p-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2026. № 1
e-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2026. № 1

UDC 004.8

METHOD FOR CORRECTION OF MULTISUBJECTIVE MULTI-FACTORIAL ENVIRONMENTS OF SOFTWARE COMPLEXES' SUPPORT

Pukach A. I. – PhD, Assistant of Automated Control Systems Department, Institute of Computer Sciences and Informational Technologies, Lviv Polytechnic National University, Lviv, Ukraine. ROR: https://ror.org/0542q3127. ORCID: https://orcid.org/0009-0001-8563-3311.

Teslyuk V. M. – Doctor of Sciences, Professor, Head of Automated Control Systems Department, Institute of Computer Sciences and Informational Technologies, Lviv Polytechnic National University, Lviv, Ukraine. ROR: https://ror.org/0542q3127. ORCID: https://orcid.org/0000-0002-5974-9310.

ABSTRACT

Context. The problem of correction of multisubjective multifactorial environments of software complexes' support is considered in this research, necessary to provide the possibility(-ies) of adjusting the perception's subjectivization of the support object (the supported software, as well as related processes of its complex support), caused by the influence of relevant impact factors. **The object of research** is a process of correction of multisubjective multifactorial environments of software complexes' support. **The subject of research** are methods and means of correction of a multisubjective multifactorial environments of software complexes' support, as well as methods of an artificial neural networks (in particular: a multilayer perceptron).

Objective – is the development of method for correction of multisubjective multifactorial environments of software complexes' support.

Method. The development of a method for correction of multisubjective multifactorial environments of software' support is proposed. which provides possibilities for the necessary adjustments of the perception subjectivization of the researched support objects (which could be either the supported software itself, as well as the related processes for its comprehensive support) relevant (directly or directly) interacting subjects, who provide and implement this comprehensive support of the researched supported software product, in order to provide the possibility(-ies) of further automation and intellectualization of its comprehensive support.

Results. The results of functioning of the developed method – are relevant models of adjusted multisubjective multifactorial environments of software complexes' support, obtained in result of solving a relevant scientific and applied problem of adjusting such class of environments. The developed method provides the opportunity(-ies) for studying the processes of collective perception's subjectivization (caused by the influence of existing impact factors) of the objects of comprehensive support by the appropriate related subjects, which directly provide and implement this support, and also facilitates and ensures for further automation and intellectualization of such complex support of various software products and complexes in this separate and exact functional and procedural segment. As a practical approbation of the developed method, – the results of solved applied practical task of determination and further correction the impact factors of maximum imbalance of the researched multisubjective multifactorial environment (representing the technician team of the supported software product) are given.

Conclusions. The developed method solves the declared problem of correction of multisubjective multifactorial environments of software complexes' support. At the same time, the obtained results of performed practical approbation of the developed method confirm its functionality in solving a range of scientific and applied tasks based on the processes of collective perception's subjectivization of support objects (the supported software complexes, as well as the processes of their comprehensive support), which (those tasks), in turn, are included into the cluster of a more valuable scientific and applied problem of software products' comprehensive support automation and intellectualization.

KEYWORDS: software product, comprehensive support, impact factors, automation, correction, multisubjective multifactorial environment, neural networks, multilayer perceptron.

ABBREVIATIONS

ABC is an Artificial Bee Colony Algorithm;
ACO is an Ant Colony Optimization;
AI is an Artificial Intelligence;
ANN is an Artificial Neural Network;
CCS is a Cartesian Coordinate System;
CI/CD is a Continuous Integration / Continuous Delivery;
DevOps is a Development and Operations;
DM is a Data Mining;
EAI is a Explainable Artificial Intelligence;
FA is a Firefly Algorithm;
GA is a Genetic Algorithm;
HA is a Hybrid Algorithms;
HCA is a Hill Climbing Algorithm;
MP is a Multilayer Perceptron;
ML is a Machine Learning;
NLP is a Natural Language Processing;
PCS is a Polar Coordinate System;
PSO is a Particle Swarm Optimization;
SVM is a Support Vector Machine.

NOMENCLATURE

Δ is an adjustment parameter;

$\Delta_{[i][j]}$ is an adjustment parameter component for j-th impact factor of personal multifactor portrait of i-th subject;

$(\rho^{Dest}_{[i,j]};\varphi^{Dest}_{[i,j]})$ is a component target coordinates (in the PCS) of the j-th impact factor of the i-th subject's personal multifactor portrait, which can be achieved using the adjustment parameter $\Delta_{[i,j]}$;

DOI 10.15588/1607-3274-2026-1-17

$(\rho^{Curr}_{[i,j]};\varphi^{Curr}_{[i,j]})$ is a component current coordinates (in the PCS) of the j-th impact factor of the i-th subject's personal multifactor portrait;

$FCSP_{[j]}(Obj)$ is a component of the j-th impact factor inside the subjectivization function of the personalized perception (of the researched support object) by the current interaction subject;

$Fsubj_{[i]}$ is a nonlinear subjectivization function of personalized perception of the support object by the i-th interaction subject;

Obj is a variable-identifier of the researched support object;

$PCoPsMfSE_{[j]}$ is a parametric characteristics of a multisubjective multifactorial environment of the supported software complex;

$(x^{C}_{[i,j]};y^{C}_{[i,j]})$ is a component current coordinates (in the CCS) of the j-th impact factor of the i-th subject's personal multifactor portrait;

$(x^{D}_{[i,j]};y^{D}_{[i,j]})$ is a component target coordinates (in the CCS) of the j-th impact factor of the i-th subject's personal multifactor portrait, which can be achieved using the adjustment parameter $\Delta_{[i,j]}$;

$xc^{2}_{[i,j]}$ is a x-coordinate (in the CCS) of the adjustment parameter's current position (of the j-th impact factor of the i-th subject's personal multifactor portrait);

$xd^{2}_{[i,j]}$ is a x-coordinate (in the CCS) of the adjustment parameter's target position (of the j-th impact factor of the i-th subject's personal multifactor portrait);

$yc^{2}_{[i,j]}$ is a y-coordinate (in the CCS) of the adjustment parameter's current position (of the j-th impact factor of the i-th subject's personal multifactor portrait);

$yd^{2}_{[i,j]}$ is a y-coordinate (in the CCS) of the adjustment parameter's target position (of the j-th impact factor of the i-th subject's personal multifactor portrait).

INTRODUCTION

One of the major and key components of the life cycle of any software product – is its comprehensive support, which includes, in particular, such elements as: development, testing, implementation, environment configuration, and processing of requests (both external from outside customers' companies, and internal from the inside members of the development company itself).

At the same time, the automation of this complex support of various software products – is a complex scientific and applied problem, which includes a whole range of relevant scientific and applied tasks, including, among others, tasks based on the processes of collective subjectivization of the perception of support objects (the supported software complexes, as well as the processes of their comprehensive support), which arise as a result of presence of various impact factors, that lead to a distortion of the objective perception (of the object of support) by the relevant subjects (e.g. personnel), which, in fact, directly provide and implement this comprehensive support.

Thus, all available subjects of the comprehensive support (of any supported software complex) form a corresponding multisubjective multifactorial support environment, which is synthesized on the basis of a set of their individual multifactor representations of a personalized perception of the same support object.

The object of research is a process of correction of a multisubjective multifactorial environments of software complexes' support.

The subject of research are methods and means of correction of a multisubjective multifactorial environments of software complexes' support, as well as methods of an artificial neural networks (in particular: a multilayer perceptron).

The objective of the research consists in the development of method for correction of a multisubjective multifactorial environments (of software complexes' comprehensive support), which provides/ensures possibilities for the necessary adjustments of the perception's subjectivization of the researched support object (which can act as directly supported software product/complex itself, as well as processes related to its comprehensive support) by the relevant interaction subjects who directly provide and implement this comprehensive support (for the supported researched software product/complex), in order to provide the possibility(-ies) of further automation and intellectualization of suck kind comprehensive support.

