Date of publication MAR-30, 2022, date of current version JAN-14, 2022.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.21.1.2524

Software Requirements Profile Quality Model

OLEKSANDR GORDIEIEV[1], VYACHESLAV KHARCHENKO[2], DARIA GORDIEIEVA[3]

[1]Software engineering department, Lutsk National Technical Univerity, Lvivska Str., 75, Lutsk, 43018, Ukraine, (e-mail: oleksandr.gordieiev@gmail.com), https://lntu.edu.ua/

[2]Dept. of Computer Systems, Networks and Cybersecurity, National Aerospace University "Kharkiv Aviation Institute", 17, Chkalov Street, Kharkiv, 61070, Ukraine, (e-mail: v.kharchenko@csn.khai.edu), http://khai.edu

[3]Department of Computer Engineering and Cybersecurity, Lutsk National Technical Univerity, Lvivska Str., 75, Lutsk, 43018, Ukraine, (e-mail: gordeyeva.daria@gmail.com), https://lntu.edu.ua/

Corresponding author: Oleksandr Gordieiev (e-mail: oleksandr.gordieiev@gmail.com).

ABSTRACT Article opens series of works devoted to profile-oriented software quality assessment. In this article the concept of software requirements profiling for subsequent software requirements profile quality assessment is analyzed and developed. The main result of the article is the development of the software requirements profile quality model. The model describes the following: characteristics and attributes of software requirements and their classification features; characteristics and attributes of software requirements profile and their classification features; semantics and syntax of software requirements. The article is based on analysis and use of the following standards: ISO/IEC 25012:2008, ISO/IEC/IEEE 29148:2018, ISO/IEC/IEEE 29148:2011. Examples of software requirements profile quality model are introduced. The suggested approach is used for the development of terms of reference or draft of a new standard. For example, the model is used for an assessment draft of the new standard "Requirements to computer security of NPP Instrumentation and Control Systems (NPP I&C)" developed by the Ukrainian state regulatory body. As a result of the development of the software requirements profile quality model, a set of propositions for improving the quality of the standard as a branch of the profile for NPP I&C cybersecurity are implemented.

KEYWORDS software quality; software requirements profile; software requirements profile model; software quality model; ISO/IEC 25012:2008; ISO/IEC/IEEE 29148:2018; ISO/IEC/IEEE 29148 :2011.

I. INTRODUCTION

A. MOTIVATION

SOFTWARE quality depends on many factors in duration of its development, but "foundation of quality is laid" on the first stages of software development. Basis of such foundation of software quality provides the terms of reference (specification) for software development. As a rule, the basic part of notions of reference represents a set of requirements, which are logically connected by a single structure. Such structure has an established name – software requirements profile [1]. Logically, all requirements which form software requirements profile can be divided into two groups: functional and non-functional requirements.

The basis of non-functional requirements is software quality models [2, 3]. The results of studies of such models [4-9] allowed to establish an interaction between characteristics of software quality models, for example, between the pairs of characteristics "security" and "usability" [7], "greenness" and "reliability" [6] or "triplet of usability", "security" and "safety" [9] etc. [5]. Such interactions between characteristics (i.e., non-functional requirements) on the one hand have a constructive character, when characteristics complement each other, on the other hand – destructive character, when characteristics compete with each other. Results of such research give possibilities that are represented by the following:

– even individual software requirement is a complex object due to its semantic structure. Its role is more responsible rather than general sentence outside software requirements profile;

– software requirements profile unlike individual software requirement is a more complex structure. The software requirements profile combines in itself a set of software requirements in a united "harmonious" structure.

Therefore, for software quality assessment it is important not only to use an adequate software quality model but also software requirements quality model, which would allow the most exact assessment of its peculiar "skeleton".

B. RELATED WORK ANALYSIS

Existing works, in which a quality of software profile requirements was analyzed, should be divided into the following groups:

– articles, in which review of some aspects of software requirements quality models is done at the level of separate non-functional requirements [10-12];

– articles, in which review quality of software requirements profile is done at the level of taxonomic structure excluding semantic content of requirements [13-15];

– articles, which review only separate elements of quality of software requirements profile [16-18] including human-computer systems [19].

Paper [20] describes framework to improve the software development quality considering knowledge, operation experience and work with clients. Authors of [21] suggested the hierarchical fuzzy logic based model of the assessment and the criteria taking into account degree of the expert confidence.

Analysis of the works allows us to form following objectives of the paper:

– to analyze and represent elements of software requirements quality model and software requirements profile in general;

– to represent and describe software requirements profile quality model;

– to represent the software requirements quality model and the software requirements profile quality model as a single conception;

– to formulate certain stages and tasks for the development of conception of profile-oriented assessment of software requirements quality.

The paper is structured as follows: section 2 describes terms, which are used in the article, and basic elements of software requirements profile quality model; in section 3 a quality model in part of characteristics, attributes, semantics and syntax of requirements and classification features is represented; in section 4 a continuation of model in part of elements of software requirements profile is represented directly; in section 5, an example of using software requirements profile quality model is introduced.

