UDC 004.8

METHOD FOR DEVELOPMENT MODELS OF POLYSUBJECT MULTIFACTOR ENVIRONMENT OF SOFTWARE COMPLEX'S SUPPORT

Pukach A. I. – PhD, Assistant of ACS Department, Institute of Computer Sciences and Informational Technologies, Lviv Polytechnic National University, Lviv, Ukraine.

Teslyuk V. M. – Doctor of Sciences, Professor, Head of ACS Department, Institute of Computer Sciences and Informational Technologies, Lviv Polytechnic National University, Lviv, Ukraine.

ABSTRACT

Context. The task of development the models of a polysubject multifactor environment for software's complex support is considered in this research, that ensures possibilities of taking into account the influence of various impact factors onto the supported software complexes themselves, onto their complex support's processes, as well as onto the subjects (interacting with them) who provide and implement this complex support. **The object of study** are the processes of complex support of software products, the processes of automation of this complex support, the processes of influence of impact factors onto the object and subjects of the complex support of software products, as well as the processes of perception's subjectivization of the supported object by relevant subjects of interaction with it. **The subject of study** are methods and means of artificial neural networks, in particular a multilayer perceptron, as well as a computer design and modeling. **Objective** is the development of the method for building models of a polysubject multifactor environment(s) of the complex support of software product(s).

Method. The developed method for building models of a polysubject multifactor environment of software complexes' support is proposed, which makes it possible (in an automated mode) to obtain appropriate models, based on which, later on – to investigate the strengths and weaknesses of a specific researched complex support's environment(s) of a particular investigated software product(s), in order to ensure further improvement and automation of this complex support based on the study and analysis of impact factors, which form the subjective vision and perception of this complex support by those subjects who directly provide and perform it, that is, in fact, on whom this support itself depends, as well as its corresponding qualitative and quantitative characteristics and indicators.

Results. The results of functioning of the developed method are corresponding models of investigated polysubject multifactor environments of the complex support of software products, which take into account the presence and the level of influence of relevant existing and present factors performing impact onto the subjects of interaction with supported software complexes, which (subjects) directly provide and perform the complex support for the studied software products, and also form relevant researched support environments. In addition, as an example of a practical application and approbation, the developed method was used, in particular, to solve the applied practical task of determining the dominant and the deficient factors of influence of a polysubject multifactor environment of the studied software complex's support, with presenting and analyzing the obtained results of resolving the given task.

Conclusions. The developed method resolves the problem of building models of a polysubject multifactor environment of the complex support of software products, and ensures taking into account the action of various impact factors performing influence onto the supported software complex itself, onto the processes of its support, as well as onto the subjects of interaction with it, which (subjects) provide and perform this complex support. In particular, the developed method provides possibilities for modeling and investigating a polysubject multifactor environments of the "in focus" software product's complex support, which reflect the global (or the local, it depends on the specific tasks) impact of various existing factors making influence onto the object of support itself (the supported software complex, or the processes of its complex support), as well as onto the subjects which directly carry out and implement this complex support in all its possible and/or declared manifestations. The practical approbation of the developed method has been carried out by solving specific applied practical tasks, one of which is presented, as an example, in this paper, – which is the task of determining the dominant and the deficient factors of influence of a polysubject multifactor environment of the studied software complex's support, and this approbation, actually, confirms its effectiveness in solving a stack of applied practical problems related to researching the impact of factors performing influence onto the complex support of software products, using the advantages of artificial intelligence technologies, machine learning, artificial neural networks, and multilayer perceptron in particular.

KEYWORDS: software product, complex support, software product support environment, impact factor, automation, artificial neural networks, multilayer perceptron.

ABBREVIATIONS

ANN – artificial neural networks;
DevOps – Development and Operations;
IT – information technologies;
MP – multilayer perceptron;
XML – EXtensible Markup Language.

NOMENCLATURE

F[*k*][*j*] is a level of presence of the *k*-th impact factor within the *j*-th instant slice of a multifactor portrait of currently investigated subject of the complex support;

G[*i*].*F*[*j*] is a level of presence (in a normalized form of representation) of the *j*-th impact factor within the obtained multifactor portrait of the *i*-th investigated subject of the complex support, represented by the corresponding relevant *i*-th unique gradient;

InflFact[1..*p*][1..*q*] is a set of pre-agreed and declared impact factors, affecting the perception subjectivization of the supported object by relevant subjects, who provide and perform a complex support of/for this object;

PsMfEnvExt is a variable describing the expanded version (option) of mathematical representation of the model

DOI 10.15588/1607-3274-2025-2-19

of researched polysubject multifactor environment of complex support;

PsMfEnvGrad is a variable describing the matrix variant of interpretation of the gradient form of representation of the model of researched polysubject multifactor environment of the complex support;

PsMfEnvSimp is a variable describing the simplified version (option) of mathematical representation of the model of researched polysubject multifactor environment of complex support;

Subj[*i*]*PortInst*[*j*] is an *j*-th instant slice of a multifactor portrait of the *i*-th investigated subject of support;

Subj[*i*]*PortFull* is a full multifactor portrait of the *i*-th investigated subject of the complex support, formed on the basis of its corresponding instant slices;

SubjPercOfObj[*j*] is an *j*-th subjectivized characteristic of subjective perception of the supported object by a specific subject which provides and performs its complex support;

SupObjChars[*i*] is an *i*-th objective characteristic of the investigated object of complex support.

INTRODUCTION

The nomenclature and assortment of software complexes, as well as a wide variety of software in general, continues growing steadily, which is due to the global digitalization and informatization of an extremely huge amount of objects, processes, entities and phenomena. Thus, both the total amount of existing and developed software and its general complexity are increasing, as well as extremely high requirements for its competitiveness in the global information technology (IT) market. Accordingly, there is a need for a complex support of all these (both existing and those in-process of development) software complexes, the requirements for which (just as in case of the software itself) are constantly growing, because clients, customers and investors, as well as, directly, the development companies of the relevant software complexes, are interested in providing the highest quality, effective, efficient and competitive, complex and comprehensive, support for all these software products.

Complex support of software products is one of the most important stages of the software's life cycle, while the automation of this support is an actual scientific and applied problem, which includes a huge range of related scientific applied and practical applied tasks. At the same time, the main driving force behind the implementation of the complex support of software products – are, undoubtedly, the specialists and professionals of various fields: ranging from a software developers, to customer service specialists, and real client's users of these supported software products. All these specialists are the subjects of a complex support of software products. Besides, all of them, in one way, or another, are influenced by both internal and external impact factors, which, in different ways and different extents, affect their subjective perception of both the supported software complex itself as well as the processes related to its complex support. And also, all these subjects, as a single whole entity, form the corresponding researched polysubject multifactor environment of the complex support of software product. Thus, there is a need to research and investigate such environments, formalize their representation, and also ensure the possibilities of their reproducing and modeling.

The object of study are the processes of complex support of software products, the processes of automation of this support, the processes of influence of impact factors onto the object and subjects of the complex support of software products, as well as the processes of the perception's subjectivization of the supported object by relevant subjects interacting with it.

The subject of study are methods and means of artificial neural networks, in particular a multilayer perceptron, as well as a computer design and modeling.

The purpose of the work is a development of a method for building models of a polysubject multifactor environment of the complex support of software products, that take into account the presence and the level of influence of relevant existing impact factors onto the subjects, which (those subjects) are interacting with the supported software complexes, and which directly provide and perform the complex support for the "in focus" software products, and form the relevant researched complex support's environments.

1 PROBLEM STATEMENT

Let's consider the formalization of the researched problem of building models of a polysubject multifactor environment of the complex support of software products in the form of problem of a nonlinear poly-criteria dependence. In this case, the input variables of the problem are the objective characteristics of the supported software complex (or processes related to its complex support) as a direct object of support: *SupObjChars*[*i*] (*i*=[1..*n*]), where *n* – the number of characteristics of this object of support.

While the initial variables of this task are the subjective characteristics of perception of the same object of support by the subjects who directly provide and perform its complex support: *SubjPercOfObj*[*j*] (*j*=[1..*m*]), where *m* – the number of characteristics of the subjectivized perception of the object of support by a specific subject(s) which provides and performs its complex support.

Let there be an existing set of a pre-agreed and predeclared impact factors affecting the perception's subjectivization of the support object by the subjects, which provide and perform its complex support, thus forming a non-linear polycriteria dependence in the form like below:

$$SubjPercOfObj[1..m]= =InflFact[1..p][1..q](SupObjChars[1..n]), \quad (1)$$

The main necessary criterion of the researched problem is the finiteness of this set of pre-agreed and declared impact factors, that ensures the stability of the transformation function over the entire period of the research.

DOI 10.15588/1607-3274-2025-2-19

Limitation(s) of the problem:

1. The values of the input characteristics of the investigated object of complex support *SupObjChars*[1..*n*] must be represented in the format of real numbers and necessarily in a normalized form (that is, in the range of values between 0.0 and 1.0): *SupObjChars*[1..*n*] ∈ [0..1].

