ISSN 2278-3091

Volume 10, No.3, May - June 2021

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1131032021.pdf

https://doi.org/10.30534/ijatcse/2021/1141032021

A Methodology for Incorporating Quality Assurance Practices during Software Development Life Cycle

Eraj Laaraib[1], Zulfikar Ahmed Maher[1], Zulfiqar Ali Solangi[2],Muhammad Yaqoob Koondhar[1], Mukhtiar Memon[1], MansoorHyder Depar[1], Asadullah Shah[3]

[1]Information Technology Centre, Sindh Agriculture University Tandojam, Pakistan, Email erajlaaraib11@gmail.com, zamaher@sau.edu.pk,yaqoobkoondhar@gmail.com,mukhtiar.memon@sau.edu.pk, mansoor.hyder@sau.edu.pk

[2]Jubail Technical Institute-Colleges and Institutes Sector Royal Commission Jubail, Saudi ArabiaEmail: zulfs@hotmail.com

[3]Faculty of Information and Communication Technology, International Islamic University Malaysia, Email: asadullahshah@iium.edu.my

ABSTRACT

The aim of quality assurance in software development is to ensure that a software product meets its practical business specifications and quality requirements. A comprehensive overview of different software quality requirement and quality assurance practices have been performed to find out the most appropriate software quality assurance practices at different stages of the software development life cycle (SDLC). A methodology along with its implementation during SDLC has been proposed in this research paper. The proposed methodology will help software development teams to achieve quality assurance goals during software development.

Key words: Software Quality Assurance, Software Development Life Cycle, SQA Factors

1. INTRODUCTION

Software quality relates to a software which is fairly bug free or defect free, delivered in specified/proper time and within desired/expected budget, which at the end meets with the stated business requirements and is also maintainable. Functional quality and structural/non-functional quality are the two aspects of software quality in the context of software engineering. Software Functional Quality ensures the quality of a design based on functional specifications or requirements of services or products. Software Quality ensures the quality of a design based on non-functional specifications or requirements of services or products. It supports the functional specifications or requirements of services or products such as maintainability, robustness, and the degree of correctness to which the services or software has produced.

The word quality is actually very hard to define or describe but it tells us that end system/ software/product should meet the needs of clients and customers and should fulfill their expectations. Quality actually relates to the system/ software/product design, functionality, usability, durability, reliability and price of the system/product/software. The word assurance is nothing but a positive declaration to and gives confidence to clients and customers about system/product/software that this system/product/software will works as expected. Software quality assurance (SQA) is a set of activities which ensure the quality of a process followed in developing software products. It is process based action and the activities used in software quality assurance (SQA) help to evaluate and organize the processes that produce software services or products. In software quality assurance the set of activities are proactive and staff function which only prevents services or products form defects whereas in software quality control the set of activities are reactive, line function which actually finds defects in services or products. Software testing is actually software quality control (SQC). In software testing products are tested and are released in live environment after proper testing

Functional testing is a technique in which the functionality of the entire system or software is being tested. This testing technique has many types like unit testing, smoke testing and regression testing etc. It is same as black box testing in which the expected result/output is known without the information of how it will be produced. Standardization technique gives an assurance that all modules of the software should follow proper standards, i.e., like documentation standards, development standards and quality control standards. It also ensures and improves quality of the whole software.Non-functional testing is also performed on the entire software. Performance testing is main type of non-functional testing. An example of performance testing is stress testing which is performed to check how the entire software or system works under heavy load.There are several advantages of software quality assurance (SQA). The main advantage is that it increases the client confidence. This is done by properly checking the quality of software at different levels like review, audit and inspection etc. It involves the interest of stakeholders and by giving weekly/monthly reports of requirement metrics and defects to clients helps in satisfying that work is being done on time. Software quality assurance (SQA) also saves money by finding defects in early builds of the software. After fixing in time increases client satisfaction and hence improves the reputation of the product in customers. Agile methodology improves the quality of the product, where both development and testing are done in parallel, improves the efficiency of the product by fixing the found bugs at early stages.

2. RELATED WORK

Software Quality Assurance (SQA) is a coordinated and comprehensive strategy to ensuring that software operations, goods and services confirm SQA's objectives of enhancing the quality of software by carefully controlling all software and production processes to ensure complete conformity with the principles and specifications established [1]. The quality assurance of applications must be used to align quality with productivity. Improvements in efficiency impact the efficiency of activities in different respects, such as growing sales, reducing expenses and enhancing productivity. Quality has been recognized in any sector as one of the key factors of sustainable strategy [2]. For the past two decades, software production has become one of the fastest rising industries. As the number of app production companies has risen at a much faster rate, global rivalry has become much more serious. Tech vendors need to distinguish their offerings in ways that are important to their clients in order to succeed in this global competition climate. An established way to accomplish this distinction is consistency. Three critical sources, namely humans, technology and management, generate quality in the software development [3].

