7th IFAC Conference on Manufacturing Modelling, Management, and Control
International Federation of Automatic Control
June 19-21, 2013. Saint Petersburg, Russia

Quality Assurance for Nuclear Power Plant Control System Software

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: Most important issues of implementation works on quality assurance of software for plants of high risk operation by use of an example of software for nuclear power plants are considered.

Keywords: NPP, Upper-level control system, Software quality

1. INTRODUCTION

The process of development of automation of complex technological plants with high operation risk in the power engineering and other branches of industry is characterized by a tendency of development and adoption in the make-up of regular tools of upper level of automated process control systems (APCS) of systems of operator information support (Byvaikov et al., 2006, Poletykin et al., 2006).

In the last decade, automatic process control systems have led to a qualitatively new level of development. Such a level is concerned with an increased level of the automation of control plants and, as a consequence, a growth of the number of control and diagnostic signals processed by the control system per time unit. From another hand side, practically linear growth of the capacity of computer systems that may be used in APCS has enabled one to implement considerably more complex algorithms of control an analysis of data by use of high-performance soft- and hardware tools on computations. However, the qualitative jump in the make-up of solved problems, which has been the case, made one to reconsider the relationship components of the life cycle of the software.

Such changes are clearly traced by use of an example of development of software for NPP APCS with required life time being not less than 30 years. This considerably exceeds the average time of life and storing of hardware, achieved at present, and makes one to pay more attention to careful development of the stage of modification and maintenance of software (SW) developed (Jharko, 2011).

Software quality assurance is a persistent process within whole life cycle of SW, which covers:

- methods and tools of analysis, design, and coding;
- technical reports, implemented at each stage of SW development;
- procedure of multi-level testing;
- monitoring software documentation and changes introduced into the documentation;
- procedures of assurance of correspondence to standards in the branch of the SW development, correspondence to which is defined in the assignment on developing this SW;
- algorithms of measurements and composing reports.

The software quality may be defined as correspondence to explicitly stated functional and operational requirements, explicitly indicated standards of the development, and to implicit characteristics that are expected from professionally developed software. Such a definition of the software quality underlines three important circumstances:

- requirements to the software is a basis with respect to which the SW quality is determined;
- these standards define the set of criteria, which defines the SW development style;
- there exists a manifold of implicit requirements, which are not frequently mentioned about (for instance, maintainability and updatability). If a software meets to explicit requirements to its development but is not in position to meet explicit requirements, then the SW quality is doubtful.

These circumstances are most sharply traced with regarding software of highly reliable systems, to which, in particular, NPP APCS subsystems are related to, since besides complete correctness, the software possesses other characteristics being of interest to a consumer of this SW, such as absence of errors under execution, integrity of data, time characteristics, accuracy, correctness of types, completeness, functional reliability, safety, maintainability, intelligibility, updatability, and others.

2. CLASSIFICATION OF SYSTEMS IMPORTANT FOR THE NUCLEAR POWER PLANT SAFETY

Under development of systems for power engineering, where the operation period of the main equipment is dozens of years, one should apply such solutions in the APCS, which would enable one to operate, repair, and update the installed equipment without stopping the technological process. Besides this requirement, providing high reliability, survivorship, and safety are the key requirements.

Analysis of advanced requirements, present status of hardware and software, tendencies of development enabled one to formulate a common approach to constructing systems for the power engineering: the systems are to be constructed

either by use of own technologies, or by use of imported technologies. Meanwhile, the technologies imported are to be subject to an adaptation process that is to make them transparent and controllable to such a degree so as a supplier could expand his/her warranty obligations of duration of several dozens of years to them.

In accordance to the international classification (Byvaikov et al., 2006), systems important for the NPP safe are separated from the point of view of functions implemented by these systems:

Category A involves functions that play the main role in achieving or supporting the NPP safety in order to prevent development of emergencies to inadmissible consequences.

Category B – involves functions that play supplementary role with regard to the functions of category A in achieving or supporting the NPP safety, in particular the functions that are needed for operation under achieving a controlled status in order of preventing development of design events (DE) to inadmissible consequences or to moderate DE consequences.

Category C involves functions that play an auxiliary or indirect role in achieving or supporting the NPP safety.