1 PROBLEM STATEMENT

Let's consider the formalization of given problem of analysis a multisubjective multifactorial support environment – in the relevant form of a nonlinear polycriterial dependence task.

Thus, in considered case, the input variables of the problem – are nonlinear functions of subjectivization of the personalized perception (of the support object) by each of the interaction subjects: $Fsubj_{[i]}=[FCSP_{[j]}(Obj)]$ ($i\in[1..n]$, $j\in[1..m]$), where: Obj – variable-identifier of the researched support object; n – number of subjects interacting with the support object; m – number of declared impact factors.

The output variables of given problem – are the parametric characteristics of the multisubjective multifactorial support environment: $PCoPsMfSE_{[j]}$ (j=[1..m]), де m – number of declared impact factors.

Let us have a set of functions of subjectivization of the personalized perception of the support object by each of the subjects interacting with this object, that form relevant parametric characteristics of the multisubjective multifactorial support environment of the software complex:

DOI 10.15588/1607-3274-2026-1-17

$$PCoPsMfSE_{[j]} = \frac{\sum_{i=1}^{n} Fsubj_{[i]}[FCSP_{[j]}(Obj)]}{n}. \quad (1)$$

The main mandatory and necessary criterion of the given problem – is the finiteness of the set of support subjects, as well as the finiteness of the declared impact factors' set, which is due to the possibility of operating (in scope of given problem) with only a certain constant number of these pre-determined support subjects and impact factors.

Limitation of the problem:

1. The value of a personalized perception (of the support object) by each of the interacting subjects $Fsubj_{[i]}$ must be given as real numbers in a normalized representation form (that is, in the range of values between 0.0 and 1.0): $Fsubj[1..n] \in [0..1]$.

Expression (1) provides the possibility of interpreting the given problem of analyzing a multisubjective multifactorial support environments.

However, such an interpretation requires an additional mechanism to provide the possibility of correction (including balancing, or others) of the obtained multisubjective multifactorial support environments, taking into account the individual specifics and peculiarities of each such environment at the stage of forming/construction its interpretation.

Thus rise corresponding scientific and applied problem of correction of a multisubjective multifactorial environments of software complexes' support, for the purpose of solving which, in fact, a corresponding specialized method has been developed and presented in this research.

The main purpose of this article is to highlight the developed method, as well as the corresponding models for adjusting the investigated multisubjective multifactorial support environments, which together provide the possibility(-ies) of solving given scientific and applied problem of correction a multisubjective multifactorial support environments of software complexes.

2 REVIEW OF THE LITERATURE

The analysis of existing researches and publications was implemented both in the direction of automation of the components of software products' complex support, as well as in the research direction of the process(-es) of perception subjectivization of supported software complexes. Based on the analysis, the following interpreted results and conclusions have been obtained, presented below. In particular, the most common and basic areas of automation of comprehensive support for any software product(s) are: testing automation, DevOps automation, and automation of request processing (both external and internal).

The authors of the work [1] carried out a comprehensive comparison (specifically in the context of software testing automation) of such known Machine Learning and Data Mining algorithms as:

– HCA;
– ABC;
– FA;
– PSO;
– GA;
– ACO;
– ANN;
– SVM;
– HA.

In scope of research [2], author analyzed a batch of works related to application of AI in software testing and debugging, as well as the prospects for the application of artificial intelligence in these areas, and provided a brief summary of the methodologies, methods and approaches currently used in this field, in particular:

– using deep learning for creating test cases;
– usage of EAI for debugging;
– using reinforcement learning for test set optimization;
– usage of NLP for requirements assessment;
– use of artificial intelligence for test automation and continuous testing;
– and finally – usage of CI/CD, in order to simplify the whole process.

Authors of research [3] investigate the application of machine learning methods to increase the efficiency of software testing automation systems, based on a methodology that includes a comparative analysis of conventional testing methods with an integrated approach to machine learning, measuring performance through accuracy, execution speed, and resource utilization indicators, and, as a result, – authors confirm a significant increase in testing efficiency with using machine learning approach, methods and means.

The study [4] evaluates various AI techniques (including ML and NLP) and their application for test cases creation, software testing process optimization, and software defect(s) prediction, and the obtained results – highlight the efficiency and quality improvements achieved through software testing using AI.

Continuing an overview, authors of research [5] explore key aspects of AI and ML usage in DevOps, especially such kind of usage as: automated source code quality analysis, as well as predictive analytics for deployment and self-healing systems, and also study of tools and technologies that facilitate DevOps based on artificial intelligence, including, in particular, relevant machine learning frameworks (such as TensorFlow), and observation platforms (such as Datadog).

The work [6] explores the contribution of artificial intelligence to various aspects of DevOps, including source code management, CI/CD pipelines, deployment infrastructure, software testing infrastructure, logging mechanisms, data analysis tools, and comprehensive reporting systems, and also studies the impact of artificial intelligence on team communication, collaboration, and workflow orchestration in DevOps environments.

Authors of research [7] have conducted a systematic literature review using the PSALSAR Framework as a

DOI 10.15588/1607-3274-2026-1-17

tool for researching these relevant sources of information based on the SCOPUS database (including the Elsevier, Research Gate, and Semantic Scholar databases), starting from 2012 to 2022, which represents a comprehensive picture of usage of an artificial intelligence and machine learning technologies precisely in the context of automation of users' request(s) processing.

In the context of work [8], the integration of Apache Kafka (which is an existing platform for instant data streaming) with complex machine learning methods is considered – in order to ensure adaptive change(s) and improve customer support responses, which allows to significantly improve the efficiency and customization of contacts with customers (end users).

However, in the context of existing researches, there is no appropriate analysis (as well as, in fact, synthesis) of relevant methods and means of correction a multisubjective multifactorial environments for supporting software complexes, for which appropriate automation technologies (of their comprehensive support) are being implemented, including: testing, DevOps, or processing requests (and/or appeals) from clients/customers and end users.

In turn, this leads to the emergence of an appropriate relevant actual scientific and applied problem of correction of multisubjective multifactorial environments of software complexes' support, necessary to provide the possibility(-ies) for correction the perception's subjectivization of the support object (the supported software, as well as related processes of its complex support), caused by the influence of numerous relevant existing impact factors.

3 MATERIALS AND METHODS

The formation of a multisubjective multifactorial support environments (for software complexes' comprehensive support) is carried out on the basis of relevant personal multifactor portraits of separate subjects [9], who directly "shape" (i.e. form) the researched support environment. Thus, a multifactor portraits of support subjects (e.g. support personnel), in fact, – constitute an elementary structural and functional component of the relevant researched multisubjective multifactorial support environments of software complexes.

Accordingly, all corrections of the support environment are carried out directly through these structural and functional elementary components, which are – the relevant personal multifactor portraits of the existing subjects of each particular researched support environment. In the context of the conducted research, two variants of adjustment models (of the researched multisubjective multifactorial environments of software complexes' comprehensive support) have been developed and proposed, presented below.

In particular, one of the model options involves adjustment of the researched multisubjective multifactorial support environment – by direct adjustment of personal multifactor portraits of those subjects (e.g. support personnel) who form this support environment.

The main advantage of this option is the absence of need to encapsulate any additional (e.g.: "new", "non-native", "external" or "foreign") subjects, maintaining, in such way, the conservatism and integrity of the researched environment(s).

This particular version of the model (of adjusting multisubjective multifactorial environments of software complexes comprehensive support) is represented by the following expression (2) given below:

$$PCoPsMfSE_{[j]} = \frac{\sum_{i=1}^{n} Fsubj_{[i]}[FCSP_{[j]}(Obj) + \Delta_{[i][j]}]}{n}, \quad (2)$$

At the same time, various/different systems of distribution of the value of the correction parameter Δ – are possible.

