Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.2 : 2024
ISSN : **1906-9685**

A SURVEY ON ENHANCING SOFTWARE QUALITY ASSURANCE IN SOFTWARE DEVELOPMENT

P.Rukmanidevi Research Scholar, Department of Computer Science, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore, TamilNadu.Email id: devimaha.rm@gmail.com
Dr. M. Chandran Associate Professor, Department of Computer Science, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore, TamilNadu. Email id: onchandran@gmail.com

Abstract : To make a good reputation in the software industry,quality is an essential thing for any organization. It totally depends on how satisfied the customers are with the product. The target can be achieved only through proper standards and procedures. The big and renowned countries are making progress in this field day by day. The different organizations are trying their best to develop quality software. For this purpose, they made standards, but still, there can be different issues. There are multiple reasons for the lower quality of software. In this paper, different problems were addressed, corresponding to these issues; different techniques were elaborated, and corresponding solutions were also defined, which led towards the quality of software.

IndexTerms - Software Process Improvement,SQA Metric,GP,SMMM,quality tools.

INTRODUCTION

In the development of any kind of project, quality plays the most important role. To achieve a good position and reputation in IT, organisations really need to provide products that retain quality. A product's quality can be corrected before its shipment. Once the product is in the hands of clients, its correction takes a lot of money and time. Moreover,they need to provide extra packages like warranties, replacement products, etc. Errors can be avoided if the proper quality plan that is made before the practical is completely and properly all steps.If a business moves towards the launch of automated products, then product quality must be maintained for the business organization.When working on small projects, CMMI and ISO models can be used for large projects but not for small ones because they take a high cost. The quality problems are growing in different countries. They show a correlation between team and corporation among them.[1]

If any error is pointed out, then it is very difficult to find defects in design because all digital systems that are used are not continuous, and variables are not enough to test those errors. Testing is the basic method to detect errors in the software. Once the error is found, it is now possible to remove it to maintain the quality. If an error or issue is found, then it will indicate how much budget they will require to maintain the quality. As soon as the error is found, the budget will not touch the sky. If effective and careful testing is done, then errors can be minimised and the quality of the software increased [2]. The different types of testing techniques and methods are done,along with the different models associated with them, which guide what type of model can be used for what kind of product. These models also have a few advantages and disadvantages. Whatever kind of testing is used,it's really need it in every phase of the implementation of the quality plan. The errors could be of any type. The errors could be present in the documentation, code, design, or even the plan. One testing technique might not be suitable for different techniques. It means different projects require different types of testing. The outsourcing of projects that are given to a third party requires high quality of the product and no compromise over qualifications, skills, or experience. The company should resolve these problems [3]. So, planned strategies must be adopted to produce a quality product for outsourcing. All activities require complete knowledge and skills. Software quality not only depends on a quality plan, but it also depends on quality management, verification, and the quality of a product. Different people think and expect that this software will definitely help them do the job properly. The product quality is characterised by reusability, portability, reliability, manageability, flexibility, and usability [4]. The implementation of the peer objectives and differentevolution techniques are followed. All members must be trained, responsibilities must be assigned to each member, and products must be given to a quality assurance member. The checklist must be made so that it is possible to check for errors.During the review, the moderator needs to write down all defects, but there is no type of review that requires the product to be redesigned.

