International Journal of Computer Science and Software Engineering (IJCSSE), Volume 8, Issue 2, February 2019
ISSN (Online): 2409-4285 www.IJCSSE.org Page: 40-47

Is Customer Satisfaction Enough for Software Quality?

Shahzad Hussain[1], Shahid Farid[2] and Iqra Mumtaz[3]

[1] Department of Information Technology, Bahauddin Zakariya University, Multan, Pakistan

[2] Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan

[3] Department of Computer Science, COMSATS University Vehari Campus, Vehari, Pakistan

ABSTRACT

Quality is not only crucial to the survival and success of software system but also a prime concern of software development. Therefore, Delivering the best software quality for a product is a challenging task for a software development organization. The overall software quality depends on various factors, like reliability, performance, functionality and customer satisfaction of a product. Generally, the local software development SMEs focus only on customer satisfaction. They pursue further development or may even stop when the customer says so. Yet, the overall software quality means more than just customer satisfaction including factors like reliability, maintainability, etc. This paper takes into account the other software quality factors that also need to be focused on making sure best quality of a software product, so that the quality of software production in the industry can be improved.

Keywords: *Software, Quality, Customer Satisfaction, Metrics, Practices, Pakistan.*

1. INTRODUCTION

Software quality is most of the fundamental components of Software Engineering (SE) as the quality of software products is accepted universally and now measured to be a vital component in commercial achievement as well [1]. Software quality is still immature, however, substantial research in the arena of SE has revolved around methodologies, standards, and techniques for measuring, improving and assuring the software quality[2, 3].Similarly, it is also crucial need to develop and utilize rigorous assessment models and mechanisms in order to provide facility and certify the quality of software systems [4]. Therefore, competition in the software industry is increasing steadily, but no organization can capture the software market until unless these do not develop quality systems and services [5]. most of the organizations need to evaluate and enhance early fault-detection at some stage in Software Development Life Cycle (SDLC) to enhance the implementation process as well as quality so that allows you to compete for the growing demands of software markets[6]. Such as software end users are demanding a high quality of software product and a lot of them are ready to pay a higher price for the best quality software product [2]. In fact, better quality software is not only becoming just a competitive advantage but also an essential element for the success of companies.

There is challenging task to define the quality of a software product as the term quality is very broad and consist on various dimensions such as service quality, information quality and system quality [7]. Furthermore, there are various stakeholders such as software engineers, software developers, testers, quality managers, project managers, database administrators and customers/users of a software system. There are diverse variations in the quality requirements and point of view of every stakeholder for the quality assurance of a software product. It represents that the quality depends on the context and the interpretation of the quality attributes and the association between those attributes within that particular situation[8]. Thus, measuring the quality of a software product is crucial in order to get appreciated results in software systems that are efficient, understandable, reliable, and acceptable for their investors [9]. Therefore, it is not significant or logical to consider that one definition fits the needs of all groups. Hence it is not possible to express the quality of a software product by simply setting the definition without understanding the various characteristics such as reliability, efficiency, usability, availability, user satisfaction and security about the software product.

Since the 1970s, practitioners, and researchers have been looking for ways to characterize software quality. They have found that software artifact can be broken down into constructs or quality characteristics that can be assured and measured. This enables evaluation of quality through the evaluation of more detailed characteristics [10]. These quality characteristics collectively reflect the overall quality of the system [11]. According to quality means conformance to predefined specifications that meet the customers' needs i.e. perception of a user or customer that up to what extent the software product meets their need and expectations. Similarly, it is urged by [12] quality refers to the extent or degree to which a

customer's requirement is met. In other words, quality reflects the user satisfaction. Whereas, it is argued by [13] quality of software system cannot be measured with a single factor like user satisfaction. Hence it is required to consider other quality factors like availability, complexity, size, flexibility, accessibility, reliability, maintainability and etc. Therefore, this study focuses on the exploration of the efficacy perceived by the different stakeholder for quality practices (characteristics) adopted by the localized software development SMEs to assure the software quality and account the quality with various characteristics required for the assurance.