Expression (1) provides the possibility of interpreting the given problem with help of a multilayer perceptron, where *SupObjChars*[1..*n*] – is interpreted by neurons of the input layer of MP ANN; *InflFact*[1..*p*][1..*q*] – is interpreted by neurons of the hidden layers of MP ANN; *SubjPercOfObj*[1..*m*] – is interpreted by neurons of the output layer of MP ANN.

However, such interpretation requires additional training and further testing of a multilayer perceptron, as well as formation of a new quality data (based on the received information) which will provide the possibility of assessing the complex influence of impact factors onto the entire support environment in general, including each of the subjects forming this environment as a single whole.

Thus raises a relevant scientific and applied task of forming and modeling a polysubject multifactor environment of the complex support of software products, in order to take into account the influence of various impact factors affecting the supported software complex itself, the processes of its complex support, as well as the subjects, who directly provide and perform this complex support, and the processes of subjectivization of their perception of the supported software complex or the processes of its complex support. In order to resolve scientific and applied problem a suitable method for building models of a polysubject multifactor environment of the complex support of software products has been developed.

The purpose of this paper is to highlight the developed method for building models of a polysubject multifactor environment of the complex support of software products, as well as the corresponding developed algorithm for building these models, which both and together provides possibilities to solve the scientific and applied problem of forming and modeling a polysubject multifactor environment(s) of the complex support software products, in order to take into account the influence of impact factors onto the objects, the processes, and the subjects of the complex support of software products.

2 REVIEW OF THE LITERATURE

The analysis of existing researches and publications was carried out in two main directions, namely: in the direction of automating the complex support of software products, as well as in the direction of the application of artificial intelligence technologies (in particular a multilayer perceptron type of artificial neural networks) in the processes of the complex support of software product.

One of the basic options for ensuring the mandatory requirements for the functionality, reliability, and competitiveness of software products is, actually, the automation of the complex support of these software products, which includes, in particular, such key points as: testing automation; DevOps automation; automation by means of knowledge bases and chatbots; automation of processing of a customer/user requests. Each of these basic directions in its own way investigates and improves the support of software products, while all of them together provide a comprehensive (complex) support for these products. At the same time, artificial (or computational) intelligence and/or machine learning technologies are increasingly becoming effective mechanisms for implementing this automation. All the information, obtained by the results of processing of each of the directions, is presented below.

In particular, authors of research [1] carried out an extended review of the current state of general research and implementation in the field of software testing with usage of machine learning, in order to identify and classify (by classification categories, by clustering and by anomaly detection) existing approaches, methodologies and tools of such use, and their application in various types of testing, such as test creation, test execution and defect prediction. The research [2] represents introductory information on the application of artificial intelligence technologies to improve the quality of implemented software, including the way by analyzing any anomalies in the behavior of the researched software complexes and systems. For example, authors of another research [3] investigate the issue of improving test automation using guided machine learning, namely the "The K nearest neighbors" method for setting priorities and configuring testing scenarios and relevant test cases. While, research [4] investigates the end-to-end automation of DevOps, emphasizing its potential for optimizing the entire life cycle of software: from the commit of correction code into the appropriate version control system's repository, and up to deployment in the client's main operational environment(s). In addition, author of research [5] conducts a generalized review of the application of DevOps best practices to improve the entire life cycle of software complexes, both for systems that require a high level of automation and support, and for all others, which are, actually, not characterized by the presence of highly complex processes inside of their operational routines. The research [6] examines the advantage of DevOps compared to Agile, as well as their combination, since, according to the author, Agile and DevOps function as complements, and also author emphasizes that for obtaining a better effect DevOps should be adopted rather as an organizational culture then just a technique, which allows to increase the overall efficiency of operational process and reduce costs due to cooperation improvement and tasks automation. In turn, author of research [7] reviewed the existing methods, tools, and applications for building knowledge bases, while, another author of the following considered research [8] investigated the semantic parsing of natural language phrases and sentences in such a manner that they were compatible with the used sources of structured data, such as: ConceptNet, Quasimodo, ATOMIC, and ontology like a WordNet, by extracting meaning from unstructured texts and representing it in the form of knowledge graphs, which are transformed into the first-order logical formulas that can be further used in order to answer user queries. At the same time, authors of

DOI 10.15588/1607-3274-2025-2-19

research [9] conducted a systematic study (in the context of a human-machine interaction approach) of how users interact with text chatbots, describing how users (alive people) perceive a chatbots within categories of satisfaction, involvement and trust, how and why they accept and use this technology, the issue of emotional involvement of such an interaction, as well as the issue of the disadvantages of such type of interaction. In addition, work [10] represents the results of the authors' research on the issue of automation of IT-incident's forecasting (these incidents are, in fact, the appeals of real users of software complexes, that come to the relevant customer support services of these complexes) and solving them in the shortest time, by using an artificial intelligence models. In addition, authors of research [11] proposed a neural network, based on a convolutional neural network, for the automated classification of customer support service tickets of software complexes, which provides possibilities to solve such problems of traditional ticket classifiers as sparsity, nonlinearity, overfitting and manual work of functions.

Thus, based on performed analysis of existing researches, the considerable influence of artificial intelligence and machine learning technologies, used precisely for the purpose of automating the component processes of the complex support of software products, is confirmed. However, at the same time, we also state the lack of consideration of the peculiarities of the subjects interacting with the processes of the complex support of software products, as well as various (both internal and external) impact factors which affect the supported software complex itself, the processes of its support, as well as the subjects of this support. After all, in fact, impact factors form the very environment inside of which each particular supported software complex is constantly located and exists. So, this is the way how a corresponding actual scientific-applied task of researching this polysubject multifactor environment (of the complex support of software products) arises. And one of the effective tools for its resolution is, actually, the development of an appropriate relevant model which provides possibilities for describing, researching and modeling such environments, and, accordingly, taking into account the effect of various impact factors that significantly affect the object, processes, and subjects of the complex support of software products.

3 MATERIALS AND METHODS

One of the most important components of development the method, highlighted in current one research, is the preliminary development of the algorithm for building a model(s) of a polysubject multifactor environment(s) of the complex support of software products.

The general step-by-step mechanism for building a model of a polysubject multifactor environment of the complex support of software products involves the following mandatory stages, which, actually, form the basis of the developed algorithm.

At the first stage, the construction of a generalized support model of the software complex's support is carried out, followed by the encapsulation of an artificial neural network (ANN) of the multilayer perceptron (BP) type, as well as the implementation of the corresponding training of this encapsulated ANN BP based on a previously prepared dataset(s).

At the second stage, the levels of presence of each of the impact factors (from the pre-declared set of impact factors) in the corresponding neurons of the hidden layers of the encapsulated and trained BP ANN are calculated.

At the third stage, personal multifactor portraits of the subjects of support, formed on the basis of the corresponding instant slices of these portraits, are built.

At the fourth (and the final) stage, the construction of the model of a polysubject multifactor environment of the complex support of studied software product is implemented, based on the subjects' personal multifactor portraits obtained at the previous stage, which, actually at the same time, form the researched support environment.

Thus, the implementation of these presented above four basic stages of the developed algorithm makes it possible to obtain (at the output) a corresponding model of the researched polysubject multifactor environment of the complex support of corresponding software product.

Figure 1 below represents the block diagram visualization of the developed algorithm, with a detailed representation of each of the aforementioned stages.

The developed algorithm provides both: options for representing the model of a polysubject multifactor environment of the complex support (from among two possible developed options: simplified or expanded), as well as the forms of representing the model (from among possible developed forms, such as: mathematical, linguistic and gradient). Each of these possible fluctuations of options and forms is described in more detail in the following paragraphs of current section of this research.

The next fundamental component of the developed method is, in fact, the development of a model of polysubject multifactor environment for supporting software complexes. The main feature of the developed model is that it requires preliminary decomposition into simpler components, such as:

– at the first (external) decomposition level – personal (individual) multifactor portraits of each of the subjects interacting with the object of support (the supported software complex itself, or the processes of its support);

– at the second (internal) decomposition level – instant slices of these multifactor portraits of each of the subjects interacting with the object of support.

Accordingly, in order to ensure the possibility of building a correct and complete model of a polysubject multifactor environment for supporting any software complex, it is necessary and mandatory to fully implement its two levels of decomposition, starting from the most detailed (i.e., in this case – aforementioned internal, or second, decomposition level), and ending with the least detailed (i.e., in this case – aforementioned external, or first, decomposition level).

Using the appropriate components, such as, in particular: the mathematical model presented in research [12], as well as the model of decomposed insulating dominance

DOI 10.15588/1607-3274-2025-2-19

presented in research [13], together with the information model described in research [14], it becomes possible to obtain the second (internal) level of decomposition of the developed model of a polysubject multifactor environment of the complex support of researched supported object by modeling one specific case of perception's subjectivization of this object by the relevant researched subject interacting with it.

