Empirical Assessment of Software Quality Assurance Performs and Challenges in A Development

Muhammad Masood Ahmad Khan[1], Nighat Naz[2], Anita Venaik[3], Ahmed Bin Sana[4]

[1,2]Phd Scholar, Department of Information Technology, Islamia University, Bahawalpur-Pakistan.
[3]Professor, Amity Business School, Amity University Noida-India.
[4]Instructor, Logical Creations Education Research Institute, Quetta-Pakistan.
masoodkhan12345@gmail.com, nighat.naz90@gmail.com, avenaik@gmail.com

ABSTRACT

Quality assurance is critical in the process of software development because it ensures that the software system and other software products are reliable and simple to maintain. Quality standards, Quality control, process automation, quality management, quality planning, and improvement are only a few of the tasks that make up software quality assurance. While our work focused on quality planning, observance to standard methods, and the issues that come with them, our scope is to comprise quality control, software process development, and participation in global quality standards associations. The goal is to produce more reliable findings that can help the software community make better decisions. To collect data from software practitioners, a qualitative research technique was used, particularly the use of questionnaire research instruments.

INTRODUCTION

During software development, software quality assurance (SQA) is a planned and methodical decoration of actions that provides enough confidence that a software product complies with specifications. SQA is a set of approaches and techniques for assessing software development methods, as well as the procedures, tools, and technologies required to assure the quality of the product. To ensure the integrity and durability of software, SQA is normally performed with well-defined standard procedures, including tools and techniques for quality control. SQA model, certification, SQA case studies, appraisal, and SQA standards and models, improvement, data analysis, and management of risk are just all areas of research in the turf of software validation, verification, and analysis.

The use of a methodical, disciplined, and quantitative method to the creation, operation, and care of software, as well as the education of these methods, is known as software engineering (IEEE Standard 1990). It is an engineering field, according to Somerville (2007), which is concerned with all rudiments of software creation, from the primary phases of system description through system keep after it has been put into use.

LITERATURE REVIEW

Quality

Bell Laboratories was the first to explicitly implement quality in 1916, and it progressively infiltrated into software development in the 1970s, once military applications were, being developed (Lewis 2008). In the software engineering sector, quality is not constrained to established standards, as it is in other engineering disciplines such as manufacturing; excellence in the semantic of software engineering, as articulated by Lewis (2009), means 'meeting conditions' and 'fitness for use.' This means that the programmer satisfies the users' needs as specified in the requirements requirement.

Management of Quality

Altogether planned systematic actions and procedures for producing, regulating, and insuring quality are included in quality management. It is more than a duty; it is a habit that must be established in the culture of a firm (Ebert 2011). It also tries to track and improve the development process, assuming that the excellence of the development process has a direct impact on the quality of the final product.

Process of Software quality assurance

We defined process of Software quality assurance such as, repeatable process for studying core control devices and adherence to software values and procedures that is link with project administration and software development lifespans. The process' area is to ensure constraint adherence, reduce risk, evaluate interior controls, and improve quality while remaining within the specified time and budget confines (Khazanchi & Owens 2009).

Experiments

- SQA initiation occurs prior to the start of a project, with the SQA team being alerted and the relevant quality control and audit mechanisms being developed.
- SQA top reasons the software quality assurance plan's goals and objectives, as well as the quality processes or techniques to be surveyed, the standards and metrics to be used, and the reviews and audits to be conducted.
- Authentication of requirements for feasibility, testability, and fullness (requirements assurance).
- Design assurance ensures that the design meets the requirements and that the intended methodologies are followed.
- Assurance of development, ensuring that the expansion team adheres to the established development process and coding standards.
- Assurance Testing verifies that sufficient testing was performed and that faults were recorded, tracked, and corrected.
- Implementation guarantee ensures that all required application processes have been fulfill both before and after implementation.
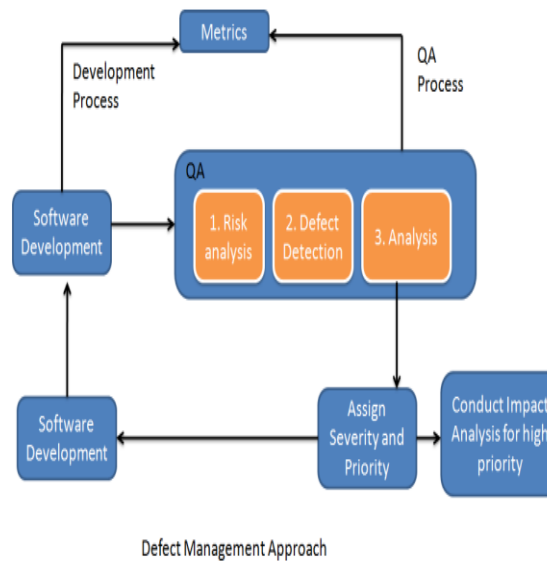
Figure 1: Quality Management process

Planning of Quality

In this process of creating a customized quality strategy for a certain project. It entails a set of structural principles tailored to the software project at hand as well as the development method to be employed.