In particular, within the framework of this research, the following variants have been developed and proposed, based on:

1) the PCS;
2) the CCS;
3) a vector system;
4) the Archimedes spiral.

Let's consider in more detail each of the proposed variants of systems for distributing the value of the correction parameter Δ.

The polar coordinate system variant declares the adjustment parameter Δ as follows (3):

$$\Delta_{[i,j]} = (\rho^{Dest}_{[i,j]}; \varphi^{Dest}_{[i,j]}) - (\rho^{Currr}_{[i,j]}; \varphi^{Currr}_{[i,j]}). \quad (3)$$

The main feature of the variant which is based on the polar coordinate system – is its advantages in solving situations and tasks when the adjustment of the component of required impact factor (of the investigated subject's personal multifactor portrait) is easier (and/or more convenient) to interpret as a vector of the deviation angle between the current and target values of the researched impact factor, since in this case there is no need to interpret such a relationship by using more complex trigonometric equations.

Another variant which is based on the Cartesian coordinate system is a little bit similar to the previous one, with the only significant difference consisting in a fact that the target and the current coordinates (of the component of j-th impact factor of the i-th subject's personal multifactor portrait) are actually given in the Cartesian coordinate system, and, in particular, can be represented by expression (4) below:

$$\Delta_{[i,j]} = (x^{D}_{[i,j]}; y^{D}_{[i,j]}) - (x^{C}_{[i,j]}; y^{C}_{[i,j]}). \quad (4)$$

The main feature of the option which is based on the CCS – is its advantages in solving tasks and situations in which the adjustment of the required impact factor's component can be carried out with a minimum set of iterations of simple movement between balancing points with previously known (given) exact coordinates.

DOI 10.15588/1607-3274-2026-1-17

The variant which is based on the vector system – represents the distribution of the value of correction parameter Δ as the interference of decomposition vectors of the components representing impact factors of personal multifactor portrait of the researched subject, and is described by the following expression (5):

$$\Delta_{[i,j]} = \sum_{k=1}^{m-1} Fr(\sqrt{((v_k)^2 + (v_{k+1})^2 - 2 \cdot v_k \cdot v_{k+1} \cdot \cos(\alpha_k^{k+1})))}, \quad (5)$$

where m – number of interference vectors; k – current interference vector's pointer counter; Fr – the recursive function of the interference of the current and the next vectors of the set; v_k – k-th vector (from the declared set of vectors) of influence onto the researched subject's personal multifactor portrait; α_k^{k+1} – the angle between the vectors v_k та v_{k+1}.

The main feature of the considered variant of the adjustment parameter Δ value distribution – is its advantages in solving those adjustment problems where the adjustment parameter is most expedient and/or convenient to be displayed precisely as an interference (superposition, or summation) of vectors representing appropriate decomposition components of the impact factors of the researched subject's personal multifactor portrait.

The variant which is based on the Archimedes spiral – actually represents a radial coordinate system, where the distribution of correction parameter Δ value is possible only within the framework of its existence on the plane (or in space) of the concentric circles of the Archimedes spiral, the transition between which is possible only along a radial trajectory, which, in fact, describes the dynamics of the: correction parameter's value distribution (relative to the investigated object and the subject); as well as the existing impact factors.

Accordingly, this variant of the adjustment parameter's distribution can be described by the following expression (6):

$$\Delta_{[i,j]} = (\sqrt{xd_{[i,j]}^2 + yd_{[i,j]}^2}) - (\sqrt{xc_{[i,j]}^2 + yc_{[i,j]}^2}). \quad (6)$$

The main feature of the considered variant (based on the Archimedes spiral) of the correction parameter's Δ value distribution – is its advantages in solving a cluster of problems related to the correction of those multisubjective multifactorial environments in which the trajectory of change (e.g. dynamics) of impact factors has a spiral (or radial, or concentric) form, due to the peculiarity of their interaction both: with each other, as well as with the subject and the object of the researched environment.

Thus, the proposed variants of the adjustment parameter's Δ value distribution (for the above-presented model option, which provides the adjustment of the researched multisubjective multifactorial support environment by direct adjusting the particular personal multifactor portraits of those subjects who form this environment, which is implemented through corresponding adjustment parameter Δ), are considered.

While another model option involves adjusting existing researched multisubjective multifactorial support environment by encapsulating (into this environment) additional new subjects with such personal multifactor portraits, which would ensure a shift in the common/general portrait of the entire environment (into which they are encapsulated) in the required vector/direction.

In turn, main advantage of this option is the preservation of the primary (original) portraits of already existing subjects that form the researched environment, that is ensuring a conservatism in relation to these subjects.

Accordingly, this version of the model for adjusting multisubjective multifactorial environments (of software complexes' comprehensive support) is represented by expression (7) below:

$$PCoPsMfSE_{[j]} = \frac{\sum_{i=1}^{n+n'} Fsubj_{[i]}[FCSP_{[j]}(Obj)]}{n+n'}, \quad (7)$$

where n' – number of additional new subjects, encapsulated into existing currently researched multisubjective multifactorial environment of complex support, necessary for appropriate adjustment(s) of this environment.

4 EXPERIMENTS

The experiment consists of the constructing a multisubjective multifactorial environment (for the researched support object), which is performed on the basis of the preliminary constructed personal multifactor portraits [9] of each of the subjects who form this entire environment. After that, the obtained multisubjective multifactorial environment is analyzed for necessity of its correction(s), and the required (appropriate) option and variant of correction are selected (in accordance to the characteristics of both the environment itself and the correction tasks) in order to select the most optimal option and variant. At the final stage of the experiment, the obtained identification results are presented in an arbitrary (convenient) form.

5 RESULTS

The main task/purpose of the developed method, presented in this research, is obtaining an adjusted multisubjective multifactorial environment according to set/predefined requirements (assessments, criteria, tasks, etc.). Let's consider obtained results on the example of solving a practical applied task of determination and further correction the impact factors of maximum imbalance of the researched multisubjective multifactorial environment (representing the technician team of the supported software product).

Figure 1 below presents a visualization of personal multifactor portraits of all subjects – team members of the investigated support environment. It should be also noted that such visualization actually represents nothing more than an expanded form of representation of the researched multisubjective multifactorial environment itself, with a detailed palette of all its constituent components of decomposition (both impact factors and subjects which "shape"/form it).

