

УДК 004.58

МАТЕМАТИЧНА МОДЕЛЬ СУБ'ЄКТНОГО СЕРЕДОВИЩА ПІДТРИМКИ ПРОГРАМНИХ КОМПЛЕКСІВ

А. І. Пукач, В. М. Теслюк

*кафедра АСУ, Інститут Комп'ютерних Наук та Інформаційних Технологій,
Національний Університет «Львівська Політехніка», м. Львів, Україна*

Розглянуто науково-прикладну проблему моделювання середовищ підтримки програмних комплексів, що базуються на основі складових суб'єктів, котрі, власне, безпосередньо забезпечують та реалізують цю комплексну підтримку. Розроблена відповідна математична модель суб'єктного середовища підтримки програмних комплексів, що забезпечують можливість представлення досліджуваних середовищ підтримки програмних комплексів в якості набору моделей персональних портретів всіх тих суб'єктів, які безпосередньо формують досліджуване середовище підтримки, забезпечуючи та здійснюючи комплексну підтримку програмного продукту. Реалізовано апробацію розробленої моделі на прикладі вирішення прикладної практичної задачі ідентифікації миттєвого зрізу тенденції розвитку досліджуваного середовища по відношенні до задекларованої множини факторів впливу. Розглянуто перспективи подальших досліджень в контексті глобальної проблематики автоматизації й інтелектуалізації комплексної підтримки програмних продуктів.

Ключові слова: підтримка, програмний комплекс, середовище, суб'єкт, фактори впливу, модель.

Постановка проблеми. Основною досліджуваною в даній роботі науково-прикладною проблемою є необхідність простого та водночас дієвого механізму моделювання полісуб'єктних мультифакторних середовищ досліджуваних процесів комплексної підтримки програмних продуктів, здатних відобразити симбіоз таких основних складових цих процесів, а саме: об'єкта підтримки (самого програмного продукту), суб'єктів підтримки (персонал, що забезпечує та реалізує комплексну підтримку), а також факторів впливу (що впливають на суб'єктивізацію сприйняття об'єкта суб'єктами).

Аналіз досліджень та публікацій. В умовах сьогодення продовжуємо бути свідками невинного росту кількості та складності існуючих та розроблюваних програмних продуктів та комплексів. Паралельно з цим, на глобальному ринку інформаційних технологій (ІТ) зростає і конкуренція між цими продуктами. В свою чергу, це призводить до необхідності покращення (зокрема, й автоматизації) процесів комплексної підтримки програмних продуктів.

Комплексна підтримка програмних продуктів включає в себе будь-які активності в контексті підтримуваного програмного продукту, що забезпечують його життєздатність на ринку ІТ.

Серед ключових напрямків комплексної підтримки програмних продуктів можна виокремити такі найбільш популярні категорії як:

- опрацювання запитів користувачів (як зовнішніх, тобто «клієнтських», так і внутрішніх, тобто всередині персоналу компанії-розробника), представлених, до прикладу, в працях [1-3];
- тестування програмного забезпечення, передові дослідження якого викладені, зокрема в працях [4-6];
- автоматизація процесів життєвого циклу з використанням методологій Agile (висвітлених, зокрема, в працях [7,8]) та DevOps (висвітлених у працях [9,10]).

При цьому, важливим аспектом в контексті будь-яких (та, навіть, більше того, практично всіх) активностей і проявів комплексної підтримки програмних продуктів являються численні фактори впливу, зокрема деякі з них розглянуті в працях [11-13].

Водночас, в напрямку моделювання середовищ підтримки (як в контексті програмних продуктів, так і в цілому, зокрема в контексті систем прийняття рішень) вже існує ряд певних досліджень (зокрема, представлених в працях [14-15]), проте, на жаль, дані дослідження потребують значної актуалізації як (меншою мірою) у зв'язку із плином часу, що минув з моменту їх виходу, так і (значно більшою мірою) у зв'язку з необхідністю введення додаткових складових, зокрема таких сутностей як: об'єкт підтримки, суб'єкти підтримки, а також фактори впливу суб'єктивізації сприйняття об'єкта суб'єктами.

