IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

*International Advanced Research Journal in Science, Engineering and Technology*
*Vol. 3, Issue 6, June 2016*

# The Effectiveness of Implementing Total Quality Management in Software Development

**Samar Ahmed Alamri[1], Azrilah Abdul Aziz, PhD[2]**

Department of Information Systems, Faculty of Computing and Information Technology,

King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia[1, 2]

**Abstract:** The following paper examines the effectiveness of total quality management regarding the software development process. Sufficient discussion focuses on the real-world issues as well as the methodological concepts of software development that affect quality. While a significant portion is dedicated to the examination of scholarly articles and publications on the subject matter, this paper includes problem statements and presents several alternatives, solutions, and approaches for effective quality management. The conclusion consolidates the research and recommendations discussed throughout the paper.

**Keywords:** quality; quality management; stakeholders; communication; software; development process; failure; tools.

## I. INTRODUCTION

Since competition in the computer industry has increased and cheap applications have become widely used, the importance of software development has grown significantly. Software development is a task that involves a number of attributes ranging from the processes involved to the final product required. This means that software development activities have to be performed carefully to achieve meaningful, useable software as the final product. Any software development team should be equal to the task and evolve with the needs of the customer for the software to be of high quality. This means that following the specific methods of TQM is an ideal procedure for realizing customized, high-quality software and a necessary condition in the competitive market (Li, Chen, & Cheung, 2000).

The specific set of tools and procedures used in software development is highly dependent on the nature of the desired product and the method of implementation. These procedures, often referred to as the methodology or software development model, aim to lay down the process for software development. In addition, the tasks and stages are well defined. A good methodology enhances quality by incorporating all its elements, as required by the philosophy of total quality management (Mcmanus & Wood- Harper, 2007).

## II. LITERATURE REVIEW

Bon and Verheijen (2006) state that many scholars and practitioners in the software development field have published works that address the issue of quality. While some of them pay particular attention to the development of software applications, most of them adopt an organizational approach in an attempt to incorporate the contextual people issues and technical issues that are likely to be encountered in software development.

Subramanian, Jiang, and Klein (2007) discuss the use of the capability maturity model (CMM), which enables the enhancement of software processes through the implementation of total quality management and is used in most organizations. In addition, the introduction of this strategy allows continuous improvement of the quality of products as well as an increase in investment returns. TQM is the prime interest for organizations within this sector, since it enhances the implementation of projects globally. However, companies in this sector face challenging factors, such as improper training of the employees, unfavourable managerial strategies, and poor software quality resulting from a poor motivational approach.

Li, Chen, and Cheung (2000) argue that the quality of software products has traditionally been guaranteed by the application of software quality assurance techniques (SQA) in the development process. Today this method is not enough to produce the quality of software products demanded by clients, hence the use of TQM strategies in all software development companies. Consequently, software development teams provide an on-the-job training programme to help workers to gain experience and knowledge for the purpose of building product quality. Nevertheless, the effective implementation of TQM has been affected by the communication gap between the developers and the users of software products as well as an organization culture that believes that quality is meant only for workers and not for managers.

Glowalla and Sunyaev (2015) recognize that the high level of failure of software and information systems programmes is a current issue with great importance for researchers. In addition, the quality of the product and development process is an essential part of software project success. Total quality management has been adopted to provide causal structures enabling and enhancing continuous quality improvement of the software development process as well as the resulting product. Nonetheless, the lack of reusability methods in the

development process has led to waste and constant errors in software products.

According to Kan, Basili, and Shapiro (1994), quality involves both product quality and customers' satisfaction, and in the software development process, productivity analysis for software programs is an important strategy followed to improve quality. In addition, despite variations in its implementation, companies in this sector have adopted TQM to achieve customer satisfaction by studying their needs, gathering their requirements, and managing their fulfilment.

Jha (2006) summarizes the quality concerns in software development by underscoring the need to stay focused on the strategic goals of the organization. Without this focus, the software development process would not be customer-centric, thereby undermining quality.