DOI 10.15588/1607-3274-2026-1-17

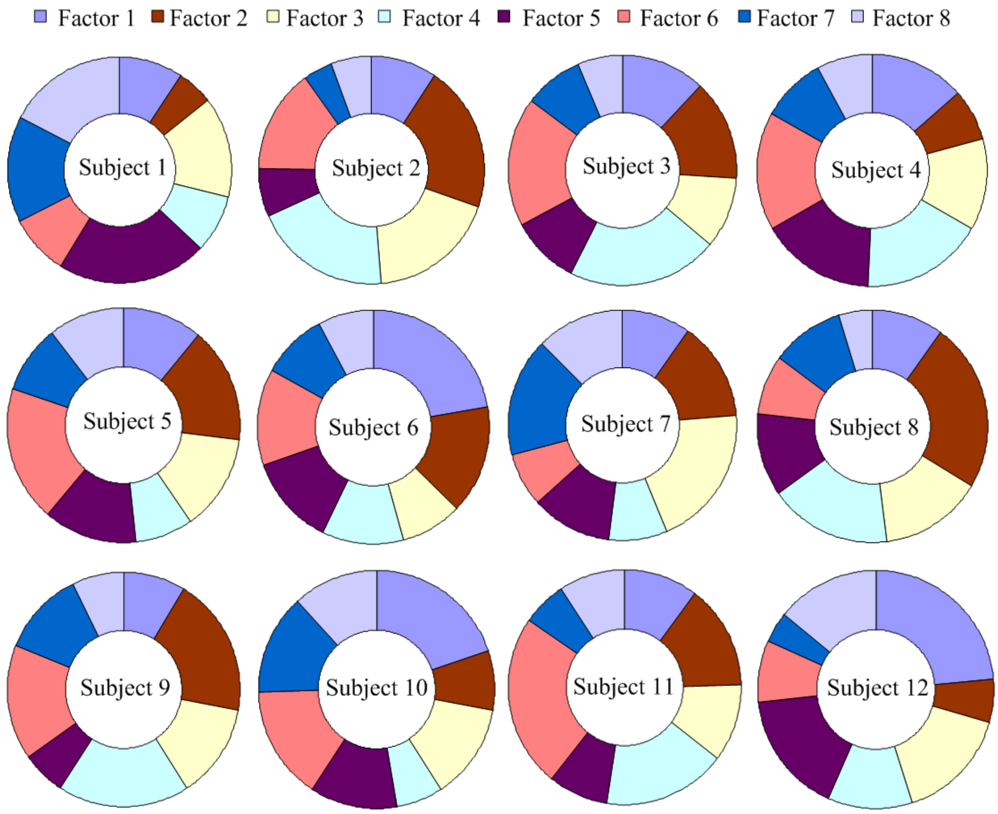

Figure 1 – Visualization of personal multifactor portraits of all subjects – team members of the investigated support environment.

In turn, the following Figure 2 represents the dynamics of the polysubject distribution of impact factors (within the framework of currently researched multisubjective multifactorial environment).

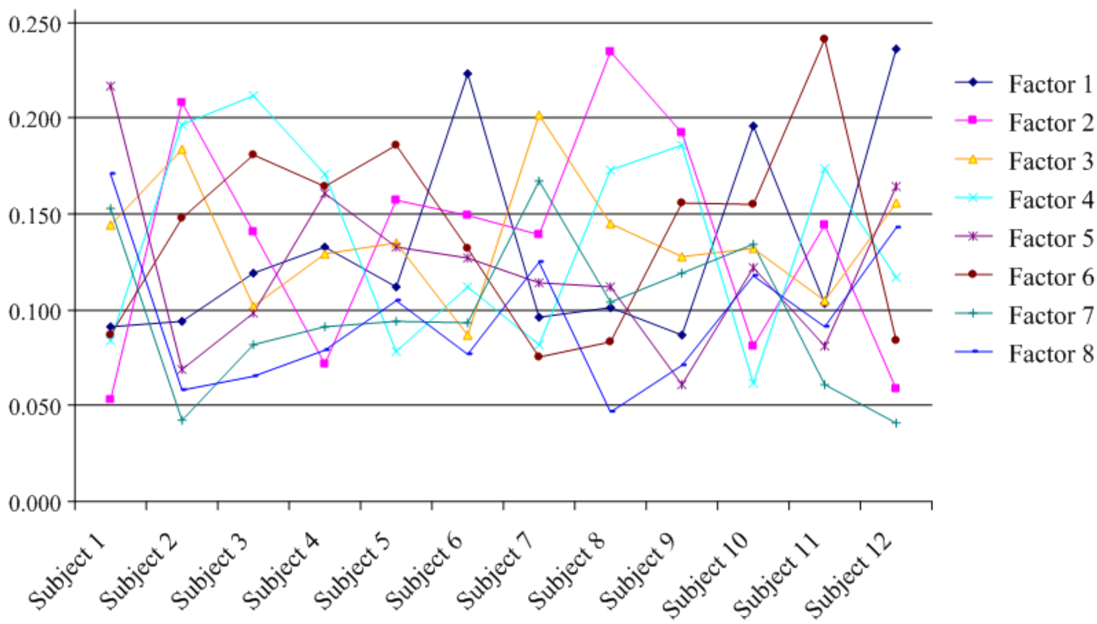

Figure 2. – Visualization of the dynamics of the polysubject distribution of impact factors (within the framework of currently researched multisubjective multifactorial environment)

Table 1 below displays the numerical data obtained for the personal multifactor portraits of all subjects – team members of the investigated support environment. In turn, the last row ("Average") of Table 1 contains the average values for all considered impact factors.

DOI 10.15588/1607-3274-2026-1-17

Table 1 – Numerical data obtained for the personal multifactor portraits of all subjects – team members of the investigated support environment

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6	Factor 7	Factor 8
Subject 1	0.091	0.053	0.144	0.084	0.217	0.087	0.153	0.171
Subject 2	0.094	0.208	0.184	0.197	0.069	0.148	0.042	0.058
Subject 3	0.119	0.141	0.102	0.212	0.098	0.181	0.082	0.065
Subject 4	0.133	0.072	0.129	0.171	0.161	0.164	0.091	0.079
Subject 5	0.112	0.157	0.135	0.078	0.133	0.186	0.094	0.105
Subject 6	0.223	0.149	0.087	0.112	0.127	0.132	0.093	0.077
Subject 7	0.096	0.139	0.202	0.082	0.114	0.075	0.167	0.125
Subject 8	0.101	0.235	0.145	0.173	0.112	0.083	0.104	0.047
Subject 9	0.087	0.192	0.128	0.186	0.061	0.156	0.119	0.071
Subject 10	0.196	0.081	0.132	0.062	0.122	0.155	0.134	0.118
Subject 11	0.103	0.144	0.105	0.174	0.081	0.241	0.061	0.091
Subject 12	0.236	0.059	0.156	0.117	0.164	0.084	0.041	0.143
Average	0.133	0.136	0.137	0.137	0.122	0.141	0.098	0.096

At the same time, Table 2 below presents corresponding calculation results of the differences in values of each of the impact factors (for each separate subject's individual multifactor portrait) from the average value (of the same impact factor among all these subjects).

Table 2 – Calculation results of the differences in values of each impact factors (for each separate subject's individual multifactor portrait) from the average value (of the same impact factor among all these subjects)

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6	Factor 7	Factor 8		
Subject 1	–0.042	***–0.083***	0.007	–0.053	***0.095***	–0.054	0.055	0.075		
Subject 2	–0.039	0.072	0.047	0.060	–0.053	0.007	–0.056	–0.038		
Subject 3	–0.014	0.005	–0.035	0.075	–0.024	0.040	–0.016	–0.031		
Subject 4	0.000	–0.064	–0.008	0.034	0.039	0.023	–0.007	–0.017		
Subject 5	–0.021	0.021	–0.002	–0.059	0.011	0.045	–0.004	0.009		
Subject 6	***0.090***	0.013	–0.050	–0.025	0.005	–0.009	–0.005	–0.019		
Subject 7	–0.037	0.003	0.065	–0.055	–0.008	–0.066	0.069	0.029		
Subject 8	–0.032	***0.099***	0.008	0.036	–0.010	–0.058	0.006	–0.049		
Subject 9	–0.046	0.056	–0.009	0.049	–0.061	0.015	0.021	–0.025		
Subject 10	0.063	–0.055	–0.005	***–0.075***	0.000	0.014	0.036	0.022		
Subject 11	–0.030	0.008	–0.032	0.037	–0.041	***0.100***	–0.037	–0.005		
Subject 12	***0.103***	***–0.077***	0.019	–0.020	0.042	–0.057	–0.057	0.047		
*recommended values of adjustment parameter: Δmin=–0.083–(–0.061); Δmax=0.103–0.085									min/ max	aver–age
Min	*–0.046*	*–0.083*	*–0.050*	*–0.075*	*–0.061*	*–0.066*	*–0.057*	*–0.049*	–0.083	–0.061
Max	*0.103*	*0.099*	*0.065*	*0.075*	*0.095*	*0.100*	*0.069*	*0.075*	0.103	0.085

At the same time, the last two rows ("Min" and "Max") of Table 2 contain (appropriately): the minimal and the maximal values present in each column of the previous rows of the same table (i.e., among the values of the obtained differences of each of the impact factors).