The participants need to ensure that they are fit for this project and the work associated with it [6]. The lack of communication among group members is a big issue. One reason could be the different languages spoken by team members. If the physical distance is large, then it will also count as a lack of communication; if there is less communication, then participants will not understand issues. The time zone also matters because how much mature process is being used also counts [7]. The management is not good if there is weak planning and coordination among team members. If there are changes needed, either in schedule or planning, it will result in delays and errors. [8]. The software quality, not the single method, is good, but it will require combinations of many methods to produce a quality product. Only testing is not a solution. There is a special need to solve bugs in the database [9]. The product quality depends on strong and systematic standards and procedures. [10]. Distributed software development is a special methodology that has a strong impact on software development [11]. If documentation and risk handling are not proper, then it will lead to the failure of the project. The software quality assurance in the aspect of cost then decreases cost but benefits increase [13]. Not only does quality review require software quality assurance, but each phase also requires a quality process. The output will increase if a quality process is adopted. [14]. Quality management is basically the most important thing. It is particularly important for large and complex projects. The quality documentation shows the progress of a project, but for smaller projects, documentation is not needed so much and the focus is on establishing culture. We cannot wait for specifications to improve before giving attention to them. We must put quality procedures into place to improve the quality of a product. To ensure the quality of software data, accuracy is one of the important factors.Don't collect unnecessary data. What the answer should beThose questions must be decided before the distribution of the questionnaire. The people must be told what data should be
ng collected. Don't rely on thememory write down all answers. If it is possible, then consult with professionals. Software organisations use many error detection and prevention techniques to describe those problems [15]. The organisations need to clearly define quality standards before judging them [16]. Organisational structure has a strong impact on product quality. After analysing and reading thoroughly, the result describes that a quality product can be produced if there is a quality plan, which will be followed later. The different error removal techniques must be applied; as soon as errors are found, they will be good to that level. For all progress, communication must be there among team members. The following techniques can be used to solve quality issues and provide a good software product: This paper is organised in sections: Section I gives the introduction to the topic; Section II defines the literature review; Section III compares all techniques used to solve multiple problems; Section IV describes the result, which comes after comparison; and last but not least, Section V describes the conclusion.

LITERATURE REVIEW

In the literature review,different techniques are discussed.

CMMI

In this paper, many problems were discussed related to software quality assurance. To improve the quality of software, they must be provided with software where the team can work and perform their responsibilities. There must be some ways to improve the quality of the software. In this research paper, the authors discuss the main issues of budget, time, less use of standards to gain quality, internal political issues, requirements engineers, and unrealistic deadlines. Keeping in mind these issues, they proposed a technique called the CMMI Model, a certified and specialised SQA team. Requirement engineers must have domain knowledge and must avoid internal political knowledge. The specialist knows exactly how to deal with a problem. Requirements were cleared when members gained complete knowledge. CMMI took many resources and time, but it removed errors. Secondly, a specialised team first finds an error, then minimizes it. Due to no compromise on quality, the team might have to face pressure, but it has proved beneficial [1]. **Models**

In their research paper, different quality factors are discussed and different problems are highlighted, e.g. The occurrence of errors in the soft requirement document, design document, test plan, etc. Further, they explained that requirements are not specified; they might consider it a big issue, so they kept away from compliance. No standard was followed. Moreover,management issues were highlighted. Different SDLC models are used: waterfall model, spiral model, prototype model, rapid model, and V model; moreover, different testing techniques, e.g., black box testing, white box testing, and grey box testing, are applied in each model. He explained that to improve the quality of software, effective testing can be done to minimize errors. Different techniques were applied, which may reduce errors, but in some ways, a few techniques need to be measured because they may not be clearly attributed. So far, this V type model has been used. Testing can be applied in all phases, but not at one particular stage [2]. Systematic methodology In this research paper, the authors explain related outsourcing projects and identify different problems, e.g.

Development costs can’t be reconsidered; what type of

Hardware and software would be used in the project; the requirements are not clear and incomplete. To solve these issues, different techniques were made that may prove to be effective, like at the start of the project. Incremental approaches were used. They made versions, and the next version is based on the previous version's output. They did comparative analysis, and different supporting tools were used; moreover, the systematic methodology was adopted. He gave solutions regarding these issues. He recommended soft metrics, testing the product using static and dynamic methods, and freezing further.

requirements to assure quality. Due to these techniques, large but sophisticated projects were being developed. If we freeze the requirements, then no more features can be added, and later,it could prove to be low-quality software [3]. Divide and ConquerParnas explained in his paper how to find and reduce errors froma problem, and make sure that it’s a quality product. The main methodology or technique he used was dividing and conquering. He explained in his research that if you are purchasing software that is free, then its quality is much better than others. He gave a solution to the software inspection tool. Quality can be improved on the basis of feedback. Indeed, the divide and conquer method takes time to find errors, but it will still find errors to solve an issue. This method was used to find the quality of software, not a process. [4]

