Software Quality Metrics for CRM: A Quantitative Approach

S.Sivagami Sundari
Thiagarajar College of Engineering,
Madurai.

G.Sri Arjuna Ragini
Thiagarajar College of Engineering,
Madurai.

T.Chandrakumar
Thiagarajar College of Engineering,
Madurai.

S.Parthasarathy
Thiagarajar College of Engineering,
Madurai.

ABSTRACT

Software development is similar to other engineering disciplines: a managed approach is likely to improve the quality of the product. As software becomes more and more pervasive, there has been a growing concern in the information technology industry about software quality. Though various software quality models are in place to measure and manage the software quality, there is no effective relation among software quality models and the practice of assessing the software quality remains unsolved. The objective of this paper is to assess characteristics of software quality quantitatively for CRM based applications. We use the quality attributes defined by the International Standard for Software Product Evaluation ISO/IEC 9126 and a set of software metrics proposed in this research study have been applied to these attributes for its measurement. The results are then validated using statistical techniques. The statistical analysis gives an input to the software quality assurance team to focus at the specific software quality attributes which leads to the poor quality of the software. The software quality metrics are applied on the CRM applications. The significance of our research study is illustrated with suitable datasets and future research directions are indicated.

Keywords

Software quality, software metrics, quality model, software quality characteristics, quality assessment, CRM.

1. INTRODUCTION

The significance and need of high-quality software has expanded with the fast transition of humanity to information-oriented era. [7] [19]. Software quality is defined as conformance to explicitly state functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software. Software

Quality Assurance is a sunshade undertaking that is applied all through the software process. Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project. In general, software quality metrics are more closely associated with process and product metrics than with project metrics. The growing complexity of software systems, construct mostly integrating components offering services, rises the need of more rigorous approaches for considering the software quality quantitatively. Therefore, it is nowadays of foremost significance to fulfill not only the customer's obligations that the software accomplishes, but also to take into account the quality of the software. A software system that undergoes continuous change, such as having new functionality added to its original design, will eventually become more complex and can become disorganized as it grows, losing its original design structure [5]. The object deployment of distributed software has a great impact on its performance [8]. Knowing your customers better will enable you to serve them better and keep them loyal forever. CRM is relationship orientation, customer retention and superior customer value created through process management [35]. CRM is the building of a customer-oriented culture by which a strategy is created for acquiring, enhancing the profitability of, and retaining customer that is enabled by an IT application; for achieving mutual benefits for both the organization and the customers [36].

According to ISO 9126, quality is defined as a set of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs. Software quality can be viewed from many different, equally valid perspectives; they are Contractual, attributes and User-Focused. Software quality metrics for CRM are a subset of software metrics that focus on the quality aspects of the product, process, and project. Product metrics are also referred to as software metrics. They are directly associated with the product itself and attempt to measure product quality or characteristics of the product that can be connected with product quality. Process metrics concentrate on the process of software development and measure process structures with the aim of either distinguishing problems or pushing forward effective practices. Resource metrics are associated with the properties that are essential for the development of software systems and their realization.

The objective of this research study is to enable the software engineering group to is consigned to the customer. Though not all quality characteristics are of identical significance to a software product and the quality characteristics are generally not part of the stakeholders' terminology, it is essential to ensure that software products encounters international standards. Since 1920, software metrics and software quality models are known as major reference in the field of software quality assessment. But there is still no clear procedure for considering software quality via software metrics for CRM. In the first part of the paper, we review the quality models in practice, some problems and solutions available. In the second part, we introduce a practical example of software quality assessment based on a set of software metrics for CRM proposed in this research study. With increasing demands on software functions, software systems become more and more complex [32]. CRM architecture is shown in Figure1.

