https://orcid.org/0000-0001-7756-0004
https://orcid.org/0009-0000-4803-6393
https://orcid.org/0009-0008-2309-7569

UDC 519.718

O.V. FEDUKHIN[*], M.M. REDKOVSKA[*], O.O. SKRYPNIKOVA[*]

ATTRIBUTE MODEL OF SOFTWARE DEPENDABILITY ASSESSMENT AND ITS PRACTICAL APPLICATION

[*]Institute of Mathematical Machines and Systems Problems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine

Анотація. *Стаття присвячена питанню використання атрибутивної моделі гарантоздатності (АМГ) для кількісної оцінки рівня гарантоздатності програмного забезпечення (ПЗ). Побудова АМГ ПЗ є одним із найважливіших етапів оцінки гарантоздатності і дозволяє визначити необхідність використання тих чи інших характеристик (атрибутів) програмного продукту. Від повноти та адекватності застосовуваної системи характеристик залежить достовірність одержуваної оцінки гарантоздатності. Розглянуто два приклади програмного забезпечення (ПЗ) медичного призначення. Від лабораторної інформаційної системи (1-ЛІС), яка автоматизує роботу медичної лабораторії, до критичної медичної системи (2-МС) контролю життєдіяльності людини після реанімації, яка включає комплексне рішення, що дозволяє оптимізувати процес лікування пацієнта. Проведено порівняння узагальнених характеритик гарантоздатності. Атрибутивна модель гарантоздатного ПЗ (АМГ ПЗ) зроблена і розширена спеціальними атрибутами і метриками відповідно до вимог ДСТУ ISO/IEC 25051:2016 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Вимоги до якості готового для застосування програмного продукту (RUSP) та інструкції щодо його тестування (ISO/IEC 25051:2014, IDT). У статті використовується математичне подання АМГ, яке призначене для обчислення рівня гарантоздатності ПЗ. Використовується функціонал* $G_{АМГ}$*, складовими якого є експертні нормовані значення кількісних оцінок атрибутів та метрик із відповідними коефіцієнтами ваги та коефіцієнтами впливу. З метою оцінки роботоздатності атрибутивної моделі гарантоздатності ПЗ було проаналізовано два її варіанти — з урахуванням вагових коефіцієнтів критеріїв оцінки метрик і без них. Крім того, атрибутивна модель гарантоздатності ПЗ була апробована на двох типах програмного забезпечення — з низькими вимогами до гарантоздатності (перший пакет ПЗ) і високими вимогами (другий пакет критичного ПЗ).*
Ключові слова: *атрибутивна модель, гарантоздатність, атрибути, метрики, критерії, рівень гарантоздатності ПЗ.*

Abstract. *The article is devoted to the issue of using the attribute model of dependability (AMD) to quantify the level of software dependability. Building an AMD of software is one of the most important stages of reliability assessment that allows for determining the need to use certain characteristics (attributes) of a software product. The completeness and adequacy of the used system of characteristics determine the reliability of the obtained reliability assessment. Two examples of a medical software are considered in the paper. From the laboratory information system (1-LIS), which automates the work of a medical laboratory, to the critical medical system (2-MS) for monitoring human life after resuscitation, which includes a comprehensive solution that allows for optimizing the patient's treatment process. A comparison of the generalized characteristics of assurance was made. The attribute model of dependable software AMD was created and extended with special attributes and metrics in accordance with the requirements of DSTU ISO/IEC 25051:2016 Engineering of systems and software, requirements for the quality of systems and*

ISSN 1028-9763. Математичні машини і системи. 2024. № 3-4

software and its evaluation (SQuaRE), and quality requirements for a ready-to-use software product (RUSP) and instructions for its testing (ISO/IEC 25051:2014, IDT). The article uses the mathematical representation of the AMD, which is intended to calculate the level of software dependability. The functional G_{AMD} is used, the components of which are the expert normalized values of quantitative assessments of attributes and metrics with the corresponding weighting factors and influence factors. In order to evaluate the performance of the attribute model of software reliability, two variants of the model were analyzed, with and without weighting the metrics evaluation criteria. In addition, the attribute model of software reliability was tested on two types of software — with low reliability requirements (the first software package) and high requirements (the second critical software package).
Keywords: *attribute model, dependability, attributes, metrics, evaluation criteria, level of software dependability.*

DOI: 10.34121/1028-9763-2024-3-4-109-123

1. Introduction

In modern practice, the concept of «attribute model of software dependability» (hereinafter referred to as AMD) is used to describe and assess the dependability of software, which is the basis for both the formal description of dependability characteristics and their relations and for assessing the dependability of software. A software dependability model is a structured set of interrelated characteristics and relationships between them. The structure of the software AMD is described by a hierarchy, the elements of which are a set of characteristics and subordination relations between them.

