POSITION AND ROLE OF FEEDBACK IN THE SOFTWARE QUALITY LIFE CYCLE

Sandra Sproge, Rudite Cevere
Latvia University of Agriculture, Jelgava, Latvia
E-mail: Sandra.Sproge@llu.lv, Rudite.Cevere@llu.lv

Abstract

The current research relates with findings for a new possibility to improve quality of software development end product. Not any of the software development stages can be imagined without human participation. Therefore the knowledge of the development staff has an essential impact on the software quality. Teaching of information technologies specialists the impact of the particular development activity on software quality simultaneously with the acquiring programming, date base formation and other development skills is a possibility to improve the quality of particular products already during their development stage. Such teaching requires modifying special courses of the bachelor's study programmes by adding knowledge from aspects of quality assurance. Teaching staff of these courses should already perfect their knowledge in software quality. The research is devoted to development of the proposals for a higher education institution for teaching information technology specialists. Inclusion the teaching of the quality model of the end product into the study programme is described. The main idea is based on the teaching to the students the total software quality assurance aspects in parallel with uniform discussion of the quality features of the end product in all special study courses and the impressionability of them doing the particular development activities. The research contains a version of preparation work for implementation of the quality model's acquisition in the study courses. The adjacency matrix of software quality characteristics and study courses gives an idea on which particular characteristics may be influenced in every study course. The feedback data from software developers, in turn, identify the typical quality requirements of the customers, and the issues to be taught definitely in every study course to ensure and to increase the quality of software under development.

Key words: *software quality, quality model, quality characteristic, study programme.*

Introduction

Software quality life cycle includes three large stages – quality of development processes, software internal/external quality, and software quality in use. Software is a product requiring a time-consuming and expensive development and introduction process. Therefore, possible preventive activities of all types play an essential role apart from the quality evaluation and improvement measures undertaken during the development process. The improvement and advancement of staff quality, which starts from the study period, is one of such activities. Already during the first stage of training staff involved in software development, new specialists should acquire knowledge on the software quality life cycle and skills to influence the quality of end-product in each particular stage of the product.

Problem of Research

Software product is an abstract product where different users - customers, project managers, developers, and end users – advance their requirements towards the quality of software product. It is paramount to perceive that every stakeholder has its own definite understanding on the quality of end-product. Software development includes two key quality definitions:

1. Product is qualitative if it conforms to the requirements set previously (Crosby, 1979);
2. Product is qualitative if it matches a degree up to which a user's needs or expectations are satisfied (Deming, 1988).

The impact of software is highly relevant considering its extensive application in multiform spheres of life. Therefore, software quality issues have already been outstanding from the initial stage of software development. One of the approaches for solution of quality assurance aspects is related with the quality of processes. This approach is based on larger expectations and probability that ensuring the development processes of higher quality it is possible to obtain a more qualitative end-product. All software life cycle processes are viewed and managed, both those directly related with the development, and those related with the staff qualification and training. This has resulted in the development and wide practical application of several approaches like the ISO 9000, CMMI, Total Quality Management etc. to the quality assurance and evaluation of an organisational operation (Paulk, 1995; Abrahamsson, 1999, Baldassarre & Caivano & Pino & Piattini & Visaggio, 2012). However, the meaning of qualitative software is not specified in details when speaking about the quality of an end-product under the mentioned view.

The attempts to describe the software quality itself based on the notion of a quality model have also progressed in parallel (Al-Qutaish, 2010; Ortega & Perez & Rojas, 2003). These multiform models have been analysed and summarised for several years resulting in the elaboration of the ISO 9126 and the ISO 15485 series of standards. The most essential issues of the summary refer to the definition of software product quality notion in the life cycle and the development of a hierarchical general internal and external quality model as well as a quality model for software in use (ISO/IEC 9126, 2001). Further development of quality models characterises with an extensive application of the quality model defined in the mentioned standards by tailoring it to various software and evaluation situations (Behkamal & Kahani & Akbari, 2009; Fam & Luo & Wu & Fu, 2010). The theoretical understanding on model and quality requirements also continues to develop; this understanding is documented in new more comprehensive series of quality standards SQuaRE, ISO/IEE 25000.

Software quality life cycle starts with the quality of processes and ends with the quality in use. Speaking about quality, it shall be considered that human participation is unavoidable in every process and every stage of software life cycle. Therefore, specialists involved in the development of software product primarily ensure its quality. Hence, ensuring the staff qualification is one of the aspects for the assurance of software product quality. For example, the role of specialists in the software product quality is emphasised by Tervonen and Kerola (Tervonen & Kerola, 1998) who recommend three cycles for improving understanding on software quality:

1. Different perspectives;
2. Quality as individual experience;
3. Software quality as team/organisational knowledge creation.

