

- 66-78, ISSN 0952-1976. (<https://doi.org/10.1016/j.engappai.2008.04.007>).
- [10]. Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, "VGGFace2: A Dataset for Recognising Faces across Pose and Age," 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, China, 2018, pp. 67-74, doi: 10.1109/FG.2018.00020.
- [11]. F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.
- [12]. T. Baltrušaitis, P. Robinson and L. P. Morency, "OpenFace: An open-source facial behavior analysis toolkit," 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 2016, pp. 1-10, doi: 10.1109/WACV.2016.7477553.
- [13]. J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 10, pp. 5962-5979, 1 Oct. 2022, doi: 10.1109/TPAMI.2021.3087709.
- [14]. Wang, J., Yuan, Y., Li, B., Yu, G., & Jian, S. (2018). SFace: An Efficient Network for Face Detection in Large Scale Variations. ArXiv, abs/1804.06559.
- [15]. Anil Jain, Karthik Nandakumar, Arun Ross, Score normalization in multimodal biometric systems, Pattern Recognition, Volume 38, Issue 12, 2005, Pages 2270-2285, ISSN 0031-3203. (<https://doi.org/10.1016/j.patcog.2005.01.012>).
- [16]. Perlibakas, V. (2004). Distance measures for PCA-based face recognition. Pattern Recognit. Lett., 25, pp. 711-724. (<https://doi.org/10.1016/j.patrec.2004.01.011>).
- [17]. Günther, F., & Fritsch, S. (2010). neuralnet: Training of Neural Networks. R J., 2, 30. (<https://doi.org/10.32614/RJ-2010-006>).
- [18]. Fasnacht, L. (2018). mmappickle: Python 3 module to store memory-mapped numpy array in pickle format. J. Open Source Softw., 3, 651. (<https://doi.org/10.21105/JOSS.00651>).
- [19]. Ye, H., Li, X., Yao, Y., & Tong, H. (2022). Towards Robust Neural Graph Collaborative Filtering via Structure Denoising and Embedding Perturbation. ACM Transactions on Information Systems, 41, pp. 1-28. (<https://doi.org/10.1145/3568396>).
- [20]. Nguyen, H., & Bai, L. (2010). Cosine Similarity Metric Learning for Face Verification., pp. 709-720. (https://doi.org/10.1007/978-3-642-19309-5_55).

RESEARCH ON THE EFFICIENCY OF COMBINED EMBEDDINGS FOR FACIAL VERIFICATION

In the era of digital authentication, facial verification systems have become a cornerstone of security protocols across various applications. This study explores the performance synergy from concatenated embeddings in enhancing biometric authentication accuracy. By leveraging the Celebrities in Frontal-Profile dataset (CFP), we investigate whether the fusion of embeddings generated by models such as VGG-Face, Facenet, OpenFace, ArcFace, and SFace can result in a more robust authentication process. The approach involves computing the L2 distance between normalized concatenated embeddings of an input face image and an anchor, thereby determining the authenticity of the individual. Experiments are designed to compare the performance of singular model embeddings against concatenated embeddings, employing metrics such as accuracy, False Acceptance Rate (FAR), and False Rejection Rate (FRR). The findings of this research could significantly contribute to the development of more secure and reliable facial verification systems by using multiple existing models without the need for new model research, designing, and training.

Keywords: Facial Verification, Biometric Authentication, Neural Networks, Concatenated Embeddings.

Ханін Денис Олегович, асистент кафедри захисту інформації Національного університету «Львівська політехніка».

Denys Khanin, Assistant at the Department of Information Security, National University "Lviv Polytechnic".

E-mail: denys.o.khanin@lpnu.ua.

Orcid ID: 0009-0001-4009-0202.

Отенко Віктор Іванович, к.т.н., доцент кафедри захисту інформації Національного університету «Львівська політехніка».

Viktor Otenko, PhD, Associate Professor at the Department of Information Security, National University "Lviv Polytechnic".

E-mail: viktor.i.otenko@lpnu.ua.

Orcid ID: 0000-0003-4781-7766.

DOI: [10.18372/2410-7840.26.18832](https://doi.org/10.18372/2410-7840.26.18832)

УДК 004.422.4:005.3:004.4

МУЛЬТИАСПЕКТНІСТЬ ТА СТРАТЕГІЧНЕ ПЛАНУВАННЯ ПРИ СТВОРЕНІ БАГАТОЦІЛЬОВИХ МОДЕЛЕЙ ОЦІНКИ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ

Антон Шантир, Ольга Зінченко, Максим Фесенко, Віктор Вишнівський

У сучасному інформаційному суспільстві проблема оцінювання якості програмних систем (ПС) є однією з ключових. Мета даної статті полягає в ретельному розгляді особливостей процесу оцінювання якості ПС з використанням принципів мультиаспектності та стратегічного планування. Для досягнення поставленої мети визначені чотири основні цілі. По-перше, стаття пропонує докладно розглянути основні етапи принципу мультиаспектності в підході до оцінювання якості ПС при створенні багатоцільових моделей якості. По-друге, здійснюється спроба надати математичні пояснення щодо того, як може бути представлена му-

льтиаспектність в моделюванні якості ПС. По-третє, стаття розглядає основні аспектні етапи стратегічного планування багатоцільових моделей із оцінки якості ПС. Нарешті, вона пропонує аналізувати фундаментальні формули, які можна використовувати для математичного вираження різних аспектів якості ПС в межах стратегічного планування та реалізації механізму оцінки якості ПС. Отримані результати статті сприятимуть подальшому розвитку області оцінки якості програмного забезпечення та підвищенню ефективності розробки та управління програмними проектами. Дослідження показало, що використання принципів мультиаспектності та стратегічного планування в оцінюванні якості програмних систем є дієвим підходом, що дозволяє створювати комплексні та адаптивні моделі оцінки. Ретельний аналіз основних етапів принципу мультиаспектності в підході до оцінювання якості ПС, а також розгляд аспектів стратегічного планування багатоцільових моделей, підкреслює важливість розуміння різноманітних потреб користувачів та інших зацікавлених сторін. Математичні пояснення стосовно представлення мультиаспектності в моделюванні якості ПС виявилися корисними для розробки формул та механізмів оцінки. Аналіз фундаментальних формул дав змогу розробити інструменти для математичного вираження різних аспектів якості ПС та їх використання у стратегічному плануванні. На підставі отриманих результатів можна зробити висновок, що застосування розглянутих підходів до оцінювання якості ПС сприятиме підвищенню ефективності управління проектами та розробки програмного забезпечення. Такі моделі дозволять враховувати різноманітні потреби та очікування різних зацікавлених сторін, що є ключовим для досягнення успіху в сучасному програмному індустрії.

Ключові слова: програмне забезпечення, метрики якості, потреби користувачів, інформаційні технології, методологія Agile, підтримка системи, математичний апарат.

ВСТУП

У сучасному цифровому віці програмне забезпечення (ПЗ) виявляє значний вплив на різні аспекти нашого життя, починаючи від корпоративного сектору та закінчуючи особистими смартфонами. Щодня з'являються нові програмні продукти, від яких користувачі очікують не тільки надійності та продуктивності, але й відповідності їхнім потребам та очікуванням. Це ставить перед розробниками програмного забезпечення та менеджерами проектів немалі виклики. Одним із головних завдань у цьому контексті є оцінка якості ПС. Шлях до створення ідеальних програмних продуктів вимагає глибокого розуміння якості та способів її вимірювання.

Сучасний стан у межах питання «Мультиаспектність та стратегічне планування при створенні багатоцільових моделей оцінки якості програмних систем» відображається в різноманітних аспектах.

По-перше, із зростанням складності програмного забезпечення та збільшенням очікувань користувачів зростає потреба в комплексному підході до оцінки якості. Стандартні методи оцінки, які фокусуються на окремих аспектах, можуть бути недостатніми для повного охоплення всіх вимог та очікувань.

