

Research Article

Software Quality Assurance in Software Projects: A Study of Pakistan

¹Faisal Shafique Butt, ¹Sundus Shaukat, ¹M. Wasif Nisar, ¹Ehsan Ullah Munir,

²Muhammad Waseem and ¹Kashif Ayyub

¹Department of Computer Sciences,

²Department of Mathematics, COMSATS Institute of Information Technology Wah Cantt.,
47040, Pakistan

Abstract: Software quality is specific property which tells what kind of standard software should have. In a software project, quality is the key factor of success and decline of software related organization. Many researches have been done regarding software quality. Software related organization follows standards introduced by Capability Maturity Model Integration (CMMI) to achieve good quality software. Quality is divided into three main layers which are Software Quality Assurance (SQA), Software Quality Plan (SQP) and Software Quality Control (SQC). So In this study, we are discussing the quality standards and principles of software projects in Pakistan software Industry and how these implemented quality standards are measured and managed. In this study, we will see how many software firms are following the rules of CMMI to create software. How many are reaching international standards and how many firms are measuring the quality of their projects. The results show some of the companies are using software quality assurance techniques in Pakistan.

Keywords: Pakistan, performance, Software Quality Assurance (SQA)

INTRODUCTION

Software quality is specific property which tells what kind of standard software should have. Quality is very important for the success of software. Quality makes the small difference in successful and failed software projects. There is no device in existence to measure the quality of software project. But there are standard set by ISO 9001/9000-3, CMMI and many others to acquire good quality software. To create effective quality software, there are eight steps of software quality management which are as follows (Humphrey, 2008):

- Establishment of quality strategies, achievements and policies.
- Appropriately guide, instruct and encourage the software developing team.
- Launch and sustain requirements of quality management process.
- Create and preserve numerical control of the software engineering process.
- Assist, examine and estimate all the software products.
- Search all errors for correction and categorize, repair and avoid similar mistakes.
- Establish and preserve software configuration management and adjust the control system.

- Continuously advance the developing process.

Related work: Quality of software is the reason of success and decline of a software related organization. Many researches have been done regarding quality management, measurement and assurance in different countries.

Florac (1992) discussed software quality measurement and the related problems and defects. The defects were used in understanding and predicting the software quality product and software efficiency. In this study, the problems and defects were discussed in detailed.

Devnani-Chulani (1998) is discussing cost, schedule and quality with the help of two sub-models of quality model. This study discussed initial stages of the model which tells great deal about the foundation of numerical problems related to the model parameters. And different models are discussed related to the measurement parameter. This study also discussed the research on defect removal model and further research on this model.

Jacquet and Abran (1998) explained that the expression metrics validation is used in many different meanings. They discussed different methods are analyzed and compared. This analysis was based on the process model for measurement method to the exploitation of the measurement result.

Corresponding Author: Faisal Shafique Butt, Department of Computer Sciences, COMSATS Institute of Information Technology, G,T Road Wah Cantt., 47040, Pakistan, Tel.: +92-321-5029899

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

In 2003, Eldon Y. Li, Houn-Gee Chen and Waiman Cheung discussed and explained the principles of total quality management and its implementation. It briefly explained the similarities in software development process and product development process (Li *et al.*, 2000).

Ifikhar and Ali (2011) deliberated the role of quality measurement in software industries of Pakistan. It was discussed how quality assurance is measured in different industries and how they are compared. In this study, a survey has been conducted to differentiate more experience firms from less experience firms on the basis of quality assurance.

Eldrandaly (2008) discussed different quality models and how industries should choose these models according to their software. He introduced a prototype knowledge based advisory system for checking SQA of software industries. In this prototype, different steps were introduced through which industries or their members can choose quality model according to their requirements.

Stavrinoudis and Xenos (2008) discussed how to measure the internal and external software quality. Quality metrics and surveys were conducted to measure customers' software quality reviews.

