An Empirical Study of SQA Function Effectiveness in CMMI Certified Companies Implementation, Problems, and Recommendations

Zille Subhan and Ali Saeed Khan

Abstract—**The most vital component for any software development process is, "quality", as it ensures the reliability and effectiveness of new software. Software Quality Assurance (SQA) techniques as well as a standardized qualitative metric known as Capability Maturity Model Integration (CMMI) are used to ensure this quality. The purposes of both the practices are same as both make efforts for end product's quality. In spite of this, CMMI certified organizations have SQA function, but face a lot of issues, which resulted in lowering the quality of the products. Standards usually provide documentation, but SQA consider testing as a chief element and also documentation only for authentication and appraisals. The relationship of the SQA function with CMMI has not attended much in common literatures. This paper is centered on investigation conducted through data collection from diverse CMMI certified software development firm to check the practice of SQA function.**

Keywords— **Software Development Process, SQA, CMMI and Software Quality Frameworks**

I. INTRODUCTION

The purpose of this project report is to present a synopsis of contemporary literature and issued papers on the subject matter of practices regarding SQA/CMMI and general management. With regard to the purpose, it is to outline the significance of employing these two practices, i.e. CMMI and SQA function, together and to end with getting suggestions and paramount practices for SQA function and CMMI collective application in software development firms. It is based on studying the contemporary best practices and their application circumstances. For the purpose of study, we arranged some interviews with project managers and their teams, along with the filled questionnaire method. We examined projects managers and team members' perspectives about the two functions through questions. Grounded on their perspectives over the subject, we developed some hypotheses to test. The main purpose, here is to demonstrate the inability of firms to make distinction between CMMI and SQA, which impedes the capability to get most out of these techniques.

This project report delivers significant advices/suggestions, which can be applied by these firms to augment the quality of software development practice, software product and in managing the customers' requirements.

II. BACKGROUND AND MOTIVATION

In this survey data is collected from 50 respondents, Table 1 outlines statistics of the data collection. The purpose of survey was to get the prevalent trends of quality measures in software development firms. The information synthesized from gathered data explicitly demonstrates that 60% of participants disregard any distinction between SQA and CMMI. Whereas participants who claim that software quality assertion denotes to the testing were 75% and who recommended the combination of these were 50%. Following table represent the questionnaire result.

TABLE I
DATA COLLECTION SOURCES

Total Participants	SQA as CMMI	SQA as Testing	Combined Activity
50	30	38	25

The facts above mentioned evidently reveals that software development firms do not identify any divergence in the CMMI and SQA function. They misconceived about the basic differences among these two disparate functions. Such facts permit us to demonstrate the disparities among the two practices CMMI and SQA, which collectively intended at certifying software quality. Software development firms should employ these practices altogether to enhance their products' quality.

Zille Subhan, National University of Computer and Emerging Science, Lahore, Pakistan, (Email: zsubhan@hotmail.com)
Ali Saeed Khan is working as a senior software engineer with Web-tronix, Islamabad, Pakistan, (Phone: +92-3348481514; Email: alisaeed101@gmail.com)

III. RESEARCH METHODOLOGY

A. Selection of a Research Methodology

In this project report, aimed at "analysis of SQA function effectiveness in CMMI certified software development firms" we used qualitative methods to gather data and then employed manual enquiry to recognize the relationship in data.

B. Data Collection

The data required for this project was collected through visiting the chosen software development firms and conversed with managers of CMMI certified software development firms with the help of interviews, for comprehending their SQA and CMMI strategies/functions. Through interviews and filled questionnaire, we collected facts and information regarding the outcomes of applying SQA practices, such as major and minor NCs (discussed later in the paper), and defects.

C. Time frame for Data Collection

We visited 3-4 firms for data collection, and among these 10-15 projects were studied. The project related data is also collected, which is quite recent in time, mostly in previous 2 years.

D. Source of Data

The data sources used were of different types, such as management, team members, staff members, and previously completed project reports regarding some particular software development firms. Nonetheless, the primary data for this project was collected from management of software development firms. We enquired some questions and considered the elementary requirements concerning software quality matters and difficulties faced throughout the entire lifespan.

Another data source to collect information was staff members of those specific software development firms. For data collection we interviewed the staff with respect to get the insights about matters in software quality assurance procedure and evaluated the chief matters they usually encounter during the quality software development process.

