Information Technology of Evaluating the Sufficiency of Information on Quality in the Software Requirements Specifications

Tetiana Hovorushchenko [1[0000-0002-7942-1857]] and Oksana Pomorova [2[0000-0002-8193-8759]]

[1] Khmelnitsky National University, Institutska str., 11, Khmelnitsky, Ukraine
tat_yana@ukr.net
[2] University of Lodz, Uniwersytecka str., 3, Lodz, Poland
o.pomorova@gmail.com

Abstract. The paper presents the research of information flows at the early stages of the software development lifecycle on the subject their impact on the expected level of software quality. The result of the research is the identification of the problem of evaluating the sufficiency of information on quality in the software requirements specifications (SRS), and the development of information technology that addresses this problem. Information technology is based on the use of standards ISO 25010, ISO 25023 for the development of set-theoretical and ontological models of software quality. During the use of this information technology, the developers receive: the assessment of the sufficiency of information on quality in the SRS; request about the complement of the SRS with the necessary information with the purpose to increase the sufficiency of the volume of information on quality. The use of ontologies provides the automation of the processes of forming the requests for the complement of requirements, which reduces the subjective influence of professionals and preserves their experience and knowledge in the software company.

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

1 Introduction

At the current stage of development and implementation of information technologies in various fields of human activity, there are decisive changes, as there emerged the powerful technical resources for the accumulation and processing of large volumes of information. But the application of known methods and tools for the processing of such volumes of information doesn't justify the expectations of developers, leads to over-resources, losses of important information and conflicts between customers expectations and the obtained results [1-4].

At the same time, directions of intellectualization of the processing the information and data are actively developing: machine learning [5], cognitive computing [6], deep learning [7], semantic web [8]. They provide the solving new classes of tasks based on available information resources, as well as the development of methods of combining the various technologies and tools to increase the efficiency of the use of information resources.

All of the above are the prerequisites for the transition to a new qualitative level of information processing, and accordingly for the development of information technologies of the new generation.

However, the specifics and peculiarities of the subject domains, for which the information technologies are being developed, significantly affect the content and methods of information processing. Therefore, the expectations of universal approaches to creating the effective information technologies for various domains of activity are premature today. An approach, which is based on the study of the characteristics and features of the subject domains and the development of new information technologies just for the concrete domains, remains justified [9].

The special attention in the direction of the development and implementation of information technology today requires the domain of software engineering, in particular, to solve the problems of software quality assurance. The achievement of high values of software quality is a key factor in its efficient use and one of the main needs of customers. The need for quality assurance is based on the fact that software bugs and failures are threatened by catastrophes that lead to human casualties, environmental disasters, significant time and financial losses [10, 11].

As statistics [11-13] show, today there are problems in the field of software quality assurance – large projects still have time or cost overruns, developed software often doesn't have the necessary functionalities, its performance is low, and its quality doesn't satisfy consumers. A significant quantity of bugs is introduced in software at the stage of formation and formulation of requirements - these bugs make up 10-23% of all bugs, and the greater the size of software, the more bugs are introduced at this stage [12, 13]. The vast majority of software-related crashes occurred due to false requirements, not because of bugs of encoding. The cost of correcting the incorrect requirements in the specification, which were discovered after the release of the product, is almost 100 times the cost of correcting the defects of the specification, which were found at earlier stages of the formation and formulation of requirements [12].

Software projects often fail because of bugs at the early stages of the software lifecycle, namely [13-16]: inadequate formulation of the requirements; unsuccessful design or ineffective planning; incorrect understanding or insufficient analysis of the specification and project; unrealistic project plans; incorrect model of the lifecycle.

One of the most important stages in project management is gathering business requirements from stakeholders. Requirements are often vague because it is difficult for customers to articulate their needs before they see the end product [15].

Analysis of the bugs of the embedded and application software, carried out in [17], showed that the incorrectness and incompleteness of the SRS led to: the impossibility of launching the "Cosmos-419", the impossibility of undocking the "Mars 2" station, the crash of the helicopter "Chinook", "lethal" sessions of radiation therapy with "Therac-25", the explosion of rocket "Ariane 5", the falling rating on the exchange trading of company "Dow Jones Industrial Average", the failure of New York Bank system, Y2K error, accidents at AT&T telecommunication company, which led to death of people, significant financial losses, loss of reputation of companies.

In [12] the fact is confirmed, that the causes of many incidents and accidents through software are in the SRS. Consequently, the majority of accidents through software are the result of false of requirements, rather than through coding errors. In [12] the experiment was found that the software versions, written by different developers for the same requirements, contain a number of joint errors associated with errors or inaccuracies of requirements (of SRS). Analysis of a large number of software projects, carried out in [18], also found that the main place of bugs in software is the stage of requirements formulation (SRS). Such experimental evidence directly leads to the need to deepening the analysis and evaluation of the SRS.

Moreover, the need to deepening the analysis and evaluation of requirements takes place both for traditional design and for Agile methodologies (especially for requirements regulating non-functional software characteristics). During using Agile methodologies, the initial requirements stack is initially idenificated based on business-objectives. The functional requirements are detailed later, during development cycles in model storming sessions and via Test Driven Development (TDD) [15]. The Agile methodology includes several document-oriented strategies, including the Just Barely Good enough (JBGE) artifacts strategy, which is that the documentation needs to be sufficient for the situation at hand and no more [15, 19]. Consequently, the functional requirements during agile development of software are constantly changing, refined and supplemented, which adversely affects on the non-functional requirements (in particular, requirements for software quality).