Mekprasertvit (2004) discussed how software quality assurance plan are made. In this study, different techniques and steps were discussed. How documentation is done? What will be done in different phases? Who will be the organizer and who will the developer? What tasks are to be done? Which tools are used?

Farooq *et al.* (2011) explained the importance of software quantitative and qualitative metrics. In this study, software characteristics were explained and discussed how they are tested using software metrics. This study emphasized on software testing process.

Jørgensen (1999) discussed the importance of software quality measurement in software quality. There are different theories about software quality should be measured. There are some common quality definitions and some measurement theories are discussed. Some quality factors are also discussed.

Boegh *et al.* (1999) discussed internal and external quality characteristics and the link between them. Squid data model was discussed. Some specification relating software quality was also discussed.

Kececi and Modarres (1998) discussed the different problem arising in the software quality through a life cycle model. In this model, relationships between different phases of software were mentioned.

Bawane and Srikrishna (2010) discussed the quantitative quality model needed for the software development. The goal was achieved through using various quality measurement and metrics. This study also discussed about the quality requirement and its three main attributes which are: portability, reusability and maintainability.

In 2008, Asad Imam, Shakeel A. Khoja and Imranuulla Shariff explained that it is difficult to

improve industrial software quality standards only by relying on few approved models. In this study, they introduced a form or a template by using quality standards like CMMI etc to build an effective working system (Imam *et al.*, 2007).

Software quality parameters: Software quality has two types which are (Sacha, 2005).

- **Software functional quality:** It is a property which tells the functional requirements of software and how those requirements are satisfied.
- **Software non-functional quality:** It is property which tells how structural requirements are met and how they support the functional requirements. To measure and manage quality of software project, some parameters are laid down which are:
 - **Testability:** How software can be tested and what are limitations of testing software.
 - **Usability:** It tells that the software is user friendly.
 - **Understandability:** How easy it is to understand software and its functions by layman.
 - **Consistency:** It tells that how far software and functions are running uniformly.
 - **Efficiency:** How much work software can perform within specific time using minimum resources?
 - **Effectiveness:** To which amount software is satisfying user and meeting its requirements.
 - **Accuracy:** How precisely the software working gives correct results?
 - **Maintainability:** How software can be maintained that is fixing bugs and adding new feature etc.
 - **Reliability:** When software is used under different conditions, how much software is reliable?
 - **Portability:** Software can be easily elated from one environment to another.
 - **Security:** How much software is secured in terms of password data and authorization access?

MODEL

We conducted a survey from questionnaires and interviews. Model can be defined as the main structure on which the research is based. Model can include different areas of research and proposed hypothesis.

Proposed model: This proposed model is introduced to search on the basis of software quality assurance in Pakistan. This model can be named as SQA-PAK. The model of this survey is composed of different aspects of quality assurance engineering which are explained below in detail.

Quality Assurance Engineering is defined as the task done by engineers in a systematic way to test

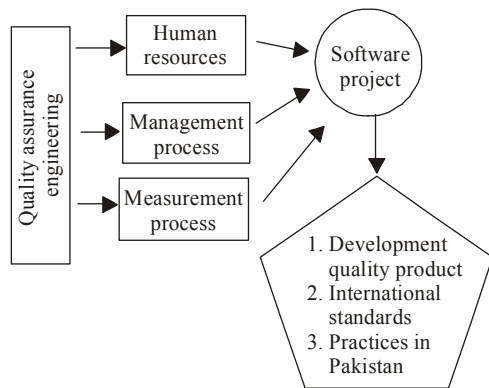


Fig. 1: SQA-PAK

quality. The quality assurance engineers' job is to make sure that associated programs in that software are working properly. The quality of should be excellent that it can locate errors before they occur. Quality assurance engineers are the link between users and programmers, so it is their job is to know the problems faced by users, those problems should be removed in the testing process. It has many topics related to this field. According to the Fig. 1, the main topics related this model is discussed below.