Basic principles of development of control systems important for the NPP safety have found the reflection in international standards (Poletykin et al., 2006, Jharko, 2011). There exists no unique classification of NPP systems. Table 1 present a comparison of safety classes of NPP systems presented in regulatory documents. In dependence on a safety class, software developed for these systems is imposed with limitations concerned with suitability of operation systems, programming languages, detailedness of documenting, etc.

The NPP APCS make-up involves systems of the 2nd, 3rd, and 4th safety classes in accordance to NP-001-97 or in accordance to the international classification (IEC 61226 ed3.0, 2009) (see table 1) systems of classes A, B, C. Thus, under development of software for NPP APCS subsystems one should follow to standards of (IEC 60880 Ed. 2, 2006) (for systems of class A), (IEC 62138 Ed. 1, 2004.) (for systems of classes B, C).

3. THE DEFINITION OF THE SOFTWARE QUALITY

The software quality is defined in standards ISO/IEC 9126-1:2001 and ISO/IEC 25010:2011 as every totality of its characteristics relating to a possibility to meet declared or assumed needs of all interested parties/persons.

One separates the notion on the internal quality, concerned with SW characteristics itself, disregarding its behavior, and the notion of the external quality, characterizing the SW from the point of view of its behavior, and the SW quality under its use within different contexts, the quality that is filled by users under specific scenarios of SW performance. For all these aspects of the quality, metrics were introduced enabling one to assess them. Besides that, to create a durable SW, the quality of technological processes of its development is essential. The interconnection between these aspects of the quality in accordance to the scheme adopted by ISO 9126 (ISO/IEC 9126-1:2001; ISO/IEC TR 9126-2:2003, ISO/IEC TR 9126-3:2003, ISO/IEC TR 9126-4:2004) is shown in Figure 1.

The SW quality may be considered as «good enough», when potentially positive results of creating or use of the SW acceptably overbalance potentially negative opinions of customers. Such an approach checks, from the point of view of the traditional SW quality notion, different variants of the implementation. Under such an approach to the SW quality, high not checked requirements are substituted with optimal ones. This approached is focused on identifying problems and improving possibilities for the decision making. Thus, the design of SW development for plants with increased risk of operation is to be mostly problem-oriented, rather than SW quality purpose-oriented. Also, one may say that the SW quality, in accordance to the notion of «good enough», is the optimal set of solutions for a given series of problems. Such a way of interpretation is to coordinate considered problems, elaborate compromise variants, contrasting them to corresponding processes of the life cycle (ISO/IEC 12207:2008). A typical order of assessment of the quality of software products, formed at advanced conditions, is presented in Table 2

Table 1. Comparison of safety classes of NPP systems

Standard or regulatory document	Safety classes (the importance degree increases from the left to the right)				
NP-001-97	Class 4	Class 3	Class 2	Class 1	
IAEA NS-R-1:2000	Systems not important for the safety	Systems important for the safety			No
		Systems concerned with the safety	Safety systems		
IEC 61226:2009	Not classified	Class C	Class B	Class A	No
IEEE 603:1998	Not class 1E			Class 1E	No

Table 2. The order of assessment of the quality of software products

Parties interested in quality assessment	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**

