PROFILE-ORIENTED ASSESSMENT OF SOFTWARE REQUIREMENTS QUALITY: MODELS, METRICS, CASE STUDY

Oleksandr Gordieiev [1), Vyacheslav Kharchenko [2)

[1) Cybersecurity Department, Banking University, Andriivska Str., 1, Kyiv, Ukraine, 04070,
alex.gordeyev@gmail.com, http://ubs.edu.ua
[2) Department of Computer Systems, Networks and Cybersecurity, National Aerospace University "KhAI", Ukraine,
61070, Kharkiv, 17, Chkalova Str, v.kharchenko@csn.khai.edu, https://csn.khai.edu/en

Abstract: The article is devoted to profile-oriented approach to assessing the software requirements profile quality. The process of assessment is implemented in accordance with software requirements profile quality model for external quality of software requirements profile and internal quality of software requirements profile. For assessment of quality of each element of such model (characteristics and attributes of software requirements and their classification features; characteristics and attributes of software requirements profile and their classification features; structure of software requirements profile; semantics and syntax of software requirements) taxonomy of metrics and indicators are introduced, which is a part of software requirements profile quality assessment model. All metrics are divided into qualitative and quantitative groups. Additive convolution method is used to go from the multi-vector values of set of connected indicators or metrics to the single scalar value. Received values of metrics and indicators are visualized by radial-metrics diagrams. An example is discussed, which represents the assessment of external quality of software requirements profile for requirements profile from new standard "Requirements to computer security of NPP Instrumentation and Control Systems (NPP I&C)" developed by Ukrainian state regulatory body.

1. INTRODUCTION

1.1 MOTIVATION

The basis of software quality is formed by terms of reference (specification) for software development. The basis of terms of reference is software requirements profile [1], which unites all functional and non-functional requirements in the single logical structure. Thus, software requirements profile quality determines fundamental quality of developed software in general.

The task of formation of qualitative software requirements profile is non-trivial and it has many particularities. To solve it, expert participation is required. For demonstration of non-triviality of such a task, we can represent the following example. The analysis of software quality models, the characteristics of which are fundamental for non-functional requirements in the software requirements profile, allowed us to establish an interaction between following characteristics (and consequently between non-functional requirements) of software quality models: between pairs of characteristics: "security" and "usability" [2], "greenness" and "reliability" [3] and "triplet usability", "security" and "safety" [4], etc. [5]. On the one hand, this interaction between non-functional requirements is constructive when the characteristics complement each other, and on the other, it is destructive when they compete with each other.

Responsibility for assurance of software requirements profile quality usually lies on quality assurance (QA) services, managers of quality management system, independent auditors and

---

This paper has been submitted for the Open Special Issue on Green Mobile Computing and IoT Systems. Assessment, Modeling, Assurance.

experts. Such participants of process will be denoted by general term "expert" in the article. Based on their own experience and the necessary tools, experts, first of all, assess the quality of existing software requirements profile and then, if necessary, work to improve its quality, i.e., the primary task in ensuring the software requirements profile quality is the task of its quality assessment.

Software requirements profile quality assessment is a separate task. Such task is connected with assurance process of software requirements profile quality. When solving such a problem, the following features should be taken into account:

– assessment of software requirements profile quality, as a rule, is not a unified procedure and has some differences for projects in terms of the nomenclature of requirements and the structure of their profile. Therefore, experts cannot be excluded from the process of assessment and their role, to a certain extent, determines the quality the process of assessment and its results;

– requirements of the software requirements profile do not always have the property of orthogonality. The interaction between separate requirements or their groups is often not obvious. Such interconnections are found by experts and can only be automated for big requirements groups, for example, in competition between usability and security;

– processing and analysis of the software requirements profile quality assessment results cannot be fully automated and, again, require the participation of corresponding experts.

1.2. RELATED WORKS ANALYSIS

The preliminary analysis of existing articles, in which questions of software requirements profile quality are reviewed allowed us to separate them into the following groups:

1. Articles, in which questions of assessment at the level of separate requirements are reviewed, that are not connected to a single software requirements profile [6-9]. Approach for assessment of single software requirement and software requirements profile has some differences. For, example, single software requirement has fewer characteristics than software requirements profile. Such situation is explained by emergence principle;

2. Papers, in which only individual elements of software requirements profile are assessed as a whole, are [10-12]. Particularly, only assessment of the semantics of requirements is considered in [13-15], i.e. complex model, which can describe a quality of software requirements profile and a corresponding model, which can assess a quality of such profile with the use of nomenclature of metrics, are absent;

3. Articles, in which authors make attempts to fully automate the process of software requirements quality assessment, are [16-18]. First of all, task must be done before automating the assessment process necessary development of approach to software requirements profile assessment.

Objectives of the paper are:

– to form the taxonomy of metrics and indicators uniting them. Such taxonomy of metrics and indicators is the basis of software requirements profile quality assessment model;

– to represent the mechanism of cast values of multi-vector indicators and metrics to a single scalar;

– to represent the variant of visualization of received results;

– to assess the quality of software requirements profile on the real example.

