QUALITY MODELS IN SOFTWARE PRODUCT DEVELOPMENT LIFE CYCLE

SANDRA SPROGE, RUDITE CEVERE

Latvia University of Agriculture, Latvia
Sandra.Sproge@llu.lv; Rudite.Cevere@llu.lv

Abstract: *Within the software product development life cycle, one may speak on qualities of four types – process quality, internal quality, external quality, and quality in use. Various quality models have been developed in the sphere of software engineering to evaluate internal and external quality of software product. All models are based on the defined hierarchic link of quality characteristics and sub-characteristics. The quality in use model of a product is also based on certain characteristics allowing evaluation of the mentioned quality. The development of quality systems and certifying of organisations in compliance with the ISO 9001 quality standards is one of the widely applied approaches in the world practice for assurance of process quality. Quality systems shall be developed according to the defined guidelines; however, no quality models are established for evaluation of process quality. The aim of the present paper is to demonstrate that quality models with earmarked quality characteristics and sub-characteristics may also be developed for the evaluation of software product development process quality.*

Keywords: software quality, software process, software process quality model.

Introduction

Software development sphere is characterised by several peculiarities:

- software product, in essence, is an abstract product, which is impossible to weight physically, to measure, or to assess it in any other way (Pressman, 2010);
- software product development period is lasting and the final result could be seen relatively late; thus, possibly causing certain obscurity and excitement from the part of a customer (Gibbs, 1994);
- requirements for the final product may be changed during the product development period (Pfleeger, 2009);
- development has a process approach (Ashrafi, 2003).

Therefore, the development of a new software product requires setting of not only quality requirements and methods for measuring the quality of a new product but also the quality assurance of its development processes. The sphere of software engineering does not discuss the quality of each individual process but focuses on the quality management systems the elaboration and maintenance of which should assure the development of a qualitative product. The mentioned sphere provides elaborated methods and an approved set of international standards determining guidelines for the development of software product. The entire standards may be classified into 4 groups consistent with the type of their use (Borzovs, Viļums, Čevere, Plūme, 1997):

1) standards determining the documentation of software;
2) standards determining the documentation of software development process;
3) standards determining the implementation of software quality assurance measures;
4) other standards related to the sphere.

Methods and standards, determining the implementation of measures for software quality assurance, describe the software quality life cycle, a quality model with the earmarked quality characteristics and sub-characteristics as well as the ways for setting and evaluation of quality requirements. However, in relation to the software development process, a model has not been sufficiently developed to evaluate the quality of processes. Several approaches have been developed for the definition, maintenance, and improvement of processes. These approaches are recorded in standards, the most popular standards being ISO 9000 and CMM (Mark, 1995).

The authors in the present paper would like to offer a method how the hierarchic quality model of software product may be transformed to the evaluation of software product development process quality. The essence of method is based on the selection of the necessary software development process for quality assurance from the universal model containing hierarchic quality characteristics.

Materials and methods

Definition of the quality model, which initially was provided by the ISO series standards and specified in the SQuaRE series standards (ISO/IEC 25030, 2007), is the following: "defined set of characteristics, and relationships between them, which provides a framework for specifying quality requirements and evaluating quality."

Historically, many software quality models have been developed and supplemented each other, and it is possible to determine and measure the quality of product by means of these models. The most well-known models are as follows:

- McCall's model;
- Boehm's model;
- FURPS model;
- Dromey's model;
- ISO 9126 model.

The entire models are based on the definition of hierarchically related characteristics and sub-characteristics of the quality indicators. These models are widely applied for the evaluation of various software products (Al-Qutaish, 2010; Khayami et al., 2009). Different modifications of models are developed for the needs of individual products (Rawashdeh and Matalkah, 2006). One of the key postulates defined by the ISO 9126 standard is the notion of software quality life cycle, which is viewed in relatively few versions trying to expand the software quality model with the process attributes (Ortega et al., 2003; Rawashdeh and Matalkah, 2006). Processes are mainly viewed as instrument furthering hopes that an adjusted and well organised process would ensure a more qualitative result.

There are also attempts to use such models for the evaluation of quality in other spheres parallel to the diversity of developed quality models. Similarly developed models have been applied in linguistics to evaluate the quality of translation since the beginning of the 1990s (King, 2003). An analogue quality model for the quality evaluation of the study process and study courses is used at Latvia University of Agriculture (Čevere and Sproģe, 2010; Čevere and Sproģe, 2011).

