ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

ВОЙЧУР ЮРІЙ ОЛЕКСІЙОВИЧ

УДК: 004.9 : 004.05

ДИСЕРТАЦІЯ

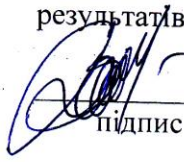
МЕТОДИ І ЗАСОБИ ПРОГНОЗУВАННЯ РІВНЯ ЯКОСТІ ТА БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

123 Комп'ютерна інженерія
(шифр і назва спеціальності)

12 Інформаційні технології
(галузь знань)

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних проваджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.



_____ Войчур Юрій Олексійович
підпис

Науковий керівник Медзатий Дмитро Миколайович, кандидат техн. наук, доцент

Хмельницький – 2024

АНОТАЦІЯ

Войчур Юрій Олексійович. Методи і засоби прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 123 – Комп'ютерна інженерія. – Хмельницький національний університет, Хмельницький, 2024.

На даний момент часу спостерігається суперечність між зростаючою відповідальністю, яка покладається на програмне забезпечення комп'ютерних систем (ПЗКС), та розширенням вимог до якості ПЗКС, з одного боку, і недосконалістю методів та засобів прогнозування якості та безпеки ПЗКС, яка особливо проявляється на ранніх етапах життєвого циклу, з іншого боку. Відтак, в дисертаційній роботі розв'язується актуальна науково-прикладна задача прогнозування і оцінювання рівня якості та безпеки ПЗКС на ранніх етапах життєвого циклу шляхом розроблення методів і засобів прогнозування рівня якості та безпеки ПЗКС.

Об'єкт дослідження – процеси прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем.

Предмет дослідження – методи та засоби прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем.

Метою дисертаційного дослідження є забезпечення оцінювання наявного набору вимог з позиції прогнозованого рівня якості та безпеки програмного забезпечення комп'ютерних систем, яке планується до реалізації за таким набором вимог, шляхом розроблення методів та засобів прогнозування рівня безпеки та якості програмного забезпечення комп'ютерних систем.

У дисертаційній роботі вперше розроблено метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп'ютерних систем, який відрізняється від відомих структуруванням вимог за атрибутами якості, та забезпечує вибір значень атрибутів якості ПЗ з природомовної специфікації вимог

до ПЗ, які використовуються для оцінювання значень характеристик якості ПЗ та для комплексного оцінювання якості ПЗ; розроблений метод дозволяє автоматизувати опрацювання вимог та мінімізувати участь людини у процесах аналізу вимог та оцінювання якості/безпеки.

У дисертаційній роботі також вперше розроблено метод прогнозування рівня якості програмного забезпечення комп'ютерних систем, який відрізняється від відомих тим, що дозволяє прогнозувати рівень якості майбутнього програмного забезпечення на основі значень атрибутів якості зі специфікації вимог. Таким чином, запропонований метод дозволяє порівнювати специфікації вимог, одразу відмовлятись від реалізації ПЗКС на основі невдалих специфікацій (економія коштів та часу, зменшення ймовірності провальних і проблемних проєктів) та виконувати ґрунтовний вибір специфікації для наступної реалізації ПЗКС саме високої якості (за умови, що помилки не будуть внесені при подальшому виконанні програмного проєкту).

В дисертації удосконалено метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем, який, на відміну від відомих, виконує аналіз припинення функціонування ПЗКС та функційних можливостей ПЗКС, які потенційно можуть бути вразливостями, та забезпечує висновок щодо того, чи є припинення функціонування ПЗКС відмовою, і, якщо так, то який тип вона має, а також забезпечує висновок, чи є функційна можливість вразливістю, і, якщо так, то який тип вона має.

В роботі удосконалено метод прогнозування рівня безпеки програмного забезпечення комп'ютерних систем, який, на відміну від відомих, встановлює залежність значення безпеки ПЗКС від значень атрибутів якості та формує прогнозоване числове значення безпеки ПЗКС на основі атрибутів, і забезпечує прогнозування рівня безпеки ПЗКС на основі отриманого числового значення, а також порівняння специфікацій вимог за прогнозованим рівнем безпеки розроблюваного ПЗКС та можливість відбраковування невдалих специфікацій.