To this day, the approaches and methods for building a software AMD are based on the formation of a hierarchical structure of characteristics. At the top level, there are the characteristics of reliability, detailed by indicators of lower levels, until the decomposition leads to measurable indicators. The differences lie in the proposed number of hierarchy levels (two or more), as well as in the characteristics of the top level of the hierarchy.

The expediency of the hierarchical structure of the software AMD is explained, firstly, by the fact that the multilevel structure of reliability indicators provides a systematic description of the requirements for a software tool, allowing stakeholders to set the desired properties (characteristics) of a software product. Secondly, most of basic indicators (attributes, inherent essential properties of an object) of assurance, such as functional compliance, reliability, safety, etc., cannot be directly measured and evaluated. In order to obtain estimates of these indicators, they can be represented by a set of narrower lower-level characteristics (estimated indicators), which can also be disaggregated. Such disaggregation is carried out as long as the characteristics of the lower level of the hierarchy are easy to assess and provide objective quantitative estimates.

Based on the results of a comparative analysis of a modern software AMD, the article shows the feasibility of building a basic dependability model for critical software based on the standard set of requirements of ISO/IEC 25051. However, to be applied to specific types of software, it should be adapted to meet the specific requirements by selecting relevant attributes, metrics, and criteria, as well as possibly adding additional dependability indicators.

The aim of the paper is to apply a methodological approach to quantifying the level of dependability of computer systems based on the attribute model of dependability in vector and scalar forms of representation [1], the assessment should be carried out in accordance with the requirements of [2] using a universal approach to assessing software attributes and metrics, similar to [3].

2. Requirements for building a software AMD

The construction of the software AMD is carried out through a detailed and consistent description from top to bottom of the multilevel structure of indicators, from the characteristics of the upper level of the hierarchy to the evaluation elements (single indicators). At the same time, the

evaluation element should provide a direct determination of the presence of a particular property of the software.

The assessment of the achieved dependability indicators goes in the opposite direction: from the assessment of individual assessment elements to the aggregate assessment of the higher dependability indicators of the software AMD.

Today, there is no generally accepted methodology for building a software AMD that allows reducing the attributes of dependability to a final set of quantitative assessment indicators, the values of which would be easily and objectively assessed.

At the same time, the practice has developed general requirements and recommendations for the formation of a system of software dependability characteristics. These requirements can be formulated as follows:

• it is recommended to build the AMD of software on the basis of modern international standards regulating the indicators of software dependability, taking into account the functional purpose, specifics, and requirements of the field of application;

• the system of software dependability characteristics should be formed on the basis of the standard characteristics of the upper level of the hierarchy, taking into account the requirements of all stakeholders;

• it is advisable to design a generalized basic AMD for several groups (classes) of software with the maximum possible list of dependability indicators and, in each case, based on the basic model, build an AMD for a particular group or specific software, expanding or narrowing the range of basic dependability indicators;

• the system of software dependability characteristics should describe all the main properties and features of the software of a given class and have the possibility of further refinement and detailing;

• the assessment characteristics of dependability should be formed on the basis of the principle of understandability and measurability of values;

• large groups of software dependability characteristics should be divided into logically related subgroups;

• each dependability characteristic should describe the importance of the software tool in this class;

• the characteristics of dependability should not overlap or duplicate each other;

• single indicators of dependability expressed in physical units should either be converted to relative units that lie in the range from 0 to 1 (normalized) or assessed by experts in the same range.

3. The main part of the study

A comprehensive assessment of the level of software dependability is carried out according to the AMD of software in the form of a linear functional containing estimates of all model attributes [3]. As a mathematical model of the AMD, the following functionality is used to calculate the level of system dependability:

$$G_{AMD} = \sum_{i=1}^{n} B_i A_i , \tag{1}$$

where n is the number of AMD attributes, B_i is the coefficient of influence of the i-th attribute, and A_i is the quantitative assessment of the i-th attribute.

