue: 1

https://numl.edu.pk/journals/nijec

Print ISSN: 2788-9629

E-ISSN: 2791-3465

DOI:https://doi.org/10.52015/nijec.v3i1.59

Systematic Literature Review on Methodologies for Improving Software Quality in Software Development Process

Inam-Ul-Haq[1], Waseem Abbas[2], Mehran Shakoor[3], Wasi Haider Butt[4]

Department of Software Engineering[1,2,3,4]
College of Electrical & Mechanical Engineering, NUST Rawalpindi,
Pakistan[1,2,4] Abasyn University, Islamabad Pakistan[3]
iuhaq.cse19ceme@ce.ceme.edu.pk[1], wabbas.cse19ceme@ce.ceme.edu.pk[2], mehranshakoor007@gmail.com[3],
wasi@ceme.nust.edu.pk[4]

Submitted	**Revised**	**Published**
1-July-2024	03-Dec-2024	30-Dec-2024

Abstract

In the competitive software industry, product quality is a critical factor for a company's reputation. The quality of a product is directly linked to customer satisfaction, and it is measured by how well the product meets customer expectations. Achieving high-quality standards requires effective policies and foundational principles. Growing economies continue to improve their product quality daily, aiming to enhance customer satisfaction. Every organization strives to design software that best meets customer needs. Although fundamental principles and policies have been established to improve quality, various challenges and reasons contribute to poor product quality. For this study, we adopted a quantitative research approach, which focuses on data measurement and evaluation. This paper addresses the various issues associated with these challenges, explaining several techniques and alternative solutions that can lead to improved product and software quality. Additionally, we conclude that while systematic and incremental approaches are effective, their limitations arise from the difficulty in accommodating changes once requirements are fixed. Ultimately, all approaches have a significant impact on enhancing software quality.

Keywords—Software Enhancement Mechanism, Software Quality Assurance Parameter, Quality tools, Capability Maturity Model Integration *(CMMI)*

1. Introduction

Managing quality of software becomes essential part of project management. During the development of every project or software, quality of the software is the most important factor. For achieving better position and prominence in IT field, organization needs to deliver software products that have high quality.

Quality of a product can be rectified before the product or software is delivered. After delivery of software to the clients, if someone wants to rectify the software then correction will take too much time and it will be too much expensive. Quality plans should be developed in early and followed at every step during the development of software product. Capability Maturity Model Integration model will be used if you are working on the small working products and for large software products International Organization for Standardization ISO model should be used. We do not use ISO model for small software's projects because they will become very costly for them. Problem of quality of software becomes increasing day by day within different organizations as well in countries. These problems describe interactions among teams and companies or societies among them [1]. If a defect occurs, then it would not be easy to detect that defect or error in the design just because of the digital systems that would be used are not continuous and variables will not be enough to find the errors. Errors that arise in the software's can be detected through testing and after detecting the error in software it will become possible to correct that error in order to achieve good quality of product. If an issue is detected, then with the help of testing approach we will make predictions about how much cost will be required to remove that defect and to maintain the software quality. If the error is detected in early stages during requirement elicitation phase or design phase that cost for removing that error will be minimum. If testing is performed very effectively and carefully then it will reduce the chances of errors and hence quality of product will be good [2]. Several categories of testing and methods can be used along with several models. Every model has some advantages as well some disadvantages.

It does not matter that which testing technique is used, but the thing that matters a lot is that every implementation phase must be tested according to the quality planning. Detected defects can be of any category. Some of the defects may be present in the documentation some may be present in code and design. Testing the product just once may not be enough. Doing same testing technique for every kind of software may not help you to detect different defects. Outsourcing projects that are developed for delivering outside the organization must have good quality because third party who requires software do not compromise on the quality and experience. Company must resolve such kind of problems [3]. So, for resolving these issues planning should be required in order to attain a good quality software for outsourcing. Software quality

Quality of software do not depend only to build and follow quality plans. It also depends on managing the quality, verifying quality and product verification. People who require software support in their work always think that software product will help them in doing their job properly. Quality of product can be characterized by some features of software like whether the code of the software can be used to build any feature for other software, how long the software will work without any error, whether the software is user friendly or not etc. [4].

Alternative evolution approaches are used to develop peer objectives of product. Members who are responsible for developing the features must have experienced and trained. Different tasks should be assigned to the team members and at last quality of product should be checked by quality assurance team member. Checklist should be made to check possible defects and to identify that no necessary feature should be absent [5].