A. Issues and difficulties relating to TQM implementation in software development

The failure of managerial strategies within software development companies has contributed to the lack of proper communication between businesses and their users (Subramanian et al., 2007). In addition, employees have not been involved in the management process; hence, a lack of motivation has contributed to the poor quality of the products. Moreover, the incompetence of the management has led to poor implementation strategies, which have resulted in errors in projects and reduced returns on investments.

The inherent problem highlighted by Li, Chen, and Cheung (2000) is that employees' motivation induces them to aspire to produce quality products. Nonetheless, poor materials, ineffective plans, broken tools, and delayed schedules affect the workmanship during the software development process. In addition, the lack of management support has affected the implementation of TQM. Workers lack appropriate knowledge of statistical methods that help them to understand the nature of variations, their causes, and how to manage them.

Glowalla and Sunyaev (2015) research argues that the top management is only committed to success, risk factors, and project issues; hence, it is not interested in software project quality. For this reason, it acts as a barrier to the implementation of TQM within an organization. In addition, the use of a standard method to address individual cultural and social issues tends to be ineffective. Consequently, there is a problem of a high level of business competency in modern information technology, which results in software developers' reliability, hence avoiding the involvement of users within the project development.

Kan et al. (1994) argue that the application of TQM as a strategy and improvement method of operational quality in software engineering is rarely understood and in most cases ignored. In addition, there is a challenge concerning researchers transferring software technologies to development corporations, hence filling the gap between the state of art and the practice. Firms tend to consider TQM as a short-term financial investment and the responsibility for quality as lying with the workers rather than the management. Consequently, poor leadership, differentiated participation, and poor communication between developers and users has led to multiple errors recognized in most software projects.

Since any software development company should have the capability to develop quality software that evolves with the desires of the customer, using the definite approaches of TQM is a perfect procedure for acquiring tailored, high-quality software, which a significant requisite in a competitive marketplace. Recent research by Mcmanus and Wood- Harper (2007) also contends that the application of TQM as a strategy and improvement method of operational quality in software engineering is rarely understood and in most cases ignored.

Bon and Verheijen (2006) stipulate that it is not worthwhile carrying out certain innovative transformation procedures with a low-performance software development organizational structure. In addition, software isolation works unproductively and marginalizes productivity in TQM. Such excellence centres on developing and firming the software systems and the progression of an IT organization to increase its performance and build value for its shareholders.

Jha (2006) states that little analysis and research have been conducted regarding the perspective of TQM as an idea or business excellence implementation approach seeking to integrate the notion of enterprise resource planning (ERP). Classifications of quality have transformed with the elapse of time with varying customer desires and requirements.

B. Solutions

Subramanian et al. (2007) suggest the need to enhance the academic research on software development through the use of the CMM. In addition, organizations should increase their efforts to create a positive and dynamic working environment and apply quantitative methods and analytical techniques. Computer software organizations' management needs to focus on their employees' requirements for the purpose of improving the quality of products. Consequently, companies should ensure simplicity in their products as well as meeting the requirements. This strategy aims to enhance customer satisfaction by tending to use less complex software obtained through the use of diverse information system strategies.

Li et al. (2000) emphasize that software development requires a joint effort between the developers and the users. For this reason, open communication among the different functional areas and knowledge in various disciplines are essential for effective TQM implementation. Additionally, strategy transformation needs to be adopted by each member, since everybody belongs to a team. It is therefore recommendable for an organization to change the culture that controls the performance of its employees. Organizational goals and work principles that focus on numbers and not quality should be eliminated. Non-conforming software development projects and services should not be tolerated, and the executive should enhance accuracy by reducing the errors, waste, and rework among employees through motivation and continuous learning.

Glowalla and Sunyaev (2015) suggest the need to differentiate and involve the users because of the increased requirements for social interaction in TQM. In addition, clients' involvement has a substantial and conclusive effect on software development success. Consequently, the developers and users should have a basic knowledge of the strategic goals of such projects and understand the importance of quality at the high supervision level. For this reason, the executive needs to develop higher-quality systems that need to be recognized and well managed.

Kan et al. (1994) recommend ensuring good communication between the developers and the users as well as within the team. Improved communication will reduce the errors occurring at different stages of the development process if the developers run the application domain and the clients run the design realities at the same time. Further, modern information technology should be adopted to build effective communication change for a quality flow of information. Moreover, the management needs to have a proper tracking system for the whole development process to address the data validation issue, hence improving the quality of the data. Methods of measurement for software projects should be evaluated well before being applied, since some might be harmful to the quality enhancement actions of the company.