The sum of the influence coefficients of all attributes is equal to the number of attributes:

$$\sum_{i=1}^{n} B_i = n . \tag{2}$$

Each attribute A_i corresponds to its own set of metrics M_{ij} with weighting coefficients β_{ij}:

$$A_i = \sum_{j=1}^{m_i} \beta_{ij} M_{ij} \cdot \tag{3}$$

The sum of the weights of the metrics in a particular attribute is equal to

$$\sum_{j=1}^{m_i} \beta_{ij} = 1. \tag{4}$$

Each metric corresponds to a set of criteria K_{jk} which are used to evaluate the software metrics. K_{jk} is assumed to be equal to the expert value $\hat{E}_{å}$. When several expert assessments of sub-criteria are used to estimate K_{jk}, the score $\hat{E}_{å}$ is calculated using formula (5).

Note 1. The overall expert assessment of the criterion is formed from the average score of the sub-criteria k_j [3] when it is possible:

$$K_{å} = \frac{\sum_{j=1}^{q} k_j}{q}, \tag{5}$$

where q is the number of sub-criteria.

The metrics are calculated by the following formulas:

$$M_{ij} = \sum_{k=1}^{p} \nu_{jk} K_{jk}, \tag{6}$$

or

$$M_{ij} = \sum_{k=1}^{p} K_{jk}, \tag{7}$$

where $k = 1 \div p$ is the sequence number of the k-th criterion in the j-th metric, p is the number of criteria in the j-th metric, and ν_{jk} is the weighting factor of the k-th criterion in the j-th metric.

After calculating the metric scores, each attribute is evaluated, and the general AMD formula looks like this:

$$G_{AMD} = \sum_{i=1}^{n} B_i \sum_{j=1}^{m_i} \beta_{ij} M_{ij}, \tag{8}$$

$$G_{AMD} = \sum_{i=1}^{n} B_i \sum_{j=1}^{m_i} \beta_{ij} \sum_{k=1}^{p} \nu_{jk} K_{jk}, \tag{9}$$

$$G_{AMD} = \sum_{i=1}^{n} B_i \sum_{j=1}^{m_i} \beta_{ij} \sum_{k=1}^{p} K_{jk}. \tag{10}$$

3. Examples of the software AMD application

Let's consider the options for using the software AMD on the examples of software from two medical systems for automating the activities of an institution.

Example 1. Quantitative assessment of the level of dependability of a laboratory information system (LIS) that automates the work of medical laboratories. The LIS can be used to register a client's (patient's) visit and order with automatic labeling.

An example of calculating attributes and metrics, taking into account (6).

Table 1 ─ The main metrics of the attribute *Performance level*

Attribute K_i^A	Evaluation attribute	Metric name M_{ij}	Metric weight factor V_{ij}^M	Metric evaluation M_{ij}	Criterion weight factor v_{jk}^K	Overall evaluation of the criterion $K_{jk} = K_{å}$ or $K_{jk} = K_{å} = \frac{\sum_{j=1}^{q} k_j}{q}$	Name of the criterion/sub-criterion k_j	Expert evaluation of the criterion K_e
Performance level	0.3· 0.76 + 0.4· 0.95 + 0.3· 0.85 = **0.863**	Temporal characte-ristics	0.3	0.6·0.8 + 0.4·0.7 = 0.76	0.6	0.8	Compliance with response time requirements	0.8
					0.4	0.7	Compliance with processing time requirements	0.7
		Use of resources	0.4	0.5·0.9 + 0.5·1= 0.95	0.5	0.9	Volume consumption requirements for performing functions	0.9
					0.5	1	Types of resources for performing functions	1
		Potential opportu-nities	0.3	1·0.85= 0.85	1	(0.8 + 0.8 +0.9+ 0.85 + 0.9)/5 = 0.85	Compliance with the requirements of the limit parameters of the product or system	0.85
							1. The number of	0.8

Continuation of Table 1

								stored items	
								2. The number of users working in parallel	0.8
								3. Channel capacity	0.9
								4. Transaction throughput	0.85
								5. Database size	0.9

The final data of attribute and metric calculations, taking into account (6).

