Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Evaluation of the Quality of a Program Code for High Operation Risk Plants

Elena Ph. Jharko

V.A. Trapeznikov Institute of Control Sciences, 65 Profsoyuznaya, Moscow 117997, Russia (Tel: +7 495 334 8990; e-mail: zharko@ipu.ru).

Abstract: The present paper is, first of all, oriented to the internal evaluation of the quality of a program code, based on the conception of evaluation of the software quality using both the series of international standards ISO/IEC 9126 and Russian standards in the branch of the software quality as a system of quality criteria and a set of metrics applied to evaluating a program code. *The paper has been supported by a grant of the Russian Foundation for Basic Researches (RFBR): project 12-08-01205-a.*

Keywords: verification, quality assurance, program code, software, high operation risk plants, nuclear power plant

1. INTRODUCTION

Industrial process control systems have achieved a qualitatively new level of the development concerned with the increased degree of automation of control plants and, as a consequence, increasing the number of diagnostic and control signals processed by the control system per time unit. From another hand side, practically linear growth of the capacity of computing systems that may be applied within an APCS has enabled one to implement considerably more complicated algorithms of control and data analysis by use of complex soft- and hardware tools.

A software (SW) for high operation risk systems, having a large size of both the program code and software documentation, has, in the majority of cases, problems of the maintenance, while Customers of such a software are expecting the high quality of the software product. However, the tasks of the evaluation and assurance of the SW quality are hard due to ambiguous understanding the software quality. The ISO/IEC 9126 standard (ISO/IEC 9126-1:2001, ISO/IEC TR 9126-2:2003, ISO/IEC TR 9126-3:2003, ISO/IEC TR 9126-4:2004) has been developed to resolve issues concerned with the software quality, and it describes a software product with characteristics and sub-characteristics of the quality and proposes criteria and metrics to evaluate them. This standard is universal one and may be applied to any type of the software under accounting particularities of a SW.

The quality assurance is a continuous process in the course of the whole SW life cycle, which covers:

- Methods and tools of the analysis, design, and coding;
- Technical reports being implemented at the each step of the software development;
- Procedure of multi-level testing;
- Monitoring the software documentation and changes introduced in it;
- Procedures of assuring the correspondence to standards in the branch of the software development, meeting which is defined in the assignment on specific SW development;
- Algorithms of measurement and forming reports.

The main task of increasing the SW quality is based on the adoption in organizations-developers of the software life-cycle processes: “Software Quality Assurance” (SQA), “Quality Management” (QM), “Verification and Validation” (V&V).

SQA is concerned with two kinds of the activity:

- Adoption of quality standards and corresponding procedures in the development of software complexes;
- Evaluation of the commitment to these standards and procedures.

Starting from 1995, this activity is regulated by the international standard of ISO/IEC 12207 (at present, ISO/IEC 12207:2008 is acting). SQA investigation objects are, basically, life-cycle processes of software complexes rather than software products. To monitor the software product quality, the V&V process is intended. In Fig. 1, a place of the V&V process in the assurance of the quality of systems being important for the nuclear power plant (NPP) safety is displayed (Jharko, 2011b; Jharko, 2013).

An experience of developing soft- and hardware complexes (Byvaikov et al., 2006, Poletykin et al., 2006, Jharko, 2008, Jharko et al., 2011a) gathered has enabled one to analyze the SW development process to the reveal a number of problems concerned with the process of integration of software complexes at the stage of their development, solving which is achievable by implementing the software verification and validation in the course of the whole SW life-cycle. Meanwhile, the stage of creating the program code is of particularly emphasized.

Copyright © 2014 IFAC

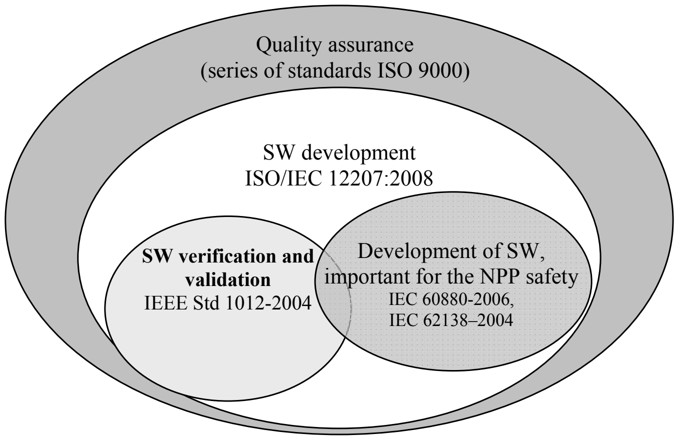


Fig. 1. The place of V&V of the software in the quality assurance of SW important for the NPP safety