The first cycle introduced the five perspectives as a source of difficulties in understanding quality. The second cycle introduced different learning styles as a problem of understanding. If the first and third cycles mainly focus on the company developing software and company operation, then the second cycle particularly relates closely with the study process.

Hilburn and Towhidnejad (Hilburn & Towhidnejad, 2000) underline that the most academic study programmes offered by higher education institutions concentrate on software development languages, technologies, tools, and environment. The emphasis is being laid on the process how the product is developed and not on the aspect how to create a qualitative product. During the studies, when training new specialists in information technologies, the emphasis should be initially laid on software quality and it should be discussed in more details within all the following study courses related with software development and introduction.

In turn, speaking about the training quality of specialists in information technologies, it shall be noted that nowadays the quality assurance issues in the sphere of education have become especially topical. Quality systems for the evaluation and improvement of education are being developed and certified also in this sphere (Thonhauser & Passmore, 2006; Lundquist, 1999). The application of the EFQM model in education (Osseo-Asare & Longbottom & Murphy, 2005; Rosa & Amaral, 2007) has gained broad popularity as well as the application of Education Criteria for Performance Excellence offered by the Malcom Baldrige National Quality Award.

The main problems in the entire diversity of quality assurance approaches and models are caused by several aspects:

1. The diversity of approaches, models and quality notions is extremely large in issues related with the quality of software end-product;
2. Speaking about the quality of education including also the quality of training specialists in information technologies, there are no concrete recommendations for defining and evaluation of exactly the quality of study courses and study programmes, namely, the quality of study content;
3. No detailed and verified in practice techniques and recommendations based on the experience gained in software development and study process organisation are available as well as there are no techniques and recommendations for the implementation of quality assurance in practical operation.

Research Focus

Specialists in software development are initially trained during studies through the acquisition of a study programme related with information technologies. Predominantly, the main objective of study programmes is the provision of knowledge on software product development tools and techniques. The development processes and their organisation, management and quality assurance are viewed in parallel. Thereby, not every individual study course examines the impact of the action to be acquired on the quality of end-product. Long-term experience both in the development of information technology systems and the study programmes of this sphere allows the authors of the paper to raise a hypothesis that it would be necessary to provide profound knowledge to the future employees of information technology sphere already during the studies on:

- The possibilities to evaluate the quality of end-product;
- The influence caused by different solutions chosen and decisions made in the development process;
- The possibilities to improve the quality.

The present research concentrates on two main questions: 1) how to determine which activities in the software quality life cycle leave an essential impact on the quality of product; and 2) how to relate software (and the information system in general) development techniques with the software quality assurance already during the studies.

The feedback from the evaluation of the software end-product and by-products in IT companies is chosen as the main source of information for the improvement of the study course content. Recommendations for inclusion of quality assurance activities in the study courses are made based on the feedback results and through a targeted involvement of the academic staff.

Methodology of Research

General Background of Research

The ISO/IEC 9126 standard examines the quality assurance and improvement of software product during its development from the aspect of process quality, product quality, and quality in use. The company personnel and its qualification influence both the development process and the quality of product. The study process is the primary source for the qualification of personnel, when students master the respective study programme in information technologies. Hence, basic knowledge on the quality of product, quality evaluation and improvement shall be acquired during the study process.

A hierarchic quality model is offered for the product quality and quality in use evaluation in the ISO/IEC standard. The transformation of this quality model allowed obtaining a universal quality base model (Figure 1) that may be applied both for the quality evaluation of software product itself and its development processes as well as the quality evaluation of the study programme content (Čevere & Sproģe, 2010, 2011).

A two-level hierarchic structure was created for this quality model:

- Level 1 – basic characteristics of quality;
- Level 2 – sub-characteristics of quality characteristics that are divided into: basic sub-characteristics; additional sub-characteristics, and optional sub-characteristics.

Depending on the items evaluated by means of this model, both the number of quality characteristics and sub-characteristics may change as well as their description (name) and meaning. Metrics applied for the evaluation of the respective characteristics will also differ significantly. For example, if a suitability metric "Functional implementation completeness" is used for the evaluation of product functionality; this metric answers the question "How complete is the implementation according to the requirement specifications?" The method of application is counting the number of missing functions detected in evaluation in comparison to the number of functions described in the requirement specifications. In case of quality evaluation of processes, the use of all inputs and acquisition of outputs will be evaluated instead of the specified functions. In turn, the quality evaluation of study programme content will encompass the evaluation of implementation of themes offered by study courses during the study process.

