Volume 6, Issue 7, July 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Quality Management Impact in Software Engineering - A Product Based Approach

Minimol Anil Job
Faculty of Computer Studies
Arab Open University, Kingdom of Bahrain

Nouhad Amaneddine
Faculty of Computer Studies
Arab Open University, Lebanon

Abstract—Ensuring Quality Management is very important in Software Projects. Quality is achieved to the extent that products that is fit for their purpose and of sufficiently high quality so that they meet the customer's requirements and expectations. Also there is a need to produce a high-quality product, but at a predetermined cost and within a predetermined timescale. This research paper focuses on how to assess the quality of a software product during its development and on its completion. A proposed quality management model using a number of software quality factors and metrics as ways of measuring quality of the development of an appropriate software product is introduced in this research. In this paper, to analyses quality factors during software development stages, the researchers used an example of an online Learning Management System (LMS) as a software product.

Keywords — software quality, quality control, quality factors, quality management, internal and external quality

I. INTRODUCTION

Quality is defined as "the totality of features and characteristics of a product or a service that bear on its ability to satisfy the stated and implied needs" [15]. Software that is fit for its purpose and is of sufficiently high quality is said to be of appropriate quality and it will be in conformance to the requirements and expectations of the customer.The software product should deliver the required functionality (*functional requirements*) with the required quality attributes (*non–functional requirements*) [1][2].A customer's requirements for software can take many forms. In general, they appear in a requirements statement that is part of the contract between developer and client. The requirements can also arise and be documented during the stages of software development process. To establish whether quality is appropriate, various measures can be used such as "how the product operates against what is required and expected by the customer", "quality and development standards have been followed during the development of the product" and "best practice of software engineers, as professionals operating to professional standards, has been followed during development"[3][4]. Quality is related to the customer's requirements; it is required to provide working definitions of what factors affect product quality, together with ways of measuring them.

II. EXTERNAL AND INTERNAL QUALITY

External quality or functional quality refers to the quality features seen by an end user and relate to fitness for purpose. The primary method of assessing external quality is by testing that the software fulfils each of its requirements.Internal Quality or structural quality relates to how well the software wasproduced. The International Organization for Standardization (ISO) has published standards relating to software quality. ISO/IEC 25010:2011 describes software quality in terms of eight characteristics namely: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability and portability [5]. For example; an online LMS, awebapplication,external quality includes features such as accessibility,browser compatibility, usability, and privacy.During the initial requirements of gathering sessions of the software development process where the client requires and demonstrate their expectationsaccessibility features, browsers types, and privacy features to be identified.Accessibility and browser compatibility are design features[6][7]. Usability is ensured by applying various testing, both automated tools and manual methods may be suggested by the clients. For an online LMS, a web application, internal quality include features such as functionality, usability, reliability, efficiency, maintainability and portability and a set of sub-characteristics per each characteristic are to be identified and can be measured and also evaluated by static attributes of documents such as specification of requirements, architecture of the system model, system designs, source code etc.[8].

ISO 9126 Model

Fig-1 shows a view of the relationship between internal, external and quality in use attributes[11,[12]. The ISO 9126 model [13] was based on the McCall and Boehm models. The model has two main aspects consisting of "the attributes of internal and external quality" and "the quality in use attributes". Internal quality attributes are referred to the system properties that can be evaluated without executing. External quality refers to the system properties that can be in its execution. For example, for an online LMS, these properties are experienced by users when the system is in operation

and also during maintenance. The quality in use aspects are referred to the effectiveness of the LMS, productivity, and security offered to the applications and satisfaction of users.

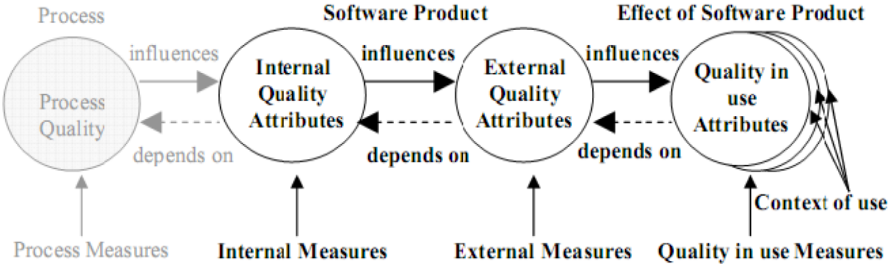

Fig-1: Quality in the lifecycle ISO 9126

III. SOFTWARE QUALITY FACTORS

McCall has written very widely on the subject of software quality. In a joint paper, McCall and Cavano (1978) identified three general types of requirements – either stemming from the customer, or from other sources – that impinge on software product quality; these types still stand today[9][10]. The types are product operation requirements, product revision requirements and product transition requirements. In addition, McCall described which attributes of a software product, or software quality factors (SQFs) as he termed them, are affected by these three types of requirements.