A high level of specialization is critical to the realization of quality goals. This is because the larger objectives are often broken down into smaller tasks, which form the yardsticks for the evaluation of progress and quality (Bon & Verheijen, 2006). These teams should be based on specialization, knowledge, abilities, and skills. Such an approach enables the entire development team to harness the strengths and competence of every member and stakeholder (Jha, 2006). As a result, every constituent task or process will be performed well, leading to the realization of quality. The principles of total quality management require specific tasks to be allocated to specific personnel, who have knowledge and experience in such undertakings. Teamwork promotes knowledge sharing (Donaldson, Siegel, & Donaldson, 2001).

Computer software organizations' management needs to focus on their employees' requirements for the purpose of improving the quality of products. Such companies should be able to host large and mostly inclusive developer assessments and performance metrics in the industry. From definite time- and goal-oriented procedures seeking individual, ideal solutions to fragmentary incremental quality development processes, TQM can help to transmute a business. It is believed that open competition and a civic- and quality-centred approach can provide exciting and fresh products and accompaniments that are inadequate only due to the essence of open innovation.

## III. IDENTIFICATION AND QUANTIFICATION OF QUALITY METRICS AND ATTRIBUTES

According to Jha (2006), while quality is difficult to quantify, it can be defined and qualified using a set of attributes. Usually, these attributes seek to measure the extent to which the software product meets the expectations and solves the problem at hand. In most cases, time and performance are critical metrics of quality. In the non-computing world, high expectations are placed on software products in regard to performance and the time-saving benefits that they offer through faster and more dynamic processing and retrieval of data. Actually, these are the very reasons that most businesses seek to develop software applications. If these expectations are not met, all the other attributes of the deliverable are rendered useless.

The availability of additional features is another metric of quality. Clients and users of applications are often impressed when the delivered software application can perform more tasks and offer more features than it was intended to. Such features, most of which are rarely detailed in the requirement specification and analysis stage, include different levels of access, audit trails, and the ability of an application to roll back changes. All these are metrics of quality.

Perhaps the most important element of the identification of quality metrics is the enlisting and rating of the desired attributes. For example, a certain attribute can be rated from zero to ten. In such an approach, the quality of the developed application can be expressed as the aggregate of these values. Notably, some attributes should be weighted more than others. For example, a hospital application should be evaluated more on its ability to produce statistical data to identify trends than on its ability to roll back erroneous changes made by a counsellor at a clinic. Without such approaches, the implementation of quality would be ineffective.

## IV. USER INVOLVEMENT FOR QUALITY MANAGEMENT

Usability and acceptability are a critical element of quality, just like performance and functionality. If a software application is to be used by a number of people within the field of application, then it is imperative for user involvement to feature prominently in the design phases. Secondly, if the application is to be used by people who are not well versed in computing, then the design of the interface should be aimed at this specific set of novice users (Vliet, 2010). The interface should be intuitive enough to mitigate the chances of user rejection and consequently project failure. Sometimes, systems fail not because of their inherent weaknesses in performance and functionality but because of the people issues surrounding them, such as acceptability and usability among novice users.

Therefore, the involvement of users in the specification of the requirements is imperative and by far the most important element. Communication is as important as effectiveness and competence in the delivery of quality software applications (Subramanian et al., 2007). If the development process does not involve all the stakeholders, then they would feel that the system does not address their needs and is likely to introduce undesired change. This is more of a social and psychological than a technical issue in quality management.

Like the construction of any other computing artefact, be it hardware or software, faults in the design phase cannot be

addressed effectively later. Therefore, the initial stages should pay attention to the needs and anticipated experience of the users; failure to do so would result in the artefact being rejected or adopted reluctantly, thereby compromising its usage and quality. Feedback and follow-up are critical elements of user involvement in the implementation of total quality in software development. It is not uncommon for a consensus to be reached among developers and users only for the team to realize that the two parties had different understandings of the underlying concepts and concerns. Therefore, feedback and follow-up should be included at every possible stage, especially during the testing and prototyping phases. For example, a working or non-functional prototype complete with a user interface can be developed and presented to the users so that they can review it critically. This enables the users to gain a feel of what to expect and voice their concerns about the application under development.