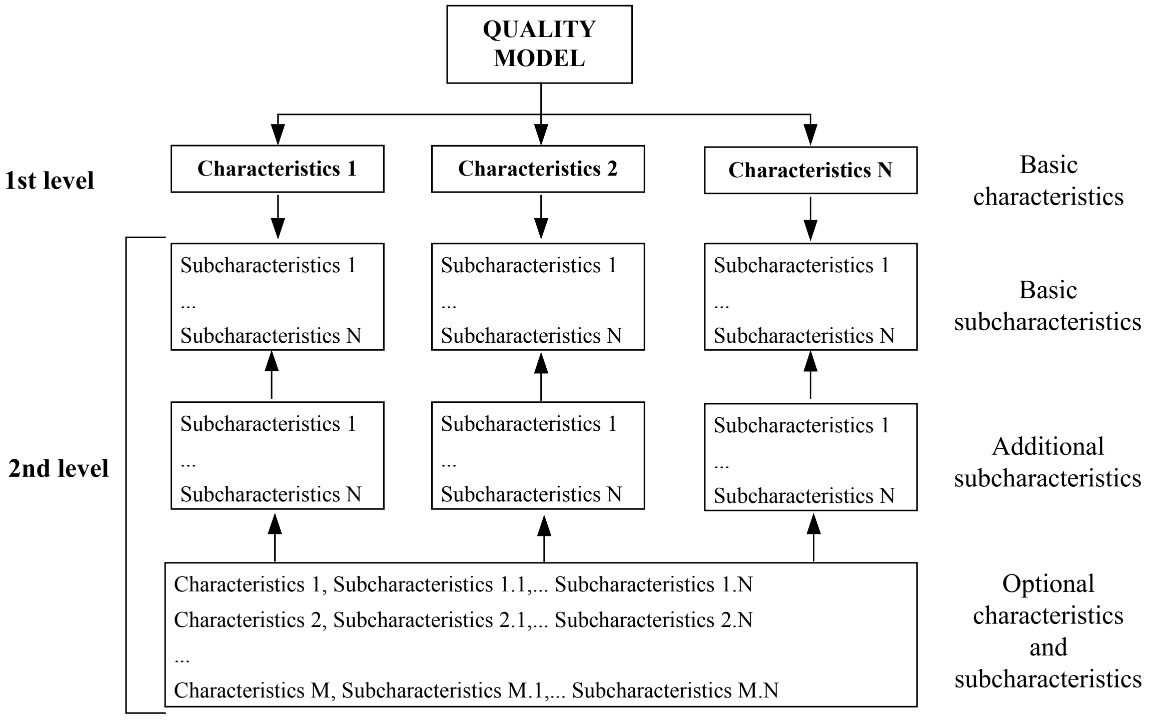

Figure 1: Base Quality model (Sproge & Cevere 2012).

The base model applies the same quality characteristics and sub-characteristics for the evaluation of a product quality as defined by the quality model ISO-9126 (ISO/IEC 9126, 2001):

- Functionality: *Suitability, Accuracy, Interoperability, Security, Functionality Compliance;*

- Reliability: *Maturity, Fault Tolerance, Recoverability, Reliability Compliance;*
- Usability: *Understandability, Learnability, Operability, Attractiveness, Usability Compliance;*
- Efficiency: *Time Behaviour, Resource Utilisation, Efficiency Compliance;*
- Maintainability: *Analysability, Changeability, Stability, Testability, Maintainability Compliance;*
- Portability: *Adaptability, Installability, Co-Existence, Replaceability, Portability Compliance.*

Quality in use is determined by a set of four characteristics - effectiveness, productivity, safety, and satisfaction.

Each of these quality characteristics through the application of certain metrics may be evaluated for a product under the development or a ready for use product. Frequently, the improvement of an individual quality characteristic may require an essential configuration of product, which, in turn, results in overrun of development time-limit and project costs. Therefore, it is very significant to know factors that directly affect a characteristic of a certain quality and what aspects have to be considered to achieve the respective quality characteristic already at the initial stage of a product development.

Software Quality Characteristics in Study Courses

The development and application of quality systems focus on the assumption that there are larger expectations and probability for obtaining a more qualitative end-product by ensuring the development processes of higher quality (ISO9001, EFQM). A similar assumption states that if training of specialists involved in software development were organised so that the impact of activities on software quality is viewed simultaneously with the performance of activities, then the quality of certain products would also be essentially increased during their development process. It is impossible to devote additional time and resources for understanding and evaluation of individual quality characteristics and their mutual impact when working on a particular project. The study process, in turn, is the first step towards training of new specialists. Supplementing the content of present study courses included in the study programmes, it would be possible to master software product quality significance, its impact and evaluation possibilities in all study courses related with software development (Figure 2). Hence, it is a requisite to determine the particular quality characteristics which are preferable for mastering in particular study courses.