For this study, two approaches to assessing the quality of software were chosen - software quality assessment based on SQuaRE model and software quality assessment based on the processing of results of the metric analysis. These approaches are actual for the domain of software quality assessment at the current time.

As the software quality assessment by the model of ISO 25010 (SQuaRE) [20] directs the software development to specification and evaluation of requirements, so exactly this standard was chosen for further work. The evaluation of software quality according to ISO 25010:2011 [20] is as follows: the comprehensive evaluation of software quality is based on the quality characteristics and subcharacteristics, which, in turn, are evaluated on the basis of measures (specified in ISO 25023: 2016 [21]).

One of the approaches, which are proposed by the authors, for prediction and assessment of the software quality at the early stages of the life cycle is using the results of metric analysis [22]. Quality is calculated on the basis of 24 metrics with exact or predicted values at the design stage [22]. The comprehensive assessment of software quality is calculated on the basis of the values of metrics, which, in turn, are calculated based on the indicators. All required quality measures and indicators are in the software requirements specification, that is, already on the basis of the SRS, we can assess the sufficiency of information for further determination and achievement of the software quality.

Then *the information on quality* consists of quality measures and quality indicators, which are specified in the specification. *The sufficiency of information on quality* *in the SRS* is the availability in the SRS of all the information elements (measures and indicators), which are required for software quality assessment.

The analysis of known models, methods and tools of providing the information on quality in the SRS [22] has shown that they don't solve the problem of evaluating the sufficiency of information on quality in the SRS. All of them belong to different methodological approaches and don't integrate with each other, that is, today there is no information technology of evaluating the sufficiency of information on quality in the SRS.

Consequently, the lack of models, methods and tools of evaluating the sufficiency of information on quality in the SRS is *the actual scientific problem*. One of the ways of its solution is the development of the information technology of evaluating the sufficiency of information on quality in the SRS.

2 Modeling the Information Flows Movement in Generating the Software Requirements Specification

The lifecycle of software development begins with the formation of the set of requirements, and the SRS on their basis.

The main sources of information at the stage of formation and formulation of requirements for software are the business requirements of the customer, the requirements of the subject domain, standards, descriptions of the process of developing and implementing such software, etc. All of the above is *the set of input information (SII)*, which describes the functions of the future software, its features and limitations. Business requirements describe the goal of the system creation, the criteria for achieving this goal, the key requirements for the results and their priorities and limitations [23]. All this information is presented as a text (verbal) description of business processes at the levels of business functions and operations for various functional parts of the software, sets of business rules and a set of models of the subject domain, for example, conceptual models, scenarios, diagrams, etc.

Also, in the SII it is necessary to take into account the text-based requirements of the standards for software development and the standards of the subject domain, for which the software is being developed. In this case, the information losses often held due to the incomplete and different understanding of the needs and context of information. Especially such losses are significant for software projects, which are developed at the junction of subject domains (for example, software for medicine).

On the basis of the set of input information (SII) the SRS is formed. Consequently, characteristics of the SRS are largely determined by the characteristics of the SII. If the SII has insufficient information, or information is inaccurate, ambiguous or contradictory, then there is a high probability that all these deficiencies will be in the SRS. But software projects, whose specifications are based on such information, cannot be successful.

For software quality assurance, it is necessary to research the characteristics of the SII with the purpose of assessing the completeness of displaying information, which describes the functions, features and limitation of the future software. Also, it is necessary to identify the facts of the insufficiency of information on quality in the SRS.

Information flows, which describes the requirements, that regulate quality characteristics, are formed on the basis of the high-level goals of the organization of customers and of the goals, tasks, business rules, that include corporate policies and industrial standards requirements, which is presented as the document about object and boundaries of project, constraints of design and implementation. Such information flows consist of the requirements of the SII, which determine the characteristics of the quality – for example, the requirement “At least 25% of the bandwidth of the processor and operative memory, which are available for use, should not be used in the conditions of the planned peak load” determines the quality characteristic “Performance Efficiency”; the requirement “Only users with privileges at the level of the Auditor should be able to view the customer transactions” determines the quality characteristic “Security”.

The requirements that determine the quality characteristics are contained in the SII. All information from the SII enters *the module of information processing*, which executes the construction and filling of the software quality models. The set-theoretical models of the software quality, presented in [22], contain information about characteristics-subcharacteristics-measures of quality, as well as information about metrics and indicators of quality (Fig. 1).