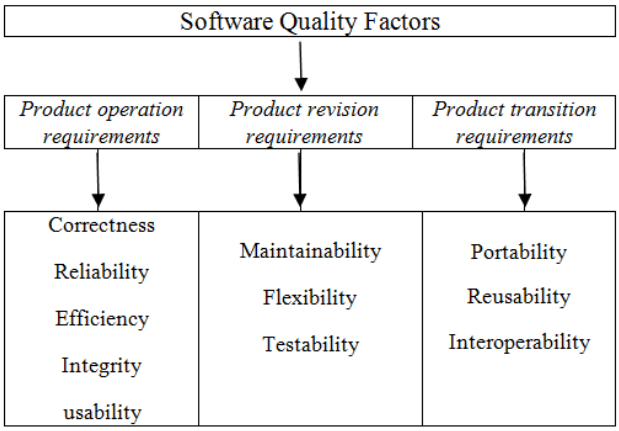

Fig-2: Software Quality Factors

A. Product operation requirements:

Concern how the product will be used. SQFs that are affected by product operation requirements are correctness, reliability, efficiency, integrity and usability.

- **Correctness:** determines how well a system fulfils the customer's overall objectives, the software system is expected to satisfy its specification
- **Reliability:** determines the likelihood with which a system can be expected to perform its intended function
- **Efficiency:** determines the level of computing resources (including time) required by a system to perform its function, that is how well it runs on the customer's hardware
- **Integrity:** refers to extent to which access to software or secure data by unauthorized persons. Due to the excessive use of e-commerce transactions integrity plays important role in network based application.
- **Usability:** determines the effort required to learn about, to operate, to prepare input for, and to interpret the output of a system, which means software system is to be easy to use for human user.

B. Product revision requirements:

Concern how the product will be changed. SQFs that are affected by product revision requirements are maintainability, flexibility and testability.

- **Maintainability**: refers to easily and inexpensively the maintenance tasks can be performed. Maintainability is the ability of the system to undergo changes with a degree of ease. These changes could impact components, services, features, and interfaces when adding or changing the application's functionality in order to fix errors, or to meet new business requirements.
- **Testability:** means that the ability to verify requirements. Testability is a measure of how well system or components allow you to create test criteria and execute tests to determine if the criteria are met.
- **Flexibility:** determines the effort required to modify an operational system, that is how easily the system can be changed while in service

C. Product transition requirements:

Are requirements relating to how the product will be modified for different operating environments. SQFs that are affected by product transition requirements are portability, reusability and interoperability.

- **Portability:**determines the effort required to transfer the system from one hardware platform and/or software environment to another, that is how easily the system can be used on another machine should the customer change their platform or should other customers require it
- **Reusability**:determines the extent to which a system (or system component) can be reused in other applications, how easy it is to reuse some of the software to make future developments more cost-effective.
- **Interoperability**: is the ability of a system or different systems to operate successfully by communicating and exchanging information with other external systems written and run by external parties.

For a particular development, some of the SQFs may be of more relevance than others[14][15]. For example, an online LMS, web server will probably require high efficiency due to the excessive amounts of course materials and assignments uploads and downloads [16]. High integrity is expected, to ensure integrity of the uploaded assignments and the grades, a safety-critical machine controller is required to ensure the problem of loss of students' data. On the other hand, the server requires accurate and highly reliable performance, of course, with expected efficiency.Software product quality attributes can be summarized into six characteristics: functionality, reliability, usability, efficiency, maintainability and portability.

IV. PRIMARY SOFTWARE QUALITY FACTORS

Software quality metrics focuses on the quality aspects of process, product and project. The project parameters such as software developers' knowledge expertise, the project schedule, the size, complexity (lines of code increase complexity increases, and the organization structure affects the quality of developed product[17][18]. For everyday software products, Tom Gilb (1988) has identified correctness, integrity, maintainability and usability as being the primary SQFs. If we wish to be able to measure the quality of a typical software product, we need some way of assessing the levels of these four attributes.