The article is structured as the follows: section 2 describes terms, which are used in the article, and basic elements of model of assessment of software requirements profile quality; in section 3 taxonomy of metrics and indicators is represented, which is used for assessment of software requirements profile quality directly; in section 4 the variant of visualization of received values as radial-metrics diagrams (RMD) is represented and in section 5 real example of assessment of software requirements profile quality is reviewed.

2. PROFILE-ORIENTED APPROACH TO SOFTWARE REQUIREMENTS QUALITY ASSESSMENT

2.1 SOFTWARE REQUIREMENTS PROFILE QUALITY MODEL

The idea of a profile-oriented approach has appeared during the development of the quality model of an individual software requirement and further was developed after the presentation of the software requirements profile quality model. We enumerate elements of model of software requirements profile quality briefly and introduce terms. The model includes the following elements: 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. We introduce the following terms:

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

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

– "requirement characteristic". It is a set of features, that define distinctive particularities of a requirement;

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

– "requirement attribute". It 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". It is a tool of formal description and combination of elements software requirements profile quality.

Note, that model of software requirements profile quality was developed using some standards [19-21].

2.2 MODEL OF SOFTWARE REQUIREMENTS PROFILE QUALITY ASSESSMENT

Structure of the model of software requirements profile quality assessment consists of the set of elements of software requirements profile quality

model (Fig. 1, upper triangle) and the set of metrics and indicators for assessment the corresponding requirements profile elements (Fig. 1, bottom triangle). The assessment elements of software requirements profile, metrics and indicators are divided into 2 groups (Fig. 1): external quality of software requirements profile and internal quality of software requirements profile. Such separating into two groups is necessary because assessment is connected to each requirements separately (software requirements profile internal quality) and software requirements profile as with general object, which combines all requirements in a single structure (software requirements profile external quality) with additional characteristics, attributes, etc.

The main task of model is the assessment of software requirements profile quality based on the calculation of the set of designated metrics and indicators. Assessing the quality of the entire software requirements profile leads to assessing the elements of software requirements profile quality in accordance with software requirements profile quality model.
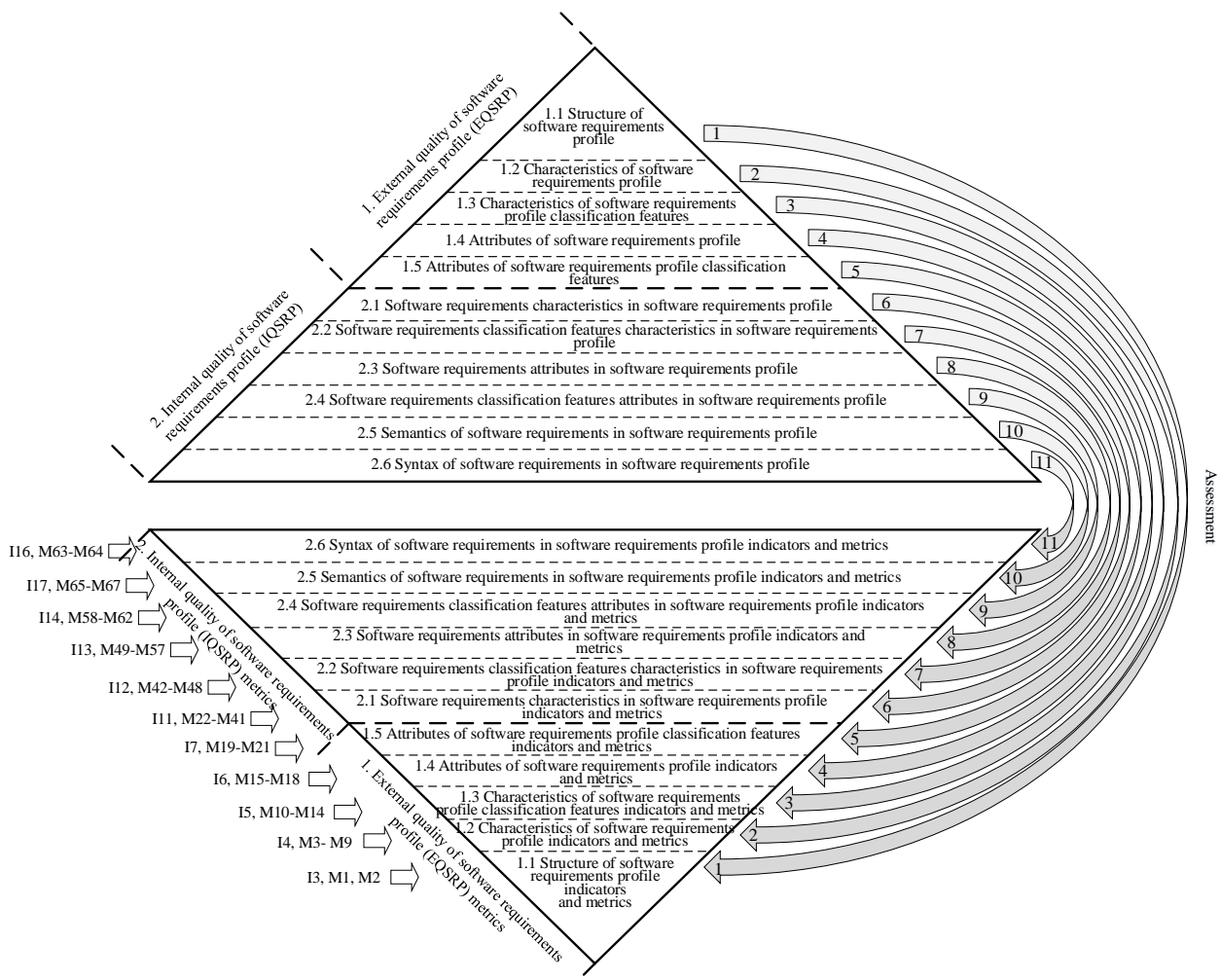
Figure 1 – Structure of a model of software requirements profile quality assessment