Among all the results obtained (and presented in Table 2) of the differences between the values of each of the impact factors and the average value of the same impact factor, in the context of given practical applied task (determination and further correction the impact factors of maximum imbalance of the researched multisubjective multifactorial environment representing the technician team of the supported software product) – the maximum level of interest is represented by those values that represent two key indicators, namely: the minimum difference value, as well as the maximum difference value. Because these values, actually, indicate those impact factors which bring the maximal imbalance within the framework of the expanded representation form of the researched multisubjective multifactorial environment (that represents the technician team of the supported software product).

Accordingly, correction of these indicators makes it possible to ensure an increase in the balance level of the entire investigated multisubjective multifactorial environment of the researched software complex's support team. At the same time, the recommended value of the adjustment parameter will be the difference between the current value (of one of the two key indicators) and the corresponding arithmetic mean value (for the same key indicator).

Figure 3 below provides a corresponding graphical interpretation, visualized in the form of a relevant histogram, of the data from Table 2, which represents the results of calculating the differences in values between each of the impact factors and the average value of the same impact factor.

DOI 10.15588/1607-3274-2026-1-17

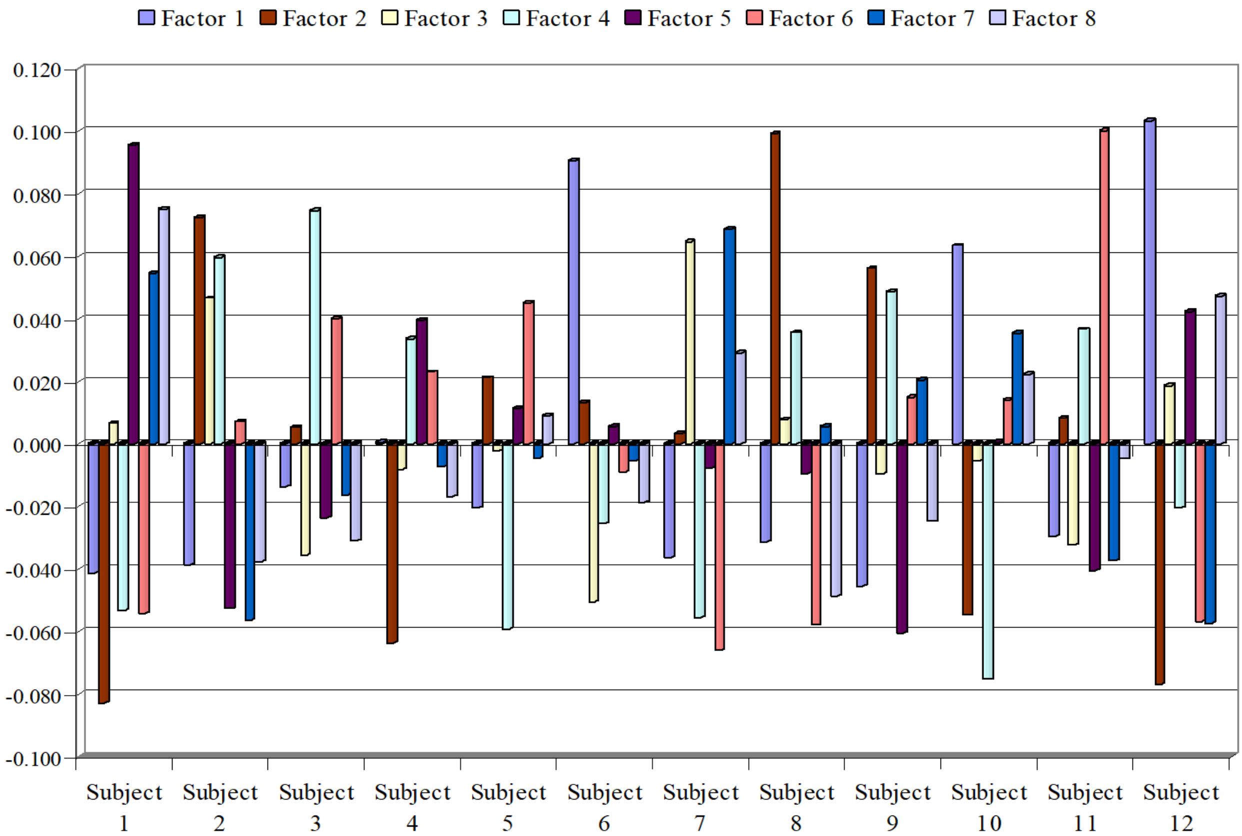

Figure 3 – Visualization of calculation results of the differences in values between each of the impact factors and the average value of the same impact factor.

Besides that, it is also possible to additionally establish a certain (necessary or expedient) threshold deviation value (both in the absolute and relative scales), above which the value of the corresponding impact factor of the relevant subject will be considered as critically affecting the imbalance of the investigated multisubjective multifactorial support environment – and, therefore, requires correction(s).

Thus, for example, based on the data represented above in Table 2, as well as in Figure 3, the identified impact factors which have been considered as such that should be corrected (in the context of given / "being solved" practical applied task of determination and further correction the impact factors of maximum imbalance of the researched multisubjective multifactorial environment representing the technician team of the supported software product) are the following:

– factor 1 within subject 12;
– factor 2 within subject 1;
– factor 2 within subject 12;
– factor 2 within subject 8;
– factor 5 within subject 1;
– factor 6 within subject 11;
– factor 1 within subject 6;
– factor 4 within subject 10.

Thus, determining the relevant impact factors influencing the maximum imbalance of the existing multisubjective multifactorial environment (of the researched software complex' support team), as well as determining the recommended value of the correction parameter of these impact factors – both together provide the possibility of their further correction using the developed method for correction of multisubjective multifactorial environments of software complexes' support.

Figure 4 below presents a visualization of calculation results of the differences in values for all considered impact factors (from the average value of the same impact factor) after the relevant correction(s) have been done for the previously determined impact factors (of maximal unbalancing of the investigated multisubjective multifactorial environment representing the research software complex' support team).

Therefore, as follows from the obtained results of the histogram visualized in Figure 4, as a result of performed correction and application of the developed method, – a reduction of the differences (between the values of each of the impact factors and the average value of the same impact factors) was ensured.

For a more clear understanding of the significance of obtained results, – Figure 5 below presents: the absolute values of the differences for each of the identified impact factors (for which correction has been applied), as well as the differences between the minimal / maximal values and the corresponding arithmetic mean value.