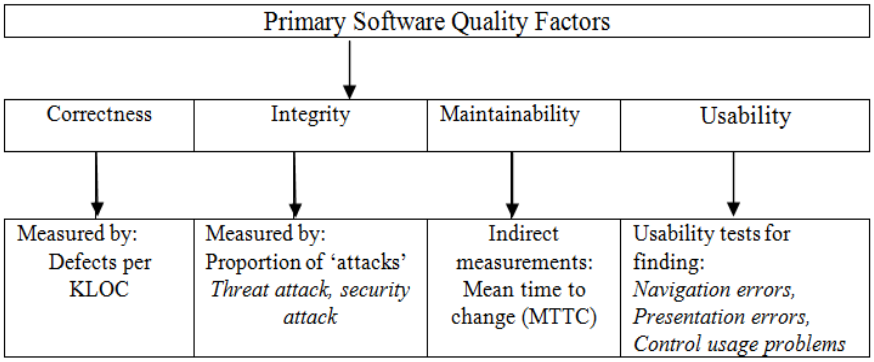

Fig 3: Primary Software Quality Factors and their measurements'

Correctness

The oldest metric for software projects is that of "lines of code" (LOC). A popular measure for assessing correctness is defects per thousand lines of code (defects per KLOC), where a defect is defined as a verified lack of conformance to requirements. Software quality is measured in terms of "defects per KLOC" where "K" was the symbol for 1000 lines of code.

For example, defects are reported by a user of the online LMS, the system has been released for general use, and are counted over a standard period of time; say one month.

Integrity

Integrity is measured by considering the proportion of 'attacks' on a product as opposed to intended uses. Some examples of attacks are: "denial of service" - in which a legitimate user is prevented from accessing a service to which they are entitled, "accidental attacks" - in which failures of the product or its users are responsible for the attack, as, for instance, when a previous privileged user doesn't log out properly from a terminal, and"input attacks" - where data is chosen to cause an application to function incorrectly. For a particular type of attack, threatattack is defined as the likelihood that an attack of this type will occur within a given time and securityattack is defined as the likelihood that an attack of this type will be repelled. For each type of attack, both threatattack and securityattack can be given values between 0 and 1, which can be calculated from historical data, such as that contained in a log file. The integrity against attacks of this type, integrityattack, is defined to be[19].

$integrity_{attack}$ =1 -$threat_{attack}$ (1 -$security_{attack}$) The integrity of the software product is defined to be the sum, over all attack types, of $integrity_{attack}$: integrity = $\sum integrity_{attack}$

For example, for an online LMS, if there are n different types of attacks, then integrity will be a value between 0 and n; the higher the value, the better the integrity.

Maintainability

Maintainabilityrefers to how easily and inexpensively the maintenance tasks can be performed. Maintainability is the ability of the system to undergo changes with a degree of ease. For example, for an online LMS, these changes could

impact components, services, features, and interfaces when adding or changing the application's functionality in order to fix errors, or to meet new academic requirements. There is no way to measure maintainability directly and it must be measured indirectly. A simple measure is "Mean Time To Change (MTTC)", which is the average of the times it takes to analyze a bug report, design an appropriate modification, implement the change, test it and distribute the change to all users. The lower the MTTC the more maintainable the software product is.

Usability

The software system is to be easy to use for human user. Normally client much emphasis on the user interfaces of software system. Usability can be ensured by adopting usability testing methods. Usability testingaims to ensure that the application is easy to use [20][21]. Checks for clearinstructions, simple and consistent navigation, fonts and colors, site map and searchtools.Usability tests are conducted to identify any design inconsistencies or usabilityissues in the user interface and content areas of a website's pages [22].

For example, for an online LMS, potential sources of error may include such as "Navigation errors" – a user fails to locate functions, takes toomany keystrokes to complete a task, or fails to follow therecommended sequence of web pages, "Presentation errors" – failure to locate important information onthe page such as 'assignment submission', failure to select the correct action due to labeling ambiguities.Control usage problems – incorrect use of buttons such as use of back button symbol instead of forward button symbol, tabs andtoolbars, or failure to enter data into the correct field such as assignment type.