A question on the quality of software as a system element emerges with the development of software product quality models. The ISO/IEC 25030 standard offers a diversity of quality models within the system model. The authors offer supplementing the set of models with the quality model of software development processes (Fig. 1), since, consistent with the definition provided by the ISO/IEC 9126-1 standard, the quality of processes affects the quality of product (product quality develops from).

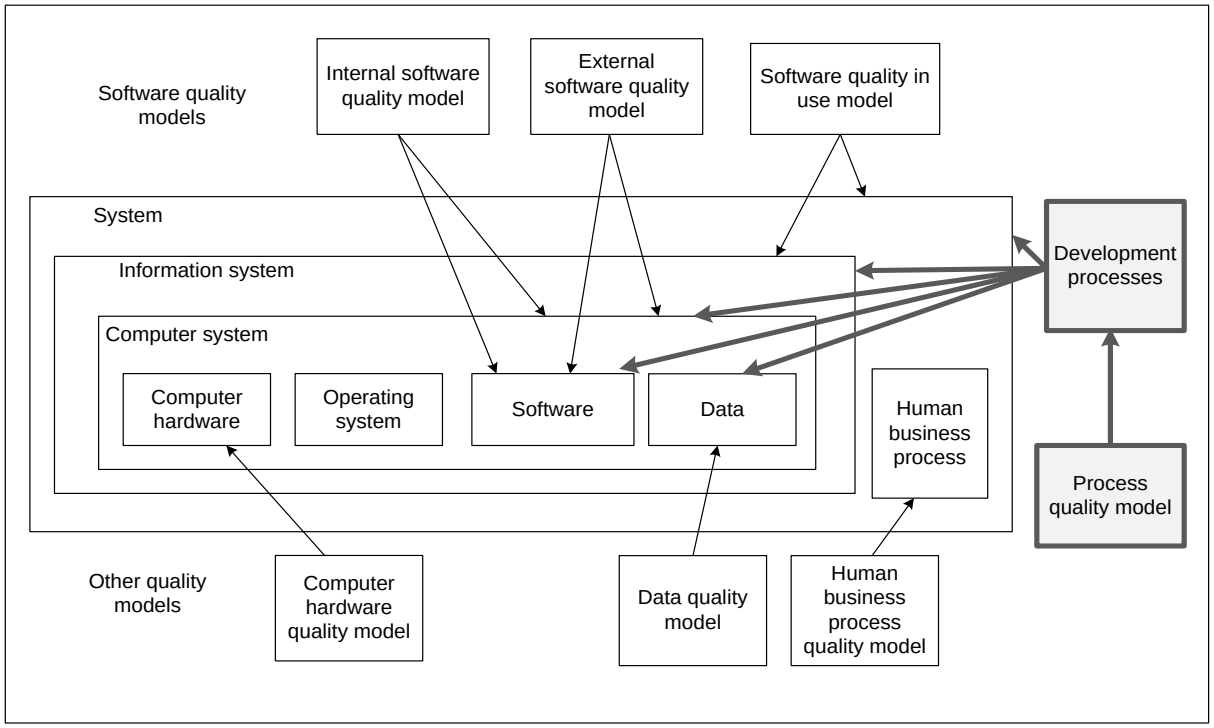

Fig. 1. **Set of system quality models** (ISO/IEC 25030, 2007).

The development and application of software product quality models in various other spheres ensure so many outputs that it is practically impossible to study all of them and evaluate their suitability prior to solving a particular aspect. According to Abran et al. "it is not seen to be economically feasible for either industry or the research community to investigate each of the hundreds of alternatives" (Abran et al., 2006).

Considering the diversity of quality aspects, different and various quality requirements of stakeholders as well as different evaluation contexts, the attempts to establish an identical quality model, which could serve a sufficiently broad sphere of evaluation, would turn inefficient. Instead, it is necessary to choose the base version of quality model and to develop procedure for its transformation to various authorities in use based on the previous personal and international experience. The present research develops an approach for the use of a single quality model for the evaluation of the entire system elements. The authors propose a base quality model, which

is supported by the ISO 9126 standard, and the procedure for transformation of this model consistent with each individual necessity.