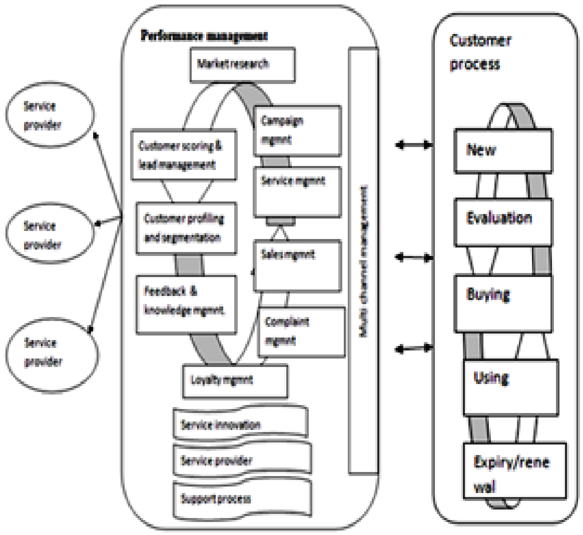

Fig 1.CRM Architecture

We define CRM as an approach that achieves an optimum balance between corporate investments and the satisfaction of customer needs in order to generate maximum profits. This entails: 1. acquiring and continuously up dating knowledge on customer needs, motivations, and behavior over the lifetime of the relationship. 2. Applying customer knowledge to continuously improve performance through a process of learning from successes and failures. 3. Integrating marketing, sales, and service activities to achieve a common goal. 4. The implementation of appropriate systems to support customer knowledge acquisition, sharing, and the measurement of CRM effectiveness. To integrate marketing, sales, and service activities, CRM requires the strong integration of business processes that involve customers .Their performance is predominantly influenced by the underlying supply of knowledge on products, markets, and customers.

2. LITERATURE REVIEW

The quality model in ISO/IEC 9126 [2] has been developed through the models from Boehm [1], McCall [15]. The first software metrics were suggested in the mid ’70s and since then, a great number of metrics have been recommended in the following years. Capability Maturity Model (CMM) developed by the Software Engineering Institute (SEI) is a model of 5 levels of process 'maturity' that determine effectiveness in delivering quality software. Most software quality metrics are defined at the level of individual software components; there is a need for aggregation methods to summarize the results at the system level [25]. It can be used to guide process improvement over a task, a division, or an entire organization. CMMI helps to integrate conventionally distinct organizational functions, set process enhancement goals and main concerns, provide guidance for quality processes, and provide an issue of reference for appraising current processes. During the development [28] systematically performed measurements on the source code, using software metrics that have been proved to be correlated with software quality, such as the Chidamber and Kemerer suite and Lines of Code metrics. The important elements of the quality of the software that is considered a service to a business function are analyzed [34].

In [1], a conceptual framework is proposed for the characteristics of software quality. From [3], it is evident that software quality characteristics can be measured using software metrics. In [4], a method is used to identify the most important quality characteristics defined by ISO 9126 Standard by means of a risk assessment. [6] believes that the software quality metrics can be used to detect and remove problems with the software processes. Ion I. [31] presented a model and proposed standards for measuring software quality and also few software metrics for software complexity. In [12], it is reported that application of software metrics acts as an improvement driver for Capability Maturity Model (CMM). In [10], a quantitative software quality evaluation model is proposed with respect to the Component Based Development (CBD) methodology. In [14], it is mentioned that the CMM, a popular software quality model is meant for improving the software processes during the software development. [17] provides a methodology for identifying and analyzing software quality requirements. Thomas E. Murphy [20] focused on software quality and testing tools to manage software quality assurance activities. Wojciech Basalaj [21] proposed coding standards compliance as a measure of internal quality of a software system. [27] Presents a novel metric for measuring the readability of Software Source Code (SC).

The ISO/IEC 9126 Standard has been used in a number of previous studies: measuring the quality of architectural design [13], R&D project evaluation [16], identifying relevant criteria for evaluating software products [18] and measuring customer’s perception of software quality [22]. From the literature review we observe the following: (a) Few quality models are found in the literature and no systematic methodology is found to quantitatively assess the software quality to improve the software processes there by the software product (b) Software metric is the most suitable methodology to assess the software quality characteristics (c) Of all the software quality models discussed here, the ISO/IEC 9126 quality model is appropriate quality model for our research study. (d) Investigation of the quality attributes defined in the software quality model is very much essential to ensure customer satisfaction and increase product quality. (e) ISO/IEC 9126 quality model possesses quality characteristics to measure the software processes involved during the software development as well as the software product. Activity-based quality models break down this complex concept into concrete definitions, more precisely facts about the system, process, and environment as well as their impact on activities performed on and with the system [23]. According to the IEEE standard glossary of software engineering, Object-Oriented design is becoming more important in software development environment and software Metrics are essential in software engineering for measuring the software complexity, estimating size, quality and project efforts [26]. Production of high quality software is considered as one of the key factor in the rapid growth of Global Software Development (GSD) [28].

