Introduction of software product and process quality aspects in the study courses of information technologies

Rudite Cevere[1], Sandra Sproge[2]

[1,2] Faculty of Information Technology, Latvia University of Agriculture, Liela str. 2, Jelgava, LV-3001, Latvia
Rudite.Cevere@llu.lv, Sandra.Sproge@llu.lv

Abstract: *Quality assurance aspects related to software product and its development processes is one of the recent topicalities in the sphere of software engineering. Various quality models and approaches to quality evaluation and improvement have been developed, standardised, and documented in any other way. These activities absolutely enable the quality improvement of products in the sphere of information technologies and, thus, simultaneously they enable the quality of business sphere applying the particular information systems. At the same time, it is generally known that the development and introduction of information systems is very labour intensive and expensive process. Hence, it is difficult to find additional time and resources for individual quality evaluation and improvement measures. On such conditions, all the possible preventive measures play an essential role. The research focuses on the possibility to include quality aspects in all the study courses of information technologies, applying quality models of intermediate products, final products, and processes as one of such measures. The impact of individual activities and intermediate results on the quality of final products is exemplified based on the metrics used in the evaluation of quality characteristics. The article outlines activities used for the evaluation of process quality and identified basing on the experience in the evaluation of the software product quality.*

Keywords: software engineering, quality assurance, process quality.

Introduction

Quality assurance aspects related to software product and its development processes have always been topical in the sphere of software engineering. Various quality models and approaches to quality evaluation and improvement have been developed, standardised, and documented in any other way (Thomas et al., 1996). These activities absolutely enable the quality improvement of products in the branch of information technologies, and thus, simultaneously they enable the quality of business sphere applying the particular information systems. The development and introduction of information systems is very time consuming and expensive process. Detection of faults and deficiencies in the end product is very expensive; therefore quality assurance in the course of the development is very important. Analysis and comparison of the quality models (Deissenboeck et al., 2009; Saini et al., 2011; Jamwal, 2010), as well as the development of new ones is still going on (Rawashdeh, Matalkah, 2006). Nevertheless, usage of quality models in practice, and implementation of quality assurance activities are tasks that are not resolved enough.

In such situation, all the possible preventive activities aimed at for decreasing the possibility of fault arising play essential role. The quality improvement activities described in the article are bases on more than ten years' experience in the position of head manager in quality assurance in large IT Company in Latvia. The Company has been ISO 9001 plus TickIt certified since 1999. In accordance to it twice a year the surveillance audit has been carried out. During these audits the progress of active development and maintenance projects has been controlled. The audits have been performed by TickIt certified auditor of the Lloid's Register who continuously performed similar audits in many countries in Europe. Therefore during the audits there was a possibility to get acquainted with actual international trends and requirements in software development. Providing IT business operations of the Company in accordance with requirements of the quality system, over these years the measures that ensured implementation of the quality requirements have been developed and realized in the projects.

In parallel with the implementation of real projects, the proposal arose from this experience how to improve teaching the questions of quality assurance and evaluation in the bachelor's study programmes of information technologies. The traditional way is to teach these topics in separate study courses like *Software Engineering*, *Software Projects Management*, *Software Product Quality*, etc. Teaching such courses is necessary, but is not enough to acquire knowledge about implementation of quality requirements. The experience from the development projects shows that the quality assurance becomes the most effective when these activities become a part of each development task. The approach how to improve the study courses of the information technologies on the bases of this experience is described in the particular research.

Materials and methods

Aim and tasks of the quality system

In accordance with the quality life cycle of software products, the first stage of quality assurance of the end product is related to quality assurance of the development processes. Traditional way of its realization is the development and introduction of the quality system of the company, for example, in accordance with ISO 9001 requirements (Stelzer et al., 1997). The development and maintenance of quality system of an organization has proved itself being a successful decision in the software development (Stevenson, Barnes, 2001). It has also demonstrated long-term experience in IT Company.
Successful quality system has two main prerequisites:

- quality system documentation provides to all staff the minimum substantive guidance as to work in a consistent, disciplined and effective manner;
- regular training is carried out with the staff, ensuring quality assurance measures as a regular part of the work.