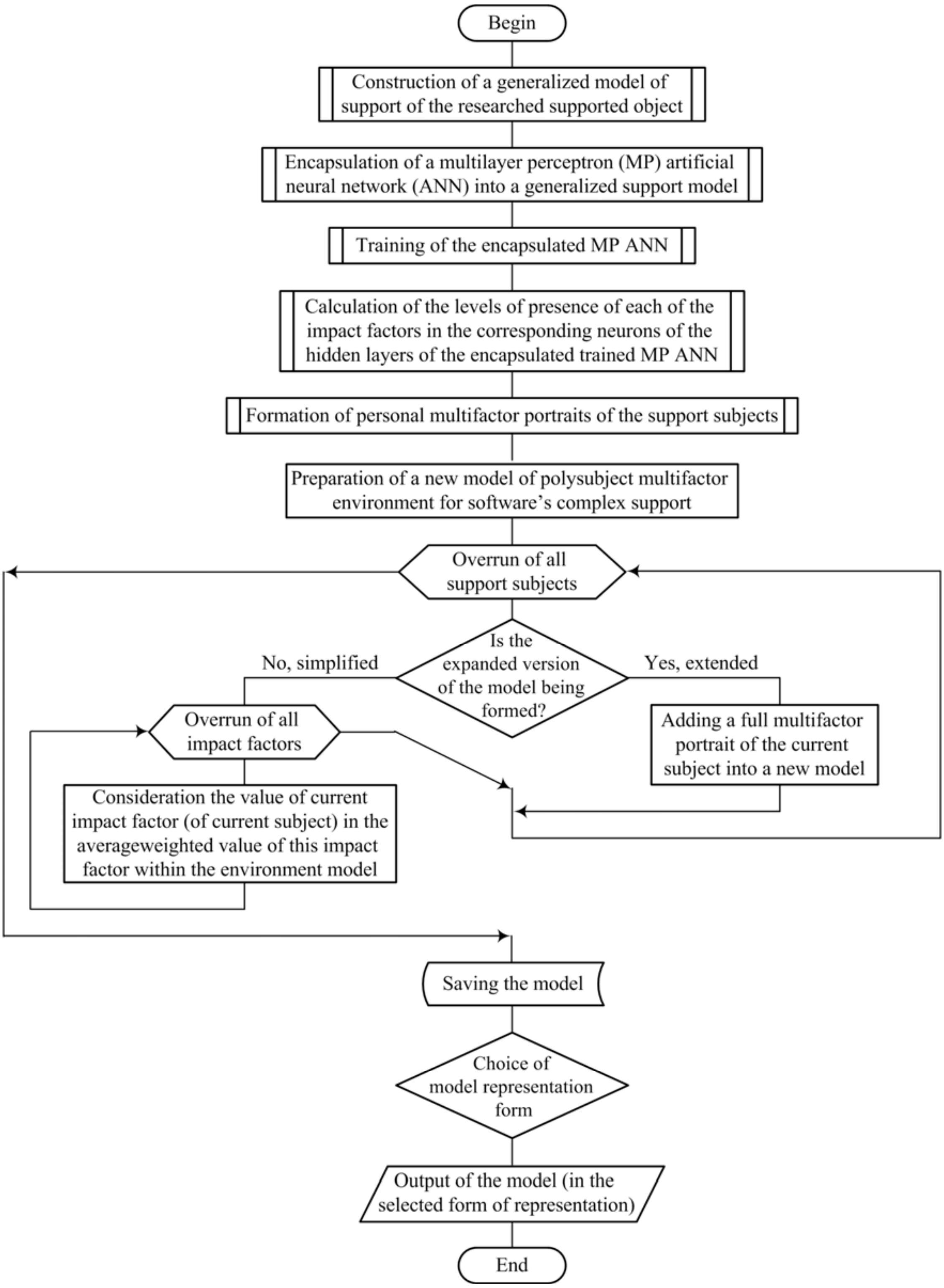

Figure 1 – Block diagram of the developed algorithm for building a model of polysubject multifactor environment of the complex support of software products

DOI 10.15588/1607-3274-2025-2-19

In turn, the set of modeling results of all available cases of perception subjectivization of the supported object by the corresponding researched subject of interaction – provides all the necessary data for building a personal (individual) multifactor portrait of this specific researched subject, which represents the first (or external) level of decomposition of the developed model of a polysubject multifactor environment of researched object's support.

Few variants of representation forms of the model of a polysubject multifactor environment of the complex support of software products have been developed and proposed, in particular, such as: mathematical form of representation; linguistic form of representation; and gradient form of representation.

At the same time, the developed model of a polysubject multifactor environment of the complex support of software products, regardless of its form of presentation, can exist in scope of two possible options of its presenting: a simplified option (in which only the average values of the presence levels of each of the impact factors in the researched environment are represented), as well as expanded option (in which there are separate values of the presence levels of each of the impact factors for each of the subjects which form the researched environment).

The mathematical form of representation of the model of a polysubject multifactor environment of the complex support of software products contains the following components, which are described in detail in the text below.

The basic fundamental unit of measurement for the future representation of any researched polysubject multifactor environment – is an instant slice of the multifactor portrait of each specific individual researched subject, represented by the following single expression:

$$Subj[i]PortInst[j](i \in 1..n)=[F1[j],\ F2[j],\ \ldots\ Fm[j]], \quad (2)$$

where $F[k](k \in 1..m)[j]$ – the level of presence of the impact factor $F[k]$ inside the instant slice of a multifactor portrait $PortInst[j]$ of the investigated subject $Subj[i]$; n – the number of subjects forming the researched polysubject multifactor environment of complex support; m – the number of pre-agreed and declared impact factors within researched environment of complex support.

The next important component of constructing of any researched polysubject multifactor environment – is a complete multifactor portrait of the researched subject(s), formed on the basis of its instant slices, and represented by the following expression:

$$Subj[i]PortFull(i \in 1..n)=[\sum(F1[1..o])\ /\ o,\ \sum(F2[1..o])\ /\ o,\ \ldots\ \sum(Fm[1..o])\ /\ o], \quad (3)$$

where $\sum(F[j][1..o])/o$ $(j \in 1..m)$ – the average arithmetic value of the level of presence of a specific (currently considered) impact factor $F[j]$ inside the multifactor portrait of current researched subject $Subj[i]PortFull$, formed on the basis of its instant slices (with total amount of slices = o); m – the number of pre-agreed and declared impact factors within researched environment of complex support; o – the number (amount) of instant slices of the currently researched multifactor portrait.

Accordingly, on the basis of formed full multifactor portraits of subjects (which, in fact, directly form the researched environment of the complex support of software product), the simplified option of mathematical representation of the polysubject multifactor environment's model of complex support is described by following expression:

$$PsMfEnvSimp=[\sum(Subj[1..p]PortFull[F1])/p,\ \sum(Subj[1..p]PortFull[F2])/p,\ldots\ \ldots\sum(Subj[1..p]PortFull[Fm])/p], \quad (4)$$

where $\sum(Subj[1..p]PortFull[F[j]])/p$ $(j \in 1..m)$ – the average arithmetic value of the level of presence of a specific current impact factor $F[j]$ in the current model of the researched polysubject multifactor environment of the complex support, formed on the basis of values of presence levels of this impact factor in the available full multifactor portraits of all subjects $Subj[1..p]PortFull$, which (subject) form this specific support environment; p – the number of full multifactor portraits of subjects which form the researched support environment; m – the number of pre-agreed and declared impact factors within researched environment of complex support.

Meanwhile, the expanded option of mathematical representation of the polysubject multifactor environment model of complex support is described by the following expression:

$$PsMfEnvExt=[Subj[i]PortFull(i \in 1..n)], \quad (5)$$

where $Subj[i]PortFull(i \in 1..n)$ – are full multifactor portraits of all subjects $Subj[1..p]PortFull$, which form this specific support environment; n, p – the number of subjects, which form this specific support environment.

In other words, the expanded option of mathematical representation of the polysubject multifactor environment's model of complex support – is nothing more than just an array of pre-formed full multifactor portraits of the subjects which, in fact, form this support environment.

The main advantage of the developed and described mathematical form of representation of the polysubject multifactor environment's model of complex support – is basically its universality and uniformity, which makes it possible to use it also beyond the limits of the only particular area of software complexes' support, or even the field of information technologies as a whole.

The *linguistic form* is the next one developed and introduced form of representation of the polysubject multifactor environment model of complex support. This representation form involves the partial or full use of any existing, modified, or completely new linguistic constructions, to describe a polysubject multifactor environment(s). One of such existing variants of linguistic constructions is, in particular, XML (EXtensible Markup Language), since XML still remains as one of the most widely used and popular markup languages for representing a wide variety of objects and structures regardless of their degree of

DOI 10.15588/1607-3274-2025-2-19

complexity, so it could be used both in the classic form of representation and as a basis for various modifications, including the considered linguistic form of a polysubject multifactor environment's model representation.