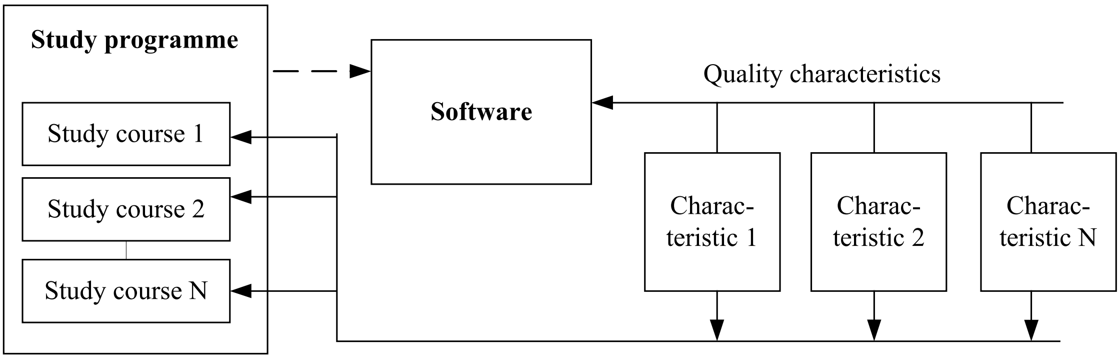

Figure 2: The interrelation between the quality of study programme and software.

Instrument and Procedures

The interrelation between a software product quality characteristic and a study course is determined in three ways:

1. Theoretical background;
2. Opinion of the academic staff;
3. Opinion of software developers.

Theoretical Background

Study courses are selected and a matrix is created on their interrelation based on the metrics of quality characteristics prescribed by the ISO/IEC 9126 standard and the content of study courses to be mastered. The matrix is aimed to identify which characteristics should be mastered in more details in which study courses (for example, impact on software high operation speed and maintainability).

Figure 3 outlines the sequence of judgements starting from the quality in use and ending with the study courses. The quality in use model names satisfaction as one of the most significant software quality characteristics. On the level of metrics, "satisfaction metrics assess the user's attitude towards the use of product in a specified context of use. Satisfaction is influenced by the user's perception of properties of the software product (such as those measured by external metrics)" (ISO 9126-4). The most appropriate external and internal quality characteristics are Usability and Maintainability. The analysis of metrics offered for the evaluation of these characteristics, in turn, reveals that the items to be valued include user's interface, software source code, and user's documentation (help). Next direction leads to the study courses which ensure mastering the development of these items. Knowing the content of every study course and the development approaches studied in them it is possible to select the particular study courses which should include teaching the assurance of these characteristics.

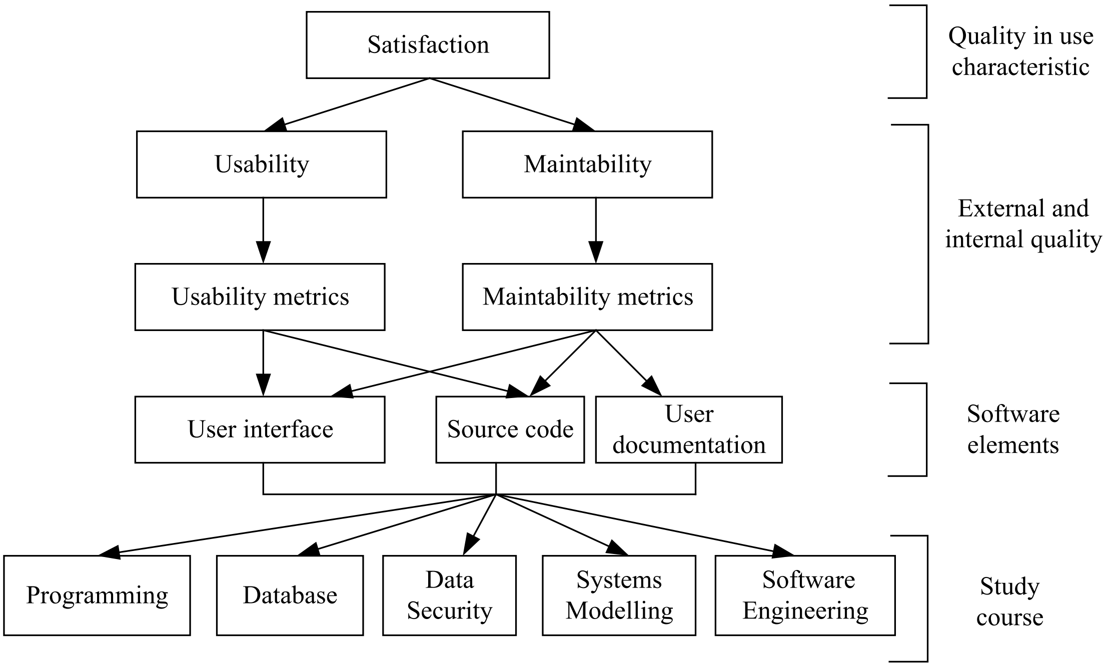

Figure3: Relation of the quality characteristic "Satisfaction" with the study courses.