## V. TQM IMPLEMENTATION TOOLS

This section will identify the specific total quality management (TQM) tools that have been used in the implementation of process improvements.

- Capability maturity model (CMM)
- TQM method
- Deming's TQM method
- Quality seven (Q7) and management seven (M7) tools

Models of the development process to improve quality during development:

- Total quality control (TQC)
- Six sigma strategy
- Market-driven quality
- Plan–do–check–act
- Quality improvement paradigm (QIP)/experience factory (EF) organization
- Lean enterprise management

Subramanian et al. (2007) point out the difference in software development management brought about by the use of the capability maturity model (CMM). The CMM is one of the tools or methods used in software process improvement (SPI). It consists of five levels, which are initial, repeatable, defined, managed, and optimized. This means that these processes, which lead to the development of software, will be manageable and the development team will deliver the finished product within the desired time frame and produce software in accordance with the specification. It is a periodic measurement tool to evaluate the ability of the organization's processes.

In their article Li et al. (2000) suggest that the TQM method refers to management methods used to enhance quality and productivity. It should be used to succeed in software development. This means that the processes involved in TQM are vital to the achievement of quality. It also suggests that Deming's TQM method is vital for organizations to provide quality satisfaction for customers. Glowalla and Sunyaev (2015) propose that agile software development methods should be used in IS development automation. The paper does not suggest any specific tool but rather proposes the evaluation of factors that have a strong impact on the failure or success of IS projects. These factors include the quality responsibility and quality policy that govern IS projects. Through several themes, the paper brings out these influential aspects that dictate project outcomes in the information domain. Process management efficacy, stakeholder participation, and the management infrastructure are the three main constructs into which the themes are categorized.

Kan et al. (1994) discuss TQM as a management philosophy along with its key elements: customer focus, process improvement, the human side of quality, data, measurement, and analysis. They then focus on software development, which should consider the elimination of errors to achieve quality. This is because of the coordination involved in the team. The process tends to check on the quality of the final product, since it considers aggressiveness in addressing errors in the processes and the product. The CMM is the tool used to implement TQM in IS projects. The paper also mentions different models that aim to improve quality during the development process, such as total quality control (TQC), the six sigma strategy, market-driven quality, plan–do–check–act, quality improvement paradigm (QIP)/experience factory (EF) organization, and lean enterprise management.

Jha (2006) research centres on the basis of the major study conducted in the area of TQM and business excellence (BE) in terms of software development. The distresses and concerns for TQM as well as ERP execution are discussed. Minor case research of the first company in Asia to win the desired Deming Prize regarding the incorporated Japanese model for BE is also discussed. The research attempts to provide a holistic standpoint of ERP execution as a measure of TQM or BE strategy application.

Bon and Verheijen (2006) state that the rapidity of growth in the IT management domain and TQM has provided public education programmes with limited chances to tackle the current demands of companies. This is because they are not capable of developing processes that cover well-rooted training in IT (software development) management matters, let alone providing or delivering quality personnel who have undergone training as part of their systematic education programme.

### A. Reasons for the choice of total quality management (TQM) tools or methods

Jha (2006) explains that technological and software innovations have softened geographical borders, resulting in more knowledgeable customers. A sustainable business environment is becoming increasingly multifaceted and the market base has transformed from indigenous to global. Accordingly, constant pressure is exerted on organizations to develop their competitiveness by reducing their operating cost and humanizing their logistics.

Bon and Verheijen (2006) insist on the precedent that TQM is not the sole issue that challenges software development companies. Almost every developer in virtually all enterprises encounters similar glitches. Administration in these establishments is often fairly unaccustomed to some of the central frameworks and systems (instruments).

In their article Subramanian et al. (2007) explain that the reason for the use of the capability maturity model (CMM) in software development is based on the need to improve the quality of the final software product. This means that, to achieve quality, a stringent measure that emphasizes the quality of the product through management commitment, process monitoring, team behaviour, and the involvement of the customer is important. In this way it is important to ensure that the model takes into consideration all the best qualities of software development.