Практичне значення отриманих результатів полягає в доведенні теоретичних результатів дисертаційної роботи до реалізації та у безпосередньому використанні

їх на підприємстві. Реалізована система прогнозування рівня якості ПЗКС надає користувачу прогнозовані оцінки восьми характеристик якості ПЗКС, комплексний показник прогнозованої якості ПЗКС та висновок про рівень якості майбутнього програмного забезпечення комп’ютерних систем, на основі якого можна виконати порівняння специфікацій вимог та обґрунтований вибір специфікації вимог для подальшої реалізації.

Реалізована система прогнозування рівня безпеки ПЗКС забезпечує аналіз вимог, на основі якого надає користувачу прогнозовану оцінку безпеки ПЗКС (як характеристики якості) та висновок про рівень безпеки майбутнього ПЗКС, на основі якого можна виконати порівняння специфікацій вимог та обґрунтований вибір специфікації для подальшої реалізації.

Реалізована система ідентифікації та класифікації відмов і вразливостей ПЗКС надає висновок щодо того, чи є припинення функціонування ПЗКС відмовою ПЗКС; висновок щодо того, чи є функційна можливість ПЗКС вразливістю ПЗКС; висновок про тип відмови та/або вразливості за їх наявності.

Результати дисертаційної роботи впроваджено у (Додаток Б): ПП «Авіві» (акт впровадження від 10.01.2024 р.); ТОВ «Деймос» (акт впровадження від 15.02.2024 р.); ГО «ІТ Кластер м. Хмельницького» (акт впровадження від 22.03.2024 р.); у навчальному процесі Хмельницького національного університету (акт впровадження від 13.05.2024 р.); при виконанні держбюджетних тем Хмельницького національного університету «Самоорганізована розподілена система виявлення зловмисного програмного забезпечення в комп’ютерних мережах» (ДР № 0121U109936), «Система виявлення ЗПЗ та комп'ютерних атак в корпоративних мережах з використанням хибних об'єктів атак та пасток» (ДР № 0124U000980).

Ключові слова: програмне забезпечення комп’ютерних систем, якість програмного забезпечення комп’ютерних систем, безпека програмного забезпечення комп’ютерних систем, вразливість програмного забезпечення, відмова програмного забезпечення.

ANNOTATION

Voichur Yurii. Methods and tools for predicting the level of quality and security of computer systems' software. – Manuscript copyright.

Thesis on competition of scientific degree of Doctor of Philosophy by specialty 123 – Computer Engineering. – Khmelnytskyi National University, Khmelnytskyi, 2024.

Currently, we have the contradiction between the increasing responsibility, assigned to computer systems' software, and the expansion of software quality requirements, on the one hand, and the incompleteness of methods and tools for predicting the computer systems' software quality and security, especially in the early stages of the software project's life cycle, on the other hand. Therefore, the dissertation solves the actual scientific and applied problem of predicting and assessing the level of quality and security of software at the life cycle's early stages by developing methods and tools for predicting the level of quality and security of computer systems' software.

The object of research is the processes of predicting the level of quality and security of computer systems' software.

Subject of research are methods and tools for predicting the level of quality and security of computer systems' software.

The aim of the dissertation research is to provide an assessment of the existing set of requirements from the point of view of the predicted level of quality and security of computer systems' software that is planned to be implemented according to such a set of requirements by developing methods and tools for predicting the level of security and quality of computer systems' software.

In the dissertation, the method for analyzing requirements for computer systems' software for the search for values of quality attributes was first time developed, which differs from the known ones by imposing certain restrictions on the formation of a specification of software requirements by structuring requirements containing quality attributes and provides a selection of values of attributes of software quality from natural language specification of software requirements, which are used to evaluate the values of software quality characteristics and for a comprehensive assessment of software quality; the developed method is important for automating the processing of requirements and minimizing

subjective influence and human participation in the processes of information processing and software quality and security assessment.

The thesis also first developed a method for predicting the quality level of computer systems' software, which differs from the known methods in that it allows predicting the value of quality level of developed software on the basis of the processing of values of attributes of software quality, which are in the specification of requirements. Thus, the proposed method allows comparing software requirements specifications, immediately refusing to implement a software system based on unsuccessful specifications (reducing the likelihood of failed and challenged projects, saving time and money), and making a well-grounded choice of specifications for the further computer systems' software realization and implementation with high quality (if errors and bugs are not introduced at the next work during computer systems' software development).