Javed et al. [4] reported that emerging countries such as Pakistan are competing harder in the international market to sustain software quality. For several factors, software output is compromised. This article states that if experts don't successfully perform their assigned jobs, efficiency goes down. Performance may also be enhanced by preparing the persons who work on the software items. Educated individuals should pay attention when the product is being produced. Parnas[5] clarified that the divide and conquer approach may be used in applications to detect and eliminate problems/errors. In his study, David et al. [6] mentioned factors influencing efficiency. His analysis covers the following considerations: Duration of Project Period, Fund spent, less use of standards to attain consistency, Lack of relevant Field Experts, Duration of Project, and Determine efficiency regardless of inferior advantages.

As per IS0 DIS 13407 the user cantered design helps in achieving quality of a system. It states that user cantered design is an activity which has multiple disciplines, which include human factors, people's efficiency, people’s knowledge, and the techniques. The main objective here to enhance the efficiency and effectiveness of the system by improving the working conditions of humans. It also prevents the possible negative effects these working conditions on human’s mental and physical health. It also protects the human health performance and safety [7].

Human factors are the certain conditions which directly influences the quality, cost, and duration of the product. Human factors are very diverse and if some organization ignores it or not mitigating properly in a project then it will affect the project seriously [8]. Human factors can be environmental, technical, organizational, and managerial. Technological factors are the most important aspects in software development life cycle and results in the success of software product [9]. Many organizations which are developing software’s are in continuous struggle to tackle the issues related to human resources. These issues totally depend upon the office culture, training and team dynamics. [10] Software quality relates to a software which is fairly bug free or defect free, delivered in specified/proper time and within desired/expected budget, which at the end meets with the stated business requirements and is also maintainable. Functional quality and structural/non-functional quality are the two aspects of software quality in the context of software engineering.

Software Functional Quality ensures the quality of a design based on functional specifications or requirements of services or products. Software Structural Quality ensures the quality of a design based on non-functional specifications or requirements of services or products. It supports the functional specifications or requirements of services or products such as maintainability, robustness, and the degree of correctness to which the services or software has produced. The word quality is actually very hard to define or describe but it tells us that end system/ software/product should meet the needs of clients and customers and should fulfill their expectations. Quality actually relates to the system/ software/product design, functionality, usability, durability, reliability and price of the system/product/software. The word assurance is nothing but a positive declaration to and gives confidence to clients and customers about system/product/software that this system/product/software will works as expected. Software quality assurance (SQA) is a set of activities which ensure the quality of a process followed in developing software products. It is process-based action and the activities used in software quality assurance (SQA) help to evaluate and organize the processes that produce software services or products [11].

Software quality control (SQC) is something different from software quality assurance (SQA). It is a set of activities which ensure the quality in software products. It is product-based action and the activities used in software quality control (SQC) only focus on discovering defects in the actual/end services or products which is produced by following the activities used in software quality assurance (SQA). As an approach to resource optimization, SQA will minimize costs and, in economic terms, maximize benefits. A significant element in the life cycle of software production is SQA. Quality assurance should be included at each level of the development process. Monitoring and evaluation are used for device identification and avoidance of defects. The key objective of SQA is to specify the targets and planned performance and outcome verification requirements and to include the appropriate structure to ensure a reliable alternative to SQA across the construction phase, so that both software quality assurance and software engineering approaches can be used to improve the efficacy of both systems and technology designers [12,13].

DevOps is a series of approaches wherein designers and procedures interact and work easily, efficiently and with greater consistency to deliver applications and services[14,15]. DevOps shares roles and duties within a team that is entirely responsible for their operation and the related stack of technologies, from creation to implementation and maintenance. Cross features, joint roles and trust are encouraged in a DevOps setting. In turn, DevOps applies the agile group's agile growth targets through continuous delivery and launch. DevOps facilitates optimization of the phases of transition, setup and activation to facilitate continuous updates [16]. The demand for software production has seen strong growth with rapid improvements. Client expectations and improvement demands presented in a constructive way are the rationale for the accelerated changes. Application approach has solved this and several organizations have switched to application framework in order to facilitate regular launches with greater customer loyalty. While most businesses have adopted agile techniques, a few businesses have used such agile practices to mitigate the bottlenecks in the current phase. In Speedup, companies like IBM, Facebook and Microsoft began their own growth. Constant implementation is not an easy task, as there is a substantial effect on device reliability, as per [17].