1.1 Literature Review

By the IEEE Standard Appendix of Software Engineering terminology quality is evaluated as, *"The degree to which a system, component, or process meets specified requirements"*, IEEE further explained quality as *"The degree to which a system, component, or process meets customer or user needs or expectations"*. It is urged by [14]*" a product's quality is a function of how much it changes the world for the better"*. Quality is conformance to predefined specifications that meet the customers' needs. it is urged by [12] quality refers to the extent or degree to which a customer's requirement is met. It is clear from these definitions that quality of software refers to the measurement of various characteristics or elements of software system ranging from the requirement to implementation. These characteristics include availability, reliability, functionality, efficiency, usability and users' satisfaction etc. However, quality of a software product can be evaluated with methodologies, standards, and techniques. For measuring, improving and assuring the software quality [2, 3]. Quality divided into two parts as internal quality, external quality. Internal quality is gauged by software professionals in the phase of Developing and testing, whereas external internal quality can be achieved by the user's end [15]. It is clear from these definitions that quality of software refers to the measurement of various characteristics or elements of software system ranging from the requirement to implementation. According to [16], criteria defined by [17], that several elements constitute the quality of a system. These elements include availability, reliability, functionality, efficiency, usability and users' satisfaction etc.

Quality itself is difficult to measure and the quality of a software system is really a critical matter to assure due to its various dimensions such as information quality, system quality and service quality [7]. According to [18], nearly 94% of quality issues originate from a faulty system. As there are many stakeholders (including end users, administrators, managers, data operators, software engineers, testers etc.) of any software system and every stakeholder has their own views and needs of quality according to their specific requirements. The aim of a quality system is not only to fulfill the user requirements only, but other stakeholders must also be considered while designing a quality system. Moreover, the system must be attuned to the organization's overall strategic policies [19].

It is relatively easy to discuss quality and quality assurance, but it is quite difficult to measure the various characteristics of quality in the different phases of SDLC. The quality is generally described in terms of a quality model which characterizes the quality characteristics and relationship between them [20]. Therefore, quality of any software system can be measured and assessed in term of its several quality characteristics. However, it is difficult to measure all quality characteristics of a system. One way to assure the quality of a software product is to set minimum standards. There exist various software quality models in the literature to explicitly set minimum standards for the quality assurance of a software product. Unfortunately, existing quality models are deficient in addressing the customers' satisfaction explicitly. However, it can be gauged implicitly measuring various quality characteristics of the model.

Moreover, this study also identifies whether customer satisfaction is enough to assure the quality of software product. Therefore, these research questions have been formulated in order to encounter the objectives of this study in the context of the localized environment of Pakistan.

RQ.1: What are the crucial characteristics required for the quality assurance of a software product?

RQ.2: how user/developer perceive the Quality of software product?

1.2 Software Quality Models

In order to assure the quality of a software product, different researchers came up with various solutions in term of software quality models. These quality models include McCall's quality model[21], Boehm quality model[22], Dromey's quality model[23], FURPS quality model[24], ISO 9126 quality model[24] and so on [3]. Some of the familiar and standard quality models are discussed briefly.

McCall software quality model is considered to be the first software quality model in the world of SE. It has been proposed by McCall in late seventies [25]. The model describes the quality of software product as a hierarchy of factors, metrics, and criteria. Three major perspectives have been formulated to cover the various aspects of a software product. These perspectives include Product Operation (correctness, efficiency, reliability, usability, and integrity), Product Transition (portability, reusability and interoperability) Product Revision (maintainability, flexibility, and testability). There are 23

quality criteria to describe the internal view of the software (developer's view) and eleven quality factors to define the external view of the software (user's view). The metrics in the lower level of McCall's quality model are neither completely nor clearly defined as connected to the upper level of the quality model.

[26] that describes the characteristics of software on a larger scale as compared to McCall's model [27]. This model tries to qualitatively define the quality of software using a predefined set of metrics and properties. Boehm's quality model constitutes the hierarchical structure of characteristics, each of which leads to the total quality. This model consists of three high-level characteristics (including As-is-utility; Portability and Maintainability) seven intermediate-level characteristics (including reliability, efficiency, human engineering, portability, testability, understandability, and modifiability) and 15 lowest-level characteristics which contribute to the overall quality level in a structured manner [11]. The lowest-level characteristics can be used to provide a base for defining quality metrics.