In the reviewing phase, moderator will write down defects. Participant of team must be experienced, fit for the project and can work like a team in the project [6]. Communication gaps between members within the groups arises complex problems. Some of the reasons are difference of languages and terminologies of members in the team. If team members belong to different areas, then it will become also a factor of communication gaps, so team members will not be able to communicate properly. This could lead towards severe issues. Time zone differences and type of procedures also affects the quality of product [7]. Management will suffer if they have weak coordination's and planning among other members within the team. If there is a need for changes in planning or in schedule, then the result of changes will delay the finish time of product as well may be some errors occurs due to these changes [8]. Quality of software product cannot be obtained by using one method, for good quality product combination of several methods will be used. Testing of software or product is not only a solution. There is a special need for solving different bugs in database [9]. Quality product can be obtained through the systematic procedures and by setting strong standards [10]. Distributed development of software is one of the special methodologies that has very strong effect on development of software [11]. If risk minimization techniques and documentation of product is not accurate then there are chances of project failure. Quality assurance will increase benefits and it provides chances to reduce the cost of product, if it is done properly [13]. It is not enough to perform quality reviews in order to check and improve quality of product, each phase of product must have a quality process. Quality of product will improve if each phase has a quality process [14].

Managing quality is one of the critical things. Quality is most important factor for complex and big projects. Project progress can be traceable from project documentation, quality of documentation matters a lot. We must perform some quality attributes to increase product quality. Accuracy of data is also an important aspect of quality. Collect only relevant data that is related to the product. Before making questions for surveys, you must know right answers to those questions and write those answers on paper. Do not think that you can remember them. If possible then take advice from experts. There are several error detection and error prevention approaches used in implementation of product by software organizations to explain the issues [15]. Organization or company requires to clearly explain quality standards before making prediction about them [16]. Structure of organization leaves very strong effects on quality of product. If some quality preplans exist that can be taken on later then quality of product will increase. Errors should be removed as early as possible by using different error detection and error removal approaches this will be best for quality of product. Progress and communication should be there between members of team. Following are the techniques that can resolve quality problems to develop quality software.

Paper is formulated in different sections: first section explains topic introduction, second section describes the methodology, third section describes previous related work, fourth section describes or shows a comparison among techniques needed to resolve different issues, fifth section explain result of comparisons and at last there is conclusion.

2. Methodology

The objective of this paper is to perform a comprehensive review of improving software quality approaches. For this purpose, quantitative research approach is used. The articles selected for this research included different research approaches and methodologies and every approach has a certain impact on the results. We examined almost hundred papers. From those, we have selected paper related to project management and then we further categories them and select papers related to the quality of software and quality assurance.

A. Data Source and Search Strings

("Quality" OR "Standard" AND "Attributes OR Parameter") AND ("Improving" OR "Enhancing" AND "Technique" OR "Methodology") in Software Development.

TABLE 1: JOURNAL AND CONFERENCE

1	***IEEE Explore***	https://ieeexplore.ieee.org/
2	***Research Gate***	https://www.researchgate.net/
3	***Science Direct***	https://www.sciencedirect.com/
4	***ACM Library***	https://dl.acm.org/
5	***Springer***	https://link.springer.com/
6	***Semantic scholar***	https://www.semanticscholar.org/

3. Literature review

Different approaches for producing quality product are explained in this section. This paper includes explanation related to different issues that arises in quality assurance.

For improvement of software quality, different kinds of software are used, and project team should work together to perform their operations. There are several ways for the software quality to be improved. Paper includes some big issues regarding quality like political conflicts, cost, requirement engineers, time and unrealistic deadlines. For these problems, an approach called CMMI model was proposed, some other approaches as well were also introduced like specialized software quality assurance members in team, requirement engineers should be domain experts and they must neglect internal politics among each other with in the team. Specialized workers have experience about how to handle different issues and problems. When team member gets accurate and complete picture of requirements then requirements will become easily understandable. A lot of resources are required for CMMI model, but it will remove the errors and increase the quality of software. Secondly experienced team has a goal that they will try to find errors as early as possible and then correct and reduce them. As quality is the important aspect, team will have to bear pressure, but it will be beneficial to them [1].

Different types of models are also being used to improve quality of product. Several quality factors and different problems that occurs in product to reduce quality are discussed in the paper. For example, errors in design document, requirement document or baseline requirement document, errors in test plan or requirements may not be properly specified due to non-standardization. Some management problems were also highlighted in previous research papers. SDLC models can be used, like Spiral Model, Rad Model, Waterfall Model and Prototype Model. There are several testing approaches like black box testing in which we only check the functionality of the system, white box testing in which we check or test the internal working of code. These techniques are used in the above models. Previous research describes that to develop quality product, proper testing must be done in order to reduce errors. Testing may be used in every phase of development [2].

