Information Technology for Assurance of Veracity of Quality Information in the Software Requirements Specification

Conference Paper *in* Advances in Intelligent Systems and Computing · September 2018

DOI: 10.1007/978-3-319-70581-1_12

1 author:

Tetiana Hovorushchenko
Khmelnytskyi National University

Information Technology for Assurance of Veracity of Quality Information in the Software Requirements Specification

Tetiana Hovorushchenko[1[0000-0002-7942-1857]]

[1] Khmelnitsky National University, Khmelnitsky, Institutska str., 11, 29016, Ukraine
tat_yana@ukr.net

Abstract. The aim of this study is the development of information technology of evaluating the sufficiency of quality information in the software requirements specification (SRS) for assurance of veracity of quality information in the SRS. This study is also devoted to design and research of the subsystem of evaluating the sufficiency of the SRS information for software quality assessment based on the comparative analysis of ontologies. The developed information technology and subsystem provide: evaluating the sufficiency of the SRS information for software quality assessment by the standard ISO 25010:2011 and based on the metric analysis; identifying the missing (in the SRS) measures and (or) indicators (if the SRS information is insufficient); prioritization of the addition of the missing measures and (or) indicators in the SRS; quantify evaluating the veracity of the available in the SRS information for software quality assessment; increasing the veracity of the quality information in the SRS; increasing the software quality assessment at the early lifecycle stages.

Keywords: Software, Software Requirements Specification (SRS), Software Quality, Software Quality Information, Sufficiency of Quality Information, Ontology.

1 Introduction

Today almost all spheres of human activity are connected with information systems, the basis of which is software. A key factor in ensuring the effective using of software products and one of the main user requirements to modern software is to achieve high values of its quality. The need to ensure the quality of software follows from the fact that software bugs and failures threaten by catastrophes resulting in human casualties, environmental disasters, significant time and financial losses.

As statistics [1-5] show, there are currently problems in the field of software quality assurance – the large projects are still performed with the lag of schedule or cost overruns, the developed software products often lack the necessary functionality, their performance is low, and the quality doesn't suit consumers.

A large number of software bugs occurs at the stage of requirements formation and formulation – these errors constitute 10-23% of all bugs, and the greater the size of software, the more errors are made in the stage of requirements formation and formulation [2, 6]. The vast majority of software-related crashes occurred due to false requirements, not because of coding bugs [2, 6]. The earlier the defect (bug, trouble, drawback, malfunction) will be revealed, the cheaper it will cost its correction – the cost of correcting the incorrect requirements of the specification, discovered after the release of the product, is almost 100 times the cost of correcting the defects of the specification, assumed in the process of formation and formulation requirements [4].

In the process of formating and formulating the requirements there are the information losses due to incomplete and different understanding of the needs and context of information – especially these losses are significant for software projects that are developed at the junction of subject domains (for example, software for medicine), when it is necessary to consider as standards for development of software, and the standards of the subject domain, for which software is being developed. It's difficult to implement such standards, and it is even more difficult to verify the degree of consideration of the recommendations of these standards.

Software projects with incomplete requirements and specifications cannot be successful [2]. Under such circumstances, the analysis of the SRS, the ability to "cut off" the software projects with the incomplete (with insufficient information) specification is the actual and very important task. Sufficiency of information is one of the most important aspects of software quality assessment. The quality and success of the software project implementation significantly depend on the SRS, and on the sufficiency of the SRS information (the presence of all the information elements, which are necessary to the software quality assessment). The insufficiency, inaccuracy and distortion of the SRS information lead, respectively, to a decrease in the veracity of software quality assessments, as well as to increase the gap of knowledge about software that results in unpredictable emergent properties of software systems.

Currently, the software quality evaluation by standard ISO 25010:2011 [7] is as follows (Fig. 1) – the software quality is evaluated on the basis of the characteristics, the characteristics are evaluated on the basis of subcharacteristics, the subcharacteristics are evaluated on the basis of measures, that are described in ISO 25023:2016 [8]. Evaluation of software quality and complexity based on the metric analysis is as follows (Fig. 2) – the software quality and complexity are calculated on the basis of the metrics, and the metrics are calculated on the basis of the indicators. For metric analysis, 14 quality metrics and 10 complexity metrics with exact or predicted values at the design stage have been selected [9]. The software quality measures, the software quality and complexity indicators, which are defined in the SRS, constitute the quality information of the SRS.

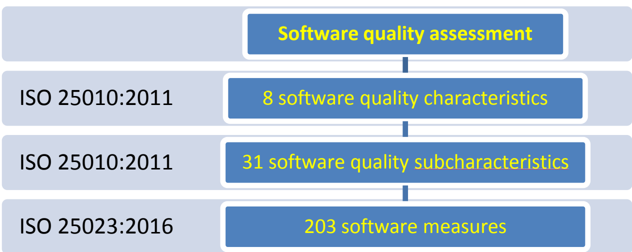

Fig. 1. The modern concept of software quality assessment by ISO 25010:2011

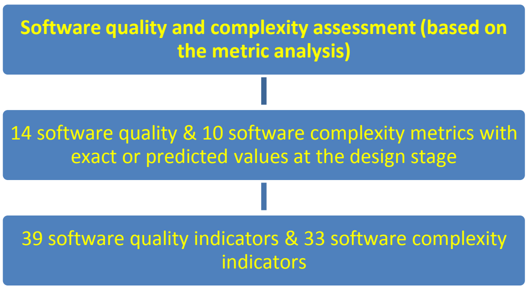

Fig. 2. The modern concept of assessment of software quality and complexity based on the metric analysis

So, *the sufficiency of quality information in the SRS* is the presence in the specification of all information elements (measures and indicators), which are necessary to the software quality assessment.

Today the evaluation of measures for the software quality subcharacteristics and characteristics, indicators for the software quality and complexity metrics is conducted only at the stage of the quality evaluation for the ready source code [5]. But the software requirements determine the required characteristics of the software quality, and also affect the methods of quantitative evaluation of software quality [5]. So, the SRS have all measures and indicators, which are needed to the subcharacteristics and metrics calculation [5]. So the information sufficiency for future software quality assessment can be evaluated on the basis of the SRS. And if some measures or indicators are absent, then the SRS has insufficient information for software quality assessment and the developers have to make the necessary adjustments in the SRS.

The conducted analysis of standards [7, 8] showed that they are presented in natural language in the textual form, so there is no mechanism for verification of the results of the implementation of these standards in the software development process. It has been established that quality information is conveniently presented as ontologies, which provide the reflection of cause-effect relationships between concepts.