Другим важливим аспектом є швидкість змін у технологічному ландшафті, що вимагає постійного оновлення та адаптації моделей оцінки. Стратегічне планування в цьому контексті виявляється ключовим, оскільки дозволяє враховувати майбутні тренди та вимоги користувачів ще на етапі розробки програмного продукту.

Третій аспект полягає у використанні передових методів та технологій, зокрема штучного інтелекту та аналізу даних, для побудови багатоаспектних моделей оцінки якості. ШІ може допомогти у зборі та обробці великих обсягів даних, а також у виявленні складних зв'язків між різними аспектами якості.

Нарешті, потрібно враховувати зростаючу увагу до питань безпеки та етики в інформаційних технологіях. Багатоаспектні моделі оцінки повинні враховувати ці аспекти, щоб забезпечити стабільність та довіру до програмного забезпечення.

У цілому, сучасний стан демонструє необхідність інтеграції принципів мультиаспектності та стратегічного планування для створення ефективних та адаптивних моделей оцінки якості програмних систем.

ПОСТАНОВКА ЗАДАЧІ

У даній статті ми докладно розглянемо процес оцінювання якості ПС з використанням двох ключових принципів: мультиаспектності та стратегічного планування. Ця тема стає надзвичайно актуальною, оскільки сучасне суспільство ставить все більше вимог до якості та функціональності програмного забезпечення. Із зростанням складності програмних систем зростає й необхідність у здатності враховувати різноманітні потреби та очікування користувачів.

З технічної точки зору порушена проблема потребує не лише розгляд особливості оцінки якості програмних систем, але й вказання на шляхи її оптимізації та покращення. Розгляд використання принципів мультиаспектності та стратегі-

чного планування допоможе зрозуміти, як досягти більшої ефективності та точності у процесі оцінки якості програмного забезпечення.

Додаткове дослідження цих принципів дозволить розробникам та менеджерам проєктів створювати більш адаптивні та комплексні моделі оцінки якості програмних систем. Розуміння основних етапів мультиаспектності та стратегічного планування дозволить ефективно аналізувати різні аспекти якості ПЗ, що в свою чергу сприятиме розробці інструментів та методологій для їх вдосконалення.

Окрім того, розгляд фундаментальних формул для математичного вираження різних аспектів якості програмних систем в межах стратегічного планування відкриває нові можливості для розробки інструментів автоматизованого аналізу та оцінки. Це дозволить забезпечити більш об'єктивні та консистентні результати в процесі розробки та управління програмними проєктами.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

В праці [1] відмічається, що при використанні принципу мультиаспектності у стратегічному плануванні при створенні багатоцільових моделей оцінки якості ПС враховуються різноманітні аспекти якості, такі як продуктивність, безпека, надійність тощо. Проте автори даної праці вказують, що нині є проблема, яка зводиться до недостатності повної узгодженості та однозначності визначення критеріїв оцінки для різних аспектів якості. Різні зацікавлені сторони можуть мати різний погляд на те, як кожен аспект якості слід оцінювати, що може призвести до конфліктів у визначенні метрик. Відповідно [2] врахування різноманітних аспектів якості за допомогою принципу мультиаспектності дозволяє створити комплексні моделі оцінки, які відображають потреби різних зацікавлених сторін та вимоги до ПС. Згідно позиції авторів вище зазначеної праці в межах практичного застосування врахування різних аспектів якості ПС на практиці дозволяє розробникам та менеджерам забезпечити більш повне і комплексне оцінювання рівня якості програмних продуктів. Проте даний підхід має потребу в додатковому часі та ресурсах для збору та аналізу різних аспектів якості може призвести до затримок у розробці. Також може виникнути складність у стратегічному визначенні пріоритетів між різними аспектами. Згідно позиції авторів праці [3] при розробці багатоцільових моделей оцінки якості з використанням принципу мультиаспектності важливо забезпечити баланс між різ-

ними аспектами, щоб уникнути перекоксу в оцінці. Відповідно поглядів авторів вище зазначеної праці стратегічне планування багатоцільових моделей з врахуванням принципу мультиаспектності дозволяє визначити оптимальні метрики та інструменти оцінки для досягнення поставлених цілей.

В праці [4] відмічається, що використання принципу мультиаспектності у стратегічному плануванні оцінки якості ПС важливо враховувати змінні умови та вимоги, щоб забезпечити адаптивність моделей. Згідно поглядів авторів вище зазначеної праці використання стратегічного планування для багатоцільових моделей оцінки якості дозволяє керівництву визначити пріоритети та ресурси, необхідні для досягнення цілей. Це в свою чергу надає можливість оптимізації використання ресурсів, управління ризиками та проблемами, забезпечення високої ефективності в процесі оцінки якості. Проте недостатня адаптивність до змін в умовах ринку, або технологічному середовищі може призвести до недооцінки деяких аспектів якості, або непродуктивного використання ресурсів.

В праці [5] відмічається, що мультиаспектність дозволяє уникнути однобічного підходу до оцінки якості ПС та забезпечити більш повне уявлення про їх стан, що в свою чергу призводить до зменшення ймовірності пропуску критичних проблем, покращення задоволення користувачів та підвищення загальної якості продукту. Проте збільшення складності процесу оцінки та аналізу, можливість конфліктів між різними аспектами, що вимагає більш складного стратегічного планування в межах управління та прийняття рішень. Авторським колективом праці [6] вказується, що стратегічне планування дозволяє виявити ключові напрямки удосконалення якості ПС та розробити стратегії для їх реалізації. На практиці забезпечення систематичного підходу до покращення якості, зменшення ризику втрат часу та ресурсів на непродуктивні заходи. Проте існує можливість неправильного визначення пріоритетів, що може призвести до витрат на непродуктивні заходи або недооцінки важливих аспектів.

В праці [7] зазначається, що мультиаспектність дозволяє враховувати різноманітні перспективи та інтереси зацікавлених сторін у стратегічному плануванні оцінки якості програмних систем. Проте при використанні принципу мультиаспектності у стратегічному плануванні багатоцільових моделей оцінки якості програмних систем необхідно уникати перекриття між аспектами та

забезпечити повноту покриття всіх важливих аспектів.

В праці [8] досить слушно відмічено, що стратегічне планування з використанням принципу мультиаспектності сприяє розробці гнучких та адаптивних моделей оцінки якості, які враховують змінні умови та потреби. Зокрема принцип мультиаспектності дозволяє враховувати різноманітні вимоги та очікування різних зацікавлених сторін у процесі стратегічного планування оцінки якості ПС.

В працях [9–12] відмічається, що використання штучного інтелекту (ШІ) може сприяти розробці багатоаспектних моделей оцінки якості, оскільки алгоритми машинного навчання можуть аналізувати різноманітні аспекти якості та допомагати у прийнятті стратегічних рішень. Проте незважаючи на потенційні переваги використання ШІ у розробці багатоаспектних моделей оцінки якості ПС, існують деякі недоліки і виклики з позиції стратегічного планування. Зокрема в праці [10] вказано, що ШІ може потребувати значної кількості даних для тренування та навчання моделей. Тому для ефективного використання машинного навчання у стратегічному плануванні, необхідно мати достатню кількість експертного знання для належного визначення цілей, параметрів та оцінок якості. В праці [11] відмічається потреба у великій кількості даних: моделі машинного навчання часто потребують великої кількості даних для навчання, щоб бути ефективними. Якщо дані недоступні або неповні, це може призвести до недостатньо точних, або недостовірних результатів. Автори праці [12] вказують на проблему інтерпретованості результатів: деякі алгоритми машинного навчання, особливо нейронні мережі, можуть бути складними для інтерпретації. Це може ускладнити розуміння причин виявлених проблем або рекомендацій для покращення якості. В праці [14] відмічається проблема наявності прихованих залежностей: машинне навчання може виявляти складні залежності та патерни в даних, які можуть бути непросто виявити або врахувати в стратегічному плануванні. Це може призвести до непередбачуваних результатів або неправильних висновків. В праці [13] порушується питання етики та відповідальності: використання машинного навчання у стратегічному плануванні може поставити питання етики та відповідальності. Наприклад, неправильне використання алгоритмів може призвести до біасів, або недопустимих рішень. Таким чином, при використанні ШІ в межах стратегічного планування у

створенні багатоаспектних моделей оцінки якості ПС необхідно уважно враховувати ці недоліки та виклики для досягнення ефективних та надійних результатів.