To assess the elements of the software requirements profile quality model the following components of the assessment process are needed:

– object of assessment. In this case objects of assessment are the elements of software requirements profile quality model;

– expert, who makes the assessment. As a rule, he assesses the quality of requirements profile from the point of an independent organization, which makes expertise of software in general;

– metrics and primitives for its calculation. Metrics are divided into 2 groups: qualitative and quantitative. Qualitative metrics are calculated on the basis of corresponding techniques. Quantitative metrics are calculated on the basis of formula whose initial data are primitives. Metrics can be calculated either manually or using automated procedures. Each group of metrics has a rating scale - qualitative or quantitative;

– indicators. Such indicators generalize groups of metrics into taxonomy and reduce values of metrics to a single scalar depending on the taxonomy level;

– techniques of the assessment. Techniques of assessment are presented by the procedure for evaluating software requirements profile quality elements, which cannot be estimated using metrics. The choice of techniques for assessing the qualitative metrics is made by an expert based on his experience and practical skills. Necessary to distinguish between expert assessment (manually technique) and automated assessment;

– scale of the assessment, as a rule, corresponds to the type of metrics. Scale of the assessment is divided into 2 groups: quantitative scale of the assessment which corresponds to quantitative metrics with acceptable range of values [0,1], qualitative scale of the assessment which corresponds to qualitative metrics with the following values: corresponds/does not correspond or corresponds/partially corresponds /does not correspond.

3. METRICS AND INDICATORS FOR ASSESSMENT OF SOFTWARE REQUIREMENTS PROFILE QU ALITY

Part of the software requirements profile quality model is presented by metrics and indicators. All metrics are divided into groups and are organized by relevant indicators into a single taxonomy (Fig. 2). At the beginning, let us examine in more detail indicators of taxonomy. In Table 1 such indicators are represented by 3 columns: identifier (ID), abbreviation and full name. To switch from multi-vector values of the set of connected indicators or metrics (Fig. 2) to single scalar value, the additive convolution method is used when calculating all

indicators. Such method consists in "folding" commensurable indicators (metrics) of this level into an indicator higher level of taxonomy (Fig. 2).
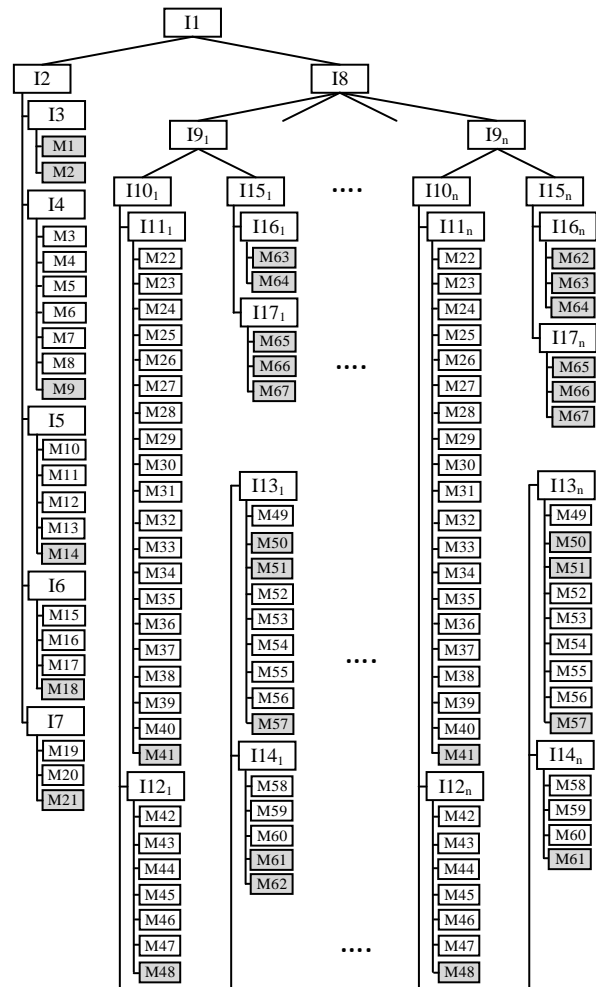