The analysis of known ontological models in the field of software engineering has shown that, at present, the ontological models of profile for software certification [10], ontological models of intelligent decision support systems [11], ontological models for a single coherent underpinning for all ISO/IEC JTC1's SC7 standards [12, 13], the model of domain ontology for ISO/IEC 24744 [14] and the model of domain ontology in the software analysis and reengineering tools [15] have been developed. But nowadays there aren't ontological models of software quality based on ISO 25010:2011, ontological models of software quality and complexity based on metric analysis, and ontological models of the SRS in terms of the availability of information for the software quality assessment.

The analysis of known methods showed that the methods of software development on the basis of ontological models of tasks [16], methods of formation of normative profile in the software certification [10], ontological approach to specification of properties of software systems and their components [17], and methods of the SRS analysis (Using natural language processing technique, Using CASE analysis method, QAW-method, Using global analysis method, O'Brien's approach, Method to discover missing requirement elicitation, Selection of requirements elicitation technique, Comparison and categorization of requirements elicitation techniques, Techniques for ranking and prioritization of software requirements) [18-20] have been developed. But these methods are devoted to monitor the implementation of requirements rather than on evaluating the sufficiency of the quality information in the SRS.

The analysis of known tools has shown that the number of tools have been developed, in particular, the tools for constructing the software systems based on ontological models of tasks [16], and the automated tools of the SRS analysis (IBM Rational RequisitePro, IBM Rational/Telelogic DOORS, Borland Caliber RM, Sybase PowerDesigner, Open Source Requirements Management Tool, Sigma Software, DEVPROM) [18-20]. But these tools are not oriented to assessing the sufficiency of the quality information in the SRS.

Consequently, the known models, methods and tools don't solve the problem of evaluating the sufficiency of quality information in the SRS. In addition, they all belong to different methodological approaches and don't integrate among ourselves, that is, nowadays the information technology of evaluating the sufficiency of quality information in the SRS is absent.

The lack of the information technology of evaluating the sufficiency of quality information in the SRS creates the *actual scientific problem*, one of the ways of solving which is the development of the models, methods and tools of analyzing the sufficiency of quality information in the SRS. Therefore, the *aim of this study* is the development of the information technology (models, methods and tools) of evaluating the sufficiency of quality information in the SRS.

2 Information Technology of Evaluating the Sufficiency of Quality Information in Software Requirements Specification

The structure of the information technology (models, methods and tools) of evaluating the sufficiency of quality information in the SRS can be represented as follows – Fig. 3.

Fig. 3 shows that the developed information technology consists of: 1) mathematical and ontological models of the software quality by the standard ISO 25010:2011 (were developed and represented in [21]); 2) mathematical and ontological models of the software complexity and quality based on the metric analysis (were developed and represented in [22]); 3) mathematical and ontological models of the SRS (were developed and represented in [21, 22]); 4) methods of evaluating the sufficiency of the SRS information for software quality assessment (by the standard ISO 25010:2011) based on the ontologies; 5) methods of evaluating the sufficiency of the SRS information for software complexity and quality assessment (on the basis of the metric analysis results) based on the ontologies; 6) method of design results assessment and software characteristics prediction (was developed and represented in [23]); 7) subsystem of evaluating the sufficiency of the SRS information for software quality assessment based on the comparative analysis of ontologies; 8) subsystem of software complexity and quality evaluation and prediction based on the metric analysis results (was developed and represented in [23]).

Information technology of evaluating the sufficiency of quality information in the SRS				
Object – quality information in the software requirements specification	**Goal** – the solution of the actual scientific problem of ensuring the veracity of quality information in the SRS	**Models:** - mathematical and ontological models of the software quality by the standard ISO 25010:2011; - mathematical and ontological models of the software complexity and quality based on the metric analysis; - mathematical and ontological models of the SRS	**Methods:** - methods of evaluating the sufficiency of the SRS information for software quality assessment (by the standard ISO 25010:2011) based on the ontologies; - methods of evaluating the sufficiency of the SRS information for software complexity and quality assessment (on the basis of the metric analysis results) based on the ontologies; - method of design results assessment and software characteristics prediction	**Tools:** - subsystem of evaluating the sufficiency of the SRS information for software quality assessment based on the comparative analysis of ontologies; - subsystem of software complexity and quality evaluation and prediction based on the metric analysis results

Fig. 3. The structure of information technology of evaluating the sufficiency of quality information in the SRS

The practical implementation of the developed base (universal) ontological model of the subject domain "Software Engineering" (part "Software Quality") is the base ontology of the subject domain "Software Engineering" ("Software Quality"), the concept of which is represented on Fig. 4. The practical implementation of the developed base (universal) ontological models of the subject domain "Software Engineering" (part "Sofware Quality and Complexity") is the base ontology of the subject domain "Software Engineering" (part "Software Quality and Complexity"), the concept of which is represented on Fig. 5. The components of the base ontology of the subject domain "Software Engineering" ("Software Quality") are represented in [21], and the components of the base ontology of the subject domain "Software Engineering" ("Software Quality and Complexity") are represented in [22].

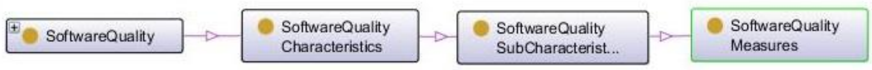

Fig. 4. Concept of the base ontology of the subject domain "Software Engineering" ("Software Quality")

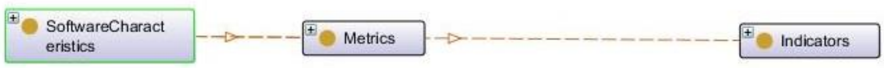

Fig. 5. Concept of the base ontology of the subject domain "Software Engineering" (part "Software Quality and Complexity. Metric analysis")

Methods of evaluating the sufficiency of the SRS information for software quality assessment (by the standard ISO 25010:2011) based on the ontologies were developed and detail represented in [21, 24].

The scheme of the method of evaluating the sufficiency of the SRS information for software quality assessment (by the standard ISO 25010:2011) based on the ontology is represented on Fig. 6.