В праці [15] в межах піднятої тематичної спрямованості частково порушено питання, щодо забезпечення автоматизованого тестування та валідації: інструменти автоматизованого тестування можуть бути використані для перевірки різноманітних аспектів якості ПС, що відповідає підходу мультиаспектності. Незважаючи на те, що використання інструментів автоматизованого тестування може мати значні переваги з погляду забезпечення якості ПС, існують деякі недоліки та виклики з позиції стратегічного планування. Зокрема в праці [16] відмічається необхідність адаптації до різних аспектів якості: інструменти автоматизованого тестування можуть бути спроектовані для перевірки конкретних аспектів якості, але можуть бути не ефективними для інших аспектів. Це може вимагати додаткових зусиль для розробки або інтеграції додаткових інструментів для валідації всіх потрібних аспектів. В праці [17] підкреслюється проблема, щодо складності налаштування та підтримки: деякі інструменти автоматизованого тестування можуть вимагати значних зусиль для налаштування та підтримки. Це може включати в себе написання скриптів тестування, налаштування тестових середовищ та інші завдання, які можуть бути складними для управління на стратегічному рівні. В праці [18] відмічається необхідність обґрунтованості тестового покриття: при використанні автоматизованих тестів необхідно гарантувати, що тестове покриття відповідає потребам всіх аспектів якості програмних систем. Це вимагає уважного аналізу та планування, яке може бути витратними зусиллями на стратегічному рівні.

В праці [19] відмічено що інтеграція автоматизованих тестів у вже існуючі процеси розробки може бути складною, особливо якщо вони вимагають значних змін у процесах або інфраструктурі. Це може призвести до затримок у розробці та викликів у плануванні. Зважаючи авторів праць [12–19] варто відмітити, що незважаючи що автоматизоване тестування може бути важливим елементом стратегічного планування для забезпечення якості ПС, його використання також вимагає уваги до перерахованих недоліків для ефективного впровадження та управління. В праці [20] відмічено, що використання контейнерів та мікросервісів дозволяє виділяти різні аспекти функціональності програмного забезпечення в окремі

компоненти, що сприяє врахуванню різних аспектів якості. Проте незважаючи на те що використання контейнерів та мікросервісної архітектури може мати значні переваги з погляду розподілу функціональності ПЗ на окремі компоненти, існують деякі недоліки та виклики з позиції стратегічного планування. Зокрема в праці [21] відмічена проблема складності управління багатомірністю: з використанням контейнерів та мікросервісів збільшується кількість окремих компонентів, що може призвести до складнощів у керуванні та координації їх розвитку та взаємодії. В праці [5] відмічається проблема необхідності забезпечення стандартизації інтерфейсів: щоб забезпечити ефективну взаємодію між мікросервісами, необхідно визначити стандартизовані інтерфейси та протоколи, що може вимагати додаткових зусиль на стратегічному рівні.

В праці [22] вказується на проблему, щодо високої складності тестування та валідації: Збільшення кількості компонентів у системі також ускладнює процес тестування та валідації, оскільки необхідно переконатися, що кожен мікросервіс працює коректно як окрема частина системи, а також у взаємодії з іншими компонентами. Також в працях [1-12] частково була порушена проблема стратегічного врахування потенційної залежності від сторонніх постачальників: використання мікросервісної архітектури може вимагати використання сторонніх сервісів або ресурсів, що може призвести до залежності від сторонніх постачальників та ризику недоступності або змін у їх послугах, та відповідного розроблення додаткових стратегій щодо усунення даної проблеми. Авторами праці [15] порушено питання масштабованості та продуктивності: Хоча мікросервіси дозволяють гнучко масштабувати окремі компоненти системи, необхідно враховувати питання ефективності та продуктивності при розробці та управлінні масштабованими додатками.

Отже, використання контейнерів та мікросервісної архітектури для розподілу функціональності ПЗ може вимагати уваги до цих недоліків та викликів на стратегічному рівні для досягнення ефективної та надійної архітектури системи.

В працях [17-18] також порушено питання підвищення культури якості (Quality Culture): впровадження культури якості, такої як DevOps або Agile, може стимулювати врахування різних аспектів якості на різних етапах розробки програмного забезпечення. В праці [22] вказується про важливість врахування розширення оцінки якості на весь цикл життєвого циклу ПЗ: мульти-

аспектність передбачає оцінку якості на всіх етапах життєвого циклу програмного забезпечення, що відповідає ідеї розширення оцінки якості на весь цикл розробки.

Отже, останні досягнення у сфері розробки ПЗ підтримують принципові концепції мультиаспектності та стратегічного планування, які важливі для створення ефективних багатоцільових моделей оцінки якості програмних систем. Проте існує нагальна потреба в їх аналізі.

Мета статті полягає в розгляді особливостей оцінювання якості ПС з використанням принципів мультиаспектності та стратегічного планування.

Реалізація поставленої мети передбачає вирішення наступних цілей:

1. Розглянути основні етапи принципу мультиаспектності в підході до оцінювання якості ПС при створенні багатоцільових моделей якості;
2. Надати математичні пояснення стосовно того як може бути представлена мультиаспектність в моделюванні якості ПС;
3. Розглянути основні аспекtnі етапи стратегічного планування багатоцільових моделей із оцінки якості ПС;
4. Розглянути фундаментальні формули, які можна використовувати для математичного вираження різних аспектів якості ПС в межах стратегічного планування та реалізації механізму оцінки якості ПС.

ОСНОВНА ЧАСТИНА

Відповідно [1] принцип мультиаспектності в підході до оцінювання якості ПС полягає в тому, щоб розглядати якість з різних точок зору, або аспектів, замість того, щоб розглядати її, як єдиний показник.

В праці [2] розглядається можливість уникнення ієрархії, в межах принципу мультиаспектності, яка в свою чергу означає відмову від ієрархічної структури аспектів якості та розгляд аспектів на рівність. Згідно з [3] визначення аспектів якості включає ідентифікацію різних аспектів, які впливають на якість ПС. Наприклад, це можуть бути аспекти продуктивності, ефективності, надійності, зручності використання, безпеки тощо. У відповідності до [4] після визначення аспектів якості потрібно встановити критерії, за якими буде оцінюватися кожен аспект. Наприклад, для аспекту продуктивності критеріями можуть бути час відгуку системи, швидкість виконання тощо. Згідно з [6] на етапі збору даних в межах застосування принципу мультиаспектності в підході до оцінювання якості ПС здійснюється збір даних

про різні аспекти їх якості. Це може включати вимірювання різних параметрів, або оцінку за допомогою спеціальних інструментів чи тестів. У відповідності до [8] після збору даних проводиться їх аналіз і оцінка для кожного з аспектів якості. Цей етап дозволяє зрозуміти, наскільки задоволені критерії для кожного аспекту. Зважаючи на погляди авторів праці [10] варто погодитися із тим, що на основі аналізу результатів можуть бути прийняті рішення, щодо подальших дій. Наприклад, можуть бути заплановані покращення в певних аспектах якості, або розроблені стратегії для забезпечення балансу між різними аспектами. Згідно із [16] принцип мультиаспектності підтримує постійне вдосконалення ПС шляхом вдосконалення всіх аспектів якості, забезпечуючи таким чином загальну покращену якість програмного продукту. На рис.1. наведено основні етапи принципу мультиаспектності в підході до оцінювання якості ПС при створенні багатоцільових моделей якості. Наведені (рис. 1) етапи утворюють цикл постійного вдосконалення, який допомагає забезпечити високу якість ПС в умовах швидких змін середовища.