Table 2 — Attribute *Functional suitability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Functional completeness	0.66	0.5	0.33
2	Functional correctness	0.8	0.3	0.24
3	Functional expediency	0.7	0.2	0.14

$$A_{FS} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.71.$$

Table 3 — Attribute *Performance efficiency*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Temporal characteristics	0.76	0.3	0.228
2	Use of resources	0.95	0.4	0.38
3	Potential opportunities	0.85	0.3	0.255

$$A_{PE} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.863.$$

Table 4 — Attribute *Compatibility*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Coexistence	0.76	0.4	0.304
2	Functional compatibility	0.95	0.6	0.57

$$A_{C} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.874.$$

Table 5 — Attribute *Usability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Determination of eligibility	0.96	0.2	0.192
2	Researchability	0.75	0.15	0.1125
3	Controllability	0.8	0.15	0.12
4	Security	0.8	0.2	0.16
5	User interface aesthetics	0.8	0.2	0.16
6	Accessibility	0.8	0.1	0.08

$$A_U = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.8245.$$

Table 6 — Attribute *Reliability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Completeness	0.9	0.3	0.27
2	Readiness	0.9	0.3	0.27
3	Fault tolerance	0.9	0.2	0.18
4	Renewability	0.8	0.2	0.16

$$A_R = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.88.$$

Table 7 — Attribute *Security*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Privacy	0.7	0.2	0.14
2	Integrity	0.9	0.2	0.18
3	Genuineness	0.9	0.2	0.18
4	Traceability	0.8	0.2	0.16
5	Authenticity	0.9	0.2	0.18

$$A_S = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.84.$$

Table 8 — Attribute *Maintainability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Modularity	0.85	0.3	0.255
2	Reusability	1	0.1	0.1
3	Analysability	0.84	0.2	0.168
4	Modifiability	0.9	0.3	0.27
5	Testability	0.8	0.1	0.08

$$A_M = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.873.$$

Table 9 — Attribute *Portability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Adaptability	0.9	0.4	0.36
2	Installability	0.7	0.4	0.28
3	Interchangeability	0.8	0.2	0.16

$$A_P = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.8 .$$

Table 10 — Attribute *Effectiveness*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Accuracy and completeness of goal achievement	0.9	1	0.9

$$A_E = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.9 .$$

Table 11 — Attribute *Performance*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Relationship between accuracy and completeness of goal achievement	0.9	1	0.9

$$A_{Per} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.9 .$$

Table 12 — Attribute *Usefulness*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Fullness	0.8	0.3	0.24
2	Trust	0.7	0.3	0.21
3	User satisfaction	0.8	0.2	0.16
4	Comfort	0.7	0.2	0.14

$$A_{Uf} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0{,}75 .$$

Table 13 — Attribute *Absence of risks*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Mitigating the negative consequences of the economic situation	0.7	0.4	0.28
2	Mitigating the negative health and safety impacts	0.7	0.3	0.21

Continuation of Table 13

3	Mitigating the negative consequences of environmental risk	0.8	0.3	0.24

$$A_{AR} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.73.$$

Table 14 — Attribute *Context coverage*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij} M_{ij}$
1	Context completeness	0.8	0.6	0.48
2	Flexibility	0.7	0.4	0.28

$$A_{CC} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.76.$$

Table 15 — Attribute *Redundancy*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij} M_{ij}$
1	Temporary redundancy	0.9	0.3	0.27
2	Information redundancy	0.8	0.3	0.24
3	Structural redundancy (multi-version) with control coincidence (verification) of results	0.9	0.4	0.36

$$A_R = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.87.$$

Table 16 — Attribute *Self-control*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij} M_{ij}$
1	Self-recovery	0.9	0.2	0.18
2	Self-verification	0.7	0.1	0.07
3	Preventing errors	0.8	0.2	0.16
4	Error tolerance	0.9	0.2	0.18
5	Efficiency	0.85	0.1	0.085
6	Self-adjustment	0.9	0.2	0.18

$$A_{SC} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.855.$$

Note 2. Let's assume that the influence coefficients of the i-th attribute B_i are the following: functional suitability B_1=2; performance efficiency B_2=1; compatibility B_3=0.5; usability B_4=2; reliability B_5=1; security B_6=1; maintainability B_7=1; portability B_8=1; effectiveness B_9=1; performance B_{10}=1; usefulness B_{11}=1; absence of risks B_{12}=1; context coverage B_{13}=0.5; redundancy B_{14}=0.5; self-control B_{15}=0.5.

