

**A. I. Pukach, T. V. Teslyuk***Lviv Polytechnic National University, Lviv, Ukraine*

METHOD FOR BALANCING MULTISUBJECT POLYFACTORIAL ENVIRONMENTS OF SOFTWARE PRODUCTS' COMPLEX SUPPORT

Within the scope of current research, the scientific and applied problem of balancing multisubject polyfactorial environments of software products' complex support was considered, in the context of a more global scientific and applied problem of automation and intellectualization of the complex support of software products, as well as a human-machine interaction. The relevant object of research in this work – is the process of balancing multisubject polyfactorial environments of software products' complex support. An appropriate method has been developed to ensure the possibility of balancing (and further potential automation of these processes) of multisubject polyfactorial environments, which provides the possibility of solving the declared scientific and applied problem of current research. In particular, the necessary algorithm for balancing the investigated multisubject polyfactorial environments has been developed, as well as the corresponding basic mathematical model, which provides the possibility of interpreting, researching, and modelling the processes of balancing of the multisubject polyfactorial environments of software products' complex support. A practical approbation of the developed method has been carried out on the example of solving the experimental applied problem of identification and monitoring “of-the-trend” dynamics of the deficient component during the automated balancing of the investigated multisubject polyfactorial support environment of the research software complex. The prospects for further research directions related to possible ways of potential improvement(s) and enhancement(s), as well as practical application of the developed method (for balancing multisubject polyfactorial environments of software products' complex support), have been considered. In particular, as one of the potential directions of further research, we see the development of additional algorithmic and software supply that would ensure possibilities for better modelling the researched processes of various balancing multisubject polyfactorial environments for different software's complex support, in order to find potential evolution ways of the proposed concept.

Keywords: software, comprehensive software support, balancing of software support environments, intellectualization of comprehensive software support.

Introduction

The modern world is filled with a variety of software tools and systems aimed at simplifying people's lives. However, the number and complexity of such software products continue to grow relentlessly, generating new challenges for researchers. One of these challenges is the subjectivity in the perception of both the software products themselves and the processes of interaction with them (united under the concept of “comprehensive support”) by the subjects directly implementing the components of such support. These subjects ensure the correct operation and viability of the software in today's aggressive and competitive information technology (IT) market. At the same time, the main complexity and problem of comprehensive software support lie in the fact that it is carried out by different subjects (members of the respective teams: developers, testers, business analysts, system engineers, etc.), each of whom has their own subjective, personalized “vision” (or perception) of the support object – whether it is the software product itself or the processes involved in its comprehensive support. In turn, this

personalized perception by the subjects is shaped by differences in the proportion of influence exerted by each factor (from the common set of influencing factors shared by all subjects). For one subject, certain factors may be dominating and dominant, while for another – completely different ones may prevail. As a result, we observe an imbalance in the entire network (group, team, or in general – the whole multisubject environment) of subjects involved in comprehensive software support, which leads to negative consequences and related support issues. Hence arises a relevant applied scientific problem.

The object of the study is the process of balancing multisubject polyfactor environments of comprehensive software support.

The subject of the study is the methods and tools of software and computer simulation, as well as artificial intelligence methods and tools.

The aim of this paper is to develop a method for balancing multisubject polyfactor environments of comprehensive software support, which will enable improvement in the balance of such environments and,

consequently, enhance the quality of the support provided within them.

To achieve this goal, the following main *research tasks* were set:

- To analyze the subject area of balancing multisubject polyfactor environments of comprehensive software support.
- To develop an algorithm for balancing multisubject polyfactor environments of comprehensive software support.
- To create a basic mathematical model for balancing multisubject polyfactor environments of comprehensive software support.
- To conduct practical testing of the proposed method by solving an experimental applied task.

This paper presents the complete development process of a method designed to address the declared applied scientific problem of balancing multisubject polyfactor environments of comprehensive software support.

Analysis of recent studies and publications. The analysis of recent studies and publications was carried out across several key research directions, such as:

automation of components of comprehensive software support (DevOps and Agile methodologies);

- software testing; incident handling); issues of human-machine interaction;
- problems related to influencing factors on the components of the comprehensive software support;
- as well as, the study of multisubject polyfactor environments where such comprehensive support of any software products is directly implemented.

Such a comprehensive approach to research analysis enables a deep and representative understanding of the overall problem domain. In particular, regarding the topic of automation in comprehensive software support components, the following papers should be noted. Paper [1] provides a general overview of the current state of application of DevOps automation practices, while paper [2] offers a deeper insight into modern DevOps trends by exploring their integration with generative artificial intelligence technologies. Meanwhile, paper [3] addresses the highly relevant issue of automating the development of the component of comprehensive software support through the integration of an AI-based program code analysis subsystem. At the same time, another fundamental component of comprehensive software support – software testing – although arguably one of the oldest, continues to evolve. This is confirmed by studies in papers [4] and [5], in which the authors propose relevant optimization solutions based on artificial intelligence and virtual reality technologies, respectively.

Additionally, another key area of automation within the context of comprehensive software product support is incident processing – both “external” (originating from end users, clients, or customers of the software) and “internal” (created internally by employees of the companies developing these software products). For instance, the author of paper [6] studied the possibilities of automated incident

response using machine learning and artificial intelligence technologies within the DevOps methodology. Another study [7] focused on predictive incident management, also applying machine learning methods and tools. Meanwhile, the authors of paper [8] conducted a comprehensive review of the use of machine learning approaches in systems for software users request registration and handling. An additional noteworthy study of incident processing [9] presents a practical solution for technical support management, aimed at enhancing service delivery in a technical university setting. The authors of [10] carried out an in-depth investigation into the use of generative AI in technical support systems.