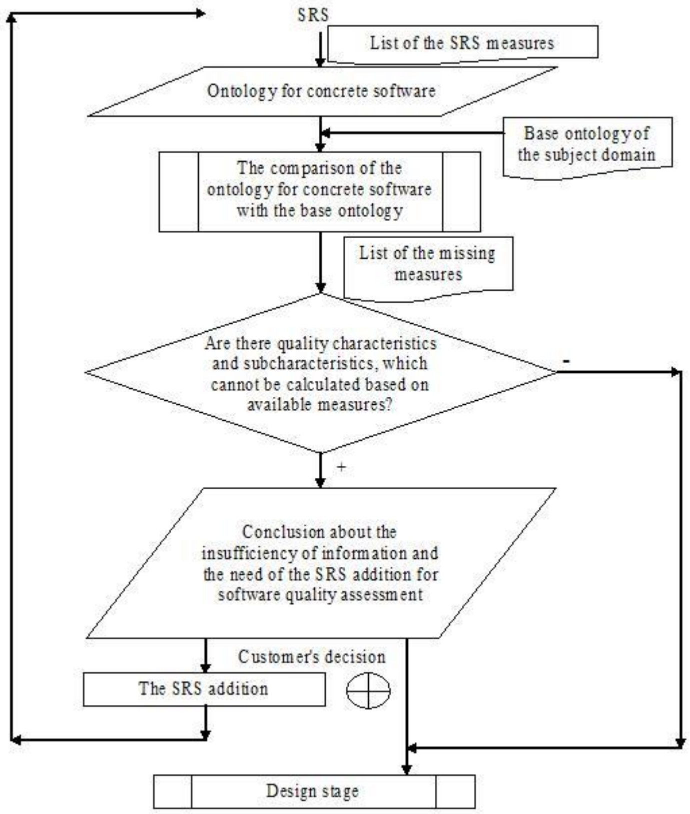

Fig. 6. The scheme of the method of evaluating the sufficiency of the SRS information for software quality assessment (by the standard ISO 25010:2011) based on the ontology

For the eliminate of the subjective evaluation and formal satisfaction of the software quality, it's necessity to consider the degree of severity of quality characteristics and subcharacteristics, and their significance. One of the problems of the known quality models is the calculation of the significance of the quality measures and characteristics. Quality characteristics and subcharacteristics correlate with each other by the measures. It was proven during the above software quality modeling. The existence of such correlations between subcharacteristics increases the significance and weight of software quality measures. Scheme of the method of evaluating the weights of software quality measures is represented on Fig. 7.

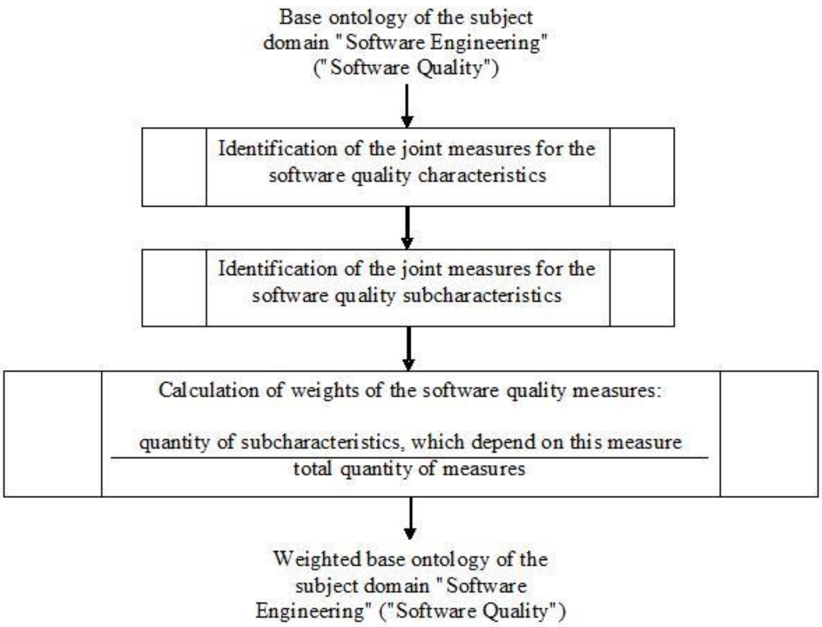

Fig. 7. Scheme of the method of evaluating the weights of software quality measures

The weights of the software quality measures were estimated by the method of evaluating the weights of software quality measures [21]. During the software quality assessment by ISO 25010:2011, it's important to satisfy the availability of measures with larger weights in the SRS for ensuring the appropriate level of veracity of infromation. Weighted ontology of the subject domain "Software Engineering" (part "Software quality") is the ontology, in which the software quality measures have weights with purpose of the recommendation of further satisfaction of these measures in the SRS. The weighted base ontology of the subject domain "Software Engineering" (part "Software quality") was developed on the basis on the base ontology of the subject domain "Software Engineering" (part "Software quality") with addition of information about the weights of software quality measures [21].

The scheme of the method of evaluating the sufficiency of the SRS information for software quality assessment (by the standard ISO 25010:2011) based on the weighted ontology is represented on Fig. 8.

For forming the logical conclusion about sufficiency of the SRS information for software quality assessment by ISO 25010:2011 the production rules were formed on the basis of the developed base and the weighted base ontologies for subject domain “Software engineering” (part “Software quality”). 138 production rules (for the each of measures) have the form “if-then” and were constructed as follows: if measure is missing in the concrete SRS, then: the counters of missing measures for appropriate subcharacteristics are increased by 1 and the counters of missing measures for appropriate characteristics are increased by the quantity of subcharacteristics of this characteristic, for calculation of which the SRS information is insufficient; the weight of the focused measure is assigned to the element of array of the missing measures weights (index of which is the focused measure).

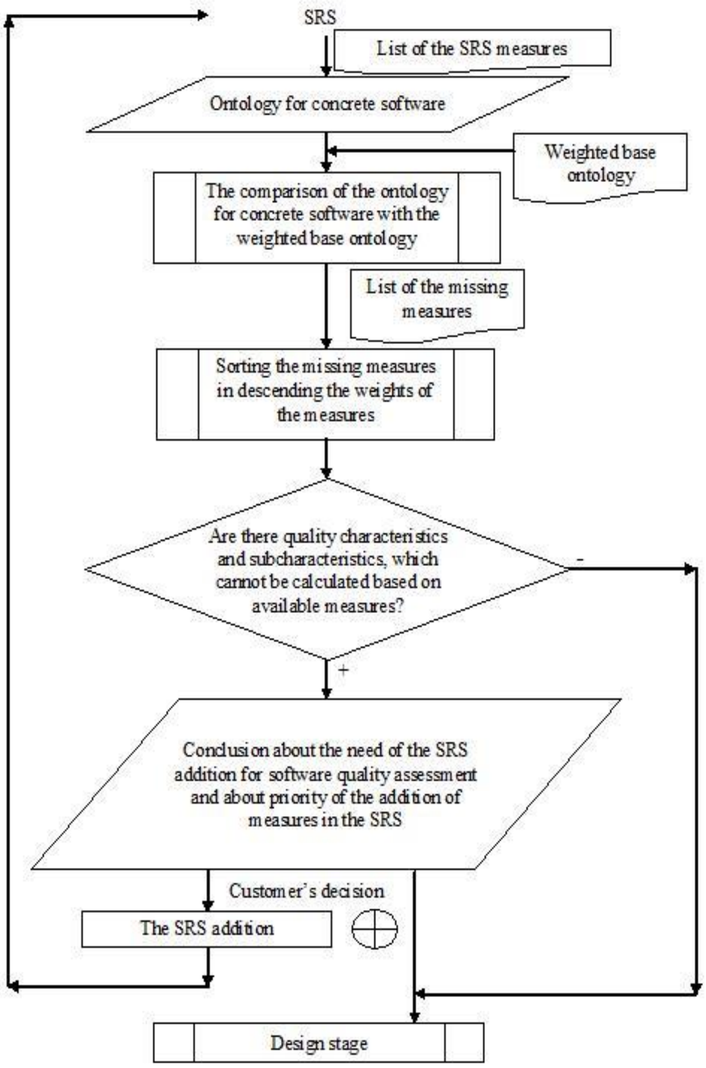

Fig. 8. Scheme of the method of evaluating the sufficiency of the SRS information for software quality assessment (by the standard ISO 25010:2011) based on the weighted ontology