Таким чином розглянуті на рис.1 етапи допомагають забезпечити комплексний підхід до оцінювання якості ПС, уникнути вузького уявлення про якість та забезпечити покращення в усіх аспектах, які мають значення для користувачів.

Мультиаспектність в моделюванні якості ПС може бути представлена математично наступним чином: нехай $Q_{\text{мульти}}$ – це загальна метрика якості ПС із позиції врахування принципу мультиаспектності. Кожен аспект якості, такий, як продуктивність P , надійність R , ефективність E , зручність використання U та безпека S , може бути виражений окремою метрикою якості.

Тоді ми можемо виразити загальну метрику якості $Q_{\text{мульти}}$, як комбінацію різних аспектів якості з вагами, що відображають їхню важливість (1):

$$Q_{\text{мульти}} = w_P \cdot P + w_R \cdot R + w_E \cdot E + w_U \cdot U + w_S \cdot S, \quad (1)$$

де: w_P, w_R, w_E, w_U, w_S – ваги, що відображають важливість кожного аспекту якості; P, R, E, U, S – метрики якості для кожного аспекту.

Кожен аспект може бути додатково розкритий через визначення відповідних моделей, або параметрів, які враховують його вплив на загальну якість ПС. Цей підхід дозволяє систематично враховувати різні аспекти якості при моделюванні

ні ПС і дозволяє встановлювати пріоритети для поліпшення різних аспектів відповідно до їхнього важливості.

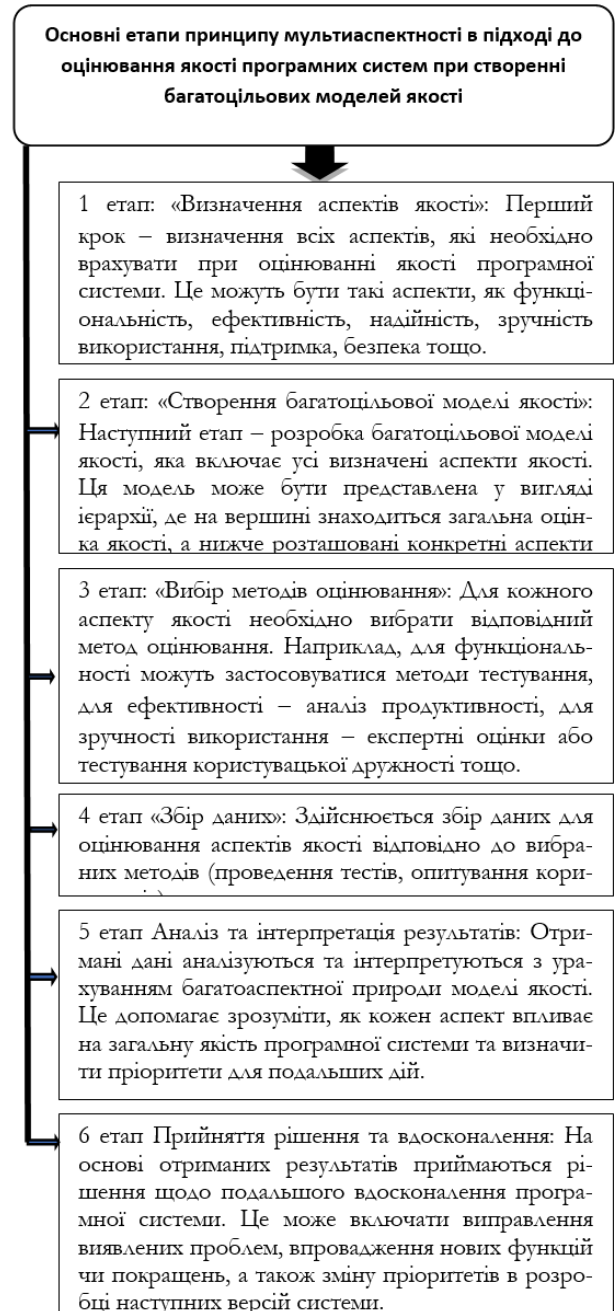


Рис.1. Основні етапи принципу мультиаспектності в підході до оцінювання якості ПС при створенні багатоцільових моделей якості

Контекстуальність в моделюванні якості ПС може бути математично виражена шляхом врахування специфічного контексту використання системи. Нехай $Q_{\text{кон}}$ це загальна метрика якості ПС із позиції врахування принципу контекстуальності. Контекст використання системи може впливати на рівень якості через різні характеристики, такі як внутрішнє середовище, характеристики користувачів та їхні очікування. Тоді ми мо-

жемо врахувати контекст шляхом додавання додаткових параметрів до $Q_{\text{Кон.}}$ (2):

$$Q_{\text{Кон.}} = f(Q_1, Q_2, \dots, Q_n, C_1, C_2, \dots, C_m), \quad (2)$$

де: Q_1, Q_2, \dots, Q_n – метрики якості ПС; C_1, C_2, \dots, C_m – параметри, що відображають контекст використання системи.

Зазначена вище функція може враховувати вплив різних аспектів контексту на загальну якість системи. Наприклад, характеристики користувачів можуть впливати на вимоги до зручності використання, внутрішнє середовище може впливати на надійність системи тощо.

Такий підхід дозволяє більш точно моделювати вплив контексту на якість системи і враховувати специфічні особливості її використання при оцінці якості ПС.

Інтеграція моделей якості може бути виражена математично через забезпечення взаємодії та зв'язку між різними метриками якості. Припустимо, що ми маємо кілька моделей якості Q_1, Q_2, \dots, Q_n , які оцінюють різні аспекти якості ПС. Для інтеграції цих моделей можемо використати наступний підхід: нехай $Q_{\text{Інтег.}}$ – загальна метрика якості ПС із позиції врахування принципу забезпечення інтеграції моделей якості. Ця метрика може бути представлена, як комбінація метрик якості на рівні моделей якості (3):

$$Q_{\text{Інтег.}} = f(Q_1, Q_2, \dots, Q_n). \quad (3)$$

Така функція f може бути визначена шляхом комбінування різних моделей якості з урахуванням їхньої вагомості або важливості. Наприклад, ми можемо використовувати зважену суму (4):

$$Q_{\text{Інтег.}} = w_1 \cdot Q_1 + w_2 \cdot Q_2 + \dots + w_n \cdot Q_n, \quad (4)$$

де: w_1, w_2, \dots, w_n – ваги, що відображають важливість кожної моделі якості із врахуванням аспектів перебігу їх інтеграції.

Варто відмітити, що зазначений вище підхід дозволяє поєднувати різні аспекти якості ПС та отримувати повний обсяг інформації про якість системи, враховуючи інтеграційний взаємозв'язок між різними моделями якості. Зважаючи на вище вказане інтеграція моделей якості дозволяє отримувати більш об'єктивну та комплексну оцінку якості ПС.

Вимірюваність в контексті моделювання принципів багатоцільових моделей якості ПС може бути математично виражена через використання вимірювальних показників для кожного аспекту якості. Припустимо, що для кожного аспекту якості у нас є відповідні метрики, які дозволяють об'

ективно оцінити рівень цього аспекту. Тоді ми можемо використовувати ці метрики для визначення відповідних вимірювальних показників.

Нехай M_i – це вимірювальний показник для аспекту якості i . Для оцінки загальної якості ПС $Q_{\text{Вим.}}$ із позиції врахування практичного забезпечення принципу вимірюваності ми можемо використовувати ці показники з врахуванням їхньої важливості, або ваги w у відповідності до виразу (5):

$$Q = w_1 \cdot M_1 + w_2 \cdot M_2 + \dots + w_n \cdot M_n, \quad (5)$$

де: w_1, w_2, \dots, w_n – ваги, що відображають важливість кожного вимірювального показника якості ПС.