The theoretical background is prepared by selecting those 11 study courses from the undergraduate (bachelor) study programme which would be most appropriate to encompass factors impacting on software product quality characteristics. Names of the study courses are taken based on the study courses of an undergraduate study programme offered by the Faculty of Information Technologies, Latvia University of Agriculture.

Opinion of the Academic Staff

The Faculty of Information Technologies, Latvia University of Agriculture when training the future specialists in information technologies basically applies the approach that individual study courses (for example, Software Engineering, Software Testing, Software Product Quality) of the study programme include the teaching of aspects related with software quality. Quality assurance measures are mentioned within the scope of individual study courses (for example, Programming); however, no single approach and quality describing terminology are used in general. Therefore, the previously prepared adjacency matrix of study courses and software quality indicators also simultaneously performs a targeted training of the academic staff. The academic staff members were offered to mark in the matrix those characteristics and sub-characteristics of the quality model the teaching of which they believe useful within the scope of a particular study course. The first survey was carried out by involving university teachers lecturing in special study courses and asking them to evaluate not only the study courses they teach but the entire study courses related with software development. The adjacency matrix included study courses named during the theoretical background stage and all characteristics/sub-characteristics of the quality model. In total, the survey encompassed 12 academic staff members from the Department of Computer Systems.

Opinion of Software Developers

It is envisaged to obtain the feedback from software users basing on the practical work experience of software developers. Software developers may have information on the fact which quality characteristics particularly interest end users. Personal experience of developers, in turn, allows specifying which development activities influence these indicators most significantly, and thus, they shall be paid more attention to during the study process. A modified adjacency matrix is being prepared for obtaining the opinion of software developers; here, the names of study courses are substituted by standard tasks of software development. It is planned to carry out this survey in software developing companies of Latvia after a coordinated view on the inclusion of quality indicators in the study programme courses is developed as a result of the Faculty internal survey.

Results of Research

The summary of the academic staff survey revealed that students are taught about software product quality; although, mastering a concrete theme, it is not emphasised which particular quality characteristics of a product are influenced thereby. The quality of an end-product is being spoken about indirectly or each lecturer using his/her own terminology. It is not wrong considering the existing multiform approaches towards quality assurance issues; although, it does not ensure a systemic approach in the study programme in general. The summary on the number of lecturers having mentioned concrete quality sub-characteristics which should be discussed in definite study courses allowed obtaining a joint list of characteristics for every study course (Figure 4). For each quality subcharacteristic the number of responses is shown in which there is a notification that this quality characteristic should be taught in the corresponding study course.

Software quality characteristic / **Study course**	*Programming Basics*	*Windows Programming*	*Database Access Applications*	*Algorithms and Structures*	*Database Technologies*	*Large Databases*	*Web Programming*	*WWW Technologies*	*Data Security*	*Computer Networks*	*Administration of Computer Network*
Functionality	**15**	**28**	**27**	**10**	**21**	**28**	**29**	**15**	**16**	**18**	**20**
suitability	4	7	8	3	5	4	8	4	2	0	1
accuracy	5	5	3	3	4	5	5	2	3	1	2
interoperability	3	7	7	0	2	6	5	3	2	6	5
security	1	5	5	1	4	8	6	3	7	7	8
functionality compliance	2	4	4	3	6	5	5	3	2	4	4
Reliability	**7**	**17**	**19**	**9**	**10**	**18**	**17**	**11**	**18**	**17**	**13**
maturity	2	4	4	2	1	4	3	4	3	5	1
fault tolerance	4	6	8	4	2	4	7	3	5	3	4
recoverability	1	4	3	2	4	6	4	3	4	3	3
reliability compliance	0	3	4	1	3	4	3	1	6	6	5

Figure 4: An excerpt from the adjacency matrix of study courses and quality indicators.

The opinion of all teachers who participates in the survey is represented in Figure 5. This chart shows the relations between the quality characteristics and the relative study courses in which development impact to them should be taught. The longer the column of a respective characteristic in the chart, the more material attention should be paid to this quality characteristic within the study course.