3. METHODOLOGY

In this research, We identified a number of factors from literature review which are described below by two diagrams with detailed explanation named as organizational factors, technological factors human factors, factors at requirement stage, factors at design stage, factors at implementation/Coding, factors at testing stage and factors at deployment/maintenance stage. Figure 1 showing identified technological and organizational Factors.

4. IDENTIFIED SOFTWARE QUALITY ASSURANCE FACTOR

The factors affecting quality of software are identified in three different categories. The categories include Organizational factors, Technological Factors and Human factors. The explanation of each of them is given below in detail.

4.1 Organizational factors

As per IS0 DIS 13407 the user centered design increases the quality of a system by improving the working conditions of humans by keeping in mind that this will not disturb the human mental and physical health. So, by protecting the human health from any type of pressure resulted from the workload will results in better performance and speed in the development and testing process and thus will result in good quality product.Human factors are the certain conditions which directly influences the quality, cost, and duration of the product. So, an organization should not ignore it while developing a software product. Human factors also depend upon the environment of the software developing company, its culture, training, and team aspects [10].

4.2 Technological factors

ISO 9000 standard concerns with the quality of the product in quality assurance. The standard provides confidence and helps to assure that the resulted product will meets the desired product and requirement to satisfy the clients. Literally the quality depends upon the person who is responsible of making the requirement specification of a product. Quality directly correlates with the standard of requirement specification if it is inappropriate then it will reduce the quality of the end-product [15]. As per Garvin, the manufacturing quality is described by requirement specification and high-quality end-product confirms the appropriate requirement specification [18].

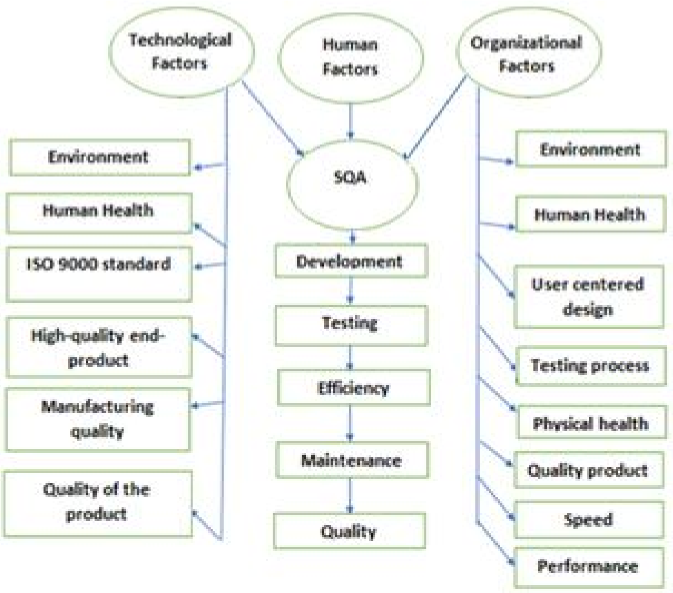

Figure 1: Technological and Organizational Factors

4.3 Human factors in SQA

The quality characteristics of a software directly relates with the quality, use of the software and on user. Functional and non-function aspects of the product clearly depend upon the type of user and number of users. Usability, efficiency, reliability, portability, and maintainability come from end user perspective. Quality here again depends upon the end user satisfaction and the performance of the product. For quality both degree of measurement and care should be considered, if a quality is measured under different environments then generalizing the consequences of quality measurement will provide care. Various takes implemented on the intermediate and end-product also helps in enhancing the quality of product. Tasks are compromised on different activities which are performed to obtain the desired goals. The approach of human factors in quality directly relates with the user center design and usability. User center design is something related with prototypes and mockups whereas usability relates with user interface. Thus, user center design and usability help in improving the quality of product.

2.3 Factors identified at different stages of SDLC

Following factors have been identified in this study for ensuring software quality assurance at different stages of SDLC (figure 2)