Quality management system together with other parts of corporate management systems shall be integrated into a single corporate management system via common elements.
Students have to gain the idea on diversity of quality systems, their significance in the company, advantages and disadvantages of quality systems as well as existing or non-existing risks during their studies when mastering software development and project management requirements, methods and the best practice (for example, software engineering).

Quality assurance activities in software development processes

In a company, the prime business of which is the development of information systems and provision of various information and communication technologies services, the development processes may be defined consistent with recommendation of the ISO 12207 standard (ISO/IEC 12207, 2008). The standard prescribes groups of basic, support, and organisational processes. Basic processes (specification, design, implementation, and testing) are the processes ensuring the development of software. Four support processes – configuration management, change management, review and documentation are the most significant from the quality assurance point of view. The main characteristic of these processes lies in the fact that they are not implemented as sequence of independent operations executed by one performer. Operations necessary for implementation of processes are performed during the basic processes by adding activities required for implementation of support processes to the tasks of basic processes and providing preparation of required documents and records. Therefore, it is necessary to define clearly the interaction of each basic development process with support processes.
The observations of each audit were analysed and the measures were developed for prevention of further origination of similar notes. In such way during several years the necessary quality assurance actions were defined. Their implementation in all projects ensured successful passing of the surveillance audits constantly. The following article contains an analysis of each software product development core process interaction with the support processes, developed on a personal experience.

Requirements specification

Requirements defined during the specification process may be documented in any electronic environment, the content of which and form of documentation shall be coordinated with a customer prior to start of requirements analysis. During the audits the main attention focuses to version control of all developed and received documents, to change control of the deliverables, and to internal reviews of intermediate results and deliverables.
In order to ensure compliance with the specification process quality requirements, the interaction of specification process with four support processes should be implemented. Configuration management, change control and documentation activities are connected to an individual specification activities. Reviews are included as an additional activity in specification process (Table 1).

Table 1

Activities assuring the specification process quality

Specification operation	Related quality assurance activity	Supporting process	Performer	Notes
Collection and summarisation of source documents	Maintenance of source document versions	Configuration management	System analysts	Ensure identification and maintenance of source document versions
Preparation of intermediate results and results	Preparation of transferable documents	Documenting	System analysts	Ensure qualitative form of transferable documents

Specification operation	Related quality assurance activity	Supporting process	Performer	Notes
Preparation of intermediate results and results	Maintenance of transferable document versions	Configuration management	System analysts	Ensure identification of transferable documents
Preparation of intermediate results and results	Maintenance of transferable document changes	Change management	System analysts	Ensure maintenance of change notes of transferable documents
Preparation of intermediate results and results	Reviews of transferable documents and intermediate results	Review process	Group of developers	Ensure quality assessment of transferable documents

Design

The aim of design process is to transform system requirements within design of information system to be developed. Similar to the specification process, the main issues of the audit during the design process are version control of all developed documents, change control of the deliverables, and internal reviews of intermediate results and deliverables. Audit requirements in the design process are usually stronger by require performing design description reviews already in situations when it is not intended as deliverable, and require performing joint reviews.

Thus, during the design, similar to the specification process, it is necessary to ensure the versioning and identification of all design documents, results and intermediate results. Qualitative forms of the design documents, correct change involvement into design description, and requirements traceability to the design elements should be ensured. It is implemented by interaction of the design process with the support processes (Table 2).