The results of the survey show that total opinion of the lecturers is that all software product quality characteristics should be taught in all study courses. Of course not equal attention to them should be paid in all courses. From the lecturers' point of view certain techniques of implementation of different quality characteristics should be taught much more in such study courses as Web Programming, Database Access Applications and Windows Programming. That is logically because these courses are directly related to the software development. During teaching software coding the future specialists should be taught also a techniques how to increase or to decrease particular quality subcharecteristics. For example, teaching the coding it should be taught how reasonable comments, correct definition of variables and parameters, and clearness of the code increase or decrease maintainability of the end software.

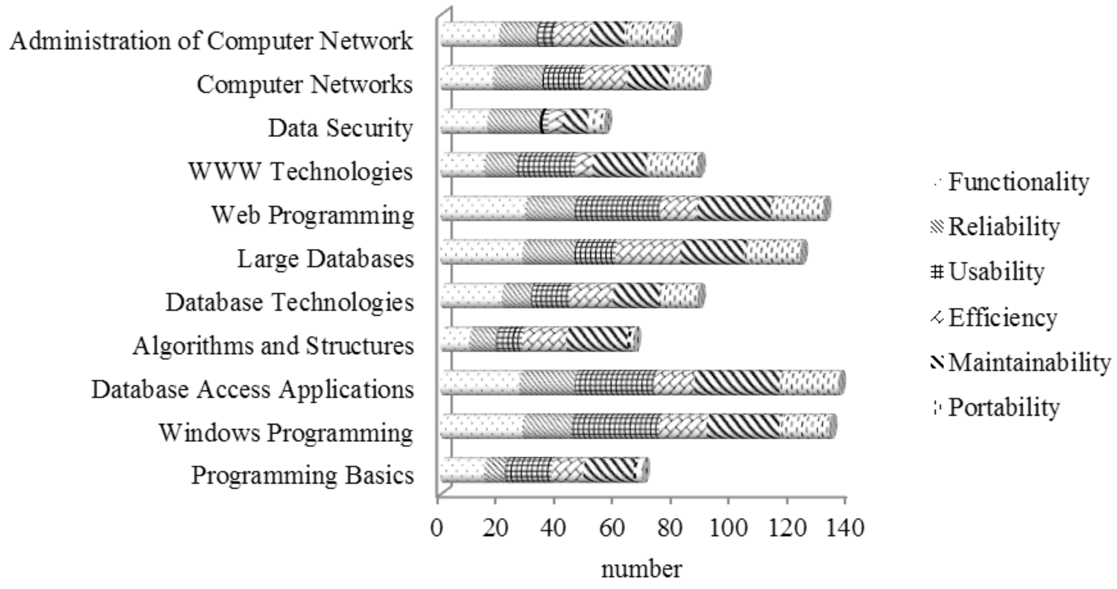

Figure 5: Impressibility of software quality characteristics in the study courses.

Summary data of each quality characteristic in all reviewed study courses of the bachelor's study programme shows (Figure 6), that the main emphasis in study process should be directed to such software quality characteristics as Functionality (21%), Maintainability (19%) un Usability (17%).

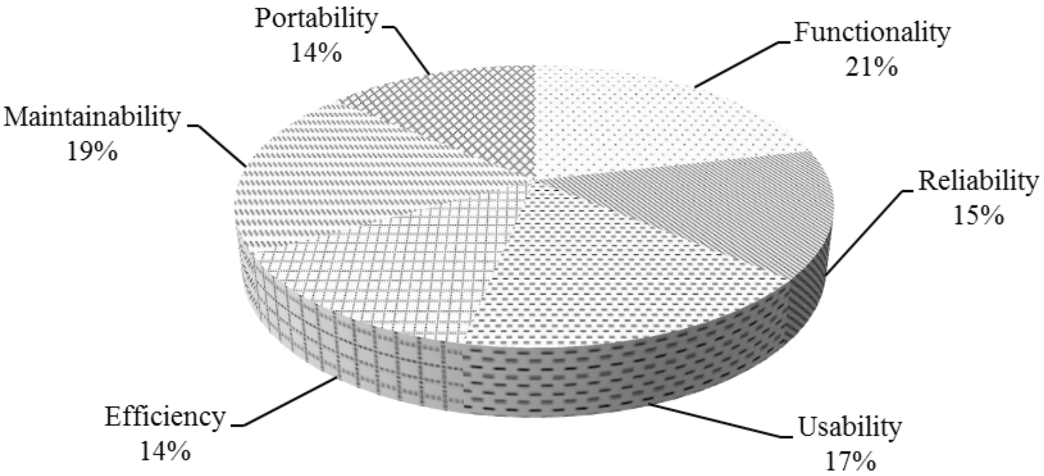

Figure 6: Quality characteristics in the study programme.