- Human resources:** In the common language, human resources mean the workers which establish an organization. Management of human resources is known as Human Resources Management (HRM) or HR (Offen and Jeffery, 1997). HRM is a department in an organization which focuses on recruiting workers and assigning their tasks. HR manages the workers or officers of an organization and makes sure that they are satisfied working. For an organization, their workers are an asset, they do not want to lose their assets, for that different activity like training and promotion, bonus etc are arranged by HR. When a person is satisfied with his/her job then he will be able to give good effort to that organization, only then an organization can acquire a good quality product. In this model, the role of HR is observed in Pakistan.
- Management process:** Management is a process of planning and organizing the activities. In an organization, manager mostly manages the activities and project. In managing the software quality, there are three layers which should be kept in mind. These layers are SQA, SQP and SQC. Software quality management is used to ensure that specified level of quality is achieved. In this model, the quality management model is being observed in the firms in Pakistan.
- Measurement process:** Measurement is a process of measuring something. Software measurement is talented field of software engineering. It may

include planning, controlling and development process (Briand *et al.*, 1996). For measuring the quality, metrics are used. Those metrics can be estimation of code of software or the risk analysis. Mainly used metrics are of two types which are software metrics and performance metrics. Software metrics are used to measure a part of software or its estimations. Performance metrics are used to measure the performance of the organization. Software metrics is one of the basic practices of software engineering. Nowadays many customers are specifying the demand of software or quality metric in their requirement (Linda Westfall and The Westfall Team, 2005). There are some metric for quality analysis.

For example: (Pressman and Ince, 1992) Metrics are used to access the quality of the analysis model. Metrics given below are used to determine number of quality requirements:

$$N_r = N_f + N_{nf}$$

where,

N_r = Total No of requirements

N_f = Total Functional requirements

N_{nf} = Total Non functional

To determine the lack of ambiguity of requirements, we have:

$$Q1 = N_{ui}/N_r$$

where,

N_{ui} = Total no. of requirements which are agreed upon

N_r = Total No of requirements

The closer the value to 1, the lower the ambiguity of the specification.

All these quality principles are implemented into creating a successful software project as shown in Fig. 1. From this model, we can conclude the following hypothesis questions:

- To find what type of practices are followed in Pakistan regarding software quality?
- To what extent software quality needs are fulfilled?
- Are our software houses familiar with software standards of CMMI?

DISCUSSION AND RESULT

Our survey is conducted through interviews and questionnaires. Survey questionnaires consist of eight categories. The data acquired through these questionnaires is analyzed by statistical tools which are given below:

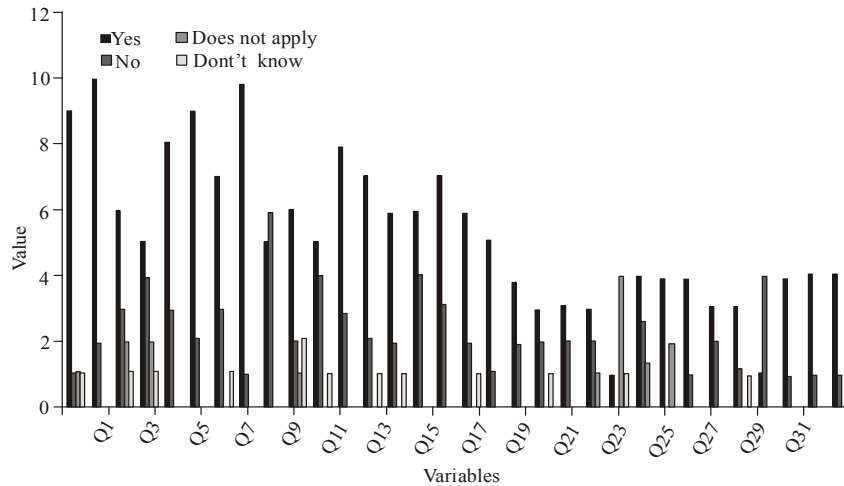


Fig. 2: Bar graph

Table 1: Frequency table of requirement management

Requirement management				
	Frequency	%	Valid%	Cumulative %
Yes	6	50.0	50.0	50.0
No	3	25.0	25.0	75.0
Doesn't apply	2	16.7	16.7	91.7
Don't know	1	8.3	8.3	100.0
Total	12	100.0	100.0	