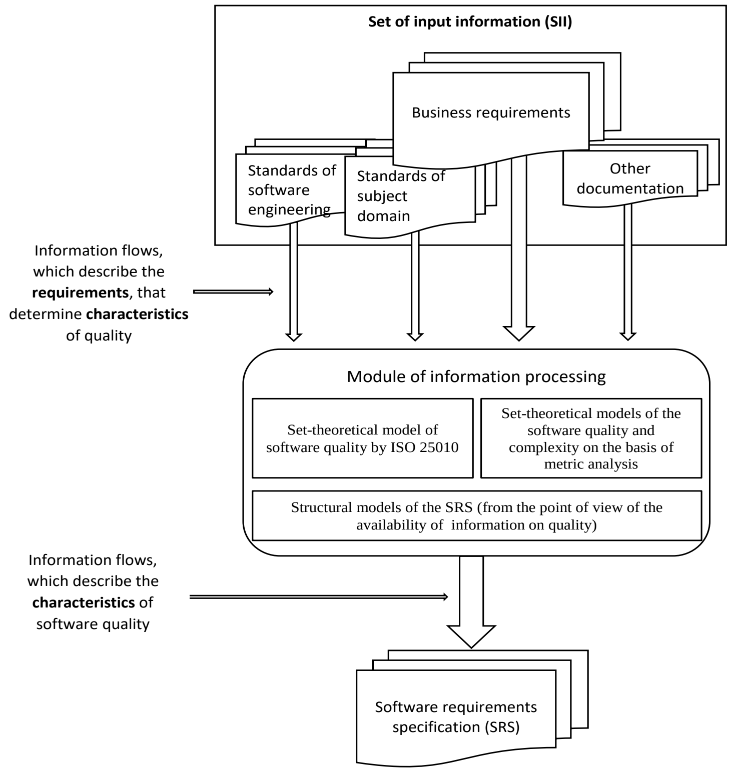

Fig. 1. Transformation of the information flows during the development of the SRS on the basis of the SII.

The result of the work of the module of information processing is the information flows, which describe the characteristics of the software quality. They contain the systematized information on quality which then comes into the SRS. For example, the requirement from the SII “At least 25% of the bandwidth of the processor and operative memory, which are available for use, should not be used in the conditions of the planned peak load” has the following representation in the SRS: “Mean Amount of Throughput is 75%”,”Maximum Memory Utilization is 75%”, “IO Loading Limits is peak load”, which reflect the quality measures, from which characteristic “Performance Efficiency” depends. The SII requirement “Only users with privileges at the level of the Auditor should be able to view the customer transactions” has the following representation in the SRS: “Access Controllability – privilege levels”, and also on the basis of the requirements of this type the following measures are calculated: “Number of Events Requiring Non-Repudiation Property” “Number of Provided Authentification Methods”, from which the characteristic “Security” depends.

Further evaluating the sufficiency of information on quality in the SRS is based on ontologies, that provide the identification and reflection of the cause-effect relationships between requirements, which determine the quality characteristics, between directly quality characteristics and subcharacteritics, between quality metrics, etc. [24, 25]

The base (universal) ontologies of the subject domain "Software Engineering" (parts "Software quality", "Software quality. Metric analysis") were developed for directly evaluating the sufficiency of the volume of information on quality in the SRS [22, 26] – for example, the component of base ontology for Functional Suitability is represented on Fig. 2. Similar ontologies have been developed for the remaining 7 software quality characteristics (in accordance with ISO 25010), and for assessing the quality and complexity of the software project and for predicting the quality and complexity of future software based on the processing of results of metric analysis [22, 26]. The base ontologies reflect the required information on quality (measures and indicators), that should be available in the SRS for providing the sufficiency of its information on quality. The validity of base ontologies is confirmed by the fact that ontologies (ontological knowledge bases) are filled on the basis of information from the standards ISO 25010, ISO 25023, and on the basis of information from classical papers on the software quality metrics [18, 27, 28] respectively.

On the basis of the base ontologies, the weighted base ontologies are also developed in [22]. In the weighted ontologies, the measures and indicators of software quality have the weights for recommending the further satisfaction of these measures and indicators in the SRS. The method of evaluating the weights of software quality measures and indicators is represented in [29]. The numerator of the weight of each measure or indicator indicates the number of software quality subcharacteristics or metrics, which depend on this measure or indicator. The denominator of the weight contains the total number of non-duplicate measures or indicators.

During the development of the concrete software, in addition to the base ontologies, it is necessary the ontology of this software, that displays the available information on quality (measures and indicators) in the SRS for this software.

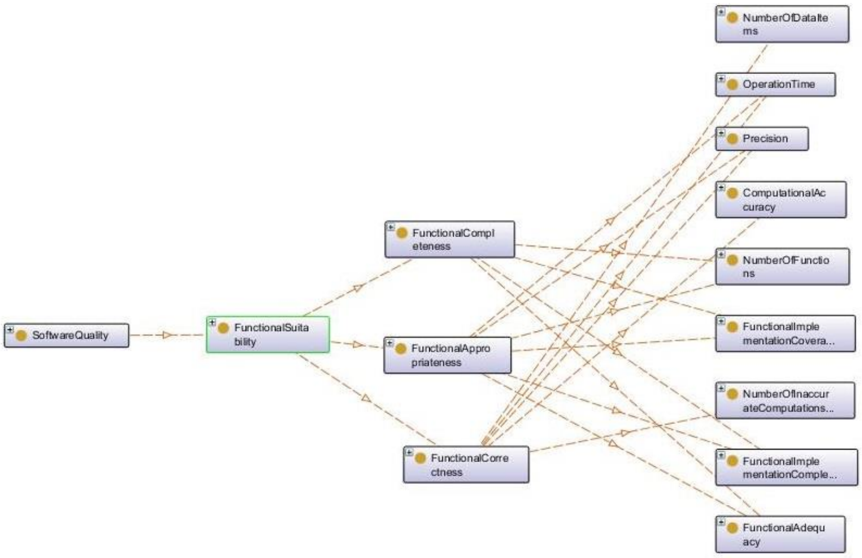

Fig. 2. The component of base ontology for Functional Suitability [22]

Comparison of the developed ontology for the concrete software with the base ontologies provides determining the insufficiency (Fig. 3) or sufficiency (Fig. 4) of the information on quality in the SRS for concrete software.