The present paper is, first of all, oriented to the internal evaluation of the program code quality, based on the conception of the software quality evaluation in accordance to both the series of international standards of ISO/IEC 9126 and Russian standards in the branch of the software quality as systems of quality criteria and a set of metrics applied to evaluate a program code. The conception of the quality evaluation involves a three-stage approach associating, at the system, level characteristics of the code quality, criteria and metrics in order to reflect the importance of indicators under evaluation of the software quality indicators (Jharko, 2011).

2. THE SOFTWARE QUALITY

The software quality is defined in the standards of ISO/IEC 9126-1:2001 and ISO/IEC 25010:2011 as any totality of software characteristics related to the possibility to meet refereed or assumed consumptions of all interested parties.

One distinguishes the notions of the internal quality, concerned with SW characteristics as itself, disregarding its behavior; the external quality that characterizes SW from the point of view of its behavior; and the SW quality under use in different contexts, the quality that is filled by users under specific scenario of the SW performance. For all these aspects of the quality, metrics were introduced that enable to evaluate them. Besides that, to create a dependable SW the quality of technological processes of its development is very essential. The interconnections between these aspects of the quality in accordance to the scheme adopted by ISO/IEC 9126 is displayed in Fig, 2.

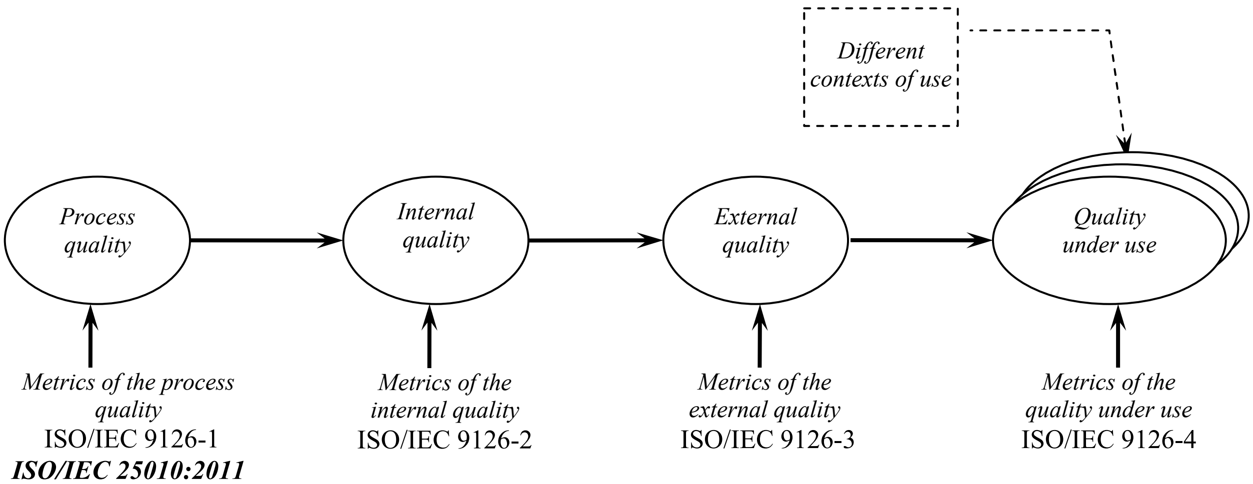


Fig. 2. Basic aspects of the software quality in accordance to standards of ISO/IEC 9126-1:2001 and ISO/IEC 25010:2011

Table 1. The order of evaluation of the quality of software products

Parties interested in quality evaluation	Stages of the life cycle of a software					
	Development	Tests	Replication	Adoption	Maintenance	Operation
Developer	Yes	Yes	Yes	Yes	Yes	Yes
Test and certification centers	-	Yes	-	Yes	-	Yes
User	-	-	-	-	-	Yes