Figure 2 – Taxonomy of metrics and indicators of software requirements profile quality assessment

Herewith for the value of each indicator must be set weight coefficients $k=(k_i, k_{i+1},...,k_n)$, where k – weight coefficient, i – ordinal index of weight coefficient, n – ordinal index of last weight coefficient. Notes, that the sum of all weight coefficients must be equal 1 (1):

$$\sum_{i=0}^{n} k_i = 1, k_i \geq 0 ,\qquad(1)$$

where k – weight coefficient, i – ordinal index of weight coefficient, n – ordinal index of last weight coefficient in one formula. Thus, each indicator will be calculated according to the following formula (2):

$$\text{Im} = \sum_{i=0}^{n} (k_i * l_i) ,\qquad(2)$$

where k – weight coefficient, l – indicator, i – ordinal index of weight coefficient and indicator, n –

ordinal index of last weight coefficient and indicator in one formula, *m* – ordinal number of indicator, for which casculate value. As a rule, weight coefficients are set by expert.

An example of a calculation formula for indicator *I2* is represented:

$$I2 = k_1 * I3 + k_2 * I4 + k_3 * I5 + k_4 * I6 + k_5 * I7, \quad (3)$$

where *I2* – software requirements profile external quality generalized indicator, *I3* – software requirements profile structure quality indicator, *I4* – software requirements profile characteristics quality indicator, *I6* – software requirements profile attributes quality indicator, *I7* – software requirements profile classification features attributes quality indicator, k_1-k_5 – weight coefficients.

Table 1. Indicators

ID	Abbreviation	Full name
1. External quality of software requirements profile (EQSRP) indicators		
I1	*SRPQGI*	Software requirements profile quality generalized indicator
I2	*SRPEQGI*	Software requirements profile external quality generalized indicator
I3	*SRPSQI*	Software requirements profile structure quality indicator
I4	*SRPCQI*	Software requirements profile characteristics quality indicator
I5	*SRPCFCQI*	Software requirements profile classification features characteristics quality indicator
I6	*SRPAQI*	Software requirements profile attributes quality indicator
I7	*SRPCFAQI*	Software requirements profile classification features attributes quality indicator
I8	*SRPIQGI*	Software requirements profile internal quality generalized indicator
2. Internal quality of software requirements profile (IQSRP) indicators		
I9₁	*SRPQIRGI*	Software requirements profile quality individual requirement generalized indicator
I10₁	*SRPEQIRGI*	Software requirements profile external quality individual requirement generalized indicator
I11₁	*SRPCQIRI*	Software requirements profile characteristics quality individual requirement indicator
I12₁	*SRPCFCQIRI*	Software requirements profile classification feature characteristics quality individual requirement
		indicator
I13₁	*SRPAQIRI*	Software requirements profile attributes quality individual requirement indicator
I14₁	*SRPCFAQIRI*	Software requirements profile classification feature attributes quality individual requirement indicator
I15₁	*SRPIQIRGI*	Software requirements profile internal quality individual requirement generalized indicator
I16₁	*SRPSQIRI*	Software requirements profile syntax quality individual requirement indicator
I17₁	*SRPSEQIRI*	Software requirements profile semantics quality individual requirement indicator

The model of software requirements profile quality assessment (Fig. 1) includes 67 metrics (lower level of taxonomy) (Fig. 2): 49 qualitative and 18 quantitative metrics. Quantitative metrics were marked with grey background in Fig. 2. Quantitative metrics were calculated according to formulas which are represented in Table 3. For definition of the value of qualitative metric the expert has to propose a technique that would allow us to obtain its qualitative values. It should be noted that values of metrics can be converted from one scale of assessment to another, for example, from qualitative scale of assessment to quantitative and conversely. Examples of such normalizations are represented in Table 2. As a rule, the correspondences between the values of metrics, which are represented in different assessment scales, are determined by the expert.

Qualitative metrics are not represented in the article, because their descriptions and a description of the techniques corresponding to them is a topic of a separate statement and requires significant descriptive volumes. The article provides a description of quantitative metrics represented as their ID, name, abbreviations, formulas and primitives (Table 3). It should be noted, that ID for metrics in Table 3 corresponds to taxonomy of metrics and to indicators of assessment (Fig. 2). All quantitative metrics (Table 3) are divided into 2 groups: external quality of software requirements profile (EQSRP) indicators and internal quality of software requirements profile (IQSRP) indicators.

Proposed metrics are not mandatory in the profile-oriented assessment, because their nomenclature and quantity are determined by the expert who makes the assessment. The only condition for the use of metrics is the mandatory use of minimum one metric for one element of software requirements profile quality model. Proof of

completeness of nomenclature metrics and indicators is not included in the article, because such task can be the separate research (i.e., a separate article).