A working framework has been proposed by [28] to evaluate the quality of the software when each software product has a different quality than the others. It consists of four software product properties including Correctness (functionality and reliability), Internal (maintainability, efficiency, and reliability), Contextual (maintainability, reusability, portability and reliability) and Descriptive (maintainability, efficiency, reliability and usability). Dromey's approach is significant as it allows developers and software engineers to verify models. But extensibility is not addressed as an explicit characteristic to represent future growths.

FURPS model of quality actually proposed by [29] and then its updated version FURPS+ release by IBM Rational Software. The quality characteristics that have been considered in this model are i) Functional requirements which are defined by expected input & output and ii) Non Functional requirements in which U stands for Usability, R stands for Reliability, P stands for Performance and S stands for Supportability.this model can't provide the facility of software portability and not apply completely on any system, it should customize before use. In addition, domain-specific attributes are also not addressed in this model. The FURPS quality model is a special-purpose model [1]. It addresses the quality requirements of a specific organization only as it has been developed and extended to be employed in the IBM Rational Software Company.

ISO 9126 [9] is an international standard for the evolution of software since it has been developed based on the international consensus and agreement from all the country members of ISO organization. ISO 9126-1 is an extension of previous work done by McCall, Boehm, FURPS etc. ISO 9126-1 comprises of **a)** Internal and external quality characteristics and **b)** Quality in use characteristics. The internal quality attributes are those properties of the system that can be measured without executing the system while external quality attributes can be evaluated by observing the system during its execution. There are six characteristics determine by the first part (functionality, reliability, usability, efficiency, maintainability, and portability), whereas the second part of the model highlights four qualities in use characteristics (effectiveness, productivity, safety, and satisfaction). The quality in use attributes refers the properties of the system that can be observed by the users when the system is in operation or in its maintenance phase. The standard recommends measuring the characteristics directly, but how to measure these characteristics is still a question. *There is a lack of a principal for determining which factors should be included in the quality definition.* Moreover, there is no description given that how the lowest level indicators are composed of an overall assessment of higher level quality characteristics [30].

1.3 Analysis of the Existing Software Quality Models

Most of the existing quality models (McCall, Boehm, Dromey, FURPS, ISO 9126 etc.) have been proposed for general application systems to measure the several quality characteristics of a software product explicitly. However, various characteristics can be deployed to gauge the customer or user satisfaction implicitly. Whereas it is urged by [31] customers' satisfaction is graded as one of the crucial factors for the software success and can be strongly linked to accomplishment of all functions of a business [32]. Therefore, companies can generate higher revenue by improving the quality of software product and increasing customer satisfaction [2]. On the other hand, it is difficult to explore the concept of user satisfaction due to its qualitative nature. Factors of satisfaction impacting on it need to be captured and used as a covariate. As measures of satisfaction are focused on the system and influential factors [33].

1.4 Customer Satisfaction

Evaluation of a product or service with regard to the needs and expectations of the user is considered as a user or customer satisfaction [34]. In other words, it can be stated that the physiological output of a customer during his/her experiences with the execution of a software product. It is usually measured as a gap between customer views before and after the interaction of a customer with the product. It usually implies as an accumulated temporary and sensory response about an artifact. According to [35], customer satisfaction can be categorized into five dimensions such as loyalty, satisfaction, recommendation, favorite and priority

options. Therefore, it is a difficult task for an organization to accomplish customer relations. As the level of satisfaction varies from customer to customer due to their diverse viewpoints about a software product, e.g. for an instance the efficiency of data entry is one of the important features for a data entry operator, whereas a software engineer may focus more on the correctness of the actions performed by the software product. Hence, in order to achieve the customer satisfaction, various viewpoints are required to be considered while designing and developing a software product.

The state-of-the-art literature identifies an array of service quality factors that are essential for customers. These factors include a) timeliness and convenience in providing better services to customers, b) vitality of personal attention for each customer to glue him with otherwise the user might choose alternative, c) level of reliability and dependability of a customer on the support staff, that lets the organization to mirror its friendliness nature to its customers and d) tangibles like physical facilities, equipment and appearance of the personnel are also key factors in developing good customer relations for an organization. Hence, customer satisfaction is one of the essential factors for an organization to successfully build well-engineered software products. Customer satisfaction in flip hinges at the best and consequences in their experiences and the services or goods they receive while using the product.