In table 1, the order of the software quality evaluation is presented. The SW quality may be considered as "good enough", when potentially positive results of the creation or use of a SW prevails potentially negative opinions of customers. Such an approach checks up, from the point of view of the conventional SW quality notion, different variants of implementations. Under such an approach to the SW quality, high unverified requirements are substituted with optimal ones. This approach is focused on initiating tasks and on improving possibilities for decision making. Thus, the design of SW development for high operation risk plants is to be sooner problem-oriented rather than purposeful to the SW quality. Also, one may be said that the SW quality, in accordance to the notion of "good enough", is the optimal set of solutions for this series of problems. Such a way of the interpretation is to coordinate the considered problems, is to elaborate compromise variants, setting them against the corresponding life-cycle processes (ISO/IEC 12207:2008).

3. SOFTWARE QUALITY EVALUATION

The maturity of the development process of soft- and hardware tools of a system (Goodman, 2005; Madachy, 2008) may be characterized as a degree of clearness of defining, management, measurement, monitoring, and implementation of the software development process. The SW quality is "a totality of object characteristics that are related to its ability to meet set and assumed requirements". Using the term "meeting", the ISO/IEC 9126 standard implicates "software possibilities to meet users in a given context of use". Fig. 3 displays factors and attributes of the external and internal software quality in accordance to ISO/IEC 9126. In Fig. 4, the model of evaluation in accordance to ISO/IEC 9126 is presented. In the paper of Dubey et al. (2012), a comparison of different models of the software quality evaluation has been performed.

The conception of the evaluation of the program code quality is oriented to the internal quality, which is defined by the six characteristics:

1) Functionality, which is concerned with the fact what software does to implement requirements of users, both set and assumed ones;
2) Dependability, which is concerned with the evaluation of abilities of software to maintain a certain level of the performance quality;
3) Usability (convenience of use), which is concerned with the evaluation of how understandable and suitable considered software is for use;
4) Effectiveness, which is concerned with the evaluation of abilities of software to provide a required capacity related to the quantity of required resources;
5) Maintainability, which is the evaluation concerned with abilities of modification of software;
6) Mobility, which is an evaluation of the dependence of software from basic software.

