Evolution of Software Quality Models in Context of the Standard ISO 25010

Chapter · January 2014

DOI: 10.1007/978-3-319-07013-1_21

4 authors:


Oleksandr Gordieiev
Banking University


Vyacheslav Kharchenko
KhAI - Aerospace university


Nataliia Fominykh
Plekhanov Russian Academy of Economics


Vladimir Sklyar
KhAI - Aerospace university

Evolution of software quality models in context of the standard ISO 25010

Oleksandr Gordieiev[1] and Vyacheslav Kharchenko[2] and Nataliia Fominykh[1] and Vladimir Sklyar [3]

[1] Sevastopol Institute of Banking of University of Banking of the National Bank of Ukraine, 6 Parkova street, Sevastopol, Ukraine
alex.gordeyev@gmail.com
[2] National Aerospace University «KhAI», 17 Chkalova street, Kharkiv, Ukraine
V.Kharchenko@khai.edu
[3] Research and Production Corporation Radiy, 29 Geroev Stalingradu , Kirovpgrad , Ukraine
v.sklyar@radiy.com

Abstract. Evolutionary analysis of software (SW) quality models (QM) over the past forty years, from one of the first software QM by McCall to the model presented in the standard ISO 25010 is performed. 9 models were chosen for the analysis and divided into sets of basic and corporate QMs according to the completeness, detailing and significance. The choice of basic models McCall (1977), IEEE 1219 (1993), ISO9126-1 (2001), ISO 25010 (2010) is grounded. QM structure is described by hierarchy whose elements are sets of characteristics (subcharacteristics) and relations of subordination between them. To assess the complexity and completeness of SW QM and to compare them with the latest ISO 25010 model special particular and general metrics are introduced. Analytic dependence of the growth of model complexity represented by a linear function is obtained. Analysis of some characteristics evolution (operational suitability, effectiveness, reliability, usability, safety, etc) is performed.

Keywords: evolutionary analysis, software quality models, complexity metrics, ISO 25010.

1 Introduction

1.1 Motivation

Beginning of active development and use of software as an integral part of computers can be considered the middle of the last century. The term Software Engineering appeared in 1968 at NATO Software Engineering Conference [1] where corresponding concept was formed. During the period of software engineering development as an independent direction in Engineering practice, and then forming it as a systematic science, one of the key issues was software (SW) quality.

Software quality is a degree to which a software product satisfies stated and implied needs when used under specified conditions [2]. SW quality model is [3] usually defined as a set of characteristics and relationships between them which ac-

adfa, p. 1, 2014.

tually provides the basis for specifying the requirements of quality and evaluating quality. A lot of software quality models (SWQM) have been introduced for the last decades [4]. Quality models structure is described by hierarchy whose elements are sets of characteristics (CHs) (subcharacteristics (SubCHs)) and relations of subordination between them. Characteristics (subcharacteristics) included into the models as usual are the basis of software projects requirements.

The motive of writing this paper is release of ISO25000 series as a new generation of International Organization for Standardization requirements document connected with software standardization and its quality evaluation. It was not just a very important step to improve SW QM, but a significant event in models evolution that reflected changes in software engineering as well.

1.2 Analysis Methods of SW Quality Models

A lot of SW quality models have been introduced for almost half century history of Software Engineering, but only a part of them became widely known and is used requirements elicitation. Preliminary analysis of the publications describing software quality models and techniques of SW QM analysis [2-14], allowed determining the most significant of them. Information about these models indicating the bibliographic data and two important parameters (quality hierarchy number and characteristics/subcharacteristics number levelization) is systemized in table 1.

Table 1. The most well-known SW Quality Models

№	SW QM name	Publication year	Model levels number	CHs / SubCHs number	Author	Reference/ Resource
1	McCall	1977	2	11/35	John McCall	[5]
2	Boehm	1978	3	3/8/18	Boehm	[6]
3	Carlo Ghezzi	1991	1	8	Carlo Ghezzi	[7]
4	FURPS	1992	2	5/25	Grady R. & Hewlett Packard	[8]
5	IEEE	1993	2	6/19	IEEE	[9]
6	Dromey	1995	2	4/13	Dromey	[10]
7	ISO 9126-1	2001	2	6/19	ISO	[13]
8	QMOOD	2002	1	6	Bansiya	[12]
9	ISO 25010	2010	2	8/31	ISO	[2]

SW quality models diversity causes a lot of works aimed at their comparative analysis. These researches intensified in the 90 years before ISO 9126 publication and in first years of its practical use.