Table 2: Mean and standard deviation of requirement management

Requirement management				
	Q1: Requirement collection	Q2: Necessary adjustments	Q3: Written policy	Q4: Measurements
Mean	1.50	1.17	1.83	1.92
Standard deviation	1.000	0.389	1.030	0.996
Variance	1.000	0.152	1.061	0.992

Table 3: Frequency table of software project planning

Software project planning				
	Frequency	%	Valid %	Cumulative %
Yes	8	66.7	72.7	72.7
No	3	25.0	27.3	100.0
Total	11	91.7	100.0	

Table 4: Mean and standard deviation of software project planning

Software project planning				
	Q5: Sufficient resources	Q6: Review of project manager	Q7: Estimates	Q8: Activities
Mean	1.27	1.18	1.55	1.09
Standard deviation	0.467	0.405	0.934	0.302
Variance	0.218	0.164	0.873	0.091

Table 5: Frequency table of software project tracking and oversight

Software project tracking and oversight				
	Frequency	%	Valid%	Cumulative%
Yes	5	41.7	45.5	45.5
No	6	50.0	54.5	100.0
Total	11	91.7	100.0	

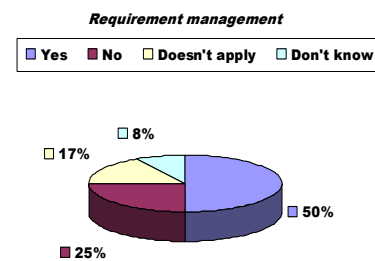


Fig. 3: Pie diagram of requirement management

Bar graph provides visual presentation of data (Fig. 2). Bar graph shown above presents data that is gathered from the survey. Horizontal axis or X-axis is representing number of questions asked in the survey. Vertical axis or Y-axis is representing frequency of the answers given by the subject. The answer consists of four choices which are Yes, No, Does Not Apply and Don't Know. Every choice is shown by a color which is shown in the bar graph. Different colored bars are representing those choices.

Requirement management is the first category of our survey questionnaires. It contained four questions whose statistical analysis is given below.

The frequency table of requirement management is given in Table 1. Four choices are given which are Yes, No, Doesn't Apply, Don't Know whose frequency is given 6, 3, 2 and 1 respectively. The percentages of these choices are 50, 25, 16.7 and 8.3 respectively.

Mean and standard deviation table of requirement management is given in Table 2. It consists of four questions whose mean is 1.50, 1.17, 1.83 and 1.92 respectively and standard deviation is 1, 0.389, 1.030 and 0.996 respectively.

Figure 3 shows pie diagram of requirement management. Diagram shows frequency percentage of four options which are Yes, No, Doesn't Apply and Don't Know. According to pie diagram, frequency of Yes is 50%, No is 25%, Doesn't Apply is 17% and Don't Know is 8%.

Table 6: Mean and standard deviation of software project tracking and oversight
Software project tracking and oversight

	Q9: Comparison between result and estimation	Q10: Corrective action	Q11: Known changes	Q12: Assigning responsibility
Mean	1.55	1.91	1.70	1.27
Standard deviation	0.522	1.221	0.949	0.467
Variance	0.273	1.491	0.900	0.218

Software project planning is the second category of our survey questionnaires. It contained four questions whose statistical analysis is given below.

The frequency table of software project planning is given in Table 3. Four choices are given which are Yes, No, Doesn't Apply, Don't Know whose frequency is given 8, 3, 0 and 0 respectively. The percentages of these choices are 66, 25, 0 and 0 respectively.