Quantitative	Qualitative
First variant	
(0,6-1)	Corresponds
(0,5-0)	Does not correspond
Second variant	
(0,8-1)	Corresponds
(0,4-0,7)	Partially corresponds
(0-0,3)	Does not correspond

4. VISUALIZATION OF RECEIVED RESULTS

In this section radial-metrics diagrams (RMD) are considered in more details and used for visualization of received results (Fig. 3) [22]. In RMD each indicator (metric) is represented by a separate vector, on which the value of the indicator or metric is marked. Value of the indicator (metric) and value of its weight coefficient are indicated near the scale of vector. The value of the indicator is marked on the scale of vector directly and can fall into one of three sectors: does not correspond (unsatisfactory quality – red sector), partially

corresponds (satisfactory quality – yellow sector) or corresponds (high quality – green sector). The intervals of such sectors can be adjusted by an expert with consideration of particularities of the evaluated software requirements profile. Choice of RMD for visualization of received visual information was caused by simplicity in representation and understanding of visual information.
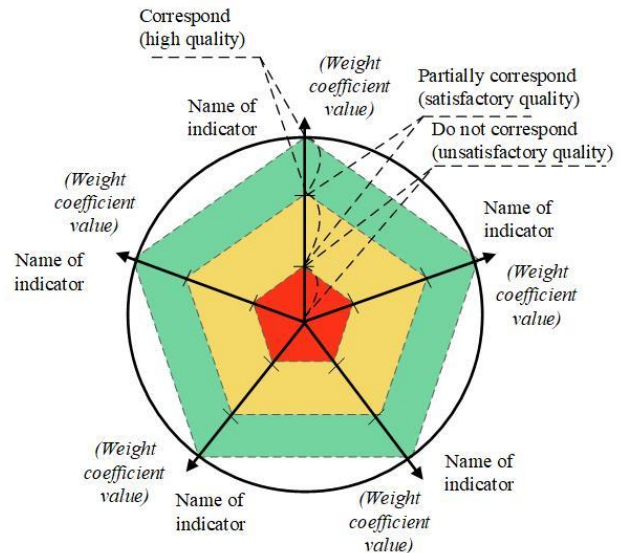

Figure 3 – Overall view of radial-metrics diagram

Table 3. Quantitative metrics

ID	Name (Abbreviation)	Formula	Description of primitives
1. External quality of software requirements profile (EQSRP) metrics			
M1	Software requirements profile taxons number (*SRPTN*)	$SRPTN = \dfrac{SRPTRN}{SRPTRQN}$	*SRPTRN* - software requirements profile taxons real number, *SRPTRQN* - software requirements profile taxons required number
M2	Software requirements profile classification features number (*SRPCFN*)	$SRPCFN = \dfrac{SRPCFRN}{SRPCFRQN}$	*SRPCFRN* – software requirements profile classification features real number, *SRPCFRQN* – software requirements profile classification features required number
M9	Software requirements profile characteristics quality (*SRPCQ*)	$SRPCQ = \dfrac{SRPPCQ}{SRPACQ}$	*SRPPCQ* – software requirements profile performed characteristics quantity, *SRPACQ* – software requirements profile all characteristics quantity
M14	Software requirements profile classification features characteristics quality (*SRPCFCQ*)	$SRPCFCQ = \dfrac{SRPCFPCQ}{SRPCFACQ}$	*SRPCFPCQ* – software requirements profile classification features performed characteristics quantity, *SRPCFACQ* – software requirements profile classification features all characteristics quantity
M18	Software requirements profile attributes quality (*SRPAQ*)	$SRPAQ = \dfrac{SRPPAQ}{SRPAAQ}$	*SRPPAQ* – software requirements profile performed attributes quantity, *SRPAAQ* – software requirements profile all attributes quantity
M21	Software requirements profile classification features attributes quality (*SRPCFAQ*)	$SRPCFAQ = \dfrac{SRPCFPAQ}{SRPCFAAQ}$	*SRPCFPAQ* – software requirements profile classification features performed attributes quantity, *SRPCFAAQ* – software requirements profile classification features all attributes quantity