And then calculate the generalized characteristic of dependability (9):

$$G_{AMD} = \sum_{i=1}^{n} B_i \sum_{j=1}^{m_i} \beta_{ij} M_{ij} = B_1 A_{FS} + B_2 A_{PE} + B_3 A_C + B_4 A_U + B_5 A_R + B_6 A_S + B_7 A_M + B_8 A_P + B_9 A_E + B_{10} A_{Per} +$$

$$+ B_{11} A_{Uf} + B_{12} A_{AR} + B_{13} A_{CC} + B_{14} A_R + B_{15} A_{SC} =$$

=2·0.71+1·0.863+0,5·0.874+2·0.8245+1·0.88+1·0.84+1·0.873+1·0.8+1·0.9+1·0.9+1·0.75+ 1·0.73+0.5·0.76+0.5·0.87+0.5·0.855=**12.2845.**

Below, there is an example of the calculation of attributes and metrics taking into account (7) in accordance with the new AMD form without using the weighting factors of the evaluation criteria.

Table 17 — The main metrics of the attribute *Performance efficiency*

Attribute K_i^A	Attribute evaluation	Name of the metric M_{ij}	Metric weighting factor V_{ij}^M	Evaluation of the metric M_{ij}	Name of the criterion/sub-criterion	Criterion evaluation K_e
Performance efficiency	0.3* 1.5+ 0.4* 1.9+ 0.3* 0.85= **1.465**	Temporal characte-ristics	0.3	0.8+0.7= 1.5	Compliance with the response time requirements	0.8
					Compliance with the processing time requirements	0.7
		Use of resources	0.4	0.9+1=1.9	Volume consumption requirements for performing functions	0,9
					Types of resources for performing functions	1
		Potential opportu-nities	0.3	0.85	Compliance with the requirements of the boundary parameters of the product or system	0.85

Let's calculate the generalized characteristic of dependability (10):

$$G_{AMD} = \sum_{i=1}^{n} B_i \sum_{j=1}^{m_i} \beta_{ij} M_{ij} = B_1 A_{FS} + B_2 A_{PE} + B_3 A_C + B_4 A_U + B_5 A_R + B_6 A_S + B_7 A_M + B_8 A_P + B_9 A_E + B_{10} A_{Per} +$$

$$+ B_{11} A_{Uf} + B_{12} A_{AR} + B_{13} A_{CC} + B_{14} A_R + B_{15} A_{SC} =$$

2·1.42+1·1.465+0.5·1.74+2·1.125+1·1.33+1·0.84+1·1.64+1·1.44+1·0.9+1·0.9+1·0.99+1·1.01 +0.5·1.52+0.5·0.87+0.5·1.89=**18.615.**

Example 2. Quantitative assessment of the level of dependability of the medical system (DMS) for monitoring human vital signs in the intensive care unit (including a comprehensive solution that allows for the diagnosis and treatment process to be carried out and optimized).

The final data of the attributes and metrics calculations, taking into account (6):

Table 18 — Attribute *Functional suitability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Functional completeness	0.96	0.4	0.384
2	Functional correctness	0.9	0.3	0.27
3	Functional expediency	0.9	0.3	0.27

$$A_{FS} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.924 .$$

Table 19 — Attribute *Performance efficiency*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Temporal characteristics	0.94	0.3	0.282
2	Use of resources	0.75	0.2	0.15
3	Potential opportunities	1	0.5	0.5

$$A_{PE} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.932 .$$

Table 20 — Attribute *Compatibility*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Coexistence	0.76	0.6	0.456
2	Functional compatibility	0.95	0.4	0.38

$$A_{C} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.836 .$$

Table 21 — Attribute *Usability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Determination of eligibility	0.96	0.2	0.192
2	Researchability	0.9	0.15	0.135
3	Controllability	0.8	0,15	0.12
4	Security	0.8	0.2	0.16
5	User interface aesthetics	0.8	0.2	0.16
6	Accessibility	0.8	0.1	0.08

$$A_{U} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.847 .$$

Table 22 — Attribute *Reliability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Completeness	1	0.3	0.3
2	Readiness	1	0.3	0.3
3	Fault tolerance	1	0.3	0.3
4	Renewability	1	0.1	0.1