At the same time, within the research domain of human-machine interaction, several recent works have made valuable contributions. For example, [11] offers a thorough analysis of human-machine interaction and augmented intelligence in the paradigm of interactive machine learning in educational software. Paper [12] provides a systematic literature review on artificial intelligence in human-machine interaction, while in [13] the authors present practical guidelines for combining human-machine interaction with AI.

Moreover, various influencing factors undoubtedly play a significant role in the context of comprehensive software support and the study of multisubject polyfactor environments associated with such support, as they affect comprehensive software support processes in one way or another, to a greater or lesser extent. For instance, the authors of paper [14] propose a tool comprising 79 elements across 13 different social and human influence factors to measure perception of these influences on software development performance. In [15], the authors conduct a comprehensive analysis of literature in the context of influence factors from two generalized categories: organizational and human, impacting software development productivity in development companies. In [16], the authors examine factors affecting the implementation of secure software development lifecycle (SSDLC) practices in small and medium-sized companies practicing remote work.

While the state of research regarding the previously discussed key directions appears quite promising (in the context of ongoing scientific investigations within those areas), the situation is somewhat more complex when it comes to the study of multisubject polyfactor environments of comprehensive software support. This complexity stems from the relatively limited number of existing studies on the topic. Among the few available studies, references [17–20] consider such environments only as technological or technical constructs – sets of tools or components – without accounting for the subject dimension. In other words, they fail to take into account the presence of human or organizational actors who, alongside the technical components, actively participate in the support processes and thereby shape these very multisubject polyfactor support environments.

Based on this comprehensive analysis of current research and publications in the context of the studied issues, the

following can be concluded: the automation and intelligent enhancement of each component of comprehensive software product support is a highly relevant applied scientific problem today, particularly in the context of human-machine interaction. However, the influence factors affecting the subjects responsible for delivering this support remain significantly underestimated. This results in evident research gaps, particularly in studies that could facilitate the resolution of derivative applied scientific problems and challenges associated with multisubject polyfactor environments of comprehensive software support, including the issue of balancing such environments.

On a more optimistic note, some progress has been made to fill this research gap. Notably, study [21] proposes a method for generating individual polyfactor “profiles” (or models) of support agents using a multilayer perceptron. The creation of such individual polyfactor profiles for all agents involved in the studied multisubject

environment enables further exploration of the balancing problem, which is discussed in greater detail in the next section of this paper.

Research results and their discussion

The initial stage and primary step in addressing the declared problem of balancing multisubject polyfactor environments of comprehensive software support is the development of an appropriate balancing algorithm. Specifically, Fig. 1 below presents a block diagram of the developed algorithm.

The next step in solving the stated problem of balancing multisubject polyfactor environments of comprehensive software support is the development of a corresponding baseline mathematical model, which consists of a set of expressions (1)–(4) described and explained in detail further below.

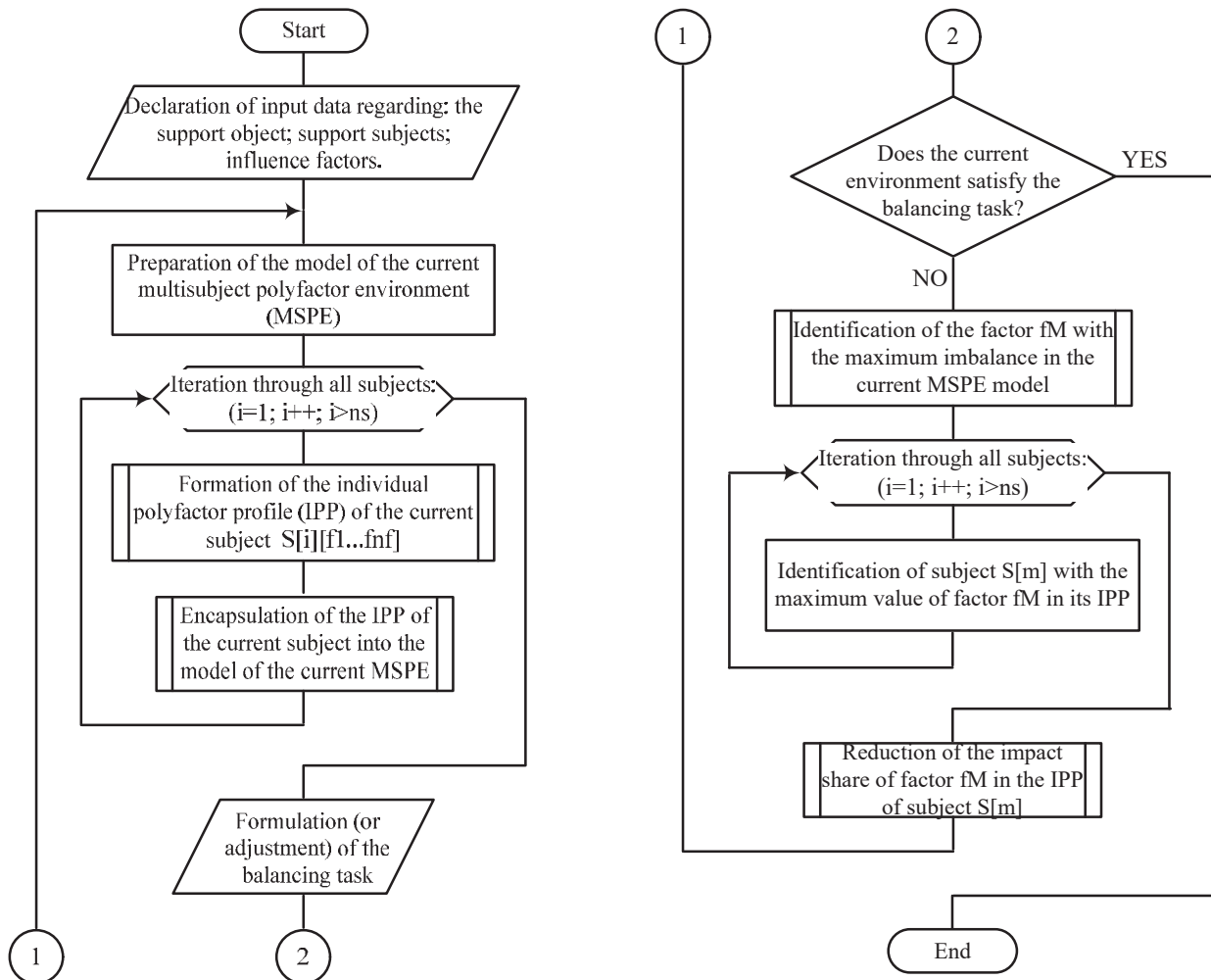


Fig. 1. Block diagram of the balancing algorithm for multisubject multifactor environments of software products comprehensive support