	2. Internal quality of software requirements profile (IQSRP) metrics		
M41	Software requirements profile software requirement conforming characteristic (*SRPSRCC*)	$SRPSRCC = \dfrac{REQCGSP}{GQRE}$	*REQCGSP* – requirement elements quantity conforming for general spelling pattern, *GQRE* –general quantity of requirement elements
M48	Software requirements profile software requirement characteristics quality (*SRPSRCQ*)	$SRPSRCQ = \dfrac{SRPSRPCQ}{SRPSRACQ}$	*SRPSRPCQ* – software requirements profile software requirement performed characteristics quantity, *SRPSRACQ* – software requirements profile software requirement all characteristics quantity
M50	Software requirements profile classification feature software requirement characteristics quality (*SRPCFSRCQ*)	$SRPCFSRCQ = \dfrac{SRPCFSRPCQ}{SRPCFSRACQ}$	*SRPCFSRPCQ* – software requirements profile classification feature software requirement performed characteristics quantity, *SRPCFSRACQ* – software requirements profile classification feature software requirement all characteristics quantity
M51	Software requirements profile software requirement stakeholder priority attribute (*SRPSRSPA*)	$SRPSRSPA = \dfrac{RSQ}{PSGQ}$	*RSQ* – requirement stakeholder quantity, *PSGQ* – project stakeholder general quantity
M57	Software requirements profile software requirement risks assessment attribute (*SRPSRRAA*)	$SRPSRRAA = \dfrac{RRQ}{RRGQ}$	*RRQ* – requirement risks quantity, *RRGQ* – project stakeholder general quantity
M61	Software requirements profile software requirement attributes quality (*SRPSRAQ*)	$SRPSRAQ = \dfrac{SRPSRPAQ}{SRPSRAAQ}$	*SRPSRPAQ* – software requirements profile software requirement performed attributes quantity, *SRPSRAAQ* – software requirements profile software requirement all attributes quantity
M62	Software requirements profile software requirement classification features attributes quality (*SRPSRCFAQ*)	$SRPSRCFAQ = \dfrac{SRPSRCFPAQ}{SRPSRCFAAQ}$	*SRPSRCFPAQ* – software requirements profile software requirement classification features performed attributes quantity, *SRPSRCFAAQ* – software requirements profile software requirement classification features all attributes quantity
M63	Software requirements profile software requirement syntactical structure permanent elements (*SRPSRSSPE*)	$SRPSRSSPE = \dfrac{RPSSEQ}{2}$	*RPSSEQ* – requirement permanent syntactical structure elements quantity
M64	Software requirements profile software requirement syntactical structure variable elements (*SRPSRSSVE*)	$SRPSRSSAE = \dfrac{RASSEQ}{6}$	*RASSEQ* – requirement all syntactical structure elements quantity
M65	Software requirements profile software requirement mandatory semantic structures (*SRPSRMSS*)	$SRPSRMSS = \dfrac{1}{MSSQ}$	*MSSQ* – mandatory semantic structures quantity
M66	Software requirements profile software requirement admissible semantic structures (*SRPSRASS*)	$SRPSRASS = 1 - \dfrac{1}{1 + \dfrac{1}{ASSQ}}$	*ASSQ* – mandatory semantic structures quantity
M67	Software requirements profile software requirement undesirable semantic structures (*SRPSRUSS*)	$SRPSRUSS = 1 - USSQ$	*USSQ* – undesirable semantic structures quantity

5. CASE STUDY

In full, proposed approach is unified and can be used to assess the quality of any software requirements profile. We considered the possibility of applying the proposed approach to assess the quality of requirements profile of the new standard "Requirements to computer security of NPP Instrumentation and Control Systems (NPP I&C)". It was calculated that in order to assess the quality of the designated profile, consisting of 299 requirements, it is necessary to calculate and determine the values for 27 metrics and indicators of assessment of external quality of software requirements profile and 16744 metrics and assessment indicators of internal quality of software requirements profile.

Thus, to assess the entire profile of requirements for the new standard, 16771 metrics and indicators must be calculated. The calculation of the maximum quantity of metrics and indicators, visualization of the results obtained may qualify for a separate research.

The quantity of metrics that are used in the assessment can be reduced by choosing the necessary metrics.

Thus, it was decided to calculate all values of metrics and indicators to obtain only the total value of indicator *I2* (software requirements profile external quality generalized indicator). Value of *I2* allows us to evaluate external quality of software requirements profile generally and preliminarily calculate all the necessary intermediate indicators that are required to calculate *I2*.

First of all, we define the initial data (values of metrics and weight coefficients) for indicator *I3*: $M1=0,4$; $M2=0,6$; $k_1=0,5$; $k_2=0,5$ and calculate indicator *I3* (4):

$$I3 = k_1 * M1 + k_2 * M2 = \\ = 0,5 * 0,4 + 0,5 * 0,6 = 0,5 \quad (4)$$

Further we define the initial data for indicator *I4*: $M3=0,7$; $M4=0,6$; $M5=0,5$; $M6=0,4$; $M7=0,3$; $M8=0,4$; $M9=0,7$; $k_1=0,2$; $k_2=0,1$; $k_3=0,1$; $k_4=0,1$; $k_5=0,1$; $k_6=0,1$; $k_7=0,3$ and calculate indicator *I4* (5):

$$I4 = k_1 * M3 + k_2 * M4 + k_3 * M5 + \\ + k_4 * M6 + k_5 * M7 + k_6 * M8 + k_7 * M9 = \\ = 0,2 * 0,7 + 0,1 * 0,6 + 0,1 * 0,5 + 0,1 * 0,4 + \\ + 0,1 * 0,3 + 0,1 * 0,4 + 0,3 * 0,7 = 0,57 \quad (5)$$

We define the initial data for indicator *I5*: $M10=0,3$; $M11=0,5$; $M12=0,5$; $M13=0,2$; $M14=0,5$;

$k_1=0,3$; $k_2=0,1$; $k_3=0,1$; $k_4=0,3$; $k_5=0,2$ and calculate indicator *I5* (6):

$$I5 = k_1 * M10 + k_2 * M11 + k_3 * M12 + \\ + k_4 * M13 + k_5 * M14 = 0,3 * 0,3 + 0,1 * 0,5 + \\ + 0,1 * 0,5 + 0,3 * 0,2 + 0,2 * 0,5 = 0,35 \quad (6)$$