In systematic methodology research paper author describes problems related to the outsourcing of projects and discover different issues like cost of development may not be reconfirmed, type of physical device such as hardware that an help in building product cannot be identified, requirements may not be accurate and complete. In order to solve these problems many tools exist. We will identify the most effective one among these tools. In systematic approach author provides solution to several problems and predict that product should be tested using static methods and dynamic methods. Once all the errors are resolved then freeze the requirements to ensure quality. With the help of these approaches large and sophisticated software projects were implemented. If the base requirements are freeze then adding more features for betterment of project will be restricted. Hence quality of product will suffer [3].

A. Divide and Conquer

Researcher Parnas describe in a paper the way to identify and reduce issues of a problem to increase the quality of product. Main technique used by him was divide and conquer. He described a method in the research, if you want to buy a software which is free of cost then quality will be better as compared to others. He said that quality will be better and improved based on the feedback. This approach will require time for finding issues, so that issues can be resolved. This approach can be used to identify quality of software product rather than a process [4].

B. Automation Techniques

The author discussed the errors arises because of unclear communication issues and people on the other end will understand different meaning of the same thing. Due to this, chances or errors will increase. Author suggested use of automation techniques that depends upon committed algebra and statistical analysis. These approaches help in increasing the quality and provides faster delivery of product by reducing the communications that are not clear. These mathematical ways are not suitable because these ways are not testing approaches. With these approaches program execution can be accurate, but we are unable to say that every requirement is fine and accurately met [5].

C. Software Quality Analysis (SQA) Metric

Hurdles in way of developing quality products are the issues and problems. For example, if someone used the software but he did not give the response or feedback. Requirements might be unclear and incomplete. Developers are sometimes unable to understand customer needs or what the client is thinking about the product. Duplication of requirements may arise in the project. Author suggests that checklist is the method used to handle these issues. Checklist supports SQA Metrics and quality assurance. It can also be used to detect defects. While in verification requirements that are written for implementation process are satisfied. Duplication can be removed by doing efforts [6].

D. Capability Maturity Model (CMM)

In CMM author discussed how to perform implementation framed for the software and how its result will be shown. How to remove errors. For improving quality of product different approaches can be used like testing, or CMM capability maturity model. CMM models for software shows code framing for it. Development approaches and collection of other different approaches were also used. XP can also be used with CMM to improve quality of software. CMM is a good solution as it clearly shows that either all requirements are met or not. It takes time for making this decision. Consideration of quality product cannot be done by itself. It also evaluates quality during development. Using XP interactive technique is good in getting feedback. Risks can be mitigated at very early stages and quality verification related to customers' needs can be checked. Indirectly, XP contains rules that supports software quality [7].

E. Development Standard

During a workshop, triple constraints were discussed. They said that there is a connection among cost or product, schedule of product and its quality. If any one factor is changed then it will affect the other two factors as well. Like if quality will increase then schedule and cost may also vary from previous planning. Several different standard development approaches exist that can be used to enhance the quality of product.

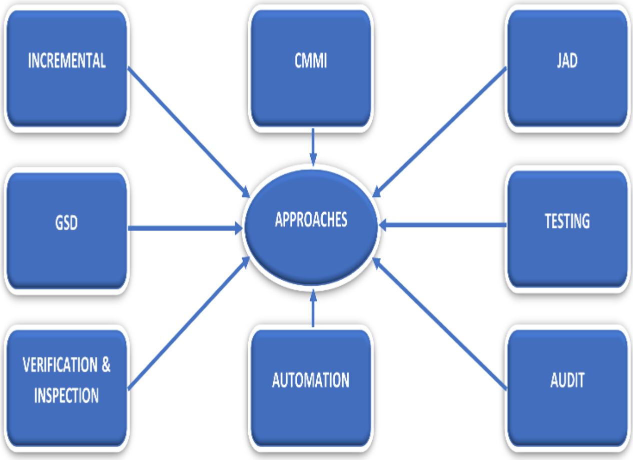

Figure 1: Techniques for Quality Assurance

F. Global Software Development (GSD)

Author described several issues that are faced by developers and project team members. People of different areas speaks different languages and sometime due to communication gaps, if some changes are made in the product, then it will become hard to manage those changes. Cost of product will increase by increasing the quality of software. Some security as well errors in documentation may also occurs. These types of hurdles can be removed with the help of GSD technology. You can use agile development or methodology when chances of error are very little or when your product is error free. GSD approach is very good for quality product, still a question can be arising related to quality evolution for product. GSD provides facility to communicate continuously over broadband. In GSD models' rules are not defined [8].