1.5 Customer's View of software quality

Quality is difficult to measure due to its variety of dimensions but crucial for a software product to be viable. Controlling the customer's view of quality is an important task. There is diversity in the perception of quality in terms of the product features from various customers. Certain factors contribute to achieving this goal. It has been usually seen that the first customers who receive a new product release often have a higher probability of highlighting a software-related issue than later customers. This can be due to lack of installation skills. This early-customers fashion means that the nice of the present day launch is anticipated conservatively before everything because at that time simplest clients who had deployed right away after General Availability (GA) are displayed via this metric [36, 37].

New users of a software product accompany the product's usability as the first dimension of quality. The initial learning time of understanding how the product operates dictates how easily this product can be used. Reduced learning time can improve the customer's insights about the product's quality. As the initial learning time is over, the customers focus on how quickly they can accomplish their desired tasks. Therefore, in order to achieve a quality product, it is vital for software engineers to utilize the appropriate software quality mechanisms (metrics) during each phase of SDLC. As software quality metrics have far-reaching effects on the quality of software product[6]. A couple of decades ago there were no mature quality assurance practices (metrics) available to assure the quality of software products to achieve the customer satisfaction. Once the product was ready to deliver, its quality was then assured using different methods, which often used to result in software failure or the software did not meet the customers' requirements[31, 38, 39].

The quality attribute is a difficult element to measure but it is crucial especially for a software system with the main objective to enable it to be viable or sustainable. The quality issue is considered as one of the major issues in a recent business scenario in general and for distributed systems (like e-learning systems) specifically [40].

Hence, the overall view of the customer satisfaction can be traced by considering two important factors. These are the in-process defects and field defects [41]. The former type of defects can occur during the development of the product and later usually happen due to limitations of the domain or field knowledge about a problem by the organization. Both these factors need to be at their lowest possible levels, in order to guarantee the best quality software product.

1.6 Research Method

In this research, focus on the achieving of software quality by the survey from the different software expert because survey can produce a massive amount of data timely and quickly for a fairly low cost and based totally on empirical statistics[42, 43].

After selecting the key quality requirement that focuses on the survey, next create the survey terms for assuring the quality, the questionnaire starts with an introduction that describes the cause of the survey and includes the instructions for finishing the survey for achieved the quality of the product. The example represents, Is Customer's Satisfaction enough for the Quality Assurance of Software product? First, describe the terms that are using into the survey and then their subterms. Survey Questionnaire has two portions internal and external factor for achieving the quality of software. The first portion includes on functionality, reliability, usability, efficiency, Maintainability, and portability,[5, 8] 2nd portion contains external factors that is Effectiveness, productivity, safety, security and user satisfaction [44]. All these terms are main factors for achieving the quality of software that attached in Appendix A.

The terms are measured on Likert Scales 1 to 5 for achieving the quality due to easy to construct, more likely to produce an extremely reliable scale, handy to read and complete for the participant. In this, each specific term can have its response analyzed separately [45, 46]. We will be able to evaluate the outcome as a whole using

descriptive records, and also the distinctive outcome for all phrases that normally treated as ordinal data.

Who to Ask?

The responder to our survey is very dependent on the relationship with the software product. Our targeted people in the survey were the Software engineer, designer, developer, tester and end user that were fully related to software usage and manipulation. We take the observation, how we can improve the quality of a software product by using these terms.

Data collection

Data were collected in form of survey from the 80 peoples for improving the quality of software product from different software house and universities of Pakistan, wherever used a structured survey which was derived from the review of literature, the questionnaire was about measuring and improve the software quality internally and externally by the 1 to 5 scale measurement.

Data Analysis

The data analysis and statistics for the survey shows the result that is very interested in software Quality, as well as depends on the external quality and internal characteristics both have a very important role in SQA. The composite reliability of this survey observes ranging between 4.4875 and 3.375. Where, the maximum mean value is (4.4875) for customer's satisfaction, while (4.325) the second largest value for security/safety in external factors. Customer's satisfaction and security/safety is the most important factor for quality of the product.

In the Internal quality factor, the Functionality and Reliability are the most important factor with the value of 4.225 and 4.2875, And reliability has more important than functionality.