Below is a variant of the linguistic representation of an instant slice of the multifactor portrait of investigated subject interacting with the supported object:

```
<Subj>
  <PortInst>
    <F1>..</F1>
    <F2>..</F2>
      ...                                  (6)
    <Fm>..</Fm>
  </PortInst>
</Subj>
```

At the same time, the linguistic form of representation of the full multifactor portrait of the investigated subject is described by the following expression:

```
<Subj>
  <PortFull>
    <F1>∑(<PortInst>.<F1>[1..o])/o</F1>
    <F2>∑(<PortInst>.<F2>[1..o])/o</F2>
      ...                                  (7)
    <Fm>∑(<PortInst>.<Fm>[1..o])/o</Fm>
  </PortFull>
</Subj>
```

Accordingly, the linguistic representation of a simplified version (option) of the polysubject multifactor support environment's model will have the following form:

```
<PsMfEnvSimp>
  <F1>∑(<Subj>[1..p].<PortFull>.<F1>)/p</F1>
  <F2>∑(<Subj>[1..p].<PortFull>.<F2>)/p</F2>
    ...                                    (8)
  <Fm>∑(<Subj>[1..p].<PortFull>.<Fm>)/p</Fm>
</PsMfEnvSimp>
```

While, the linguistic representation of the expanded version (option) of the model of a polysubject multifactor support environment will, accordingly, look like below:

```
<PsMfEnvExt>
  <Subj>
    <PortFull>
      <F1>∑(<PortInst>.<F1>[1..o])/o</F1>
      <F2>∑(<PortInst>.<F2>[1..o])/o</F2>
        ...                                (9)
      <Fm>∑(<PortInst>.<Fm>[1..o])/o</Fm>
    </PortFull>
  </Subj>
</PsMfEnvExt>
```

The main advantage of the developed and proposed linguistic form of representing of the model of a polysubject multifactor support environment is its adaptability for use precisely within the framework of almost any possible further software implementation and/or computer simulation and modeling.

The *gradient form* is the next one developed and introduced form of representation of the models of a polysubject multifactor environment of the complex support of software products. The main idea of this form of representation is the most concise and rational numerical representation of the researched model of the complex support environment with the simultaneous selection of data by each of the subjects forming this environment.

At the same time, the data of each individual subject are represented by the range of values of its (subjects's) personal gradient, which can act as a separately taken unique segment of the numerical range of values, for example: (10.00–20.00) – the gradient of subject 1, 20.00–30.00 – the gradient of subject 2, 30.00–40.00 – the gradient of subject 3, and so on; as well as, for example, a unique combination of symbols, for example: AA – the gradient of subject 1, AB – the gradient of subject 2, AC – the gradient of subject 3, and so on.

However, the numerical values (themselves) of the levels of presence of each of the impact factors inside the formed multifactor portraits of the subjects must remain in a single common coordinate system, and also, necessarily, in a single normalized form of representation (the last one means they must be represented by real numbers in the value range [0.00 – 1.00]).

Thus, each gradient, representing a multifactor portrait of a separate subject, receives its own personal range of saturation values for each of the impact factors, however (necessarily) within the framework of a common evaluation system (coordinates) as well as a common normalized scale for comparison of data of all other gradients.

Two variants of interpretation of the gradient form of representation (of the models of polysubject multifactor support environments) have been developed and proposed in this research: the matrix variant of interpretation and the graphic variant of interpretation.

In particular, the matrix variant of interpretation of the gradient form of representation (of the models of polysubject multifactor support environments) is described using the following expression:

$$\begin{aligned} PsMfEnvGrad=(&G[1].F[1], G[1].F[2], \dots G[1].F[f],\\ &G[2].F[1], G[2].F[2], \dots G[2].F[f],\\ &G[..].F[1], G[..].F[2], \dots G[..].F[f],\\ &G[s].F[1], G[s].F[2], \dots G[s].F[f]), \end{aligned} \quad (10)$$

where $G[i].F[j]$ ($i€1..s$, $j€1..f$) – the level of presence (in a normalized form of representation) of the impact factor $F[j]$ within the formed multifactor portrait of the subject $Subj[i]$, represented by its unique gradient $G[i]$; f – the number of declared available impact factors; s – the number of subjects forming the researched polysubject multifactor support environment.

While, the graphical variant of interpretation of the gradient form of representation (of the models of polysubject multifactor support environments) can be demonstrated with the help of the Figure 2 below, at the same time, all the notations used in this figure, are absolutely the same as in the expression (10) provided above.

DOI 10.15588/1607-3274-2025-2-19

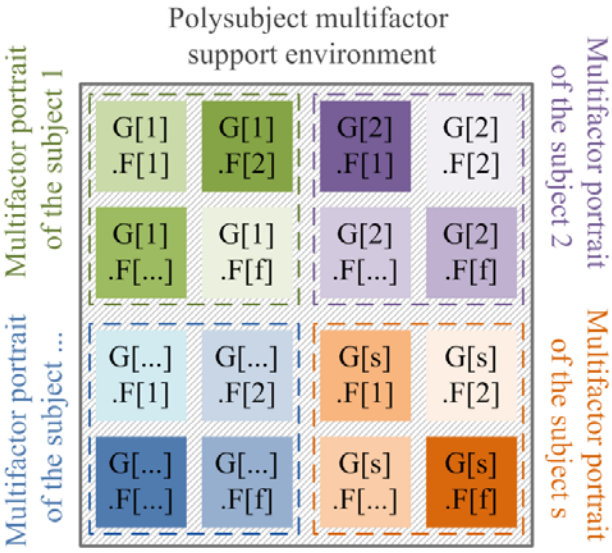

Figure 2 – Demonstration of a graphical variant of interpretation of the gradient form of the representation of a polysubject multifactor support environment's model.

The main idea of the developed and proposed graphical variant of interpretation of the gradient form of representation the support environment's model is, first of all, its simplicity of visual perception, as well as the ability to "combine" subjects in a different (and/or needed) ways/manner in order to form and obtain the necessary structure of the researched environment itself.

The gradient of each subject is indicated (in the proposed graphic variant) with a corresponding unique color, while the level of presence of each impact factor within formed multifactor portrait of this subject is represented with a gradient shades of this color. Hue saturation represents the normalized value of the presence fraction: the more saturated the hue – the closer this value is to 1, and the less saturated the hue – the closer this value is to 0.

The main advantage of the developed and described gradient form of representation of the model of a polysubject multifactor environment for supporting software complexes is its rationality, brevity, as well as dynamism and accessibility of perception, provided, among other things, by the developed and proposed graphical variant of interpretation of this form.

4 EXPERIMENTS

The experiment consists in the step-by-step execution of all the stages described in this research, namely:

– selection of an object of complex support;
– definition of the subjects of support;
– determination of existing impact factors;
– determination of the objective characteristics of the supported object;
– determination of the subjective characteristics of the perception of the supported object by relevant subjects;
– design and encapsulation of the appropriate relevant MP ANN;
– preparation of the datasets for training and testing the encapsulated MP ANN;
– formation of the instant slices of a multifactor portraits of the subjects of researched support environment;
– formation of a full multifactor portraits of the subjects of researched support environment (based on their instant slices);
– development of a model of the researched polysubject multifactor support environment based on obtained full multifactor portraits of the subjects forming this environment;
– presentation of the obtained model in an arbitrary form from among the developed, proposed and described forms of representation.

5 RESULTS

The main results of functioning of the developed method – are the appropriate models of the researched polysubject multifactor support environments for various supported software complexes. The obtained models provide opportunities both for the representation of the investigated polysubject multifactor support environment in a convenient form of representation (from a set of developed and proposed ones), as well as for any further research of such important components of the complex support of software products as: the impact of the researched support environment onto the supported object itself (directly the supported software complex itself, or the processes related to its support); the impact of individual subjects onto the researched support environment; as well as the influence of individual impact factors onto the researched support environment through prior influence onto the subjects forming the same environment.

In addition, the models of the researched polysubject multifactor support environments, obtained using the developed method, make it possible to get specific numerical values of the levels of presence of each of the impact factors within the specific researched support environment, and, therefore – the detection of any disproportions, anomalies, regularities, features, or any other characteris-

DOI 10.15588/1607-3274-2025-2-19

tics of the polysubject multifactor structure of the researched environment, in order to provide opportunities for further correction(s) of this environment, in such a manner to improve the complex support of the "in focus" software products in automated mode. In addition, the presence of the developed and proposed simplified and expanded options of representation of the obtained models ensures the variability of the detalization degree of corresponding researched polysubject multifactor structure of the simulated support environment(s). At the same time, the expanded option of representation also provides the possibility of additional clustering of the structure of researched support environment both by the criteria of support subjects and by the criteria of impact factors.

Figure 3 below presents an example of visualization of both simplified and expanded (extended) options of representation of a polysubject multifactor support environments based on relevant multifactor portraits of the subjects, which actually form these environments.

At the same time, depending on the obtained results of the complex representation of the researched polysubject multifactor support environment (obtained on the basis of processing the corresponding multifactor portraits of the subjects which form this environment) various options for the classification of such environments are possible.