II. INITIAL NOTIONS AND APPROACH TO ASSESSMENT OF QUALITY OF SOFTWARE REQUIREMENTS PROFILE

The following basic notions are represented in the article:

– software requirement is a statement which translates or expresses a need and its associated constraints and conditions;

– software requirements profile is a set of requirements, which are combined in a single structure;

– requirement characteristic is a set of features defining distinctive particularities of a requirement;

– a quality of requirements profile is a complex notion, which combines in itself on the one hand a quality of each requirement at the profile, on the other hand – a quality of all set of requirements at software requirements profile. It is a constituent of software quality;

– requirement attribute is a field of entity (software requirements, software requirements profile, classification features) that can be distinguished quantitatively or qualitatively by human or automated means;

– software requirements profile quality model is a tool of formal description and combination of elements software requirements profile quality.

Results of preliminary analysis [19-21] give a possibility to represent the following nomenclature of elements, which are combined in two groups. The first group of elements of requirements quality model in software requirements profile is an internal quality of software requirements profile (IQSRP), which includes: software requirements characteristics; software requirements classification features characteristics; software requirements attributes; software requirements classification features attributes; semantics and syntax of software requirements.

The second group of software requirements profile quality model is an external quality of software requirements profile (EQSRP), which includes: structure of software requirements profile; characteristics of software requirements profile; characteristics of software requirements profile classification features; attributes of software requirements profile; attributes of software requirements profile classification features.

Representation and description of software requirements profile quality model is a first stage of the process, which is directed at an assessment of software requirements profile quality. This article deals with a representation and description only of software requirements profile quality model.

III. SOFTWARE REQUIREMENTS PROFILE QUALITY MODEL (FIRST GROUP OF ELEMENTS - IQSRP)

A. CHARACTERISTICS AND CLASSIFICATION FEATURES OF SOFTWARE REQUIREMENTS

As a result of the analysis characteristics in [22-24] and their ordering, there was formed a general set of characteristics of requirements in the context of software

requirements profile. It consists of the following 20 features: accuracy; completeness; consistency; credibility; correctness; accessibility; compliance; confidentiality; traceability; understandability; necessary; implementation free; unambiguous; consistent; singular; feasible; verifiable; appropriate; correct; conforming.

Formal designation for the set of requirement characteristics in context software requirements profile is as follows:

– $SIRCSRP = \{sircsrp_j\}_{j=1}^{20}$ – a set of software individual requirement characteristics in software requirements profile (*SIRCSRP*), sircsrp – a software individual requirement characteristic in software requirements profile.

It is worth noting that the characteristics and the attributes of software requirement and its classification feature will be different. It is connected with the difference in their intended purpose. Software requirement is a more complex construction and the task of its forming is more responsible. The task of the classification feature is to exactly determine a place of a requirement in the general structure of software requirements profile. Accordingly, elements of the quality model for requirements and their classification features differ and are considered separately.

The set of classification features characteristics of software requirements consists of the following elements: uniqueness; evidence; stability; simplicity of discovery; positioning; accuracy; compliance; understandability; univocacy; consistency.

Formal designation for the set of classification features characteristics of software requirements profile is as follows:

– $SIRCFCSRP = \{sircfcsrp_j\}_{j=1}^{10}$ – a set of software individual requirement classification features characteristics in software requirements profile (*SIRCFCSRP*), *sircfcsrp* – software individual requirement classification features characteristic in software requirements profile.

B. ATTRIBUTES FOR SOFTWARE REQUIREMENTS AND ATTRIBUTES OF THEIR CLASSIFICATION FEATURES

A set of attributes for requirements in the context of software requirements profile was formed. Such set includes the following attributes: identification, stakeholder priority, risk, source, rationale, difficulty, type, version number and dependency. Therefore, software requirements profile can have the corresponding 8 attributes. Let us introduce formal designation for such set of attributes:

– - $SIRASRP = \{sirasrp_j\}_{j=1}^{9}$ – a set of software individual requirement attributes in software requirements profile (*SIRASRP*), *sirasrp* – software individual requirement attribute in software requirements profile.

A set of attributes for requirements classification features in the context of software requirements profile was formed. Such set includes the following elements: identification and dependency. Let us introduce formal designation for such set of attributes:

– $SIRCFASRP = \{sircfasrp_j\}_{j=1}^{2}$ – a set of software individual requirement classification feature attributes in software requirements profile (*SIRCFASRP*), *sircfasrp* – software individual requirement classification feature attribute in software requirements profile.

C. REQUIREMENTS FOR SEMANTICS AND SYNTAX OF SOFTWARE REQUIREMENTS

Semantic constructions for each software requirement can be logically divided into the following groups: mandatory, admissible and undesirable. A more detailed analysis of each group of semantic constructions is presented below:

– a group of mandatory semantic constructions consists of more important semantic elements of the requirement. For example, the most widespread and mandatory element for the requirement is “shall”;

– a group of admissible semantic constructions consists of the elements, which do not contradict to the meaning of the requirement and can be used in it. For example, “should”, “may” etc.;

– a group of undesirable semantic constructions are the elements of the requirements, which can be a source of equivocation of a requirement interpretation, distortion of its meaning, an incompleteness of the requirement, its inaccuracy etc. For example, application words or phrases in a superlative degree are “best”, “most”; subjective phrases – “such as user friendly”, “easy to use”, “cost effective”, “shall be able to select”; ambiguous phrases – “almost always”, “significant”, “minimal”; unverifiable phrases – “provide support”, etc.