$$A_R = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 1.$$

Table 23 — Attribute *Security*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij} M_{ij}$
1	Privacy	1	0.2	0.2
2	Integrity	1	0.2	0.2
3	Genuineness	1	0.2	0.2
4	Traceability	1	0.2	0.2
5	Authenticity	0.9	0.2	0.18

$$A_S = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.98.$$

Table 24 — Attribute *Maintainability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij} M_{ij}$
1	Modularity	0.85	0.3	0.255
2	Reusability	1	0.1	0.1
3	Analysability	0.84	0.2	0.168
4	Modifiability	0.9	0.3	0.27
5	Testability	0.8	0.1	0.08

$$A_M = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.873.$$

Table 25 — Attribute *Portability*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij} M_{ij}$
1	Adaptability	0.9	0.4	0.36
2	Installability	0.7	0.4	0.28
3	Interchangeability	0.8	0.2	0.16

$$A_P = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.8.$$

Table 26 — Attribute *Effectiveness*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij} M_{ij}$
1	Accuracy and completeness of goal achievement	0.9	1	0.9

$$A_E = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.9.$$

Table 27 — Attribute *Performance*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Relationship between accuracy and completeness of goal achievement	0.9	1	0.9

$$A_{Per} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.9 .$$

Table 28 — Attribute *Usefulness*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Fullness	0.8	0.3	0.24
2	Trust	1	0.3	0.3
3	User satisfaction	0.8	0.2	0.16
4	Comfort	1	0.2	0.2

$$A_{Uf} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.9 .$$

Table 29 — Attribute *Absence of risks*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Mitigating the negative consequences of the economic situation	1	0.4	0.4
2	Mitigating the negative health and safety impacts	1	0.3	0.3
3	Mitigating the negative consequences of environmental risk	0.8	0.3	0.24

$$A_{AR} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.94 .$$

Table 30 — Attribute C*ontext coverage*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Context completeness	0.8	0.6	0.48
2	Flexibility	1	0.4	0.4

$$A_{CC} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.88 .$$

Table 31 — Attribute *Redundancy*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij}M_{ij}$
1	Temporary redundancy	0.9	0.3	0.27

Continuation of Table 31

2	Information redundancy	0.8	0.3	0.24
3	Structural redundancy (multi-version) with control coincidence (verification) of results	0.9	0.4	0.36

$$A_R = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.87 .$$

Table 32 — Attribute *Self-control*

№	Metrics	Evaluation of the metric M_{ij}	Weight β_{ij}	$\beta_{ij} M_{ij}$
1	Self-recovery	0.9	0.2	0.18
2	Self-verification	0.9	0.1	0.09
3	Preventing errors	0.87	0.2	0.174
4	Error tolerance	0.9	0.2	0.18
5	Efficiency	0.85	0.1	0.085
6	Self-adjustment	0.9	0.2	0.18

$$A_{SC} = \sum_{j-1}^{m_i} \beta_{ij} M_{ij} = 0.889 .$$

Note 3. Let's assume that the influence coefficients of the i-th attribute B_i are the following: functional suitability $B_1 = 1$; performance efficiency $B_2 = 0.5$; compatibility $B_3 = 0.5$; usability $B_4 = 0.5$; reliability $B_5 = 3$; security $B_6 = 3$; maintainability $B_7 = 0.5$; portability $B_8 = 0.5$; effectiveness $B_9 = 1$; performance $B_{10} = 1$; usefulness $B_{11} = 0.5$; absence of risks $B_{12} = 1$; context coverage $B_{13} = 0.5$; redundancy $B_{14} = 0.5$; self-control $B_{15} = 1$. Note that in this example, the influence factors for the reliability and security attributes are increased.

Let's calculate the generalized characteristic of dependability (9):

$$G_{AMD} = \sum_{i=1}^{n} B_i \sum_{j=1}^{m_i} \beta_{ij} M_{ij} = B_1 A_{FS} + B_2 A_{PE} + B_3 A_C + B_4 A_U + B_5 A_R + B_6 A_S + B_7 A_M + B_8 A_P + B_9 A_E + B_{10} A_{Per} +$$

$$+ B_{11} A_{Uf} + B_{12} A_{AR} + B_{13} A_{CC} + B_{14} A_R + B_{15} A_{SC} =$$

$$= 1 \cdot 0.924 + 0.5 \cdot 0.932 + 0.5 \cdot 0.836 + 0.5 \cdot 0.847 + 3 \cdot 1 + 3 \cdot 0.98 + 0.5 \cdot 0.873 + 0.5 \cdot 0.8 + 1 \cdot 0.9 + 1 \cdot 0.9 + 0.5 \cdot 0.9$$
$$+ 1 \cdot 0.94 + 0.5 \cdot 0.888 + 0.5 \cdot 0.87 + 1 \cdot 0.889 = \mathbf{13.962.}$$

To evaluate the model's response in the same examples, we eliminate the weighting coefficients on the criteria variables (7).