Table 2

Activities assuring the design process quality

Designing operation	Related quality assurance activity	Supporting process	Performer	Notes
Preparation of intermediate results and results	Preparation of transferable documents	Documenting	System analysts	Ensure qualitative form of transferable documents
Preparation of intermediate results and results	Maintenance of transferable data item (document) versions	Configuration management	System analysts	Ensure identification of transferable items
Preparation of intermediate results and results	Maintenance of transferable document changes	Change management	System analysts	Ensure maintenance of change notes of transferable documents
Preparation of intermediate results and results	Internal reviews of transferable documents and intermediate results	Review process	Group of developers	Ensure quality of transferable data items (documents) and conformity with requirements
Preparation of intermediate results and results	General reviews of transferable documents and intermediate results	Review process	Group of developers and a customer	Ensure conformity of transferable data items (documents) with requirements

Coding and debugging

In coding process all agreed and approved documents become especially strict configuration management controlled items. Maintenance of the versions, identification and change control should be ensured for all code items. Forms of their realization differ from those of the specifications and design descriptions, but the compliance with these requirements are more stringent. The main auditable issues for coding are requirements traceability, i.e. the compliance of the developed units of the software code with the requirements and design, and compliance of the code with the requirements of a unified coding style. In particular, code reviews are strictly supervised activities. Involvement of the documentation process allows solving the problems with self-documentation of the code and realization of embedded electronic documentation (help). Interaction of the design process with the support processes is described in Table 3.

Table 3

Activities assuring the coding process quality

Coding operation	Related quality assurance activity	Supporting process	Performer	Notes
Preparation of source code data items	Documenting of source code	Documenting	Programmers	Ensure self-documenting of source code
Preparation of source code data items	Maintenance of source code data item versions	Configuration management	Programmers	Ensure identification of software data items
Preparation of executive code data items	Maintenance of executive code data item versions	Configuration management	Programmers	Ensure identification of executive code
Preparation of source code data items	Internal reviews of source code	Review process	Group of developers	Ensure quality of software and conformity with requirements
Preparation of software versions	Reviews of transferable versions	Review process	Group of developers and a customer	Ensure conformity of transferable versions with requirements

Testing

Testing process plays an essential role not only as one of the basic processes of software development but also as a significant activity for quality assurance of final results.
Considering the test document and record diversity, the audit focuses attention to related issues. One of the most popular schemes is an inspection of random problem report for traceability in both directions - from the report to initial requirements and from the report to the program code. In order to ensure these requirements configuration management and change control actions are involved into test documents and records. All necessary reviews should be carried out. Besides the common task of reviews to evaluate current object, reviews in testing should be organized for preparation of management information and for decision making about ending, modification or interruption of testing process. Configuration management has been related with versioning of the test objects, and also with test environment preparation and maintenance. Necessary interaction of the design process with the support processes for testing is described in Table 4.

Table 4

Activities assuring the testing process quality

Testing operation	Related quality assurance activity	Supporting process	Performer	Notes
Preparation of testable objects	Preparation of testing object versions	Configuration management, coding	Testers, system analysts	Ensure timely preparation of testable objects
Preparation of testing documentation	Determination of types for testing notes	Documenting	Testers, system analysts	Ensure preparation of testing records
Preparation of testing documentation	Maintenance of testing document versions	Configuration management	Testers, system analysts	Ensure maintenance of testing records
Preparation of testing notes	Determination of types for testing notes, implementation environment and identification	Configuration management, documenting	Testers, system analysts	Ensure monitoring of testing process information
Preparation of testing notes	Maintenance of testing notes	Change management	Testers, system analysts, programmers	Ensure registration of problem reports and change demands
Preparation of testing documentation and testing notes	Quality assurance of testing documentation and notes	Reviews	Project managers, quality manager, testers	Ensure quality of testing documentation and notes (hence, the entire testing process)
Organisation of testing process	Compliance monitoring of the course of testing performance	Reviews	Project managers, quality manager, testing manager	Ensure adequate procedure of testing process

Activities included in the primary processes provide software quality because maintain an orderly flow of information for all items involved in the development. However, along with the benefits, developers should always keep in mind that quality assurance activities require additional resources. It is unacceptable that this resource becomes unreasonably large. There appears the next aspect of quality assurance, which is related to

quality assessment of the process. The following section describes the approach how the quality model may be used for description of the quality of primary and support processes.