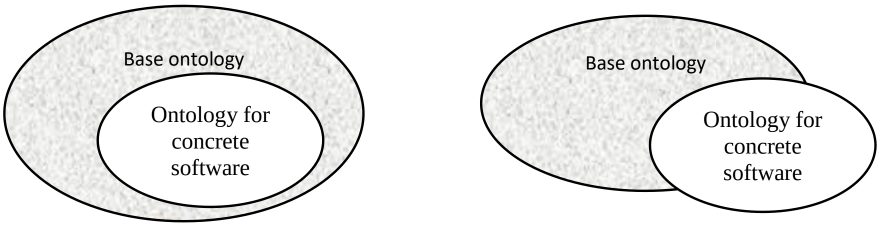

Fig. 3. Two cases of insufficiency of information on quality

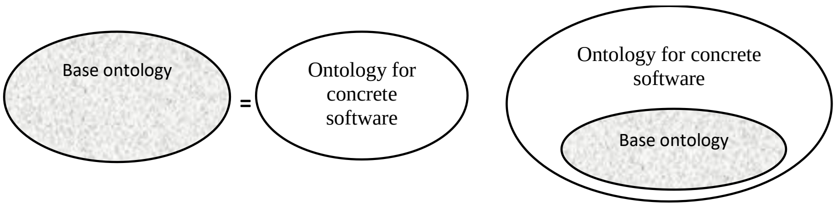

Fig. 4. Two cases of sufficiency of information on quality

The criterion of the sufficiency of information on quality in the SRS. Let SAMI is the set of missing measures and indicators:

$$\text{SAMI} = \text{Base ontology} \setminus (\text{Base ontology} \cap \text{Ontology for concrete software}). \quad (1)$$

Then:

1) if SAMI=∅, then information on quality in the SRS is sufficient;

2) if SAMI≠∅, then information on quality in the SRS is insufficient, and the SRS may require being complemented by information on quality (measures and/or indicators).

Numerical evaluation of the sufficiency of the volume of the available information on quality (measures) in the SRS can be calculated by the formula:

$$D_{chr} = (ksc-((qtmm_1/qtnm_1)+\ldots+(qtmm_{ksc}/qtnm_{ksc}))) / ksc, \quad (2)$$

where $qtmm_i$ – the quantity of missing (in the SRS) measures for i-th software quality characteristic, $qtnm_i$ – the quantity of required measures for i-th software quality characteristic (is determined by the set-theoretical model of software quality [22], and moreover $(qtnm_1+\ldots+ qtnm_{ksc}) = 203$ – total quantity of measures (including and those repeated), from which the software quality characteristics depend), ksc – quantity of software quality characteristics (according to ISO 25010:2011, ksc=8).

Such approach to modeling the information flows movement provide to solve the problem of systematizing all available information on software quality and bringing it to the single unified form (by developing the set-theoretical and ontological models of the subject domain “Software Engineering” (parts “Software quality”,”Software quality. Metric analysis”,”Software requirements specification”) [22, 26]), and the development of the criterion of the sufficiency of information on quality in the SRS based on the basis of the use of ontologies.

The obtained results illustrate the universal approach to evaluating the sufficiency of information on quality in the SRS. At this stage, the information on the subject domain is contained in the developed in [22] models. The impact of any subject domain, for which the software is being developed, on the assessment of software quality and on the sufficiency of information on quality is not considered.

3 Process of Evaluating the Sufficiency of Information on Quality in the Software Requirements Specifications

The result of the evaluating the sufficiency of information on quality in the SRS based on the criterion of the sufficiency may be the conclusion about the absence of the required information in the SRS. For eliminating the fact of insufficiency of information, it is necessary to formulate the repeated request about the requirements, which determine the software quality characteristics. On the basis of such request, the developers of SII should make the necessary complements to the SII (Fig. 5).

The subsystem of the evaluating the sufficiency of information on quality in the SRS on the basis of comparative analysis of ontologies carries out the evaluation of the sufficiency of information on quality. Its functioning is based on the methods and tools of the evaluating the sufficiency of information on software quality, and on the ways (moduses) of filling the ontological models of quality of the concrete software.

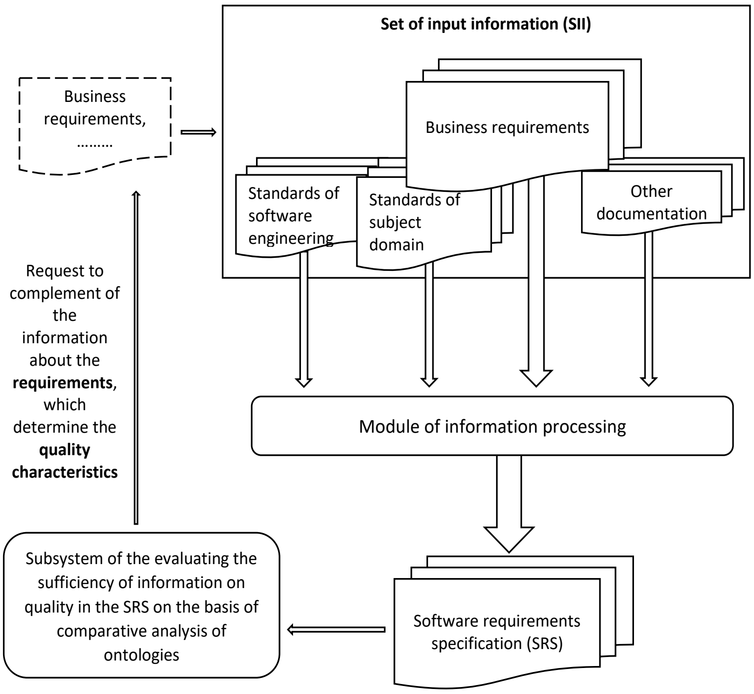

Fig. 5. Repeated requests to developers of SII on complementing the information about the requirements, which determine the quality characteristics