DOI 10.15588/1607-3274-2026-1-17

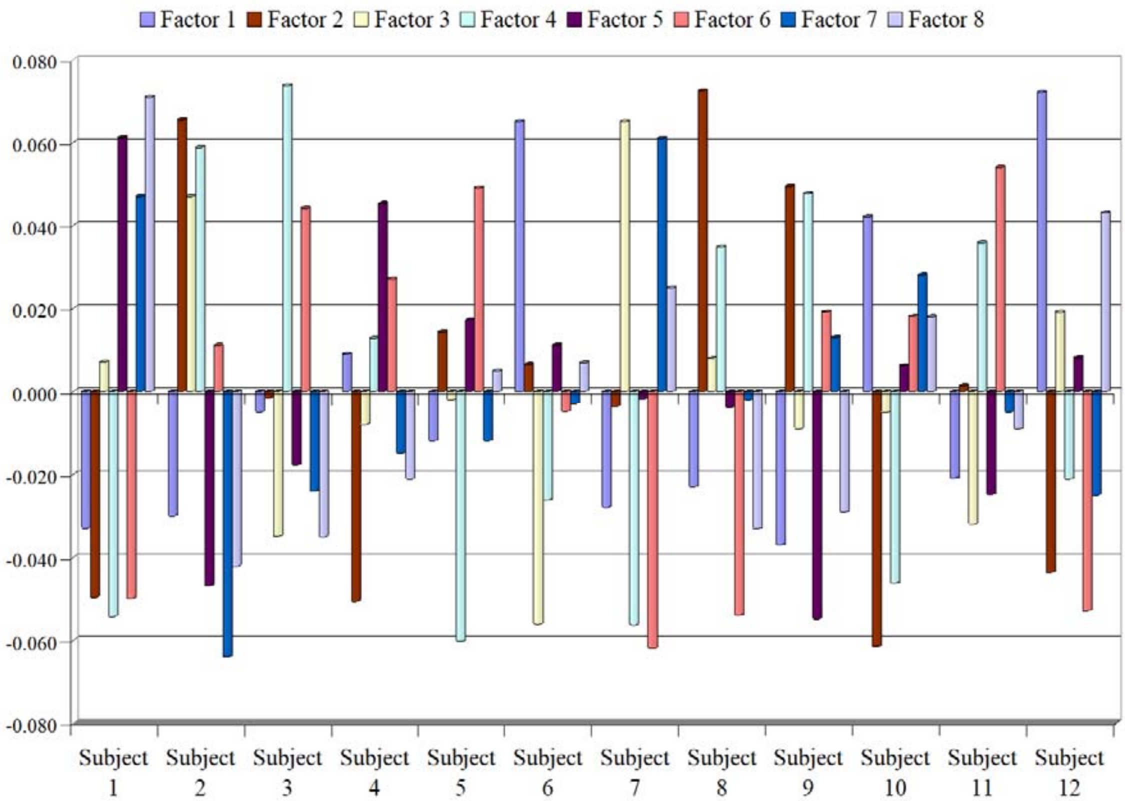

Figure 4 – Visualization of calculation results of the differences in values for all considered impact factors (from the average value of the same impact factor) after the relevant correction(s) have been done for the previously determined impact factors (of maximal unbalancing of the investigated multisubjective multifactorial environment representing the research software complex' support team)

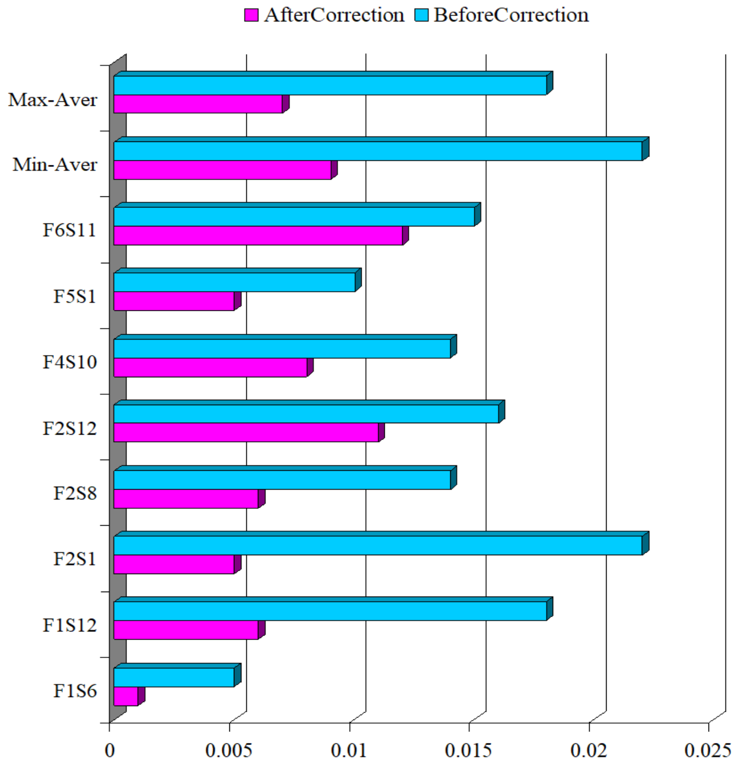

Figure 5 – Comparison visualization between differences' values before and after correction of the identified maximal imbalance's impact factors of the researched multisubjective multifactorial environment

DOI 10.15588/1607-3274-2026-1-17

Thus, as a result of applying the developed method (for correction of multisubjective multifactorial environments of software complexes' support) for solving the given relevant example practical applied task (of determination and further correction the impact factors of maximum imbalance of the researched multisubjective multifactorial environment representing the technician team of the supported software product) a significant improvement(s) in the indicators of the difference between the values of each of the identified "maximum imbalance" impact factors and the corresponding arithmetic mean values of these same factors, have been obtained.

In particular, the following specific improvement achievements have been obtained:

– for factor 1 within subject 6 – 80% improvement;
– for factor 1 within subject 12 – 67% improvement;
– for factor 2 within subject 1 – 77% improvement;
– for factor 2 within subject 8 – 57% improvement;
– for factor 2 within subject 12 – 31% improvement;
– for factor 4 within subject 10 – 43% improvement;
– for factor 5 within subject 1 – 50% improvement;
– for factor 6 within subject 11 – 20% improvement;
– in the context of the minimal deviation value – 59% improvement;
– in the context of the maximal deviation value – 61% improvement.

In addition, the developed method can be used (both in the context of solving the given practical applied tasks, as well as in general) using a recursive (cyclic) approach, where (at each separate cycle of recursion) the determination and correction of the corresponding necessary impact factors are implemented, thereby providing the possibility of "branching", parallelization and research of alternative variants of a multicriterial optimization (including balancing, for example) of any studied multisubjective multifactorial environments.

6 DISCUSSION

Research [10] confirms an importance of taking into account the impact factors (in particular, such global groups of factors as: human and social, as well as technical) on such processes of comprehensive software support as Requirements Engineering. It provides some idea(s) about the importance's distribution structure (of the influence of each of the categories of declared impact factors onto these processes), forming a more or less holistic picture of the environment for interaction of various subjects involved in these processes.

Another research [11] provides a clear list of key factors that affect the productivity of teams who provide and implement such a component of comprehensive software product support as Agile. At the same time, the peculiarity of this research is that it studies the factors that influence the productivity of entire teams (i.e., groups of subjects), but not a separate individuals.

At the same time, research [12] proposes a model (which is based on three dimensions and twelve key factors) that significantly affect Agile software development, and the results of the conducted research provide structuring an Agile environment(s) by taking into account all critical factors of success. At the same time, the presence of dimensions allows to adapt the proposed model in accordance with such important parameters of the developed and supported software products as their: size, nature and financial constraints.

So, absolutely all of the above studies, unfortunately, do not reveal the issues of forming, or, even more so: correcting environments based on these present and declared impact factors, while, to a greater extent, they just focus their main attention on these factors as a separate components, regardless of their integration with each other, and with the researched object(s).

However, at the same time, absolutely all of these studies confirm the criticality of the need of considering and taking into account factors influencing various processes, which (those processes) are the components of comprehensive support for software products.

While the developed and presented in current research method for correction of multisubjective multifactorial environments of software complexes' support allows not only to take into account these impact factors (by the way: without any restrictions on their set, equally processing each necessary and previously declared set of factors), but also to combine them into more significant components – a multisubjective multifactorial environments, but also: to provide the possibility(-ies) of correcting such environments, thereby influencing the quantitative and qualitative indicators of comprehensive support of the investigated supported software products and complexes.

As a further application of the developed method, we see its perspectives and promises both in solving a number of relevant practical applied tasks and problems, including, in particular, the problems of identifying and further correcting problematic factors of influence and/or subjects of the studied multisubjective multifactorial support environments, as well as scientific problems of intellectualizing the processes of software complexes' automated support of various both existing and developed software products.

Thus, given the wide range of relevant and related practical and/or applied problems, the feasibility of further investigation in the considered direction is pretty justified. Furthermore, the proposed method can be adapted for use in other areas of science which study and research any polysubject (or heterogeneous) multifactor environments, such as, for example: socionics, conflictology, social psychology, etc.

CONCLUSIONS

The method for correction of multisubjective multifactorial environments of software complexes' support has been proposed, developed and represented in this research. A key scientific & applied problem, which has been successfully resolved by the proposed method, – is the problem of correction of a multisubjective multifactorial environments of software complexes' support is considered in this research, necessary to provide the possibility(-ies) of adjusting the perception's subjectivization of

DOI 10.15588/1607-3274-2026-1-17

the support object (the supported software, as well as related processes of its complex support), caused by the influence of relevant impact factors.