Evaluation of process quality

During the development of the quality system all processes going on in an organisation should be identified and the procedures for their implementation defined. This action results in the development of process descriptions, which form a part of quality system documentation. A description shall be developed for each process irrespective of the fact whether a process is implemented as a sequence of independent operations or as activities added to basic operation of other processes. General description of process includes process inputs, outputs, sequence of the necessary operations resulting in the transformation of input into output as well as resources and control necessary for the process implementation.

General requirements set for process description state that a process shall be described in terms of its purpose and outcomes, and in any process description the set of process outcomes shall be necessary and sufficient to achieve the purpose of the process (ISO/IEC 15504-2, 2003). Results of every process may be necessary to be used for the input of other processes.

Development and description of processes is an indispensable prerequisite for the establishment of quality system. Though, it is insufficient to really implement quality assurance. For quality system's requirements realization in the primary processes of the development the quality activities described above were defined. These activities implement measures necessary for quality assurance of the end product. However, the quality of the development processes in not guarantees that the developed final product - the software will be of good quality. It is necessary yet to take the software product quality and process quality assessment. Thus, it appears another question as to what is considered a high-quality process or product.

Realizing any theory based quality assurance activities, it is necessary to make sure that the potential benefits outweigh the investment of resources. This effect makes urgent need to assess the quality of the process itself.

In ISO 9000 standards there is not defined in detail what process can be thought of as a quality, and how to evaluate the quality of the existing or desired process. These issues are outside the ISO 9000 standard scope. However, the software product quality assessment and quality improvement issues in the field of software development are well developed (Khayami et al., 2009). On the bases on software quality models developed during many years the generalized form of quality model and evaluation techniques are coordinated and approved. They are included in standards of ISO/IEC 9126 and ISO/IEC 14598 series. This approach is further developed in standards of the SQuaRE series (group of ISO/IEC 25000 standards). Approach described in the mentioned standards defines the internal and external quality of software in accordance with quality model of the ISO 9126.

Considering, that during software development it is necessary in different ways to focus on the quality of processes and their outcomes, the approach, described in the current research includes proposal of using unique model for quality description in all cases. The model consists of hierarchically related quality characteristics and subcharacteristics. Therefore the process model also can be defined as hierarchical structure of characteristics and subcharacteristics. (Fig. 1)

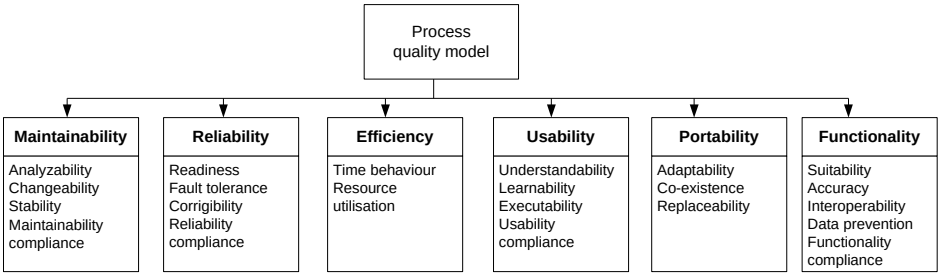

Fig. 1. **Process quality model**.

The same quality characteristics, which are determined for the software product, may also be used for quality evaluation of processes. Certainly, considering that an explanation of the notion may slightly or even greatly differ from the explanation used in the evaluation of products. The largest difference is related with the metrics applied evaluating characteristics or sub-characteristics. For example, a suitability metric "Functional implementation completeness" may be used evaluating functionality of a product. It answers the question "How complete is the implementation according to requirement specifications?" and the method of application is counting the number of missing functions detected in evaluation and comparison with the number of functions described in the requirement specifications. The use of all inputs and acquisition of all outputs will be evaluated instead of specified functions when evaluating the quality of processes.

Particular importance in the successful implementation of the quality system has rational design of the quality assurance processes. Experience gained during the surveillance audits gives a good illustration related to the review process.