In particular, expression (1) below represents the model of a multisubject polyfactor environment (of comprehensive software support), which is formed on the basis of the individual polyfactor profiles of all the actors constituting this environment:

$$MPS = \left\{ \frac{1}{ns} \sum_{i=1}^{ns} IPS_{[i]}^{[j]}, j = \overline{1, nf} \right\}, \quad (1)$$

where MPS – a variable-identifier denoting the studied multisubject polyfactor environment (comprehensive

support for the software product under consideration); $IPS_{[j]}^{[i]}$ – the share of influence of the j -th factor within the individual polyfactor profile of the i -th subject (within the studied multisubject polyfactor environment); ns – the number of subjects forming the studied multisubject polyfactor environment; nf – the number of influence factors shaping the individual polyfactor profile of each subject (within the studied multisubject polyfactor environment).

Expression (2) below represents the identification of the imbalance level (in general, although component-wise identification is also possible) compared to the acceptable level of imbalance, which is determined / set by the current conditions of the balancing task (either during its initial formulation or during subsequent adjustment):

$$MPS_{[disbalance]} = fMaxF(MPS) - \Delta, \quad (2)$$

Imbalance

where $MPS_{[imbalance]}$ – the identified level of imbalance (in this particular case – in general, although it is also possible to identify imbalance for each individual component, which in such a case would be additionally indexed in the upper register) of the current studied multisubject polyfactor environment; $fMaxF$ – a function for identifying the potential factor of maximum imbalance (for the current studied multisubject polyfactor environment); Δ – the acceptable imbalance value, established by the current conditions of the balancing task.

Expression (3) below details the function for detecting a potential factor of maximum imbalance:

$$fMaxF(MPS) =$$

$$\left\{ MPS_{[j]} \mid MPS_{[j]} > MPS_{[i]}, i \in [1..nf], i \neq j; j = \overline{1, nf} \right\}, \quad (3)$$

where $MPS_{[j]}$ – the value of the share of influence of the j -th factor within the current studied multisubject polyfactor environment (i.e., the software product under study).

The following expression (4) represents the identification of the subject with the maximum value of the indicator of a specific factor of influence (denoted as “ fM ”), the share of influence of which was returned by the previously described function $fMaxF$:

$$IPS_{[k]}^{[fM]} = IPS_{[j]}^{[i]}, \forall$$

$$\left((j = \overline{1, ns}) \wedge (i \in [1..nf]) \wedge (IPS_{[j]}^{[i]} = fMaxF(MPS)) \right), \quad (4)$$

where k – a variable-identifier of the subject whose share of influence of the fM -th factor must be reduced (within the current balancing cycle, in the context of the current multisubject polyfactor environment under investigation).

Thus, expressions (1)–(4) represent the developed basic mathematical model for balancing multisubject polyfactor environments of comprehensive software product support. In turn, the developed algorithm for balancing multisubject polyfactor environments of comprehensive software product support, along with the above mathematical model, collectively form the fundamental basis of the developed method for balancing multisubject polyfactor environments of comprehensive software product support.

At the same time, the final stage in the development of the proposed method is its validation through solving a practical applied task – identifying and monitoring the trend dynamics of a deficit component during the balancing process of the investigated multisubject polyfactor environment of support. The solution to this declared experimental task is carried out in several stages (fully aligned with the developed and presented algorithm for balancing multisubject polyfactor environments of comprehensive software support), the first of which is the construction of individual polyfactor profiles for all the subjects that form the studied multisubject polyfactor environment of software support and are involved in the given applied task. Fig. 2 below presents a graphical interpretation of the obtained individual polyfactor profiles of the subjects in the environment.

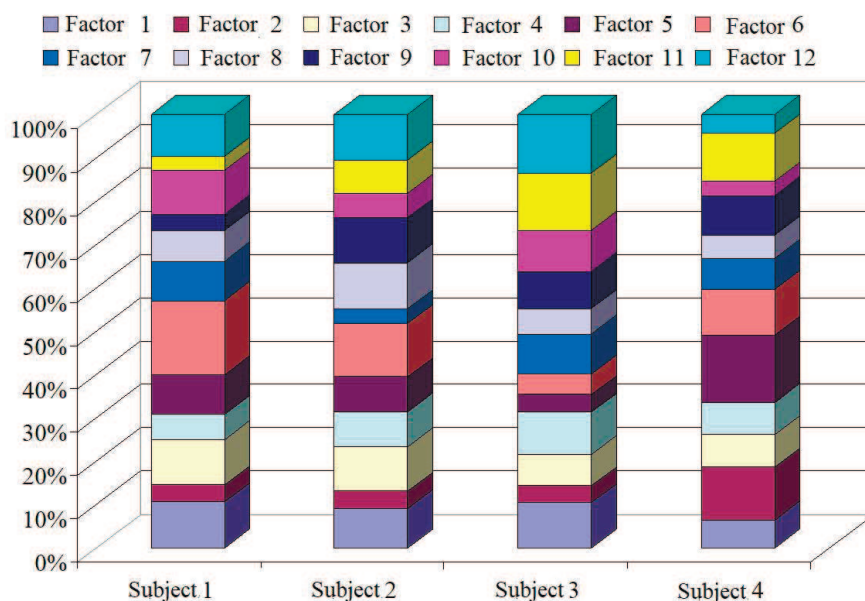


Fig. 2. Graphical interpretation of the obtained individual polyfactor profiles of the subjects in the studied environment