The dissertation further develops the method of identifying and classifying failures and vulnerabilities, which, unlike the known ones, identifies and classifies failures and vulnerabilities and provides a conclusion as to whether a failure has occurred, and, if a failure has occurred, the user is given its type. In addition, the developed method of ensuring the security of computer system software by identifying and classifying failures and vulnerabilities provides a conclusion as to whether a functional capability is a vulnerability, and, if the functional capability is a vulnerability, the user is given its type.

The dissertation improves the method for determining the security level of computer systems' software, which, unlike the known, establishes the dependence of the value of security of the software on values of the quality attributes and generates a predicted numerical value of the security of the software on the basis of attributes, and provides prediction of the security level of the software based on the obtained numerical value, and also provides comparison of requirements specifications according to the predicted security level of the developed software and the possibility of rejecting unsuccessful specifications.

The practical significance of the obtained results is to bring the theoretical results of the thesis to implementation and use its at the enterprises. The realized system for predicting the level of quality of computer systems' software provides the predicted estimates of 8 quality characteristics of the computer systems' software, a comprehensive indicator of the

computer systems' software quality and a conclusion about the level of developed computer systems’ software quality, on the basis of which compares the specifications of requirements for computer systems’ software and make a well-grounded choice of the requirements specification for next implementation.

The implemented system for predicting the security level of the computer systems' software provides the predicted assessment of the value of security of the computer systems' software (as a quality characteristic) and a conclusion about the security level of the future computer systems' software, based on which compares the specifications of requirements and make a well-grounded choice of the requirements specification for next implementation.

The implemented system for identifying and classifying failures and vulnerabilities provides a conclusion on the presence/absence of failure(s) of the computer systems' software; a conclusion on the presence/absence of vulnerability(s) of the computer systems' software; a conclusion on the type of failure and the type of vulnerability.

The results of the dissertation are implemented in (Appendix B): PE “Avivi” (implementation act of 10.01.2024); LLC “Deymos” (implementation act of 15.02.2024); NGO “IT Cluster of Khmelnytskyi” (implementation act of 22.03.2024); in the educational process of Khmelnytskyi National University (implementation act of 13.05.2024); in the implementation of state budget topics of Khmelnytskyi National University “Self-organized distributed system for detecting malicious software in computer networks” (State Research Project No. 0121U109936), “System for detecting malware and computer attacks in corporate networks using false attack objects and traps” (State Research Project No. 0124U000980).

Key words: computer systems’ software, computer systems’ software quality, computer systems’ software security, software vulnerability, software failure.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у періодичних виданнях, включених до категорії «А» Переліку наукових фахових видань України, або у закордонних виданнях, проіндексованих у базах даних Web of Science Core Collection та/або Scopus:

1. Hovorushchenko T., Medzatyi D., Voichur Yu., Lebiga M. Method for forecasting the level of software quality based on quality attributes. *Journal of Intelligent & Fuzzy Systems*. 2023. vol. 44, no. 3, pp. 3891-3905. (https://doi.org/10.3233/JIFS-222394) *(індексована в наукометричних базах Scopus, Web of Science (Q2 by Scimago Journal & Country Rank))*

2. Hovorushchenko T., Voichur Yu., Medzatyi D., Boyarchuk A. Information Technology for Prediction Software Quality Level. *Radioelectronic and Computer Systems*. 2023. No. 3. Pp. 238-254. (https://doi.org/10.32620/reks.2023.3.19) *(індексована в наукометричній базі Scopus (Q3 by Scimago Journal & Country Rank))*

3. E. Zaitseva, T. Hovorushchenko, O. Pavlova, Yu. Voichur. Identifying the Mutual Correlations and Evaluating the Weights of Factors and Consequences of Mobile Applications Insecurity. *Systems*. 2023. Vol. 11. Issue 5. Article No. 242. (https://doi.org/10.3390/systems11050242) *(індексована в наукометричній базі Scopus (Q3 by Scimago Journal & Country Rank))*

Статті у наукових виданнях, включених до Переліку наукових фахових видань України:

4. Медзатий Д.М., Войчур Ю.О., Войчур О.Ю. Технологія ідентифікації та класифікації відмов і вразливостей програмного забезпечення. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2023. №1. С. 53-57. (https://doi.org/10.31891/2219-9365-2023-73-1-8)

5. Ю. Войчур, Д. Медзатий. Метод аналізу вимог до програмного забезпечення на предмет пошуку значень атрибутів якості. Вимірювальна та обчислювальна техніка в технологічних процесах. 2024. №1. С.146-151. (https://doi.org/10.31891/2219-9365-2024-77-18)