We define the initial data for indicator *I6*: $M15=0,6$; $M16=0,5$; $M17=0,4$; $M18=0,4$; $k_1=0,2$; $k_2=0,2$; $k_3=0,3$; $k_4=0,3$ and calculate indicator *I6* (7):

$$I6 = k_1 * M15 + k_2 * M16 + k_3 * M17 + \\ + k_4 * M18 = 0,2 * 0,6 + 0,2 * 0,5 + \\ + 0,3 * 0,4 + 0,3 * 0,4 = 0,46 \quad (7)$$

We define the initial data for indicator *I7*: $M19=0,4$; $M20=0,5$; $M21=0,6$; $k_1=0,4$; $k_2=0,3$; $k_3=0,3$ and calculate indicator *I7* (8):

$$I7 = k_1 * M19 + k_2 * M20 + k_3 * M21 = \\ = 0,4 * 0,4 + 0,3 * 0,5 + 0,3 * 0,6 = 0,49 \quad (8)$$

And finally we define the initial data for indicator *I2*: *k1=0,2; k2=0,2; k3=0,1; k4=0,3; k5=0,2* and calculate indicator *I2*:

$$I2 = k_1 * I3 + k_2 * I4 + k_3 * I5 + \\ + k_4 * I6 + k_5 * I7 = 0,2 * 0,5 + \\ + 0,2 * 0,57 + 0,1 * 0,35 + \\ + 0,3 * 0,46 + 0,2 * 0,49 = 0,485 \quad (9)$$

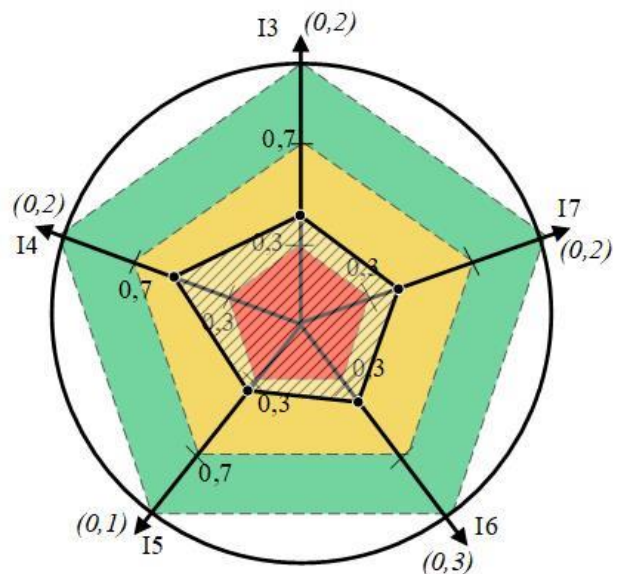Visualization of results is represented as RMD (Fig. 4).

Figure 4 – Visualization of received values as RMD

The received results of calculations indicate the following:

– the values of all indicators are located in the yellow sector of RMD (Fig. 4). This is the evidence of partially correspondence, i.e. satisfactory quality of software requirements profile;

– the value of indicator *I5* is closest to the red sector. This is the evidence of almost unsatisfactory quality of software requirements profile classification features characteristics quality (Fig. 4);

– the value of indicator *I2* is located in the yellow sector of RMD. This is the evidence of satisfactory external quality of software requirements profile.

The article does not describe procedures aimed at improving the quality of software requirements profile. The material of the article is limited only to the description of the approach to assessing the quality software requirements profile. More details of such example is represented in [23]. In the work [23] all needed calculations were represented for assessment of the 5th chapter "Assurance of computer security on stage of development" of the draft of new standard "Requirements to computer security of NPP Instrumentation and Control Systems (NPP I&C)" developed by Ukrainian state regulatory body. Such requirements profile (5th chapter of draft of new standard) includes 26 requirements. Tool for automation of such assessment was represented in [23].

6. CONCLUSION

Profile-oriented approach to assessing the quality of software requirements profile is represented in the article. Such an approach is based on the model for assessing the software requirements profile quality. The model includes taxonomy of metrics and indicators of the software requirements profile quality assessment. The results of calculation of metrics and assessment indicators of quality of software requirements profile classification features characteristics for the real software requirements profile show that a full assessment involves the calculation of a large number of metrics. For assessing them manually, it requires large human and time resources. Consequently, the probability of making mistakes is increased.

In order to automate the process of assessing the software requirements profile quality in the future it is necessary to develop a tool that will support such an assessment process. Automatization has to simplify the assessment process in general, reduce the time for assessment and, therefore, exclude potential mistakes of the expert.

The represented approach can be used for assessment of software cybersecurity and safety requirements in embedded systems as well.

7. 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] K. T. Al-Sarayreh, L. A. Hasan, K. Almakadmeh, "A trade-off model of software requirements for balancing between security and usability issues," *International Review on Computers and Software,* vol. 10, issue 12, pp. 1157-1168, 2016.

[3] G. Lami, F. Fabbrini and M. Fusani, "Software sustainability from a process-centric perspective," *Proceedings of the EuroSPI 2012, Communication in Computer and Information Science CCIS* Springer, vol. 301, 2012, pp. 97-108.