G. Audit Technique

Author says you will face several hurdles during development process that will obstruct your way in developing a quality product. Some risks are technical, and some are complicated, they are hard to handle and maintain. During process of managing risks the estimated cost of product will increase and time schedule may also slip. For overcoming these issues audit technique with some testing techniques can be used. In audit techniques we evaluate standards and some procedures. Several validation techniques and verification techniques can be used. For identification of risk some testing techniques can also be used. Status of product can be obtained by using audit technique and it gives a review quality assurance process and management process. Medical system will use this technique during development of their products. This technique in not best technique for detecting errors. It should be done throughout the development life cycle of product [9].

H. Quality Function Deployment (QFD), Joint Application Development (JAD)

Author describes multiple errors like errors that occurs in test planning and test cases. It mostly arises in graphics due to unclear requirement specification. Issues that arise in databases and warehouses will increase cost. For these kind of defects different approaches are used like JAD and QFD. In JAD we do analysis at the base from where the error is generated. In QFD verification of product and validation of product can be done [11]. Code inspection can also be done in QFD to mitigate defects. By the help of QFD and JAD issues are estimated, and maintenance and complexity standards are improved. For quality product, quality can not be produced by using only one approach, you must have to use different approaches to resolve different issues [10].

I. Four Step Approach Model of Inspection (FAMI)

In FAMI approach, issues that are related to product quality and ways to maintain the defects are discussed. FAMI contains full package for quality improvement. FAMI is based upon three Ps. This will totally remove all errors and defects, but it does not support by innovative products. Author discussed issues of triple constraints and apply different approaches as well different metrics. All these approaches were enforced by accuracy rate and proper process. There are several models for improvement of quality exists which can be used to enhance the quality of product.

J. Software Development Life Cycle (SDLC)

In SDLC, quality assurance operations in which systematic actions are performed in order to estimate the quality of process. In SDLC model every process that will be done during software development will be monitored and data integrity issues are also discussed.

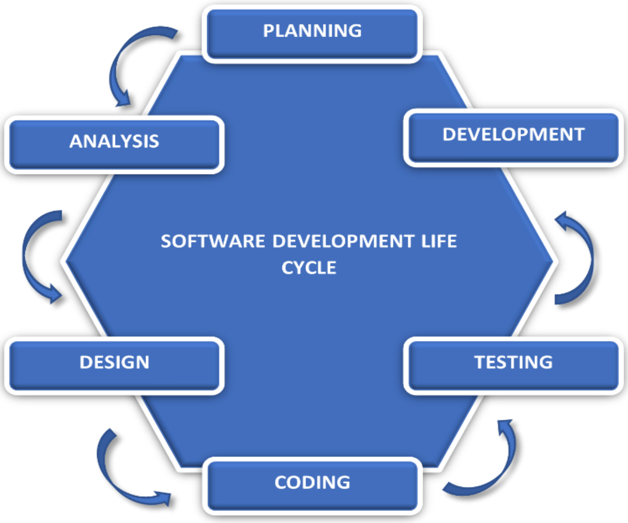

Figure 2: Software Development Life Cycle Plan

K. Quality Metrics

Author explained different quality parameters of products. He explained multiple measures that can affects firm structure. If planning is done before development, then there is a chance that schedule of product and cost may be affected later. Delay might occur but this will be difficult as it relies on variables. Decision making is also a strong factor in building quality product. If decision making is weak then there will be chance that the team will divert from planned development. Complex patterns of designs also obstruct development of software [18].

L. Multidimensional Technique

Computer system implementation uses some standards, modeling, performance parameters and some kind of analysis ways to evaluate system. These ways determine computer system quality. This will include multidimensional characteristics like technology and people. These approaches deal with the different aspects that are related to the groups and employees etc.