Публікації, які засвідчують апробацію матеріалів дисертації:

6. Hovorushchenko T., Popov P., Medzatyi D., Voichur Yu. Method and Technology for Ensuring the Software Security by Identifying and Classifying the Failures and Vulnerabilities. *CEUR-WS*. 2022. Vol. 3309. Pp. 338-348. *(індексована в наукометричній базі Scopus)*

7. Hovorushchenko T., Voichur Yu. Method for Determining the Number of Lines of Manually Written Source Code. *CEUR-WS*. 2023. Vol. 3628. Pp. 520-525. *(індексована в наукометричній базі Scopus)*

8. T. Hovorushchenko, Yu. Voichur, D. Medzatyi, A. Boyarchuk, A. Hnatchuk. Method for Determining the Security Level of Software. *CEUR-WS*. 2024. Vol. 3675. Pp. 72-85. *(індексована в наукометричній базі Scopus)*

Публікації, які додатково відображають наукові результати дисертації:

9. А. с. 113734 Україна. Нейромережна модель прогнозування якості програмного забезпечення / Т. О. Говорущенко, М. М. Лебіга, Ю. О. Войчур. 2022.

10. А. с. 118851 Україна. Метод прогнозування рівня якості програмного забезпечення на основі атрибутів якості / Т. О. Говорущенко, М. М. Лебіга, Ю. О. Войчур. 2023.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПЗ – програмне забезпечення

КС – комп’ютерна система

ПЗКС – програмне забезпечення комп’ютерних систем

ЖЦ – життєвий цикл

ШНМ – штучна нейронна мережа

ВСТУП

Актуальність теми. Зростаючий інтерес до галузі інформаційних технологій приводить до створення великої кількості програмного забезпечення (ПЗ) різного функціонального призначення, складність та обсяг якого постійно збільшуються [1].

Ефективне функціонування комп'ютерних систем визначається якістю їх програмної, математичної, технічної, правової, інформаційної складових. Однією з найважливіших складових є програмне забезпечення, менеджмент якості якого суттєво впливає на якість комп'ютерної системи [2].

Постійний розвиток існуючих комп'ютерних систем (КС) та створення нових КС з використанням хмарних технологій, штучного інтелекту, доповненої та віртуальної реальності підвищують вимоги до процесів прогнозування та оцінювання якості програмного забезпечення комп'ютерних систем.

Ринок програмного забезпечення для комп'ютерних систем (ПЗКС) зростає дуже швидко, що робить його найбільш динамічним у галузі інформаційних технологій. Програмне забезпечення стає все більш складним, і на нього покладається все більша відповідальність. Зі зростанням надійності апаратного забезпечення та зростаючою складністю програмного забезпечення, якість і безпека програмного забезпечення комп'ютерних систем викликає дедалі більше занепокоєння як у розробників, так і у користувачів, щонайменше виходячи з прагнення досягти поставлені бізнес-цілі.

Сучасна індустрія ПЗКС характеризується високою конкуренцією. Із зростанням глобалізації та вільних ринків користувачі програмного забезпечення стають все більш впливовими, маючи можливість купувати або відмовлятися від програмного забезпечення через його незадовільну якість. Очевидно, що найбільш зацікавленим у якості програмного забезпечення є кінцевий користувач, який його використовує. Забезпечення необхідного рівня якості та безпеки програмного забезпечення стало стратегічним завданням у життєвому циклі сучасних проєктів програмного забезпечення комп'ютерних систем.

Оцінювання та прогнозування якості та безпеки ПЗКС є одним з ключових завдань, що виникають при розробці програмного забезпечення комп'ютерних систем через необхідність всебічного врахування його впливу на якість апаратних та програмних компонентів критично важливих систем.

Якість ПЗКС – це ступінь, в якому ПЗ відповідає потребам користувача при його застосуванні у визначених умовах [3]. Безпека ПЗКС як характеристика якості ПЗ розглядається як забезпечення захисту від потенційних загроз (несанкціонований доступ, зловживання даними тощо) [3].