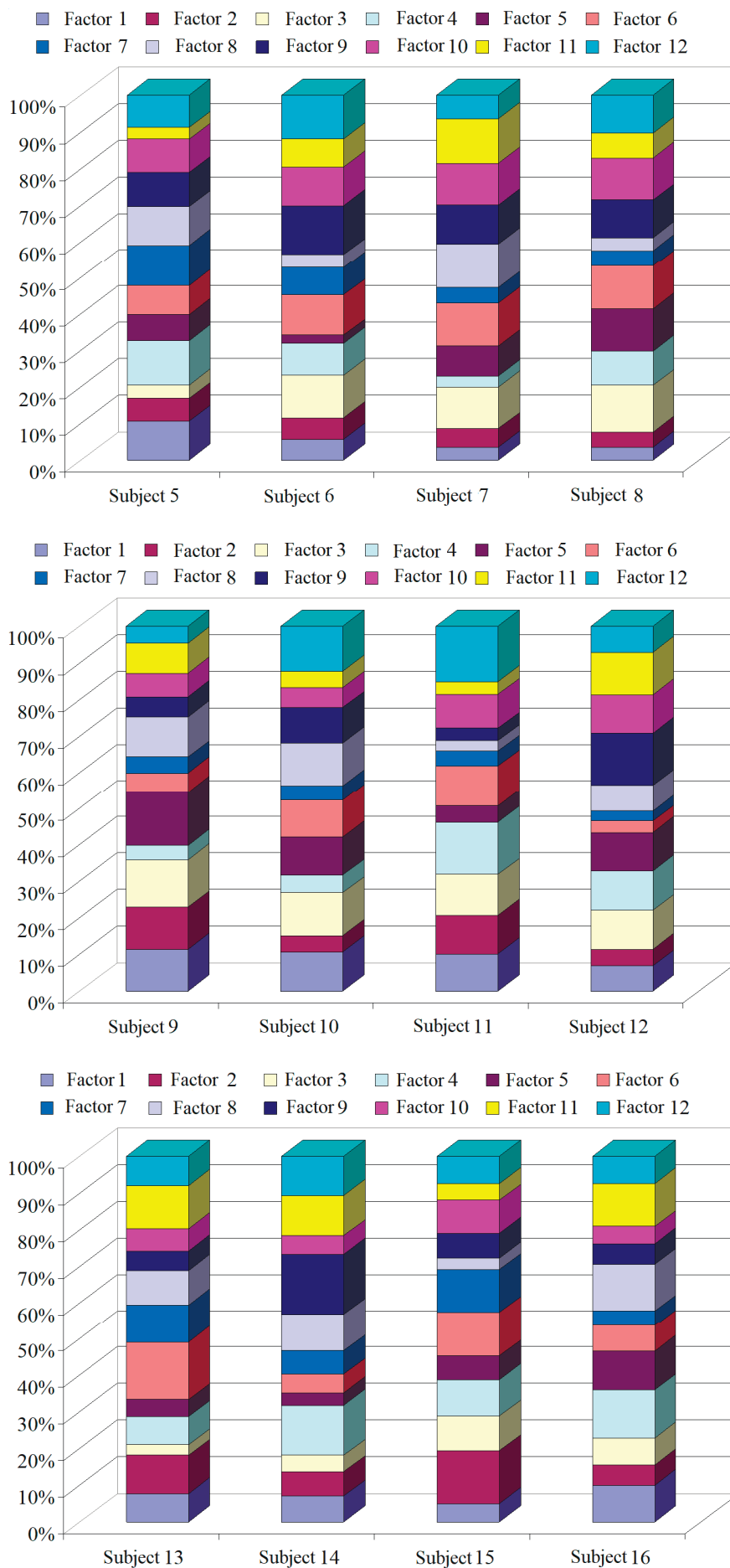


Fig. 2. Graphical interpretation of the obtained individual polyfactor profiles of the subjects in the studied environment (continued)

The next stage is the formation of a generalized profile of the current multisubject polyfactor environment of comprehensive support (based on the previously obtained

individual polyfactor profiles of all its constituent subjects), with its corresponding graphical interpretation shown below in Fig. 3.

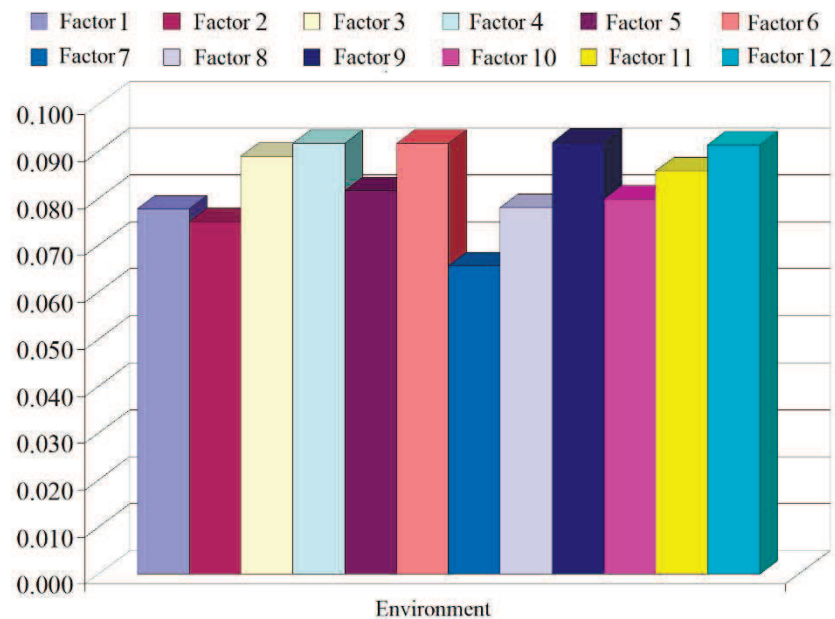


Fig. 3. Graphical interpretation of the obtained generalized profile of the current studied multisubject polyfactor environment

The final stage of solving the declared practical applied task is the identification and monitoring of the trend dynamics of the deficit component during the balancing process of the studied multisubject polyfactor support environment.