Mean and standard deviation table of software project planning is given in Table 4. It consists of four questions whose mean is 1.27, 1.18, 1.55 and 1.09 respectively and standard deviation is 0.218, 0.164, 0.873 and 0.091 respectively.

Figure 4 shows pie diagram of software project planning. Diagram shows frequency percentage of four options which are Yes, No, Doesn't Apply and Don't Know. According to pie diagram, frequency of Yes is 73%, No is 27%, Doesn't Apply is 0% and Don't Know is 0%.

Software project tracking and oversight is the third category of our survey questionnaires. It contained four questions whose statistical analysis is given below.

The frequency table of software project tracking and oversight is given in Table 5. Four choices are given which are Yes, No, Doesn't Apply, Don't Know whose frequency is given 5, 6, 0 and 0 respectively. The percentages of these choices are 41.7, 50, 0 and 0 respectively.

Mean and standard deviation table of software project tracking and oversight is given in Table 6. It consists of four questions whose mean is 1.55, 1.91, 1.70 and 1.27, respectively and standard deviation is 0.522, 1.221, 0.949 and 0.467, respectively.

Figure 5 shows pie diagram of software project tracking and oversight. Diagram shows frequency percentage of four options which are Yes, No, Doesn't Apply and Don't Know. According to pie diagram, frequency of Yes is 45%, No is 55%, Doesn't Apply is 0% and Don't Know is 0%.

Software configuration management is the fourth category of our survey questionnaires. It contained five questions whose statistical analysis is given below.

The frequency table of software configuration management is given in Table 7. Four choices are given which are Yes, No, Doesn't Apply, Don't Know whose frequency is given 7, 2, 0 and 1 respectively. The percentages of these choices are 58.3, 16.7, 0 and 8.3, respectively.

Table 7: Frequency table of software configuration management
Software configuration management:

	Frequency	%	Valid %	Cumulative%
Yes	7	58.3	70.0	70.0
No	2	16.7	20.0	90.0
Don't know	1	8.3	10.0	100.0
Total	10	83.3	100.0	

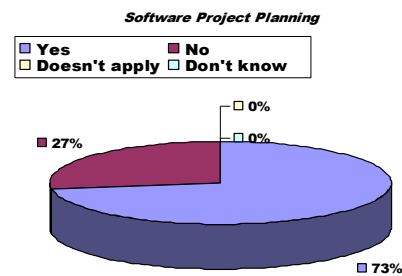


Fig. 4: Pie diagram of software project planning

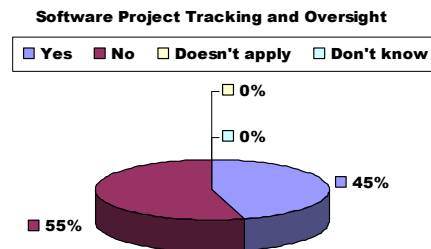


Fig. 5: Pie diagram of software project tracking and oversight

Mean and standard deviation table of software configuration management is given in Table 8. It consists of five questions whose mean is 1.50, 1.56, 1.40, 1.30 and 1.56, respectively and standard deviation is 0.972, 1.014, 0.516, 0.483 and 1.014, respectively.

Figure 6 shows pie diagram of software configuration management. Diagram shows frequency percentage of four options which are Yes, No, Doesn't Apply and Don't Know. According to pie diagram, frequency of Yes is 70%, No is 20%, Doesn't Apply is 0% and Don't Know is 10%.

Training program is the fifth category of our survey questionnaires. It contained four questions whose statistical analysis is given below.

The frequency table of training program is given in Table 9. Four choices are given which are Yes, No, Doesn't Apply, Don't Know whose frequency is given 3, 2, 0 and 1 respectively. The percentages of these choices are 25, 16.7, 0 and 8.3 respectively.