Formal designation for such groups are as follows:

– $MSC = \{msc_i\}_{i=1}^{n}$ – a set of mandatory semantic constructions, *msc* – a mandatory semantic construction;

– $ASC = \{asc_i\}_{i=1}^{n}$ – a set of admissible semantic constructions, *asc* – an admissible semantic construction;

– $USC = \{usc_i\}_{i=1}^{n}$ – a set of undesirable semantic constructions, *usc* – an undesirable semantic construction.

Results of the general analysis of semantic classification features of individual software requirements show us, that specific requirements to their semantics are absent. Thus, the absence of requirements to semantics can be represented as follows:

– $SIRCFSR = \varnothing$ – a set of software individual requirement classification feature semantic requirements.

Each requirement consists of one and more sentences. It is necessary to consider and formulate the requirements for the structure of the sentences included in an individual

requirement. After the analysis of the following sources [22-29], in which is presented a review of possible variants of syntactic structures of requirements, a maximally general syntactic structure for software requirement was formed. Such general syntax structure for the software requirement (Fig. 1) consists of 6 elements. A detailed description of each element in the syntactic structure was represented as the following ones:

1. Condition. Usually, the condition is put at the beginning of the requirement. Condition can begin from the following words: "when", "if", "while", "where", "which";

2. Topic. Usually, the topic of requirement is needed for the definition of its assignment. For example, "information system", "software", "computer-based system", etc.;

3. Action. This is an action of the requirement, which has to be performed in the context of this requirement. For example, "install", "disable", "switch", "reset", "form", "test", "run", etc.;

4. Object is an object of the requirement, i.e., an object, for which this requirement was intended. For example, "indicator", "signal", "carrier frequency", "switch", etc.;

5. Limitation is a limitation, which is important when performing the specified action. Worth noting, limitations are usually used together with a condition. For example, "with regarding to", "if installed", "while in progress", etc.;

6. Value is a value, which the object has to receive when the requirement is fulfilled. For example, "yes" or "no", "0" and "1", [0…..1], "on" and "off".

Thus, a requirement in part of its structure is a set of 6 possible elements. Formal designations for the set of requirement structure elements are as follows:

– $RSSE = \langle rsse \rangle_{i=1}^{n}$ – a set of requirement syntactical structure elements (*RSSE*), rsse – a requirement syntactical structure element. Elements of the set *RSSE* are represented as a tuple, because their sequence is important and must not be broken.

The quantity of requirement structure elements (or a power of a set *RSSE*) is not permanent, i.e., $|RSSE| \neq const$. Therefore, it is evident, each requirement can have different syntactical structure elements quantity, i.e., the quantity of structural elements can change depending on a requirement. When the syntactical structure of requirement is forming, the following peculiarities were identified:

– an element of $rsse_5$ set (Limitation) can be represented in syntactical structure as 2 variants: a limitation with a condition and a limitation without a condition. Thus, for the identification of such variants, an additional sub-index was added. Variants were denoted in the following way: a limitation with a condition – $rsse_{5.1}$, a limitation without a condition – $rsse_{5.2}$;

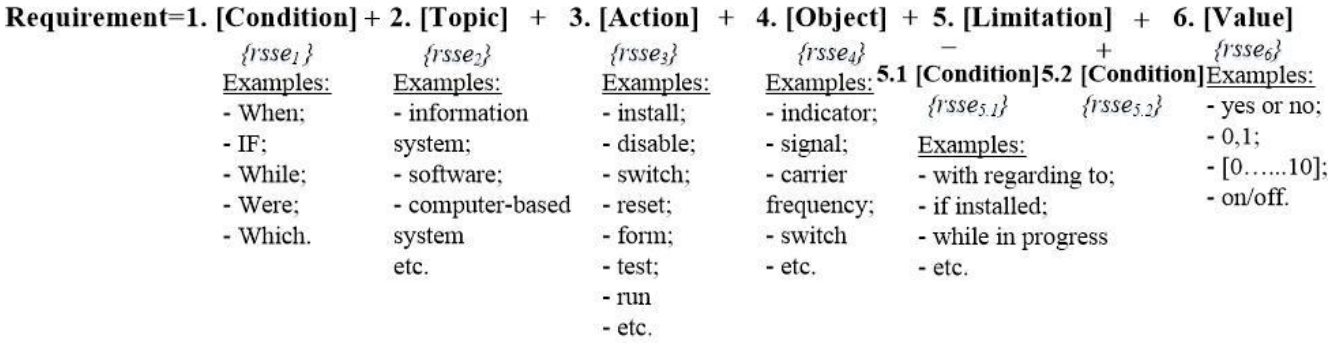

Figure 1. Summarized syntactical structure of software requirement

– elements of a set of requirement syntactical structure, which are included in all variants, i.e., they are permanent. There are 2 elements: $rsse_2$ (Topic) and $rsse_3$ (Action).

Thus, all elements of the syntactical structure were divided into 2 following groups:

– $RSSEP = \langle rssep \rangle_{i=1}^{n}$ – a set of permanent elements of requirement syntactical structure, *rssep* – a permanent element of requirement syntactical structure ;

– $RSSEN = \langle rssen \rangle_{i=1}^{n}$ – a set of non-permanent elements of requirement syntactical structure, *rssen* – a non-permanent element of requirement syntactical structure.