Розвиток сучасних технологій та методологій проєктування та розроблення ПЗКС вимагає динамічного вдосконалення засобів оцінки, а особливо засобів прогнозування якості та безпеки ПЗКС на ранніх етапах життєвого циклу, щоб запровадити превентивні заходи для зменшення кількості помилок та збоїв у програмному забезпеченні; щоб обгрунтовано обирати той чи інший набір вимог з множини різних альтернатив, запропонованих різними розробниками для розв'язку однієї й тієї ж задачі; щоб визначити, чи можна розробити якісне ПЗ на основі зібраних вимог (оскільки версії ПЗ, створені різними розробниками за однаковими вимогами, зазвичай містять ряд спільних помилок, зумовлених помилками або неточностями в самих вимогах) [4-8].

Питанням прогнозування та оцінювання безпеки і якості програмного забезпечення комп'ютерних систем присвячено ряд досліджень як українських, так і іноземних вчених: Д. Маєвського [9-11], О. Гордєєва [12-16], В. Яковини [17-20], Н. Лисої [21-23], А. Кудряшової [24-27], О. Одарущенка [28-31], О. Пастуха [32-34], Н. Падхі (Padhy) [35-38], І.Арори (Arora) [39, 40], С. Гойала (Goyal) [41-45], С. Хуанга (Huang) [46, 47], К. Шеорана (Sheoran) [48, 49], М. Хана (Khan) [50, 51], С.-Я. Чо (Cho) [52, 53], С. Ямади (Yamada) [54-57], Х. Саді (Sadya) [58, 59].

Досліджені методи та засоби прогнозування безпеки та якості ПЗКС мають великий потенціал для розв'язання різних задач, можуть бути використані в різних контекстах, проте вони не забезпечують обчислення і не задають залежності значень характеристик якості від значень атрибутів, не забезпечують обчислення і не задають залежності значення якості від значень характеристик якості та не

забезпечують прогнозування рівня якості та/або безпеки ПЗКС на основі отриманих кількісних значень якості та/або безпеки.

Отже, на даний момент часу спостерігається суперечність між зростаючою відповідальністю, яка покладається на програмне забезпечення комп'ютерних систем (ПЗКС), та розширенням вимог до якості ПЗКС, з одного боку, і недосконалістю методів та засобів прогнозування якості та безпеки ПЗКС, яка особливо проявляється на ранніх етапах життєвого циклу, з іншого боку. Відтак, прогнозування і оцінювання рівня якості та безпеки програмного забезпечення комп'ютерних систем на основі атрибутів якості на початкових етапах життєвого циклу є *актуальною науково-прикладною задачею,* одним із шляхів розв'язання якої є розроблення методів і засобів прогнозування рівня якості та безпеки ПЗКС.

Зазначена науково-прикладна задача відповідає предметній області Стандарту вищої освіти України зі спеціальності 123 – Комп'ютерна інженерія для третього (освітньо-наукового) рівня вищої освіти, зокрема, такому об'єкту вивчення та діяльності, як «процедури та засоби підтримки та керування життєвим циклом, забезпеченням якості, надійності та безпеки».

Зв'язок роботи з науковими програмами, планами, темами. Дослідження, результати яких викладено в дисертації, виконано під час виконання науково-дослідних робіт за держбюджетними темами Хмельницького національного університету «Самоорганізована розподілена система виявлення зловмисного програмного забезпечення в комп'ютерних мережах» (ДР № 0121U109936), «Система виявлення ЗПЗ та комп'ютерних атак в корпоративних мережах з використанням хибних об'єктів атак та пасток» (ДР № 0124U000980), у яких автор був виконавцем.

Мета і задачі дослідження. *Об'єкт дослідження* – процеси прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем.

Предмет дослідження – методи та засоби прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем.

Метою дисертаційного дослідження є забезпечення оцінювання наявного набору вимог з позиції прогнозованого рівня якості та безпеки програмного

забезпечення комп’ютерних систем, яке планується до реалізації за таким набором вимог, шляхом розроблення методів та засобів прогнозування рівня безпеки та якості програмного забезпечення комп’ютерних систем.

Для досягнення поставленої мети слід розв’язати такі *задачі:*

– проаналізувати відомі методи та засоби прогнозування безпеки і якості програмного забезпечення комп’ютерних систем;

– розробити метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп’ютерних систем;

– розробити метод прогнозування рівня якості програмного забезпечення комп’ютерних систем на основі атрибутів якості;

– розробити метод прогнозування рівня безпеки програмного забезпечення комп’ютерних систем;

– розробити метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем;

– спроєктувати та реалізувати системи прогнозування рівня якості та безпеки програмного забезпечення комп’ютерних систем;

– спроєктувати та реалізувати систему ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем.