Review of works related to SW QM analysis techniques [3, 12-14] shows that modern ISO 25010 standard, where a new SW quality model was adopted by the international institution, is not taken into account. Besides:

– the works are usually reduced to an analysis of characteristics and subcharacteristics are not considered enough or are not a subject of the analysis at all;

– formal techniques of SW QM description are based on table representation of models, but their description in sets and relations between them is not analyzed;

– procedure of SW QM analysis is not formalized.

Considering the importance of quality models base, it requires further researches in this field.

1.3 Problem Statement

Every new model appearing is a reflection of the quality of the constantly changing (rising, expanding and specifying) software requirements. It is caused by:

– dynamic development of software products and technologies;

– rising impact of software tools on functional capabilities, performance, reliability, security and other characteristics of the computer systems, and the technical systems in general;

– development of the regulatory framework at the international, national and corporate levels.

Variety of software quality models and frequency of their appearance encouraged the authors to analyze the evolution of the well-known software quality models in the context of ISO 25010.

Regarding this, the goal of the paper is, firstly, to develop a formalized description of the SW QM and metrics to present them in a compact form, and, secondly, to perform a comparative analysis of software quality models and to study their changes for more than 40 years of evolution.

2 Systematic SW QM description

2.1 Description Principles

The primary task while analyzing the evolution of software quality models is the problem of transition from their verbal and structural-table representation to the formal description in terms of the algebra of sets and relations. To describe the models, sets of software quality model elements (SSQME - Set of Software Quality Model Elements) and relationships of model elements (SRSQME - Set of Relationships Software Quality Model Elements) were defined. Set of software quality model elements may be presented as the following:

$$SSQME_i = \{EM_i^j, EM_i^{j+1}, ..., EM_i^n\}, \quad (1)$$

where i is index of model, j is index of model element EM_i^j.

To describe the relationship between the characteristics of different levels predicate (relation) R (“parent-child”) is used:

$$EM_i^j R EM_i^k, \quad (2)$$

where EM_i^j - parent, EM_i^k - child.

Thus, every software quality model (SQM) is described by two sets (the set of model elements and the set of model elements relations - R) and the table of the model semantic content is needed to determine SSQME sets elements accordance and their names. For example, let us describe ISO 25010 software quality model using this approach:

$$SSQME_9 = \left\{ \begin{array}{l} EM_9^1, EM_9^2, EM_9^3, EM_9^4, EM_9^5, EM_9^6, EM_9^7, EM_9^8, EM_9^9, EM_9^{10}, \\ EM_9^{11}, EM_9^{12}, EM_9^{13}, EM_9^{14}, EM_9^{15}, EM_9^{16}, EM_9^{17}, EM_9^{18}, EM_9^{19}, EM_9^{20}, \\ EM_9^{21}, EM_9^{22}, EM_9^{23}, EM_9^{24}, EM_9^{25}, EM_9^{26}, EM_9^{27}, EM_9^{28}, EM_9^{29}, EM_9^{30}, \\ EM_9^{31}, EM_9^{32}, EM_9^{33}, EM_9^{34}, EM_9^{35}, EM_9^{36}, EM_9^{37}, EM_9^{38}, EM_9^{39}, EM_9^{40} \end{array} \right\}$$

$$SRSQME_9 = \left\{ \begin{array}{l} EM_9^1 REM_9^2, EM_9^1 REM_9^6, EM_9^1 REM_9^{10}, EM_9^1 REM_9^{13}, EM_9^1 REM_9^{20}, \\ EM_9^1 REM_9^{25}, EM_9^1 REM_9^{31}, EM_9^1 REM_9^{35}, EM_9^2 REM_9^3, EM_9^2 REM_9^4, \\ EM_9^2 REM_9^5, EM_9^6 REM_9^7, EM_9^6 REM_9^8, EM_9^6 REM_9^9, EM_9^{10} REM_9^{11}, \\ EM_9^{10} REM_9^{12}, EM_9^{13} REM_9^{14}, EM_9^{13} REM_9^{15}, EM_9^{13} REM_9^{16}, EM_9^{13} REM_9^{17}, \\ EM_9^{13} REM_9^{18}, EM_9^{13} REM_9^{19}, EM_9^{20} REM_9^{21}, EM_9^{20} REM_9^{22}, EM_9^{20} REM_9^{23}, \\ EM_9^{20} REM_9^{24}, EM_9^{25} REM_9^{26}, EM_9^{25} REM_9^{27}, EM_9^{25} REM_9^{28}, EM_9^{25} REM_9^{29}, \\ EM_9^{25} REM_9^{30}, EM_9^{31} REM_9^{32}, EM_9^{31} REM_9^{33}, EM_9^{31} REM_9^{34}, EM_9^{35} REM_9^{36}, \\ EM_9^{35} REM_9^{37}, EM_9^{35} REM_9^{38}, EM_9^{35} REM_9^{39}, EM_9^{35} REM_9^{40} \end{array} \right\}$$