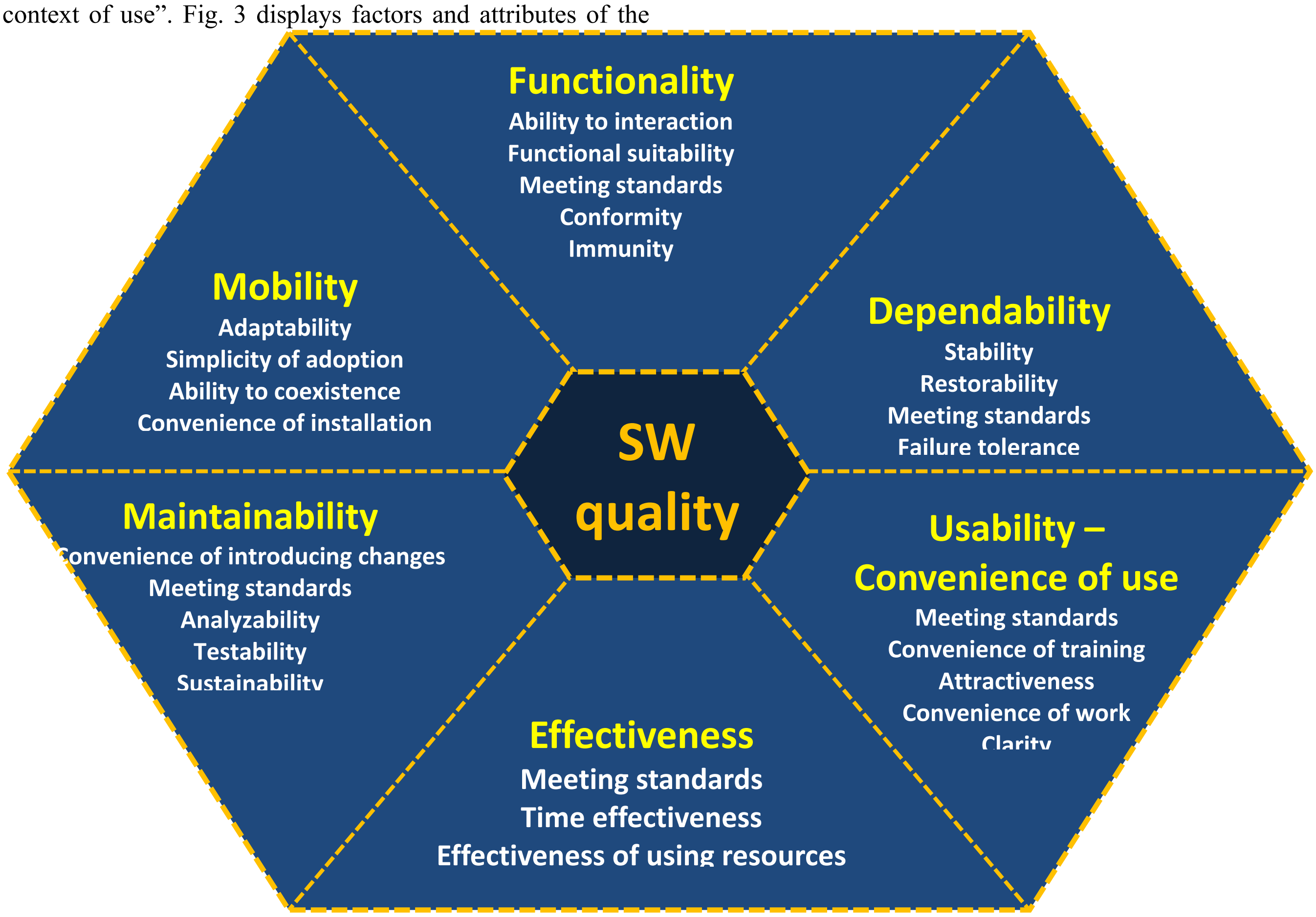


Fig. 3. Factors and attributes of the external and internal software quality in accordance to ISO/IEC 9126