Input data for the proposed method – are individual multifactor portraits of those researched subjects, who form the corresponding investigated multisubjective multifactorial support environment, as well as the corresponding tasks, defined for correcting this environment.

Two variants of models for adjusting the researched multisubjective multifactorial environments (of software complexes' comprehensive support) have been introduced, namely:

– variant by direct adjusting the personal multifactor portraits of those existing subjects who already form this environment;

– as well as a variant of adjustment the entire environment by encapsulating (into this environment) additional new subjects with such personal multifactor portraits that would ensure a "shift" in the portrait of the entire environment (into which they will be encapsulated) in the required correction vector/direction.

Besides that, additional variants of the correction parameter's Δ value distribution's system have been presented, in particular: based on the polar coordinate system; based on the Cartesian coordinate system; based on the vector system; and based on the Archimedean spiral, and their main features and practical advantages have been given as well.

The developed method provides the possibility(-ies) for studying the processes of collective perception's subjectivization (caused by the influence of various existing impact factors) of objects (of software complexes' comprehensive support) by the relevant subjects (e.g.: support personnel) who directly provide and implement this comprehensive support, and also makes it possible to further automate this complex support of software products in scope of the considered functional and procedural activity segment.

The scientific novelty consists in development of method for correction of multisubjective multifactorial environments of software complexes' support, which ensures possibility(-ies) for researching the factors influencing the processes of complex support of various software products, and also provides opportunities for further intellectualization of the automation processes of such comprehensive support. **The practical significance** consists in development of basic models for adjusting the researched multisubjective multifactorial environments (of software complexes' comprehensive support), as well as related variants of the necessary and suitable distribution system(s) for the value(s) of the adjustment parameter Δ for its better adaptation to the given tasks.

Prospects for further research consist in the designing of an extra specialized dedicated algorithmic and software supply for modeling the correction processes (for example, balancing, and others) of the studied multisubjective multifactorial environments (of software products' and complexes' comprehensive support), as well as in further application of the developed method in solving a range of relevant practical applied problems and tasks, including, in particular, the problems of identifying and further correcting problematic factors of influence and/or subjects of the studied multisubjective multifactorial support environments, as well as scientific tasks and problems of intellectualizing the processes of automation of the software products' and complexes' comprehensive support.

ACKNOWLEDGEMENTS

This research is proactive. It was carried out as a part of the scientific activity of the authors outside of the working hours at their main positions.

DECLARATIONS

Conflict of interest: The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

Authors' contributions: Andrii Pukach: the method for correction of multisubjective multifactorial environments of software complexes' support, methodology, resources, data curation, project administration, modeling, visualization, correspondence; Vasyl Teslyuk: conceptualization, supervision.

Data availability: The original contributions presented in this study are included in the article.

Software availability: The manuscript has no associated software.

Use of artificial intelligence tools: The authors confirm that they did not use artificial intelligence technologies in creating the submitted work.

REFERENCES

1. Salam M., Abdel-Fattah M., Moemen M. A survey on software testing automation using machine learning techniques, *International Journal of Computer Applications,* 2022, Vol. 183, Iss. 51, pp. 12–19. DOI: 10.5120/ijca2022921919
2. Garg U. Exploring the Use of Artificial Intelligence for Software Testing and Debugging, *International Journal of Electrical Engineering and Technology,* 2020, Vol. 11, Iss. 1, P. 94–102. – DOI: 10.17605/OSF.IO/43AE2
3. Kathiriya S., Karangara R., Challla N. Optimizing automated software testing with Machine Learning Techniques, *International Journal of Science and Research,* 2018, Vol. 7, Iss. 3, pp. 1960–1964. DOI: 10.21275/sr24304113021
4. Prathyusha N. Integrating AI in testing automation: Enhancing test coverage and predictive analysis for improved software quality, *World Journal of Advanced Engineering Technology and Sciences,* 2024, Vol. 13, Iss. 1, pp. 769–782. DOI: 10.30574/wjaets.2024.13.1.0486
5. Oyeniran O. C., Adewusi A. O., Adeleke A. G. et al. AI-driven DevOps: Leveraging machine learning for automated software deployment and maintenance [Electronic resource], *Engineering Science & Technology Journal,* 2023, Vol. 4, Iss. 6, pp. 728–740. Mode of access: https://www.researchgate.net/profile/Adams-Adeleke-2/publication/383915954_AI-driven_devops_Leveraging_machine_learning_for_automated_software_deployment_and_maintenance/links/66e0bf2a64f7bf7b19a5c2ed/AI-driven-devops-Leveraging-machine-

DOI 10.15588/1607-3274-2026-1-17

learning-for-automated-software-deployment-and-maintenance.pdf (date of access: 14.03.2025). Title from screen.
6. Ghantous G.B. Enhancing DevOps using AI [Electronic resource], *Information Control Systems & Technologies,* 2024, pp. 1–13. Mode of access: https://ceur-ws.org/Vol-3790/paper12.pdf (date of access: 14.03.2025). – Title from screen.
7. Wijaya A. S., Oktavia T. Machine learning approaches for helpdesk ticketing system: a systematic literature review [Electronic resource], *Journal of Theoretical and Applied Information Technology*, 2024, Vol. 102, No. 5, pp. 1831–1842. Mode of access: https://www.jatit.org/volumes/Vol102No5/14Vol102No5.pdf (date of access: 14.03.2025). – Title from screen.
8. Upadhyaya N. Enhancing real-time customer service through Adaptive Machine Learning, *International Journal of Advanced Research in Science, Communication and Technology*, 2024, Vol. 4, Iss. 1, pp. 630–636. – DOI: 10.48175/ijarsct-19381
9. Pukach A. I., Teslyuk V. M. Method of forming multifactor portraits of the subjects supporting software complexes, using a multilayer perceptron, *Radio Electronics, Computer Science, Control,* 2025, Vol. 1, pp. 130–141. DOI: 10.15588/1607-3274-2025-1-12
10. Hidellaarachchi D., Grundy J., Hoda R. et al.The Influence of Human Aspects on Requirements Engineering: Software Practitioners Perspective, *ACM Transactions on Software Engineering and Methodology*, 2021, Vol. 1, No. 1, pp. 1–37. DOI: 10.1145/1122445.1122456
11. Turic M., Celar S., Dragicevic S. Productivity Factors in Agile Software Development Projects, *Proceedings of the 34th DAAAM International Symposium on Intelligent Manufacturing and Automation,* 2023, P. 0004–0008. DOI: 10.2507/34th.daaam.proceedings.001
12. Muhammad A. et al. Investigating crucial factors of Agile software development through composite approach, *Intelligent Automation & Soft Computing,* 2021, Vol. 27, Iss. 1, pp. 15–34. DOI: 10.32604/iasc.2021.014427

Received 11.05.2025.
Accepted 09.01.2026.
Published 27.03.2026.

УДК 004.8

МЕТОД КОРЕКЦІЇ ПОЛІСУБ'ЄКТНИХ МУЛЬТИФАКТОРНИХ СЕРЕДОВИЩ ПІДТРИМКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (КОМПЛЕКСІВ, СИСТЕМ, ПРОДУКТІВ, ТОЩО)

Пукач А. І. – канд. техн. наук, асистент кафедри Автоматизованих Систем Управління Інституту Комп'ютерних Наук та Інформаційних Технологій Національного Університету «Львівська Політехніка», Львів, Україна. ROR: https://ror.org/0542q3127. ORCID: https://orcid.org/0009-0001-8563-3311.