Control of Quality

In this process of observing the software development process and verifying the invention or deliverables in order to ensure that, the development team adheres to the quality plan and administrative ethics and techniques. Quality control refers to a collection of software engineering procedures for ensuring that individually work product fulfils its quality objectives (Pressman 2011). It frequently entails the use of software metrics for measurements. Any discovered breach of quality standards is noted and submitted to the appropriate persons for repair.

Team work of software quality assurance

Despite the fact that each member of the general project team is accountable for ensuring project quality, there is still a critical need for a specialized staff devoted to quality assurance. Quality assurance used to be the job of the person who developed the product, but that is no longer the case, (God bole, 2004).

Software quality assurance Costs and assists

It is impossible to overstate the importance of software quality assurance. It has been proven that a lack of it is one of the leading causes of software project failure. It is extremely important in the software development life cycle and may significantly improve the chances of a project's achievement. It also aids in the reduction of potential dangers (Owens & Khazanchi 2010).

Software quality isn't accomplished by accident; a product doesn't meet all of the requirements by accident. It is the outcome of time, money, and effort spent on purposeful acts and steps. While guaranteeing quality has a cost, so does an absence of quality. The cost of quality may be broken down into three categories: prevention, assessment, and failure. The expenses of planning and coordinating actions in the SQA are included in the preventative costs. Internal failure costs are based on faults discovered before to shipping to the customer, whereas external failure costs are based on problems discovered after deployment at the client's location (Pressman 2011).

Challenges & implementation of software quality assurance

In their effort to develop great-quality software and achieve client approval, software businesses regularly confront several hurdles (Elgebeely 2014). Inadequate tools to help automate the process, a lack of familiarity and information of the process, a lack of management exercise on quality standards, an absence of a quality management framework in the institution, top management disapproval, opposing beliefs and opinions, and the process' previous futility.

Pitfalls in SQA

A number of typical problems done by software businesses in an attempt to assure quality were discovered and described in this sub-section based on the literature research.

Without first establishing effective software quality assurance practices within individual departments, software firms frequently rush into implementing a software quality assurance process (Scarping 2012). Quality assurance should be imposed initially at the departmental level before some all comprehensive procedure is implemented at the top level, in principle.

SQA should not be regarded as only the job of the SQA team, but rather as a collective obligation of everyone participating in any activity across the software development. Every employee should be well educated about what is expected of them in order to make sure quality in whatever role they play. Secondly, SQA encompasses much more than testing and should not be left until the conclusion of the project; rather, it should be adopted from the start.

METHODOLOGY

In this study, the quantitative research approach was used. The qualitative data was evaluated using evocative statistics after the survey approach was utilized. To create the study enquiries and the investigation tool, a closed-ended feedback form, a thorough works evaluation of the activities complicated in software quality assurance controlling was undertaken. To verify the questions' acceptability, a software quality specialist and a calculator evaluated, validated, and confirmed them.