Таким чином, виникає необхідність розроблення спеціалізованого рішення: відповідної моделі (зокрема математичної, як і, впринципі загалом, будь-якої іншої) що дасть змогу врахувати актуальні вимоги в контексті поставленої задачі моделювання суб'єктних середовищ підтримки програмних комплексів.

Мета статті. Метою статті є розроблення моделі (зокрема, в даному випадку саме математичної моделі) суб'єктного середовища підтримки програмних комплексів, що реалізує формалізації представлення з забезпеченням можливості подальшого моделювання полісуб'єктних мультифакторних середовищ досліджуваних процесів комплексної підтримки програмних продуктів, та враховує ключову особливість суб'єктивізації сприйняття об'єкта(-ів) комплексної підтримки тими суб'єктами, котрі її безпосередньо забезпечують та реалізують.

Виклад основного матеріалу дослідження. Розроблення моделі суб'єктного середовища підтримки програмних комплексів можна умовно поділити на три основні етапи, детальніше представлені нижче по тексту.

Первинним етапом розроблення моделі суб'єктного середовища підтримки програмних комплексів є декларування структури будь-якого такого середовища – як множини персональних моделей, власне, самих тих суб'єктів, котрі формують відповідне середовище. Саме такий підхід забезпечує реалізацію необхідного повноцінного симбіозного зв'язку між сутностями «об'єкт підтримки» та «суб'єкти підтримки».

Наступним етапом є інкапсуляція в отриманий (на попередньому кроці) симбіозний зв'язок наступної фундаментальної сутності – тобто «факторів впливу» (суб'єктивізації сприйняття об'єкта суб'єктами).

Напрацювання, представлені в роботі [16], власне, забезпечують таку можливість, даючи змогу отримати конкретні числові показники присутності впливу кожного із факторів впливу на суб'єктивізацію сприйняття об'єкта підтримки кожним окремо взятим суб'єктом взаємодії, в рамках суб'єктного наявного середовища підтримки.

Таким чином, отримуємо модель персонального мультифакторного портрету суб'єкта підтримки, подану виразом (1) нижче:

$$S_{[j]}^{[e]} = [f_{[j]}^{[1]}, f_{[j]}^{[2]}, \dots, f_{[j]}^{[nf]}], \quad (1)$$

де: $S_{[j]}^{[e]}$ – персональний мультифакторний портрет j -ого суб'єкта в межах e -ого середовища підтримки; $f_{[j]}^{[nf]}$ – числовий показник присутності (тобто, іншими словами – конкретне значення величини впливу) nf -ого фактора впливу в рамках j -ого суб'єкта; nf – лічильник кількості задекларованих (в контексті досліджуваного середовища) факторів впливу.

Відповідно, «згорнута» модель середовища підтримки, сформованого суб'єктами підтримки, є не що інше як множина моделей цих суб'єктів, що може бути представлено наступним виразом (2):

$$MM^{[e]} = [S_{[1]}^{[e]}, S_{[2]}^{[e]}, \dots, S_{[ns]}^{[e]}], \quad (2)$$

де: $S_{[j]}^{[e]}$ – персональний мультифакторний портрет j -ого суб'єкта в межах e -ого середовища підтримки; ns – лічильник кількості суб'єктів в межах досліджуваного суб'єктного середовища підтримки.

Водночас, «розгорнута» форма розробленої математичної моделі суб'єктного середовища підтримки програмних комплексів може бути представлена виразом (3) нижче:

$$MM_{\{exp\}}^{[e]} = \begin{bmatrix} f_{[1]}^{[1]} & f_{[1]}^{[2]} & \dots & f_{[1]}^{[nf]} \\ f_{[2]}^{[1]} & f_{[2]}^{[2]} & \dots & f_{[2]}^{[nf]} \\ \dots & \dots & \dots & \dots \\ f_{[ns]}^{[1]} & f_{[ns]}^{[2]} & \dots & f_{[ns]}^{[nf]} \end{bmatrix}, \quad (3)$$

де: $f_{[j]}^{[i]}$ – числовий показник присутності (тобто, іншими словами – конкретне значення величини впливу) i -ого фактора впливу в рамках j -ого суб'єкта (в контексті досліджуваного суб'єктного середовища підтримки « e »); nf – лічильник кількості задекларованих (в контексті досліджуваного середовища) факторів впливу; ns – лічильник кількості суб'єктів в межах досліджуваного суб'єктного середовища підтримки.