Table 8: Mean and standard deviation of software configuration management

Software configuration management:					
	Q13:Planned activities	Q14: Software work product	Q15:Configuration unit	Q16: Written organizational policy	Q17:Determining measurement
Mean	1.50	1.56	1.40	1.30	1.56
Standard deviation	0.972	1.014	0.516	0.483	1.014
Variance	0.944	1.028	0.267	0.233	1.028

Table 9: Frequency table of training program

Training program				
	Frequency	%	Valid %	Cumulative %
Yes	3	25.0	50.0	50.0
No	2	16.7	33.3	83.3
Don't know	1	8.3	16.7	100.0
Total	6	50.0	100.0	

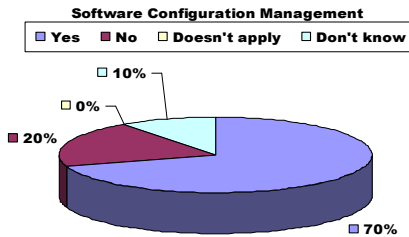


Fig. 6: Pie diagram of software configuration management

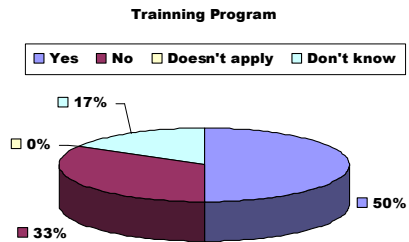


Fig. 7: Pie diagram of training program

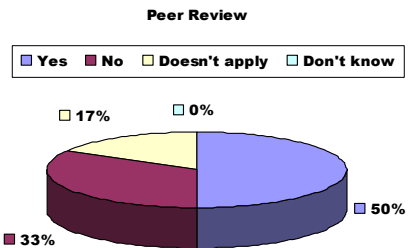


Fig. 8: Pie diagram of peer review

Mean and standard deviation table of training program is given in Table 10. It consists of four questions whose mean is 1.17, 1.33, 1.83 and 1.67, respectively and standard deviation is 0.408, 0.516, 1.169 and 0.816, respectively.

Figure 7 shows pie diagram of training program. Diagram shows frequency percentage of four options

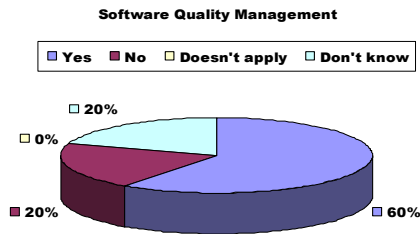


Fig. 9: Pie diagram of software quality management

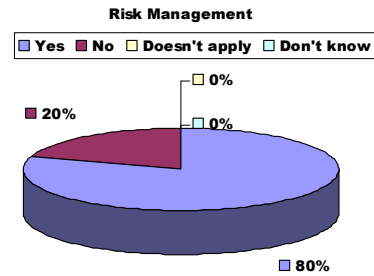


Fig. 10: Pie diagram of risk management

which are Yes, No, Doesn't Apply and Don't Know. According to pie diagram, frequency of Yes is 50%, No is 33%, Doesn't Apply is 0% and Don't Know is 17%.

Peer review is the sixth category of our survey questionnaires. It contained four questions whose statistical analysis is given below.

The frequency table of peer review is given in Table 11. Four choices are given which are Yes, No, Doesn't Apply, Don't Know whose frequency is given 3, 2, 1 and 0 respectively. The percentages of these choices are 25, 16.7, 8.3 and 0 respectively.

Mean and standard deviation table of peer review is given in Table 12. It consists of four questions whose mean is 1.67, 2.83, 1.50 and 1.67, respectively and standard deviation is 0.816, 0.983, 0.837 and 1.033 respectively.