Reviews are one of the activities which are very carefully checked during every surveillance audit. A review report, the preparation of which may be quite time consuming, is one of records demanded compulsory. For example, in first years of maintenance of the quality system during a number of audits projects received comments on the lack of the review report in spite that in quality system the review reporting form was designed, that includes all the information required. Therefore, in number of active projects there were carried out observations related to situation with review reports. It showed that, for example, if in a small-scope project one review is performed by 3 employees and the duration of review takes about 1.5 hours, than preparation of a review report according to the form stated by the quality system, may take additional 30-45 minutes for one employee. In addition, for the developers themselves, this neatly presented report is not required, whereas they use their records made during the review. This considerably reduces employees' motivation to perform reviews in general. After a careful analysis of information required by the review report and the information arising as the result of implementation of project tasks, it turns out that only a review register shall be established additionally. Most of information is already recorded in the environment of work flow control, a project plan, e-mail correspondence, minutes of regular meetings, and other records. This example shows necessity for the quality managers to base decisions on a certain understanding of the quality (in the example the assessment was based on the efficiency of the process). For the improvement all employees should master skills of using information of certain type, which is recorded in various places. In any case one should avoid creation a new document with a title of required information, which in fact repeatedly rewrites the existing information. These skills shall be mastered already during the studies.

Using quality model given in Fig. 1 for the quality evaluation of the process, it is desirable to organize decision-making, using the metrics defined by sample of software product quality model (Table 5).

Table 5

Time behaviour metric of the process

Metric name	**Purpose of the metrics**	**Method of application**	**Measurement, formula and data element computations**	**Interpretation of measured value**	**Input to measurement**
Documentation time	What proportion of the time do developers spend preparing the review process documentation?	Observe the reviews of various items carried out in projects. Measure the time it takes to develop the review report.	X = Td / Tr Td = time spent for documentation Tr = total time of reviews	0<= X The smaller the better.	Review report Operation report showing documentation

Results and discussion

In Latvian University of Agriculture, Faculty of Information Technology students are taught in Programming and Computer Science undergraduate programs. In the special courses of the programmes the approach developed by authors of learning the quality assurance issues in-depth gradually are being implemented. The essence of the approach is that in most special courses of information technologies along with special topics should be included the questions about the quality characteristics of the product and about how to change (to increase or reduce) the quality of the end product (Sproge and Cevere, 2012). The reviewed literature has failed to find a description of a similar approach, only a few articles have noted that there should be more attention to the issue of quality learning during the study period (Hilburn and Towhidnejad, 2000). The authors in their research have dealt with issues of how to determine in which courses and to what extent it would be desirable to include software product quality issues.

This article focuses on the acquisition of quality of the basic and support processes of software development. A possible realization of this learning in courses of bachelor's degree program is summarized in Table 6.

Metrics used for software quality characteristics and sub-characteristics may be used for acquisition of software quality indicators, in order to illustrate how particular decisions in code development allows to move towards achievement of quality requirements. For example, testability is one of the sub-characteristics of software maintainability, the influence of which may be determined in the study course by applying metrics given in Table 7. In description of the metric the question tried to answer by help of the metric is formulated, and the quality characteristic to be evaluated is pointed. Besides them, the method of data acquisition and interpretation is described.

Table 6

Quality assurance activities included in primary processes

Software development process	Study Courses	Knowledge and Skills
Specification	*Systems Modelling, Algorithms and structures*	Acquire requirements analysis and modelling tools and techniques or ways necessary for the software development which can also be used to document requirements
	Basics of Software Engineering	Provides knowledge of the organization of specification process and of quality requirements of the specification
	Software Engineering	Acquire configuration management, change control and documentation processes, the practical work contains Reviews of requirements specification
	Programming, Database Technology	In the execution of practical work and course work to follow the minimum requirements of the support process are required (document identification, change control)
Design	*Programming, Database Technology*	The implementation of the practical work should comply with requirements of the configuration management, change management and documentation process, the review of design should be implemented
	Systems Modelling, Algorithms and structures	In practical work review of design is carried out
Coding	*Programming Fundamentals, Programming in Windows environment, Database Application Programming, WWW technologies Programming*	Acquire form and content of the mandatory and recommended comments of source code, maintenance of code changes, and provision of the testability and maintainability of the program. Implement code review (source code data retrieval operator reviews, class libraries reviews, deliverable versions reviews)