V. QUALITY MANAGEMENT PROCESS – A PROPOSED MODEL IN THE DEVELOPMENT OF A WEB APPLICATION

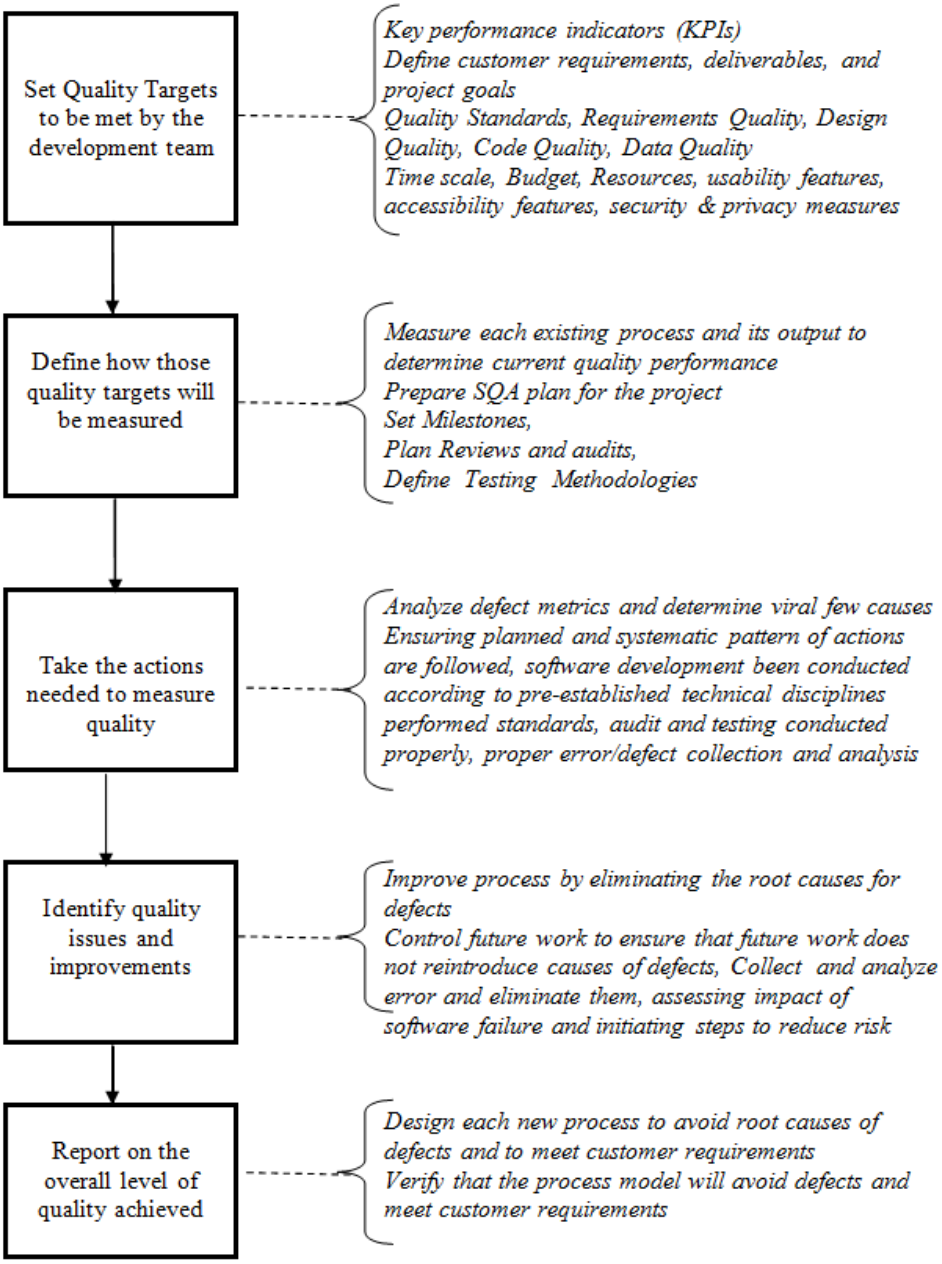

Fig-4: Quality Management Process – A proposed Model

Quality Management Process -Analysis of the development of an online Learning Management System (LMS)

Set Quality Targets to be met by the development team

The academic institution should define quality of their onlineLMS online website to match the students' needs. This include: identify appropriate quality standards,identify functional and non-functional requirements of the system. They should identify the students' academic requirements from the system; identify an interface design, login screen to enter the authentication details to ensure security, set quality standards for the data, set time scale for the completion of the development ,identify budget, identify resources, define usability and accessibility features.

Outcome: The development team will be trained with a definition of quality of the system and the importance of developing a system fits within the given time, resources and budget by ensuring quality.