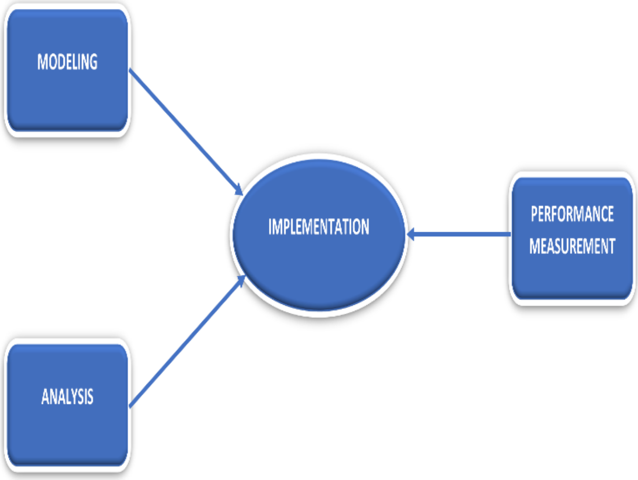

Figure 3: Implementation of Analysis, Standards and Approaches

M. Engineering Principles

Software engineering is discipline which uses principles of engineering to maintain the quality of software. Errors that occur in software are very expensive to fix. For these issues, there are several quality assurances approaches. These approaches will improve quality of software and improve quality attributes.

N. System & Software Quality Assurance (SSDA)

Usability means how easy a software is for use and learn. If usability of the software product is high then many users will interact with it effectively, this can be done when organization target users within numbers. So, for this purpose layman should be considered, in order to keep them before the usability should never be limited. Strict SQA methods were also followed. Testing has also been performed for the approval of this technique.

O. Total Quality Management (TQM)

TQM is a quality tool used for quantitative methods as well for human resources that are needed to supply to organization for betterment. TQM is a combination of some basic approaches, standard tools and improvement efforts that will focus on improvement of product continuously.

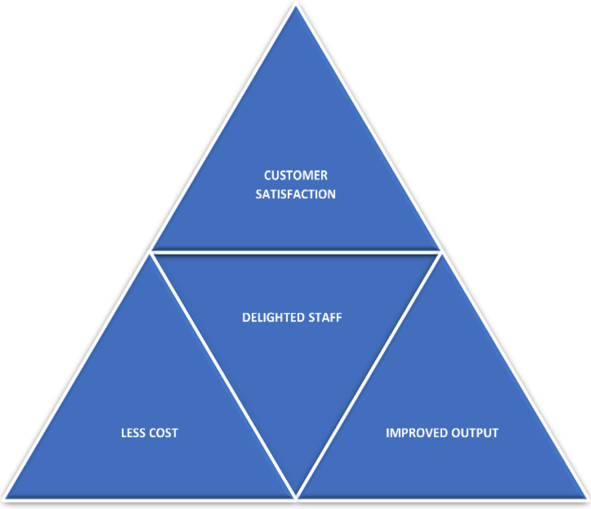

Figure 4: TQM Pyramid

P. Automated Analysis

Researchers for development of a product, firstly all requirements need to be clarified that are related to the product. As all requirements are cleared and finalized then a baseline requirement will be developed. Basically, requirements have two broad categories. Functional requirements that will tell what the system will do. It includes functionalities and features. While nonfunctional requirements are the constraints upon development team for building the system. It includes system design, attributes and quality of product.

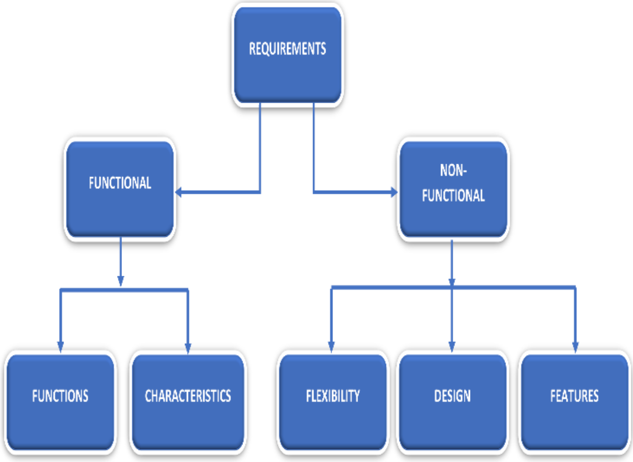

Figure 5: Requirement Types

Q. Requirement, Aspects & Software Quality (REASQ*)*

Author described object-oriented design and analysis of system. Its main concern is with the functional requirements or functionalities and simply avoids non function concerns. It results in rough code that will become hard to understand and maintain.

R. Object Oriented

Product code, product quality and design describe attributes of product. Some attributes are shown in figure below. Basic attributes explain characteristics of project. These attributes cover system aspects like coupling, inheritance, polymorphism, classes, encapsulation, and structures. This approach leaves very good impacts on quality of product.

4. COMPARISON

In the below table some quality techniques and their outcomes are discussed.