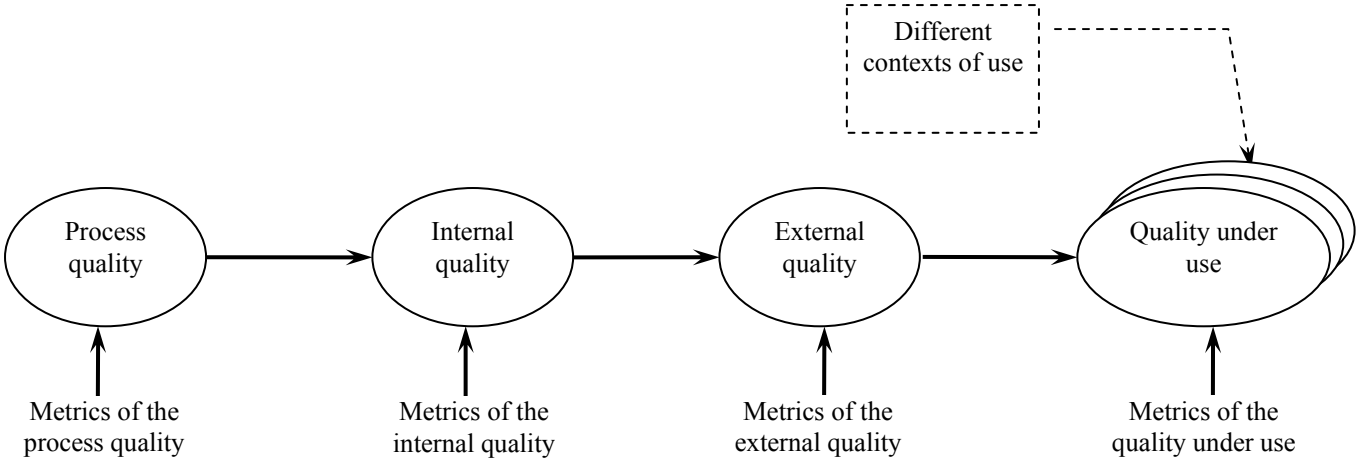

Fig. 1. Basic aspects of the software quality in accordance to standards of ISO/IEC 9126-1:2001 and ISO/IEC 25010:2011

The software quality definition helps:

- to assess software products;
- to assess principles of organization of the software;
- to improve processes of creating the software.

One separates the following stages of assessment of the software quality:

- determining characteristics of the software quality;
- development of indicators to determine the characteristics of the quality;
- recording values and comparison with preceding values;
- introducing changes into the software to improve its quality.

In Figure 2, the process of «measurement» of a software product is displayed. It is impossible to measure some characteristics directly; however these may be measured via the quality indicators. In Figure 3, the dependence between characteristics and indicators of the software is displayed.

Conditions that are to be met under measuring the characteristics of the software via the quality indicators:

- the quality indicators are to be measure accurately;
- the dependence is to be expressed in the form of a formula or model.

Quality indicators have two types: indicators of control and indicators of prediction.

The indicators of control are used by management to manage the process of development of the software. These indicators are provided with information on the process quality. These are not particular for the software: any processing manufacture may be managed and monitored in accordance to such indicators. The indicators of control involve, for instance, the size of works, calendar time spent or use factor for partial tasks, percentage of operators checked, etc.

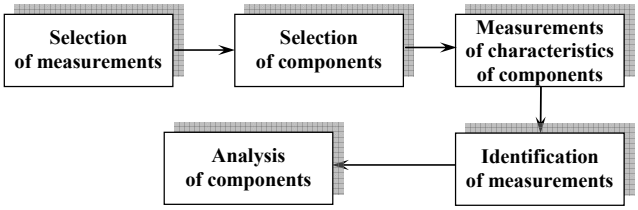

Fig. 2. The process of measurement of a software product

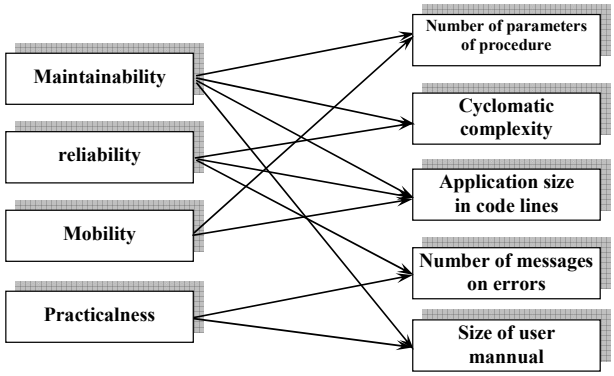

Fig. 3. The interconnection between indicators and characteristics of the quality

The indicators of prediction determine product characteristics that predict the product quality. Quality characteristics are

predicted, if corresponding to them indicators are determined. One separate two type of the indicators of prediction: dynamic and static indicators.

The dynamic indicators collect measurements made within the process of execution of an application for the efficiency and reliability. These are closely concerned with the software quality and are convenient for determination. The efficiency may be calculated on the basis of measuring the execution time, while the reliability may be calculated on the basis of the number of faults of the system and types of the faults.

The static indicators collect measurements made within the process of representation of the system to assess complexity, intelligibility, and maintainability. These indicators have indirect connection with the software quality, which assumes that there exist dependence between the characteristics and indicators of the quality.

4. SOFTWARE QUALITY ASSURANCE

The software provides a considerable impact into functions implemented by systems important for the safety. The software may support supplement functions introduced in accordance to the design of a developed or already performed system (for instance, initialization and monitoring of hardware, connection between subsystems, etc.). For systems important for the NPP safety, the life cycle of the software safety is closely concerned with the life cycle of the safety of the system itself. As well as specification of requirements to the software is a part of the system specification. In Figure 4, works relating to the SW in the life cycle of the system are displayed.