This subsystem works as follows:

1. templates of ontologies for determining the quality of the concrete software (by ISO 25010 and based on the metric information) are generated and filled;
2. the ontologies for determining the quality of the concrete software are compared with the appropriate base ontologies of the subject domain "Software Engineering" (part "Software quality", "Software quality. Metric analysis");
3. on the basis of the criterion of the sufficiency of information on quality in the software requirements specification, the conclusion about the sufficiency or insufficiency of information is drawn;
4. if the information on quality in the specification is sufficient, then the work on software project by this specification are continued;
5. if the information on quality in the specification is insufficient, then the request for complementing the information in the set of input information is made. This request is the list of missing measures and (or) indicators, and the recommended priority for complementing these measures and (or) indicators to the specifications (depending on the importance and significance of the measure and (or) indicator).

The next step is an iteration of the complement of the specification by the information on software quality.

The scheme of process of evaluating the sufficiency of information for determining the software quality by ISO 25010:2011 is presented on Fig. 6.

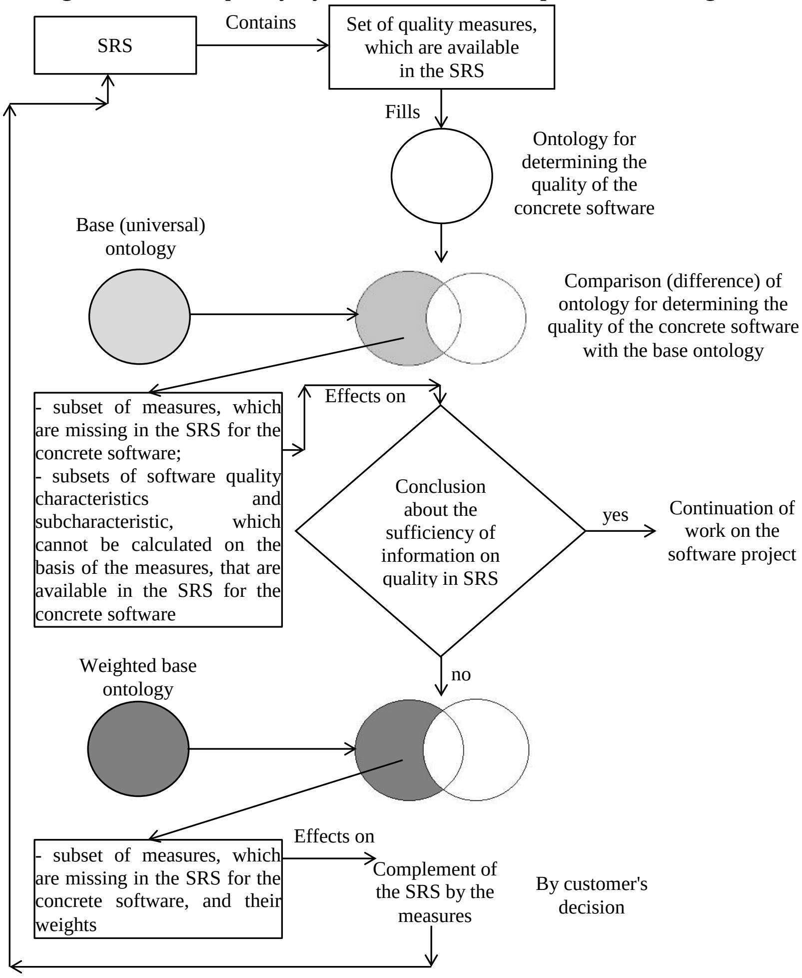

Fig. 6. The scheme of process of evaluating the sufficiency of information for determining the software quality by ISO 25010:2011

The scheme reflects the movement of information flows in the process of evaluating the sufficiency of information on quality in the SRS.

Consequently, the process of evaluating the sufficiency of information for determining the software quality by ISO 25010 consists of: 1) comparison of ontology for determining the quality of the concrete software with the base (universal) ontology – with the purpose of the identification of the measures, which are missing in the ontology for the concrete software, i.e. are missing in the SRS, on the basis of which this ontology was built; and with the purpose of the identification of the software quality characteristics and subcharacteristics, which cannot be calculated on the basis of the measures, which are available in the SRS for concrete software; 2) forming the conclusion about the insufficiency of the information on quality in the SRS for concrete software, if there are measures, which are missing in the SRS, and if there are software quality characteristics and subcharacteristics, which cannot be calculated on the basis of available measures in the SRS for concrete software; 3) comparison of ontology for determining the quality of the concrete software with the weighted base ontology (modified base ontology, in which the software quality measures have weights with the purpose of the recommending the further satisfaction of these measures in the SRS [22]), if the conclusion about insufficiency of information on quality was made - with the purpose of identification of weights of measures, which are missing in the SRS for the concrete software, for further identifying the priority of complement of measures in the SRS.