The rule No. 139 has the form: if counters of missing measures for the all 31 software quality subcharacteristics are simultaneously equal to 0, then the SRS information is sufficient for calculation of all software quality subcharacteristics, else: the SRS information is insufficient for the calculation of some software quality subcharacteristics (with indicating the subcharacteristics, for calculation of which the SRS measures are insufficient). The rule No. 140 has the form: if counters of missing measures for the all 8 software quality characteristics are simultaneously equal to 0, then the SRS information is sufficient for calculation of all software quality characteristics, else: the SRS information is insufficient for the calculation of some software quality characteristics (with indicating the characteristics, for calculation of which the SRS measures are insufficient); array of the missing measures weights should be sorted in descending the values of elements (weights of missing measures); indices of those elements of the sorted array of the missing measures weights, which aren't equal 0, should be displayed – as the recommended priority of addition of the missing measures in SRS [24].

The scheme of method of forming the logical conclusion about sufficiency of the SRS information for software quality assessment by ISO 25010:2011 is represented on Fig. 9.

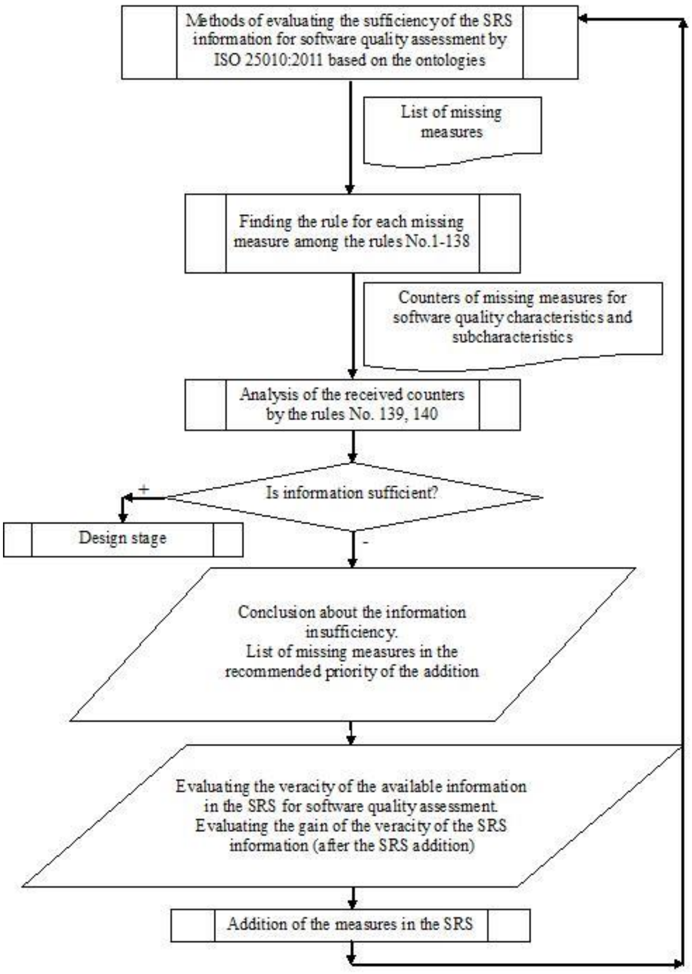

Fig. 9. Scheme of the method of forming the logical conclusion about sufficiency of the SRS information for software quality assessment by ISO 25010:2011

The methods of evaluating the sufficiency of the SRS information for software complexity and quality assessment (on the basis of the metric analysis results) based on the ontologies were developed and detail represented in [22, 25]. These methods are similar to the above methods of evaluating the sufficiency of the SRS information for software quality assessment (by the standard ISO 25010:2011) based on the ontologies (Fig. 6, Fig. 8). The weights of software complexity and quality indicators were calculated in [22] by the method of evaluating the weights of software quality measures (Fig. 7).

For forming the logical conclusion about sufficiency of the SRS information for software quality and complexity assessment by the metric analysis results the production rules were formed on the basis of the developed base and the weighted base ontologies for subject domain “Software engineering” (part “Software quality and complexity. Metric analysis”). 42 production rules (for the each of indciators) have the form “if-then” and were constructed as follows: if indicator is missing in the concrete SRS, then: the counters of missing indicators for appropriate metrics are increased by 1; the weight of the focused indicator is assigned to the element of array of the missing indicators weights (index of which is the focused indicator). The rule No. 43 has the form: if counters of missing indicators for the all 24 software quality and complexity metrics are simultaneously equal to 0, then the SRS information is sufficient for calculation of all software metrics, else: the SRS information is insufficient for the calculation of some software metrics (with indicating the metrics, for calculation of which the SRS indicators are insufficient); array of the missing indicators weights should be sorted in descending the values of elements (weights of missing indicators); indices of those elements of the sorted array of the missing indicators weights, which aren't equal 0, should be displayed – as the recommended priority of addition of the missing indicators in the SRS [25]. The method of forming the logical conclusion about sufficiency of the SRS information for software complexity and quality assessment by the metric analysis results is similar to the above method of forming the logical conclusion about sufficiency of the SRS information for software quality assessment by ISO 25010:2011 (Fig. 9).

Concept of method and subsystem of design results assessment and software characteristics prediction is represented on Fig. 10.

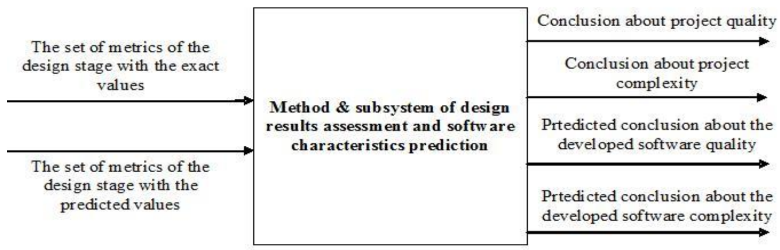

Fig. 10. The concept of method and subsystem of design results assessment and software characteristics prediction

For the completion of the proposed information technology of evaluating the sufficiency of quality information in the SRS, it’s necessary to develop (to design and realize) the subsystem of evaluating the sufficiency of the SRS information for software quality assessment based on the comparative analysis of ontologies.