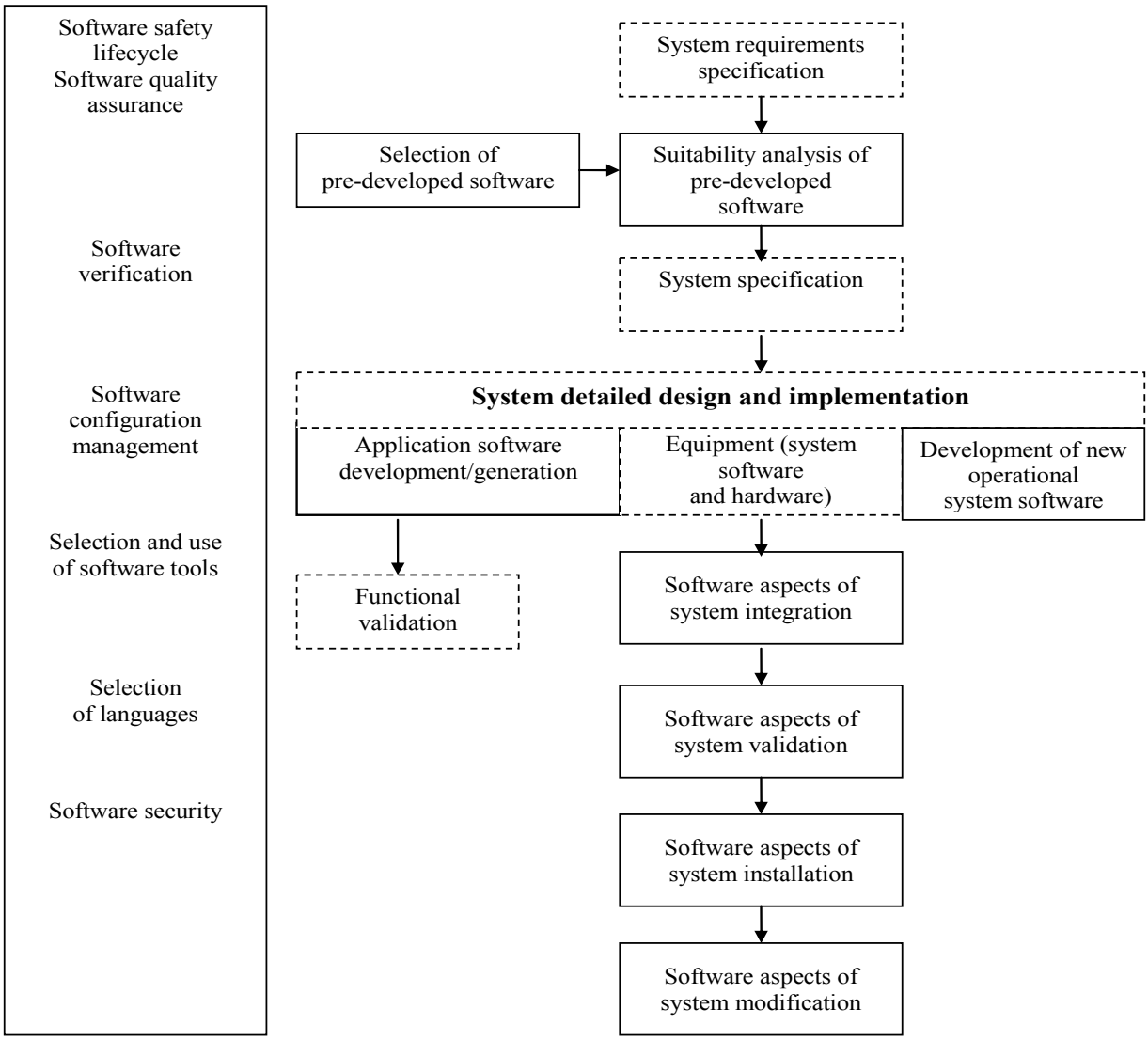

Fig. 4. Software related activities in the system safety lifecycle

The required software quality is hardly achievable since the process of obtaining a required SW quality is concerned with the process of development, methods and control of the process. The SW quality is achieved due to applying the development methodology and applying methods of verification and validation during the life cycle of the SW development for systems important for the NPP safety. In Figure 5, the place of verification and validation of the software in the context of the quality assurance and the hierarchy of standards is displayed.