TABLE 2: COMPARISON TABLE OF TECHNIQUES

#	PROBLEM	TECHNIQUE	RESULT
1	Budget, Time, Standards, and internal politics cause hurdles.	CMMI, Specialized team, and Domain knowledge.	CMMI took many resources and time consuming but remove errors. The team might have to face pressure because of no compromise about quality.
2	Incomplete and development cost cannot reconsider.	Incremental approach and systematic methodology.	If we freeze the requirements, then no more features can be added.
3	Finding and reducing Errors.	Divide and conquer	This method was used to find the quality of software, not a process.

4	Unclear communication n causes major hurdle in quality software.	Automation Techniques	These are not enough because these are not testing methods. Execution could be correct but no surety about the fulfillment of requirements.
5	Duplication of Features.	Verification and inspection methods and SQA metrics.	More effort would be needed to remove duplication.
6	The change occurred but difficult to manage.	GSD and Agile methodologies.	Indeed, GSD is good but there are no specific defined rules.
7	The occurrence of errors in graphics, testing plans and cases, defects in the database and data warehouse.	JAD, QFD, verification and validation, code inspection.	Different combinations are required to solve these major issues.
8	Risk Management	Offshore Distribution Model	Support offshore decision-making.
9	Cost, schedule, and quality of product.	Standard development Methodologies.	High level Quality.
10	Errors in documentation.	Quality assurance methods	Enhanced the quality.
11	Issues related to data Integrity.	SDLC development process	Secure data.
12	Issues related to quality & defect maintenance.	FAMI	Leads to zero Defects.
13	Issues related to schedule, budget, and quality of software.	Metrics	To enhance the quality of the product.
14	Measurements that affect the organization structure.	Pre-planning about development	Delay can occur but this is difficult.
15	Defects, poor decision- making.	The rework to solve defects	Too much time consuming.
16	Complexity of design.	Analyze procedures, MOOD.	Effect quality Characteristics.
17	Predict presence of Errors.	Coupling and inheritance	Strong impact on the software.
18	Reliability Issues.	Genetic Programming	Validation of software quality.
19	Design issues.	Different methodologies	Artifacts are given.
20	Gap b/w software Practices.	Statistics	Not supported for all scenarios.
21	Issues related to group, technology.	Performance measurements, modeling, and analysis.	Determine the quality of system.
22	Maintenance.	SMMM	Improve quality.
23	Usability limitations.	Quality assurance methods, Testing	Increase usability.
24	Software direction is not defined.	Quality methods were used	Define direction.
25	How to improve product quality.	TQM techniques, tools	Increase quality.
26	How software quality is taken in higher education.	Multiple techniques are used	Fulfill Requirements.
27	How software risks are reduced.	Testing techniques	Reduce problems, but no single technique is perfect.
28	How to handle both business and technical aspects.	MDE	Improve quality and reduce cost.
29	Inadequate communication, technical knowledge issues.	Knowledge must be improved.	It will welcome everyone to communicate with each other, eventually, CM will improve.

30	Unclear requirements	Semantic web technologies, Onto SRS	Secure SRS Document.
31	Improvements in SRS document.	Automated Analysis	Supports vector Machine.
32	It ignores nonfunctional requirements which result in tangled code and it became difficult.	RWASQ	It keeps a balance between them.
33	Software validation.	Metrics	Measurements are validated.
34	How to handle FURPS for code.	Object Oriented Metrics	Improves quality and leave good influence.

5. Result and Decision

Various approaches have been explored to improve software quality, each offering distinct benefits. Systematic and incremental approaches are effective but face limitations in accommodating changes due to frozen requirements. The FAMI approach, however, stands out by focusing on defect prevention and aiming for zero defects, showing promising results in software quality enhancement. Our findings indicate that combining different approaches leads to the best outcomes, as no single method can fully address all challenges.

To optimize software quality assurance processes, organizations should adopt a more flexible approach, such as integrating agile practices, which allow for continuous feedback and adaptation to changing requirements. Additionally, automating testing can improve efficiency and reduce human error, while risk-based testing ensures that critical areas are prioritized. Implementing Continuous Integration and Continuous Deployment (CI/CD) practices can also help identify defects earlier in the development cycle, improving overall quality. Lastly, fostering collaboration between development and QA teams, alongside proactive defect prevention strategies like static analysis and code reviews, can significantly enhance the quality of the final product.

These strategies, when aligned with industry standards, can help organizations streamline their software quality assurance processes, reduce defects, and improve efficiency.