The inquiry forms were then deliverer to players in software development in Lagos, which is the core of Nigeria's software industry and household to about half of the country's software enterprises (Saroyan and Heems 2004). The information gathered was compiled and reviewed.
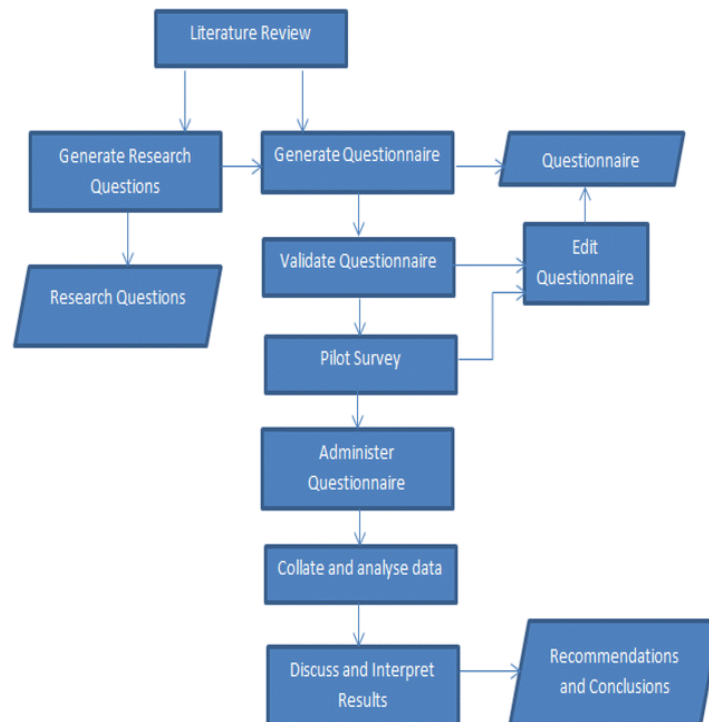
**Figure 2: Diagram of Methodology of SQA**

RESULTS

In calculation to the above findings, it was discovered that quality assurance methods are mostly ignored, which might be a contributing factor in poor patronage. Likewise, software consultants are ignorant of international values organizations or the essential process upgrading techniques; as a consequence, their demanded standards which is not fulfill the aligned with those of credited bodies, and are incomplete to their local knowledge and understanding, making them suspect. The results were comparable when compared to Turkey, indicating that they are typical of poor countries. The internal consistency of the study instrument was verified using Cronbach's alpha, and it was founded to be dependable.

This section summarizes the entire project's outcomes. The answers to the extra research questions are also included. There were 86 surveys in all that were evaluated. The internal consistency of the study instrument was assess using identical Cronbach's alpha, commonly known as the coefficient alpha, to determine its reliability. Since they slow discrete components of the SQA and had dissimilar Likert scales, this was compute on different portions of the questionnaire.

Analysis of Gathering Data about Quality Maintain:
Report the findings on the quality standards table and figure. Just 2.3 percent of defendants specified they have no sympathetic what standards of quality are, though 11.6 percent indicated their firms did not survey them.
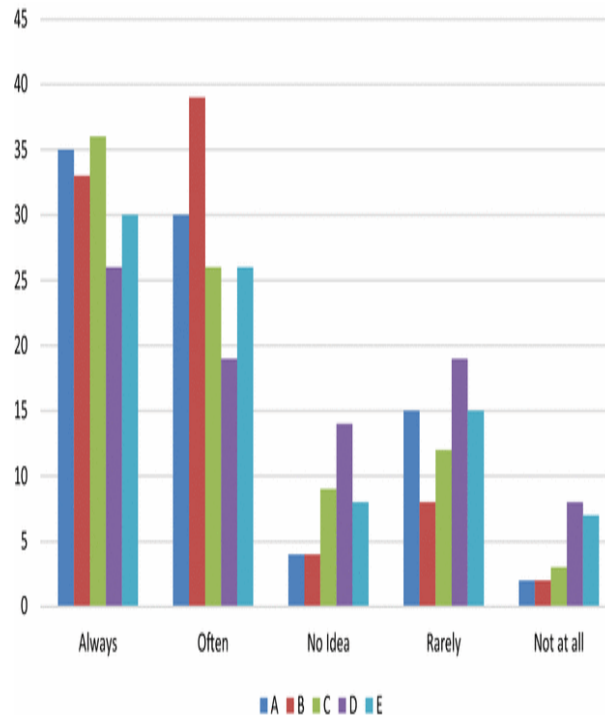
Quality Standard:
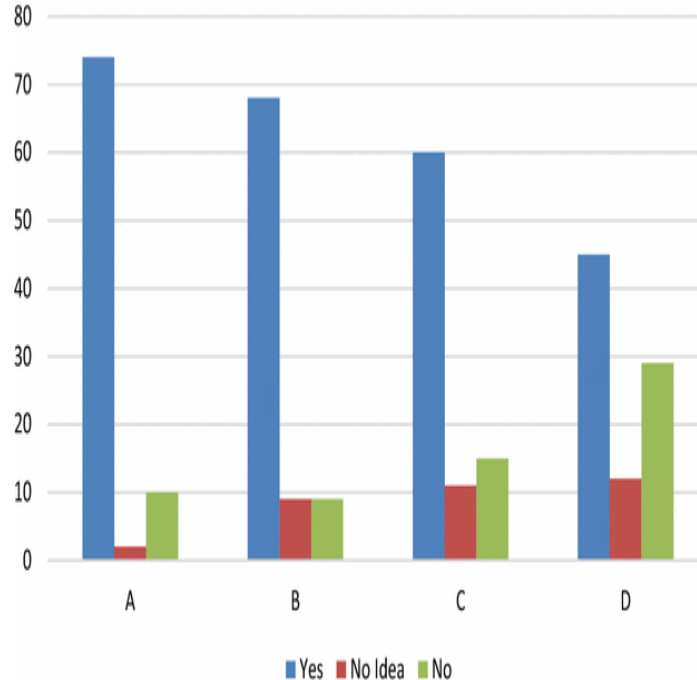
Graphs 1: Quality Standard:

Only 33.7 percent of companies do not have a distinct SQA team from the development team, and 30.1 percent either do not have or are unaware of such a team.

Table and Fig show the results of the quality planning survey. Overall, 22.1 percent of respondents said they do quality planning either seldom or never, while only 36 of the 86 said that they do risk management activities all of the time.

Table: 1 Quality Planning:

| Questions	Always	Often	No idea	Rarely	Not at all
Do you develop a quality plan for new software projects before their commencement? (A)	35	30	4	15	2
How often do you use a project schedule plan for each software project? (B)	33	39	4	8	2
Do you carry out risk management activities for your software projects? (D)	36	26	9	12	3
Do you use a budget plan for every software project? (E)	26	19	14	19	8
Do you plan the tests and reviews (e.g. write the test cases) in advance	30	26	8	15	7

Graphs 2: Quality Standard:

Quality Planning

The CONTROL of QUALITY AND MEASUREMENT ACTIONS ARE PERFORM, AS SEEN IN TABLE AND FIGURE, NONETHELESS ONLY 22% REPORT THAT THEY EMPLOY AN OUTSIDE REVIEW SIDE ON THEIR PROJECTS. SOFTWARE TESTING, PERIODIC REVIEWS, & CODE WALKTHROUGHS, ON THE OTHER HAND, ARE JUDICIOUSLY PERFORM.

Table 2: Quality control/measurement

Questions	Yes	No idea	No
Do you carryout periodic reviews on on-going software projects? (A)	77	5	4
Do you employ an external review team for your software projects? (B)	19	8	59
Do you carryout review of your software documentations? (C)	63	6	17
Do you carry out testing of software before releasing it to customers? (D)	82	1	3
Do you carryout regression testing of software after making any modifications? (E)	66	11	9
Do you engage in code walkthroughs as a means of inspecting codes? (F)	57	16	13
Do you engage in design walkthroughs before coding? (G)	53	21	12
Do you carry out software quality assurance audits? (H)	57	18	11
Do you have a configuration management and change control system? (I)	44	25	17
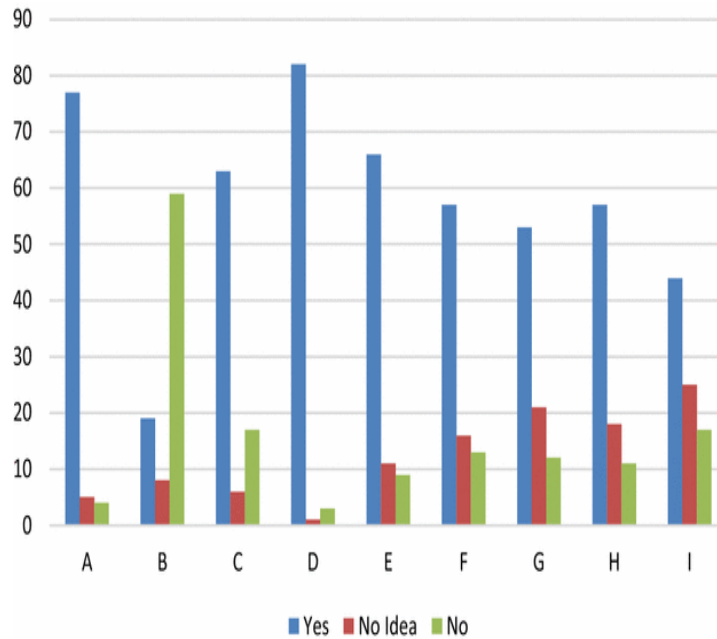
Graphs 2: Measurement/ Quality control