Thus, the full set of possible variants of syntactical structure of requirement taking into account the variable number of its elements can be formed and represented in a more formalized view. But such set of variants will not be presented in this article. Let us just present as an example several variants for such structures:

$$RSSE = \langle rsse_2, rsse_3 \rangle,$$

$$RSSE = \langle rsse_1, rsse_2, rsse_3, rsse_4, rsse_{5.2}, rsse_6 \rangle.$$

General syntax analysis of classification features of software requirements showed us that special requirements in part of their syntax are absent too. Thus, the absence of the requirements for the syntax of classification features can have the following representation:

– $SIRCFSYR = \varnothing$ – a set of requirements to syntax of software individual requirement classification feature (*SIRCFSYR*).

IV. SOFTWARE REQUIREMENTS PROFILE (SECOND GROUP OF ELEMENTS - EQSRP)

A. STRUCTURE OF SOFTWARE REQUIREMENTS PROFILE

The structure of software requirements profile is represented in Fig. 2. The structure of software requirements profile includes 2 interconnected sets: a set of software requirements (semantic taxons) and a set of classification features [13]. The semantic aspect of software requirements profile in this part of the article will not be considered, because it was considered earlier. The set of software requirements (semantic taxons), the set of classification features and connections between them were analyzed. Particularities of software requirements profile structure according to types of their taxonomic structures [13] were analyzed. Formal designations for representation of analysis results are as follows:

– $ST_{tts(i)}$ – a set of semantic taxons (*ST*), *tts* – a type of taxonomic structure – hierarchy (H) or facet (F), *i* – a number of taxonomic structure;

– $STE_{tts(i),j}$ – a set of semantic taxon elements (*STE*), *tts* – a type of taxonomic structure, *i* – a number of taxonomic structure, *j* – a number of element in order;

– $SCF_{tts(i)}$ – a set of semantic classification feature *(SCF)*, *tts* – a type of taxonomic structure, *i* – a number of taxonomic structure;

– $SCFE_{tts(i),j}$ – a set of semantic classification feature elements (*SCFE*), *tts* – a type of taxonomic structure *i* – a number of taxonomic structure, *j* – a number of element in order.

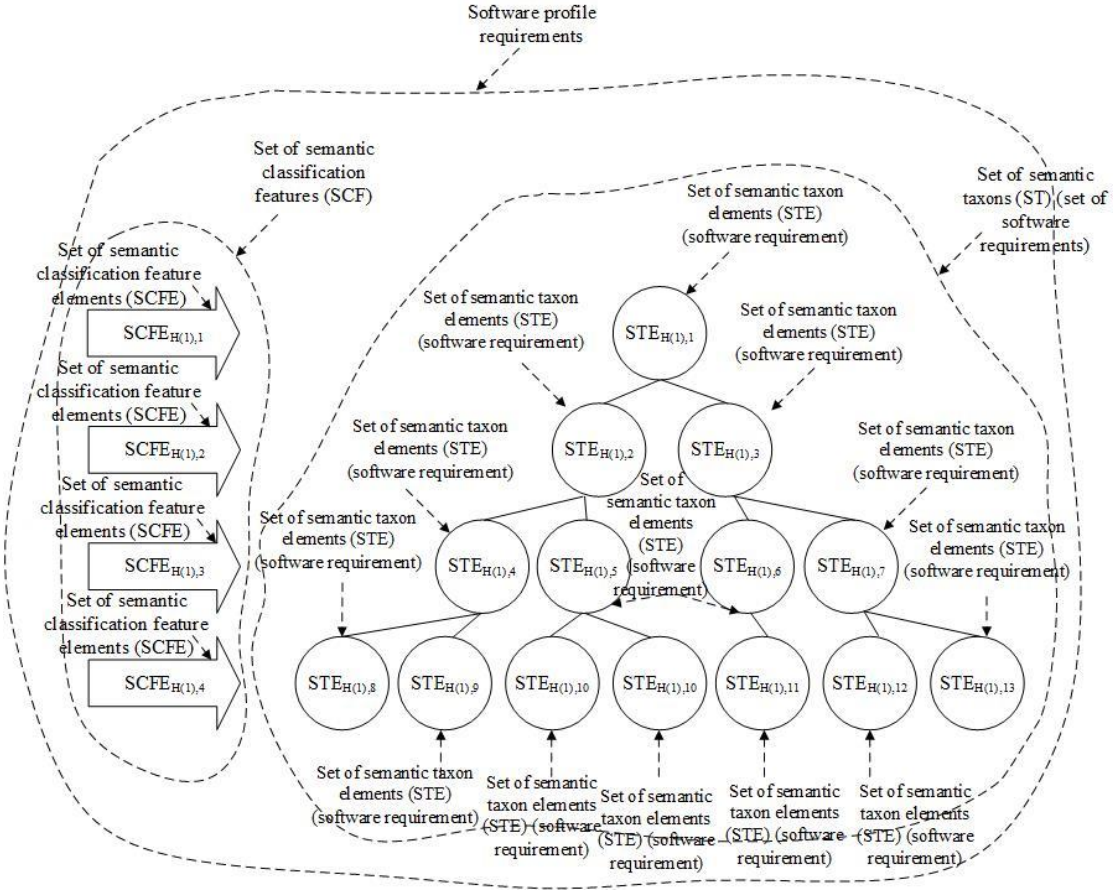

Figure 2 – Structure of software requirements profile (for example, hierarchical structure)