3 Subsystem of Evaluating the Sufficiency of Software Requirements Specification Information for Software Quality Assessment Based on the Comparative Analysis of Ontologies

The inputs of the subsystem of evaluating the sufficiency of the SRS information for software quality assessment based on the comparative analysis of ontologies are the sets: 1) $\{qms_1,\ldots,qms_{nm}\}$ ($nm\leq138$) available in the SRS software quality measures (according to standards [7, 8], the software quality subcharacteristcs depend on 203 measures, but only on 138 different measures); 2) $\{sqcxi_1,\ldots,sqcxi_{ni}\}$ ($ni\leq42$) available in the SRS software quality and complexity indicators (the selected in [9, 23] software metrics depend on 72 indicators, but only on 42 different indicators).

The results of the developed subsystem are: 1) conclusion about the sufficiency of the SRS information for software quality assessment by the standard ISO 25010:2011; 2) recommendations about necessity and priority of the addition of the measures in the SRS for software quality assessment by ISO 25010:2011; 3) evaluation of the veracity of the available in the SRS information for software quality assessment by ISO 25010:2011; 4) conclusion about the sufficiency of the SRS information for software complexity and quality assessment by the metric analysis results; 5) recommendations about necessity and priority of the addition of the indicators in the SRS for determining the software complexity and quality by the metric analysis results; 6) evaluation of the veracity of the available in the SRS information for software quality assessment by the metric analysis results.

The concept of subsystem of evaluating the sufficiency of the SRS information for software quality assessment based on the comparative analysis of ontologies is represented on Fig. 11. The structure of subsystem of evaluating the sufficiency of the SRS information for software quality assessment based on the comparative analysis of ontologies is represented on Fig. 12.

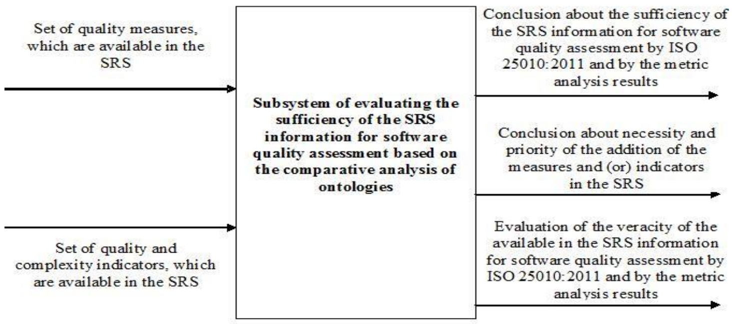

Fig. 11. The concept of subsystem of evaluating the sufficiency of the SRS information for software quality assessment based on the comparative analysis of ontologies

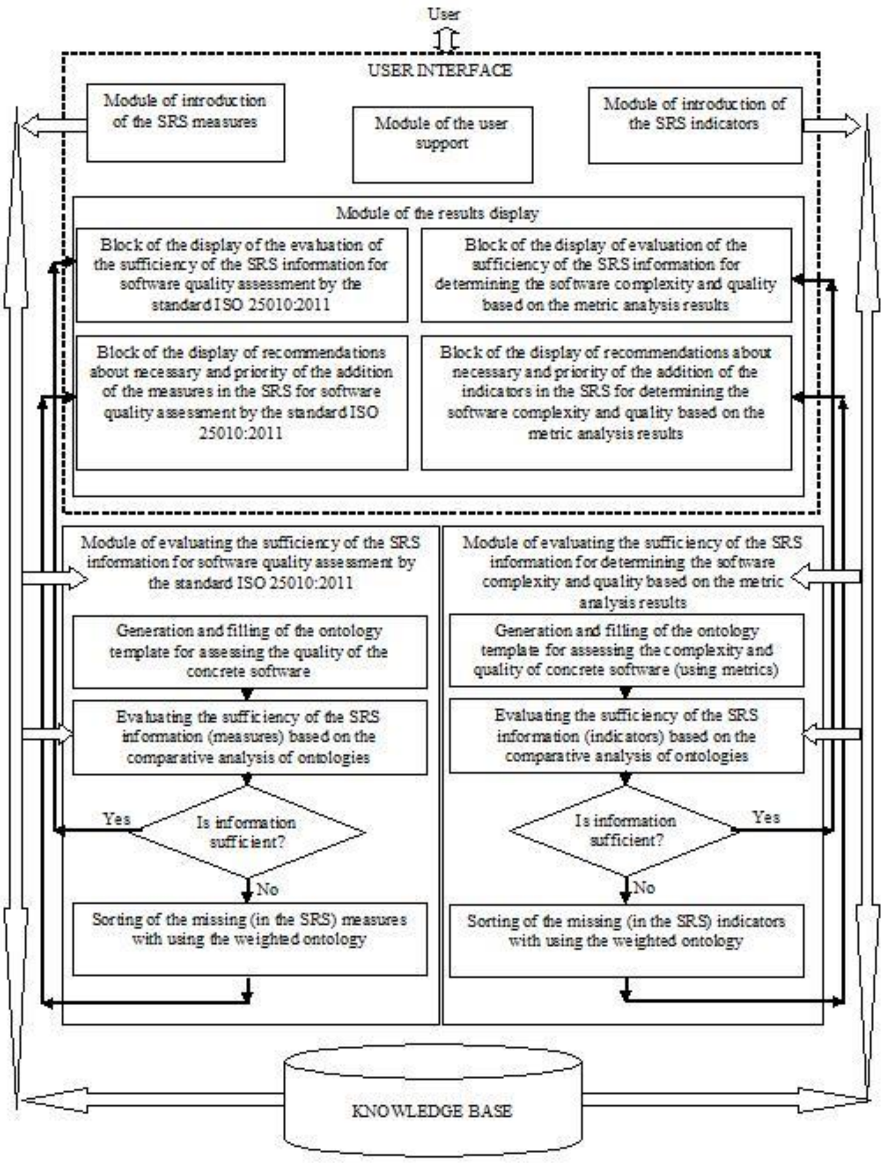

Fig. 12. The structure of subsystem of evaluating the sufficiency of the SRS information for software quality assessment based on the comparative analysis of ontologies