Такий підхід дозволяє об'єктивно оцінювати рівні якості для різних аспектів програмної системи і порівнювати їх між собою. Вимірювальні показники дозволяють стандартизувати процес оцінки якості та забезпечують можливість об'єктивного порівняння рівнів якості між різними системами або версіями програмного забезпечення.

Аналіз ризиків в контексті стратегічного моделювання оцінки якості ПС може бути математично виражений через врахування можливих ризиків та розробку моделей для їх передбачення та управління. Для цього можна використати підходи, які побудовані на основі теорії ймовірностей та статистики.

Нехай R – це загальний ризик ПС, який може бути виражений, як сума ризиків (6):

$$R = R_1 + R_2 + \dots + R_n, \quad (6)$$

де n – кількість ризиків, які враховуються.

Варто зауважити, що при оцінці якості ПС враховуються різноманітні ризики, які можуть вплинути на їх ефективність, надійність та щодо їх відповідності до вимог експлуатації. Автори праці [9] відмічають, що важливо враховувати «ризик зміни вимог»: вимоги до ПС можуть змінюватися протягом процесу розробки, що може призвести до несумісності, або втрати функціональності.

Натомість автори праці [17] звертають увагу на потребу удосконалення та оптимізації шляхом врахування «ризик недооцінки та переоцінки ПС» та «ризик технічних помилок». Згідно їх позиції недооцінка чи переоцінка ресурсів (часу, бюджету, кадрів) може призвести до непрогнозованих затримок, або втрати якості, тоді, як неправильне функціонування програми через технічні помилки, або дефекти може негативно позначитися на якості.

Автори праці [8] рекомендують зверну увагу на врахування «ризиків безпеки ПС». З даного приводу варто відзначити, що недостатня захищеність від кіберзагроз може призвести до компрометації конфіденційності, цілісності, або доступності даних.

В праці [5] відмічається, потреба додаткового врахування «ризиків відмови у процесі експлуатації». Дану потребу дослідники пояснюють тим, що під час експлуатації програмного забезпечення можуть виникати відмови через різні причини, такі як перевантаження, несправність апаратного забезпечення тощо.

Натомість в праці [14] вказується, що додаткового має бути врахована ще ціла група ризиків, а саме: «ризик несумісності», «ризик продуктивності», «ризик відмови в резервуванні та відновленні», «ризик втрати даних», «ризик недостатньої документації», «ризик недостатньої користувацької підтримки», «ризик недооцінки користувацьких потреб».

Потребу врахування «ризиків несумісності» дослідники пов'язують із тим, що програмне забезпечення може бути несумісним із певними операційними системами, пристроями чи іншими програмами. Потребу врахування «ризиків продуктивності» пояснюють тим, що при експлуатації ПС під час великих завантажень можуть виникати проблеми з продуктивністю програми, такі як зниження швидкості відповіді або перевищення обсягу ресурсів.

Відповідно потреба врахування «ризиків відмови в резервуванні та відновленні» пояснюється тим, що недостатньо ефективна система резервного копіювання та відновлення може призвести до втрати даних та довести до недовіри користувачів. Тоді, як проблематика наявності «ризиків втрати даних» зв'язана із тим, що ПЗ може піддаватися ризику втрати даних через помилки у програмі, кібератаки, або інші причини.

Натомість потреба врахування «ризиків недостатньої документації» пояснюється тим, що недостатня, або некоректна документація може ускладнити розуміння та підтримку програмного забезпечення [3].

Проблематика врахування «ризиків недостатньої користувацької підтримки» пов'язана із тим, що недостатня користувацька підтримка може призвести до незадоволеності користувачів і втрати популярності продукту [10]

Відповідно [1] потреба врахування «ризиків недооцінки користувацьких потреб» пояснюється тим, що недостатнє врахування потреб користу-

вачів може призвести до розробки програмного забезпечення, яке не відповідає їхнім очікуванням.

Натомість автори праці [6] відмічають, що потрібно також враховувати «Ризик відсутності масштабованості». Зазначену потребу вони пояснюють тим, що на практиці недостатня масштабованість програми може ускладнити її адаптацію до зростаючих потреб, або обсягів користувачів.

Зважаючи на різнобічний захист прав розробників ПС автори праці [4] відмічають, що є нагальна потреба врахуванні на практиці «ризик юридичної відповідності»: невідповідність програмного забезпечення законодавчим вимогам, або ліцензійним умовам може призвести до юридичних проблем.

З технічної сторони експлуатації ПС автори праці [15] вказують на потребу врахування «ризиків недостатньої тестування ПС»: Недостатня кількість, або якість тестів може призвести до пропуску помилок або дефектів перед випуском програмного забезпечення в експлуатацію.

Узагальнюючи вище відмічене зі свого боку варто зазначити що наведений перелік ризиків не є вичерпними, і реальні експлуатаційні ризики можуть відрізнятися в залежності від специфіки проекту та середовища його розробки та використання.

Зважаючи на вище відмічене для кожного окремого ризику R_i можна розробити модель, яка оцінює ймовірність виникнення цього ризику та його вплив на якість програмної системи. Наприклад, якщо P_i – ймовірність виникнення ризику R_i , а I_i – його вплив на якість, то ризик R_i може бути виражений, як добуток цих параметрів (7):

$$R_i = P_i \cdot I_i. \quad (7)$$

Таким чином, загальний ризик R може бути виражений, як сума добутків ймовірностей та впливу кожного окремого ризику. Це дозволяє аналізувати різні ризики та їх вплив на якість програмної системи.

Для управління ризиками можна розробити стратегії, які базуються на виявлених ризиках та їх оцінці. Наприклад, можна розробити моделі для прогнозування ризиків, а також плани дій для мінімізації впливу цих ризиків на якість програмної системи.

Адаптивність моделей якості може бути математично виражена через здатність моделей відповідати змінним вимогам, технологічному середовищу та стратегіям розробки. Для цього можна використовувати методи адаптивних систем та алгоритмів машинного навчання.

Один із підходів до адаптивності - це використання методів машинного навчання для автоматичного адаптування моделей якості до нових умов. Наприклад, можна використовувати алгоритми регресії чи класифікації для прогнозування змін у якості системи в залежності від змінних вимог, або технологічного середовища.

Для математичного вираження адаптивності можна використовувати параметризовані моделі, де параметри можуть автоматично змінюватися або оновлюватися в залежності від нової інформації або зміни умов. Наприклад, можна використовувати параметризовані функції, де параметри регулюються алгоритмами оптимізації з метою максимізації або мінімізації певних критеріїв якості.

Такий підхід дозволяє моделям якості ПС адаптуватися до змінних умов і забезпечувати стабільну та ефективну оцінку якості ПС навіть у змінному середовищі розробки.

Для математичного вираження адаптивності багатоцільових моделей із оцінки якості ПС можна використати поняття параметризованих функцій.

Нехай $Q_{\text{адап.}}(x, \theta)$ – модель якості ПС із врахуванням аспектів адаптивності багатоцільових моделей із оцінки якості, де x – вхідні параметри системи (наприклад, характеристики системи), а θ – параметри моделі. Тоді адаптивність може бути виражена через автоматичне оновлення параметрів θ в залежності від нових даних, або зміни умов. Наприклад, ми можемо використовувати алгоритм градієнтного спуску для оновлення параметрів θ з метою мінімізації функції втрат або максимізації певного критерія якості (8):

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla J(\theta^{(t)}), \quad (8)$$

де $\theta^{(t)}$ – значення параметрів на кроці t , α – крок навчання, $J(\theta)$ – функція втрат, або критерій якості, $\nabla J(\theta)$ – градієнт цієї функції.

Такий підхід дозволяє моделям якості адаптуватися до змінних умов і забезпечувати стабільну та ефективну оцінку якості ПС навіть у змінному середовищі розробки.