Li et al.'s (2000) article highlights the importance of management and why it should be ready to train employees on a continuous cycle. This is because quality keeps improving and the best practice is to ensure that the employees have the knowledge to incorporate any change and work towards achieving better integration of the results. Training is vital to ensure that new processes work for the improvement of the final product.

Glowalla and Sunyaev (2015) mention the importance of considering all the factors that contribute to both the success and the failure of an IS project and state that the process highlights the importance of all the stakeholders being fully committed to the process to deal with the sense of quality software development. Quality is enhanced in the model by process tests and working on the results to deliver quality. The results are assessed after the system has been tested, and this means that the quality of the product depends on the continuous testing of the system. The commitment of the top management in any firm carrying out IS projects is imperative in determining the success of such projects.

According to Kan et al. (1994), one key aspect of IS projects is that quality simply implies conformance to the anticipations or specifications of the customer. Training is important to the employees or the team. This is based on the processes involved in TQM because of the complexity of the system and the need to incorporate the changes brought about by customer preferences and the logistics of the processes. The customer is involved in the project since he or she is the determinant of the quality of the product. Therefore, he or she must be available to provide the necessary information regarding his or her satisfaction, since the main goal of the project is to satisfy the customer. This means that the satisfaction of the customer is the measure of quality.

Communication is one of the key principles involved in the use of TQM in software development management. It is the main driver of the project, and it requires all the departments to have an open link to it. This means that there is no procedure for communication when involving TQM in development management, because communication has to reach the associated party in the required time and the response must be received instantly. The project depends entirely on communication; therefore, there is need for a structure in which communication between the management, the development team, and the customer is not limited.

The method used in the paper is the TQM method, which includes Deming's TQM method and the quality seven (Q7) and management seven (M7) tools. TQM is a management tactic that was patented in the 1950s. It became progressively more prevalent from the early 1980s. As such, total quality is an explanation of the culture, environment, and institution of a company that endeavours to offer its customers quality aspects that gratify their needs.

B. Alternative tools and solutions to help organizations to achieve their business goals

Just like other engineering projects, a managed approach is likely to increase the quality of software products, which require the introduction of manufacturing-based ideas into the software development process. The TQM philosophy, as suggested by various authors, can be imperative in improving the quality of any given software product. Developing software through total quality management procedures and objectives is ideal for meeting the quality standards of the client due to the aggressive involvement in achieving quality. TQM entails the following of procedures and enables the development team to meet its goals. The procedures involved include flexibility of the procedures as well as the flow of information (Rothenberger, Kao, & Van Wassenhove, 2010). Rothenberger et al. (2010) further observe that different environments bring about or support differing degrees of quality in the software development process. In some environments quality is achieved through developing error-free code or code that has minimal errors. In other environments, however, the same is achieved through improving the code-testing process. Therefore, improving the software development process ultimately improves the quality of the final project.

Lee and Chang (2005) assert that traditional software development processes place considerable stress on testing techniques but reveal great weaknesses in the planning process. Now organizations depend more on methods such as the ISO 9001 management system for quality, the CMM, and the TQM quality management approaches, which can be applied effectively in the process of quality improvement. Knowledge management technology can transform these quality documents into knowledge that exists in a readily actionable format, which can then be used effectively throughout the software development process.

A better choice is to use the CMM, as it has become more widely institutionalized in software development. Agrawal and Chari (2007) highlight the use of the CMM as a popular methodology for improving the quality of software through improving the processes used in the development cycle. CMM projects that are at level 5 are greatly influenced by highly mature processes that have a significant effect on quality, effort, and cycle time. The most important attribute of this procedure is ensuring that the project guarantees quality. Quality is determined by time, and the resources are determined by the availability of information. Therefore, time has to be used to seek information that will enable the exploitation of resources to achieve the quality final product. This means that there should be a free flow of communication in the department. Therefore, the best method to be used in this category should be defined as an aggressive flow of information about customer preferences to the development team. The

development team includes the project management and the team members. Obtaining a free flow of information in relation to the customer needs should be encouraged. The management and the team have to share information about the project in an unlimited way. This means that the customer preferences, information flow, and improvement will be communicated at the same time and therefore the project will be able to accomplish the intended objectives. The importance of using the information model in this project is to ensure that all these previous requirements, problems, and even accomplishments are known. This means that the team will use the project assessment information to work on meeting the customer preferences. Furthermore, the management will be notified about any changes and therefore the organization will work to its maximum or expand positively.