4 Information Technology of Evaluating the Sufficiency of Information on Quality in the Software Requirements Specifications

On the basis of modeling the movement of information flows during the formation of the SRS and researching the information flows in the process of evaluating the sufficiency of information on quality in the SRS, the information technology of evaluating the sufficiency of information on quality in the SRS is developed. The scheme of the developed information technology is represented on Fig. 7.

Information technology of evaluating the sufficiency of information on quality in the SRS provides: 1) supporting the process of the software quality assessment at the early stages of the lifecycle on the basis of the evaluating the sufficiency of information on quality in the SRS; 2) conclusion about the sufficiency of information on quality in the SII and in the SRS; 3) the priority of complement of the SII and the SRS by the necessary information (in case of insufficient information) by creating the request for the complement of business requirements to the software; 4) numerical evaluation of the sufficiency of the volume of the available information on quality (measures) in the SRS; 5) numerical evaluation of the gain of the sufficiency of the volume of the available information on quality (measures) in the SRS after its complement; 6) the possibility to processing the information on quality in the SRS by software agents (bots), without the participation of specialists, which provides the automation of these processes and the elimination of the subjective influence of specialists, and also saving this information in the software company at the release of the specialist.

The basis of information technology of evaluating the sufficiency of information on quality in the SRS is the use of ontologies for providing the information on quality in the SRS. The advantages of using the ontologies are the automatic processing of their content, the possibility of the accessing, understanding and analyzing the information by intellectual and non-intellectual virtual agents, the processing of Web-based information, sharing of information and exchanging of it between applications, and the detection of duplication and knowledge gaps based on visualization of missing logical connections.

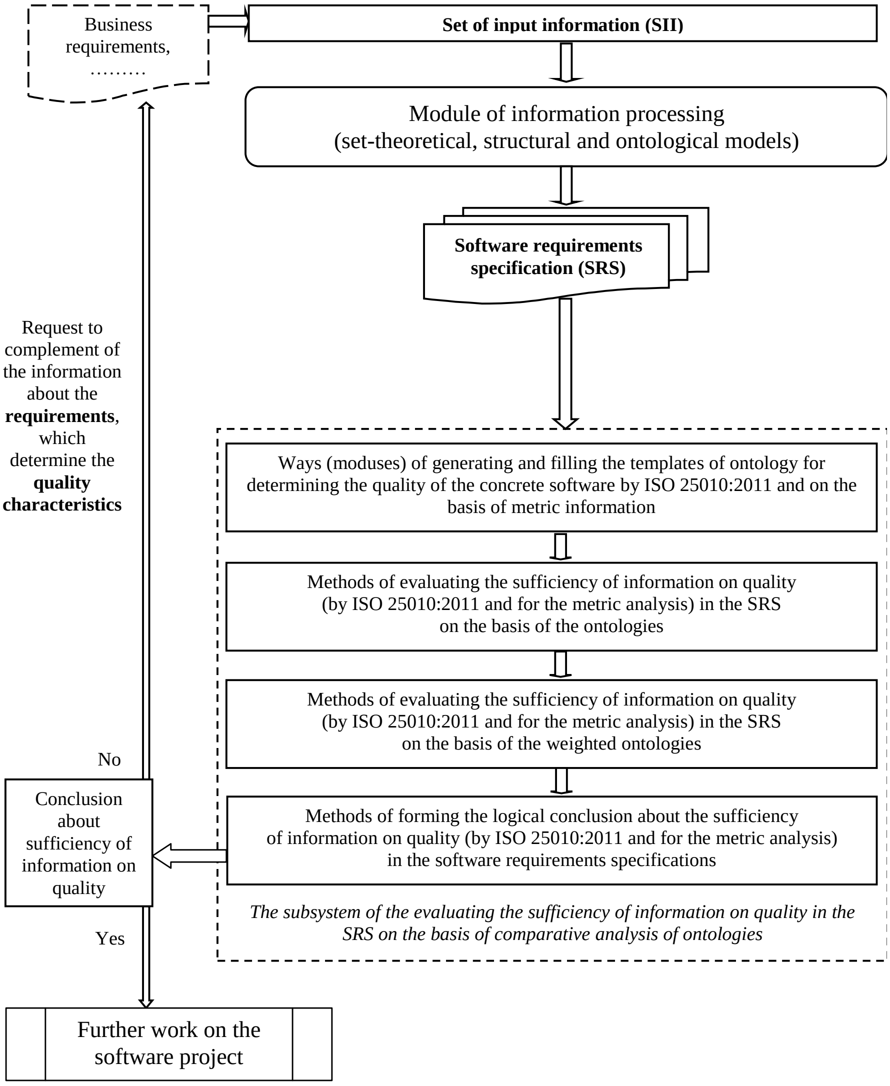

Fig. 7. Scheme of the information technology of evaluating the sufficiency of information on quality in the SRS

Nowadays, the information technologies foresee the human-machine interaction at all stages of processing information, during which all information is interpreted by the person, that often leads to subjective assessments and losses of important information. But the need of processing the large volumes of information and the development of various directions of intellectualization of the information processing are prerequisites for the transition to a new quality level of information processing, where the person is eliminated from processes of information processing and knowledge gaining.

The evolution of information technology of evaluating the sufficiency of information on quality in the SRS is possible through the development of program agents (bots) on the basis of the ontological approach. Implementation of the developed information technology in software companies will automate the process of evaluating the sufficiency of information on quality in the SRS, will provide the use of software agents for analysis and complement of the specification, which in the end provides the prerequisites for the appropriate level of software quality.