Table 7

Testability metrics (ISO 9126, 2003)

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Input to measurement
Completeness of built-in test function	How complete is the built-in test capability.	Count the number of implemented built-in test functions as specified and compare it to the number of built-in test functions in the requirements.	X=A/B A=Number of implemented built-in test function as specified confirmed in review B=Number of built-in test function required	0 <= X <= 1 The closer to 1, the more complete.	A comes from review document B comes from requirements or design document
Autonomy of testability	How independently can the software be tested?	Count the number of dependencies on other systems for testing that has been simulated with stubs and compares it with the total number of test dependencies on other systems.	X=A/B A=Number of dependencies on other systems for testing that have been simulated with stubs B= Total number of test dependencies on other systems	0 <= X <= 1 The closer to 1, the better.	A comes from review document B comes from requirements or design document

The relationship between processes execution and quality models are reflected in Fig. 2. Reviews of different levels are very important during execution of primary processes. It is useful to organize them on the basis of the uniform quality model. The approach, used for quality assessment, which provides metrics purpose formulation in form of a question, makes it possible to prepare checklists which guide each review.

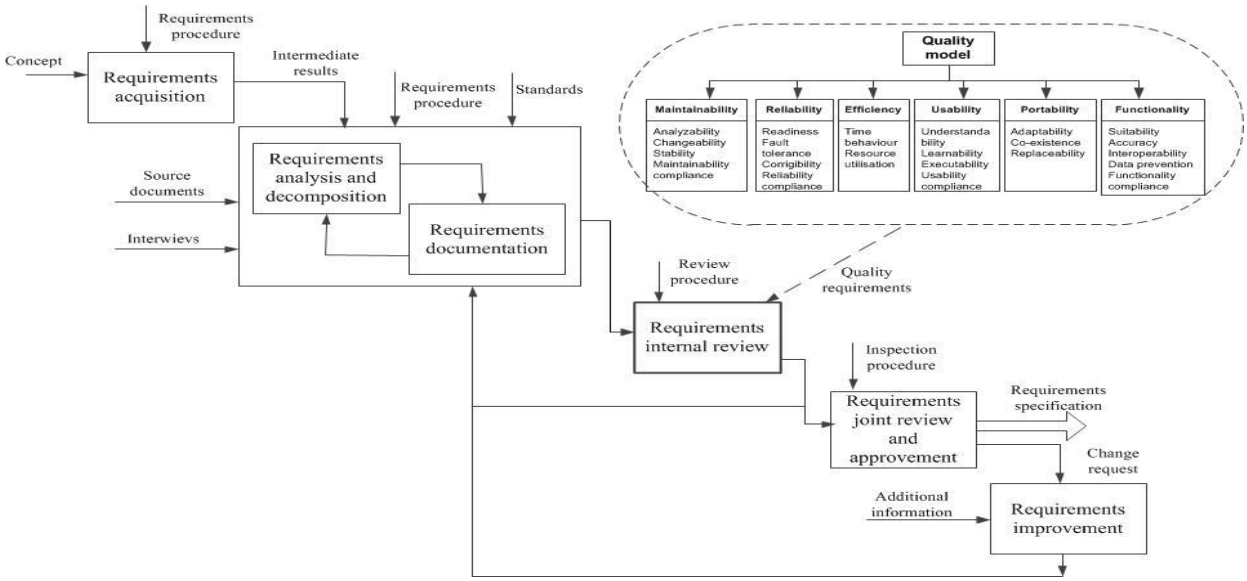

Fig. 2. **Integration of the processes description and the quality model.**

Conclusion

Preventive actions play the most significant role in the sphere of quality improvement, unlike correction of admitted mistakes. During the project preparation and implementation, much attention is being paid to the staff qualification and necessary additional training. Though, it is usually possible to master only programming languages or development environments necessary for the implementation of the particular project due to time restrictions. Therefore, it is essential already during the studies to include quality assurance aspects of corresponding processes or results, their importance and quality improvement activities in the content of field professional specialisation courses when mastering study programmes in information technologies. It is recommended to ground the studying of these quality aspects on a single quality model of a product (end product or intermediate product of software development) and quality model of processes.