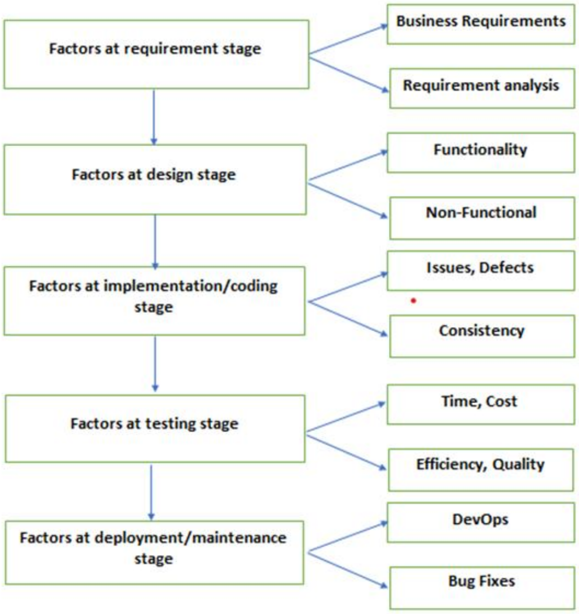

Figure 2: SQA Factors during SDLC stage

3 RESULTS AND DISCUSSION

Quality plays an important role in achieving high quality of end product. Organizations perform testing by the help of SQA Engineers in order to meet the goal and to satisfy the clients. The quality characteristics of a software directly relates with the quality, use of the software and on user. Functional and non-function aspects of the product clearly depend upon the type of user and number of users. For quality both degree of measurement and care should be considered, if a quality is measured under different environments then generalizing the consequences of quality measurement will provide care. Various takes implemented on the intermediate and end-product also helps in enhancing the quality of product. Quality increases the client confidence, and which is achieved by properly checking the quality of software at different levels like review, audit and inspection etc. It involves the interest of stakeholders and by giving weekly/monthly reports of requirement metrics and defects to clients helps in satisfying that work is being done on time. Organization saves their money by the help of Software quality assurance (SQA) to find defects in early builds of the software. After fixing in time increases client satisfaction and hence improves the reputation of the product in customers.

Each factor identified at different stages in SQA has its own role in achieving high end product. First start with business requirements stage involves the gathering of all the requirements from the business owner. It is the very early stage in software development. After collecting all the required information from the business owner there comes requirement analysis stag in which all the product designers give his/her thinking idea. These ideas are then shared with the business owner. The business owner than has right to select an idea. Therefore, design quality directly depends upon the level of designers hired in the project. Consistency plays an important role to satisfy the clients. Technical vendors need to distinguish their offerings in ways that are important to their clients in order to succeed in this global competition climate. An established way to accomplish this distinction is consistency. So technology has direct impact on the quality of the product. Using latest technologies and tools will increase the quality and project management of the product. The approach of human factors in quality directly relates with the user center design and usability. User center design is something related with prototypes and mockups whereas usability relates with user interface. Thus, user center design and usability helps in improving the quality of product. In every quality assurance activity, design phase is still the most relevant; SQA team members are responsible and accountable for designing project scope and executing the plan as well. They also were liable for the calculation of output, enhancement of quality and control of configurations. The external quality comes in product specification stage and it is a part of product validation. The system behaviour points towards external requirements of product quality whereas external metrices points towards external quality. During real time software execution these external metrices are being applied the consequences of which clearly reflects the degree of the quality of software. A product works fine in an environment it might show defects in another environment. So, quality is clearly dependent on the user-identified environments. The internal quality requirements come in design and development stage and software attributes points towards it. Internal metrics points out the internal quality and are part of product verification. So, validation process focuses on the external quality and its requirements whereas verification depends upon the internal quality and its requirements. During development process the internal quality requirements are considered. These metrices are applied during the development stage i.e., directly on the source code. The results of which helps in improving the quality of source code. Better code quality results in low quality defects. The feedbacks of both internal and external requirements help to improve the quality of code and desired product.

The competence of adeveloper also plays an important role in attaining high end product. Writing high level of software bug free code will save time and money. Also after completion of the program, testing is not mandatory, but it can be conducted at any onset of the study and expose the coding defects. Peer Evaluations play a very significant role in enhancing the efficiency of projects. This shows that user participation is often very relevant for reviews and can be changed very quickly on the grounds of this reviews app consistency. It is quite essential for business organizations to retain product consistency, when any company runs towards automation. Software testers play a very important role in providing end product to clients which is defect free. Because high level of testing will leads a high quality of software. Technology here will again help SQA team to attain client's goal. Creation of technology is a human operation; multiple techniques of production management and quality control used to produce at least value an excellent performance, accuracy, and error free product. Quality assurance should be included at each level of the development process. The general view of software testing is that the activity of finding bugs, everyone wants software quality, project manager looks at it as delivering a good software to a customer, and developer wants to produce product and getting things done fast, and user know it as stable and reliable.