The single approach means that the quality models are based on the hierarchic structure, which contains quality characteristics and corresponding sub-characteristics. The set of possible characteristics and sub-characteristics is based on the values proposed within the course of software quality valuation. Literature describes the results on revision, comparison and summarisation of many and various models. Yet, the set of these attributes is analysed once more, since the present research tries to ascribe such attributes also to the process quality models.

McCall' model is one of the most known quality models in the software engineering literature. This model serves as the basis for many other quality models. The model (McCall et al., 1977) divides all the characteristics into three major perspectives – product operations, product revision, and product transition. Respectively, 11 quality characteristics and 23 subcharacteristics are specified according to these perspectives. One and the same sub-characteristics are adjusted to several quality characteristics.

In Boehm's model, the higher level characteristics represent the actual basic usage requirements for software to which the evaluation of product shall be put – general utility of software (Berander, 2005). The higher level characteristics answer the three main questions, which are important for the evaluation of software quality:

- as-is utility – to address how well (easily, reliably and efficiently) one could use the software product;
- maintainability – to address how it is to understand, modify and retest the software product;
- portability – to address if one could use the software product when the drive environment would be changed.

The FURPS model for quality determination takes into account characteristics the first letters of which make up its name – Functionality, Usability, Reliability, Performance, and Supportability. The application of the FURPS model envisages the performance of two steps – setting of priorities and defining of quality attributes than can be measured.

Dromey's model focuses on the relations between quality attributes and sub-attributes as well as tries to relate the software product qualities to the software quality attributes (Dromey, 1995).

The ISO 9126 model defines the quality of product as a set of product characteristics. The model earmarks internal and external quality of the software product. External characteristics are characteristics influenced by the operation of a product in the provided environment, while the characteristics, which pertain to the product development aspects are internal characteristics (ISO/IEC 9126-1, 2001). The ISO 9126-1 quality model is the most useful one since it has been build.

The number of characteristics and sub-characteristics defined in McCall's, Boehm's, FURPS, ISO 9126, and Dromey's models is different (Table 1) as well as the type of their hierarchic relations is different:

- 1:n – several sub-characteristics are subjected to one quality characteristics (e.g., ISO 9126);
- n:m – each quality characteristics is related to one or several sub-characteristics (e.g., McCall).

Table 1

Description of the quality models

Model	Number of characteristics	Number of sub-characteristics	Relationships	Metrics
McCall	11	23	n:m	+
Boehm	7	15	n:m	+
FURPS	5	28	1:n	+
Dromey	7	-	n:m	+
ISO 9126	6	27	1:n	+

The summary outlines that only one quality characteristics – "Reliability" is present in all quality models and that there are several characteristics, which are present only in one model (for example, Performance). Models demonstrate that characteristics of individual models are used as sub-characteristics in other models. Table 2 summarises quality characteristics (Ch.) present in all the models, the Column "Subch." displays models, where these characteristics are used as quality sub-characteristics. According to Table 2, six quality characteristics (maintainability, reliability, efficiency, usability, portability, functionality) are present in at least three of the analysed quality models. These common characteristics coincide also with the quality characteristics of the ISO 9126 model.

The analysis of models has resulted in making of a list of found terms, which are used to denote characteristics and sub-characteristics. Individual terms, which could denote one and the same notion, are summarised as synonyms (for example, expandability, extensibility, or augmentability). It has to be considered that different stakeholders willing to evaluate the software quality may better understand one or the other name of the quality characteristics or sub-characteristics when choosing the existing quality model or developing a new one.

Table 2

Summary on the characteristics of the quality models

Ch. - charasteristics; **Subch.** – sub-charasteristics

Characteristics	McCall		Boehm		FURPS		Dromey		ISO 9126	
	Ch.	Subch.	Ch.	Subch.	Ch.	Subch.	Ch.	Subch.	Ch.	Subch.
Maintainability	+					+	+		+	
Flexibility	+									
Testability	+		+							+
Correctness	+									
Reliability	+		+		+		+		+	
Efficiency	+		+			+	+		+	
Integrity	+									
Usability	+				+		+		+	
Portability	+		+				+		+	
Reusability	+						+			
Interoperability	+									
Understandability			+							+
Functionality					+		+		+	
Performance					+					
Supportability					+					
Human Engineering			+			+				
Modifiability			+							

The summarised set of quality attributes was further used for the development of the base quality model and elaboration of its transformation procedure.