Following types of taxonomic structures are distinguished:

– hierarchical structure (Fig. 2) is a type of taxonomic structure, which is characterized by a multilevel form of object organization with strict subordination of the objects of lower level to the object of high level. Subordination in a hierarchical structure is organized due to

classification features. Thus, hierarchical structures are described by two following sets [13]:

$$SCF_{H(1)} = \begin{Bmatrix} SCFE_{H(1),1}, SCFE_{H(1),2}, \\ SCFE_{H(1),3}, SCFE_{H(1),4} \end{Bmatrix}$$ – a set of semantic classification features of software requirements profile (*SCF*), *SCFE* – a set of classification feature elements;

$$ST_{H(1)} = \begin{Bmatrix} STE_{H(1),1}, STE_{H(1),2}, STE_{H(1),3}, STE_{H(1),4}, \\ STE_{H(1),5}, STE_{H(1),6}, STE_{H(1),7}, STE_{H(1),8}, \\ STE_{H(1),9}, STE_{H(1),10}, STE_{H(1),11}, STE_{H(1),12}, \\ STE_{H(1),13} \end{Bmatrix}$$ – a set of semantic taxons (*ST*), *STE* – a set of semantic taxon elements.

Correlations in the hierarchical structure between software requirements (semantic taxons) are described by an adjacency matrix. Correlations between software requirements (semantic taxons) and classification features are described by correspondence matrix;

– facet structure is a type of classification structure, which is divided into semantic taxons by several classification features simultaneously. Orthogonality of facet structure is realized due to classification features. Thus, the following two sets are described in facet structures [13]:

$$SCF_{F(1)} = \left\{ SCFE_{F(1),1}, SCFE_{F(1),2}, SCFE_{F(1),3} \right\}$$ – a semantic classification feature of software requirements profile (*SCF*), *SCFE* – a set of classification feature elements;

$$ST_{F(1)} = \begin{Bmatrix} STE_{F(1),1}, STE_{F(1),2}, STE_{F(1),3}, \\ STE_{F(1),4}, STE_{F(1),5}, STE_{F(1),6}, \\ STE_{F(1),7}, STE_{F(1),8}, STE_{F(1),9} \end{Bmatrix}$$ – a set of semantic taxons (*ST*), *STE* – a set of semantic taxon elements.

Correlations between software requirements (semantic taxons) and classification features are described by a correspondence matrix.

It should be noted that there are still mixed structures in which elements of hierarchical and facet structures are combined. For example, facet structure can include particular hierarchical structures.

B. CHARACTERISTICS OF SOFTWARE REQUIREMENTS PROFILE AND ITS CLASSIFICATION FEATURES

Characteristics, which relate directly to the whole software requirements profile, are the following: complete; consistent; affordable; bounded; feasible; comprehensible.

Formal designation for a set of characteristics of software requirements profile is as follows:

– $CSRP = \left\{ csrp_j \right\}_{j=1}^{6}$ – a set of characteristics for software requirements profile (*CSRP*), *csrp* – a characteristic for software requirements profile.

It should be noted that compliance of the software requirements profile with the indicated characteristics prevents changes in requirements and their growth in requirements ("creep of requirements") during the software development life cycle, which will affect the cost, development time or quality of software.

Characteristics of classification features/attributes of software requirements profile are the following: all-sufficient; indivisibility; fullness; accuracy.

Formal designation for such characteristics is as follows:

– $CFCSRP = \left\{ cfcsrp_j \right\}_{j=1}^{4}$ – a set of classification features characteristics for software requirements profile (*CFCSRP*), $cfcsrp_j$ – a classification features characteristic.

C. ATTRIBUTES AND CLASSIFICATION FEATURES OF THE SOFTWARE REQUIREMENTS PROFILE

Attributes of software requirements profile are the following: software requirements profile version; software requirements profile complexity; software requirements profile independence.

Formal designation for such attributes is as follows:

– $ASRP = \left\{ asrp_j \right\}_{j=1}^{3}$ – a set of attributes for software requirements profile (*ASRP*), *asrp* – an attribute for software requirements profile.

Attributes of classification features of software requirements profile are the following: structure complexity; taxonomy type.

Formal designation for such attributes is as follows:

– $CFASRP = \left\{ cfasrp_j \right\}_{j=1}^{2}$ – a set of classification features attributes for software requirements profile (*CFASRP*), *cfasrp* – a classification features attribute for software requirements profile.

D. STRUCTURE OF SOFTWARE REQUIREMENTS PROFILE QUALITY MODEL

Thus, software requirements profile quality model ($SRPQM$) includes 16 following sets:

$$SRPQM = \begin{Bmatrix} SIRCSRP, SIRCFCSRP, SIRASRP, \\ SIRCFASRP, MSC, ASC, USC, \\ RSSE, RSSEP, RSSEN, SCF, ST, \\ CSRP, CFCSRP, ASRP, CFASRP \end{Bmatrix}$$

The general detailed structure of software requirements profile quality model is represented in Fig. 3. Such structure includes the following elements: characteristics and attributes of software requirements and their classification features; characteristics and attributes of software requirements profile and its classification features; semantic and syntax of software requirements.