Generalized characteristic of dependability (10):

$$G_{AMD} = \sum_{i=1}^{n} B_i \sum_{j=1}^{m_i} \beta_{ij} M_{ij} = B_1 A_{FS} + B_2 A_{PE} + B_3 A_C + B_4 A_U + B_5 A_R + B_6 A_S + B_7 A_M + B_8 A_P + B_9 A_E + B_{10} A_{Per} +$$

$$+ B_{11} A_{Uf} + B_{12} A_{AR} + B_{13} A_{CC} + B_{14} A_R + B_{15} A_{SC} =$$

$$= 1 \cdot 1.93 + 0.5 \cdot 1.37 + 0.5 \cdot 1.66 + 0.5 \cdot 1.17 + 3 \cdot 1.6 + 3 \cdot 0.98 + 0.5 \cdot 1.64 + 0.5 \cdot 1.44 + 1 \cdot 0.9 + 1 \cdot 0.9 + 0.5 \cdot 1.14$$
$$+ 1 \cdot 1.34 + 0.5 \cdot 1.76 + 0.5 \cdot 0.87 + 1 \cdot 1.32 = \mathbf{19.655.}$$

5. Conclusion

Building a software AMD is one of the most important stages of software dependability assessment because it allows for determining the need to use certain characteristics (attributes) of a software product when assessing software dependability. The assessment reliability depends on the completeness and adequacy of the used system of characteristics.

The choice of AMD is influenced by many different factors, whether the software is general-purpose or mission-critical. The main features of critical software that should be taken into account when building an AMD are the following:

• the increased requirements for reliability and functional safety;

• the need to guarantee that critical software provides the specified quality, safety, and ability to withstand system disruptions, failures, and errors of various types;

• critical software must have such important properties as reliability, fault tolerance, recoverability, and built-in diagnostic and testing functions;

• several different models can be used in parallel for a comprehensive assessment of the reliability of critical software.

It is clear that the implementation of dependability requirements leads to an overall increase in the cost of software, so the task of optimizing the projected level of dependability and the cost of software is quite relevant and of great social and economic importance.

In order to assess the software AMD, two variants of the software AMD were analyzed with and without weighting of the metrics evaluation criteria. In addition, the software AMD was tested on two types of software — with low dependability requirements (the first software package) and high requirements (the second critical software package). For the first software package, the model gave the following results: 12.2845 and 18.615 points, respectively. For the second software package, the scores were 13.962 and 19.655, respectively. The absence of weighting factors in the criteria leads to an increase in the assessment of the level of software dependability and is more visible and confirms the high level of software dependability for a critical application system, ensuring the patient's vital activity in the intensive care unit after surgery.

It should be noted that the task of building a model for assessing the level of software dependability does not lose its relevance over time. This can be explained by the fact that, firstly, it is impossible to develop a universal system of characteristics for all classes of software, and, secondly, a once-built AMD for a certain class of software may eventually cease to correspond to the dynamics of its functional capabilities.

REFERENCES

1. Fedukhin O.V. Expert assessment of the level of dependability of computer systems. *Mathematical machines and systems*. 2019. N 2. P. 131–147.

2. DSTU ISO/IEC 25051:2016. Systems and software engineering. Requirements for the quality of systems and software and its evaluation (SQuaRE). Quality requirements for a ready-to-use software product (RUSP) and instructions for its testing (ISO/IEC 25051:2014, IDT). URL: https://www.iso.org/standard/61579.html.

3. Redkovska M.M., Fedukhin A.V. Attribute model for assessing software dependability taking into account DSTU ISO/IEC 25051:2016. *Mathematical machines and systems.* 2023. N 3. P. 113–133.

Стаття надійшла до редакції 12.08.2024