It is worth noting that multisubject polyfactor support environments of any software products are closed systems. A specific feature of such systems is that regulating one part inevitably results in a deficit in other parts. In practice, in the context of balancing multisubject polyfactor environments, this means that reducing the influence of a dominant factor will automatically induce the rise of another (or several

other) factors, which in turn may lead to a reduction in yet another factor (or factors), etc. This is a natural and normal process (one that should not be feared or “fought against”) for closed systems such as the multisubject polyfactor environments of comprehensive software support considered in this study.

Below, in Fig. 4, a graphical visualization of the generalized trend dynamics of the deficit component in the studied multisubject polyfactor environment is presented (based on test cases [001]–[100]), without factor-by-factor detailing – that is, without specifying the unique identifier (ordinal number) of the factor, but showing only its value.

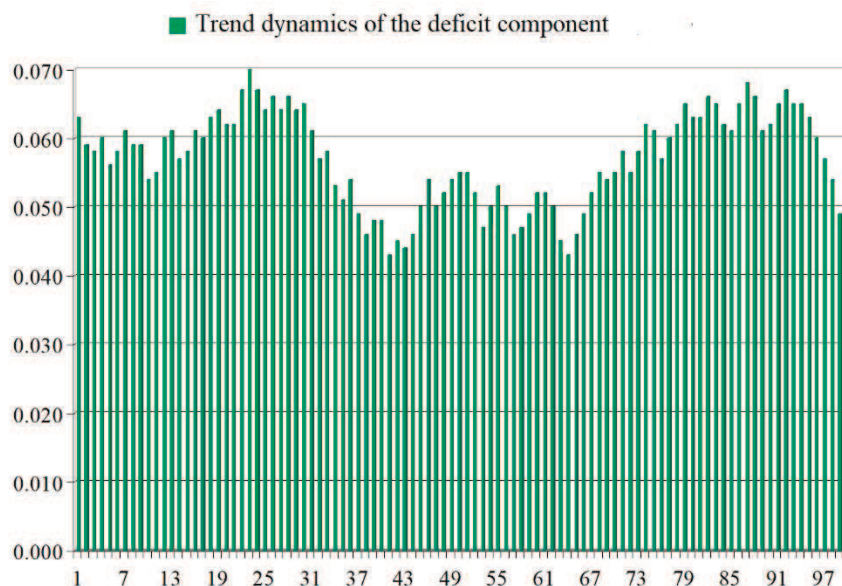


Fig. 4. Graphical visualization of the generalized trend dynamics of the deficit component

Meanwhile, Fig. 5 presents a detailed (factor-by-factor) trend dynamics of the deficit component during the balancing of the studied environment, indicating not only the factor value but also its unique identifier.