Results and discussion

The development of the base hierarchic quality model aggregated quality characteristics and sub-characteristics from all the previously mentioned quality models. A two-level hierarchic structure was created for this quality model (Figure 2):

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

The characteristics of the base model include those quality characteristics, which are present in at least three quality models. Characteristics mentioned in the ISO 9126 model were chosen as the basic characteristics, since there are many studies on the evaluation of software product quality based on the ISO 9126 model (Fam et al., 2010; Sibisi et al., 2007). Sub-characteristics presented in the other quality models and related to the corresponding quality characteristics were chosen as additional sub-characteristics to each quality characteristics, when determining the type of relation n:m (Fig. 2). Optional characteristics and sub-characteristics of the base quality model are constructed by those characteristics and sub-characteristics, which are mentioned in McCall's, Boehm's, FURPS, and Dromey's models and not included into the basic sub-characteristics and additional sub-characteristics of the base model.

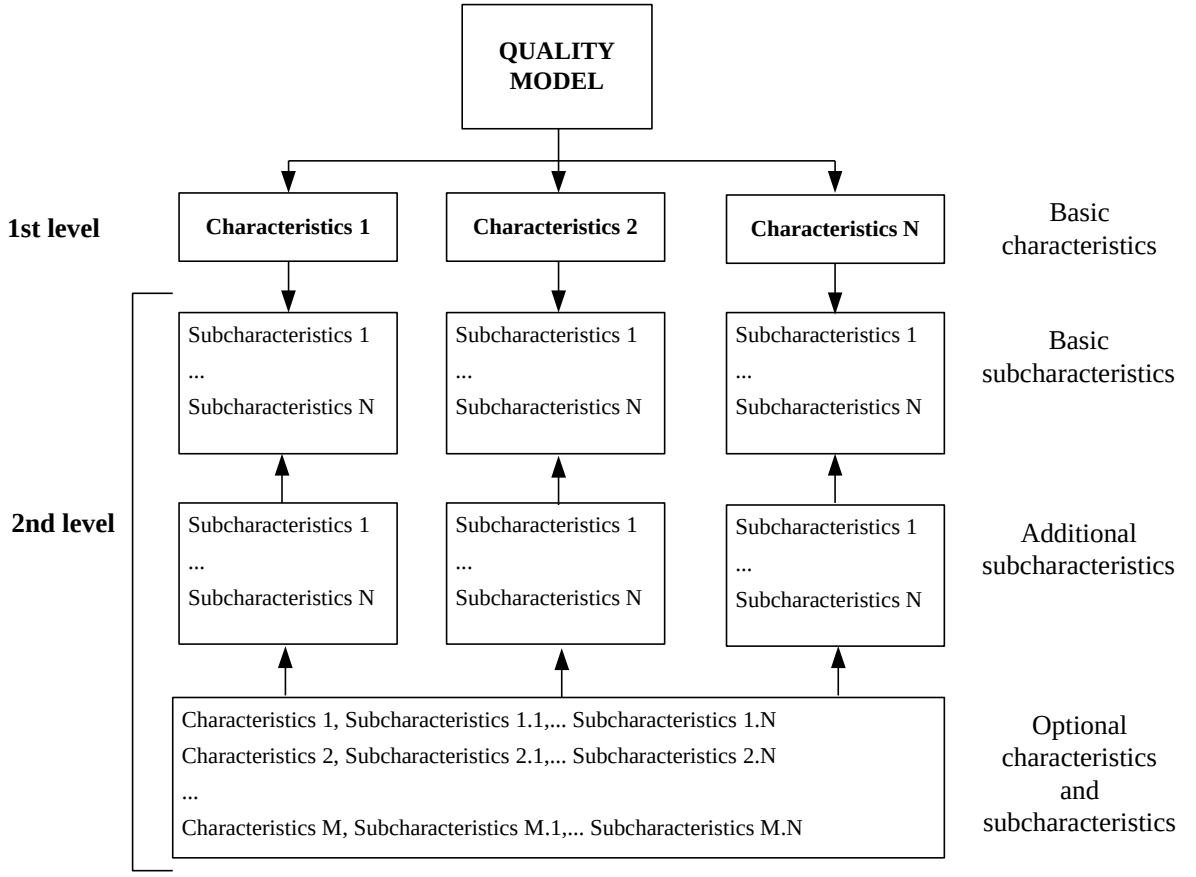

Fig. 2. **Hierarchic structure of the base quality model.**

The initial list of optional characteristics and optional sub-characteristics of the model consists of:

- Reusability – *Generality, Machine independence, Modularity, Self-descriptiveness*;
- Testability – *Accountability, Communicativeness, Self-descriptiveness, Structuredness*, Instrumentation, *Modularity, Self-descriptiveness, Simplicity*;
- Correctness – *Completeness, Consistency*, Traceability;
- Flexibility – Expandability, Generality, *Self-descriptiveness*;
- Human Engineering – *Accessibility, Communicativeness, Robustness/Integrity*;
- Integrity – Access audit, Access control;
- Interoperability – Communication commonality, Data commonality, *Modularity*;
- Modifiability – Augmentability, *Structuredness*;
- Performance – Efficiency, Resource consumption, Response time, Speed, Throughput;
- Supportability – *Adaptability*, Compatibility, Configurability, Extensibility, *Installability*, Localisability, Maintainability, Portability. Serviceability, *Testability*;
- Understandability - *Conciseness, Consistency*, Legibility, *Structuredness*.

Those quality sub-characteristics, which were not included into the number of additional sub-characteristics of the base model, are marked in *Italic* in the list of optional characteristics and sub-characteristics of the model. Therefore, double characteristics were not included into the final version of the model.

The summary on the experience in application the existing models evidence that one and the same notion may be used on different levels (as characteristics or sub-characteristics); for example, testability, efficiency, or as a sub-characteristic for different characteristics. Thus, they may repeat or overlap in the lists of proposed values.

The base quality model may be used both for the quality evaluation of software product and software development processes. Figure 3 outlines the software product quality model with earmarked quality characteristics, basic sub-characteristics, and additional sub-characteristics.