Figure 8 shows pie diagram of peer review. Diagram shows frequency percentage of four options which are Yes, No, Doesn't Apply and Don't Know. According to pie diagram, frequency of Yes is 50%, No is 33%, Doesn't Apply is 17% and Don't Know is 0%. Software quality management is the seventh category of our survey questionnaires. It contained four questions whose statistical analysis is given below.

The frequency table of software quality management is given in Table 13. Four choices are

Table 10: Mean and standard deviation of training program

Training program				
	Q18:Planning training activities	Q19: Necessary training	Q20: Skills and knowledge	Q21:Use of measurement
Mean	1.17	1.33	1.83	1.67
Standard deviation	0.408	0.516	1.169	0.816
Variance	0.167	0.267	1.367	0.667

Table 11: Frequency table of peer review

Peer review				
	Frequency	%	Valid %	Cumulative %
Yes	3	25.0	50.0	50.0
No	2	16.7	33.3	83.3
Doesn't apply	1	8.3	16.7	100.0
Total	6	50.0	100.0	

Table 12: Mean and standard deviation of peer review

Peer review				
	Q22: Planned peer review	Q23:Action to solve defects	Q24:Necessary training	Q25: Activities for review and audit
Mean	1.67	2.83	1.50	1.67
Standard deviation	0.816	0.983	0.837	1.033
Variance	0.667	0.967	0.700	1.067

Table 13: Frequency table of software quality management

Software quality management				
	Frequency	%	Valid %	Cumulative %
Yes	3	25.0	60.0	60.0
No	1	8.3	20.0	80.0
Don't know	1	8.3	20.0	100.0
Total	5	41.7	100.0	

Table 14: Mean and standard deviation of software quality management

Software quality management				
	Q26: Measurement of activities	Q27: Senior management review	Q28: Selection of new technology	Q29: Senior management review on change of activity
Mean	1.20	1.40	1.80	1.80
Standard deviation	0.447	0.548	1.304	0.447
Variance	0.200	0.300	1.700	0.200

Table 15: Frequency diagram of risk management

Risk management				
	Frequency	%	Valid %	Cumulative %
Yes	4	33.3	80.0	80.0
No	1	8.3	20.0	100.0
Total	5	41.7	100.0	

Table 16: Mean and standard deviation of risk management

Risk Management			
	Q30:Risk calculation	Q31:Plan of action	Q32:Steps to avoid risk
Mean	1.20	1.20	1.20
Standard deviation	0.447	0.447	0.447
Variance	0.200	0.200	0.200

given which are Yes, No, Doesn't Apply, Don't Know whose frequency is given 3, 1, 0 and 1 respectively. The percentages of these choices are 25, 8.3, 0 and 8.3, respectively.

Mean and standard deviation table of software quality management is given in Table 14. It consists of four questions whose mean is 1.20, 1.40, 1.80 and 1.87, respectively and standard deviation is 0.447, 0.548, 1.304 and 0.447, respectively.

Figure 9 shows pie diagram of software quality management. Diagram shows frequency percentage of four options which are Yes, No, Doesn't Apply and Don't Know. According to pie diagram, frequency of Yes is 60%, No is 20%, Doesn't Apply is 0% and Don't Know is 20%.

Risk management is the eighth category of our survey questionnaires. It contained three questions whose statistical analysis is given below.

The frequency table of risk management is given in Table 15. Four choices are given which are Yes, No, Doesn't Apply, Don't Know whose frequency is given 4, 1, 0 and 0 respectively. The percentages of these choices are 33, 8.3, 0 and 0, respectively.

Mean and standard deviation table of software risk management is given in Table 16. It consists of three questions whose mean is 1.20, 1.20 and 1.20, respectively and standard deviation is 0.447, 0.447 and 0.447, respectively.

Figure 10 shows pie diagram of risk management. Diagram shows frequency percentage of four options which are Yes, No, Doesn't Apply and Don't Know. According to pie diagram, frequency of Yes is 80%, No is 20%, Doesn't Apply is 0% and Don't Know is 0%.