6. Conclusion and Future Directions

We conclude that various issues related to software quality exist, and several solutions have been proposed by different approaches and technologies. A software product of high quality has a higher chance of securing a place within IT organizations, contributing to various roles within the firm. Client satisfaction is heavily dependent on whether the software meets their needs and requirements. Furthermore, timely delivery of the product in accordance with user needs and specifications is critical. Different approaches can be adopted to improve software product quality, and each technique has its distinct impact on enhancing software quality.

Looking ahead, future research could explore the integration of existing software quality improvement methodologies with modern AI-driven technologies. One promising direction is the application of AI for enhanced testing and fault prediction, which could facilitate proactive identification and resolution of quality issues during the software development lifecycle. Additionally, the development of smart debugging tools powered by AI could optimize the debugging process, providing more efficient solutions compared to traditional approaches. Future studies may also investigate the use of machine learning algorithms for continuous improvement in software quality assurance, enabling systems to learn from past defects and dynamically adjust testing processes. By combining these approaches, researchers can contribute to creating a more efficient and effective software quality assurance framework, ultimately improving time and cost efficiency in software development.

References

[1] Ali Javed, Muazzam Maqsood, Khurram Ashfaq Qazi,Khurram Ali Shah,” how to improve software quality assurance in developing countries” Advanced Computing” an international Journal (ACIJ), vol 3, No2.

[2] Maneela Tuteja, Gaurav Dubey, Importance of testing and quality assurance in software development life cycle (SDLC) Models, International Journal of Software computing and engineering, ISSN: 2231_2307, Volume_2 Issue-3 July 2012.

[3] Nelly Meneva,” Soft quality assurance and maintenance for outsourced software development” Institute of Mathematics and Informatics, BAS, 113 Sofia, and American university in Bulgaria, 2700 Blagoevgrad.

[4] David L.Parnas, mark Lawford, “The role of inspection in soft quality Assurance” IEEE transaction on software engineering vol.29 No8 August 2003.

[5] Steve Ross-Tolbot, “A revolution in software quality assurance and software delivery “Cognizant passion for building a stronger business.

[6] Mr.S.manivannan, Dr.S.balasubrmanan,” Software Process and Product quality assurance in software organization” International Journal of Information Technology and Management Information System, vol 1, Issue Map 2010, pp-1-11.

[7] CH.V. Phani Krishna, G.Rama Krishna and K.Rajasekhana Rao, “A panoramic Approach on software quality assurance proposed by CMM and XP”IJCSNS International Journal of Computer Science and Network Security, Vol 13 No 2 Feb 2013.

[8] Khalid Khan, Adnan Khan, Muhammad Aamir and M.N.A Khan, “Quality assurance assessment in Software Global development” world Applied Science Journal 2(11): 1449-1454-2013- ISSN -4952.

[9] Dennis devolder, Shahin Ghananshahi, Jeff Zadeh, “Software testing quality assurance”, Western Illinois University Macomb, Illinois 61455, California state university Fullerton, CA 92834, virgina state university perenbsung, VA 23806.

[10] Capers Jones, “Software quality in 2002: A survey state of the art” software productivity research and Artenis Company.

[11] Rafael prikladnicki, Marcelo Hideki Yamaguti, Risk management in Global Software Development: A Position paper” School of computer science, Pontifica Universidade Catolica do the Rio Grande do Sul, Brazil.

[12] Barry Boehm, Sunita Chulani, Jane Verner, Bernard Wong “Fifth Workshop on software Quality” IBM research.

[13] Elyezjy, Neal T. "Software Quality Assurance-SQA-Practices in Palestinian Government." Information Technology 4 (2012): 19.

[14] Glick, Bud. "An SQA quality tracking methodology." Software Maintenance, 1990, Proceedings. Conference on. IEEE, 1990.

[15] Nair, T. R. "Four-Step Approach Model of Inspection (FAMI) for Effective Defect Management in Software Development."ArXiv preprint arXiv: 1209.6466 (2012).

[16] Ward Jr, William A., and Buvaneswari Venkataraman. "Some observations on software quality." Proceedings of the 37th annual southeast regional conference (CD-ROM). ACM, 1999.

[17] Nagappan, Nachiappan, Brendan Murphy, and Victor Basili. "The influence of organizational structure on software

quality: an empirical case study. "Proceedings of the 30th international conference on Software engineering. ACM, 2008.

[18] Melo, Walcelio. "Evaluating the impact of object-oriented design on software quality. "Software Metrics Symposium, 1996, Proceedings of the 3rd International. IEEE, 1996.