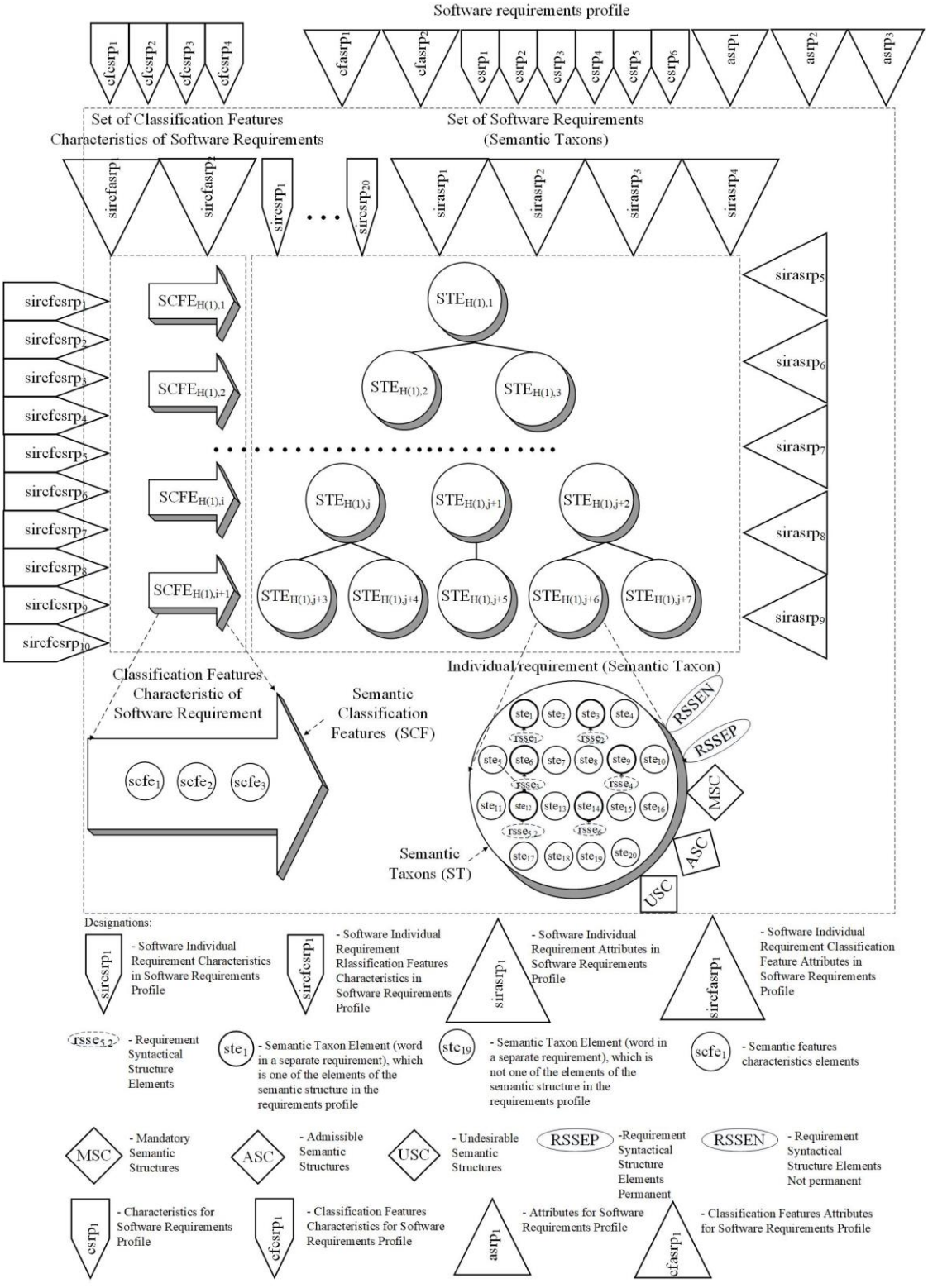

Figure 3 – Structure of software requirements profile quality model

V. CASE STUDY

Software requirements profile quality model is a necessary element for requirements profile assessment. The model contains a nomenclature of its own elements (characteristics, attributes, structure of software requirements profile, semantic and syntax). In accordance with such model, the procedure of software requirements profile quality assessment will be realized in the future.

The suggested approach is used for the development of terms of reference or a draft of a new standard. For example, a model is used for the assessment draft of the new standard "Requirements to computer security of NPP Instrumentation and Control Systems (NPP I&C)" developed by the Ukrainian State Regulatory Body [30]. As a result of the development of software requirements profile quality model, a set of propositions for improving the quality of the standard as a branch of the profile for NPP I&C cybersecurity are implemented. Basic beneficiaries and users of such approach to software requirements profile quality assessment in part of representing of software requirements profile quality model are quality assurance services, quality management system managers, independent auditors and others.