5 Experiments

For the experiment, the SRS of information system (IS) for the store and warehouse of spare parts for trucks was analyzed. The measures, which are available in this SRS, were identified. The ontology for determining the quality of this software was developed in Protégé 4.2. The comparison (in Protégé 4.2) of the developed ontology for IS for the store and warehouse of spare parts for trucks with the base ontology for subject domain “Software Engineering” (part “Software quality”) provides the conclusion, that in the ontology for this software 6 (six) measures are missing: “Operation Time”, “Number of Faults”, “Number of Failures”, “Number of Resolved Failures”, “Number of Controllability Requirements”, “Data Exchangeability”. By the method of forming the logical conclusion about the sufficiency of information on quality (by ISO 25010) in the SRS [22, 26], searching the rule for each missing measure is performed. According to these rules, the quantities of missing (in the SRS) measures for each software quality characteristic are counted.

According to the rules, the fact was established, that the available measures in the SRS of IS for the store and warehouse of spare parts for trucks are insufficient for calculation of all 8 software quality characteristics. So the information technology of evaluating the sufficiency of information on quality in the SRS gives the conclusion: “The information on quality in the SII and SRS of IS for the store and warehouse of spare parts for trucks is insufficient for software quality assessment according to ISO 25010”.

After establishing the fact of insufficiency of information of the SRS of IS for the store and warehouse of spare parts for trucks, on the basis of the comparative analysis of the ontology for the concrete software with the weighted base ontology of subject domain “Software Engineering” (part “Software quality”), the information technology of evaluating the sufficiency of information on quality in the SRS gives the conclusion: “For increasing the sufficiency of the volume of information on quality the next measures should be added in the SII and SRS in this consistency: 1) Operation Time; 2) Number of Failures; 3) Number of Reolved Failures; 4) Number of Faults; 5) Number of Controllability Requirements; 6) Data Exchaneability”.

Next, the evaluation of the sufficiency of the volume of information on quality, based on the available information in the SRS, is done – according to the numerical evaluation of the sufficiency of the volume of the available information on quality (measures) in the SRS (by equation 2):

$$D_{chr} = (8-(2/15+7/30+3/26+2/49+8/33+4/23+4/9+3/18)) / 8 = 0.81.$$

So the information technology of evaluating the sufficiency of information on quality in the SRS gives the conclusion: "The sufficiency of the volume of the available information on quality (measures) in the SRS is 81%".

Because there are software quality characteristics, for calculation of which the available information is insufficient, then the request for complementing the information in the SII and the SRS is made. After complement of the SII and SRS of IS for the store and warehouse of spare parts for trucks, the ontology (version 2) for determining the quality of this software was re-developed. The comparison of the re-developed ontology for IS for the store and warehouse of spare parts for trucks with the base ontology for subject domain "Software Engineering" (part "Software quality") provides the conclusion, that 3 (three) measures are still missing in this SRS: "Number of Failures", "Number of Resolved Failures", "Number of Controllability Requirements".

By the method of forming the logical conclusion about the sufficiency of information on quality (by ISO 25010) in the SRS [22, 26], searching the rule for each missing measure is performed. According to these rules, the quantities of missing (in the SRS) measures for each software quality characteristic are again counted. According to the rules, the fact was established, that the available measures in the SRS of IS for the store and warehouse of spare parts for trucks are still insufficient for calculation of 5 (from 8) software quality characteristics. So the information technology of evaluating the sufficiency of information on quality in the SRS again gives the conclusion: "The information on quality in the SII and SRS is insufficient for software quality assessment according to ISO 25010:2011".

After establishing the fact of insufficiency of information of the SRS of IS for the store and warehouse of spare parts for trucks, on the basis of the comparative analysis of the ontology for the concrete software with the weighted base ontology of subject domain "Software Engineering" (part "Software quality"), the information technology of evaluating the sufficiency of information on quality in the SRS gives the conclusion: "For increasing the sufficiency of the volume of information on quality the next measures should be added in the SII and SRS in this consistency: 1) Number of Failures; 2) Number of Reolved Failures; 3) Number of Controllability Requirements".

Next, the evaluation of the sufficiency of the volume of information on quality, based on the available information in the SRS (after complement), is done – according to the numerical evaluation of the sufficiency of the volume of the available information on quality (measures) in the SRS (by equation 2):

$$D'_{chr} = (8-(0/15+3/30+1/26+0/49+5/33+2/23+1/9+0/18)) / 8 = 0.94.$$

So the information technology of evaluating the sufficiency of information on quality in the SRS gives the conclusion: "The sufficiency of the volume of the available information on quality (measures) in the SRS is 94%".

The gain of the sufficiency of the volume of the available information on quality (measures) in the SRS after its complement is 13% ($\Delta D_{chr} = D'_{chr} - D_{chr} = 0.94 - 0.81 = 0.13$) for IS for the store and warehouse of spare parts for trucks.

6 Conclusions

The research of the composition and movement of information flows at the early stages of the software lifecycle showed that the functions of the future software, its features and limitations are described by business requirements and standards, which form the set of input information. On their basis the SRS is formed, therefore, characteristics of the SRS are largely determined by the characteristics of the SII. For identification of future problems with the quality of the developed software, it is necessary to research the characteristics of the SII with the purpose of eliminating the problems and disadvantages at the initial stages of the lifecycle. In the process of such research, there is the problem of evaluating the sufficiency of information on quality in the SII and in the SRS. The conducted modeling of information flows movement during the formation of the SRS and the research of information flows in the process of evaluating the sufficiency of information on quality in the SRS showed the need of the developing the information technology of evaluating the sufficiency of information on quality in the SRS. Such technology is based on the use of ontologies for evaluating the sufficiency of information on quality in the SRS.