In particular, the following options for the classification of support environments (of the complex support of software products) have been developed and proposed based on the level of their balance:

– perfectly balanced environments;
– well-balanced environments;
– satisfactorily balanced environments;
– non-satisfactorily balanced environments.

The main criterion for classifying the support environment(s) into one of these categories – is the average deviation (in percentages) of the indicator value of each of the impact factors, from the arithmetic mean value of this indicator across all impact factors.

Depending on specific tasks, environments, necessities or features, the specific numerical values of the thresholds of each of the categories of the developed classification may change and shift in one direction or another. As an example, as well as on the basis of conducted researches, the following values of the thresholds of each of the categories are proposed, in accordance to which:

– perfectly balanced environments – are those in which the average deviation (in percentages, rounded to the nearest integer according to generally accepted mathematical rounding rules) of the indicator value of each of the impact factors, from the arithmetic mean value of this indicator for all impact factors, varies within 1–2%;

– well-balanced environments – are those in which the average deviation (in %) of the indicator value of each of the impact factors, from the arithmetic mean value of this indicator for all impact factors, varies within 3–5%;

– satisfactorily balanced environments – are those in which the average deviation (in %) of the indicator value of each of the impact factors, from the arithmetic mean value, varies within 6–10%;

– non-satisfactorily balanced environments – are those in which the average deviation (in %) exceeds 10%.

Figure 4 below provides a visualization of examples of each of the categories of the developed classification of software product support's environments.

Thus, the developed classification provides the possibility of a flexible dynamic system of evaluation and determination of the balance category of any researched support environment of the relevant software products.

In addition, the developed method provides the possibility of studying the balance of any specific area of the researched support environment formed by the relevant subjects, as well as combining the activities of the subjects of this support environment in such a way to achieve a local balance improvement(s) within this specific area of the investigated support environment.

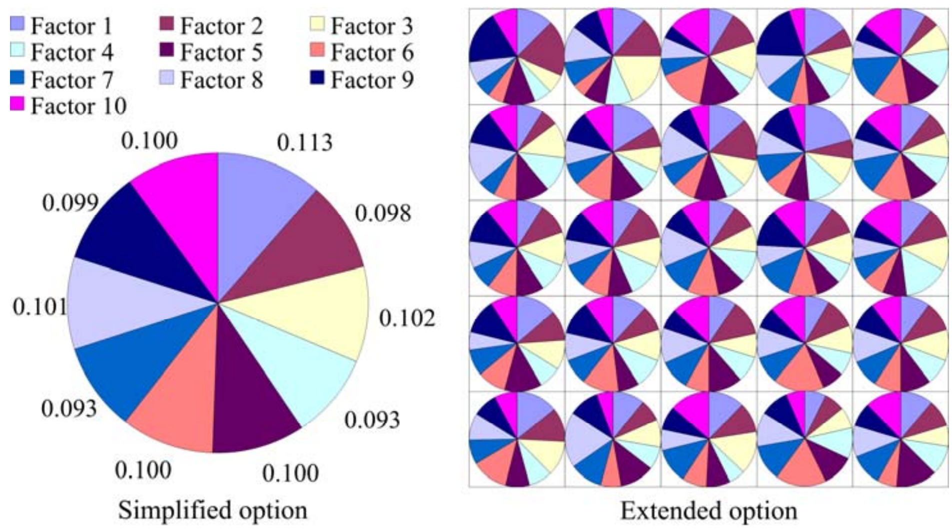

Figure 3 – Example of visualization of simplified and expanded (extended) options of representation of a polysubject multifactor support environments based on relevant multifactor portraits of the subjects which form this environment

DOI 10.15588/1607-3274-2025-2-19

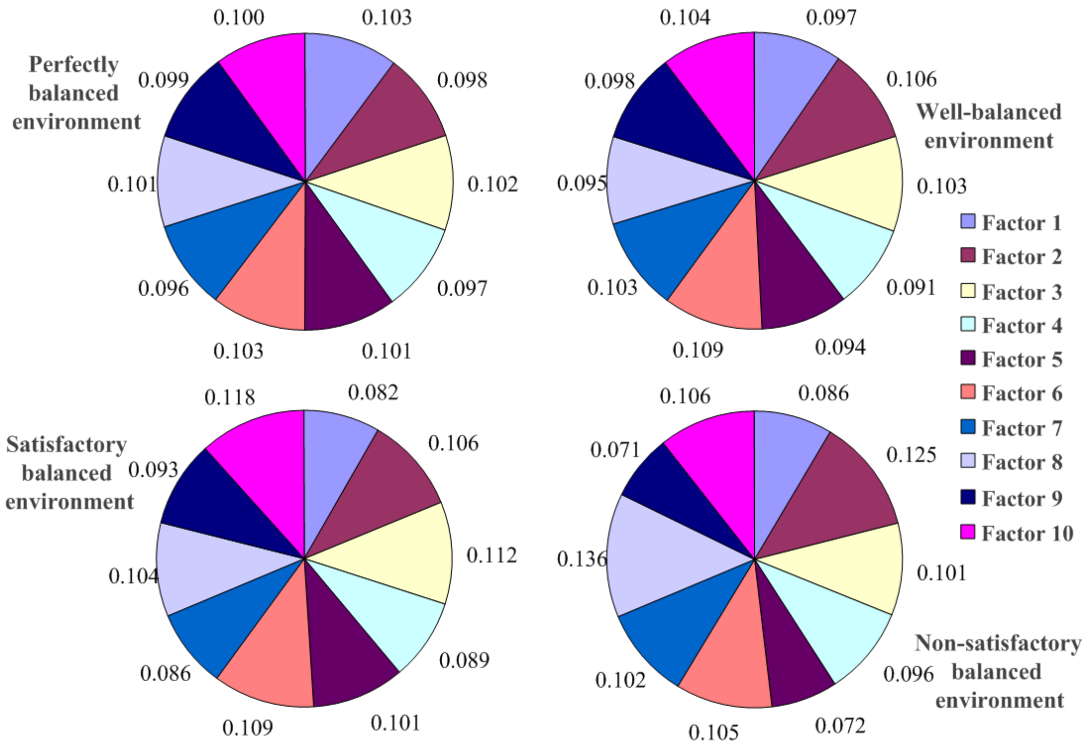

Figure 4 – Visualization examples of each of the categories of the developed classification of software product support's environments

In addition, as an example of practical application and approbation, the developed method has been used, in particular, to solve the applied practical task of determining the dominant and the deficient impact factors of a polysubject multifactor environment of the complex support of researched software product.

The resolution of the given applied practical task comes down to the application of the developed (and presented in this research) method of building models of a polysubject multifactor environment for the complex support of software products.

The formation of the model of the entire polysubject multifactor environment takes place on the basis of previously formed personal multifactor portraits of the subjects which form this support environment.

In turn, the formation of multifactor portraits of the subjects (of the support environment) is carried out on the basis of their instant slices, obtained by modeling the corresponding individual test cases of these subjects.

Figure 5 below presents the obtained results of modeling and formation of personal multifactor portraits of the subjects of the researched support environment.

While Figure 6 below presents the results of solving the given applied practical task of determining the dominant and the deficient impact factors of the polysubject multifactor environment of the complex support of the researched software product. In addition, the results of determining the balance category of the researched support environment are given, from which, by the way, another possible (alternative) way of solving the given applied practical task raises, since both the dominant and the deficient impact factors (of any researched support environment) will always have the largest indicators (in absolute equivalent) of the deviation of their value from the arithmetic mean value of all impact factors, while their polarity (that is, the real value, but not the absolute) will indicate their dominance or deficiency, which is confirmed by obtained results presented in the same Fig. 6.