For the development of software requirements profile quality model beneficiaries need to solve the following expert tasks:

– to conduct a preliminary analysis of characteristics for each software requirement and its classification features from software requirements profile;

– to determine values of attributes for each software requirement and values of attributes for its classification features from software requirements profile;

– to make an analysis of semantics for each software requirement from software requirements profile;

– to make an analysis of syntactic structures for each software requirement from software requirements profile;

– to conduct a preliminary analysis of characteristics for software requirements profile and its classification features;

– to determine values for software requirements profile attributes and values of attributes for its classification features.

VI. CONCLUSION

In the article software requirements profile quality model is presented, which includes: characteristics and attributes of software requirements and their classification features; characteristics and attributes of software requirements profile and its classification features; semantics and syntax of software requirements.

The software requirements profile quality model does not allow assessing quality. This model is a result of the first stage in the process of software requirements profile quality assessment. The next stage of this process can be the development of profile-oriented approach to the assessment of software requirements profile quality [31].

References

[1] H. Watts, *The software quality profile*, in: Taz Daughtrey, Fundamental Concepts for the Software Quality Engineer, American Society for Quality (ASQ), 2001, pp. 3-17.

[2] International Standard Systems and Software Engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, ISO/IEC 25010, International Organization for Standardization, International Electrotechnical Commission, 2011, 34 p.

[3] J. Gong, J. Lu, L. Cai, "An induction to the development of software quality model standards," *Proceedings of the Third International Conference on Trustworthy Systems and their Applications (TSA)*, Wuhan, China, September 18-22, 2016, pp. 117-122. https://doi.org/10.1109/TSA.2016.28.

[4] S. K. Dubey, S. Ghosh, A. Rana, "Comparison of software quality models: an analytical approach," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, issue 2, pp. 111-119, 2012.

[5] N. Malhotra, S. Pruthi, "An efficient software quality models for safety and resilience," *International Journal of Recent Technology and Engineering*, vol. 1, issue 3, pp. 66-70, 2012.

[6] S. Faily, J. Lyle, I. Fléchais, A. Simpson, "Usability and security by design: A case study in research and development," *Proceedings of the Workshop on Usable Security*, San Diego, CA, USA, February 8-11, 2015, pp.1-10. https://doi.org/10.14722/usec.2015.23012.

[7] J. Gong, J. Lu, L. Cai, "An induction to the development of software quality model standards," *Proceedings of the 3rd Conference on Trustworthy Systems and their Applications (TSA)*, Wuhan, China, September 18-22, 2016, pp. 117-122.

[8] O. Gordieiev, V. Kharchenko, K. Leontiiev, "Usability, security and safety interaction: Profile and metrics based analysis," *Proceedings of the 13 International Conference on Dependability and Complex Systems DepCoS-RELCOMEX*, Brunow, Poland, July 2-6, 2018, pp. 238-247. https://doi.org/10.1007/978-3-319-91446-6_23.

[9] R. Mirsalari, P. N. Robillard, "Expected software quality profile: A methodology and a case study," *Proceedings of the IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, Canada, October 13-15, 2016, pp. 774-781. https://doi.org/10.1109/IEMCON.2016.7746239.

[10] E. Knauss, C. El Boustani, "Assessing the quality of software requirements specifications," *Proceedings of the 16th IEEE International Requirements Engineering Conference*, Barcelona, Spain, September 8-12 2008, pp. 341-342. https://doi.org/10.1109/RE.2008.29.

[11] A. Toffano Seidel Calazans, R. Avila Paldes, E. Toffano Seidel Masson, I. S. Brito, K. Fialho Rezende, E. Braosi, N. Indayara Pereira, "Software requirements analyst profile: A descriptive study of Brazil and Mexico," *Proceedings of the IEEE 25th International Requirements Engineering Conference*, Lisbon, Portugal, September 4-8, 2017, pp. 204-212. https://doi.org/10.1109/RE.2017.22.

[12] V. Kharchenko, O. Gordieiev, A. Fedoseeva, "Profiling of software requirements for the pharmaceutical enterprise manufacturing execution system," in: R. Bris, J. Majernik, K. Pancerz, E. Zaitseva, *Applications of Computational Intelligence in Biomedical Technology*, Springer, 2016, vol. 606, pp. 67-93. https://doi.org/10.1007/978-3-319-19147-8_4.

[13] G. O'Regan, *Introduction to Software Quality*, Springer Nature, Switzerland, 2014, 354 p.

[14] I. Margarido, J. Faria, R. Vidal, M. Vieira. Classification of Defect Types in Requirements Specifications: Literature Review, Proposal and Assessment. *Proceedings of the 6th Iberian Conference on Information Systems and Technologies (CISTI 2011)*, Chaves, Portugal, June 15-18, 2011, pp. 1-6.

[15] G. O'Regan, *Concise Guide to Software Engineering: From Fundamentals to Application Methods (Undergraduate Topics in Computer Science)*, Springer nature Switzerland, 2014, 356 p.

[16] A. Alshazlya, A. Elfatatry, M. Abougabal. Detecting defects in software requirements specification. *Alexandria Engineering Journal*, 2014, vol. 53 (3). pp. 513-527. https://doi.org/10.1016/j.aej.2014.06.001.