Define how those quality targets will be measured

The academic institution should identify and define quality measurements and implement them to ensure the degree to which the goals and objectives are being met. Define and implement quality metrics of their online LMS to reduce defects during development. Set measurement criteria for each process and its output to determine current quality performance. The institution should prepare SQA plan for the LMS development project. The institution can prepare a Gantt chart to set milestones for activities and plan review and audit session.The institution can define appropriate testing methodologies for each development stage. Data collection process and recording mechanisms are to be defined.

Outcome: Definition of these metrics to measure quality targets of the system will keep quality as priority for the entireteam during development stages.

Take the actions needed to measure quality

The academic institution should define the software development team and individual goals to include quality of their online LMS development.The development team members perform according to their incentives, making quality improvement part of their goals reinforces desirable behavior. Implement procedures to make sure that all functional and non-functional requirements of the LMS are met. Include mechanisms to measure that the LMS project is meeting the milestones according to the schedule and within budget. Implement proper testing methods to ensure correctness, interface design such as navigation of the pages, security, accessibility, usability, performance and compatibility of the LMS. Also need to ensure the integrity of the server. Automated or manual testing methods can be adopted according to the choice of the institution by studying the pros and cons of both methods.

Outcome: Actions taken to measure quality factors of the system will keep quality as priority for the entireteam during development stages and also in the developed system.

Identify quality issues and improvements

The academic institution should assign relevant roles for quality assurance officers and the concerned managers to design the appropriate quality metrics of their online LMS and identify the appropriate standards for the web development. This will reduce the risks of failure during development and implementation. Also it is important to identify the issues to be faced in each development stages and implementation, for example, lack of resources such as domain expertise, lack of technical facilities etc. Also timely updates of project schedule will avoid expected risks. The team can optimize the use of testing methods; they can focus on the tests and implement according to the priority.

Outcome: *Actions taken to identify quality issues and improvements of the system will help the development team to complete development stages by avoiding expected risks.*

Report on the overall level of quality achieved

The academic institution should have proper mechanism to produce reports in the development stage of their online LMS. Measurements and reports are part of a system because they provide current, operational, as well as historical information of the LMS and its development stages and methodology. Report on data, such as a data dictionary, ofthe software product to be implemented to be retained. This will help infurther data analysis and thus will lead to improvements for future enhancement of the LMS as well as itsprocess management.

Outcome: *Report on the overall level of achieved qualityof the system will help the organization to help the future enhancement and maintenance the developed system.*

VI. IMPACT OF QUALITY MANAGEMENT IN SOFTWARE DEVELOPMENT

A fully documented quality assurance management system will ensure that the important requirements are met both by the clients and the software development organizations. Such organizations should ensure clients' satisfaction in delivering the desired quality product and service consistently meeting their needs and expectations. To achieve this confidence, organizationsshould meet both internal and external quality measures. They should be able to deliver the product at an optimum cost with efficient use of the available resources such as human, technology and information by ensuring the quality of the product. In this paper, the researchers used an example of an online LMS as a software product to show the impact of quality in software development process. By ensuring the need of quality during the early stages of development, the development team will be trained with a definition of quality, will keep quality as priority by knowing the definition of the quality metrics, actions taken to measure quality factors will ensure the quality, actions taken to identify quality issues and improvements will help to avoid expected risks. The reports produced on achieved quality in relevant stageswill help the organization to help the future enhancement and maintenance the developed system.

VII. CONCLUSION

Software quality factors can be measured and used to predict or indirectly measure the software quality. Software quality management activities and the measurements implemented in the early stages of software engineering development phases will help the software development teamto gain the understanding of software system to be developed as well as the importance of the quality assurance of the system. Recent studies show that the software development organizationsare implementing the SQA component to produce quality software. By following the quality standards the development organizations will be able to satisfy all the requirements of the clients.During the software development stages, as part of Quality Management Process, any quality issues are identified and resolved quickly by the development team.Product operation requirements must be ensured by applying appropriate techniques to measure the quality factors;"Correctness, Reliability, Efficiency, Integrity and Usability". Product revision requirements must be ensured by applying appropriate techniques to measure the quality factors;"Maintainability, Testability and Flexibility". Product transition must be ensured by applying appropriate techniques to measure the quality factors; "Portability, Reusability and Interoperability". Primary Quality factors Correctness, Integrity, Maintainability and Usability must be ensured by implementing appropriate manual or automated measuring techniques.