Software metrics are valuable entity in the entire software life cycle. Software metrics provides measurement of the software product and the process of software production [38] [39]. Good metrics should enable the development of models that are efficient of predicting process or product spectrum. Thus, optimal metrics should be: Simple, Objective, Easily obtainable, Valid, Robust [40]. From [41], we found functionality and portability are very much important for CRM based applications and sub-characteristics of Functionality are Suitability, Security, Compliance, Accuracy and Interoperability and sub-characteristics of Portability are Adaptability, Installablility, Conformance and Replaceability.

3. RESEARCH OBJECTIVE

Estimations are the basis from which planning is performed on a program. Planning a software quality assurance requires a frame of reference and an ability to measure against it. The program manager has three major measures with which to estimate the program: products, processes, and resources. It is argued that the quality of software product has an effect on the degree of success or failure of a software development program. We have developed a metric for measuring the quality of software management along the software quality attributes. The quality of software development management tools has improved over the past 30 years.

User satisfaction is often considered a critical outcome of quality management and the studies have shown a positive impact on organizational cost, profit and sales growth. Although the information technology market is rapidly growing, users are often dissatisfied with software quality. While several studies have developed a variety of software quality models to produce software with rich quality, there has been little or no attempt to assess the software quality characteristics defined in the quality model using appropriate methodology. The main concern of this research study is measuring the software quality characteristics defined in the quality model ISO/IEC 9126 [2] [13] using software metrics. To achieve this goal, we have defined a set of software metrics to assess the presence of each of the quality characteristics defined in the quality model. The significant feature of this study is the results obtained after the application of the proposed software metrics are validated using correlation, a statistical technique. We found from the literature [41] that software quality characteristics functionality & portability are much important for a CRM applications. Hence we have found a new metric for CRM applications.

3.1. Software Metrics and Software Quality

Software quality is defined as the quality that ensures customer satisfaction by offering all the customer deliverables on performance, standards and ease of operations. Software metrics and quality models play a significant role in software system quality measuring. Software metrics are characterized as benchmark of measurement, utilized to refer the attributes of certain thing being measured, such as quality or complexity, in a target kind, but subjective measurement of quality comes from human estimation. The most important objective of any engineering activity is to produce high quality product with limited resources and time. The quality of the product cannot be determined if it is be measured.

The quality of the end result depends upon the quality of the intermediate work products. If the requirements, design, code, and testing functions are of high quality, then the chances are that the end product will also be of good quality. At the project level, the primary focus is to measure errors and defects and derive relevant metrics such as requirement or design errors per function point, errors uncovered per review hour, errors per thousand lines of code. These metrics provide an insight into the effectiveness of the quality assurance activities at the team as well as individual level. Unlike in other engineering discipline, SE has not yet attains a level where there are standard and globally acceptable measurement metrics and models [29]

Software metrics can be classified into three categories: product metrics, process metrics, and project metrics. Product metrics describe the characteristics of the product such as size, complexity, design features, performance, and quality level. Process metrics can be used to improve software development and maintenance. Project metrics describe the project characteristics and execution. In this paper we use product metrics and process metrics to assess the software quality characteristics. In the specific case of libre (free, open source) software, the availability of a mature and reliable development community is an important factor to be considered, since in most cases both the resolvability and future fitness of the product depends on it [24]. Till date, the quality models for communities have been based on the manual examination by experts, which is time-consuming, generally inconsistent and often error-prone. In [24], an automated quality model has been proposed. The quality model used is a part of the Qual OSS quality model, while the metrics are those collected by the FLOSS Metrics project.