Теслюк В. М. – д-р техн. наук, професор, завідувач кафедри Автоматизованих Систем Управління Інституту Комп'ютерних Наук та Інформаційних Технологій Національного Університету «Львівська Політехніка», Львів, Україна. ROR: https://ror.org/0542q3127. ORCID: https://orcid.org/0000-0002-5974-9310.

АНОТАЦІЯ

Актуальність. Розглянуто задачу корекції полісуб'єктних мультифакторних середовищ підтримки програмного забезпечення (комплексів, систем, продуктів, тощо), необхідної для забезпечення можливості коригування суб'єктивізації сприйняття об'єкта підтримки (підтримуваного програмного комплексу, або процесів його комплексної підтримки), зумовленого дією відповідних факторів впливу. **Об'єктом дослідження** є процес корекції полісуб'єктних мультифакторних середовищ підтримки програмного забезпечення (комплексів, систем, продуктів, тощо). **Предметом дослідження** є методи та засоби корекції полісуб'єктних мультифакторних середовищ підтримки програмного забезпечення (комплексів, систем, продуктів, тощо).

Метою роботи – є розроблення методу корекції полісуб'єктних мультифакторних середовищ підтримки програмного забезпечення (комплексів, систем, продуктів, тощо).

Метод. Запропоновано розроблення методу корекції полісуб'єктних мультифакторних середовищ підтримки програмного забезпечення (комплексів, систем, продуктів, тощо), що дає змогу здійснювати необхідні коригування суб'єктивізації сприйняття досліджуваного об'єкта підтримки (в якості якого може виступати як безпосередньо саме підтримуване програмне забезпечення, а також процеси дотичні до його комплексної підтримки) відповідними суб'єктами взаємодії (як прямої, так і опосередкованої), котрі безпосередньо забезпечують та здійснюють цю комплексну підтримку досліджуваного програмного продукту, з метою забезпечення можливостей подальшої автоматизації й інтелектуалізації цієї підтримки.

Результати. Результатами роботи розробленого методу є моделі відкоригованих полісуб'єктних мультифакторних середовищ підтримки програмного забезпечення (комплексів, систем, продуктів, тощо), отриманих врезультаті розв'язання релевантної науково-прикладної задачі коригування середовищ даного класу. Розроблений метод забезпечує можливість дослідження процесів колективної суб'єктивізації сприйняття (зумовленої дією наявних факторів впливу) об'єктів комплексної підтримки відповідними суб'єктами, що безпосередньо забезпечують та реалізують цю підтримку, а також дає змогу здійснити подальшу автоматизацію та інтелектуалізацію цієї комплексної підтримки програмних продуктів саме в даному функціонально-процесуальному сегменті. В якості практичної апробації розробленого методу, наведено результати розв'язаної прикладної практичної задачі визначення та подальшої корекції факторів впливу максимального розбалансування полісуб'єктного мультифакторного середовища команди підтримки досліджуваного програмного комплексу.

Висновки. Розроблений метод вирішує поставлену задачу корекції полісуб'єктних мультифакторних середовищ підтримки програмного забезпечення (комплексів, систем, продуктів, тощо). Водночас, отримані результати практичної апробації розробленого методу підтверджують його функціональність при вирішенні спектру науково-прикладних задач на основі процесів колективної суб'єктивізації сприйняття об'єктів підтримки (як підтримуваного програмного забезпечення, так і процесів дотичних до його комплексної підтримки), які (ці задачі), в свою чергу, входять до кластеру більш глобальної науково-прикладної проблеми автоматизації й інтелектуалізації комплексної підтримки програмних продуктів.

КЛЮЧОВІ СЛОВА: програмний продукт, комплексна підтримка, фактори впливу, автоматизація, корекція, полісуб'єктне мультифакторне середовище, нейронні мережі, багатошаровий перцептрон.

DOI 10.15588/1607-3274-2026-1-17

ЛІТЕРАТУРА

1. Salam M. A survey on software testing automation using machine learning techniques / M. Salam, M. Abdel-Fattah, M. Moemen // International Journal of Computer Applications. – 2022. – Vol. 183, Iss. 51. – P. 12–19. – DOI: 10.5120/ijca2022921919
2. Garg U. Exploring the Use of Artificial Intelligence for Software Testing and Debugging / U. Garg // International Journal of Electrical Engineering and Technology. – 2020. – Vol. 11, Iss. 1. – P. 94–102. – DOI: 10.17605/OSF.IO/43AE2
3. Kathiriya S. Optimizing automated software testing with Machine Learning Techniques / S. Kathiriya, R. Karangara, N. Challla // International Journal of Science and Research. – 2018. – Vol. 7, Iss. 3. – P. 1960–1964. – DOI: 10.21275/sr24304113021
4. Prathyusha N. Integrating AI in testing automation: Enhancing test coverage and predictive analysis for improved software quality / N. Prathyusha // World Journal of Advanced Engineering Technology and Sciences. – 2024. – Vol. 13, Iss. 1. – P. 769–782. – DOI: 10.30574/wjaets.2024.13.1.0486
5. AI-driven DevOps: Leveraging machine learning for automated software deployment and maintenance [Electronic resource] / [O. C. Oyeniran, A. O. Adewusi, A. G. Adeleke et al.] // Engineering Science & Technology Journal. – 2023. – Vol. 4, Iss. 6. – P. 728–740. – Mode of access: https://www.researchgate.net/profile/Adams-Adeleke-2/publication/383915954_AI-driven_devops_Leveraging_machine_learning_for_automated_software_deployment_and_maintenance/links/66e0bf2a64f7bf7b19a5c2ed/AI-driven-devops-Leveraging-machine-learning-for-automated-software-deployment-and-maintenance.pdf (date of access: 14.03.2025). – Title from screen.
6. Ghantous G. B. Enhancing DevOps using AI [Electronic resource] / G. B. Ghantous // Information Control Systems & Technologies. – 2024. – P. 1–13. – Mode of access: https://ceur-ws.org/Vol-3790/paper12.pdf (date of access: 14.03.2025). – Title from screen.
7. Wijaya A. S. Machine learning approaches for helpdesk ticketing system: a systematic literature review [Electronic resource] / A. S. Wijaya, T. Oktavia // Journal of Theoretical and Applied Information Technology. – 2024. – Vol. 102, No. 5. – P. 1831–1842. – Mode of access: https://www.jatit.org/volumes/Vol102No5/14Vol102No5.pdf (date of access: 14.03.2025). – Title from screen.
8. Upadhyaya N. Enhancing real-time customer service through Adaptive Machine Learning / N. Upadhyaya // International Journal of Advanced Research in Science, Communication and Technology. – 2024. – Vol. 4, Iss. 1. – P. 630–636. – DOI: 10.48175/ijarsct-19381
9. Pukach A. I. Method of forming multifactor portraits of the subjects supporting software complexes, using a multilayer perceptron / A. I. Pukach, V. M. Teslyuk // Radio Electronics, Computer Science, Control. – 2025. – Vol. 1. – P. 130–141. – DOI: 10.15588/1607-3274-2025-1-12
10. Hidellaarachchi D. The Influence of Human Aspects on Requirements Engineering: Software Practitioners Perspective / [D. Hidellaarachchi, J. Grundy, R. Hoda et al.] // ACM Transactions on Software Engineering and Methodology. – 2021. – Vol. 1. No. 1. – P. 1–37. – DOI: 10.1145/1122445.1122456
11. Turic M. Productivity Factors in Agile Software Development Projects / M. Turic, S. Celar, S. Dragicevic // Proceedings of the 34th DAAAM International Symposium on Intelligent Manufacturing and Automation. – 2023. – P. 0004–0008. – DOI: 10.2507/34th.daaam.proceedings.001
12. Investigating crucial factors of Agile software development through composite approach / A. Muhammad et al. // Intelligent Automation & Soft Computing. – 2021. – Vol. 27, Iss. 1. – P. 15–34. – DOI: 10.32604/iasc.2021.014427

DOI 10.15588/1607-3274-2026-1-17