Then the previously competed project reports were used to collect secondary data. This secondary source delivers us a detailed impression of deficiencies in software development and management of quality assurance as well as CMMI levels management.

IV. LITERATURE STUDY

In this modern era and advanced level of computing systems the software are getting complex. Such complexity has resulted in increasing concerns and worries with regard to the security, reliability and consistency of such systems. With time, the systems are now more developed in terms of size and complications, which resulted in more inadequacies and in turn, led our lives to be threatened of the safety, reliability and consistency issues. In usual terms, the software is alienated into two rudiments: which are considered to be internal and external quality aspects. The aspects which directly interact with the users of software, are known to be external, and on the other hand, the internal quality aspects are regarded those which have no interactions with users. However, the quality should be the by-product and be free to get, as it means to conform to the product prerequisites [4]. So for this, the main function of software quality assurance is to guarantee that the standards, procedures, and measures are right for the project in addition to the correct execution. In reality, SQA and SQC are practices described within CMMI, and are the sub-processes of support procedure subject. Nonetheless, in CMMI SQA/SQC is described as practice and process of product quality assurance [5].

In the last decade, there is a rising concern with respect to the quality of software process in software development sector, which can be demonstrated by mounting numbers of estimated software process improvement (SPI) frameworks and principles. While, these standards and frameworks are rarely implemented in the actual execution of project in software industry, and most of the times the implementation is failed. Subsequently, with such dynamic environment and increasing rivalry in software industry, there is a requirement of more advanced, high quality and multifaceted systems. Nonetheless, the concerns for quality have always been tremendous [6], [7].

Furthermore, the quality assurance is a perplexing matter. A number of methods for quality assurance are there, but it is a puzzle that which one would be suitable to apply in any circumstances. Additionally, the matter becomes more heightened due to the discrepancies in the understanding of the term "quality". However, the quality assurance practice is usually denoted as the practice of investigation and authentication of software. Undoubtedly, these are significant undertakings of the quality assurance practice, but in a narrower sense, the quality assurance program is much wider than these. Furthermore, quality assurance program is intended at scheming a system with quality in the accomplished manner delivered by the software development team. The question arises here is, what would be the best way to do so? [8], [9].

In reality, software quality cannot be well-defined due to the fact that its vocabulary lacks an absolute and well demarcated definition for this. Though, research demonstrates two approaches to achieve quality software systems. So, at this point, the first technique is proactive or process based approach while the other is reactive or testing based approach. Moreover, in recent times, there are some breakthrough models and frameworks developed for the purpose of deciding the procedures and to assess the maturity or competence level of a software development firm. Among these eminent models, there comprises CMM, CMMI and ISO15504, and also these are regarded as software process evaluation or appraisal models [6], [7].

However, the Capability Maturity Model Integration (CMMI) is far extensively implemented model for process upgrading and maturity/capability level verification (Yucalar & Erdogan, 2009). Besides this, Software Engineering

Institute (SEI) delivered a report recently which stated that there was an upsurge in CMMI assessment from software enterprises from worldwide. Subsequently, it is more interesting to find out the factors for process enhancement, which led the software development firms towards success by effectively planning the SPI employment approaches [10]. In history, the CMMI model was generated and inaugurated in 2001, by Carnegie Mellon University, in partnership with U.S. Department of Defense. Although the CMMI model has been the reason for success of many companies, however the differences among models has been posing many problems with regard to the compliance and differing guidelines in attending the possible developments [6].

Despite the fact that CMMI has gradually made the performance and success in quality assurance and as a result many companies have employed CMMI, few researches have presented practical insights into the effectiveness of CMMI in terms of viability. Indeed, most of previous studies are based on the association between CMMI and organization's performance and therefore they ignored to address the usefulness of CMMI model itself. Moreover, the lack of proper directions for successful execution of the CMMI has been a crucial factor to be considered by managers [11].

Within these arguments, (Gefen et al., 2006) deliberate numerous explanations of deterrence of CMMI model. On the word of (Gefen et al., 2006), the most prevalent blame over it is that the advanced levels of capability requires extravagant documentation, which place more burdens on the implementation struggle. While ensuing the directions of regulations and formalism of CMMI, software development teams have to be less self-reliant, which resulted in lower enthusiasm and innovativeness [12].

V. HYPOTHESES

This project is based on verification of two hypotheses, which are resultant from the information gathered.