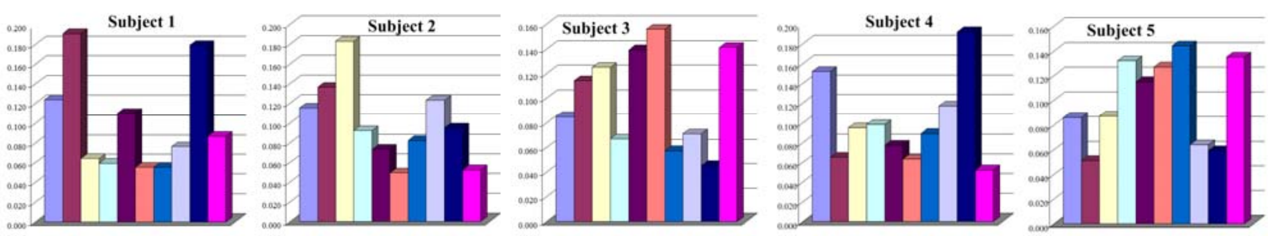

Figure 5 – Personal multifactor portraits of the subjects of the researched support environment

DOI 10.15588/1607-3274-2025-2-19

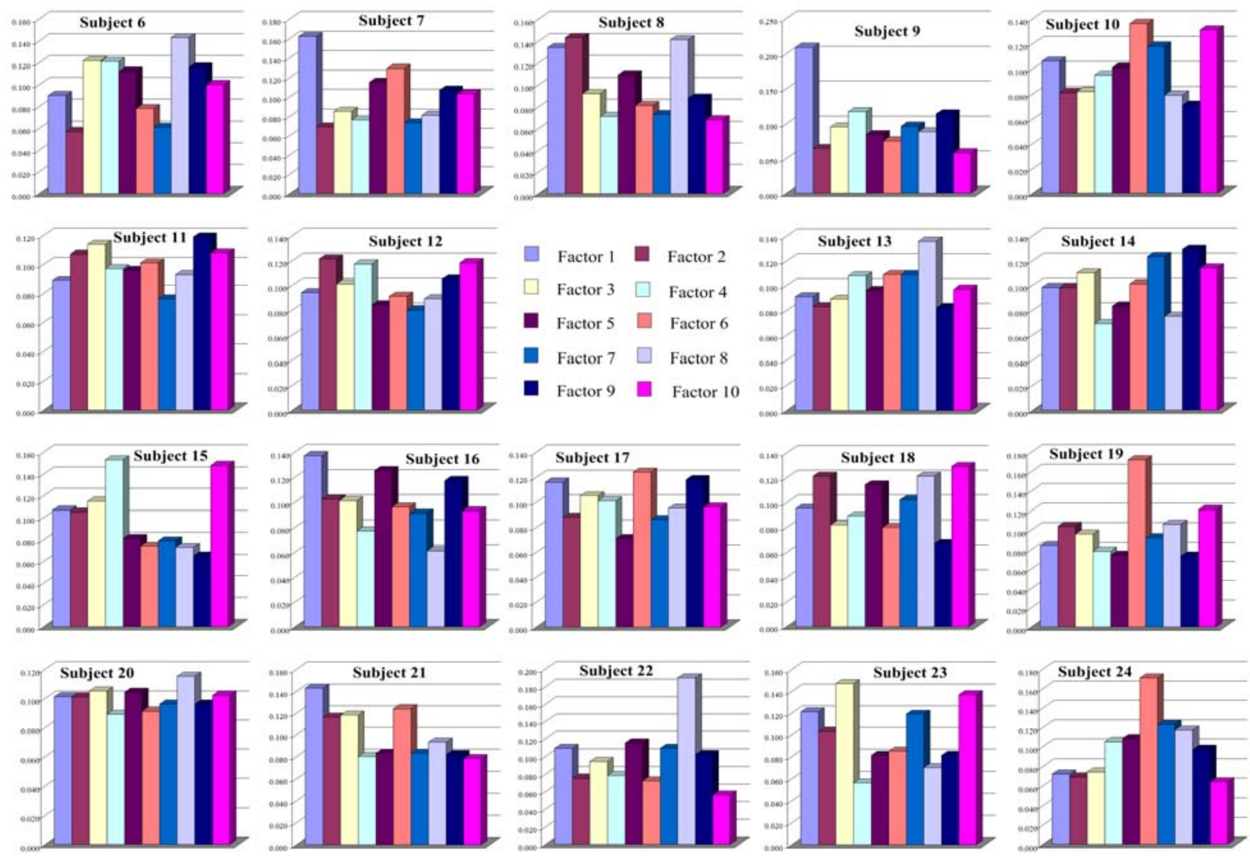

Figure 5 – Personal multifactor portraits of the subjects of the researched support environment (Continuation)

Thus, with the help of the developed method of building models of a polysubject multifactor environment for the complex support of software products, as an example of its practical application and approbation, the applied practical task of determining the dominant and the deficient impact factors of a polysubject multifactor support environment has been resolved for the specific software product. In addition, the accompanying task of identifying the balance level of the researched support environment was also resolved, which, in turn, provided the possibility of one more, additional, alternative solution to the same applied practical task.

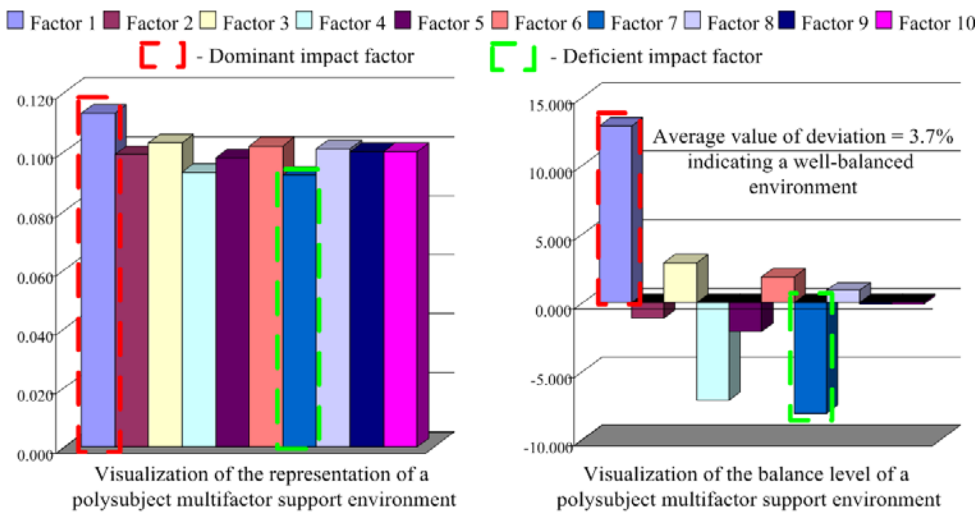

Figure 6 – Visualization of the solution of the given applied practical task

DOI 10.15588/1607-3274-2025-2-19

6 DISCUSSION

In work [15], the impact of entire procedures of Scrum and Agile technologies is studied as factors of influence onto the project management support related to the administration of development and support of software products, which takes into account, in particular, the twelve basic principles of the Agile methodology, as well as five key characteristics and three fundamental principles of the Scrum methodology, thereby demonstrating that not only external or internal separate impact factors can act as factors of influence onto the complex support of software products, but methodologies can undoubtedly be such factors as well, since each of them contains a clear list of principles that make their direct adjustments to the processes of development and complex support of software products. Another work [16] examines the influence of automated and manual testing factors onto the relevant indicators of the efficiency and effectiveness of testing the supported software product(s), where the authors take into account relevant impact factors, and as a result, it was established that only some hybrid variant of testing (which combines both manual and automated testing) makes it possible to take into account the complex influence of the impact factors of both these categories of testing, and thus obtain improvements in all indicators of efficiency and effectiveness studied by the authors. While in work [17], artificial intelligence technologies are considered as factors of influence on supported software products and their automation, and the analysis of their influence onto the automation of digital software products is carried out on example of a particular field of medical services (which is one of the most actual and profitable niches of software development nowadays) as well as the main factors affecting the implementation of artificial intelligence in systems of this class, were investigated.

Thus, in all considered cases, the relevance and significance of factors affecting supported software products, their support processes, and automation, have been confirmed. At the same time, unfortunately, the issue of a complex study of these impact factors within the framework of a single common environment of their existence and functioning remains unsolved.

At the same time, the method developed and presented in this one current research fully discloses this issue, and makes it possible to model and explore polysubject multifactor environments of the complex support of software products as a single whole indivisible entity, which directly affects the object, the processes, and the subjects of such complex support.

As at the output of the developed method, we get a model of researched polysubject multifactor environment of software's complex support, represented in a convenient form (from among those developed and proposed here in this research), which fully represents the researched environment.

As a further application of the developed method, we see the potential of its use for solving a stack of applied practical problems and tasks related to the research of a polysubject multifactor environments of the complex support of software products, which could be various investigated teams, collectives, divisions, departments, companies, or any other agglomerations of subjects interacting with the investigated object(s) of complex support. However, the potential of the developed method is not limited only to the context of software products' support, but also extends to other areas of science and practice, in which the key elements are active subjects and factors influencing their interaction. Thus, taking into account a wide range of applied problems, the expediency of further research in this direction is fully justified.

CONCLUSIONS

The method of building the models of a polysubject multifactor environment of the complex support of software products has been developed. The main scientific and applied problem solved by the developed method is the problem of forming and modeling a polysubject multifactor environment(s) of the complex support of software product(s), in order to take into account the influence of various impact factors that affect the supported software complex itself, the processes of its complex support, as well as the subjects which directly form and implement this complex support, as well as the processes of subjectivization of their (subjects') perception of the supported software complex or processes of its support. Also, the algorithm for building the model(s) of a polysubject multifactor environment(s) of the complex support of software products has been developed. In addition, several representation forms of the developed models of a polysubject multifactor environments (of the complex support of software products) have been developed and proposed, in particular, such as: mathematical form of representation (which includes additionally developed expanded and simplified representation options); linguistic form of representation; gradient form of representation (which includes additionally developed matrix and graphic versions of interpretation); and the main advantages of each of the above forms of representation are given. In addition, options for the classification of software product support's environments (based on their balance level) have been developed, which includes such categories as: perfectly balanced environments; well-balanced environments; satisfactorily balanced environments; unsatisfactorily balanced environments. The main criterion for the balance classification of the environment is the deviation average value (in percentages) of the indicator of each of the impact factors, from the arithmetic mean value of this indicator for all impact factors. The developed method provides possibility to carry out research on a polysubject multifactor environments of the complex support of software products by developing relevant models, which allows to describe, represent, investigate, and model such environments, as well as, accordingly, take into account the impact of various factors that significantly affect the object, the processes, and the subjects of the complex support of software products.