CONCLUSION

From the above given discussion and statistical analysis of the survey questionnaires, we can conclude that some software houses are well familiar with the software quality assurance techniques. The requirements to make quality software are taken in measure and are carefully implemented in the organizations. If these techniques are carefully implemented, organizations in Pakistan can develop and can compete with the organization of the developed countries and hence there will be more CMMI level five organizations in Pakistan.

REFERENCES

- Bawane, N. and C. Srikrishna, 2010. A novel method for quantitative assessment of software quality. *Int. J. Comput. Sci. Secur.*, 3(6): 508.
- Boegh, J., S. Depanfilis, B. Kitchenham and A. Pasquini, 1999. A method for software quality planning, control and evaluation. *IEEE Software*, 16(2): 69-77.
- Briand, L.C., S. Morasca and V.R. Basili, 1996. Property-based software engineering measurement. *IEEE T. Software Eng.*, 22(1): 68-86.
- Devnani-Chulani, S., 1998. Modeling software defect introduction. *Proceeding of California Software Symposium'97*. Center for Software Engineering, University of Southern California.
- Eldrandaly, K., 2008. A knowledge-based advisory system for software quality assurance. *Int. Arab J. Inform. Technol.*, 5(3): 304-310.
- Farooq, S.U., S. Quadri and N. Ahmad, 2011. Software measurements and metrics: Role in effective software testing. *Int. J. Eng. Sci. Technol.*, 3(1): 671-680.
- Florac, W.A., 1992. Software quality measurement: A framework for counting problems and defects: DTIC document. Technical Report: CMU/SEI-92-TR-022, ESC-TR-92-022.
- Humphrey, W.S., 2008. The software quality challenge. *Crosstalk J. Defense Softw. Eng.*, 21(6).
- Iftikhar, A. and S.M. Ali, 2011. Software quality assurance a study based on Pakistan's software industry. *Pak. J. Eng. Technol. Sci.*, 1(2): 65-73.
- Imam, A., S.A. Khoja and I. Shariff, 2007. *Improving Software Quality: A Benchmarking Approach*. Software Engineering Research and Practice CSREA Press, ISBN: 1-60132-034-5.
- Jacquet, J.P. and A. Abran, 1998. Metrics validation proposals: A structured analysis. *Proceeding of the 8th International Workshop on Software Measurement*, Germany.
- Jørgensen, M., 1999. Software quality measurement. *Adv. Eng. Softw.*, 30(12): 907-912.
- Kececi, N. and M. Modarres, 1998. Software development life cycle Model to ensure software quality. *Proceedings of the 4th International Conference on Probabilistic Safety Assessment and Management*. New York City, USA.
- Li, E.Y., H.G. Chen and W. Cheung, 2000. Total quality management in software development process. *J. Qual. Assur. Inst.*, 14(1): 4-6.
- Linda Westfall and The Westfall Team, 2005. 12 Steps to Useful Software Metrics. The Westfall Team. Retrieved from: floors-outlet.com/specs/spec-t-1-20111117171200.pdf.
- Mekprasertvit, C., 2004. Applying broadcasting/multicasting/secured communication to agentmom in multiagent-systems. CIS 895-MSE Project, Kansas State University.
- Offen, R.J. and R. Jeffery, 1997. Establishing software measurement programs. *IEEE Software*, 14(2): 45-53.
- Pressman, R.S. and D. Ince, 1992. *Software Engineering: A Practitioner's Approach*. Vol. 5, McGraw-Hill, New York.
- Sacha, K., 2005. *Evaluation of Software Quality*. Warsaw University of Technology, Poland, nd.
- Stavrinoudis, D. and M. Xenos, 2008. Comparing internal and external software quality measurements. *Proceedings of the 8th Joint Conference on Knowledge-Based Software Engineering (JCKBSE)*. IOS Press, Piraeus, Greece, August 25-28, pp: 115-124.