Методи дослідження. При розв’язанні поставленої науково-прикладної задачі використовувались аналіз та синтез, принципи загальної теорії систем та системного аналізу, методи аналізу та моделювання процесів, теоретико-множинні підходи, алгебра систем, апарат модельно-орієнтованих підходів, методи концептуального моделювання, принципи побудови баз знань та формування логічного висновку, загальні принципи створення систем, методи емпіричного дослідження.

Наукова новизна дисертаційного дослідження полягає у одержанні таких наукових результатів:

вперше розроблено:

1) метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп’ютерних систем, який відрізняється від відомих

структуруванням вимог за атрибутами якості, та забезпечує вибір значень атрибутів якості ПЗ з природомовної специфікації вимог, які використовуються для оцінювання значень характеристик якості ПЗ та для комплексного оцінювання якості ПЗ; розроблений метод дозволяє автоматизувати опрацювання вимог та мінімізувати суб'єктивний вплив людини у процесах оцінювання;

2) метод прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості, який, на відміну від відомих, дозволяє прогнозувати рівень якості розроблюваного програмного забезпечення на основі обробки значень атрибутів якості, доступних у специфікації вимог. Таким чином, запропонований метод дозволяє порівнювати специфікації вимог, одразу відмовлятись від реалізації ПЗКС на основі невдалих специфікацій (економія коштів та часу, зменшення ймовірності провальних і проблемних проєктів) та виконувати ґрунтовний вибір специфікації для подальшої реалізації ПЗКС саме високої якості (за умови, що помилки не будуть внесені під час подальших проєктування та розроблення ПЗ);

удосконалено:

3) метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем, який, на відміну від відомих, виконує аналіз припинення функціонування ПЗКС та функційних можливостей ПЗКС, які потенційно можуть бути вразливостями, та забезпечує висновок щодо того, чи є припинення функціонування ПЗКС відмовою, і, якщо так, то який тип вона має, а також забезпечує висновок, чи є функційна можливість вразливістю, і, якщо так, то який тип вона має;

4) метод прогнозування рівня безпеки програмного забезпечення комп'ютерних систем, який, на відміну від відомих, встановлює залежність значення безпеки ПЗКС від значень атрибутів якості та формує прогнозоване числове значення безпеки ПЗКС на основі атрибутів, і забезпечує прогнозування рівня безпеки ПЗКС на основі отриманого числового значення, а також порівняння специфікацій вимог за прогнозованим рівнем безпеки розроблюваного ПЗКС та можливість відбраковування невдалих специфікацій.

Практичне значення отриманих результатів. Практичне значення отриманих результатів полягає в доведенні теоретичних результатів дисертаційної роботи до реалізації та у безпосередньому використанні їх на підприємстві.

Реалізована система прогнозування рівня якості ПЗКС надає користувачу прогнозовані оцінки восьми характеристик якості ПЗКС, комплексний показник прогнозованої якості ПЗКС та висновок про рівень якості майбутнього ПЗКС, на основі якого можна виконати порівняння специфікацій вимог та обґрунтований вибір специфікації вимог для подальшої реалізації.

Реалізована система прогнозування рівня безпеки ПЗКС забезпечує аналіз вимог, на основі якого надає користувачу прогнозовану оцінку безпеки ПЗКС (як характеристики якості) та висновок про рівень безпеки майбутнього ПЗКС, на основі якого можна виконати порівняння специфікацій вимог та обґрунтований вибір специфікації для подальшої реалізації.

Реалізована система ідентифікації та класифікації відмов і вразливостей ПЗКС надає висновок щодо того, чи є припинення функціонування ПЗКС відмовою ПЗКС; висновок щодо того, чи є функційна можливість ПЗКС вразливістю ПЗКС; висновок про тип відмови та/або вразливості за їх наявності.

Результати дисертаційної роботи впроваджено у (Додаток Б): ПП «Авіві» (акт впровадження від 10.01.2024 р.); ТОВ «Деймос» (акт впровадження від 15.02.2024 р.); ГО «ІТ Кластер м. Хмельницького» (акт впровадження від 22.03.2024 р.); у навчальному процесі Хмельницького національного університету (акт впровадження від 13.05.2024 р.); при виконанні держбюджетних тем Хмельницького національного університету «Самоорганізована розподілена система виявлення зловмисного програмного забезпечення в комп’ютерних мережах» (ДР № 0121U109936), «Система виявлення ЗПЗ та комп'ютерних атак в корпоративних мережах з використанням хибних об'єктів атак та пасток» (ДР № 0124U000980).