The ISO/IEC 9126 standard describes a software quality model which categorizes software quality into six characteristics (factors) which are sub-divided into sub-characteristics (criteria). The characteristics are manifested externally when the software is used as a consequence of internal software attributes. The internal software attributes are measured by means of internal metrics (monitoring of software development before delivery). The quality characteristics are measured externally by means of external metrics (evaluation of software products to be delivered). As discussed in the literature review, in this research study, we have considered the software quality characteristics proposed by the ISO/IEC 9126 as it enables the software engineering community to evaluate the software product as well as the software processes involved in the software development. Measures of software quality, correctness, maintainability, integrity and usability provide useful insight.

3.2. Proposed Software Metrics for CRM

We have considered the two quality characteristics found from literature functionality and portability defined in ISO/IEC 9126 and proposed a set of software metrics to assess these characteristics and associated sub-characteristics quantitatively. The set of well defined metrics to achieve this task is shown in Table 1. Detailed explanation on ISO/IEC 9126 shall be found in [2] [13]. There are totally two quality characteristics and nine sub-characteristics.

The metrics are well defined in such a way that it can be easily validated once the results are obtained after applying it to a software product. This will increase the confidence level of the software engineering group on the measurement of software quality.

Table 1. Software Metrics to Assess Software Quality

Software Quality Characteristics	Sub -Characteristics	Proposed Metrics
1-Functionality	A-Suitability	No. of undetected functions during system testing / total no. of functions in the specification
	B-Accurateness	Total number of variations in the obtained results to the expected results for a set of given inputs
	C-Interoperability	Interaction found with no. of system / total no. of given System
	D-Compliance	No. of satisfied requirements / Total no. of requirements
	E-Security	No. of attempts (succeeded /failed) / Total no. of attempts
2-Portability	A-Adaptability	No. of platforms to which the software is applicable / Total no. of platforms
	B-Installability	Time required to install the s/w in the specific environment / standard installation time
	C-Conformance	No of (Standards/ conventions) performed to which the software adheres / Total no. of (standards / conventions)
	D-Replaceability	Adaptability + Installability

Software metrics refers to a broad range of measurements for computer software. Measurement can be applied to the software process with the intent of improving it on a continuous basis. Some software development practitioners point out that simplistic measurement can cause more harm than good. Measurement enables the Organization to improve the software process; assist in planning, tracking and controlling the software project and assess the quality of the software thus produced. It is the measure of such specific attributes of the process, project and product that are used to compute the software metrics. Metrics are analyzed and they provide a dashboard to the management on the overall health of the process, project and product. Generally, the validation of the metrics is a continuous process spanning multiple projects.

4. CASE STUDY, DATA ANALYSIS AND FINDINGS

We have applied the proposed software metrics to a CRM application, module B of Auto Mart system containing the following functions: Financial accounts, MIS Report, Payroll, Equipment maintenance, user management, Inventory; Application of the software metrics to all the modules of software will give an entire image on the degree of software quality possessed by the software product. The different modules are identified with one another. There exists intra and inter relationship among the modules. Functionality and Portability of this product were examined utilizing the proposed metrics. The correlation between the various software quality attributes has been found to analyze the impact of one quality attribute over another. The project documentation of the software product was found useful in fetching the data for the software quality metrics.

The detailed analysis was conducted with the software engineering team comprising the software quality group from the association involved in the software development and the software installation team from the Auto Mart. The results of various test cases executed by the software testing team using manual testing and automated testing tools were also found useful and recognized as inputs for the proposed software metrics. For every quality attribute, there exist sub-characteristics. The computation is done for every sub-characteristic for the module B. In some modules, some of the quality characteristics will not be covered at all during the design. In these cases, the computation cannot be done, as no input data will be available. Here, the respective quality characteristics have to be added to the software design, in concurrence with the customers. There are cases, where the customer may not require certain software quality characteristics to a greater extent and some quality attributes may not be applicable to them. All these issues must be thought seriously about by the software quality team during the quantitative appraisal of software quality using software metrics.