H_1: Implementing the SQA function (testing) solely results in flaws during the later phases.

The first hypothesis claims that if the firms execute the SQA function in isolation, which is testing, then it resulted in more defects at the later times. It is comprised over the single function of testing, thus the defects are discovered at the coding phase, and at this phase, correcting the spotted defects can be more troublesome. As a result, first one has to repeat the overall process of software development, which resulted in wastage of time.

H_2: Implementing the SQA and CMMI functions collectively results in lesser deficiencies and so quality upgrades considerably.

The second hypotheses contended that if firms execute both the functions (SQA and CMMI) altogether, then it has to face lesser amount of flaws. Moreover, if combination of both the functions has been applied then shortcomings are noticeable on the former stages, rather than later. So, it is quite easier to correct the flaws at earlier times. Moreover, the detection of defects at earlier phases, the two important resources can be saved; cost and time, which is an added benefit.

VI. HYPOTHESES ANALYSIS

A. *H_1: Implementing solely the SQA function (testing) results in flaws in later phases*

Table 2 demonstrates the data of those projects where only SQA function was employed. It is gathered from different firms with help of interviews. The project reports and documents are also used to verify the hypotheses.

The data specified in Table 2 is from small, moderate and huge size projects. We measured struggle in man hours (where 1 function point =100 man hours). In latest factor, defects are stated counter to each project. It is evidently demonstrated that with application of only SQA function (testing), we get a huge number of defects.

TABLE I
DATA FOR TESTING

Company	*Year*	*Projects*	*Effort (Man Hours)*	*FP Count*	*Duration (Months)*	*Defects*
Company A	2011	Project 1	3200	32	7	275
Company A	2011	Project 2	4500	45	6	320
Company A	2011	Project 3	2560	26	5	170
Company A	2011	Project 4	1800	18	3	65
Company B	2009	Project 5	1900	19	4	90
Company B	2009	Project 6	1600	16	5	74

Figure 1 shows association between function points and defects. In situation of employing only SQA function we can find 6 defects per FP.

.

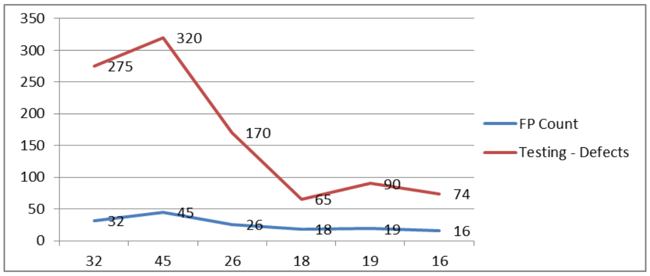

Fig. 1. Ratio of defects

Application of SQA function in isolation mode shows those defects are detectable at later stages of coding and integration. More importantly, the detection of defects at coding phase could cause higher costs and delayed implementation, as there would be need of starting from the initial step to make corrections. Though, in this phase, the revision would result in fewer defects in integration phase. The evaluation and audit practice should be made earlier in order to reduce the defects at later phases. This scenario can be observed in Figure 2 where very few defects found at earlier phases, but numerous at later ones. As we found a number of defects are detected at integration phase. Detection of such defects at the later stages poses a bigger threat, as to correct defects a revision needs to be started from first phase.

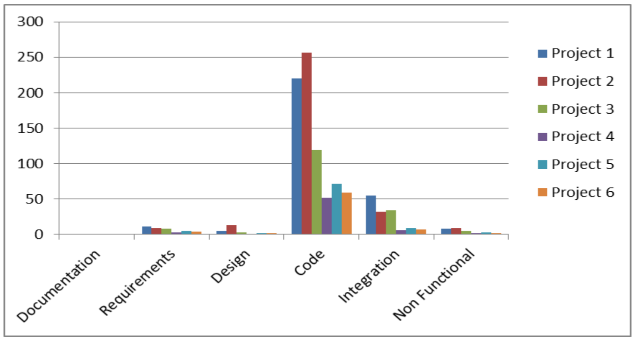

Fig. 2. Defect Categories

In Table 3 we can see different classes of defects with their ratio obtained while implementing only SQA function. Another important issue we see is that we are unable to find any defect in the documentation. However, we obtain some shortcomings at requirements and design phases, it does not imply that all the defects are discovered with regard to these phases.