Automation Techniques In this paper, they discussed that problem errors occurred due to not having clear communication, which gave a different interpretation of the same data. This causes an increase in the ratio of errors. To assure the quality of the software, he proposed automation techniques that are based on statistical and committed algebra. These techniques will definitely increase the quality of the result and speed up delivery; they also help reduce unclear communication. Further, SDLC breaks down into pieces and examines them at one time. These mathematical proofs are not sufficient because they are not testing methods. Execution of programme may be correct, but we can’t say that all requirements are met and fine [5].SQA metric problems occur when we are indented to make any quality software, so these problems proved hurdles, e.g., feedback is not given after using the software that they find. The requirements are not clear. The developer doesn’t know what customers think and what features they implemented. Duplication may occur in projects. To cover these issues, the authors highlighted the formal verification inspection method, checklists must be available to support assurance, and the SQA metric was also applied to detect errors. These methods in the process and the product quality assurance processes, which are in PL, are implemented. While if you go for the verification process, those requirements that were listed to be implemented are satisfied. Duplication occurred, so they had to remove it by doubling the effort, but still, perspectives related to this case were separately maintained [6]. CMM How implementation is framed for soft and how to result is shown. How were errors removed? To improve the quality of software, different techniques are used, e.g., testing and capability maturity model analysis, to show the code framing for software. Moreover, development methods and integration of the rest of the techniques were used. XP is also used along with CMM to ensure the quality. The capability maturity model is a better solution because it takes time to determine whether all requirements are met and the quality process is not considered itself. Instead of it, it will evaluate the quality of a development. The interactive approach proved to be effective in obtaining feedback via XP. Quality verification will check what customers require. Moreover, risk can be mitigated at early stages. Extremely complex programming needs some evolution to fulfill CMM-specified requirements in combination with CMM.Indirectly, software quality support is present in XPrules [7]. GSD Authors discussed the different issues a developer and the entire project team face, including: Communication gap among team members If any change occurs, then it proves to be problematic to manage; to increase quality, cost will increase; errors in documentation and security issues will also o occurred.The GSD methodology was actually chosen to remove these problems, which proved to be a hurdle. The agile method is used for customer collaboration because it proves to be beneficial for error-free products. GSD creates conflict on the platform that is related to code. Instead, GSD is good for quality software, but it still raises the question of the evolution of the quality of the product. GSD allows continuing communication over broadband. Still, there is no one model-defined rule [8]. Audit technique While during the development, different risks occurred, which proved to be a hurdle in quality software, like technical and programmatic risks. Some risks occurred, but those were very complicated to edit and maintain. Risk management is leading to overruns in cost and schedule. Audit techniques and testing techniques are used. The audit is a technique that evaluates procedures and standards. Different verification and validation techniques are used. Testing is also done to evaluate risk. Audit gives the status of the software product and also gives a view of management and assurance processes. These techniques are specially used for the medical system. These techniques are not enough to find the error. It must carry through the life cycle [9]. QFD, JADI In this research paper, multiple errors were discussed, e.g., the occurrence of errors in the plan made for testing and cases that frequently occur in graphics. The requirements are specified but unclear. Defects in the database and data warehouse increase costs. To resolve these problems, different types of techniques are applied, e.g., JAD (Joint Application Design). Do analysis from the root of the cause, QFD (quality function deployment), verification and validation [11], total quality management, and code inspection to remove the defects. Using these techniques, defects in estimates, complexity, and maintenance measures are improved. To ensure quality, one method is not enough, but they really choose different combinations of methods to solve the major issues. [10]. Development standard In the workshop, they discussed the relationship between cost,schedule, and quality of the product. If we increase the quality

then it’s obvious that cost and schedule will be distracted from.the planning. Different standard development methodologies Standard development methodologies were employed, which may increase the software quality to a high level. Figure 1 explains the total techniques that are used for QA.