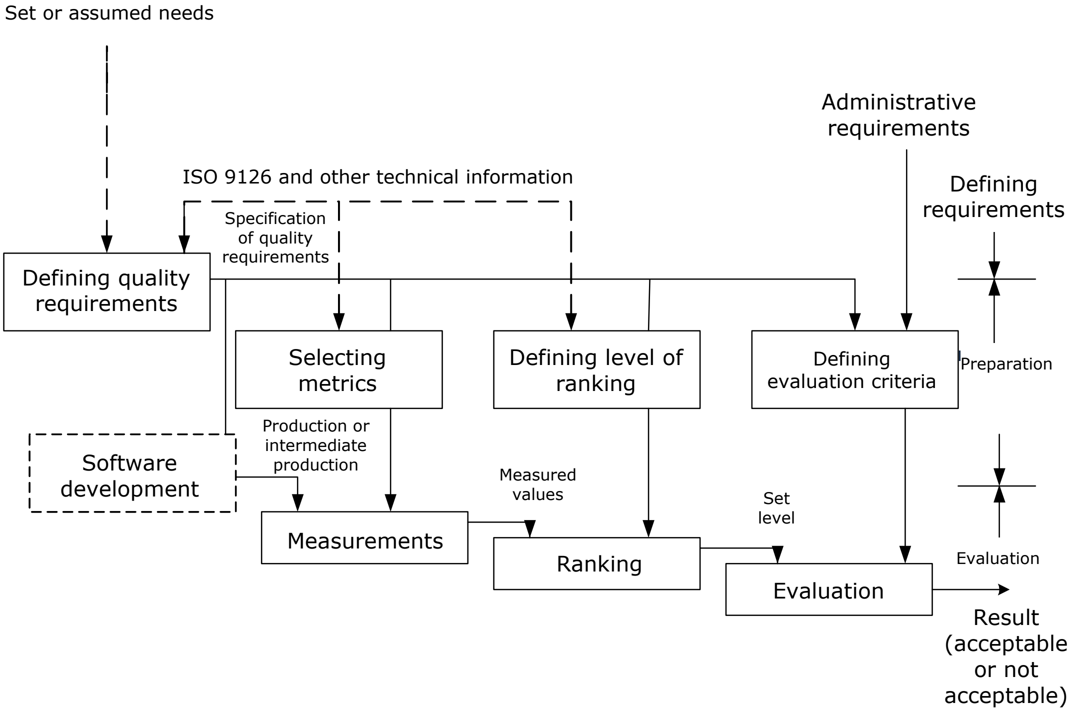


Fig. 4. Evaluation process model presented in *ISO/IEC 9126*

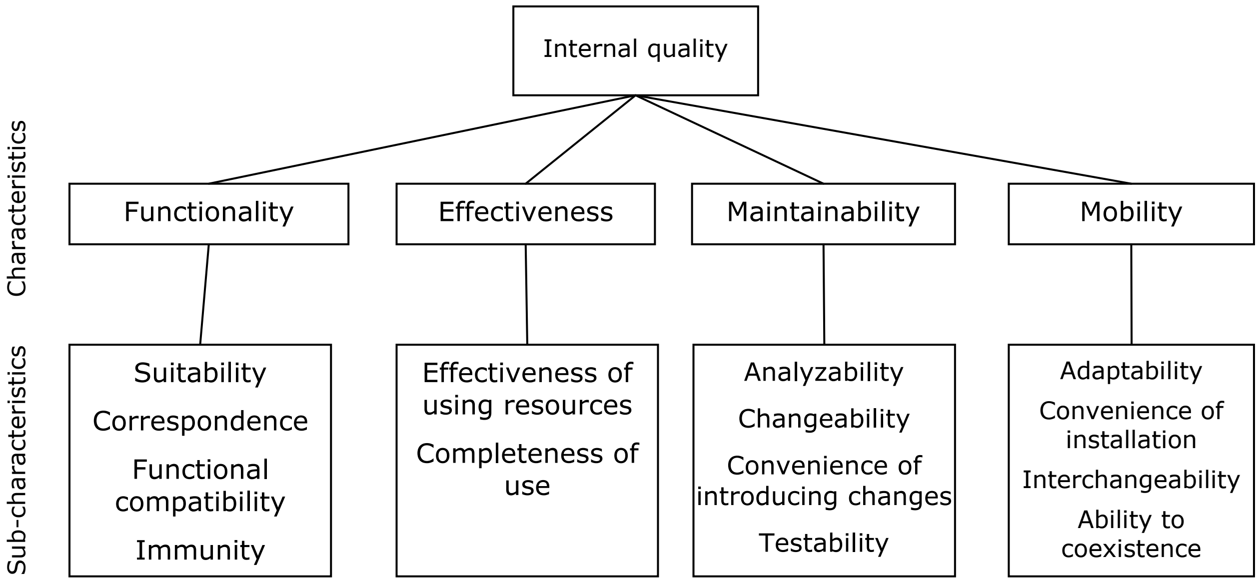
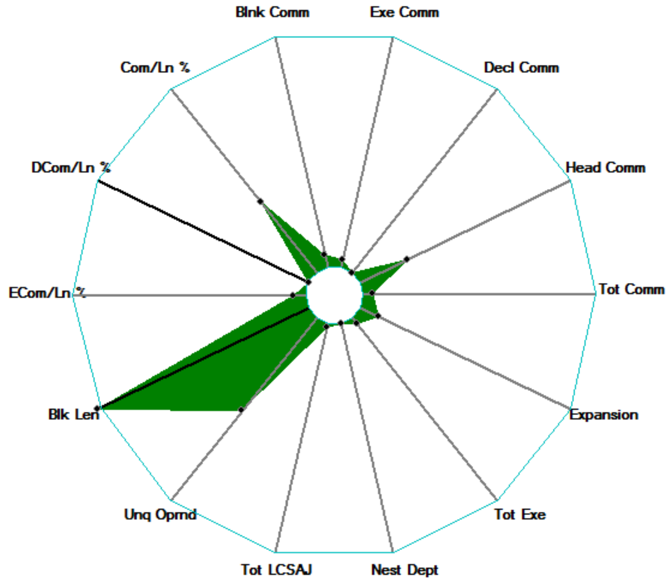


Fig. 5. External characteristics and sub-characteristics in accordance to the series of standards of *ISO/IEC 9126*