The advantage of the developed information technology is the possibility of the partial elimination of the person from the processes of processing information of the SRS, which provides the saving of this information in the company at the release of the specialist.

The evolution of the developed information technology is possible through the use of descriptive logic and reasoners for verification of built ontologies and the withdrawal of new knowledge. Implementation of the evolving information technology in software companies will help to automate the process of developing the SRS, will provide the use of software agents to complement of the SRS or obtain of new knowledge on its basis. Therefore, one of the perspective directions of further research, the authors consider the development of intelligent agents based on the ontological approach, which will automatically analyze of the developed SRS, will automatically supplement the SRS on the basis of ontologies and will automatically evaluate the initial stages of the lifecycle with purpose of the minimizing the informational and financial losses.

References

1. Sivarajah, U., Kamal, M. M., Irani, Z., Weerakkody, V.: Critical analysis of Big Data challenges and analytical methods. Journal of Business Research. 70, 263-286 (2017).
2. Luo, Y. D., Bu, J.: How valuable is information and communication technology? A study of emerging economy enterprises. Journal of World Business. 51(2), 200-211 (2016).
3. Tian, X. M.: Big data and knowledge management: a case of deja vu or back to the future? Journal of Knowledge Managament. 21(1), 113-131 (2017) .
4. Mauerhoefer, T., Strese, S., Brettel, M.: The impact of information technology on new product development performance. Journal of Product Innovation Management. 34(6), 719-738 (2017).
5. Qiu, J. F., Wu, Q. H., Ding, G. R., Xu, Y. H., Feng, S.: A survey of machine learning for big data processing. EURASIP Journal on Advances in Signal Processing. Paper 67 (2016).

6. Noor, A. K.: Potential of cognitive computing and cognitive systems. Open Engineering. 5(1), 75-88 (2015).
7. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature. 521(7553), 436-444 (2015).
8. Ristoski, P., Paulheim, H.: Semantic Web in data mining and knowledge discovery: A comprehensive survey. Journal of WEB Semantics. 36, 1-22 (2016).
9. Golub, K.: Subject access to information: An interdisciplinary approach. Libraries Unlimited, Westport (2015).
10. Osis, J., Asnina, E.: Model-Driven Domain Analysis and Software Development: Architectures and Functions. IGI Global, Hershey (2010).
11. Hastie, S., Wojewoda, S.: Standish Group 2015 Chaos Report – Q&A with Jennifer Lynch, http://www.infoq.com/articles/standish-chaos-2015, last accessed 2018/03/23.
12. 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).
13. McConnell, S.: Code complete. Microsoft Press, Redmond (2013).
14. Wiegers, K., Beatty, J.: Software requirements: 3rd edition. Microsoft Press, Redmond (2013).
15. Ambler, S. W.: Beyond functional requirements on agile projects – Strategies for addressing nonfunctional requirements. Dr Dobbs Journal. 33 (10), 64-66 (2008) .
16. Cockburn, A.: Writing effective use cases. Addison-Wesley Professional, Boston (2001).
17. Pomorova, O., Hovorushchenko, T.: Intelligent assessment and prediction of software characteristics at the design stage. American Journal of Software Engineering and Applications. 2 (2), 25-31 (2013).
18. Jones, C., Bonsignour, O.: The economics of software quality. Pearson Education, London (2012).
19. Managing Agile projects (edited by K. J. Aguanno). Multi-Media Publications, Ontatio (2005).
20. ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models (2011).
21. ISO 25023:2016. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). Measurement of system and software product quality (2016).
22. Hovorushchenko, T.: Information technology for assurance of veracity of quality information in the software requirements specification. Advances in Intelligent Systems and Computing II. 689, 166-185 (2018).
23. Delayne Stroud, J.: Business Requirements Document: A high-level review, https://www.isixsigma.com/implementation/project-selection-tracking/business-requirements-document-high-level-review/, last accessed 2018/03/23.
24. Dermeval, D., Vilela, J., Bittencourt, II., Castro, J., Isotani, S., Brito, P., Silva, A.: Applications of ontologies in requirements engineering: a systematic review of the literature. Requirements engineering. 21 (4), 405-437 (2016).
25. Dou, DJ., Wang, H., Liu, HS.: Semantic Data Mining: A Survey of ontology-based approaches. In: 2015 IEEE 9th International Conference on Semantic Computing, pp. 244-251. IEEE Press, New York (2015).
26. Hovorushchenko, T., Pomorova, O.: Ontological Approach to the Assessment of Information Sufficiency for Software Quality Determination. CEUR-WS. 1614, 332-348 (2016).
27. Fenton, N., Bieman, J.: Software metrics: a rigorous and practical approach. CRC Press, Boka-Raton (2014).
28. Abran, A.: Software project estimation: the fundamentals for providing high quality information to decision makers. Wiley-IEEE Computer Society Press, Los Alamitos (2015).
29. Hovorushchenko, T.: Method of Evaluating the Weights of Software Quality Measures and Indicators. Application and Theory of Computer Technology. 2, 2 (2), 16-25 (2017).