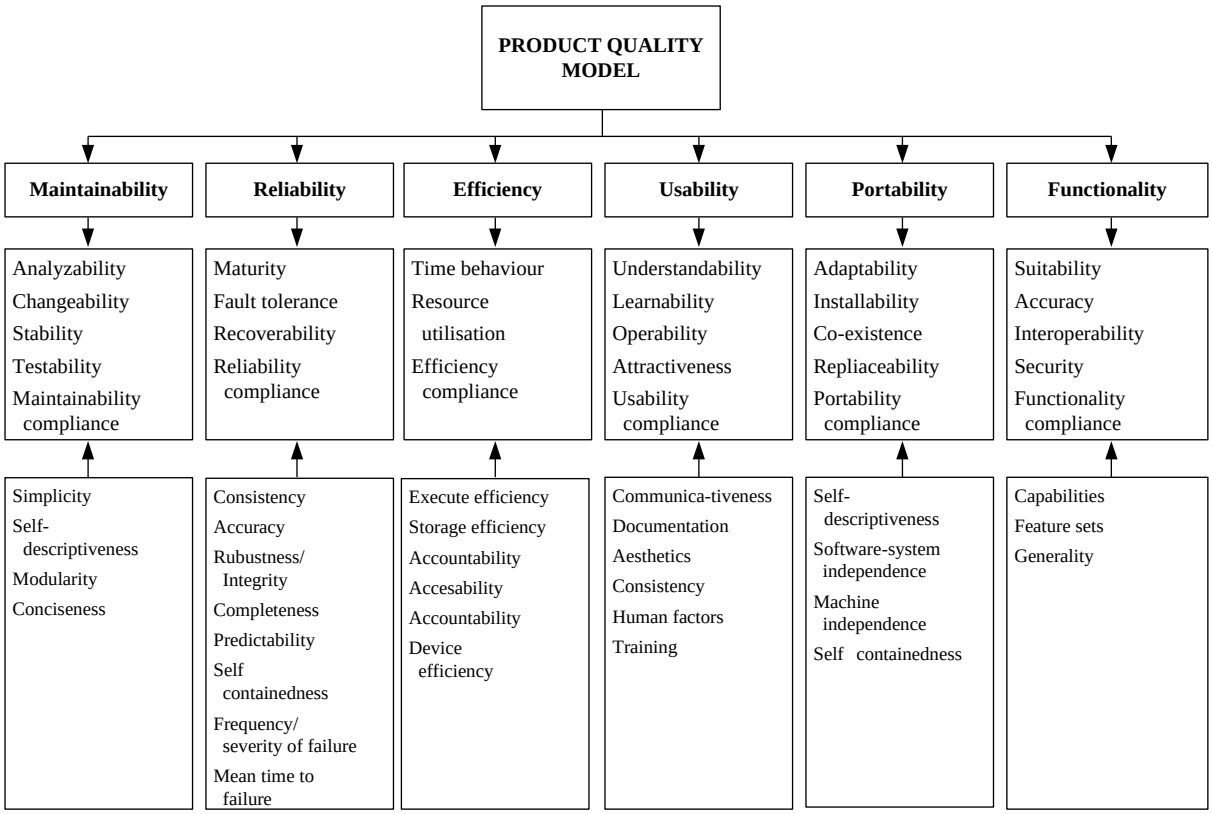

Fig. 3. **The base quality model of a product.**

The quality of development processes plays a significant role in the software product quality assurance. This is reflected both in the quality life cycle described in the ISO 9126 standard, and broad and successful implementation of quality assurance measures in IT companies based on the ISO 9000 standard, and TickIt and CMMI recommendations. The quality model of processes is not proposed; nevertheless undertaking of the respective quality assurance measures is related to the development of processes, evaluation, and implementation of continuous improvement measures. The quality model of processes and evaluation of quality is required in many cases, especially when creating and organising the course of processes as well as teaching the performance of these tasks. To establish a quality model, a process is viewed in its generally accepted form as transformation of input into output with the existing resources and defined management (Fig. 4).

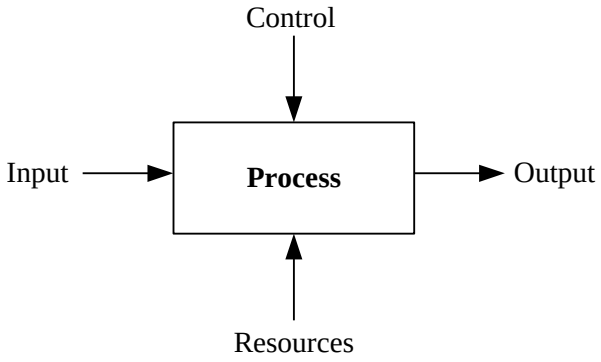

Fig. 4. **Process model.**

The evaluation of characteristics used in the base quality model leads to the conclusion that the majority of them may be used also as quality attributes of processes, certainly, considering that the interpretation of the notion may slightly or even significantly differ from the interpretation used in the evaluation of products. The basic difference relates to the metrics used in the evaluation of a characteristic or sub-characteristic. For example, if suitability metric "Functional implementation completeness" is used for the evaluation of a product functionality. This metric 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. In case of quality evaluation of processes, the use of all inputs and acquisition of outputs will be evaluated instead of specified functions. The base quality model of processes is proposed through such an approach (Figure 5). For

the comparison with the quality model of product, those sub-characteristics, which would not be suitable for the evaluation of processes, are deleted and those names of sub-characteristics, where the used term has been changed, are marked in *Italic* (for example, "data prevention" is recommended instead of "security", "readiness", →"maturity", "corrigibility" →"recoverability").