Its semantic content is presented by table 2.

Table 2. ISO 25010 Model Semantic Content Table

SSQME element	Model element	SSQME element	Model element	SSQME element	Model element
EM_9^1	System/Software Product Quality	EM_9^{15}	4.2 Learnability	EM_9^{28}	6.3 Non-repudiation
EM_9^2	1 Functional Suitability	EM_9^{16}	4.3 Operability	EM_9^{29}	6.4 Accountability
EM_9^3	1.1 Functional Completeness	EM_9^{17}	4.4 User Error Protection	EM_9^{30}	6.5 Authenticity
EM_9^4	1.2 Functional Correctness	EM_9^{18}	4.5 User Interface Aesthetics	EM_9^{31}	7 Portability
EM_9^5	1.3 Functional Appropriateness	EM_9^{19}	4.6 Accessibility	EM_9^{32}	7.1 Adaptability
EM_9^6	2 Performance Efficiency	EM_9^{20}	5 Reliability	EM_9^{33}	7.2 Installability
EM_9^7	2.1 Time Behaviour	EM_9^{21}	5.1 Maturity	EM_9^{34}	7.3 Replaceability
EM_9^8	2.2 Resource Behaviour	EM_9^{22}	5.2 Availability	EM_9^{35}	8. Maintainability
EM_9^9	2.3 Capacity	EM_9^{23}	5.3 Fault Tolerance	EM_9^{36}	8.1 Modifiability
EM_9^{10}	3 Compatibility	EM_9^{24}	5.4 Recoverability	EM_9^{37}	8.2 Testability
EM_9^{11}	3.1 Co-existence	EM_9^{25}	6 Security	EM_9^{38}	8.3 Modularity
EM_9^{12}	3.2 Interoperability	EM_9^{26}	6.1Confidentiality	EM_9^{39}	8.4 Reusability
EM_9^{13}	4 Usability	EM_9^{27}	6.2 Integrity	EM_9^{40}	8.5 Analyzability

EM_9^{14}	4.1 Appropriateness recognisability				

For the whole set of software quality models SSQM = {SQM_i} a common set of model elements using a simple sets union operation can be formed. However, this requires further analysis of the semantic content of every characteristic and subcharacteristic which is not being carried out in this article. For this reason, additional elements of indexing are used within the models. It should be noted that model level number for the formal description of software quality models isn't directly considered.

3 Metrics

To briefly characterize the proposed analysis technique, let us introduce some initial terms:

– conceptual model is a model which a model under study is compared with;

– model under study is a model which is compared with a conceptual model;

– characteristic under study is a conceptual model characteristic which is compared with model under study characteristics.

The technique is based on comparing a model under study with the conceptual model, i.e. every SW Quality Model is compared with the conceptual model. So, the analysis is equivalent to semantic comparing characteristics and subcharacteristics of a model under study and the conceptual model with regard to their structures. Selecting a reference model, is usually performed by an expert who has relevant experience and qualifications.

At the following stage comparison of models among themselves should be performed. The simplest and most obvious metrics are offered. Relationship and subordination of these metrics is presented in Fig. 1.The main objective of such metrics is comparison of models with reference model bottom up, i.e. at the level of subcharacteristics (SMM, CSCM and CMM metrics), further characteristics (CMCM metric) and models as a whole (CSQMCM metric).