SOFTWARE QUALITY ASSURANCE
SQA Techniques
Reviewing
Design Inspection
Auditing
Static Testing
SQA Techniques
Walk through
Code Inspection
Standardization

Engineering principles Software engineering is a field that applies engineering principles so that software quality can be maintained. Most software faces problems with errors, which results in high costs. For this, they proposed different quality assurance methods. It will eventually affect the quality and enhance the quality attributes. SDLC software quality assurance functions, in which systematic activities are done to evaluate the quality of the process. They tried to monitor all processes that are performed during the development of software. They also discussed issues related to data integrity. Database management was considered a result of security. The SDLC development process was also followed. Figure 2 explains the engineering principles for software quality assurance.

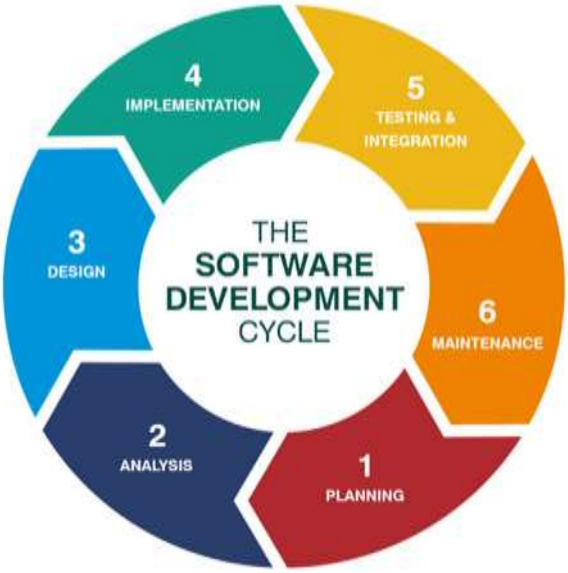


Fig. 2: SDLC Plan

FAMI

In this paper, they address issues related to the quality of the product and how to maintain the defect. FAMI was introduced, which was full of packages to enhance the quality. Basically, it depends on the 3 Ps. This basically leads to zero defects, but this is not good for innovative products. In this paper, they discussed issues related to schedule, budget, and quality of software; they applied different technologies and methods as well as metrics. But all these were applied through the proper process, and the accuracy rate was continually checked. A number of quality models were available for organisations; different models were used to enhance the quality of the product. Evaluation Technique In this approach, better quality improvements and better quality measurements are used. Table 1 shows all measurements for the model.

Table

Model	Measure				
Code closure	total	freq	Repeated freq		
Complexity	cyclomatic	Fan in	Fan out	Weighted method	Sub class
Dependency	Incoming direct	closure	Outgoing direct	closure	Layer info
Code coverage	block	Arc			
Release	visuval				

Quality metrics In this paper, they discussed the quality metrics of the product. They highlighted different measures that affect the organisation's structure. If we do pre-planning about development, then cost and schedule may affect it later. Delay can occur, but this is difficult because it depends on variables. Strong decision-making They discussed the issues for the development of software and how some conditions require rework to solve defects. But this leads us to understand how software will affect quality. If poor decision-making is done, then it will deviate from planned implementation. Complicated design patterns also affect software development [18]. Coupling and InheritanceThe main purpose of this research is to find the errors and predict their presence, so to find the errors, it may be possible to find them again. Different measurements are used for object-oriented design. But these measurements are not enough to find the fault and where it is located. Complete analysis procedures are used to validate it. Figure 3 shows the coupling of hotel management classes to application classes.