The overall means value of Functionality (4.225), Reliability (4.2875), Usability (4.2), Efficiency (4.1375), Maintainability (3.725), Portability (3.375), Effectiveness (3.875), Productivity (4.2125), safety/Security (4.325), Satisfaction (4.4875) that are illustrated in Table 1.

Table 1: Quality Survey with Mean Values

So the conclusion says that by a team of developing and using we can't achieve the quality of a product without Functionality, Reliability, Security and Customer Satisfaction.

Experimentation

Local as well as software development organizations of Pakistan were visited and several practices regarding the levels of software quality assurance (SQA) practices were observed. The role of customer's satisfaction and intrinsic quality attributes of a product were highlighted in these surveys. We looked how and where do the boundaries of the local and national software development

Factors	Response	Level of Agreement					Means
		5	4	3	2	1	
Functionality	Frequency	39	24	11	4	1	4.225
	%	48	30	13.75	5	1.25	
Reliability	Frequency	40	26	11	3	0	4.2875
		50	32.5	13.75	3.75	0	
Usability	Frequency	42	18	16	2	2	4.2
	%	52.5	22.5	20	2.5	2.5	
Efficiency	Frequency	37	24	12	7	0	4.1375
	%	46.25	30	15	8.75	0	
Maintainability	Frequency	26	24	18	6	6	3.725
	%	32.5	30	22.5	7.5	7.5	
Portability	Frequency	18	18	25	14	5	3.375
	%	22.5	22.5	31.25	17.5	6.25	
Effectiveness	Frequency	24	30	19	6	1	3.875
	%	30	37.5	23.75	7.5	1.25	
Productivity	Frequency	36	32	7	3	2	4.2125
	%	45	40	8.75	3.75	2.5	
Safety/Security	Frequency	49	16	8	6	1	4.325
	%	61.25	20	10	7.5	1.25	
Satisfaction	Frequency	50	23	5	0	2	4.4875
	%	62.5	28.75	6.25	0	2.5	

environments lie regarding software quality.

Local Practices

Several practices regarding software development can be deduced from the experimental data. It was observed, that the requirements of a client dictate the delivery date of the project. Software quality boiled down to more or less client's needs. Generally, there was no clear-cut demarcation found between the software quality and customer satisfaction. Another aspect regarding software quality of a product was the availability of software maintenance facilities. The maintenance was usually done in a shorter span after the product delivery. The organizations charged sufficiently if requirements were slightly changed or modified. This, in turn, evolved as a gap between customers and organizations. When customers experience such situation, they usually stop telling the bugs or defects in the software products- which can be alarming.

2. CONCLUSION

Hence, there may be a want to develop such reliable software program that meets the user necessities. It is representing that end-user/learner satisfaction reflects the quality of software system, but user satisfaction alone is not sufficient to assure the quality of a software system/IS [47]. However, in practice, customer's satisfaction is the sign of best quality product for the software development SMEs and software engineers as well especially in the national and localized environment. However, quality of a software product cannot be measured on the basis of customers' satisfaction only. However, the user does not know what is going on behind the screen as they work with the interface of the product.

REFERENCES

[1] Jamwal, D., Analysis of Software Quality Models for Organizations. International Journal of Latest Trends in Computing (E-ISSN: 2045-5364), 2010. 19.

[2] Boehm, B., et al. Fifth workshop on software quality. in Companion to the proceedings of the 29th International Conference on Software Engineering. 2007. IEEE Computer Society.

[3] Farid, S., et al., Gauging Quality of Software Products Using Metrics. University of Engineering and Technology Taxila. Technical Journal, 2017. 22(1): p. 121.

[4] Mavromoustakos, S. and A.S. Andreou, WAQE: a web application quality evaluation model. International Journal of web engineering and technology, 2007. 3(1): p. 96-120.

[5] Bhatti, S.N., Why quality?: ISO 9126 software quality metrics (Functionality) support by UML suite. ACM SIGSOFT Software Engineering Notes, 2005. 30(2): p. 1-5.

[6] Farooq, S.U., S. Quadri, and N. Ahmad, Software measurements and metrics: Role in effective software testing. International Journal of Engineering Science and Technology (IJEST), 2011. 3(1): p. 671-680.

[7] Alla, M.M.S.O., The Impact of System Quality in E-learning System. Journal of Computer Science and Information Technology, 2013. 1(2).