During the development and maintenance of software, it is necessary to implement different quality assurance activities, which together create configuration management, change management, documentation, and review processes. These processes are the most important support processes for software development. Simultaneously, the implementation of support process activities shall be included into the structure of every basic process of software development by adding these activities as supplement to the activities of basic process.

To teach basics of quality assurance in training of specialists in information technologies, the study courses shall be developed by analogy and include compulsory and advisable quality activities of each course, outlining their impact on the quality of a product. Quality models of software product and processes may be applied for characterisation of the impact as well as metrics recommended for the evaluation.

The Faculty of Information Technologies of Latvia University of Agriculture has started such improvement of study courses in two bachelor's study programmes "Computer Management and Computer Science" and "Programming" as well as in a master's study programme "Information Technologies".

Acknowledgements

This research has been supported by the European Social Fund within the project "Support for the Implementation of Doctoral Studies at Latvia University of Agriculture". Agreement No. 2009/0180/1DP/1.1.2.1.2/09/IPIA/VIAA/017.

References

Al-Qutaish, R., 2010. Quality Models in Software Engineering Literature: An Analytical and Comparative Study. *Journal of American Science*, 6(3), 166-175.

Deissenboeck, F., Juergens, E., Lochmann, K., Wagner, S., 2009. Software quality models: Purposes, usage scenarios and requirements. In *7th International Workshop on Software Quality* (WoSQ '09). IEEE Computer Society.

Hilburn, T. and Towhidnejad, M., 2000. Software Quality: A Curriculum Postscript? In: *Proceedings of the 31st SIGCSE technical symposium on Computer science education.* New York, 167-171

ISO/IEC 12207:2008. Information technology – Software life cycle processes. International Organization for Standardization.

ISO/IEC 14598-2: 2000. Software engineering -- Product evaluation - Part 2: Planning and Management. International Organization for Standardization.

ISO/IEC 15504-2:2003. Information technology – Process assessment –Part 2: Performing an assessment. International Organization for Standardization.

ISO/IEC 25030:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality Requirements. International Organization for Standardization.

ISO/IEC 9126-1:2001. Software Engineering – Product Quality – Part 1: Quality Model. International Organization for Standardization.

ISO/IEC 9126-3:2003. Software engineering – Product quality – Part 3: Internal metrics. International Organization for Standardization.

Jamwal, D.,2010. Analysis of Software Quality Models for Organizations. *International Journal of Latest Trends in Computing*,1(2), 19-23

Khayami, R., Towhidi, A., Ziarati, K., 2009. The Analytical Comparison of Qualitative Models of Software Systems. *World Applied Sciences Journal*, 6 (1), 1-6.

Rawashdeh, A. and Matalkah, B., 2006. A New Software Quality Model for Evaluating COTS Components. *Journal of Computer Science* **2**(4), 2006, 373-381.

Saini, R, Dubey, R., Rana, A., 2011. Analytical Study of Maintainability Models for Quality Evaluation. *Indian Journal of Computer Science and Engineering*, 2(3), 449-454.

Sproge, S. and Cevere, R., 2012. Position and role of feedback in the software quality life cycle. *Quality Issues and Insights in the 21st Century*, 1, 29-40.

Stelzer, D., Mellis, W., Herzwurm, G., 1997. A critical look at ISO 9000 for software quality management, *Software Quality Journal*, 6, 65–79.

Stevenson, T. and Barnes, F., 2001. Fourteen years of ISO 9000: impact, criticisms, costs and benefits *Business Horizons*, 44(3) (2001), 45-51.

The TickIT Guide. A Guide to Software Quality Management System Construction and Certification to ISO 9001:2000. BSI British Standards.

Thomas, S., Hurley, S., Barnes, D., 1996. Looking for the human factors in software quality management, *Software Engineering: Education and Practice*, 474-480.