Development of the matrix of relationships between software quality characteristics and study courses is the first step to start changes in teaching future software developers. Total study process has to turn from acquisition of software development languages, technologies, tools and environments to acquisition of the development of qualitative software product. The next step is related to gathering information on what should be taught about software quality assurance in each particular study course. It is very important to balance acquisition of specific matter of each study course with included software assurance methods. Essential contribution in changes of content of study courses can be done by the software developers' work experience. Working in real software development projects, the experience can be amassed on the most essential influences to software end product quality and the main activities to be taught during the study process. The next step of the research is the software developers' survey on the inclusion of teaching software quality characteristics in main study courses.

Discussion

Application of software has surrounded all spheres of life and economy, and it becomes more and more wide. Together with this extension software quality becomes more and more important. Quality evaluation and improvement is one of the main tasks of software engineering. Software life cycle consists of different stages starting with the situation when an idea of new software arises and ending with its application. Considering the diversity of these stages the viewpoints about software quality and ways of its assurance are also different. One of the best developed directions is quality improvement of the software development processes. The second viewpoint is related to quality of software end product, definition of the quality model, quality measurements and usage of the feedback for end product quality improvement. Education is the third scope in which great attention is devoted to quality just now. This direction includes development of international recommendations for study programmes of information technologies that is the first step in qualification assurance of software developers. Some researches exist about how to include teaching of the software quality assurance into study programmes of information technologies. For example, Hilburn and Towhidnejad (Hilburn & Towhidnejad, 2000) have a viewpoint, that "one could integrate quality activities throughout the undergraduate curriculum; this approach reinforces the issue of quality over a number of courses.

This means that, as students' progress through their program of study, they will be introduced to a variety of quality activities." At the same time their assessment is, that „the disadvantage of this approach is that almost all the courses in the curriculum have already been overloaded with topics that "have to be covered". The second approach recommended by them involves adding a course devoted to software quality in either the second semester of the junior year or during the senior year. In their approach quality acquisition is based on the V model for quality.

Proposals developed in the current research are generally different from the mentioned before by using equal quality models for quality assurance and evaluation in all stages of software quality life cycle. This common quality model is based on the software product quality model defined in Standard ISO 9126, and it can be used for description of quality of the software product as well as quality of the processes. Acquisition of quality topics in study programmes is planned from two viewpoints simultaneously. There are separate study courses in the study programmes which are devoted to teaching common aspects of quality assurance. In addition to them it is planned to improve acquisition of quality assurance in all specific study courses in which development of software is taught directly. Such courses are programming, data bases management, Web programming and other basic courses. The main task of teaching quality in these courses is to create to the students total understanding that each decision taken during the development has particular impact on quality of the end product.

Conclusions

Software product is unique, since it is impossible to evaluate its quality visually consistent with any specific parameters. Different approaches for the assurance of software product quality are known worldwide and they are still developing intensively. It begins with the quality assurance of development processes and ends with the evaluation of an end-product Quality in use. Diverse views on quality and varied terminology, which is used in different stages, are one of the conditions complicating the solution of issue.

Individual internal and external quality characteristics of software product are evaluated during its development by means of testing and reviews. Neither product acquirer nor users perform a material quality evaluation during this period. The main responsibility of software product quality lies on its developers. Moreover, software quality assurance may not be the closing stage during the course of process; quality assurance and improvement shall be present in all the stages of software development. Undertaking of preventive measures, including the improvement of background level of personnel developing software, is another material requirement.

More than a ten-years' experience in the position of a leading quality manager in one of the largest IT developing companies in Latvia has proved that the implementation of quality requirements is the most efficient if the necessary quality assurance activities of every individual employee are performed as the constituents of direct piece-work without the application of a specific quality assurance sphere terminology for defining these tasks. This allowed advancing an idea that it is not enough to teach individual quality assurance methods and techniques to the future IT specialists on the level of higher education institutions. Introducing the mastering of software quality characteristics also in other core study courses would ensure a conceptual base for understanding of software quality as integral part of any everyday task performance. A purposefully organised cooperation of the involved lecturers for general disclosing of quality assurance activities in the study programme is required to implement this particular improvement. The developed approach determines that the quality understanding is based on a single quality model on all levels both evaluating the process and the quality of products and by-products. The base model is developed based on the hierarchical quality model described in the ISO/IEC 9126 standard (now ISO/IEC 25000) by adjusting it to describe the quality of other items.

The initial vision on modification of study courses is based on the information being at disposal of the leading specialists and academic staff members of the study programme by following the feedback direction of the end-product quality in use characteristic to the development process activities forming it. The decisive role is prescribed to the opinions of specialists from software developing companies. The feedback from developers to an education institution training new specialists in information technologies would essentially improve the study programme content and indirectly influence the quality of software products to be developed in future.