Table 1 shows the software metrics proposed in this research study to measure the degree of quality possessed by a software product. These metrics will be applied to software after the testing procedure is finished and before it is conveyed to the client. In any case, few of the metrics could be connected as and when a part or a module of the software product is primed, rather than sitting tight for the finish of the whole software product. These metrics will measure the degree at which the developed software product meets the customer requirements as well as the other quality requirements specified in the quality model ISO/IEC 9126. The essential data source for these metrics to evaluate the software quality is the Software Requirements Specification (SRS) document and results from the software testing team. The next step is to identify a team who are involved in the software development to execute the software product under the different circumstances cited in the table 1 and compute the software metric for the respective software quality characteristics and sub-characteristics. The results are shown in Table 2 and Figure 2.

Table 2. Application of Software Metrics to Quality Characteristics

Software Quality Characteristics	Sub-Characteristics	Software Quality Estimation
1-Functionality	A-Suitability B-Accurateness C-Interoperability D-Compliance E-Security	28 52 51 73 65
2- Portability	A-Adaptability B-Installability C-Conformance D-Replaceability	62 63 74 63

From the table 2 and figure 2, we find that the module B of the Auto Mart software adheres to the software quality characteristics defined in the ISO/1EC 9126 quality model. To be specific, except the few quality attributes like suitability, adaptability and security, the other quality attributes are satisfied by the module B of Auto Mart software system to a greater extent. The application of software metrics to the module B gives good clarity of comprehension of the software quality possessed by it. This poses a positive climate for the measuring the software quality attributes using the software metrics. Using the results shown in the table 2, the software engineering team can customize the software product based on the quality attributes in which the software product is yet to improve and perform to fulfill the ISO/IEC 9126 quality requirements. Similar to the module B, the software metrics can be applied for each of the modules of the Auto Mart software system. The result of this exercise is used to validate the software metrics used for measuring the software quality characteristics.

This is done using the Pearson's Correlations Coefficient [30] for the Auto Mart software system as shown in the following table 3. Table 3 will enable the software engineering team to get an idea on the relationship between the different software quality characteristics. Positive values indicate a relationship between the two variables and negative values indicate the existence of negative correlation between the two variables. Table 3 enables us to identify the influence and the impact of one quality characteristic over the other.

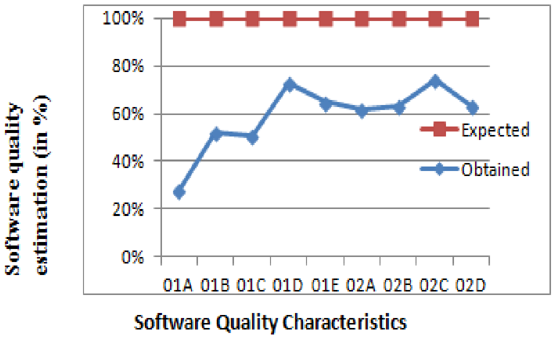

Fig. 2. Software Quality Characteristics

Using table 3, the software engineering processes can be redefined and quality requirements can be achieved for a software product.

Table 3. Correlation Matrix for Software Quality Characteristics

	Functionality	Portability
1-Functionality	1	0.641
2-Protability	0.616	1

5. CONCLUSION AND FUTURE RESEARCH SCOPE

The significant and positive impact of proposed software metrics on development quality demonstrated the contribution of certain quality attributes in improving the software development process: they may allow a semi-automated software development process, thereby introducing fewer errors in the code. The significant and positive relationship between software quality attributes indicated that, for our dataset, larger projects tended to have less defects identified during the development process; it is possible that developers are more careful when they work on large projects. The findings are highly relevant to software managers, as they illustrate the positive effect of software metrics on the software quality measurement and management. Our study was unique in that it used data on projects from a real time environment. The use of such comprehensive data for the empirical validation of the proposed software metrics enhanced the generalizability of our results. Data on real time software projects were used to validate the proposed model.

The greatest challenges in producing a quality software product are: (i) To find a suitable quality model that will provide systematic guidance for building quality into software and (ii) Assessment of software quality characteristics defined in the quality model. Thus it is evident that achieving the software quality is the need of the hour in software projects. Our approach is based on the quality characteristics of ISO/IEC 9126 quality model. We have defined a set of software metrics to assess quantitatively the quality requirements possessed by the software product according to ISO/IEC 9126 Standard. This has been applied to a software product and the results obtained are very much compelling. Correlation, a statistical technique is applied to find the degree of relationship between one quality attribute and another, thereby finding the influence of one over the other. Measurement always includes some amount of random error. Sub characteristics are relatively reliable if random measurement error minimally affects them. The above software quality metrics are well suited for CRM applications. Therefore, to give confidence to our results, it is crucial to estimate the amount of error in the software quality assessment.