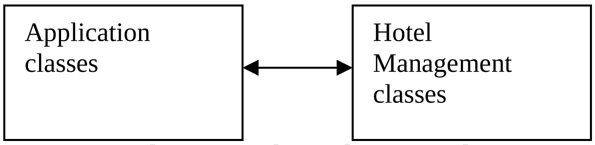


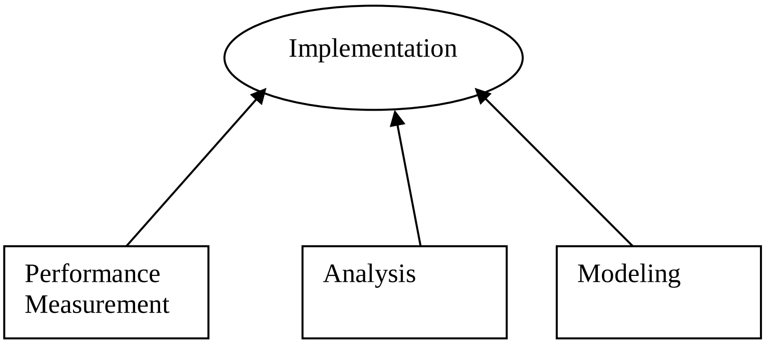
Fig 3.Coupling hotel management classes with application classes

Techniques used for this problem are coupling and inheritance measures, which will prove to be significant in predicting the location of the fault. The result shows that fault leaves a strong impact on the quality of software [19].

GP Different software quality methods are used to ensure the quality of the software. Reliability is one of the factors that is considered important for quality. To resolve reliability issues, GP (genetic programming) techniques are used. GP will generate metric input, which results in output that

predicts the existence of a fault. It will not count how many faults are found but rather predict the existence of errors. It can be used as a tool to validate software development quality. Design methodology Here, they discussed the IS design issues. It basically discusses the issues related to the methodology for design science. Different methodologies were discussed to resolve it and meet conducting issues in design science. They discussed design problems. For this artefact, it was given [21]. Development practices They discussed software development practices. They talked about the issues related to software development and traditional development; they tried to find the difference between them and further discuss whether they could translate it into practical benefits or not. It might not be supported by all scenarios. Statistics were given to resolve that in the proper context. Multidimensional Techniques Implementation of the computer system allows following the specific standard; performance measurements, modelling,and some analysis techniques are followed to evaluate he system. Basically, these techniques determine the quality of a computer. It covers the multidimensional characteristics of technology and people. These techniques deal with different properties related to the group, employees, etc. Figure 4 describes the implementation of standards, techniques, and analysis.

Fig. 4. Implementation of standards, techniques, and analysis



SMMM
Here, they discussed software maintenance to maintain the quality of the software. For this, they proposed a maintenance maturity model because it improves the maintenance and improves the quality of the software. It totally depends on the practitioner as to how much he or she knows about it. SSQA Methods If the usability of software is higher, more users can use it efficiently because they target users in large numbers. For this, alayman must be considered, so by keeping them before,usability must not be very limited. Strict software quality assurance methods were followed. Testing was also done to approve this technique. Quality Tools Software quality is a question mark: why is it necessary and what are its benefits? If a software product has no defined direction, then it is difficult to assure its quality. Different methods and tools were used. Is everything right to develop software? Is there a defined direction? TQM is the application of quantitative methods and human resources to improve materials and services supplied to an organisation and all processes within the organization, to which the needs of the customer are met. TQM combines basic techniques, improvement efforts, and tools under a disciplined approach that is focused on continuous improvement. Figure 5 represents the TQM pyramid.

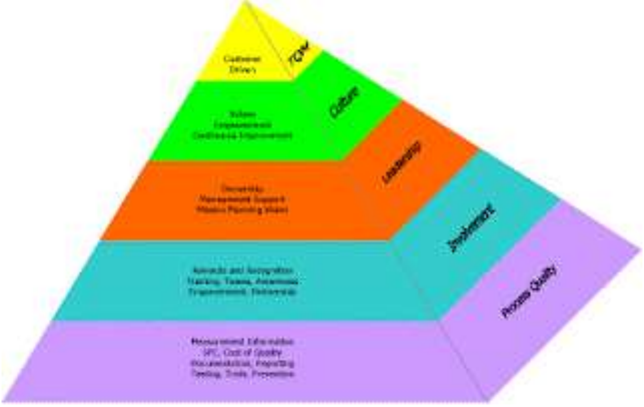

Customer Driven
TQM
Values Empowerment Continuous Improvement
Culture
Ownership Management Support Mission Planning Vision
Leadership
Involvement
Rewards and Recognition Training, Teams, Awareness Empowerment, Partnership
Process Quality
Measurement Information SPC, Cost of Quality Documentation, Reporting Testing, Tools, Prevention