TABLE III
DEFECT CATEGORIES

Project	*Documentation*	*Requirements*	*Design*	*Code*	*Integration*	*Non Functional*
1	0	11	6	220	55	8
2	0	10	13	256	32	10
3	0	9	3	119	34	5
4	0	3	1	52	7	2
5	0	5	2	72	9	3
6	0	4	1	59	7	2

B. H_2: *Implementing the SQA and CMMI functions collectively results in lesser deficiencies and so quality upgrades considerably.*

The second hypothesis contended that if companies execute both the functions (SQA and CMMI) together, then lesser number of defects is detected, where the product quality is enhanced. For this hypothesis, we have accompanied the same method as for H1. Data is collected for those projects which employed the combination of practices.

In the Table 4, the data for projects employing both practices is shown, where the NC is a measure used to evaluate the CMMI. When both practices are applied the defects are reduced about 33.33 %, which means 4 defects per FP.

TABLE IV
SECOND HYPOTHESIS DATA

Company	*Year*	*Project*	*Line of Code*	*Fault Points*	*Duration (Months)*	*NC (Major / Minor)*	*Defects*
A	2009	1	5000	50	8	16	117
A	2009	2	4700	47	7	6	112
A	2010	3	7000	70	10	12	398
A	2010	4	4000	40	6	5	110
A	2011	5	1600	16	5	4	170
B	2009	6	6000	60	10	6	240
B	2009	7	5000	50	6	3	250
B	2010	8	2500	25	4	3	117
B	2011	9	3700	37	4	8	92
C	2010	10	1600	16	5	5	73

The other benefit of employing both the practices together can be seen that the defects are realized at earlier phases of requirement and documentation, which can be corrected at earlier phases with lower cost and time consumption. Meanwhile the earlier realization of shortcomings can escape us from the revision of overall lifecycle, as demonstrated by Figure 3.

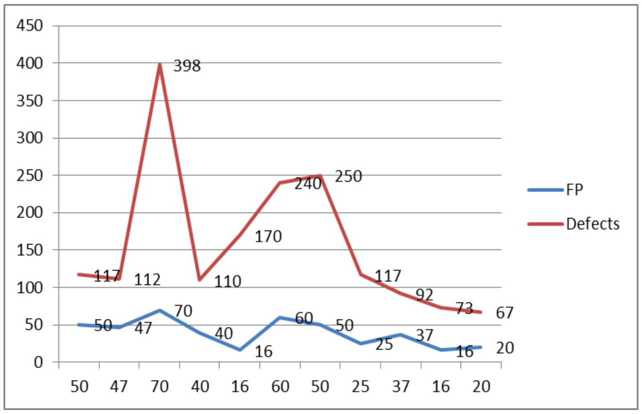

Fig. 3. FP vs Defects in CMMI and SQA

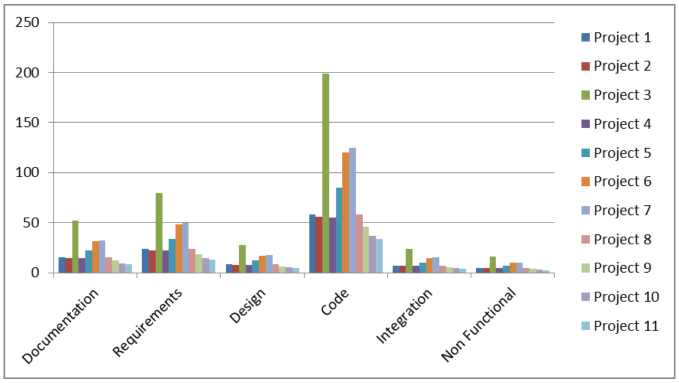

Fig. 4. Defect Categories

VII. RESULTS

In many CMMI certified companies, there is employment of SQA function but many defects are realized. In such companies, a number of audit and appraisal functions are performed along with the presence of SQA department, but huge number of shortcomings came into view resulting in lower quality of software products. The results show that SQA function is not being used by aligning it with CMMI function and such employment does not result in quality products.

VIII. RECOMMENDATIONS

It is evident that software development firms' purpose of employing CMMI certification is not as it is thought to be (quality assurance), rather they used it for marketing. They are only concerned with the promotion of their firms as to be called as CMMI certified firms, but the real practice of CMMI function is lacking in such firms. So, the marketing has to be considered at the least point, there must be focus on improving the quality. Quality assurance and delighted customers with good quality products lead towards the marketing of software itself.