As a future research work, remaining quality characteristics can be checked.

6. REFERENCES

[1] J.W. Horch, Practical Guide to Software Quality Management, Artech-House Publisher, 1996.

[2] G.C. Schulmeyer, J.I. McManus, Handbook of Software Quality Assurance, 3rd Ed., Prentice Hall, 1999.

[3] Capers Jones, Strength and Weaknesses of Software Metrics, Software Productivity Research, Version 5, March 22, 2006.

[4] Cem Kaner and Walter P. Bond, Software Engineering Metrics: What Do They Measure and How Do We Know?, 10th international software metrics symposium, metrics, 2004.

[5] Ivana Turnu, Giulio Concas, Entropy Of Some Ck Metrics To Assess Object-Oriented Software Quality, Int. J. Soft. Eng. Knowl. Eng. 23, 173 (2013).

[6] Douglas Hoffman, The Role of the Quality Group in Software Development, Pacific Northwest Software Quality Conference, 1993.

[7] Erik Van Veenendal and Rob Hendriks, Measuring Software Product Quality, SQP Vol.5, No.1, ASQ, 2002.

[8] Omid Bushehrian, Software Performance Engineering By Simulated-Based Object Deployment, Int. J. Soft. Eng. Knowl. Eng., 23, 211 (2013).

[9] Ion I Bucher, on quality and measures in software engineering, Journal of applied quantitative methods, Vol.1 No.2 Winter 2006.

[10] Jos J.M. Trienekens and Rob J. Kusters, Targets, drivers and metrics in software process improvement: Result of survey in multinational organization, Software Quality Journal, Vol. 15 pp: 135-153, Jan 2007.

[11] Kilsup Lee and Sung Jong Lee, A quantitative software quality evaluation model for the artifacts of component based development, proceeding of the sixth international conference on software engineering, IEEE, 2005.

[12] Linda Westfall, 12 steps and useful software metrics, The Westfall team, suite 270, ASQ, 2005.

[13] Manish Agarwal and Kaushal Chari, Software Efforts, Quality, and Cycle Time: A Study of CMM Level 5 Projects, IEEE Transaction on Software Engineering, Vol. 33. No.3, March 2007.

[14] Norman F. Schneidewind, New Software-Quality Metrics Methodology Standard fills measurement need, Naval Postgraduate School and 1061 Working Group Chair, April 1993.

[15] Robert A, Martin and Lawrence H. Shafter, Providing a framework for effective software quality assessment: A first step in automating assessment, the first annual software engineering conference, Software Systems Modernization, Virgina, April 1996.

[16] Thomas E. Murphy, Key Issues for Software Quality and Testing, Gartner, Inc, 2007.

[17] Wojciech Basalaj, Correlation between Coding Standard Compliance and Software Quality, Embedded Systems Conference, San Jose, 2006.

[18] Yi Liu, Jenq-Fpung Yao, Studying software metrics based on real-world Software systems , Georgia College and State University, JCSC, Vol.22, May , 2007.

[19] F. Losavio, L. Chirinos, A.Matteo, N.Levy , and A. Ramdame-Cherif , ISO Quality Standards for Measuring Architecture, Vol. 72, No. 2, 209-223, 2004.

[20] T. Miyoshi, M. Azuma , A empirical study of evaluating software environmental quality, IEEE Transaction on Software Engineering, Vol. 19, No. 5, 425-435, 1993.

[21] W, Ossadnik ,O. Lange, AHP- based evaluation of AHP-Software, European Journal of Operational Research, Vol. 118, No. 3, 578-588, 1999.[29] M.Zenos, D. Christodoulakis, Measuring perceived software quality, Information and Software Technology, 39, 417-424, 1997.