The last stage in SDLC in deployment/ maintenance of the software product.DevOps here will play an important role. It is an intermediate between developers and tester. It is a series of approaches wherein designers and procedures interact and work easily, efficiently and with greater consistency to deliver applications and services. Dev from Programmers and Operations from Service ended up with the term 'DevOps.' DevOps shares roles and duties within a team that is entirely responsible for their operation and the related stack of technologies, from creation to implementation and maintenance. DevOps is a synthesis of trends aimed to promote cooperation between growth and activities. As well as common systems and instruments, DevOps discusses shared priorities and benefits. Shared priorities and rewards cannot always be feasible due to natural conflicts between different parties. They can, though, at least be associated with one another.

4 CONCLUSION

While many software quality engineers developed software quality assurance plans. The research has focused on the different factors of SQA in SDLC. Each stage of SDLC plays important role in achieving high quality product and SQA team should be the part of each SDLC activity. Knowing the right clients' requirements, reviewing the design developed by the designer's team, helping development team in unit testing and then following the correct testing techniques, SQA can increase the quality, save time and cost of the product. Knowing the right functional and non-functional aspects of a product at requirements stage can help SQA to improve the process which will be used in quality control. It is also important that SQA team should know what is happening in design and development stage. Therefore, by involving SQA in each stage of software development life cycle the process of testing can be improved.

REFERENCES

1. Mirnalini, K. and Raya, V.R., 2010, September. Agile-A software development approach for quality software. In 2010 International Conference on Educational and Information Technology (Vol. 1, pp. V1-242). IEEE.
2. Iftikhar, A. and Ali, S.M., 2015. Software quality assurance a study based on Pakistan's software industry. Pakistan Journal of Engineering, Technology & Science, 1(2).
3. Fernandez, P.O., David, M., Duma, D.C., Ronchieri, E., Gomes, J. and Salomoni, D., 2020. Software Quality Assurance in INDIGO-DataCloud project: a converging evolution of software engineering practices to support European Research e-Infrastructures. Journal of Grid Computing, 18(1), pp.81-98
4. Javed, A., Maqsood, M., Qazi, K.A. and Shah, K.A., 2012. How to improve software quality assurance in developing countries. Advanced Computing, 3(2), p.17
5. Parnas, D.L. and Lawford, M., 2003. The role of inspection in software quality assurance. IEEE Transactions on Software engineering, 29(8), pp.674-676.
6. David L.Parnas& mark Lawford, (August 2003). The role of inspection in soft quality Assurance" IEEE transaction on software engineering, vol.29, (pp.8).
7. David L.Parnas& mark Lawford, (August 2003). The role of inspection in soft quality Assurance" IEEE transaction on software engineering, vol.29, (pp.8).
8. Dash, D., & Dash, R. (2010). Risk assessment techniques for software development. European Journal of Scientific Research, 42(4), 629–636.
9. Dhlamini, J., Nhamu, I. &Kachepa, A. (2009). Intelligent risk management tools for software development. Proceedings of the 2009 Annual Conference of the Southern African Computer Lecturers' Association, 33–40.
10. Taveras, Pedro. (2015). Human Factors in Software Engineering. 10.13140/RG.2.1.1362.0320.
11. Maher, Z. A., Sani, N. F. M., Din, J., &Jabar, M. A. (2016). Use of Security Patterns for Development of Secure Healthcare Information System. Journal of Medical Imaging and Health Informatics, 6(6), 1541-1547.
12. Elyezjy, N. T. (2012). Software quality assurance SQA-practices in

Palestiniangovernment. International Journal of Computer Applications, 58(22).

13. Koondhar, M. Y., Abdul Molok, N. N., Chandio, F., Rind, M. M., Raza, A., & Shah, A. (2015). A conceptual framework for measuring the acceptance of pervasive learning.
14. Koondhar, M., Malkani, Y., Rind, M., Chandio, F., Shah, A., &Nizamani, Q. (2015). Pervasive Learning using Cloud Computing. Sindh University Research Journal-SURJ (Science Series), 47(3).
15. Koondhar, M., Rind, M., Hub, M., &Shaikh, H. (2014). Semi-automated E-Tendering System with Web Services. A case study on tendering process in Pakistan. Sindh University Research Journal-SURJ (Science Series), 46(4).
16. Claps, G. G., Svensson, R. B., & Aurum, A. (2015). On the journey to continuous deployment: Technical and social challenges along the way. Information and Software technology, 57, 21-31.
17. Bevan, N., & Azuma, M. (1997, June). Quality in use: Incorporating human factors into the software engineering lifecycle. In Proceedings of IEEE International Symposium on Software Engineering Standards (pp. 169-179). IEEE.
18. Maher, Z. A., Shaikh, H., Khan, M. S., Arbaaeen, A., & Shah, A. (2018, November). Factors affecting secure software development practices among developers-An investigation. In 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS) (pp. 1-6). IEEE.