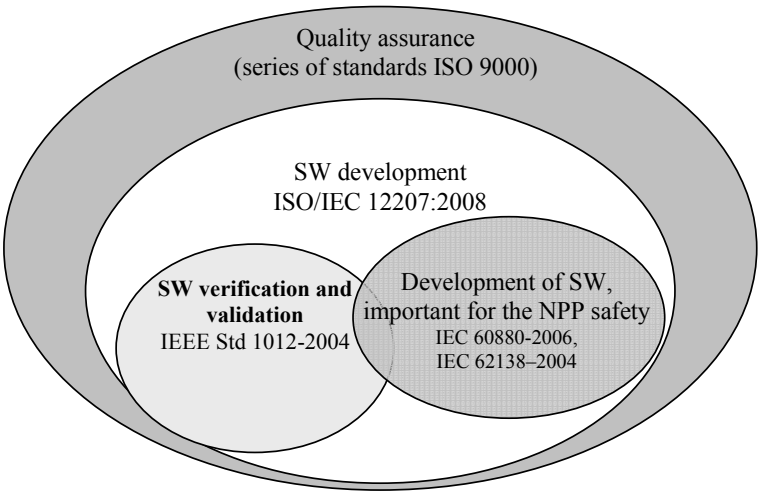

Fig. 5. The place of verification and validation of the software in the quality assurance of SW important for the NPP safety

5. THE METHODOLOGY OF THE SOFTWARE QUALITY ASSURANCE

V.A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences (ICS RAS) implements works on verification of software of systems important for the safety, relating to safety classes 2-4 in accordance to the classification of NP-001-97. The works on verification of the software of such systems are necessary to achieve the required quality of developed and modified systems.

The main purpose of verification of the considered software products is assurance of the quality of implementation of requirements to the software of systems important for the safety. The quality assurance of implementation of requirements to SW of systems important for the safety at the stages of its creation, such as development of requirements to SW, design of SW, and development of SW, is to involve checking the coordination of results obtained at each stage of the development with requirements set at preceding stages, and obtaining data providing the analysis and assessment of quality indicators of implementation of the requirements.

The verification provides:

- meeting initial requirements;
- revealing errors and drawbacks at earl stages of the design, their analysis and removing;
- decreasing expenses and sources to improve errors at subsequent stages of the life cycle.

The verification purposes are achieved by sequential implementing a combination of view, analyses of documentation created in the course of development of the software.

The analysis provides the proof of correctness of developed SW and enables one to investigate in details the functionality, practicalness, and maintainability of SW components.

Methods of the analysis and view (review) are applied for the verification of the design documentation.

The view (review) provides a qualitative assessment of the degree of correctness of developed SW and may involve methods of inspection and checking. The method of inspection is consideration of the documentation against presence of errors and abnormalities. The method of inspection is revealing a correspondence to documentation developed at the preceding stage of the design.

The views and analyses are implemented from top to down, starting with general requirements in the technical assignment to detailed requirements on modules and their interaction. The views and analyses provide an assessment of the accuracy, completeness, and verifyability of requirements, SW architecture, as well as source codes of applications.

Application texts (source application texts) are verified by the methods of view and analyses on meeting the following criteria:

- source texts of applications meet the design and requirements, are testable and correspond to requirements of programming standard;
- applications reflect the actual run of events, completeness, location of temporary and computational sources, determining errors, their detection and recovering the operability;
- applications descend from the design and requirements to the system;

– applications provide implementation of requirements concerned with the safety (requirements to SW functions, SW, diagnosis and self-monitoring, providing protection of faults and distortion, SW development process);

– source application texts contain enough comments for modification and maintenance, and are readable.

The verification methodology presented provides the high quality of software both developed by the ICS RAS and external organizations for systems important for the NPP safety.

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.

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)

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

IEC 61226 ed3.0, 2009. Nuclear power plants – Instrumentation and control important to safety – Classification of instrumentation and control functions. 2009.

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.

NS-R-1:2000. Safety of Nuclear Power Plants: Design Safety Requirements. IAEA Safety Standards Series No. NS-R-1. 2000.

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

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

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

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

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.

NP-001-97 (PNAE G-01-011-97). *General guidelines on safety assurance of nuclear power plants. OPB-88/97*, Gosatomnadzor Rossii, Moscow, 1997.