The developed subsystem consists of the next components: 1) *module of introduction of the SRS measures* – collects the user information about the available values of measures $\{qms_1,\ldots,qms_{nm}\}$ ($nm \leq 138$) in the SRS for concrete software; 2) *module of introduction of the SRS indicators* – collects the user information about the available values of indicators $\{sqcxi_1,\ldots,sqcxi_{ni}\}$ ($ni \leq 42$) in the SRS; 3) *module of the user* *support* – provides to the user the information about the structure of the SRS; about the SRS measures, which are necessary for software quality assessment by ISO 25010; about the SRS indicators, which are necessary for determining the software complexity and quality based on the metric analysis results; about the process of the forming the results of the described subsystem; 4) *module of evaluating the sufficiency of the SRS information for software quality assessment by the standard ISO 25010:2011* – works according to methods of evaluating the sufficiency of the SRS information for software quality assessment (by ISO 25010:2011) based on the ontologies. The generation and filling of the ontology template for assessing the quality of the concrete software are performed, considering introduced the available measures $\{qms_1,\ldots,qms_{nm}\}$ ($nm\leq 138$). The comparative analysis of the ontology for the concrete software with the developed base ontology for subject domain "Software engineering" (part "Software quality") is performed. The result of this comparative analysis is the list of missing measures (in the concrete SRS). If during the comparative analysis of ontologies the differences were not identified, then information of the SRS is sufficient for software quality assessment by ISO 25010. If during the comparative analysis of ontologies the differences were identified, then the available in the SRS measures are insufficient for some subcharacteristics and characteristics calculation, then the comparative analysis of the ontology for the concrete software with the developed weighted base ontology for subject domain "Software engineering" (part "Software quality") is performed, and sorting of all missing (in the SRS) measures in descending the values of weights is conducted, i.e. priority of the addition of these measures in the SRS is established. The quantitative evaluation of the veracity of the available in the SRS information for the software quality assessment is calculated; 5) *module of evaluating the sufficiency of the SRS information for determining the software complexity and quality based on the metric analysis results* – works according to methods of evaluating the sufficiency of the SRS information for software complexity and quality assessment (on the basis of the metric analysis) results based on the ontologies. The generation and filling of the ontology template for assessing the quality and complexity of the concrete software are performed, considering introduced the available indicators $\{sqcxi_1,\ldots,sqcxi_{ni}\}$ ($ni\leq 42$). The comparative analysis of the ontology for the concrete software with the developed base ontology for subject domain "Software engineering" (part "The software quality and complexity. Metric analysis") is performed. The result of this comparative analysis is the list of missing indicators (in the concrete SRS). If during the comparative analysis of ontologies the differences were not identified, then information of the SRS is sufficient for software quality and complexity assessment based on the metric analysis. If during the comparative analysis of ontologies the differences were identified, then the available in the SRS indicators are insufficient for some metrics calculation, then the comparative analysis of the ontology for the concrete software with the developed weighted base ontology for subject domain "Software engineering" (part "The software quality and complexity. Metric analysis") is performed, and sorting of all missing (in the SRS) indicators in descending the values of weights is conducted, i.e. priority of the addition of these indicators in the SRS is established. The quantitative evaluation of the veracity of the available in the SRS information for the software quality and complexity assessment is calculated; 6) *knowledge base* – contains the base and the weighted base ontologies for subject domain “Software engineering” (part “Software quality”, part "Software quality and complexity. Metric Analysis"), the formed ontologies for the concrete software, and production rules of forming the logical conclusion about the sufficiency of the SRS information for software quality assessment by ISO 25010:2011 and for software complexity and quality assessment by the metric analysis results; 7) *module of the results display* – the components of this block display the formed conclusions, recommendations and evaluations to user.

4 Experiments: Evaluating the Sufficiency of the Information of the SRS of Automated System for Large-Format Photo Print for Software Quality Assessment

For experiment the SRS of automated system (AS) for large-format photo print was analyzed. The measures, which are available in this SRS, were identified. The ontology for this software was developed [21].

The comparison (in Protégé 4.2) of the developed ontology for AS for large-format photo print with the base ontology for subject domain “Software engineering” (part “Software quality”) provides the conclusion, that in the developed ontology for the concrete software 4 measures are absent: “Number Of Functions”, “Operation Time”, “Number Of Data Items”, “Number Of Test Cases” (Fig. 13).

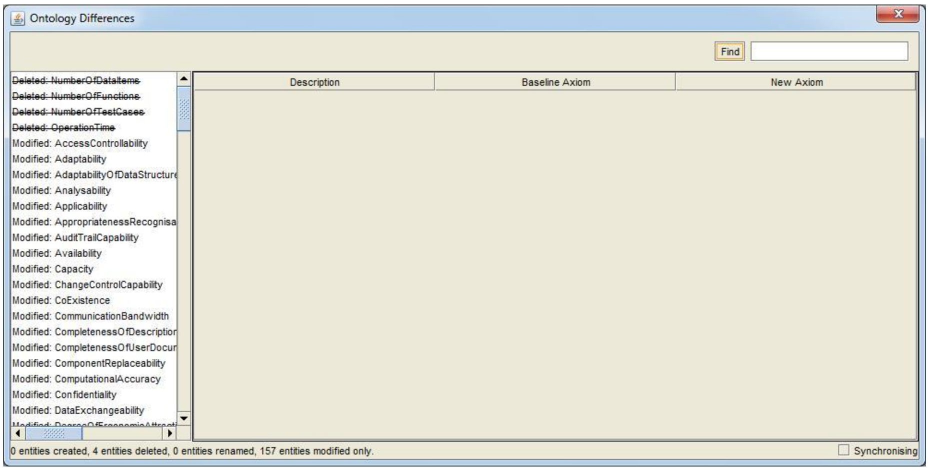

Fig. 13. Comparison of ontology for concrete software (AS for large-format photo print) with the base ontology of the subject domain "Software Engineering" (part" Software Quality")

Then the set of missing measures is: {Number Of Functions, Operation Time, Number Of Data Items, Number OF Test Cases}. The finding the rule for each element of this set among 138 rules for the measures is performed. According to these rules, the counters of missing measures are counted.

According to the rule No. 139, the fact was established, that the available measures in the SRS of AS for large-format photo print are insufficient for calculation of following subcharacteristics: Functional Completeness, Functional Correctness, Functional Appropriateness, Maturity, Availability, Fault Tolerance, Recoverability, Time Behaviour, Resource Utilization, Capacity, Appropriateness Recognisability, Learnability, Operability, Modularity, Analysability, Modifability, Testability, Confidentiality, Integrity, CoExistence, Interoperability, Adaptability, Replaceability.