Таким чином, розроблена математична модель суб'єктного середовища підтримки програмних комплексів забезпечує можливість ідентифікації всіх та кожного його (середовища) формують суб'єктів в контексті суб'єктивізації їх індивідуального персоналізованого сприйняття об'єкта комплексної підтримки (як безпосередньо самого підтримуваного програмного продукту/комплексу, так і процесів щодо його комплексної підтримки), зумовленого задекларованою множиною факторів впливу.

Розглянемо, також, апробацію розробленої математичної моделі на прикладі розв’язання прикладної практичної задачі ідентифікації миттєвого зрізу тенденції розвитку досліджуваного середовища по відношенні до задекларованої множини факторів впливу.

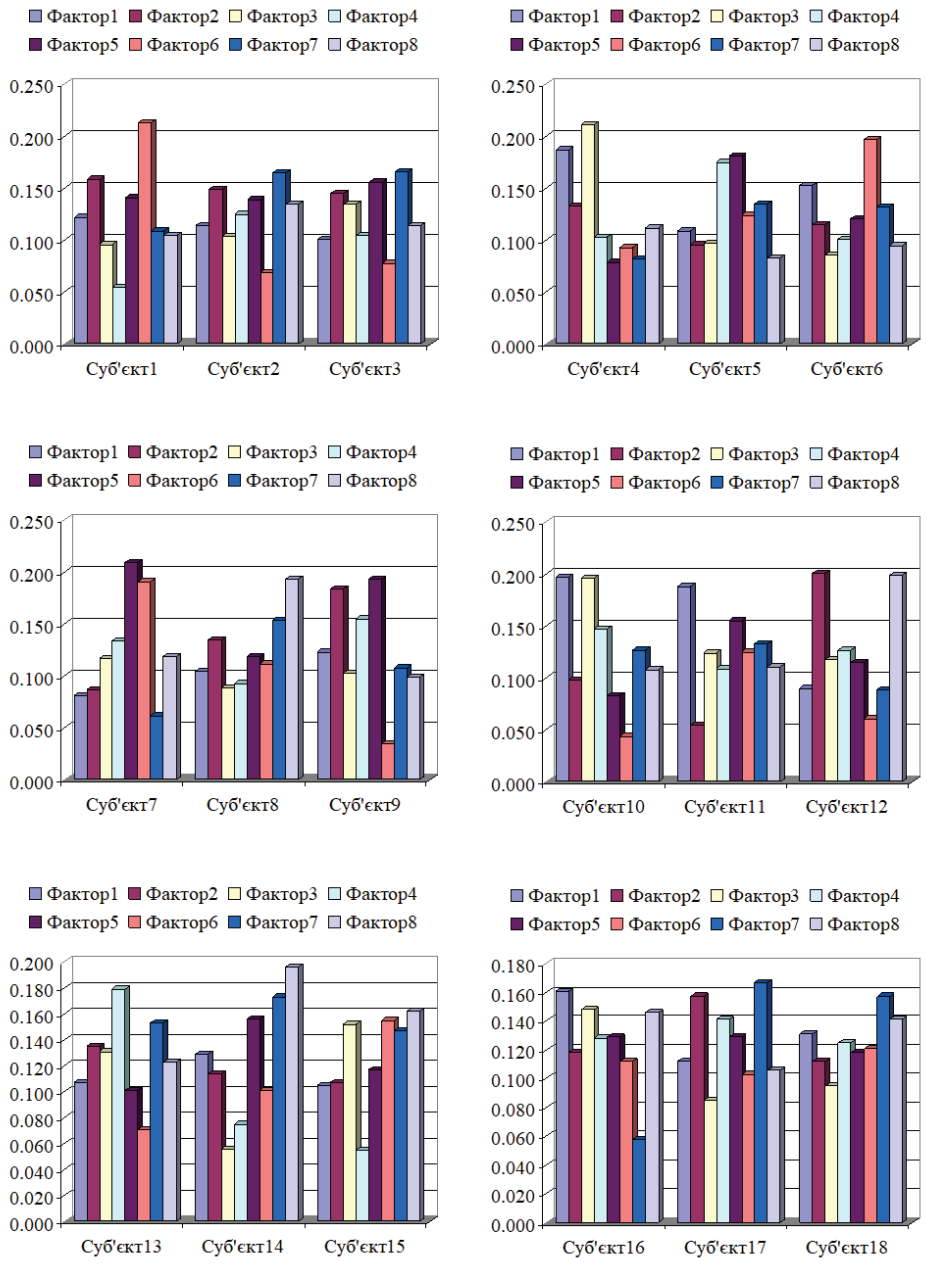


Рис. 1. Приклад графічної візуалізації розробленої математичної моделі суб'єктного середовища підтримки програмних комплексів

Зокрема, нижче, на рисунку 1, подано приклад графічної візуалізації розробленої математичної моделі суб'єктного середовища підтримки програмних комплексів, отриманого для середовища, досліджуваного в контексті розв'язання поставленої задачі.