In terms of process development initiatives, 75.6 percent said they enhance their processes founded on metrics from past projects, however no enterprise has confirmed this. Only 16 percent are listed under the ISO 9000 pledge replicas, according to the first phase of the poll, with 57 percent having no knowledge of the CMMI. According to the extra questions included in the second round of the survey, 86.3 percent of respondents are ignorant of international issues.
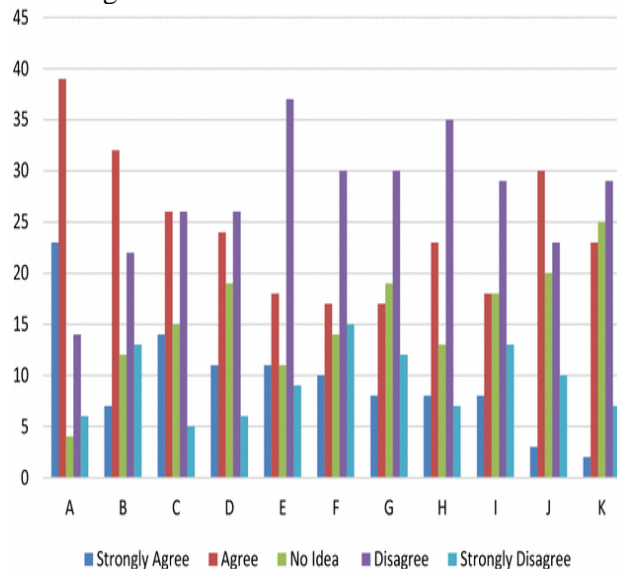
Graphs 3: Challenges

DISCUSSION

According to quality standard of development Process follow procedures. A distinct SQA team is not use by 33.7 percent of respondents. Despite the importance of a software development government's SQA team, more than half of defendants do not have one. This means that no quality checks are performed on software packages before they are sent to users, with the exception of those performed by the designers. This is extremely dangerous since detecting a problem or possible risk in a software programed normally necessitates the use of a second pair of eyes. 35 percent of respondents said they do not have a SQA squad or do not know what one is.

CONCLUSION AND RECOMMENDATIONS

Conclusion

The study examined practitioners' general software quality assurance procedures in a developing republic. The study, which was motivate by the essential to reduce software introduction into Pakistan and upsurge patronage of native software companies, has revealed some possible reasons for the industry's current state. Commendations have been made to address the current threat and enhance quality software processes, which, if followed, will result in the creation of high-quality software posts that will be popular and last.

Software assurance methods must be taken seriously in order for the software sector in developing nations to flourish and become a sustainable basis of external income since their impact is visible in the product. Furthermore, excellent frameworks and tools that demand the least amount of time and money are critical in these nations.

Recommendation

Following the discussion of the results and findings, the following suggestions are made.

Because many organizations regard quality standards and practices as an added expenditure with no recompense, the Institute of Software Practitioners of educate its affiliates on the necessity of adhering to them. They should be educated that, while ensuring quality may appear to be costly at first, it is really less expensive in the long term because not complying may be more costly.

Software firms must establish up a quality assurance team, which should ideally be independent from the development lineup. They should not be involved in any other stages of growth in order to be in the best position to spot defects in software and other crops. Memberships of the growth team should be well taught not only in practical areas, but also in the organization's quality standards and industry regulating organizations on a national and worldwide level.

REFERENCES

Swami, O.Y., Misra, S., Fernandez-Sanz, L., Crawford, B. and Soto, R. (2016). An empirical evaluation of software quality assurance practices and challenges in a developing country: a comparison of Nigeria and Turkey. Springer Plus, 5(1).

Huang, C.-Y., Leung, H., Leung, W.-H.F. and Mizuno, O. (2012). Software Quality Assurance Methodologies and Techniques. Advances in Software Engineering, 2012, pp.1–2.

Shen, P., Ding, X., Ren, W. and Yang, C. (2018). Research on Software Quality Assurance Based on Software Quality Standards and Technology Management. IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/8441142.

Hindawi. Software Quality Assurance Methodologies and Techniques. Hindawi. Available at: https://www.hindawi.com/journals/ase/si/274738/ [Accessed 11 Dec. 2021].

Research Gate. (PDF) Software Quality Assurance in Software Projects: A Study of Pakistan. Available at:
https://www.researchgate.net/publication/287092570_Software_Quality_Assurance_in_Software_Projects_A_Study_of_Pakistan.