[8] Al-Kilidar, H., K. Cox, and B. Kitchenham. The use and usefulness of the ISO/IEC 9126 quality standard. in Empirical Software Engineering, 2005. 2005 International Symposium on. 2005. IEEE.

[9] ISO, I., IEC 9126-1: Software Engineering-Product Quality-Part 1: Quality Model. Geneva, Switzerland: International Organization for Standardization, 2001.

[10] Nabil, D., A. Mosad, and H.A. Hefny, Web-Based Applications quality factors: A survey and a proposed conceptual model. Egyptian Informatics Journal, 2011. 12(3): p. 211-217.

[11] Al-Qutaish, R.E., Quality models in software engineering literature: an analytical and comparative study. Journal of American Science, 2010. 6(3): p. 166-175.

[12] Aziri, B., Managing Quality: With Special Emphasizes on SME's in The Pollog Region. Publications in International Scientific Publications: Economy & Business Journal, 2015. 9(1): p. 337-342.

[13] Hassanzadeh, A., F. Kanaani, and S. Elahi, A model for measuring e-learning systems success in universities. Expert Systems with Applications, 2012. 39(12): p. 10959-10966.

[14] Pressman, R.S., Software engineering: a practitioner's approach. McGraw Hill International Edition, 2005: p. 466-472.

[15] Jamwal, R.S., D. Jamwal, and D. Padha. Comparative analysis of different software quality models. in 3rd National Conference. 2009.

[16] Nichols, M.A. Development of a quality assurance system for e-learning projects. in ASCILITE. 2002.

[17] Garvin, D.A., What Does "hltoduct Quality" Really Mean? Sloan management review, 1984: p. 25.

[18] Deming, W.E., The new economics for industry, government, education. Cambridge: MIT Center for Advanced Engineering Study,| c1993, 1993. 1.

[19] Gilmour, P., R.A. Hunt, and H. Driva, Total quality management: integrating quality into design, operations and strategy. 2000: Corporate Insight Publishing.

[20] Suman, M.W. and M. Rohtak, A Comparative Study of Software Quality Models. International Journal of Computer Science and Information Technologies, 2014. 5(4): p. 5634-5638.

[21] McCall, J.A., P.K. Richards, and G.F. Walters, Factors in software quality. volume i. concepts and definitions of software quality. 1977, GENERAL ELECTRIC CO SUNNYVALE CA.

[22] Boehm, B.W., J.R. Brown, and M. Lipow. Quantitative evaluation of software quality. in Proceedings of the 2nd international conference on Software engineering. 1976. IEEE Computer Society Press.

[23] Dromey, R.G., A model for software product quality. IEEE Transactions on Software Engineering, 1995. 21(2): p. 146-162.