Отож, найпершим етапом розв'язання поставленої прикладної практичної задачі є, власне, застосування розробленої математичної моделі з метою отримання розгорнутого полі-факторного портрету досліджуваного суб'єктного середовища підтримки.

Наступним етапом є вже, якраз, ідентифікація миттєвого зрізу тенденції розвитку досліджуваного середовища по відношенні до задекларованої множини факторів впливу, що, фактично, з чисто практичної точки зору, полягає в формуванні зростаючої пофакторної гістограми (за зростанням значень величин факторів впливу, тобто: від найменшого значення – до найбільшого) узагальненого портрету досліджуваного середовища.

Таким чином, на рисунку 2 отримаємо візуалізацію результатів розв'язання поставленої прикладної практичної задачі.

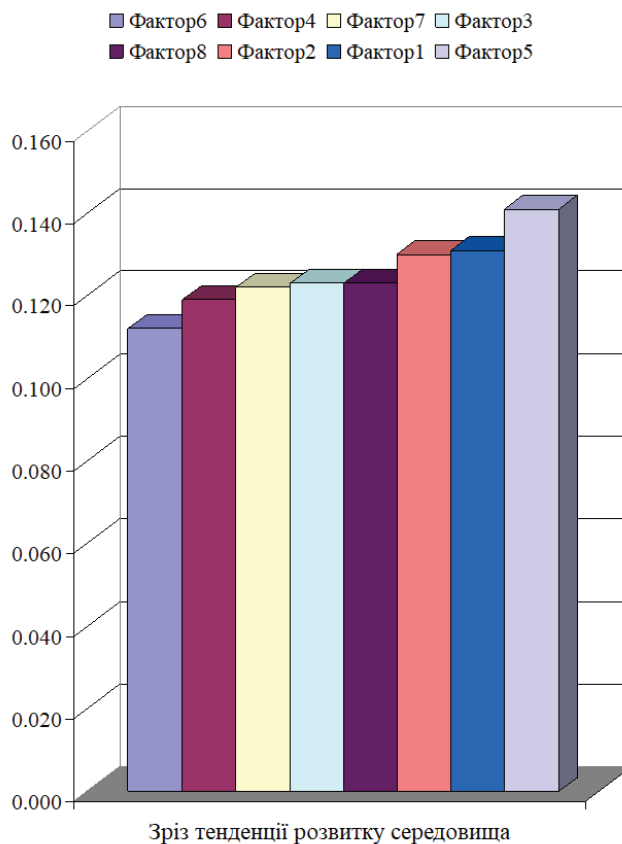


Рис. 2. Візуалізацію результатів розв'язання поставленої прикладної практичної задачі ідентифікації миттєвого зрізу тенденції розвитку досліджуваного середовища підтримки

Таким чином, апробація розробленої математичної моделі підтверджує її життєздатність та ефективність в контексті досліджуваної науково-прикладної проблеми моделювання середовищ підтримки програмних комплексів.