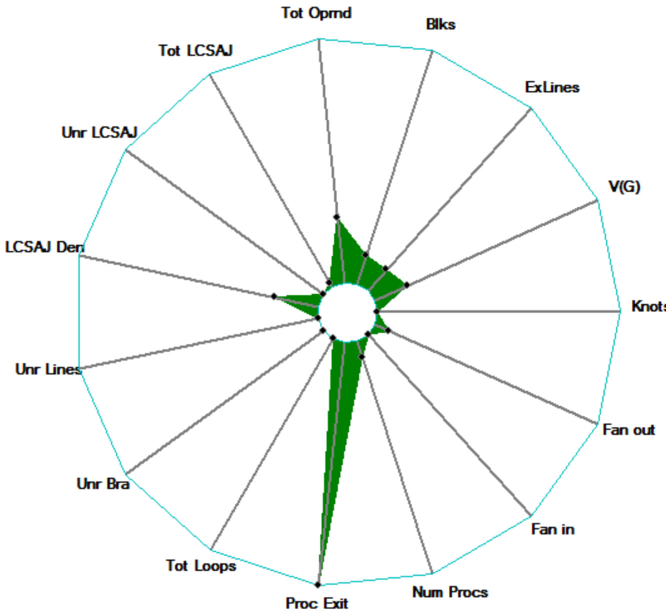
Since the branch of application of the conception is the program code evaluation, then metrics are used only that are directly related to the source code; and, as a result, some of these six characteristics, in accordance to ISO/IEC-9126, are not suitable for the evaluation. In particular, such characteristics as dependability and convenience of use (usability) are excluded from the consideration of the program code quality, since these are more concerned with the dynamic behavior of the system as a whole. Thus, the internal quality of a program code is evaluated by use of four characteristics (functionality, effectiveness, maintainability, and mobility) and sub-characteristics corresponding to them. Each of characteristics is characterized by a set of sub-characteristics that are presented in Fig. 5. This complex of the qualitative characteristics is general enough to meet the main purpose of the present paper, which is creating a conception that maintains the system of the program code quality evaluation.

In Fig. 6 and 7, examples of analysis of a program module code over the sub-characteristics *"Clarity"* (Usability) и *"Testability"* (Maintainability) correspondingly are presented.



Designation	Name
Tot Comm	Total Comments
Head Comm	Comments in Headers
Decl Comm	Comments in Declarations
Exe Comm	Comments in Executable Code
Blnk Comm	Blank Lines
Com/Ln %	Total Comments/Exe. Lines
DCom/Ln %	Declaration Comments/Exe. Lines
ECom/Ln %	Code Comments/Exe. Lines
Blk Len	Average Length of Basic Blocks
Unq Oprnd	Unique Operands
Tot LCSAJ	Number of sequences of the code and transfers
Nest Dept	Depth of Loop Nesting
Tot Exe	Executable ref. Lines
Expansion	Expansion Factor

Fig. 6. Analysis of a program module over the sub-characteristics *"Clarity"*



Designation	Name
Knots	Knots
V(G)	Cyclomatic Complexity
ExLines	Executable reformatted Lines
Blks	Number of Basic Blocks
Tot Oprnd	Total Operands
Tot LCSAJ	Number of sequences of the code and transfers
Unr LCSAJ	Number of unreachable sequences of the code and transfers
LCSAJ Den	Maximal density of sequences of the code and transfers
Unr Lines	Unreachable Lines
Unr Bra	Unreachable Branches
Tot Loops	Number of Loops
Proc Exit	Procedure Exit Points
Num Procs	Number of Procedures

Fig. 7. Analysis of a program module over the sub-characteristics *"Testability"*

The complexity of software for high operation risk plants puts forth more and more strict requirements to the quality and reliability of such software. To meet the requirements, one should organize a number of activities directed to both improving the software development process, and to assess the efficiency of their performance. Factors that influence the software quality may be related to:

– Applying the system of monitoring versions under the development;

– Organization of the process of control of requirements;

– Availability of the process of control of changes;

– Optimal type of integration of software components into the unique system;

– Modular testing;

– Automatic calculation and analysis of metrics at the stage of assembling, characterizing the source code quality.