Finally, the project time and the goals of the customer are the main aspects of every process that the organization undertakes. Time, software complexity, and the involvement of ever-increasing or larger teams lead to communication problems. The ultimate objective of TQM is to achieve systems that have no defects at all. This is achieved by continuously improving a system and removing every last defect, because it becomes problematic to increase a system's features while the already existent ones are faulty. It is also more critical to use this approach than learning to live with the faults in a system while trying to achieve software of the highest quality.

## VI. CONCLUSION AND DISCUSSION

There is more to software development than just meeting the time target and delivery of the software. Quality is the main attribute of software development, since it determines the satisfaction of the customer. This is the reason for the need to involve total quality management in the process to guarantee the approval of the customer. Quality is determined by the customer and the use of other methods incorporates the customer involvement in consultation and the approval of the software. Consequently, the ease of communication in any project determines its success. When there are no boundaries in communication among all the stakeholders working on a project, they are all able to collaborate more effectively and hence realize a high-quality product in the end.

As a TQM rule, the best method to be used in this category should be defined as an aggressive flow of information about customer preferences to the development team. This comes from the view that traditional software development processes place considerable stress on testing techniques but reveal great weaknesses in the planning process. Since different environments bring about or support differing degrees of quality in the software development process, quality is only achieved through the development of error-free code or code that has minimal errors. It can be concluded that in TQM quality is determined by time while the resources are determined by the availability of information.

## REFERENCES

[1] Agrawal, M., & Chari, K. (2007). Software effort, quality, and cycle time: A study of CMM level 5 projects. IEEE Transactions on Software Engineering, 33(3), 145–156.
[2] Bon, J. V., & Verheijen, T. (2006). Frameworks for IT management: An introduction. Zaltbommel: Van Haren Pub.
[3] Donaldson, S. E., Siegel, S. G., & Donaldson, S. E. (2001). Successful software development. Upper Saddle River, NJ: Prentice Hall PTR.
[4] Glowalla, P. & Sunyaev, A. (2015). Influential factors on IS project quality: A total quality management perspective. Retrieved from http://www.isq.uni-koeln.de/sites/wi_isq/Publikationen/ICIS_Project_Quality_final.pdf (accessed 19 February 2016)
[5] Kan, S., Basili, V., & Shapiro, L. (1994). Software quality: An overview from the perspective of total quality management. Retrieved from https://www.cs.umd.edu/~basili/publications/journals/J52.pdf (accessed 19 February 2016).
[6] Jha, V. S. (2006). Strategic planning, technology & innovation for business excellence – A conceptual research study. International Journal of Business Research, AIBE Publication, VI(2), 84–90.
[7] Lee, M.-C., & Chang, T. (2005). Applying TQM, CMM and ISO 9001 in knowledge management for software development process improvement. International Journal of Services and Standards, 2(1), 104-105.
[8] Li, E. Y., Chen, H., & Cheung, W. (2000). Total quality management in software development process. Retrieved from http://www.cob.calpoly.edu/~eli/pdf/jqai-00.pdf (accessed 19 February 2016).
[9] Mcmanus, J., & Wood- Harper, T. (2007). Software engineering: A quality management perspective. The TQM Magazine, 19(4), 315–327. doi:10.1108/09544780710756223
[10] Rothenberger, M. A., Kao, Y.-C., & Van Wassenhove, L. N. (2010). Total quality in software development: An empirical study of quality drivers and benefits in Indian software projects. Information & Management, 372–379.
[11] Subramanian, G. H., Jiang, J. J., & Klein, G. (2007). Software quality and IS project performance improvements from software development process maturity and IS implementation strategies. Journal of Systems and Software, 80(4), 616–627.
[12] Vliet, H. V. (2010). Knowledge sharing in software development. Paper presented at the 2010 10th International Conference on Quality Software. doi:10.1109/qsic.2010.78