The scientific novelty consists in the development of a method for building models of a polysubject multifactor

DOI 10.15588/1607-3274-2025-2-19

environment of the complex support of software products, which provides possibility to solve the scientific and applied problem of defining, forming, and modeling a poly-subject multifactor environment(s) of the complex support of software products, in order to take into account the influence of various impact factors affecting the supported software complex itself, the processes of its complex support, as well as the subjects which directly form, provide and implement this complex support, as well as the processes of subjectivization of their (subjects') perception of the supported software complex or its relevant support's processes.

The practical significance consists in: the developed algorithm for building a model(s) of the researched poly-subject multifactor environment(s) of the complex support of software products; in the development of forms of representation of this model (in particular, such as: mathematical form of representation; linguistic form of representation; gradient form of representation); as well as in the development of an appropriate specialized classification of complex support's environments of software products based on their balance level (which includes, in particular, such categories as: perfectly balanced environments; well-balanced environments; satisfactorily balanced environments; and non-satisfactorily balanced environments).

Prospects for further research consist in the development of appropriate additional specialized algorithmic and software dedicated for modeling the researched poly-subject multifactor environments of the complex support of software products, with the aim of automating the research of the influence of various existing impact factors performing influence on the object, the processes, and the subjects of the complex support of software products, as a component of more global scientific and applied problem of automation of the complex support of software products.

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.

REFERENCES

1. Panwar A., Peddi P. Implementation of Software Testing Using Machine Learning: A Systematic Mapping Study, *JJTU Journal of Renewable Energy Exchange*, 2023, Vol. 11, Iss. 7, pp. 58–64. – DOI: 10.58443/ijrex.11.7.2023.58-64
2. Gadani N.N. Artificial Intelligence: Leveraging Ai-Based Techniques For Software Quality, *International Research Journal of Modernization in Engineering Technology and Science,* 2024, Vol. 06, Iss. 07, pp. 757–769. – DOI: 10.56726/IRJMETS60018
3. Chaudhary J., Anand P. Predictive Modeling Of Automation In Software Testing A Machine Learning Approach For Efficient Test Case Selection [Electronic resource], *Advances in Mechanics,* 2023, Vol. 11, Iss. 2, pp. 128–139. Mode of access: https://www.advancesinmechanics.com/pdf/2023-128.pdf (date of access: 01.01.2025). – Title from screen.
4. Manchana R. The DevOps Automation Imperative: Enhancing Software Lifecycle Efficiency and Collaboration, *European Journal of Advances in Engineering and Technology*, 2021, Vol. 8, Iss. 7, pp. 100–112. DOI: 10.5281/zenodo.13789734
5. Ogala J. O. A Complete Guide to DevOps Best Practices, *International Journal of Computer Science and Information Security (IJCSIS)*, 2022, Vol. 20, No. 2, pp. 1–6. DOI: 10.5281/zenodo.6376787
6. Adnan S., Moin K., Imran J. DevOps with Agile: Best practices to improve software quality, *International Conference On Biological Research And Applied Science,* 2023, pp. 61–66. DOI: 10.37962/ibras/2023/61-66
7. Rajadhyax D. A Survey of Methods, Tools and Applications of Knowledge Base Construction (KBC) [Electronic resource], *Telecom Business Review: SIDTM Journal,* 2020, Vol. 13, Iss. 1, pp. 20–26. Mode of access: https://sidtm.edu.in/wp-content/uploads/2021/06/tbr2020.pdf#page=25 (date of access: 01.01.2025). – Title from screen.
8. Verrev M. Combining Semantic Parsing Frameworks for Automated Knowledge Base Construction [Electronic resource], *6th Workshop on Advances In Argumentation In Artificial Intelligence,* 2022, 11 pages. Mode of access: https://ceur-ws.org/Vol-3354/paper1.pdf (date of access: 01.01.2025). – Title from screen.
9. Rapp A., Curti L., Boldi A. The human side of human-chatbot interaction: A systematic literature review of ten years of research on text-based chatbots, *International Journal of Human-Computer Studies,* 2021, Vol. 151, Iss. 3, Article ID: 102630. DOI: 10.1016/j.ijhcs.2021.102630
10. Ahmed S., Singh M., Doherty B. et al. AI for Information Technology Operation (AIOps): A Review of IT Incident Risk Prediction, *9th International Conference on Soft Computing and Machine Intelligence (ISCMI)*, 2022, pp. 253–257. DOI: 10.1109/ISCMI56532.2022.10068482
11. Paramesh S. P., Shreedhara K.S. A deep learning based IT service desk ticket classifier using CNN [Electronic resource], *ICTACT journal on soft computing,* 2022, Vol. 13, Iss. 1, pp. 2805–2812. Mode of access: https://ictactjournals.in/paper/IJSC_Vol_13_Iss_1_Paper_9_2805_2812.pdf (acc. date: 02.01.2025). – Title from screen.
12. Pukach A.I. Mathematical model for analysis of influencing factors on software complexes support / A.I. Pukach, V.M. Teslyuk // Printing and Publishing. – 2024. – 1(87). – P. 75–85. – DOI: 10.32403/0554-4866-2024-1-87-75-85
13. Pukach A. I., Teslyuk V. M. Model of decomposed insulating dominance for the analysis of influencing factors of software complexes support automation, *Scientific Bulletin of UNFU,* 2024, 34(5), pp. 170–179. – DOI: 10.36930/40340521
14. Pukach A. I., Teslyuk V. M. Information model for automation of software complexes support influencing factors analysis with usage of the R-system and Python environments, *Bulletin of Lviv State University of Life Safety,* 2024, 29, pp. 54–64. – DOI: 10.32447/20784643.29.2024.06
15. Rasool N., Yousaf S., Haseeb U. et al. Scrum and the Agile procedure's impact on software project management, *Journal of Jilin University (Engineering and Technology Edition)*, 2023, Vol. 42, Iss. 2, pp. 380–392. – DOI: 10.17605/OSF.IO/MQW9P
16. Thant K. S., Tin H. H. K.The impact of manual and automatic testing on software testing efficiency and effective-

DOI 10.15588/1607-3274-2025-2-19

ness [Electronic resource], *Indian Journal of Science and Research,* 2023, Vol. 3, Iss. 3, pp. 88–93. Mode of access: https://www.ijsronline.org/issue/20230423-144118.016.pdf (acc. date: 02.01.2025). Title from screen.
17. Mehmood A. M., Murtuza A. M., Vazeer A. M. Impact of artificial intelligence on the automation of digital health system, *International Journal of Software Engineering & Applications (IJSEA)*, 2022, Vol. 13, No. 6, pp. 23–29. DOI: 10.5121/ijsea.2022.13602

Received 08.01.2025.
Accepted 18.04.2025.

УДК 004.8

МЕТОД ПОБУДОВИ МОДЕЛЕЙ ПОЛІСУБ'ЄКТНОГО МУЛЬТИФАКТОРНОГО СЕРЕДОВИЩА ПІДТРИМКИ ПРОГРАМНИХ КОМПЛЕКСІВ

Пукач А. І. – канд. техн. наук, асистент кафедри Автоматизованих Систем Управління Інституту Комп'ютерних Наук та Інформаційних Технологій Національного Університету «Львівська Політехніка», Львів, Україна.

Теслюк В. М. – д-р техн. наук, проф., завідувач кафедри Автоматизованих Систем Управління Інституту Комп'ютерних Наук та Інформаційних Технологій Національного Університету «Львівська Політехніка», Львів, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто задачу побудови моделей полісуб'єктного мультифакторного середовища підтримки програмних комплексів, що забезпечує врахування дії різноманітних факторів впливу на сам підтримуваний програмний комплекс, на процеси його підтримки, а також на суб'єктів взаємодії з ним, що забезпечують та реалізують цю підтримку. **Об'єктом дослідження** є процеси комплексної підтримки програмних продуктів, процеси автоматизації цієї підтримки, процеси впливу факторів на об'єкт та суб'єкти комплексної підтримки програмних продуктів, а також процеси суб'єктивізації сприйняття об'єкта підтримки відповідними суб'єктами взаємодії з ним. **Предметом дослідження** є методи та засоби штучних нейронних мереж, зокрема багатошарового перцептрона, а також комп'ютерного проектування та моделювання. **Метою роботи** є розроблення методу побудови моделей полісуб'єктного мультифакторного середовища комплексної підтримки програмних продуктів.