Fig 5. TQM pyramid

Technical tool Data integrity is an issue such as security,
moreover, performance and accuracy matter a lot. There is a proper sequential way to meet the quality requirements. Technical tools were used for this purpose. The focus was on nonfunctional requirements so that software quality could be enhanced. Standard How is software quality perceived in higher education? The answer is that they perceived it as the whole standard. To meet and fulfil the requirements, the focus is on all aspects of the institution.Several techniques are used [30].Testing Technique Software testing techniques are evaluated. Every testing technique contributes to reducing risks. In fact, a single technique is enough for complete risk reduction from software, while combinations of software testing techniques are used to reduce the risk associated with them. They took significant steps on a large scale; they further evaluated systematic testing techniques to make them reliable. MDE Cloud computing is the most demanding technology nowadays. Basically, it improves the quality and, at the same time, reduces the cost of system components. This will basically cover both perspective business and technical. The MDE technique is used to cover it. Basically, this technique will automate the engineering and forward engineering phases. SDLC In this paper, they discussed economic issues if costs are changed. Basically, change leaves a strong effect not just on marketing but also on a product that is intended to design, construct, and ship software to customers. Inadequate communication becomes a hurdle to software progress. Due to this, project management cannot deliver their message properly, so management becomes poor. If

Employers' have poor knowledge about their field then of course then will choose the wrong methodology, which will lead to wrong software development. Since the world is progressing day by day, technology is also on the way of progress. If technology is not compatible, change control could lead to disaster. Semantic web technologies In this paper, they discussed the software requirements of the project. When customers note down their requirements, if their requirements are not clear, it may create a hurdle to the success of the software. If changes are made, then it becomes very difficult to handle. because it requires a high cost. To handle this flaw, semantic web technologies are used to define better SRS. The main approach used here, Onto SRS, secures the flexibility,unambiguity, and traceability of the SRS document. Automated Analysis In this paper, researchers define that whenever there is an need to develop a project, there is a need to clarify all requirements related to software. Then all requirements are finalised, and the baseline is developed. All requirements are finalized. Requirements are divided into two categories. In figure 6 requirements types are discussed

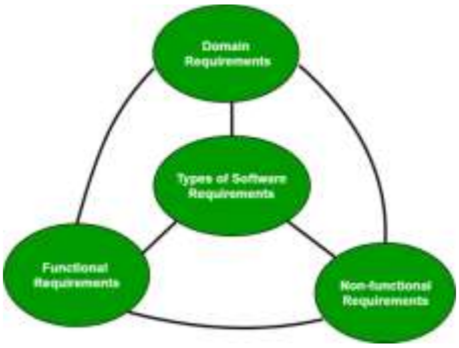


figure 6 requirements types are discussed

Functional requirements Functional requirements include the basic functionality of the system and the basic features that are demanded by customers or clients. Non-Functional Requirements Nonfunctional requirements include quality, attributes, main functionality, the design of the system, constraints, etc. Nonfunctional requirements have a strong impact on cost and schedule. In order to make improvements to the SRS document, there is a need to do automated analysis. Basically, it supports vector machines and automatically categorises requirements into further categories. REASQI In this paper, they discussed the object-oriented analysis and design of the system. Basically, it focuses on system functionality, but it ignores nonfunctional requirements, which result in tangled code that becomes difficult to handle and maintain. Figure 7 shows REASQ integration.