Водночас, перспективи подальших досліджень з допомогою розробленої та представленої математичної моделі бачимо саме в контексті глобальної проблематики автоматизації й інтелектуалізації комплексної підтримки програмних продуктів. Зокрема, існує доцільність розроблення додаткового програмного та алгоритмічного забезпечення (в тому числі, програмної реалізації) щодо розробленої математичної моделі з метою забезпечення, також і, можливостей комп'ютерного моделювання досліджуваних середовищ комплексної підтримки програмних продуктів, та виявлення відповідних потенційних шляхів її можливого покращення, вдосконалення, та подальшого еволюціонування.

Висновки. В даній роботі представлена розроблена математична модель суб'єктного середовища підтримки програмних комплексів, що забезпечують можливість представлення досліджуваних середовищ підтримки програмних комплексів в якості набору моделей персональних портретів всіх тих суб'єктів, які безпосередньо формують досліджуване середовище підтримки, забезпечуючи та здійснюючи комплексну підтримку програмного продукту.

Розроблена модель вирішує науково-прикладну проблему моделювання середовищ підтримки програмних комплексів, що базуються на основі складових суб'єктів, котрі, власне, безпосередньо забезпечують та реалізують цю комплексну підтримку.

Крім того, розроблена модель забезпечує можливість ідентифікації всіх та кожного його (середовища) формоутворюючих суб'єктів в контексті суб'єктивізації їх індивідуального персоналізованого сприйняття об'єкта комплексної підтримки (як безпосередньо самого підтримуваного програмного продукту/комплексу, так і процесів щодо його комплексної підтримки), зумовленого задекларованою множиною факторів впливу.

Здійснено апробацію розробленої моделі на прикладі вирішення прикладної практичної задачі ідентифікації миттєвого зрізу тенденції розвитку досліджуваного середовища по відношенні до задекларованої множини факторів впливу. А також, розглянуто перспективи подальших досліджень в контексті глобальної проблематики автоматизації й інтелектуалізації комплексної підтримки програмних продуктів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aguilar I.A., Polancos R. Implementation of roboc (Automated Incident Management System) to Human Centric Incident Ticketing Management Workflow. Proceedings of the 4th Asia Pacific Conference on Industrial Engineering and Operations Management. 2023. pp. 275-286. DOI: 10.46254/ap04.20230068.
2. Cooray T.M., Ariyaratne M.K., Perera, M.D. Automating incident categorization in a support request system. ERU Symposium 2021 Proceedings. 2021. pp. 1-2. DOI: 10.31705/eru.2021.12.