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

Abrahamsson, P. (1999). Commitment to Software Process Improvement - Development of Diagnostic Tool to Facilitate Improvement. *Software Quality Journal*, *8* (1), 63-76.

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

Baldassarre, M. T., Caivano, D., Pino, F. J., Piattini, M., Visaggio, G. (2012). Harmonization of ISO/IEC 9001:2000 and CMMI-DEV: from a theoretical comparison to a real case application. *Software Quality Journal*, *20* (2), 309-335.

Behkamal, B., Kahani, M., Akbari, M. K. (2009). Customizing ISO 9126 quality model for evaluation of B2B applications. *Information and Software Technology, 51*, 599-609.

Crosby, P. B. (1979). *Quality is free: the art of making quality certain*. New York: McGraw-Hill

Čevere, R., Sproģe, S. (2010). Application of Software Quality Models in Evaluation of Study Quality. *Problems of Education in the 21st Century, 21*, 37-48.

Čevere, R., Sproģe, S. (2011). Methodology for Evaluation of Internal Quality of the Study Programme. EQANIE Conference Vienna 2011. Retrieved 3/11/2012, from: *http://www.eqanie.eu/pages/events/past-events/conference-vienna-2011/proceedings.php*

Deming, W. E. (1988). *Out of the crisis: quality, productivity and competitive position*. Cambridge Univ. Press

Education Criteria for Performance Excellence 2011-2012. Retrieved 1/11/2012, from *http://www.nist.gov/baldrige/publications/upload/2011_2012_Education_Criteria.pdf*

Fam, M., Luo, Y., Wu, G., Fu, X. (2010). An Improved Analytic Hierarchy Process Model on Software Quality Evaluation. In *Proceedings 2nd International Conference Information Science and Engineering (ICISE)*. Hangzhou, China, 1838-1842.

Hilburn, T. B., & Towhidnejad, M. (2000). Software Quality: A Curriculum Postscript? In: *Proceedings of the 31st SIGCSE technical symposium on Computer science education.* New York, 167-171., doi: 10.1145/330908.331848.

ISO/IEC 25030:2007. *Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality Requirements*. International Standard.

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

ISO/IEC 9126-1:2003. *Software Engineering – Product Quality, Part 2: External Metrics*. International Organization for Standardization.

ISO/IEC 9126-1:2003. *Software Engineering – Product Quality, Part 3: Internal Metrics*. International Organization for Standardization.

Lundquist, R. (1999). Quality Systems and ISO 9000 in Higher Education. *Assessment & Evaluation in Higher Education*, *22* (2), 159-172.

Ortega, M., Perez, M., Rojas, T. (2003) Construction of a Systemic Quality Model for Evaluating a Software Product. *Software Quality Journal*, *11*, 219-242.

Osseo-Asare, A. E, Longbottom, D., Murphy, W. D. (2005). Leadership best practices for sustaining quality in UK higher education from the perspective of the EFQM Excellence Model Source. *Quality Assurance in Education*, *13* (2), 148-170.

Paulk, M. C. (1995). How ISO 9001 Compares with the CMM. *Institute for Software Research*. Paper 8. http://repository.cmu.edu/isr/8.

Rosa, M., & Amaral, A. (2007). A self-assessment of higher education institutions from the perspective of the EFQM excellence model. In D. Westerheijden, B. Stensaker, & M. Rosa (Eds.), *Quality assurance in higher education: Trends in regulation, translation and transformation* (181-207). Dordrecht: Springer.

Sproge, S., Cevere, R. (2012). Quality Models in Software Product Development Life Cycle. In: *Proceedings of the 5th International Scientific Conference on the Applied Information and Communication Technologies.* Jelgava: LLU (Latvia University of Agriculture), 2012, 69-76.

Tervonen, I., Pentti, K. (1998). Towards deeper co-understanding of software quality. *Information and Software Technology*, *39* (14-15), 995-1003.

Thonhauser, T., Passmore, D. L. (2006). ISO 9000 in Education: a comparison between the United States and England. *Research in Comparative & International Education*, *2*, 156-172.

Advised by Irina Arhipova, Latvia University of Agriculture, Jelgava, Latvia

Received: *October 25, 2012* Accepted: *December 11, 2012*

Sandra Sproge Mg.sc.ing., PhD, Latvia University of Agriculture, Faculty of Information Technology, Liela street 2, Jelgava, Latvia
E-mail: Sandra.Sproge@llu.lv
Website: http://www.llu.lv

Rudite Cevere Dr.sc.comp., Associate Profesor, Latvia University of Agriculture, Faculty of Information Technology, Liela street 2, Jelgava, Latvia.
E-mail: Rudite.Cevere@llu.lv
Website: http://www.llu.lv