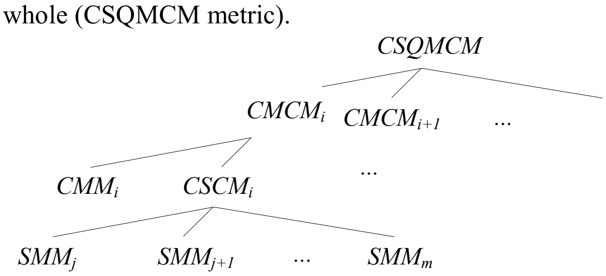

Fig. 1. Metrics relation and subordination

Features of the proposed metrics are the following:

– subcharacteristics matching metric (SMMj). Every subcharacteristic match value is identified according to the following formula SMMj = 0,5 / number of reference (conceptual) model elements subcharacteristics of the characteristic under study;

– cumulative subcharacteristics comparison metric (CSCM) is evaluated as a sum of SMM:

$$CSCM_i = \sum_{j=1}^{k} SMM_j \; ; \qquad (3)$$

– characteristics matching metric (CMM) takes the value of 0.5 in case of matching or 0 if the characteristics are different;

– cumulative matching characteristics metric (CMCM) is calculated as a sum of CMM metric and $\sum_{j=1}^{k} CSCM_j$:

$$CMCM_i = CMM_i + \sum_{j=1}^{k} CSCM_j \; ; \qquad (4)$$

– cumulative software quality models comparison metric (CSQMCM) is calculated according to the formula:

$$CSQMCM_i = \sum_{i=1}^{n} CMCM_i \; . \qquad (5)$$

4 Software quality models analysis results

Let us conduct SW QM analysis and first of all, define the reference (contextual) model. SW Quality Model ISO 25010 will be considered as uppermost and etalon regarding to all other models. It is the newest introduced model and takes into account main modern software peculiarities in point of view quality evaluation. This model is described in an international standard of top level. The reference model characteristics set consists of functional suitability, performance efficiency, compatibility, usability, reliability, security, portability, maintainability.

Results of semantic comparison of software quality models characteristics and subcharacteristics for different levels are presented in table 3.

Starting abscissa point (SW QM appearance years) is 1970. To represent information in a compact form characteristics and subcharacteristics are presented by numbers in table 3 according with table 2 for SW QM ISO 25010.

The results of quality models characteristics analysis are the following. Eevery 10 years a new quality model appears. The considered SW QMs can be divided into the following groups:

– the first group consists of the fundamental basic SW QM. These models are the result of authoritative international team work, such as ISO and IEEE. Therefore, such SW QM as IEEE, ISO 9126-1, ISO 25010 are regarded as fundamental;

– the second group consists of corporative SW QM. These models, as usual, are significant (signature models) and according to quality level (nomenclature, characteristics and their relations) and significantly inferior in quality to the basic models. McCall, Ghezzi, FURPS, Dromey, QMOOD are considered to be in this group.

Results of the models comparison using CSQMCM metrics (table 2) are presented by diagram (Fig. 2). Values of CSQMCM metrics for each model are presented in table 3.