[19] Briand, Lionel C., et al. "Exploring the relationships between design measures and software quality in object-oriented systems." Journal of systems and software 51.3 (2000): 245-273.

[20] Evett, Matthew, et al. "GP-based software quality prediction." Proceedings of the Third Annual Conference Genetic Programming, volume. 1998.

[21] Peffers, Ken, et al. "A design science research methodology for information systems research." Journal of management information systems 24.3 (2007): 45-77.

[22] Zhao, Luyin, and Sebastian Elbaum. "Quality assurance under the open source development model." Journal of Systems and Software 66.1 (2003): 65-75.

[23] Palvia, Shailendra C., Ravi S. Sharma, and David W. Conrath. "A socio-technical framework for quality assessment of computer information systems." Industrial Management & Data Systems 101.5 (2001): 237-251.

[24] Bertolino, Antonia. "Software testing research: Achievements, challenges, dreams." 2007 Future of Software Engineering. IEEE Computer Society, 2007.

[25] April, Alain, et al. "Software Maintenance Maturity Model (SMmm): the software maintenance process model." Journal of Software Maintenance and Evolution: Research and Practice 17.3 (2005): 197-
223.

[26] Hedberg, Henrik, et al. "Assuring quality and usability in open source software development." Emerging Trends in FLOSS Research and Development, 2007. FLOSS'07. First International Workshop on. IEEE, 2007.

[27] Kitchenham, Barbara A. "Software quality assurance." Microprocessors and Microsystems 13.6 (1989): 373-381.

[28] Ahire, Sanjay L., Robert Landeros, and Damodar Y. Golhar. "Total quality management: a literature review and an agenda for future research." Production and Operations Management 4.3 (1995): 277-
306.

[29] Chung, Lawrence, Brian A. Nixon, and Eric Yu. "Using quality requirements to systematically develop quality software. "Proceedings of the 4th International Conference on Software Quality. 1994.

[30] Chua, Clare. "Perception of quality in higher education." Proceedings of the Australian universities quality forum. Melbourne: AUQA Occasional Publication, 2004

[31] Chua, Clare. "Perception of quality in higher education." Proceedings of the Australian universities quality forum. Melbourne: AUQA Occasional Publication, 2004.

[32] Pham, Hoang. Software reliability. John Wiley & Sons, Inc., 2015, Herbsleb, James D., and Deependra Moitra. "Global software development."Software, IEEE 18.2 (2001): 16-20.

[33] Castañeda, Verónica, Luciana C. Ballejos, and Maria Laura Caliusco. "Improving the Quality of Software Requirements Specifications with Semantic Web Technologies." WER. 2012.

[34] Rashwan, Abdulmonem, Olga Ormandjieva, and Russell Witte. "Ontology-based classification of non-functional requirements in software specifications: a new corpus and svm-based classifier." Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual. IEEE, 2013.

[35] Castillo, Isi, et al. "Requirements, Aspects and Software Quality: the REASQ model." Journal of Object Technology 9.4 (2010): 69-91.

[36] Lazic, Ljubomir, and Nikos Mastorakis. "Cost effective software test metrics." WSEAS Transactions on Computers 7.6 (2008): 599-619.

[37] Kan, Stephen H. Metrics, and models in software quality engineering. Addison-Wesley Longman Publishing Co., Inc., 2002.

[38] Rosenberg, Linda H., and Lawrence E. Hyatt. "Software quality metrics for object-oriented environments." Crosstalk journal 10.4 (1997)

[39] Amin, M. R., and Ahmed, M. S., "A Comprehensive Review of Software Quality Assurance Techniques and Tools," International Journal of Software Engineering and Applications, vol. 13, no. 5, pp. 55-72, 2020.

[40] Sharma, R., and Singh, P., "Recent Trends in Software Quality Assurance: A Review," Journal of Software Engineering and Applications, vol. 15, no. 6, pp. 105-118, 2022.

[41] Kumar, R., and Verma, A., "Techniques for Improving Software Quality: A Systematic Review," International Journal of Computer Science and Information Security, vol. 17, no. 3, pp. 189-200, 2019.

[42] Patel, D., and Desai, R., "A Review on Software Quality Improvement: Approaches and Challenges," Journal of Software Engineering Research and Development, vol. 8, no. 2, pp. 123-135, 2021.

[43] Ravi, S., and Pandey, A., "Modern Approaches in Software Quality Assurance: Tools, Techniques, and Best Practices," International Journal of Computer Applications, vol. 182, no. 9, pp. 10-16, 2021.