The main purpose of SQA and CMMI is to prevent from defects. But firms use it for the purpose of detection and rejection, which resulted in higher costs and delayed projects. So with such application of SQA and CMMI, the quality never upgraded and defect ratio doesn't get curtailed.

In most of the firms, SQA and CMMI are set out as different departments, where the CMMI function is to document and report making, while SQA is employed for testing. As both practices are working towards same goal, so they must be integrated with each other. From broader point of view, CMMI is another sub-function of SQA. Though, it will cost higher to train employees and implement both the practices in combination, but it will reward for the long term. Following expenses have to be incurred:

- For CMMI application, most experienced resources are essential
- A particular kind of training and induction programs are needed
- The process documents are required to be up dated.

It is evident that the presence of both departments in firms does not ensure the quality enhancement, so firms are realizing the formal certification as to be less important. Nevertheless, it is just another misconception. Quality is the most competitive tool in this environment, which can be achieved by SQA and CMMI both. The only issue is ineffective implementation of these practices.

IX. CONCLUSION

It is concluded that SQA and CMMI are both such practices, which are aimed at similar goal of quality assurance. CMMI certified firms do not recognize the objectives of both the functions. CMMI is regarded to have compliance with documents' principles while SQA is thought to be involved only in testing. Or it can be said that both practices function distinctively in same firm, which leads towards realization of numerous defects affecting the product quality in negative manner. To verify this hypothesis, there is the data collected from different CMMI certified software development firms along with a SQA department as well, but still facing issues in product's quality. It is recommended that for better quality products, software development firms should develop such a setting where they can employ a combination of SQA and CMMI practices. By employing these practices in combination, the defects will be reduced and quality will be improved.

REFERENCES

[1] BlurtIt. (2011) What Is Research Methodology? http://www.blurtit.com/q462869.html

[2] Mari Murtonen, Learning of Quantitative Research Methods: University Students' Views, Motivation and Difficulties in Learning. Turku, Finland, 2005.

[3] Wilhelmina C. Savenye and Rhonda S. Robinson. (2000) Qualitative Research Issues and Methods: An Introduction for Educational Technologists. http://www.aect.org/edtech/39.pdf

[4] M Sangeetha, K.M.Senthil Kumar, C.Arumugam, and K. Akila, "Improved Software Quality Assurance Techniques Using Safe Growth Model," (IJCSE) International Journal on Computer Science and Engineering, vol. 2, no. 6, pp. 2125-2132, 2010.

[5] SQA. (2011) SQA in Theory. http://www.sqa.net/cmmi.html

[6] Faith Yucalar and Senol Zafer Erdogan, "A Questionnaire Based Method for CMMI Level 2 Maturity Assessment," Journal of Aeronautics and Space Technologies, vol. 4, no. 2, pp. 39-46, 2009.

[7] Yas A. Alsultanny and Ahmed M. Wohaishi, "Essential Characteristics of Software Model that Provide the Software Quality Assurance," International Review on Computers and Software, vol. 4, no. 5, pp. 577-585, 2009.

[8] Dan C. Boger and Norman R. Lyons, "The organization of the software quality assurance process," ACM SIGMIS Database, vol. 16, no. 2, pp. 11-15, 1984.

[9] Yi Liu and Taghi Khoshgoftaar, "Reducing over fitting in genetic programming models for software quality classification," in HASE'04 Proceedings of the Eighth IEEE international conference on High assurance systems engineering, 2004, pp. 56-65.

[10] Nuntarat Bunditwongrat, Mathupayas Thongmak, and Atcharawan Ngarmyarn, "An Empirical Framework of key Success Factors for Software Process Improvement," in Proceedings of the European Conference on Information Management & Evaluation, 2011, pp. 82-90.

[11] Shari S.C. Shanga and Shu-Fang Lin, "Understanding the effectiveness of Capability Maturity Model Integration by examining the knowledge management of software development processes," Journal of Total Quality Management, vol. 20, no. 5, pp. 509-521, 2009.

[12] David Gefen, Moshe Zviran, and Natalie Elman, "What can be Learned from CMMI Failures:," Communications of AIS, vol. 2006, no. 17, pp. 2-28, 2006