3. Ahmed S. et al. AI for Information Technology Operation (AIOPS): A review of it incident risk prediction. *Proceedings of the 9th International Conference on Soft Computing & Machine Intelligence (ISCMI)*. 2022. pp. 253–257. DOI: 10.1109/iscmi56532.2022.10068482.
4. Muqet A. et al. Assessing Quantum Extreme Learning Machines for Software Testing in Practice. *ACM Transactions on Software Engineering and Methodology*. 2024. Vol.37. Iss.4. article ID 121. pp. 1–21. DOI: 10.48550/arXiv.2410.15494.
5. Dandotiya S. Generative AI for Software Testing: Harnessing Large language models for automated and Intelligent Quality Assurance. *International Journal of Science and Research Archive*. 2025. Vol.14. Iss.1. pp. 1931–1935. DOI: 10.30574/ijrsra.2025.14.1.0266.
6. Rama S.M. Enhancing software testing efficiency through Test Diary implementation. *International Journal of Research in Computer Applications and Information Technology*. 2025. Vol.8. Iss.1. pp. 2889–2901. DOI: 10.34218/ijrcait_08_01_208.
7. Dugbartey A.N., Kehinde O. Optimizing Project Delivery Through Agile Methodologies: Balancing speed, collaboration and stakeholder engagement. *World Journal of Advanced Research and Reviews*. 2025. Vol.25. Iss.1. pp. 1237–1257. DOI: 10.30574/wjarr.2025.25.1.0193.
8. Marchesi L. et al. Responsible AI in Agile Software Engineering - An Industry Perspective. *XP 2024 Workshops, LNBIP 524*. 2025. pp. 33–41. DOI: 10.1007/978-3-031-72781-8_4.
9. Desmond O.S. Integrating Artificial Intelligence in DevOps: A Framework for Enhanced Agility and Operational Excellence. *International Journal of Innovative Research in Computer and Communication Engineering*. 2024. Vol.12. Iss.1. pp. 178–190. DOI: 10.15680/IJIRCCE.2024.1201075.
10. Joshi S. Introduction to generative AI and DevOps: Synergies, challenges and applications. *International Journal of Advanced Research in Science, Communication and Technology*. 2025. Vol.5. Iss.1. pp. 205–225. DOI: 10.48175/ijarset-23634.
11. Oliveira E. et al. Influence factors in software productivity - A tertiary literature review. *International Conferences on Software Engineering and Knowledge Engineering*. 2018. pp. 68–103. DOI: 10.18293/seke2018-149.
12. Machuca-Villegas L. et al. An Instrument for Measuring Perception about Social and Human Factors that Influence Software Development Productivity. *Journal of Universal Computer Science*. 2021. Vol.27. Iss.2. pp. 111–134. DOI: 10.3897/jucs.65102.
13. Ruiz M., Salanitri D. Understanding how and when human factors are used in the software process: A text-mining based literature Review. *Lecture Notes in Computer Science*. 2019. pp. 694–708. DOI: 10.1007/978-3-030-35333-9_54.
14. Finke A. Continuous Requirements Engineering Support Environment Model in Methodologically Heterogeneous Projects. In: Abramowicz, W. (eds) *Business Information Systems Workshops. BIS 2017. Lecture Notes in Business Information Processing*. 2017. Vol.303. Springer, Cham. 2017. pp. 207–215. DOI: 10.1007/978-3-319-69023-0_18.
15. Maschtera U. Modelling Support Environments. In: Sol, H.G., Takkenberg, C.A.T., De Vries Robbé, P.F. (eds) *Expert Systems and Artificial Intelligence in Decision Support Systems*. Springer, Dordrecht. 1987. pp. 151–173. DOI: 10.1007/978-94-009-3805-2_9.
16. Pukach A.I., Teslyuk V.M. Model of maximal weights inverse chains for the analysis of the influence factors of the software complexes support. *Radio Electronics, Computer Science, Control*. 2024. No.3, pp. 80–92. DOI: 10.15588/1607-3274-2024-3-8.

REFERENCES

1. Aguilar, I.A. & Polancos, R. (2023). Implementation of roboeoc (Automated Incident Management System) to Human Centric Incident Ticketing Management Workflow. Proceedings of the 4th Asia Pacific Conference on Industrial Engineering and Operations Management. 275-286. <https://doi.org/10.46254/ap04.20230068>.
2. Cooray, T.M., Ariyaratne, M.K. & Perera, M.D. (2021). Automating incident categorization in a support request system. ERU Symposium 2021 Proceedings. 1-2. <https://doi.org/10.31705/eru.2021.12>.
3. Ahmed, S. et al. (2022). AI for Information Technology Operation (AIOPS): A review of it incident risk prediction. Proceedings of the 9th International Conference on Soft Computing & Machine Intelligence (ISCMI), 253–257. <https://doi.org/10.1109/iscmi56532.2022.10068482>.
4. Muqet, A. et al. (2024). Assessing Quantum Extreme Learning Machines for Software Testing in Practice. ACM Transactions on Software Engineering and Methodology, 37(4), article ID 121, pp. 1-21. <https://doi.org/10.48550/arXiv.2410.15494>.
5. Dandotiya, S. (2025). Generative AI for Software Testing: Harnessing Large language models for automated and Intelligent Quality Assurance. International Journal of Science and Research Archive, 14(1), 1931–1935. <https://doi.org/10.30574/ijjsra.2025.14.1.0266>.
6. Rama, S.M. (2025). Enhancing software testing efficiency through Test Diary implementation. International Journal of Research in Computer Applications and Information Technology, 8(1), 2889–2901. https://doi.org/10.34218/ijrcait_08_01_208.
7. Dugbartey, A.N. & Kehinde, O. (2025). Optimizing Project Delivery Through Agile Methodologies: Balancing speed, collaboration and stakeholder engagement. World Journal of Advanced Research and Reviews, 25(1), 1237–1257. <https://doi.org/10.30574/wjarr.2025.25.1.0193>.
8. Marchesi, L. et al. (2025). Responsible AI in Agile Software Engineering - An Industry Perspective. (Eds.): XP 2024 Workshops, LNBIP 524, pp. 33–41. https://doi.org/10.1007/978-3-031-72781-8_4.
9. Desmond, O.S. (2024). Integrating Artificial Intelligence in DevOps: A Framework for Enhanced Agility and Operational Excellence. International Journal of Innovative Research in Computer and Communication Engineering. 12(1), 178-190. <https://doi.org/10.15680/IJIRCE.2024.1201075>.
10. 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/ijarset-23634>.
11. Oliveira, E. et al. (2018). Influence factors in software productivity - A tertiary literature review. International Conferences on Software Engineering and Knowledge Engineering, 68–103. <https://doi.org/10.18293/seke2018-149>.
12. Machuca-Villegas, L. et al. (2021). An Instrument for Measuring Perception about Social and Human Factors that Influence Software Development Productivity. Journal of Universal Computer Science, 27(2), 111-134. <https://doi.org/10.3897/jucs.65102>.
13. Ruiz, M. & Salanitri, D. (2019). Understanding how and when human factors are used in the software process: A text-mining based literature Review. Lecture Notes in Computer Science, 694–708. https://doi.org/10.1007/978-3-030-35333-9_54.
14. Finke, A. (2017). Continuous Requirements Engineering Support Environment Model in Methodologically Heterogeneous Projects. In: Abramowicz, W. (eds) Business Information