According to the rule No. 140, the fact was established, that the available measures in the SRS AS for large-format photo print are insufficient for calculation of all 8 software quality characteristics. Thus, the lack of 4 measures in the SRS led: to the impossibility of calculating the 23 (from 31) subcharactersitics, to the impossibility of calculating all 8 software quality characteristics with high veracity and, respectively, to the impossibility of software quality assessment with high veracity. After establishing the fact of insufficiency of information of the SRS of AS for large-format photo print: sorting the array of the missing measures weights in descending the values of elements was conducted; displaying the indices of those elements of the sorted array of the missing measures weights, which aren't equal 0. Sorted list of missing in the SRS measures in descending the weights: 1) Operation Time (17/138); 2) Number of Functions (11/138); 3) Number of Data Items (8/138); 4) Number of Test Cases (5/138). This list represents the recommended priority of the addition of missing measures in the SRS of AS for large-format photo print.

Next, the evaluation of the veracity of the available in the SRS information for software quality assessment is done (according to the method of forming the logical conclusion about sufficiency of the SRS information for software quality assessment by ISO 25010:2011 [21, 24]). So, for the analyzed SRS of AS for large-format photo print the conclusion about insufficient data for software quality assessment was formed by the developed subsystem, and the veracity of the available in the SRS information for the software quality assessment by ISO 25010:2011 is 76%.

Because the proposed methods of evaluating the sufficiency of the SRS information for software quality assessment (by ISO 25010:2011) based on the ontology are iterative, and there are subcharacteristics and characteristics, for calculation of which the measures of SRS are insufficient, then the addition of the necessary measures in the SRS was held. After addition of the SRS of AS for large-format photo print, the ontology (version 2) for this software was re-developed. The comparison of the re-developed ontology with the base ontology for subject domain “Software engineering” (part “Software quality”) provides the conclusion, that 2 measures were added in the SRS: “Number Of Functions” (2^{nd} in the sorted list), “Number Of Data Items” (3^{rd} in the sorted list). Then the set of missing measures is: {Operation Time, Number Of Test Cases}. The finding the rule for each element of this set is performed.

According to the rule No. 139, the fact was established, that the available measures in the SRS of AS for large-format photo print are still insufficient for calculation of 18 subcharacteristics (with indicating these subcharacteristics), but addition 2 measures in the SRS made possible the calculation of: Functional Completeness, Capacity, Appropriateness Recognisability, Analyzability, Replaceability.

According to rule No.140, the fact was established, that the available measures in the SRS of AS for large-format photo print are still insufficient for calculation of all 8 software quality characteristics. After establishing the fact of insufficiency of information of the SRS: sorting the array of the missing measures weights in descending the values of elements was conducted; displaying the indices of those elements of the sorted array of the missing measures weights, which aren't equal 0. Sorted list of missing (after addition) in the SRS measures in descending the weights: 1) Operation Time; 2) Number of Test Cases.

Next, the evaluation of the veracity of the available (after addition) in the SRS information for software quality assessment is done (according to the method of forming the logical conclusion about sufficiency of the SRS information for software quality assessment by ISO 25010:2011 [21, 24]). So, for the analyzed SRS of AS for large-format photo print the conclusion about still insufficient information for software quality assessment was formed by the developed subsystem, and the veracity of the available (after addition) in the SRS information for the software quality assessment by ISO 25010:2011 is 88%.

The process of addition the necessary measures in the SRS is iterative. It can be continued until all quality characteristics and subcharacteristics will be possible to calculate or until the conclusion will be formed, that the SRS information are insufficient for software quality assessment. The customer of developed AS for large-format photo print has decided that further complement of SRS is economically inexpedient.

The gain of the veracity of the SRS information for software quality assessment by ISO 25010:2011 after addition of necessary measures in the SRS is 12% (according to the method of forming the logical conclusion about sufficiency of the SRS information for software quality assessment by ISO 25010:2011 [21, 24]). So, *the developed information technology and subsystem of evaluating the sufficiency of the quality information in the SRS provides the increase of the veracity of the SRS information for the software quality assessment by ISO 25010:2011 by 12% for AS for large-format photo print.*

Let’s consider the functioning of the developed information technology and subsystem for evaluating the sufficiency of the quality information in the SRS for metric analysis. For experiment the SRS of AS for large-format photo print was analyzed. The indicators, which are available in this SRS, were identified. The ontology for this software metric analysis was developed [22].

The comparison (in Protégé 4.2) of the developed ontology for AS for large-format photo print with the base ontology for the subject domain "Software Engineering" (part "Software complexity and quality. Metric analysis") provides the conclusion, that in the developed ontology for the concrete software metric analysis 9 indicators are absent: “Control Variables”, “Cost Of One Line”, “Project Duration”, “Project Type”, “Quantity Of Code Lines”, “Quantity Of Links Of Each Module”, “Quantity Of Modules”, “Share Of Design Stage In Lifecycle”, “Total Quantity Of Operators”. Then the set of missing indicators is: {Control Variables, Cost Of One Line, Project Duration, Project Type, Quantity Of Code Lines, Quantity Of Links Of Each Module, Quantity Of Modules, Share Of Design Stage In Lifecycle, Total Quantity Of Operators}. The finding the rule for each element of this set among 42 rules for the indicators is performed. According to these rules, the counters of missing indicators are counted.

According to the rule No. 43, the fact was established, that the available indicators in the SRS of AS for large-format photo print are insufficient for calculation of the 20 (from 24) metrics with high veracity and, respectively, and for metric analysis with high veracity. After establishing the fact of insufficiency of information of the SRS of AS for large-format photo print: sorting the array of the missing indicators weights in descending the values of elements was conducted; displaying the indices of those elements of the sorted array of the missing indicators weights, which aren't equal 0. So, for increasing the veracity of the SRS information for software metric analysis the next indicators should be added in the SRS in this consistency: 1) Quantity Of Code Lines, 2) Quantity Of Modules, 3) Project Duration, 4) Total Quantity Of Operators, 5) Cost Of One Line, 6) Project Type, 7) Share Of Design Stage In Lifecycle, 8) Control Variables, 9) Quantity Of Links Of Each Module.

Next, the evaluation of the veracity of the available in the SRS information for metric analysis is done (according to the method of forming the logical conclusion about sufficiency of the SRS information for software quality assessment by metric analysis results [22, 25]). So, for the analyzed SRS of AS for large-format photo print the conclusion about insufficient data for metric analysis was formed by the developed subsystem, and the veracity of the available in the SRS information for the metric analysis is 42%.

The addition of the necessary indicators in the SRS was held. After addition of the SRS of AS for large-format photo print, the ontology (version 2) for this software was re-developed. The comparison of the re-developed ontology with the base ontology provides the conclusion, that 2 indicators were added in the SRS: "Quantity Of Modules" (2^{nd} in the sorted list), "Total Quantity Of Operators" (4^{th} in the sorted list). So, for increasing the veracity of the SRS information for the metric analysis the next indicators should be added in the SRS in this consistency: 1) Quantity Of Code Lines, 2) Project Duration, 3) Cost Of One Line, 4) Project Type, 5) Share Of Design Stage In Lifecycle, 6) Control Variables, 7) Quantity Of Links Of Each Module.