For each software complex, statistical analysis is implemented, involving the calculation of the number of lines (LOC – Lines of Code), using branch statements, etc., as well as the number of errors appearing in the course of the product life-cycle, fixed in systems of control of measurements.

The number of errors contained in software, as a rule, depends on the size of source code and the number of branch statements, and this fact is used to calculate certain metrics characterizing both the software product quality, and the process of software manufacturing. Such statistical data gathered enable one to implement a comparative analysis of the of the efficiency of activities on improving the software quality. As such statistical quality indicators, consistent measures of (possibly, multiple) dependence are the most appropriate to be used. The measures are to associate such indicators as the integral number of errors on the project with different measurements of the code (from the number of lines of the code and commentability to a cyclomatic complexity indicator).

4. CONCLUSIONS

At present, scientific literature in the branch of software engineering does not pay due attention to the formalization of different verbal models and technologies of the evaluation and control of the quality of the process of software development, and, basing on it, to the investigation of the branch of application of each of them.

The practice of applying the comprehensive evaluation of the program code quality for high operation risk plants, in particular, such as nuclear power plants, has shown that such an approach, finally, enables one to implement changes of the internal program structure, which do not influence its external behavior, and which are aimed to simplify understanding the software performance, what is positively reflected in the quality of developed software, in particular, in such indexes as dependability, modifiability, maintainability, and usability.

REFERENCES

Byvaikov, M.E., Zharko, E.F., Mengazetdinov, N.E., Poletykin, A.G., Prangishvili, I.V., and V.G. Promyslov (2006). "Experience from design and application of the top-level system of the process control system of nuclear power-plant", *Automation and Remote Control*, vol. 67, no. 5, pp. 735-747.

Dubey, Sanjay Kumar, Ghosh, Soumi, Rana, Ajay (2012). "Comparison of Software Quality Models: An Analytical Approach", *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 2, pp. 111-119.

Goodman, F.A. (2006) *Defining and deploying software processes*, 221 p. Auerbach Publications.

ISO/IEC 9126-1:2001. Software engineering – Software product quality – Part 1: Quality model.

ISO/IEC TR 9126-2:2003 Software engineering – Product quality – Part 2: External metrics.

ISO/IEC TR 9126-3:2003 Software engineering – Product quality – Part 3: Internal metrics.

ISO/IEC TR 9126-4:2004 Software engineering – Product quality – Part 4: Quality in use metrics.

IEC 60880 Ed. 2, 2006. Nuclear power plants – Instrumentation and control systems important to safety. Software aspects for computer-based system performing category A function. 2006

IEC 62138 Ed. 1, 2004 Nuclear Power Plants – Instrumentation and Control Computer-based systems important for safety. Software for I&C systems supporting category B and C functions. 2004.

IEEE Std 1012:2004. IEEE Standard for Software Verification and Validation. 2004.

ISO/IEC 12207:2008. Systems and software engineering - Software life cycle processes.

Jharko, E.Ph. (2008). Design of Intelligent Information Support Systems for Human-Operators of Complex Plants, *Proceedings of the 17th IFAC World Congress. Seoul, Korea, July 6-11, 2008*. pp. 2162-2167.

Jharko, E.Ph., Zaikin, O. (2011a). The Flexible Modeling Complex for an NPP Operator Support System, *Proceedings of the 18th IFAC World Congress. Milano, Italy, August 28 - September 2, 2011*. pp. 12156-12161.

Jharko, E.Ph. (2011b). "Assessment of the software quality of systems important for the NPP safety", *Information Technologies and Computing Systems*, no. 3, pp. 38-44. (in Russian)

Jharko, E.Ph. (2013). "Quality Assurance for Nuclear Power Plant Control System Software", *Proceedings of the 7th IFAC Conference on Manufacturing Modelling, Management and Control. Saint Petersburg, Russia, June 19-21, 2013*. pp. 1102-1107.

ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.

Madachy, R.J. (2008) *Software process dynamics*, 601 p. Hoboken, NJ: IEEE Press-Wiley Interscience.

Poletykin, A.G., Jharko, E.Ph., Zuenkova, I.N., Promyslov, V.G., Byvaikov, M.E., and N.E. Mengazetdinov (2006). "Software for nuclear power engineering", *Automation in Industry*, no. 8, pp. 52-56. (in Russian)