- Systems Workshops. BIS 2017. Lecture Notes in Business Information Processing, vol 303. Springer, Cham., pp.207–215. https://doi.org/10.1007/978-3-319-69023-0_18.
15. Maschtera, U. (1987). Modelling Support Environments. In: Sol, H.G., Takkenberg, C.A.T., De Vries Robbé, P.F. (eds) Expert Systems and Artificial Intelligence in Decision Support Systems. Springer, Dordrecht. pp.151-173. https://doi.org/10.1007/978-94-009-3805-2_9.
16. Pukach, A. I., & Teslyuk, V. M. (2024). Model of maximal weights inverse chains for the analysis of the influence factors of the software complexes support. Radio Electronics, Computer Science, Control, (3), 80-92. <https://doi.org/10.15588/1607-3274-2024-3-8>.

doi: 10.32403/1998-6912-2025-1-70-155-163

MATHEMATICAL MODEL OF SOFTWARE COMPLEXES' SUBJECT SUPPORT ENVIRONMENT

A. I. Pukach, V. M. Teslyuk

*ACS Department, Institute of Computer Sciences and Informational Technologies,
Lviv Polytechnic National University, Lviv, Ukraine*

The scientific and applied problem of modeling the software complexes' support environments (that are based on the constituent subjects, which, in fact, directly provide and implement this complex support) is considered in this research. Mathematical model of software complexes' subject support environment is developed, which provides possibility(-ies) of representation the researched support environments (of the supported software complexes) as a set of models of the personal portraits of all those subjects who directly form the researched support environment, providing and implementing comprehensive support for the considered software product. The approbation of the developed model has been carried out on the example of solving an appropriate relevant practical applied task of identifying an instantaneous slice of the development trend of the researched support environment in relation to the declared set of influencing factors. The prospects for further research(es) in the context of the global problems of automation and intellectualization of the comprehensive support of software products are considered as well. The developed mathematical model (of software complexes' subject support environment) provides the possibility(-ies) of identifying each and every one of its formative subjects in the context of subjectivization of their individual personalized perception of the object of complex support (both the supported software product/complex itself, as well as the processes related to its comprehensive support), determined by the declared set of influencing factors. Also, the developed model takes into account the key feature of the perception's subjectification of the object(s) (of comprehensive support) by those subjects (e.g. support personnel) which directly provide and implement this comprehensive support.

Keywords: support, program complex, environment, subject, impact factors, model.

Стаття надійшла до редакції 01.05.2025.

Received 01.05.2025.