Залучення зацікавлених сторін в процесі визначення моделей якості може бути математично виражено через інтеграцію думок та вимог різних зацікавлених сторін у визначення параметрів моделі якості.

Нехай $Q_{\text{зал.}}(x, \theta)$ модель якості ПС із врахуванням залучення зацікавлених сторін, де x – вхідні параметри системи (наприклад, характеристики системи), а θ – параметри моделі. Для враху-

вання думок та вимог різних зацікавлених сторін ми можемо використати підхід на основі багатокритеріальної оптимізації.

Допустимо, що у нас є кілька різних груп зацікавлених сторін, і кожна з них має свої вимоги до якості програмної системи. Ми можемо визначити окремі функції якості $J_1(\theta), J_2(\theta), \dots, J_k(\theta)$ для кожної групи зацікавлених сторін. Тоді можна вирішити задачу багатокритеріальної оптимізації для знаходження оптимальних параметрів моделі, які б найкращим чином задовольняли вимоги всіх груп зацікавлених сторін (9):

$$\arg \min_{\theta} [J_1(\theta), J_2(\theta), \dots, J_k(\theta)]. \quad (9)$$

Такий підхід дозволяє врахувати різні погляди та вимоги різних зацікавлених сторін при визначенні моделей якості ПС і забезпечує максимальну задоволеність усіх сторін.

Стратегічне планування якості ПС можна математично виразити через формулювання довгострокових цілей і використання моделей якості для досягнення цих цілей.

Нехай $Q_{\text{стратег.пл.}}(x, \theta)$ модель якості ПС із врахуванням аспектів стратегічного планування багатоцільових моделей із оцінки якості, де x – вхідні параметри системи (наприклад, характеристики системи), а θ – параметри моделі. Також, нехай G – довгострокова ціль якості, яку ми хочемо досягти. Одним з підходів до стратегічного планування якості є визначення оптимальних значень параметрів моделі θ , які максимізують досягнення цілі якості G . Це може бути сформульовано, як задача оптимізації: $\arg \max_{\theta} G(Q(x, \theta))$, де $G(Q(x, \theta))$ – функція, яка оцінює досягнення цілі якості на основі моделі якості.

Для досягнення цілей у межах удосконалення механізму із оцінки якості ПС також можуть бути встановлені обмеження, які потрібно врахувати при оптимізації. Наприклад, можуть бути обмеження на вартість, час виконання, чи інші параметри системи.

Такий підхід дозволяє враховувати якісні аспекти як стратегічний ресурс, формулювати довгострокові цілі якості та використовувати моделі якості для планування та досягнення цих цілей у ПС. Відповідно [11] стратегічне планування багатоцільових моделей із оцінки якості ПС є ключовим етапом для ефективного управління різноманітними проектами, програмами, або організаційними цілями.

Створення багатоцільових моделей якості на рівні моделювання можна математично сформу-

лювати через використання різних математичних підходів і методів. Автори праці [18] відмічають, що моделі якості можна виразити математично за допомогою функцій, рівнянь, або математичних виразів, які описують різні аспекти якості ПС. Наприклад, модель продуктивності може бути виражена через функцію, яка оцінює час виконання програми в залежності від різних параметрів системи (рис. 2).

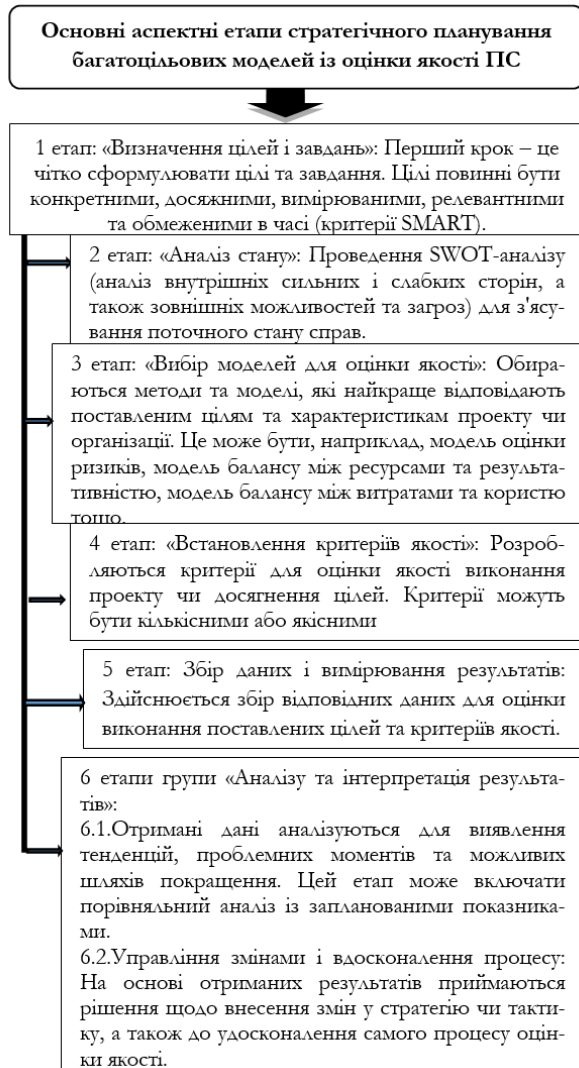


Рис.2. Основні аспектні етапи стратегічного планування багатоцільових моделей із оцінки якості ПС

Наведені на рис.2. аспекти спільно створюють цільовий фреймворк для стратегічного планування та оцінки якості, який допомагає досягти успіху в реалізації проектів та досягненні стратегічних цілей.

Автори праці [19] підкреслюють важливість врахування аспекту «багатоцільового направлення»: для розгляду ПС з різних аспектів можна використовувати різні моделі якості, які описують різні аспекти, такі як продуктивність, надійність, безпека тощо. Ці моделі можна математично ін-

тегрувати для отримання комплексного уявлення про якість системи.

Автори праці [20] відмічають на важливості врахування аспектів пов'язаних із «Оптимізацією ПС та відповідним їх управлінням»: математичні методи оптимізації можуть бути використані для вдосконалення якості ПС. Наприклад, можна використовувати алгоритми оптимізації для підбору оптимальних параметрів системи з метою максимізації продуктивності, або мінімізації кількості помилок в ПС.

Автори праці [16] рекомендують звернути увагу на врахуванні аспектів пов'язаних із «моделюванням залежностей»: математичні моделі дозволяють описати залежності між різними аспектами якості програмної системи. Наприклад, можна використовувати кореляційні аналізи для вивчення взаємозв'язків між рівнем продуктивності та рівнем надійності системи.

Отже, математичні методи можуть бути використані для створення, аналізу та управління багатоцільовими моделями якості на рівні моделювання, що дозволяє отримати комплексне уявлення про якість ПС і забезпечити їхнє покращення в різних аспектах

Ці принципи дозволяють використовувати математичні моделі для оцінки та аналізу різних аспектів якості ПС. Кожна модель представлена відповідною функцією, яка визначає залежність між метрикою якості та самою якістю

Нижче наведемо фундаментальні формули, які можна використовувати для математичного вираження різних аспектів якості ПС в межах стратегічного планування та реалізації механізму оцінки якості ПС.

У відповідності до [21] функцію продуктивності $P(x, \theta)$ багатокритеріальної моделі оцінки якості ПС можна розрахувати згідно формули (10):

$$P(x, \theta) = \frac{1}{T(x, \theta)}, \quad (10)$$

де $T(x, \theta)$ – час виконання програми з параметрами x та θ .

Автори праці [17] відмічають, що функцію надійності $R(x, \theta)$ багатокритеріальної моделі оцінки якості ПС можна розрахувати згідно формули (11):

$$R(x, \theta) = e^{-\lambda(x, \theta)t}, \quad (11)$$

де $\lambda(x, \theta)t$ - інтенсивність відмов ПС з параметрами з параметрами x та θ , t – час роботи ПС без відмов.