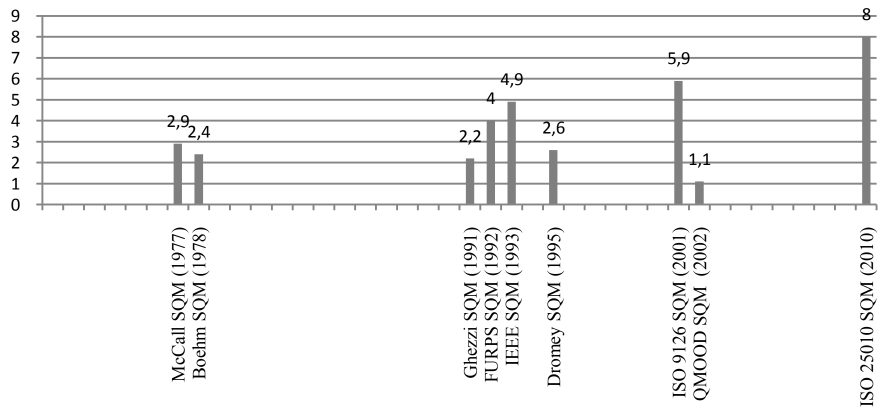

Fig. 2. Results of SW QM comparison in terms of CSQMCM

Table 3. Software Quality Models Analysis Results

Conceptual model (ISO 25010)		**ISO 9126 model**				**QMOOD model**			
CHs	Sub-CHs	CHs	Sub-CHs	CMM	SMM	CHs	Sub-CHs	CMM	SMM
1		1	-	0,5	0	4	-	0,5	0
	1.1	-	-	0	0	-	-	0	0
	1.2	-	1.2	0	0,167	-	-	0	0
	1.3	-	-	0	0	-	-	0	0
				CMCM=0,667				**CMCM=0,5**	
2		4		0,5	0	6	-	0,5	0
	2.1	-	4.1	0	0,167	-	-	0	0
	2.2	-	4.2	0	0,167	-	-	0	0
	2.3	-	-	0	0	-	-	0	0
				CMCM=0,834				**CMCM=0,5**	
3		-	-	0	0	-	-	0	0
	3.1	-	6.3	0	0,25	-	-	0	
	3.2	-	1.3	0	0,25	-	-	0	0
				CMCM=0,5				**CMCM=0**	
4		3	-	0,5	0	-	-	0	0
	4.1	-	-	0	0	-	-	0	0
	4.2	-	3.2	0	0,083	-	-	0	0
	4.3	-	3.3	0	0,083	-	-	0	0
	4.4	-	-	0	0	-	-	0	0
	4.5	-	3.4	0	0,083	-	-	0	0
	4.6	-	-	0	0	-	-	0	0
				CMCM=0,749				**CMCM=0**	
5		2	-	0,5	0	-	-	0	0
	5.1	-	2.1	0	0,125	-	-	0	0
	5.2	-	-	0	0	-	-	0	0
	5.3	-	2.2	0	0,125	-	-	0	0
	5.4	-	2.3	0	0,125	-	-	0	0
				CMCM=0,875				**CMCM=0**	
6		-	1.4	0	0,5	-	-	0	0

	6.1	-	-	0	0	-	-	0	0
	6.2	-	-	0	0	-	-	0	0
	6.3	-	-	0	0	-	-	0	0
	6.4	-	-	0	0	-	-	0	0
	6.5	-	-	0	0	-	-	0	0
				CMCM=0,5				**CMCM=0**	
7		6	-	0,5	0	-	-	0	0
	7.1	-	6.1	0	0,166	-	-	0	0
	7.2	-	6.2	0	0,166	-	-	0	0
	7.3	-	6.4	0	0,166	-	-	0	0
				CMCM=0,998				**CMCM=0**	
8		5	-	0,5	0	-	-	0	0
	8.1	-	5.2	0	0,1	-	-	0	0
	8.2	-	5.4	0	0,1	-	-	0	0
	8.3	-	-	0	0	-	-	0	0
	8.4	-	-	0	0	1	-	0,1	0
	8.5	-	5.1	0	0,1	-	-	0	0
				CMCM=0,8				**CMCM=0,1**	
		CSQMCM=5,9				**CSQMCM=1,1**			

Further, we consider the models for which the values of the CSQMCM metric were higher than the previous once according to the chronology of their emergence (Fig. 3). These models such as McCall's, IEEE, ISO 9126, ISO 25010 are landmark.

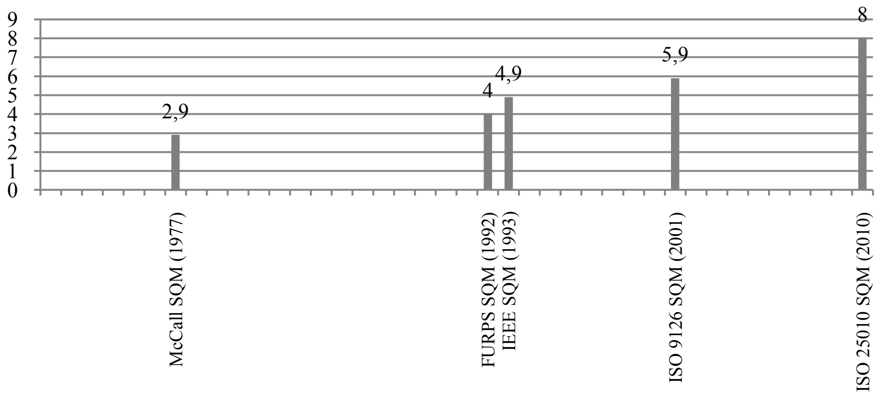

Fig. 3. Results of basic SW QM comparison in terms of CSQMCM

Let us define the functional connection between SW QM appearance year (X axis) and CSQMCM metric value (Y axis) and represent it analytically by regressive liner relationship:

$$y = ax+b, \tag{6}$$

where y is function, x - variable, a and b - regression coefficients.