[24] Sharma, K. and K. Sharma. Comparison of Various Software Quality Models. in Proc. of the Intl. Conf. on Recent Trends in Computing and Communication Engineering RTCCE. 2013.
[25] McCall, J.A., P.K. Richards, and G.F. Walters, Factors in software quality. 1977: General Electric, National Technical Information Service.
[26] Boehm, B.W., et al., Characteristics of software quality. Vol. 1. 1978: North-Holland Publishing Company.
[27] Sanjay Kumar Dubey, S.G., Prof. (Dr.) Ajay Rana, Comparison of Software Quality Models: An Analytical Approach. International journal of Emerging Technology and Advanced Engineering, 2012. 2(2): p. 111-119.
[28] Dromey, R.G., A model for software product quality. Software Engineering, IEEE Transactions on, 1995. 21(2): p. 146-162.
[29] Grady, R.B. and D.L. Caswell, Software metrics: establishing a company-wide program. 1987.
[30] Kitchenham, B. and S.L. Pfleeger, Software quality: the elusive target [special issues section]. Software, IEEE, 1996. 13(1): p. 12-21.
[31] Nafees, T., Impact of user satisfaction on Software quality in use. International Journal of Electrical & Computer Sciences, 2011. 11(03): p. 48-56.
[32] Bashir, R.S., et al., Uml models consistency management: Guidelines for software quality manager. International Journal of Information Management, 2016. 36(6): p. 883-899.
[33] Griffiths, J.R., F. Johnson, and R.J. Hartley, User satisfaction as a measure of system performance. Journal of Librarianship and Information Science, 2007. 39(3): p. 142-152.
[34] Bai, B., R. Law, and I. Wen, The impact of website quality on customer satisfaction and purchase intentions: Evidence from Chinese online visitors. International journal of hospitality management, 2008. 27(3): p. 391-402.
[35] Shahin, A., A.A. Abandi, and M.H.M. Javadi, Analyzing the relationship between customer satisfaction and loyalty in the software industry-with a case study in Isfahan System Group. International Journal of Business and Social Science, 2011. 2(23).
[36] Mockus, A. and D. Weiss. Interval quality: Relating customer-perceived quality to process quality. in Proceedings of the 30th international conference on Software engineering. 2008. ACM.
[37] Mockus, A., P. Zhang, and P. Li, Drivers for customer perceived software quality. ICSE 2005, 2005: p. 225-233.
[38] Satisfaction, C., Improving quality and Access to Service and Support in Vulnerable Neighborhoods. What The Research Tells Us, Centre for study of social policy, 2007.
[39] VanSuetendael, N. and D. Elwell, Software quality metrics. 1991, COMPUTER RESOURCE MANAGEMENT INC PLEASANTVILLE NJ.
[40] Williams, J.B. and J. Jacobs, Exploring the use of blogs as learning spaces in the higher education sector. Australasian journal of educational technology, 2004. 20(2).
[41] Chulani, S., et al. Deriving a software quality view from customer satisfaction and service data. in European Conference on Metrics and Measurement. 2001.
[42] Xin, L. and W. Rong. Survey Research on Relationship among Service Failures, Service Recovery and Customer Satisfaction. in Management Science and Engineering, 2007. ICMSE 2007. International Conference on. 2007. IEEE.
[43] Kelley, K., et al., Good practice in the conduct and reporting of survey research. International Journal for Quality in health care, 2003. 15(3): p. 261-266.
[44] Westfall, L. Software Customer Satisfaction. in Proc. Applications in Software Measurement Conference. 2002.
[45] Allen, I.E. and C.A. Seaman, Likert scales and data analyses. Quality progress, 2007. 40(7): p. 64.
[46] Jamieson, S., Likert scales: how to (ab) use them. Medical education, 2004. 38(12): p. 1217-1218.
[47] Shee, D.Y. and Y.S. Wang, Multi-criteria evaluation of the web-based e-learning system: A methodology based on learner satisfaction and its applications. Computers & Education, 2008. 50(3): p. 894-905.

Appendix A.

Software Quality achieved Survey

The company committed the quality of software XYZ base on the SQA Model. It is required for software quality assurance and improvement from Software developer, designer, engineers, testers and end user depends on characteristics and quality factor for software quality.

These terms includes Functionality (Accuracy, Interoperability, Security, Suitability, Functionality Compliance) Reliability (Maturity, Recoverability, Fault Tolerance, Reliability Compliance), Usability (Learnability, Operability, Understand Ability, Attractiveness, Usability Compliance) Efficiency (Time Behavior, Resource Utilization, Efficiency Compliance) Maintainability (Analyzability, Changeability, Stability, Testability) Portability (Adaptability, Installability, Co-Existence, Replaceability)

Effectiveness(Task Effectiveness, Task Compilation, Error Frequency) Productivity(Task Time, Task Efficiency, Economic Productivity, Productive Proportion, Relative User Efficiency) Safety(Health Safety, Use Of People Affected, Economic Damage, Software Damage, System Damage) Satisfaction(Customers Satisfaction, Satisfaction Scale, Satisfaction Questionnaire)

It measured on a scale of 1 to 5, circle the appropriate number that represents the importance of each character or attributes. A score 1 is very unimportant (VU) and 5 being very important (VI).

NO	**Characteristics**	**Preference**				
		Importance				
		VU				**VI**
1	Functionality	1	2	3	4	5
2	Reliability	1	2	3	4	5
3	Usability	1	2	3	4	5
4	Efficiency	1	2	3	4	5
5	Maintainability	1	2	3	4	5
6	Portability	1	2	3	4	5
7	Effectiveness	1	2	3	4	5
8	Productivity	1	2	3	4	5
9	Safety/Security	1	2	3	4	5
10	Satisfaction	1	2	3	4	5