[17] R. Ferenc, P. Hegedus, T. Gyimothy, "Software product quality models," in: T. Mens, A. Serebrenik, A. Cleve, *Evolving Software Systems*, Springer-Verlag Berlin Heidelberg, 2014, pp. 65-100. https://doi.org/10.1007/978-3-642-45398-4_3.

[18] M. A. Levin, T. T. Kalal, J. Rodin, *Improving Product Reliability and Software Quality: Strategies, Tools, Process and Implementation*, Wiley, 2019, 456 p. https://doi.org/10.1002/9781119179429.

[19] P. Ehrenbrink, S. Hillmann, B. Weiss, S. Moller. Psychological reactance in HCI: a method towards improving acceptance of devices and services. *Proceedings of the 28th Australian Conference on Computer-Human Interaction*, Launceston, Australia, 29 November – 2 December, 2016, pp. 478-482. https://doi.org/10.1145/3010915.3010978.

[20] M. Bychok, O. Pohudina, "Evaluation of use of design templates in the software development," *Radioelectronic and Computer Systems*, vol. 1 (97), pp. 101-109, 2021. https://doi.org/10.32620/reks.2021.1.09.

[21] I. Shelechov, N. Barchenko, V. Kalchenko, V. Obodiak, "A hierarchical fuzzy quality assessment of complex security information systems," *Radioelectronic and Computer Systems*, vol. 4 (96), pp. 106-115, 2020.

[22] *International standard Systems and software engineering – Life cycle processes – Requirements engineering, ISO/IEC/IEEE 29148:2011(E)*, International Organization for Standardization, International Electrotechnical Commission, Institute of Electrical and Electronics Engineers, 2011, 95 p.

[23] *International standard Systems and software engineering – Life cycle processes – Requirements engineering. ISO/IEC/IEEE 29148:2018(E)*, International Organization for Standardization, International Electrotechnical Commission, Institute of Electrical and Electronics Engineers, 2018, 104 p.

[24] *International standard Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model. ISO/IEC 25012:2008*, International Organization for Standardization, International Electrotechnical Commission. 2008, 13 p.

[25] A. Harwood, P. Wilkinson, A. Mavin, M. Novak, "Easy approach to requirements syntax (EARS)," *Proceedings of the 17th IEEE International Requirements Engineering Conference*, Atlanta, USA, Aug. 31 – Sept. 4, 2009, vol. 1, pp. 317-322.

[26] T. Tahvonen, E. Uusitalo, "Easy approach to requirements syntax in nuclear power plant safety design," *Proceedings of the 1st International Workshop on Easy Approach to Requirements Syntax (EARS)*, Banff, Canada, August 21, 2018, vol. 1, pp. 1-2. https://doi.org/10.1109/EARS.2018.00006.

[27] N. Urushibara, C. Sasaki, "Integration of two kinds of syntax for requirements description and its future development," *Proceedings of the 1st International Workshop on Easy Approach to Requirements Syntax (EARS)*, Banff, Canada, August 21, 2018, vol. 1, pp. 3-8. https://doi.org/10.1109/EARS.2018.00007.

[28] D. Majumdar, S. Sengupta, A. Kanjilal, A. Kanjilal, S. Bhattacharya, "Adv-EARS: A formal requirements syntax for derivation of use case models," *Proceedings of the International Conference on Advances in Computing and Information Technology (ACITY)*, Chennai, India., July 15-17, 2011, pp. 40-48. https://doi.org/10.1007/978-3-642-22555-0_5.

[29] A. Mavin, P. Wilkinson, "Ten years of EARS," *IEEE Software*, vol. 36, issue 5, pp. 10-14, 2019. https://doi.org/10.1109/MS.2019.2921164.

[30] A. Symonov, O. Klevtsov, S. Trubchaninov, O. Lazurenko, "Computer security of NPP instrumentation and control systems: Justification documents," *Nuclear and Radiation Safety*, vol.4 (84), pp. 73-81, 2019. https://doi.org/10.32918/nrs.2019.4(84).09.

[31] O. Gordieiev, V. Kharchenko, "Profile-oriented assessment of software requirements quality: models, metrics, case study," *International Journal of Computing*, vol. 19, issue 4, pp. 656-665, 2020. https://doi.org/10.47839/ijc.19.4.2001.

OLEKSANDR GORDIEIEV, Dr of Science, Professor of the Software Engineering Department, Lutsk National Technical University *"LNTU". He has a PhD from National Aerospace University "KhAI"* (Kharkiv, Ukraine) and MS from the same university. His scientific interests lays in the area of software quality assessment and assurance.

VYACHESLAV KHARCHENKO, Dr of Science, Professor and Head of the Department of Computer Systems, Networks and Cyber Security National Aerospace University KhAI (Kharkiv, Ukraine) and the Head of STC, RPC Radiy. He has a Engineer Degree in Control Systems from Kharkiv High Rocket Engineering College, PhD Degree in Testing of Digital Systems (Academy of Rocket Troops), DrS Degree in Dependable Computing (Kharkiv Military University). His scientific interests lays in critical and green computing, resilient AI.

Daria Gordieieva, PhD, Associate professor of the Computer Engineering and Cybersecurity Department, Lutsk National Technical *University "LNTU". Her scientific* interests lays in the area of software quality assessment and assurance.