Метод. Запропоновано розроблення моделей полісуб'єктного мультифакторного середовища підтримки програмних комплексів, що дає змогу, в автоматизованому режимі, отримати відповідні моделі, на основі яких, в подальшому – досліджувати сильні та слабкі сторони конкретного досліджуваного середовища комплексної підтримки того чи іншого програмного продукту, з метою забезпечення подальшого покращення та автоматизації його підтримки на основі вивчення та аналізу факторів впливу, що формують суб'єктивне бачення цієї підтримки тими суб'єктами, які її, власне, безпосередньо здійснюють, тобто, фактично, від яких залежить сама ця підтримка, а також відповідні її якісні та кількісні характеристики і показники.

Результати. Результатами роботи розробленого методу є відповідні моделі досліджуваних полісуб'єктних мультифакторних середовищ комплексної підтримки програмних продуктів, що враховують наявність та рівень впливу відповідних наявних факторів впливу на суб'єктів взаємодії з підтримуваними програмними комплексами, які (суб'єкти) безпосередньо забезпечують і реалізують цю комплексну підтримку досліджуваних програмних продуктів, та формують релевантні досліджувані середовища підтримки. Крім того, в якості прикладу практичного застосування та апробації, розроблений метод використано, зокрема, для розв'язання прикладної практичної задачі визначення домінуючого та дефіцитного факторів впливу полісуб'єктного мультифакторного середовища підтримки досліджуваного програмного комплексу, а також наведено та проаналізовано отримані результати розв'язання поставленої задачі.

Висновки. Розроблений метод вирішує поставлену задачу побудови моделей полісуб'єктного мультифакторного середовища підтримки програмних комплексів, та забезпечує врахування дії різноманітних (попередньо узгоджених та задекларованих) факторів впливу на сам підтримуваний програмний комплекс, на процеси його підтримки, а також на суб'єктів взаємодії з ним, що забезпечують та реалізують цю комплексну підтримку. Зокрема, розроблений метод дає змогу моделювати та досліджувати полісуб'єктні мультифакторні середовища комплексної підтримки програмних продуктів, що відображають глобальний (або локальний) вплив різноманітних наявних факторів як на сам об'єкт підтримки (підтримуваний програмний комплекс, чи процеси його комплексної підтримки), так і на суб'єктів, що безпосередньо здійснюють та реалізують дану комплексну підтримку в усіх її можливих та/або задекларованих проявах. Практична апробація розробленого методу здійснена на прикладі вирішення конкретних прикладних практичних задач, однією з яких є представлена в роботі задача визначення домінуючого та дефіцитного факторів впливу полісуб'єктного мультифакторного середовища підтримки досліджуваного програмного комплексу, та підтверджує його ефективність при розв'язанні стеку прикладних практичних задач дослідження впливу факторів на комплексну підтримку програмних продуктів, з використанням переваг технологій штучного інтелекту, машинного навчання, штучних нейронних мереж, та багатошарового перцептрона зокрема.

КЛЮЧОВІ СЛОВА: програмний продукт, комплексна підтримка, середовище підтримки програмних продуктів, фактор впливу, автоматизація, штучні нейронні мережі, багатошаровий перцептрон.

ЛІТЕРАТУРА

1. Panwar A. Implementation of Software Testing Using Machine Learning: A Systematic Mapping Study / A. Panwar, P. Peddi // JJTU Journal of Renewable Energy Exchange. – 2023. – Vol. 11, Iss. 7. – P. 58–64. – DOI: 10.58443/ijrex.11.7.2023.58-64
2. Gadani N. N. Artificial Intelligence: Leveraging Ai-Based Techniques For Software Quality / N.N. Gadani // International Research Journal of Modernization in Engineering Technology and Science. – 2024. – Vol. 06, Iss. 07. – P. 757–769. – DOI: 10.56726/IRJMETS60018
3. Chaudhary J. Predictive Modeling Of Automation In Software Testing A Machine Learning Approach For Efficient

DOI 10.15588/1607-3274-2025-2-19

Test Case Selection [Electronic resource] / J. Chaudhary, P. Anand // Advances in Mechanics. – 2023. – Vol. 11, Iss. 2. – P. 128–139. – Mode of access: https://www.advancesinmechanics.com/pdf/2023-128.pdf (date of access: 01.01.2025). – Title from screen.
4. Manchana R. The DevOps Automation Imperative: Enhancing Software Lifecycle Efficiency and Collaboration / R. Manchana // European Journal of Advances in Engineering and Technology. – 2021. – Vol. 8, Iss. 7. – P. 100–112. – DOI: 10.5281/zenodo.13789734
5. Ogala J. O. A Complete Guide to DevOps Best Practices / J.O. Ogala // International Journal of Computer Science and Information Security (IJCSIS). – 2022. – Vol. 20, No. 2. – P. 1–6. – DOI: 10.5281/zenodo.6376787
6. Adnan S. DevOps with Agile: Best practices to improve software quality / S. Adnan, K. Moin, J. Imran // International Conference On Biological Research And Applied Science. – 2023. – P. 61–66. – DOI: 10.37962/ibras/2023/61-66
7. Rajadhyax D. A Survey of Methods, Tools and Applications of Knowledge Base Construction (KBC) [Electronic resource] / D. Rajadhyax // Telecom Business Review: SIDTM Journal. – 2020. – Vol. 13, Iss. 1. – P. 20–26. – Mode of access: https://sidtm.edu.in/wp-content/uploads/2021/06/tbr2020.pdf#page=25 (date of access: 01.01.2025). – Title from screen.
8. Verrev M. Combining Semantic Parsing Frameworks for Automated Knowledge Base Construction [Electronic resource] / M. Verrev // 6th Workshop on Advances In Argumentation In Artificial Intelligence. – 2022. – 11 pages. – Mode of access: https://ceur-ws.org/Vol-3354/paper1.pdf (date of access: 01.01.2025). – Title from screen.
9. Rapp A. The human side of human-chatbot interaction: A systematic literature review of ten years of research on text-based chatbots / A. Rapp, L. Curti, A. Boldi // International Journal of Human-Computer Studies. – 2021. – Vol. 151, Iss. 3. – Article ID: 102630. – DOI: 10.1016/j.ijhcs.2021.102630
10. AI for Information Technology Operation (AIOps): A Review of IT Incident Risk Prediction / [S. Ahmed, M. Singh, B. Doherty et al.] // 9-th International Conference on Soft Computing and Machine Intelligence (ISCMI). – 2022. – P. 253–257. – DOI: 10.1109/ISCMI56532.2022.10068482
11. Paramesh S. P. A deep learning based IT service desk ticket classifier using CNN [Electronic resource] / S. P. Paramesh, K. S. Shreedhara// ICTACT journal on soft computing. – 2022. – Vol. 13, Iss. 1. – P. 2805–2812. – Mode of access: https://ictactjournals.in/paper/IJSC_Vol_13_Iss_1_Paper_9_2805_2812.pdf (acc. date: 02.01.2025). – Title from screen.
12. Пукач А. І. Математична модель аналізу факторів впливу підтримки програмних комплексів / А. І. Пукач, В. М. Теслюк // Поліграфія і видавнича справа. – 2024. – 1(87). – С. 75–85. – DOI: 10.32403/0554-4866-2024-1-87-75-85
13. Пукач А. І. Модель декомпонованої ізоляційної домінації для аналізу чинників впливу на автоматизацію підтримки програмних комплексів / А. І. Пукач, В. М. Теслюк // Науковий вісник НЛТУ України. – 2024. – 34(5). – С. 170–179. – DOI: 10.36930/40340521
14. Пукач А. І. Інформаційна модель аналізу факторів впливу автоматизації підтримки програмних комплексів з використанням середовищ R та Python / А. І. Пукач, В. М. Теслюк // Вісник Львівського державного університету безпеки життєдіяльності. – 2024. – 29. – С. 54–64. – DOI: 10.32447/20784643.29.2024.06
15. Scrum and the Agile procedure’s impact on software project management / [N. Rasool, S. Yousaf, U. Haseeb et al.] // Journal of Jilin University (Engineering and Technology Edition). – 2023. – Vol. 42, Iss. 2. – P. 380–392. – DOI: 10.17605/OSF.IO/MQW9P
16. Thant K. S. The impact of manual and automatic testing on software testing efficiency and effectiveness[Electronic resource] / K. S. Thant, H. H. K. Tin // Indian Journal of Science and Research. – 2023. – Vol. 3, Iss. 3. – P. 88–93. – Mode of access: https://www.ijsronline.org/issue/20230423-144118.016.pdf (acc. date: 02.01.2025). – Title from screen.
17. Mehmood A. M. Impact of artificial intelligence on the automation of digital health system / A. M. Mehmood, A. M. Murtuza, A. M. Vazeer // International Journal of Software Engineering & Applications (IJSEA). – 2022. – Vol. 13, No. 6. – P. 23–29. – DOI: 10.5121/ijsea.2022.13602

DOI 10.15588/1607-3274-2025-2-19