REFERENCES

[1] Analysis and Evaluation of Quality Metrics in Software Engineering, ZuhabGafoorZuhabGafoor Dand1, Prof. HemlataVasishtha, International Journal of Advanced Research in Computer and Communication EngineeringVol. 4, Issue 4, April 2015

[2] Al-Qutaish, Rafa E. (2010). "Quality Models in Software Engineering Literature: An Analytical andComparative Study," Journal of American Science, Vol. 6(3), 166- 175

[3] Anusha K, Ankita Sri, A Bird's Eye View of Software Quality Metrics, International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online) Vol. 3, Issue 2, pp: (1120-1124), Month: April - June 2015

[4] Ashwin B. Tomar1 and Dr.Vilas. M. Thakare, A SYSTEMATIC STUDY OF SOFTWAREQUALITY MODELS, International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.4, October 2011

[5] C. SenthilMurugan S. Prakasam Ph.D., A Literal Review of Software Quality Assurance, *International Journal of Computer Applications (0975 – 8887) Volume 78 – No.8, September 2013* ,ISO/IEC IS 9126-1. (2001). Software Engineering - Product Quality – Part 1: Quality Model. International Organization for Standarization, Geneva, Switzerland

[6] Dr. Venkata Surya Narayana Tinnaluri, An Approach of Software Quality Management, Imperial Journal of Interdisciplinary Research (IJIR) Vol-2, Issue-7, 2016 ISSN: 2454-1362

[7] Dubey, S.K &Soumi Ghosh & Ajay Rana. (2012). "Comparison of Software Quality Models: AnAnalytical Approach," International Journal of Emerging Technology and Advanced Engineering,Volume 2, Issue 2, pp 111-119

[8] Ghayathri J &Priya E. M. (2013) "Software Quality Models: A Comparative Study," InternationalJournal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE) ,Volume 2, Issue 1, pp 42-51

[9] Galin, Daniel (2004), Software Quality Assurance – From theory to implementation, Pearson –Addison Wesley, England

[10] Huai Liu , Fei-ChingKuo , TsongYueh Chen "Teaching an End-User TestingMethodology"Software Engineering Education and Training (CSEE&T), 2010 23rd IEEE Conference on9-12 March 2010

[11] ISO/IEC TR 9126-2. (2003). Software Engineering - Product Quality - Part 2: External Metrics.International Organization for Standardization, Geneva, Switzerland

[12] ISO/IEC TR 9126-3. (2003): Software Engineering - Product Quality - Part 3: Internal Metrics,International Organization for Standardization, Geneva, Switzerland

[13] ISO/IEC TR 9126-4. (2004): Software Engineering - Product Quality - Part 4: Quality in UseMetrics. International Organization for Standardization, Geneva, Switzerland

[14] Jani, Hajar Mat "Applying Case-Based Reasoning to software requirements specifications qualityanalysis system"Software Engineering and Data Mining (SEDM), 2010 2nd International Conferenceon 23-25 June 2010

[15] Jean Paul Paquin, Jean Couillard, and Dominique J. Ferrand, Assessing and Controlling the Quality of a Project End Product: The Earned Quality Method, *IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 47, NO. 1, FEBRUARY 2000* White Papers

[16] José P. Miguel1 , David Mauricio2 and Glen Rodríguez3, A Review of Software Quality Models for the Evaluation of Software Products, International Journal of Software Engineering & Applications (IJSEA), Vol.5, No.6, November 2014

[17] K. Hashim and A. Keshlaf, "An Approach to Sharing Solutions to Software Project Management Problems," Proc. of International Conference on Information Management and Engineering (ICIME '09), IEEE Computer Society, 2009, pp. 694-697.

[18] Kunal Jamsutkar1, Viki Patil2, P. M. Chawan3, Software Project Quality Management, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 ,Vol. 2, Issue 3, May-Jun 2012, pp.686-690

[19] Mrinal Singh Rawat, Arpita Mittal, Sanjay Kumar Dubey, Survey on Impact of Software Metrics on Software Quality,*(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 1, 2012*

[20] Mohammed Hasan. In'airat, Total Quality Management in Higher Education: A Review, International Journal of Human Resource Studies ISSN 2162-3058 2014, Vol. 4, No. 3

[21] SamadhiyaDurgesh& Wang Su-Hua & Chen Dengjie.(2010), "Quality Models: Role and Value in Software Engineering," *2nd International Conference on Software Technology and Engineering(ICSTE)*. Pp 320-324.

[22] Rakesh.L , Dr.Manoranjan Kumar Singh ,and Dr.GunaseelanDevaraj ,"Software Metrics: Some degree of Software Measuremen and Analysis ",(IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 2, 2010