[4] B. D. Payne and W. K. Edwards, "A Brief Introduction to Usable Security," *IEEE Internet Computing*, vol. 12, no. 3, 2008, pp. 13-21.

[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] B. Meyer, J.-M. Bruel, S. Ebersold, F. Galinier, A. Naumchev, "Towards an anatomy of software requirements," *Proceedings of the 51st International Conference, TOOLS 2019*, Innopolis, Russia, October 15-17, 2019, pp. 10-40.

[7] M. Dabbagh, S. P. Lee, R. M. Parizi, "Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches," in Emilio Corchado, Alvaro Herrero, Soft Computing, Springer, vol. 20, issue 11, pp. 4497-4520, 2016. https://doi.org/10.1007/s00500-015-1760-z

[8] Z. A. S. Al-Khanjari, "Proposing a systematic approach to verify software requirements," *Journal of Software Engineering and Applications,* no. 7, pp. 218-224, 2014.

[9] J.-Z. Tang, P.-F. Gu, S.-C. Wang, Y.-N. He, W.-H. Chen, "Nuclear power plants: Innovative technologies for instrumentation and control systems," *Proceedings of the Nuclear Power Plants: Innovative Technologies for Instrumentation and Control Systems (SICPNPP 2017)*, Chengdu, China, August 23-25, 2017, pp. 240-247.

[10] I. Dubielewicz, B. Hnatkowska, Z. Huzar, L. Tuzinkiewicz, "An approach to software quality specification and evaluation (SPoQE),: in Krzysztof Sacha, Software Engineering

Techniques: Design for Quality, Springer, Boston, MA, 2006, vol. 227, pp. 155-166.

[11] J. L. Gassto, "A Quality model for the evaluation of software requirements specifications," *Proceedings of the 2th International Conference on Information Systems Development*, Melbourne, Australia, August 29-31, 2003, pp. 281-291.

[12] E. Nazaruka, J. Osis, "The formal reference model for software requirements," *Proceedings of the 13th International Conference, ENASE 2018,* Funchal, Madeira, Portugal, March 23-24, 2018, pp. 352-372.

[13] M. Saeki, "Semantic requirements engineering," in: S. Nurcan, C. Salinesi, C. Souveyet, J. Ralyte, Intentional Perspectives on Information Systems Engineering, Springer, Berlin, Heidelberg, 2010, pp. 67-82.

[14] Y. Yang, F. Xia, W. Zhang, X. Xiao, Y. Li, X. Li, "Towards semantic requirement engineering," *Proceedings of the IEEE International Workshop on Semantic Computing and Systems,* Huangshan, China, July 14-15, 2008, pp. 67-71.

[15] T. Riechert, K. Lauenroth, J. Lehmann, S. Auer, "Towards semantic based requirements engineering," *Proceedings of the 7th international conference on knowledge management (I-KNOW)*, Graz, Austria, September 5-7, 2007, pp. 1-8.

[16] A. Mishra, A. Awal, J. Elijah, A. Abdul, U. M, Gana, I. Rabiu, "Automation of requirement analysis in software engineering," *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 5, issue 5, pp. 1173-1188, 2017.

[17] M. A. Jubaira, S. A. Mostafaa, A. Mustapha, H. Amana, M. H. Hassana, "Fully automated quality assessment metrics for software requirement specifications," *Proceedings of the Second International Conference on Trustworthy Systems and Their Applications*, Hualien, Taiwan, July 08-09, 2015, pp. 177-187.

[18] M. Kamalrudin, J. Hosking, J. Grundy, "MaramaAI: Automated and visual approach for inconsistency checking of requirements," *Proceedings of the 18th IEEE International Requirements Engineering Conference,* Sydney, NSW, Australia, September 27 - October 1, 2010, pp. 393-394.

[19] 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.

[20] 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.

[21] 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.

[22] A. Andrashov, A. Gordeyev, V. Kharchenko, V. Sklyar, "The static analysis of a program code procedure based on metrics profiling," *Electronic and Computer Systems*, no. 8(27), pp. 184-188, 2007.

[23] O. Gordieiev, D. Gordieieva, A. Tryfonov, "Method and tool for support of software requirements profile quality assessment," *Proceedings of the 9 International Conference Dependable Systems, Services and Technologies (DESSERT 2020)*, Kyiv, Ukraine, May 24-27, 2020, pp. 72-79.

Oleksandr Gordieiev, *PhD, Associate professor, is the Head of the Cybersecurity Department at the Banking University (Kyiv, Ukraine). He has a PhD from National Aerospace University "KhAI" (Kharkiv, Ukraine) and a MS from the same university.*
His scientific interests lay in area of the software quality assessment and assurance.

Vyacheslav Kharchenko, *Dr. of Science, Professor, is the Head of the Department of Computer Systems, Networks and Cyber Security at the 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 in Military Academy of Rocket Troops, and DrS Degree in Dependable Systems from Kharkiv Military University. His scientific interests lay in area of the critical and green computing.*