The values of a and b parameters can be calculated using Least Square Method. As a result we have a = 0,153, b = 1,363 and liner dependence:

$$y = 0,153x+1,363. \tag{7}$$

To analyze the changes of SW QM on the level of characteristics let's match CMCM metric value for SW QM (table 4) and represent them in table 4.

Table 4. CMCM Metric Value for basic SW QM

SW QM / Characteristics	McCall (1977), CMCM	FURPS (1992), CMCM	IEEE (1993), CMCM	ISO 9126-1 (2001), CMCM	ISO 25010 (2010), CMCM
Functional Suitability	0	0,5	0,83	0,67	1
Performance Efficiency	0,5	0,84	0,67	0,83	1
Compatibility	0	0,5	0,75	0,5	1
Usability	0,666	0	0,58	0,75	1
Reliability	0,625	0,75	0,75	0,88	1
Security	0	0,5	0,5	0,5	1
Portability	0,5	0,334	0,67	0,99	1
Maintainability	0,6	0,6	0,1	0,8	1

5 Conclusions

As a result, structural and semantic analysis technique, based on their set-theoretic description and special metrics, is proposed. This technique allows obtaining metrics values, needed for quantitative SW QM comparison that made possible evolutionary changes of SW QM.

Considering completeness and integrity metrics two SW QM groups were defined: basic and corporative. The proposed technique can be used as a basis to adapt the existed SW QMs for software companies.

Obtained liner dependence between CSQMCM metrics and SW QM appearance year describes some pattern of liner growth of SW QM complexity.

As for further development of SW QM, it is possible to make the following predictions:

– CSQMCM metric for every next model must be higher than for the previous one, i.e. characteristics nomenclature will be wider;

– SW QM characteristics structure will become more and more complicated due to their further development and specification on the subcharacteristics level, especially for certain characteristics such as reliability, security, usability and others.

It is planned for further work:

– to conduct more detailed semantic analysis of all the characteristics (subcharacteristics) in term of their unambiguous interpretation and characteristics (subcharacteristics) duplication within the same SW QM;

– to investigate metrics evolution to assess SW quality characteristics, evaluation methods and techniques;

– to perform evolutionary analysis of SW QMs separately for dependability, security, usability, etc. for different domains.

References

1. Software engineering. Report on a conference sponsored by the NATO SCIENCE COMMITTEE. Garmisch, Germany, 7th to 11th October 1968, http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF
2. ISO/IEC 25010: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, ISO/IEC JTC1/SC7/WG6, 2011
3. Sanjay Kumar Dubey, Soumi Ghosh, Ajay Rana: Comparison of Software Quality Models: An Analytical Approach. In: International Journal of Emerging Technology and Advanced Engineering, Vol. 2 (2), pp. 111-119, February 2012
4. Stefan Wagner: Software Product Quality Control. Springer, Heidelberg (2013)
5. McCall, J. A., Richards, P. K., and Walters, G. F.: Factors in Software Quality. In: Nat'l Tech. Information Service, no. Vol. 1, 2 and 3, 1977
6. Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G., and Merritt, M.: Characteristics of Software Quality. North Holland, Amsterdam (1978)
7. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli: Fundamentals of Software Engineering. Prentice Hall, New Jersey (1991)
8. Grady R. B. Practical software metrics for project management and process improvement. Prentice Hall, New Jersey (1992)
9. IEEE, std 1219: Standard for Software Maintenance. IEEE Computer Society Press, USA (1993)
10. Dromey, G. R.: A model for software product quality. In: Transactions of Software Engineering, vol. 21, no. 2, pp. 146-162, 1995
11. Hyatt, L.E., Rosenberg L.H.: A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality. In: ESA 1996 Product Assurance Symposium and Software Product Assurance Workshop, ESTEC, Noordwijk, The Netherlands, European Space Agency, 209–212
12. Bansiya, J. Davis, C.: Hierarchical Model for Object-Oriented Quality Assessment. In: IEEE Transactions on Software Engineering 28, 4-17 (2002)
13. ISO/IEC 9126-1 Software engineering – Product quality – Part 1: Quality model (2001)
14. Gordeyev, A., Kharchenko, V., Andrashov, A., Sklyar, V., Konorev, B., Boyarchuk, A.: Case-based Software Reliability Assessment by Fault Injection Unified Procedures. In. proc. of Software Engineering in East and South Europe (SEESE'08), May 13, 2008 Leipzig, Germany, pp. 1-8. ACM, New York (2008)