У відповідності до [16] функцію безпеки ПС $S(x, \theta)$ можна розрахувати згідно формули (12):

$$S(x, \theta) = 1 - \frac{V(x, \theta)}{N}, \quad (12)$$

де $V(x, \theta)$ – кількість вразливостей у системі з параметрами x та θ , N – загальна кількість вразливостей.

Згідно [15] функцію зручності використання ПС $U(x, \theta)$ можна розрахувати згідно формули (13):

$$U(x, \theta) = \frac{1}{c(x, \theta)}, \quad (13)$$

де (x, θ) – кількість кроків, які потрібно виконати користувачу для виконання певного завдання з параметрами x та θ .

Згідно [14] функцію планування ефективності використання ресурсів ПС $E(x, \theta)$ можна розрахувати згідно формули (14):

$$E(x, \theta) = \frac{B(x, \theta)}{Z(x, \theta)}, \quad (14)$$

де $B(x, \theta)$ – планування використання CPU, пам'яті, або інших ресурсів ПС, а $Z(x, \theta)$ час виконання програми з параметрами з параметрами x та θ .

Наведені вище формули можуть бути використані для оцінки різних аспектів якості ПС і можуть бути використані в багатоцільових моделях якості.

ВИСНОВКИ

Дослідження показало, що використання принципів мультиаспектності та стратегічного планування в оцінюванні якості програмних систем є дієвим підходом, що дозволяє створювати комплексні та адаптивні моделі оцінки. Ретельний аналіз основних етапів принципу мультиаспектності в підході до оцінювання якості ПС, а також розгляд аспектів стратегічного планування багатоцільових моделей, підкреслив важливість розуміння різноманітних потреб користувачів та інших зацікавлених сторін. Математичні пояснення стосовно представлення мультиаспектності в моделюванні якості ПС виявилися корисними для розробки формул та механізмів оцінки. Аналіз фундаментальних формул дав змогу розробити інструменти для математичного вираження різних аспектів якості ПС та їх використання у стратегічному плануванні. На підставі отриманих результатів можна зробити висновок, що застосування розглянутих підходів до оцінювання якості ПС сприятиме підвищенню ефективності управління проектами та розробки програмного забезпечення. Такі моделі дозволяють враховувати різ-

номанітні потреби та очікування різних зацікавлених сторін, що є ключовим для досягнення успіху в сучасному програмному індустрії.

ЛІТЕРАТУРА

- [1]. Aftab, S., Abbas, S., Ghazal, T. M., Ahmad, M., Hamadi, H. A., Yeun, C. Y., & Khan, M. A. (2023). A Cloud-Based Software Defect Prediction System Using Data and Decision-Level Machine Learning Fusion. *Mathematics*, 11(3), 632. <https://doi.org/10.3390/math11030632>.
- [2]. Yamming, C., & Shiyi, X. (2007). Exploration of complexity in software reliability. *Tsinghua Science & Technology*, 1(2), pp. 266-269.
- [3]. Sahu, K., & Srivastava, R. K. (2021). Predicting software bugs of newly and large datasets through a unified neuro-fuzzy approach: Reliability perspective. *Advances in Mathematics: Scientific Journal*, 10(1), pp. 543-555.
- [4]. Gordiciev, O., Kharchenko, V., Fominykh, N., & Sklyar, V. (2014). Evolution of Software Quality Models in Context of the Standard ISO 25010. In *The Ninth International Conference DepCoS-RELCO-MEX: Proceedings* (pp. 223-232). Wroclaw, Poland.
- [5]. Foidl, H., & Felderer, M. (2018). Integrating software quality models into risk-based testing. *Software Quality Journal*, 26, pp. 809-847.
- [6]. Aggarwal, G., & Gupta, D. V. (2013). Neural network approach to measure reliability of software modules: A review. *International Journal of Advances in Engineering Sciences*, 3(2), pp. 1-7.
- [7]. Azar, D., Harmanani, H., & Korkmaz, R. (2009). A hybrid heuristic approach to optimize rule-based software quality estimation models. *Information and Software Technology*, pp. 1365-1376. <https://doi.org/10.1016/j.infsof.2009.05.003>.
- [8]. Bharathi, R., & Selvarani, R. (2020). Hidden Markov model approach for software reliability estimation with logic error. *International Journal of Automation and Computing*, 17, pp. 305-320.
- [9]. Kapur, P. K., Pham, H., Anand, S., & Yadav, K. (2011). A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation. *IEEE Transactions on Reliability*, 60(1), pp. 331-340.
- [10]. Lee, M. (Year). Software quality factors and software quality metrics to enhance software quality assurance. *Current Journal of Applied Science and Technology*, 4(21), pp. 3069-3075.
- [11]. Miguel, J. P., Mauricio, D., & Rodriguez, G. (2014). A Review of Software Quality Models for the Evaluation of Software Product. *International Journal of Software Engineering & Applications*, 5(6), pp. 31-53.
- [12]. Pietrantuono, R. (2020). On the testing resource allocation problem: Research trends and perspectives. *Journal of Systems and Software*, 161, 42 p.
- [13]. Kemerer, C. F., & Paulk, M. C. (2009). The Impact of Design and Code Reviews on Software Quality:

- An Empirical Study Based on PSP Data. IEEE Transactions on Software Engineering, 35(4), pp. 534-550. doi:10.1109/TSE.2009.27.
- [14]. Lyu, M. R. (1997). Optimization Of Reliability Allocation and Testing Schedule for Software Systems. In Proceedings Eighth International Symposium on Software Reliability Engineering (ISSRE '97) (pp. 336-438).
- [15]. Musa, J. D. (1993). Operational Profiles in Software Reliability Engineering. IEEE Software, 10(2), pp. 14-32.
- [16]. Ohlsson, N., Helander, M., & Wohlin, C. (1996). Quality Improvement by Identification of Fault-Prone Modules using Software Design Metrics. In Proceedings Sixth International Conference on Software Quality (pp. 1-13).
- [17]. Letichevsky, A., Kapitonova, J., Letichevsky Jr., A., Volkov, V., Baranov, S., & Kotlyarov, V. (2005). Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications. In ISSRE 2004, WITUL, Rennes, 4 (pp. 112-142).
- [18]. Letichevsky A. (2005) Basic Protocols, Message Sequence Charts, and the Verification of Requerements Specifications / A. Letichevsky, J. Kapitonova, A.Letichevsky Jr., V. Volkov, S. Baranov, V. Kotlyarov // ISSRE 2004, WITUL, Rennes, 4 pp. 112-142.
- [19]. Helander, M. E. (1998). Planning Models for Software Reliability and Cost. IEEE Transactions on Software Engineering, 24(6), pp. 420-434.
- [20]. Helander M.E. Planning Models for Software Reliability and Cost/ M.E. Helander, M. Zhao, N. Ohlsson //IEEE Trans. Softw. Eng. 1998. V. 24. N. 6 pp. 420-434.
- [21]. Winther, R., Gran, B. A., & Dahll, G. (Eds.). (2005). Computer Safety, Reliability and Security: Proceedings of the 24th International Conference SAFE-COMP 2005, Friedrikstadt, Norway, September 28-30, 2005. 409 p.
- [22]. Besnard, D., Gacek, C., & Jones, B. (2006). Structure for Dependability: Computer Based Systems from Interdisciplinary Perspective. Springer Verlag London Limited. 304 p.

MULTIASPECTNESS AND STRATEGIC PLANNING IN DEVELOPING MULTI-OBJECTIVE MODELS FOR SOFTWARE QUALITY ASSESSMENT.