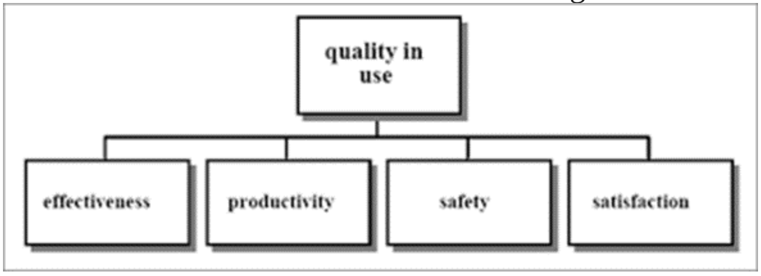


Fig 7.REASQ

It actually integrates the AOSD terms with requirements; the result of the approach is a conceptual model. Test metrics In this paper, as the population is growing, technology is also progressing, but with increasing technology, complexity ratios are also increasing. So to test a software product, high costs are also required. During testing, there is a lot of pressure on employees. There is pressure for high-quality products. There are so many chances of slippage of dates, limited offered resources, and scattered organization. We have to face team dispersion, a distributed team, and resources. Test metrics can be used for an effective and efficient software process. Basically, this paper evaluates the measurement.

model, which can be adopted according to culture and lifecycle. This survey paper basically presents a measurement of software quality from the basics of software quality metrics. These metrics basically validate the quality of the software project. All metrics must be understandable to users. Secondly, metrics related to the cost point of view, like spending 10% on metrics of total development, Before applying these metrics, they must be tested. All metrics must have a schedule. Moreover, they must be managed properly. Object Oriented If we talk about the product quality and about project code and design, then they have attributes for them. These attributes are shown in 8. The five attributes are explained, while the rest of the attributes are also self-described.

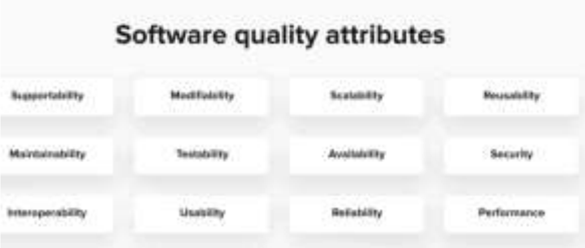


Fig. 8. Object-Oriented Attributes

Basically, these attributes describe the characteristics of the project. So for this purpose, they proposed some metrics, which are nine in number. Basically, they cover all aspects of the system, like all structures and classes they used, polymorphism, coupling, encapsulation, and inheritance. For each metric, there are a number of methods adopted, and the code and design will be workable if they are used. Eventually, it leaves a very strong effect on software quality.

III.COMPARISION

In Table 2, the problem, techniques, and all results are discussed.

REF NO	PROBLEM	TECHNIQUE	RESULT
[1]	Budget, Time, Standards, standards, and internal politics cause problems.	CMMI, specialised team, and domain knowledge	CMMI took many resources and was time-consuming, but it removed errors. The team might have to face pressure because of a lack of commitment to quality.
[2]	Requirements are incomplete, and development costs can’t 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 causes major hurdles in quality software	Automation techniques	These are not enough because they are not testing methods. Execution could be correct, but there is uncertainty about the fulfilment of requirements.
[5]	Duplication of features	Verification and inspection methods and SQA metrics	More effort would be needed to remove duplication

[6]	How to frame software implantation and reduce errors	XP,CMM	XP needs to add some evolution along CMM to fulfill requirements
[7]	The change occurred but difficult to manage	GSD and agile methodologies	Indeed,GSD is good but there are no specific defined rules
[8]	Technical and programmatic risks,cost and schedule over run	Audit technique,testing verification and validation	These techniques are not enough to find errors .we must carry it along all development life cycle
[9]	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 combination are required to solve these major issues
[10]	Risk management	Offshore distribution model	Support off-shore decision making
[11]	Cost, Schedule and quality of product	Standard development of mythology	High level quality
[12]	Errors in documentation	Quality assurance method	Enhanced the quality
[13]	Issues related to Data Integrity	SDLC development process	Secure data
[14]	Issues related to quality&& defect maintenance	FAMI	Leads to zero defects
[15]	Issues related to scheduled budgets and quality of software	Metrics	To enhance the quality of the product
[16]	Measurements that affect the organization structure	Pre-planning about development	Delay can occur but this is difficult
[17]	Defects poor decision making	The rework to solve defects	Too much time consuming
[18]	Complexity of design	Analyze procedure MOOD	Effect quality characteristics