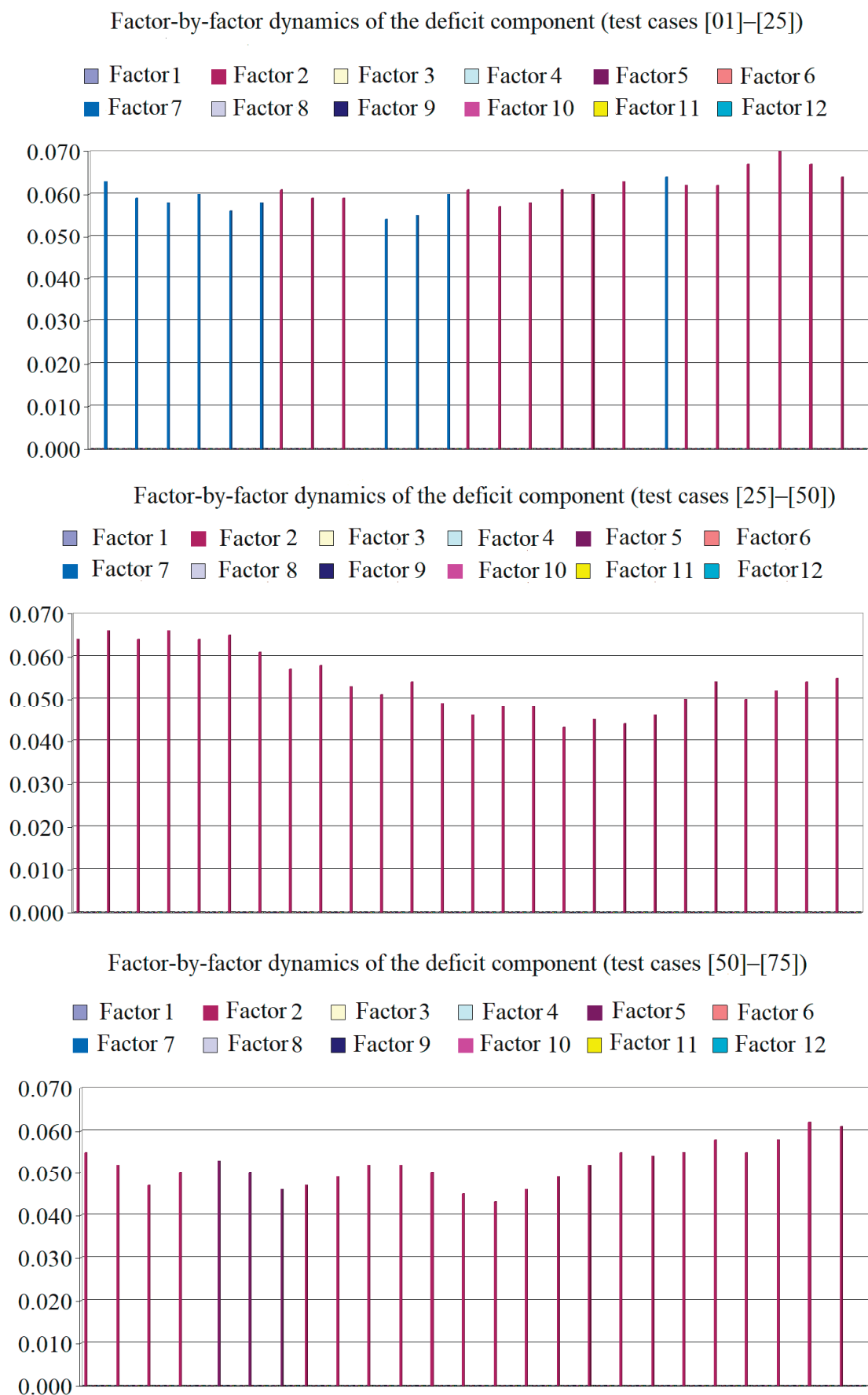


Fig. 5. Graphical visualization of the factor-by-factor dynamics of the deficit component

Factor-by-factor dynamics of the deficit component (test cases [75]–[100])

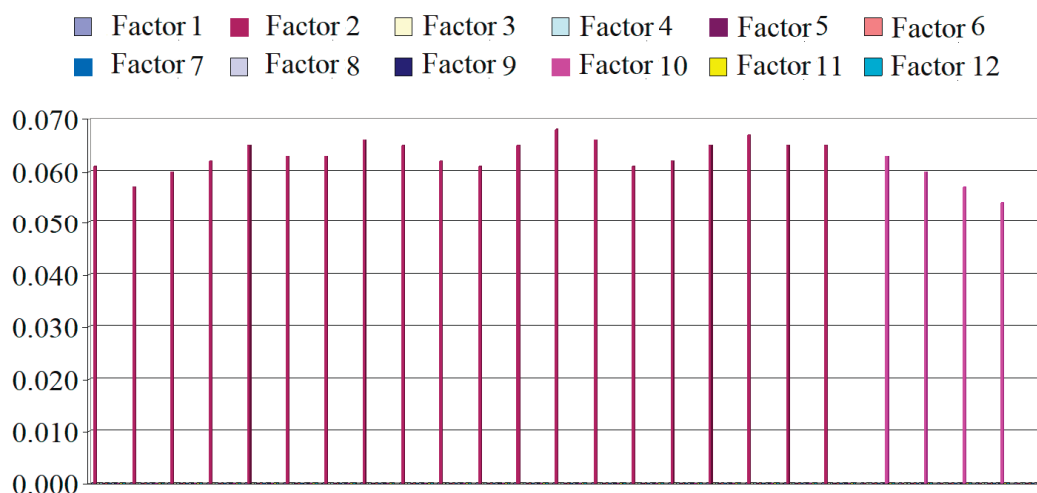


Fig. 5. Graphical visualization of the factor-by-factor dynamics of the deficit component (continued)

Thus, the declared experimental applied task of identifying and monitoring the trend dynamics of the deficit component during the balancing of the studied multisubject polyfactor support environment has been successfully resolved. In turn, the resolution of this applied task provides a successful validation of the proposed method, thereby finalizing the full cycle of its development within the framework of this research.

Discussion of the results. In paper [22], the authors investigate adaptive and multicriteria balancing within decision support environments in the context of planning the testing component of comprehensive software support. The study focuses on balancing numerous criteria within a preference-based structure, including balancing between such categories as cost, risk, and approved time limits during decision-making regarding testing planning in the context of rapid system deployment.

Meanwhile, in another study [23], the authors explore the use of information criteria such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) for balancing ARIMA (Autoregressive Integrated Moving Average) models to evaluate performance indicators in the context of predictive analytics within cloud computing environments.

At the same time, in another study [24], the author investigates, in particular, the issue of balancing computational resources (in data-absent environments using machine learning approaches and incorporating Gaussian mixture models) through several stopping criteria, including: maximum number of iterations, monitoring changes in log-likelihood, and evaluating minimal parameter changes (such as means and covariances).

In contrast, the results obtained in the course of the present research, as presented in this paper, provide a comprehensive solution specifically in the context of the narrowly focused scientific and applied challenge of balancing multisubject polyfactor environments of comprehensive software support. This includes full consideration

of the domain-specific features, peculiarities, and potential difficulties involved in solving subordinate tasks within this class of problems.

Thus, based on the results obtained, we affirm a high level of specialization of the developed solution – both in terms of the balancing method as a whole and its individual components (the algorithm and the model) – in relation to the studied scientific and applied problem of balancing multisubject polyfactor environments of comprehensive software support. Furthermore, the results of the practical validation confirm the significant applied potential of the proposed innovation – not only within the broader scientific and applied context of automating and intellectualizing comprehensive software support, but also in adjacent areas, particularly in the field of human–machine interaction.

Scientific novelty of the obtained results – For the first time, a method has been developed for balancing multisubject polyfactor environments of comprehensive software support, which provides the capability to resolve the declared applied scientific problem of the same name, within the context of the broader challenge of automating and intellectualizing comprehensive software support.

Practical significance of the research results – The developed method for balancing multisubject polyfactor environments of comprehensive software support has undergone practical validation, which confirms that this method can be used to solve a wide range of relevant applied tasks related to balancing, adjusting, or diagnosing environments of this class.

Conclusions

This paper presents a developed method for balancing multisubject polyfactor environments of comprehensive software support, enabling the resolution of the eponymous applied scientific problem within the broader context of automating and intellectualizing comprehensive software support, as well as within the domain of human-machine interaction.