Next, the evaluation of the veracity of the available (after addition) in the SRS information for metric analysis is done (according to the method of forming the logical conclusion about sufficiency of the SRS information for metric analysis [22, 25]). So, for the analyzed SRS of AS for large-format photo print the conclusion about still insufficient information for metric analysis was formed by the developed subsystem, and the veracity of the available (after addition) in the SRS information for the metric analysis is 56%.

The customer of developed AS for large-format photo print has decided that further complement of the SRS is economically inexpedient.

The gain of the veracity of the SRS information for metric analysis after addition of necessary indicators in the SRS is 14% (according to the method of forming the logical conclusion about sufficiency of the SRS information for metric analysis [22, 25]). So, *the developed information technology and subsystem of evaluating the sufficiency of the quality information in the SRS provides the increase of the veracity of the SRS information for the metric analysis by 14% for AS for large-format photo print.*

5 Conclusions

The information technology of evaluating the sufficiency of quality information in the SRS are first time proposed in this paper. It designed to the support of the software quality assessment at the early lifecycle stages. They provides: the conclusion about the sufficiency of the SRS information for software quality assessment by ISO 25010:2011 and based on the metric analysis results; the prioritization of the additions of the SRS by the measures and (or) by the indicators (if the SRS information is insufficient); the quantitative evaluations of the veracity of the available in the SRS information for software quality assessment by ISO 25010 and by the metric analysis; the increasing the software quality at the early lifecycle stages.

The subsystem of evaluating the sufficiency of the SRS information for the software quality assessment on the basis of the comparative analysis of the ontologies is first time proposed in this paper. It is the decision support system that provides the decision about: the sufficiency of the SRS information for the software quality assessment, the necessity of the addition(s) of the measures and(or) the indicators in the SRS, the veracity of the available in the SRS information for the software quality assessment by ISO 25010:2011 and by the metric analysis.

The experiments proved, that the use of the developed information technology of evaluating the sufficiency of quality information in the SRS provides the increase of the veracity of the SRS information for software quality assessment based on ISO 25010:2011 by 12%, and based on the metric analysis results by 14% even after one addition of the SRS for AS for large-format photo print.

The proposed information technology provides the increasing the veracity of the quality information in the SRS, and improving the software quality at the early stages of the lifecycle.

References

1. Jones, C., Bonsignour, O.: The economics of software quality. Boston: Pearson Education (2012).
2. Ishimatsu, T., Levenson, N., Thomas, J., Fleming, C., Katahira, M., Miyamoto, Y., Ujiie, R.: Hazard analysis of complex spacecraft using systems-theoretic process analysis. Journal of Spacecraft and Rockets. 51, 509-522 (2014).
3. Yourdon, E.: Death March: The complete software developer's guide to surviving "Mission impossible" projects: 2nd edition. Prentice Hall (2003).
4. Shamieh, C.: Systems engineering for dummies. Wiley Publishing (2011).
5. Maevskiy, D., Kozina, Y.: Where and when is formed of software quality? Electrical and Computer Systems. 18, 55-59 (2016) [in Russian]
6. McConnell, S.: Code complete. Microsoft Press (2013).
7. ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models (2011).
8. ISO 25023:2016. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). Measurement of system and software product quality (2016).
9. Fenton, N.: Software metrics: A rigorous approach (3rd edition). CRC Press (2014).

10. Shostak, I., Butenko, I. Ontology approach to realization of information technology for normative profile forming at critical software certification. Herald of the Military Institute of Kiev National University named after Taras Shevchenko. 38, 250–253 (2012).
11. Lytvyn, V.: Modeling of intelligent decision support systems using the ontological approach. Radio electronics, computer science, management. 2, 93-101 (2011) [in Ukrainian]
12. Henderson-Sellers, B., Gonzalez-Perez, C., McBride, T., Low, G.: An ontology for ISO software engineering standards: 1) Creating the infrastructure. Computer Standards & Interfaces. 36(3), 563-576 (2014).
13. Ruy, F. B., Falbo, R. A., Barcellos, M. P., Guizzardi, G., Quirino, G. K. S.: An ISO-based software process ontology pattern language and its application for harmonizing standards. ACM SIGAPP Applied Computing Review. 15(2), 27-40 (2015).
14. Hamri, M. M., Benslimane, S. M.: Building an Ontology for the Metamodel ISO/IEC24744 using MDA Process. International Journal on Modern Education and Computer Science. 8, 48-60 (2015).
15. Jin, D., Cordy, J. R.: Ontology-based software analysis and reengineering tool integration: The OASIS service-sharing methodology. In: IEEE International Conference on Software Maintenance Proceedings, pp. 613-616. IEEE, Budapest (2005).
16. Burov, E. Complex ontology management using task models. International Journal of Knowledge-Based and Intelligent Engineering Systems. 18(2), 111-120 (2014).
17. Babenko, L. Ontological approach to specification of software systems features and components. Cybernetics and System Analysis. 1, 180-187 (2009) [in Russian]
18. Chen, A., Beatty, J.: Visual models for software requirements. MS Press, Washington (2012).
19. Fatwanto, A.: Software requirements specification analysis using natural language processing technique. In: IEEE International Conference on Quality in Research Proceedings, pp. 105-110. IEEE, Yogyakarta (2013).
20. Rehman, T., Khan, M.N.A., Riaz, N.: Analysis of requirement engineering processes, tools/techniques and methodologies. International Journal of Information Technology and Computer Science. 5(3), 40-48 (2013).
21. Hovorushchenko, T.: Methodology of evaluating the sufficiency of information for software quality assessment according to ISO 25010. Accepted for publication in Journal of Information and Organizational Sciences. (2017).
22. Hovorushchenko, T.: Models and methods of evaluation of information sufficiency for determining the software complexity and quality based on the metric analysis results. Central European Researchers Journal. 2, 42-53 (2016).
23. Pomorova, O., Hovorushchenko, T.: Artificial Neural Network for Software Quality Evaluation Based on the Metric Analysis. In: IEEE East-West Design & Test Symposium Proceedings, pp.200-203. IEEE, Kharkiv (2013).
24. Hovorushchenko, T.: Forming the Logical Conclusion about Sufficiency of Information of Software Requirements Specification for Software Quality Assessment by ISO 25010:2011. In: IEEE First Ukraine Conference on Electrical and Computer Engineering Proceedings, pp.789-794. IEEE, Kyiv (2017).
25. Hovorushchenko, T. The Rules and Method of Forming the Logical Conclusion about Sufficiency of Information for Software Metric Analysis. Accepted for publication in: IEEE International Scientific and Technical Conference on Computer Science and Information Technologies. IEEE, Lviv (2017).