I. RESULTS AND DISCUSSIONS

To improve the quality of software, different techniques were adopted and followed. Different issues arose. So for different issues, different techniques and solutions were given, which led to the quality of the software. For every problem, a solution is given, so each technique is found good enough to improve the quality. If we talk about the incremental approach and systematic methodology, then it is good, but it will freeze requirements. So, further changes in specifications will prove to be problematic. FAMI found a good technique because it leads to zero defects. The different techniques can be good to use if they are combined with other methods and techniques. No single technique brings exceptional results. Otherwise, all techniques discussed with thirty-nine references leave an acceptable influence on projects.

IV.CONCLUSION AND FUTURE WORK

In this paper, different problems related to software quality assurance are identified. The different solutions were given by using different techniques. The software can get a position in the IT industry in the case of quality software. It plays different roles in organizations because customer satisfaction is totally based on whether their requirements are met or not. There is so much need to deliver the product on time and exactly according to the specifications. The different techniques were used to ensure the quality of the software. Every technique has its own impact on the quality of software.

In the future, further work will be done to assure and develop quality projects. A reputed and renowned organisation can provide opportunities if the project is delivered on time and according to demands. In the future, new techniques, their uses, and their implementation are going to be discovered in each and every phase of the life cycle. This will help to increase the advantage of quality assurance, and it will be proven for further development.

REFERENCES

[1] Ali Javed, Muazzam Maqsood, Khurram Ashfaq Qazi, and Khurram Ali Shah,"How to Improve Software Quality Assurance in Developing Countries," Advanced Computing, an International Journal (ACIJ),vol. 3, No. 2,

[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 Informatics, BAS, 113 Sofia, and American University in Bulgaria, 2700 Blagoevgrad,

[4]David L. Pranas, Mark Lawford, "The Role of Inspection in Soft Quality Assurance" IEEE transaction on software engineeringvol. 29, no. 8, August 2003.

[5]Steve Ross-Tolbot,"A reevolution in software quality assurance and software delivery: "Cognisant passion for buildinga stronger business.

[6]Mr. S. Manivannan, Dr. S. Balasubrmanan," Software Process and Product Quality Assurance in Software Organisations," International Journal of Information Technology and Management Information Systems, vol. 1, issue map 2010, pp. 1–11

[7].CH.V. Phani Krishna, G. Rama Krishna, and K. Rajasekhana Rao, "A 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, Shahi Ghananshahi, Jeff Zadeh, Software testing quality assurance, Western Illinois University, Macomb, Illinois 61455, California State University Fullerton, CA 92834; Virginia State University, Perenbsung, VA 23806

[10] Capers Jones, “Software Quality in 2002: A Survey of the State of the Art." software productivity research and Artenis Company

[11] Rafael Prikladnicki, Marcelo Hideki Yamaguti, and Risk Management in Global Software development: A Position Paper," School of computer science, Pontifica Universidade Catolica do Rio Grande do Sul, Brazil

[12]Barry Boehm, Sunita Chulani, Jane Verner, and 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] Huang, F.; Strigini, L. HEDF: A Method for Early Forecasting Software Defects Based on Human Error Mechanisms. IEEE Access **2023**, 11, 3626–3652.

[17] Khan, S.U.; Khan, A.W.; Khan, F.; Khan, M.A.; Whangbo, T.K. Critical Success Factors of Component-Based Software Outsourcing
Development From Vendors’ Perspective: A Systematic Literature Review. IEEE Access **2022**, 10, 1650–1658.

[18] Slaughter, A.E.; Permann, C.J.; Miller, J.M.; Alger, B.K.; Novascone, S.R. Continuous Integration, In-Code Documentation, and Automation for Nuclear Quality Assurance Conformance. Nucl. Technol. **2021**, 207, 923–930.

.