An appropriate balancing algorithm for the studied environments has been developed, along with a corresponding basic mathematical model (comprising four components, each of which may function as an autonomous submodel of the respective process). Together, these elements provide the capability to interpret, study, and simulate balancing processes in multisubject polyfactor environments of comprehensive software support.

As the final stage of the declared method's development, practical validation was performed through solving an experimental applied task of identifying and monitoring the trend dynamics of the deficit component during the automated balancing of the studied multisubject polyfactor support environment.

The results obtained confirm the effectiveness and efficiency of the developed solution in addressing both relevant scientific and applied challenges and practical tasks within the overarching scientific and applied domain of automation and intellectualization of comprehensive software support, as well as within the field of human-machine interaction.

Future research directions aimed at potential enhancement and practical implementation of the developed balancing method of multisubject polyfactor support environment have been considered. These include, in particular, the development of additional algorithmic and software tools to further extend the capabilities of computer-based modeling.

References

- [1] Patchamatla, P. S. S. (2025). Enhancing Software Development Efficiency: A Comprehensive Study on DevOps Practices and Automation. *Recent Trends in Information Technology and Its Application*, 8(2), 1–3. <https://doi.org/10.5281/zenodo.14916413>
- [2] Joshi, S. (2025). Introduction to Generative AI and DevOps: Synergies, Challenges and Applications. *International Journal of Advanced Research in Science, Communication and Technology*, 5(1), 205–225. <https://doi.org/10.48175/ijarsct-23634>
- [3] Ahmed, S. (2025). Integrating AI-Driven Automated Code Review in Agile Development: Benefits, Challenges, and Best Practices. *International Journal of Advanced Engineering, Management and Science*, 11(2), 01–10. <https://doi.org/10.22161/ijaems.112.1>
- [4] Kathiriyai, S., Karangara, R., & Challla, N. (2018). Optimizing Automated Software Testing with Machine Learning Techniques. *International Journal of Science and Research (IJSR)*, 7(3), 1960–1964. <https://doi.org/10.21275/sr24304113021>
- [5] Hunko, I., Muliarevych, O., Trishchuk, R., Zybin, S., & Halachev, P. (2024). The role of virtual reality in improving software testing methods and tools. *Journal of Theoretical and Applied Information Technology*, 102(11), 4723–4734. URL: <https://www.jatit.org/volumes/Vol102No11/6Vol102No11.pdf>
- [6] Raghavendran R. (2024). Machine learning and artificial intelligence in devops: applications for predictive analytics, anomaly detection, and automated incident response. *International Journal of Computer Science and Information Technology Research*, 5(3), 1–10. URL: https://ijcsitr.com/index.php/home/article/view/IJCSITR_2024_05_03_01
- [7] Mahida, A. (2022). Predictive Incident Management Using Machine Learning. *International Journal of Science and Research (IJSR)*, 11(6), 1977–1980. <https://doi.org/10.21275/sr24401231847>
- [8] Wijaya, A.S. & Oktavia, T. (2025). Machine learning approaches for helpdesk ticketing system: a systematic literature review. *Journal of Theoretical and Applied Information Technology*, 102(5), 1831–1842. URL: <https://www.jatit.org/volumes/Vol102No5/14Vol102No5.pdf>
- [9] Amanortsu, G. & Ahiawortor, D. (2024). Developing a Technical Support Management Solution for Enhanced Service Delivery at Accra Technical University. *International Journal of Innovative Science and Research Technology*, 9(12), 1939–1950. <https://doi.org/10.5281/zenodo.14576739>
- [10] Kannan, S. & Saradhi, S. (2023). Generative AI in Technical Support Systems: Enhancing Problem Resolution Efficiency Through AIDriven Learning and Adaptation Models. *Nanotechnology Perceptions*, 19(S1), 198–217. <https://doi.org/10.62441/nano-ntp.vi.4980>
- [11] Troussas, C., Krouska, A., & Sgouropoulou, C. (2025). Human-Computer Interaction and Augmented Intelligence. In *Cognitive Systems Monographs*. Springer Nature Switzerland. 431 p. <https://doi.org/10.1007/978-3-031-84453-9>
- [12] Oktafiani, R., Muklis, S. & Asmanto, Y. (2024). Artificial Intelligence for Human Computer Interaction: Systematic Literature Review. *JURNAL ADAT-Jurnal Seni, Desain & Budaya Dewan Kesenian Tangerang Selatan*, 6(2), 65–72. URL: [https://jurnaladat.or.id/public/full_paper/JurnalAdat\(65-72\)%20-%20Artificial%20Intelligence%20for%20Human%20Computer%20Interaction-%20Systematic%20Literature%20Review.pdf](https://jurnaladat.or.id/public/full_paper/JurnalAdat(65-72)%20-%20Artificial%20Intelligence%20for%20Human%20Computer%20Interaction-%20Systematic%20Literature%20Review.pdf)
- [13] Russell, D., Liao, Q., Kulkarni, C., & Glassman, E. (2023). Human-Computer Interaction and AI What practitioners need to know to design and build effective AI system from a human perspective. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23)*, 3 p. <https://doi.org/10.1145/3544549.3574170>
- [14] Machuca-Villegas, L., Gasca-Hurtado, G. P., Morillo Puente, S., & Restrepo Tamayo, L. M. (2021). An Instrument for Measuring Perception about Social and Human Factors that Influence Software Development Productivity. *Journal of Universal Computer Science (JUCS)*, 27(2), 111–134. <https://doi.org/10.3897/jucs.65102>
- [15] Oliveira, E., Conte, T., Cristo, M., & Valentim, N. (2018). Influence Factors in Software Productivity – A Tertiary Literature Review. *International Conferences on Software Engineering and Knowledge Engineering*, 2018, 68–103. <https://doi.org/10.18293/seke2018-149>
- [16] Umeugo, W., Lowrey, K. & Pandya, S. Y. (2023). Factors affecting the adoption of secure software practices in small and medium enterprises that build software in-house. *International Journal of Advanced Research in Computer Science*, 14(02), 1–7. <https://doi.org/10.26483/ijarcs.v14i2.6955>
- [17] Rasheed, Z., Waseem, M., Sami, M. A., Kemell, K.-K., Ahmad, A., Duc, A. N., Systä, K., & Abrahamsson, P. (2025). Autonomous Agents in Software Development: A Vision Paper. *Lecture Notes in Business Information Processing*, 15–23. https://doi.org/10.1007/978-3-031-72781-8_2
- [18] Gwangwadza, A., & Ridewaan Hanslo (2022). Factors that Contribute to the Success of a Software Organisation's DevOps Environment: A Systematic Review. *ArXiv (Cornell University)*, 1–15. <https://doi.org/10.48550/arxiv.2211.04101>
- [19] Veeramachaneni, V. (2020). Factors that Contribute to the Success of a Software Organisation's DevOps Environment: A Systematic Review. *International Journal For Recent Development In Science And Technology*, 04(11), 5–11. <https://doi.org/10.13140/RG.2.2.21225.20327>