[22] Levine, D.M., Berenson, M., & Stephan, D, Statistics for Managers, 2nd Edition, New Jersey: Prentice Hall, 1999.

[23] Stefan Wagner, A Bayesian network approach to assess and predict software quality using activity-based quality models, Information and Software Technology', Volume 52, Issue 11, November 2010, Pages 1230-1241.

[24] Daniel Izquierdo-Cortazar, Jesús M. Gonzalez-Barahona, Santiago Dueñas, Gregorio Robles, Towards Automated Quality Models for Software Development Communities: The QualOSS and FLOSSMetrics Case, Quatic, pp.364-369, 2010.

[25] Mordal k, Nicolas Anquetil, Jannik Laval and Alexander Serebrenik, Software quality metrics aggregation in industry, Journal of Software: Evolution and Process, 13 Aug 2012, DOI: 10.1002/smr.1558.

[26] Amit Sharma, Sanjay Kumar Dubey, Comparison of Software Quality Metrics for Object-Oriented System, International Journal of Computer Science & Management Studies, Special Issue of Vol. 12, June 2012.

[27] Bajeh, A. O., A Novel Metric For Measuring The Readability of Software Source Code, Journal of Emerging Trends in Computing and Information Sciences, VOL. 2, NO. 10, October 2011.

[28] Giulio Concas, Michele Marches, Giuseppe Destefanis, An Empirical Study Of Software Metrics For Assessing The Phases Of An Agile Project, Int. J. Soft. Eng. Knowl. Eng. **22**, 525 (2012).

[29] S.W.A. Rizvi and R.A. Khan, Maintainability Estimation Model for Object Oriented Software in Design Phase (MEMOOD), Journal of Computing, vol. 2 Issue 4, 2010.

[30] International Organization for Standardization, "ISO Standard 9126: Software Engineering Product Quality, parts 1, 2 and 3", International Organization for Standardization.

[31] Hernan Astudillo, Five ontological levels to describe and evaluate software architecture, Department of Informatics, University Of Tarapaca , Vol.13.No.1, 2005

[32] Jizhou Zhan, Xianzhong Zhou, And Jiabao Zhao, Impact Of Software Complexity On Development Productivity, J. Soft. Eng. Knowl. Eng. 22, 1103 (2012).

[33] Sadia Rehman, Siffat Ullah Khan, Swot Analysis Of Software Quality Metrics For Global Software Development: A Systematic Literature Review Protocol, IOSR Journal of Computer Engineering, Volume 2, Issue 1 (July-Aug. 2012), PP 01-07

[34] Vidan Markovic And Rado Maksimovic, A Contribution To Continual Software Service Improvement Based On

The Six-Step Service Improvement Method, Int. J. Soft. Eng. Knowl. Eng. 22, 549 (2012).

[35] L. Ryals and S. Knox, Cross –functional Issues in Implementation of Relationship Marketing Through Customer Relationship Management. European Management Jourrnal, vol19, p.534-542, 2001.

[36] K. Rababah, H. Mohd, and H. Ibrahim, A Unified Definition of CRM towards the Successful Adoption and Implementation, in the 3rd Lifelong Learning International Conference (3LInC'10), Kuala Lumpur-Malaysia, 2010.

[37]. King, S-F., Burgess, T-F. (2008) Understanding Success and Failure in Customer Relationship Management, Industrial Marketing Management, 37(2008), pp. 421-431.

[38] Rigby, D. K., Reichheld, F. F., Schefter, P. (2002) Avoid the Four Perils of CRM, Harvard Business Review, 80(2), pp. 101-109.

[39] Mrinal Singh Rawat, Arpita Mittal, Sanjay Kumar Dubey, Department of Computer Science ,Amity University, Noida, India. Survey on Impact of Software Metrics on Software Quality, International Journal of Advanced Computer Science and Applications, Vol. 3, No. 1, 2012.

[40] G. Eason, B. Noble, and I. N. Sneddon, On certain integrals of Lips Software Metrics SEI Curriculum Module, SEI-CM-12-1.1, December 1988.

[41] S.Parthasarathy, Evaluating software quality using software metrics for a software product, International Journal of Information Systems and Change Management (IJISCM), Vol. 5, No. 2, 2011