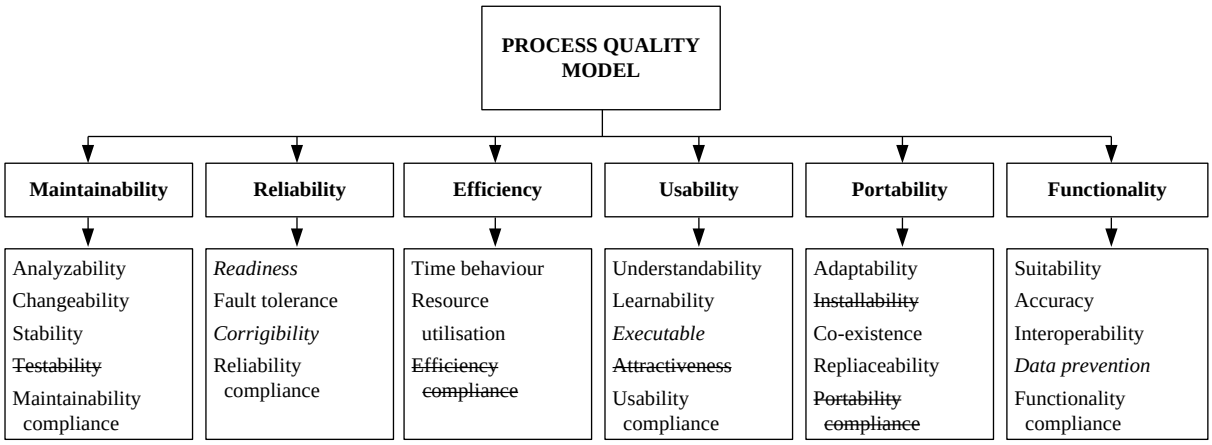

Fig. 5. **The base quality model of processes.**

It is envisaged to adjust the base quality model to the context of usage and requirements of stakeholders engaged in the evaluation in each individual case. In general, this adjustment shall be done consistent with the sequence of operations prescribed in the application procedure of the base model:

1. To identify/to choose the group that is eager to evaluate the quality of respective product/process.
2. To identify the quality requirements of a product/process set by the group of stakeholders.
3. To choose basic and additional quality characteristics and sub-characteristics.
4. In case of necessity, to choose optional characteristics and sub-characteristics.
5. To prioritise each selected characteristic.
6. To choose metrics for each quality characteristic/sub-characteristic.
7. To set the maximum value of each selected metric.
8. To provide the process of measuring for the selected metrics.
9. To analyse the obtained data – deviation from the set maximum value.
10. To prescribe activities for the improvement of a product/process quality.

Conclusions

The base quality model and its application procedure may be used both for the quality evaluation of software product and software development processes. Since different stakeholders are interested in the software quality and each of them has its own point of view and demands in relation to quality, then such a model is flexible in its usage. Quality may be evaluated by selecting the necessary characteristics and metrics.

The valuation of characteristics for a software product is based on metrics recommended by the ISO 9126 standards. Such a set of metrics is not published for the quality evaluation of processes; thus, recommendations for their description are prepared basing on the analogy with the quality evaluation of a product.

Quality may be evaluated in any phase of the software life cycle; however, in the particular research, when developing the base quality model, the main attention is paid to its application in preventive measures including quality training based on the model in training specialists of 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

Abran, A., Al-Qutaish, R., Cuadeado-Gallego, J., 2006. Investigation of the Metrology Concepts in ISO 9126 on Software Product Quality Evaluation. [online] [10.01.2012.]. Available at: *http://www.gelog.etsmtl.ca/publications/pdf-presentations/1004.pdf*

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

Ashrafi, N., 2003. The Impact of Software Process Improvement on Quality: in Theory and Practice. *Information & Management,* Vol .70(7), pp. 677-690.

Berander, P. et. al., 2005. *Software Quality Attributes and Trade-offs*. – Sweden, Blekinge Institute of Technology, 2005, p. 100.

Borzovs, J., Viļums, Ē., Čevere, R., Plūme, J., 1997. *Ieteikumi programmatūras dokumentācijas komplektam (Recommendations for the Set of Software Documentation)*. Rīga, RITI.

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

Čevere, R., Sproģe, S., 2011. Methodology for Evaluation of Internal Quality of the Study Programme. EQANIE Conference Vienna 2011. [online] [10.01.2012.]. Available at: *http://www.eqanie.eu/pages/events/conference-vienna-2011/proceedings.php*.

Dromey, R., 1995. A Model for Software Product Quality. *IEEE Transactions on Software Engineering*, 21, pp. 146-162.

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

Gibbs, W., 1994. Software's Chronic Crisis. *Scientific American* 271(3), pp. 86-95.

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.

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

King, M., 2003. Living up to Standards. In: *Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing*, Budapest, Hungary.

Mark, P., 1995. How ISO 9001 Compares with the CMM. Institute for Software Research. Paper 8. Available at: *http://repository.cmu.edu/isr/8*.

McCall, J., Richards P., Walters G., 1977. Concepts and Definitions of Software Quality, Volumes I. US Rome Air Development Centre Reports, US Department of Commerce, USA. [online] [10.01.2012.]. Available at*: http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA049014*.

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

Pfleeger, S., 2009. *Software Engineering: Theory and Practice 4th ed*., Prentice Hall, p. 792.

Pressman, R., 2010. *Software Engineering: a Practitioner's Approach 7th ed*. New York: McGraw-Hill Higher Education, p. 895.

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

Sibisi, M. Cristo van Waveren C., 2007. A Process Framework for Customising Software Quality Models. *In: Proceedings AFRICON 2007*, pp. 547-554.