- [20] Ohwada, A., Kojima, T. & Amano, H. (2021). MENTAI: A Fully Automated CGRA Application Development Environment that Supports Hardware / Software Co-design. *SASIMI 2021 Proceedings*, R1-4, 19–24. URL: https://sasimi.jp/new/sasimi2021/files/archive/pdf/p19_R1-4.pdf
- [21] Pukach, A. I., & Teslyuk, V. M. (2025). Method of forming multifactor portraits of the subjects supporting software complexes, using a multilayer perceptron. *Radio Electronics, Computer Science, Control*, (1), 130–141. <https://doi.org/10.15588/1607-3274-2025-1-12>
- [22] Valerdi, R., Sullivan, B. P. (2023). Engineering Systems Integration, Testing, and Validation. In: Maier, A., Oehmen, J., Vermaas, P. E. (eds) *Handbook of Engineering Systems Design*. Springer, 1–31. https://doi.org/10.1007/978-3-030-46054-9_20-2
- [23] Bandari, V. (2021). Predictive Analytics in Cloud Computing: An ARIMA Model Study on Performance Metrics. *Applied Research in Artificial Intelligence and Cloud Computing*, 4(1), 1–18. URL: <https://researchberg.com/index.php/araic/article/view/96>
- [24] Wani, A. A. (2024). A review of challenges and solutions for using machine learning approaches for missing data. *International Journal of Engineering Applied Sciences and Technology*, 09(05), 36–50. <https://doi.org/10.33564/IJEAST.2024.v09i05.005>

А. І. Пукач, Т. В. Теслюк

Національний університет “Львівська політехніка”, Львів, Україна

МЕТОД БАЛАНСУВАННЯ МУЛЬТИСУБ’ЄКТНИХ ПОЛІФАКТОРНИХ СЕРЕДОВИЩ КОМПЛЕКСНОЇ ПІДТРИМКИ ПРОГРАМНИХ ПРОДУКТІВ

У здійсненому дослідженні розглянуто науково-прикладну проблему балансування мультисуб’єктних поліфакторних середовищ комплексної підтримки програмних продуктів у контексті глобальнішої науково-прикладної проблематики автоматизації й інтелектуалізації комплексної підтримки програмних продуктів та людино-машинної взаємодії. Об’єкт дослідження у цій роботі – процес балансування мультисуб’єктних поліфакторних середовищ комплексної підтримки програмних продуктів. Розроблений відповідний метод для забезпечення можливості балансування (та подальшої потенційної автоматизації цих процесів) таких середовищ, що забезпечує можливість вирішення задекларованої науково-прикладної проблеми цього дослідження. Зокрема, розроблено необхідний алгоритм балансування досліджуваних середовищ, а також відповідну базову математичну модель, що забезпечують можливість інтерпретації, дослідження та моделювання процесів балансування мультисуб’єктних поліфакторних середовищ комплексної підтримки програмних продуктів. Реалізовано практичну апробацію розробленого методу на прикладі розв’язання експериментальної прикладної задачі ідентифікації та моніторингу тенденційної динаміки дефіцитної компоненти в ході автоматизованого балансування досліджуваного мультисуб’єктного поліфакторного середовища підтримки. Розглянуто перспективи подальших напрямів досліджень щодо можливих способів потенційного вдосконалення та покращення, а також практичного застосування розробленого методу балансування мультисуб’єктних поліфакторних середовищ комплексної підтримки програмних продуктів.

Ключові слова: програмне забезпечення, комплексний супровід програмного забезпечення, балансування середовищ підтримки програмного забезпечення, інтелектуалізація комплексної підтримки програмного забезпечення.

Інформація про авторів:

Пукач Андрій Ігорьович, канд. техн. наук, асистент, кафедра автоматизованих систем управління. Email: andriy.i.pukach@lpnu.ua; <https://orcid.org/0009-0001-8563-3311>

Теслюк Тарас Васильович, канд. техн. наук, асистент, кафедра інформаційних систем та мереж. Email: taras.v.tesliuk@lpnu.ua; <https://orcid.org/0000-0001-6585-3715>

Цитування за ДСТУ: Пукач А. І., Теслюк Т. В. Метод балансування мультисуб’єктних поліфакторних середовищ комплексної підтримки програмних продуктів. *Український журнал інформаційних технологій*. 2025, т. 7, № 1. С. 25–34.

Citation APA: Pukach, A. I., & Teslyuk, T. V. (2025). Method for balancing multisubject polyfactorial environments of software products' complex support. *Ukrainian Journal of Information Tecnology*, 7(1), 25–34. <https://doi.org/10.23939/ujit2025.01.025>