In the modern information society, the problem of assessing the quality of software systems (SS) is one of the key issues. The purpose of this article is to thoroughly examine the peculiarities of the process of assessing the quality of SS using the principles of multi-aspectness and strategic planning. To achieve this goal, four main objectives have been identified. Firstly, the article proposes to thoroughly consider the main stages of the multi-aspectness principle in the approach to assessing the quality of SS when creating multi-objective quality models. Secondly, an attempt is made to provide mathematical explanations regarding how multi-aspectness can be represented in the modeling of SS quality. Thirdly, the article examines the main aspectual stages of strategic planning of multi-objective models for SS quality assessment. Finally, it suggests analyzing fundamental formulas that can be used for mathematical expression of various aspects of SS quality within strategic planning and implementation of SS quality assessment mechanisms. The obtained results of the article will contribute to further development in the field of software quality assessment and increase the efficiency of software development and project management. The research has shown that the use of principles of multi-aspectness and strategic planning in assessing the quality of software systems is an effective approach that allows creating complex and adaptive assessment models. A thorough analysis of the main stages of the multi-aspectness principle in the approach to assessing SS quality, as well as consideration of aspects of strategic planning of multi-objective models, emphasized the importance of understanding the diverse needs of users and other stakeholders. Mathematical explanations regarding the representation of multi-aspectness in SS quality modeling have proven useful for developing formulas and assessment mechanisms. The analysis of fundamental formulas has enabled the development of tools for mathematical expression of various aspects of SS quality and their use in strategic planning. Based on the obtained results, it can be concluded that the application of the discussed approaches to SS quality assessment will contribute to increasing the efficiency of project management and software development. Such models will allow for consideration of diverse needs and expectations of various stakeholders, which is crucial for success in the modern software industry.

represented in the modeling of SS quality. Thirdly, the article examines the main aspectual stages of strategic planning of multi-objective models for SS quality assessment. Finally, it suggests analyzing fundamental formulas that can be used for mathematical expression of various aspects of SS quality within strategic planning and implementation of SS quality assessment mechanisms. The obtained results of the article will contribute to further development in the field of software quality assessment and increase the efficiency of software development and project management. The research has shown that the use of principles of multi-aspectness and strategic planning in assessing the quality of software systems is an effective approach that allows creating complex and adaptive assessment models. A thorough analysis of the main stages of the multi-aspectness principle in the approach to assessing SS quality, as well as consideration of aspects of strategic planning of multi-objective models, emphasized the importance of understanding the diverse needs of users and other stakeholders. Mathematical explanations regarding the representation of multi-aspectness in SS quality modeling have proven useful for developing formulas and assessment mechanisms. The analysis of fundamental formulas has enabled the development of tools for mathematical expression of various aspects of SS quality and their use in strategic planning. Based on the obtained results, it can be concluded that the application of the discussed approaches to SS quality assessment will contribute to increasing the efficiency of project management and software development. Such models will allow for consideration of diverse needs and expectations of various stakeholders, which is crucial for success in the modern software industry.

Keywords: software, quality metrics, user needs, information technology, Agile methodology, system support, mathematical framework.

Шантір Антон Сергійович, кандидат технічних наук, доцент кафедри штучного інтелекту Державного університету інформаційно-комунікаційних технологій.

Anton Shantyr, Ph.D., Associate Professor of the Department of artificial intelligence of the State University of Information and Communication Technologies.
E-mail: anton.shantyr@gmail.com.
Orcid ID: 0000-0002-0466-3659.

Зінченко Ольга Валеріївна, доктор технічних наук, доцент, завідувач кафедри штучного інтелекту Державного університету інформаційно-комунікаційних технологій.

Olha Zinchenko, Dc.S, Associate Professor, Head of the Department of artificial intelligence of the State University of Information and Communication Technologies.
E-mail: zinchenkoov@gmail.com.
Orcid ID: 0000-0002-3973-7814.

Фесенко Максим Анатолійович, кандидат технічних наук, доцент, доцент кафедри штучного інтелекту

Державного університету інформаційно-комунікаційних технологій.

Maksym Fesenko, Ph.D., Associate Professor, Associate Professor of the Department of artificial intelligence of the State University of Information and Communication Technologies.

E-mail: fesenkomaksim81@gmail.com.

Orcid ID: 0000-0001-8218-4154.

Вишнівський Віктор Вікторович, доктор технічних наук, професор, завідувач кафедри комп'ютерних наук Державного університету інформаційно-комунікаційних технологій.

Viktor Vyshnivskiy, Dc.S, Professor, Head of the Department of Computer Science of the State University of Information and Communication Technologies.

E-mail: vyshnivskiy.viktor@gmail.com.

Orcid ID: 0000-0003-1923-4344.

DOI: [10.18372/2410-7840.26.18833](https://doi.org/10.18372/2410-7840.26.18833)

УДК 004.056.5

РОЗРОБКА ЕФЕКТИВНИХ ЗАХОДІВ ВЕБ-БЕЗПЕКИ ДЛЯ МЕРЕЖІ ШЛЯХОМ ПРОВЕДЕННЯ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ OWASP

Марія Козловська, Андріян Піскозуб

З кожним кроком у розвитку технологій, веб-безпека стає більш актуальною складовою для забезпечення надійності та захисту мережевих систем. Зростання кількості кіберзагроз та потенційних порушень безпеки підкреслює необхідність удосконалення заходів захисту мережевих систем. Щоб допомогти розробникам та адміністраторам у цьому процесі, існує важливий інструмент – фреймворк OWASP (Open Web Application Security Project). Він надає широкий спектр інструментів, рекомендацій та ресурсів для забезпечення безпеки веб-додатків. Цей фреймворк допомагає розробникам перевірити веб-додатки на наявність потенційних вразливостей та знайти способи їх виправлення. Для кращого розуміння, можна уявити, що мережа – це будинок, а веб-додатки – це його двері та вікна. Якщо ці двері та вікна не міцно закриті, зловмисники можуть легко проникнути всередину та завдати шкоди. Отже для порівняння, можна сказати, що так само, як ви перевіряєте чи всі двері та вікна в мережі надійно захищені, OWASP надає засоби для перевірки веб-додатків на вразливості, що можуть бути використані зловмисниками для атак. Отже, використання фреймворку OWASP є важливим кроком у розробці ефективних заходів веб-безпеки для мережі, допомагає забезпечити надійність та захист системи від можливих кібератак та зловмисних дій.

Ключові слова: фреймворк OWASP, веб-безпека, тестування на проникнення, захист даних, кіберзахист.

ВСТУП

У сучасну цифрову епоху, коли інформація є джерелом життя бізнесу, забезпечення безпеки корпоративних мереж стало першочерговим завданням. Організації значною мірою покладаються на використання продуктів і послуг інформаційних технологій у своїй повсякденній діяльності, тому захист цих активів має вирішальне значення для їхнього успіху та загального виживання в умовах жорсткої конкуренції. Інформаційна безпека - це захист конфіденційних даних від несанкціонованого доступу, використання, розкриття, порушення, модифікації або знищення. Зважаючи на зростаючу складність і частоту кібератак, організаціям вкрай необхідно впроваджувати надійні заходи веб-безпеки для захисту своїх мереж і даних.

Оскільки Інтернет продовжує ставати повсюдним засобом комунікації та комерції, компанії все частіше ведуть бізнес онлайн. Однак зручність та ефективність цифрового ринку може

бути затьмарена зростаючими ризиками безпеки, які супроводжують онлайн-транзакції та зберігання даних. Численні переваги ведення бізнесу в Інтернеті супроводжуються складним завданням навігації в середовищі, насиченому потенційними загрозами, такими як системна корупція, шахрайство, крадіжки та віруси. Кіберзлочини призводять не лише до прямих фінансових втрат, вони також підривають довіру клієнтів і шкодять репутації бізнесу, що потенційно може призвести до довгострокових збитків.

Тому для організацій дуже важливо оцінювати та посилювати безпеку своїх мережевих систем. У цьому ландшафті цифрових загроз фреймворк OWASP стає основою безпеки веб-додатків, забезпечуючи важливий фундамент для розробки безпечних систем. Після ретельного тестування на проникнення ці стратегії утворюють надійний захист від різноманітних кіберзагроз, що дозволяє забезпечити безпеку системи. Конвергенція надійних фреймворків з комплекс-