Особистий внесок здобувача та внесок інших співавторів у спільних публікаціях. Усі наукові результати дисертаційного дослідження отримані автором особисто. Список опублікованих праць за темою дисертації представлено

в списку використаних джерел – [60-69]. У спільних публікаціях автору належать такі результати: метод прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості [60, 69], архітектура та реалізація системи прогнозування рівня якості програмного забезпечення комп'ютерних систем [61], огляд факторів, які впливають на безпеку програмного забезпечення мобільних додатків, та вразливостей мобільних додатків, а також моделювання предметної галузі оцінювання та прогнозування безпеки мобільних додатків [62], засіб ідентифікації та класифікації відмов і вразливостей програмного забезпечення [63], метод аналізу вимог до програмного забезпечення на предмет пошуку значень атрибутів якості [64], метод ідентифікації та класифікації відмов і вразливостей [65], метод визначення кількості рядків програмного коду, написаного вручну [66], метод прогнозування рівня безпеки програмного забезпечення комп'ютерних систем [67], нейромережна модель прогнозування рівня якості ПЗКС [68].

Особистий внесок інших співавторів у спільних публікаціях: у статті [60] Т. Говорущенко виконувала адміністрування та концептуалізацію дослідження, Д. Медзатий керував постановкою експериментів, М. Лебіга сумісно із автором працював над оглядом існуючих методів та рішень; у статті [61] Т. Говорущенко виконувала адміністрування та концептуалізацію дослідження, Д. Медзатий та А. Боярчук керували постановкою експериментів; у статті [62] Є. Зайцева виконувала концептуалізацію дослідження, обговорення результатів дослідження, рецензування та коригування рукопису, керівництво проєктом, Т. Говорущенко виконувала концептуалізацію дослідження, огляд відомих методів та рішень, обговорення результатів дослідження, підготовку чернетки рукопису, адіміністрування проєкту та пошук коштів на публікацію статті, О. Павлова сумісно із автором працювала над методологією дослідження та опрацьовувала результати дослідження, а також виконувала огляд відомих методів та рішень, підготовку чернетки рукопису, візуалізацію результатів дослідження; у статті [63] Д. Медзатий виконував концептуалізацію дослідження, рецензування та коригування рукопису, О. Войчур спільно із автором працював над оглядом відомих методів та рішень; у статті [64] Д. Медзатий виконував адімінстрування

та концептуалізацію дослідження, рецензування та коригування рукопису; у статті [65] П. Попов виконував концептуалізацію дослідження, обговорення результатів дослідження, рецензування та коригування рукопису, керівництво проєктом, Т. Говорущенко виконувала концептуалізацію дослідження, обговорення результатів дослідження та адміністрування проєкту, Д. Медзатий сумісно із автором працював над методологією дослідження та опрацьовував результати експериментів; у статті [66] Т. Говорущенко виконувала адімінстрування та концептуалізацію дослідження, рецензування та коригування рукопису; у статті [67] Т. Говорущенко та А. Боярчук виконували концептуалізацію дослідження, обговорення результатів дослідження, рецензування та коригування рукопису, керівництво проєктом, Д. Медзатий виконував концептуалізацію дослідження, обговорення результатів дослідження, адміністрування проєкту, А. Гнатчук сумісно із автором працювала над оглядом відомих методів та рішень; у авторському свідоцтві [68] Т. Говорущенко виконувала адімінстрування та концептуалізацію дослідження, рецензування та коригування рукопису, М. Лебіга сумісно із автором працював над моделюванням прогнозування якості програмного забезпечення; у авторському свідоцтві [69] Т. Говорущенко виконувала адімінстрування та концептуалізацію дослідження, рецензування та коригування рукопису, М. Лебіга сумісно із автором працював над патентним пошуком відносно відомих методів та рішень.

Апробація матеріалів дисертації. Основні результати дисертаційного дослідження доповідались та обговорювались на 3 міжнародних науково-технічних та науково-практичних семінарах [65-67], а саме: 5th International Workshop on Intelligent Information Technologies & Systems of Information Security IntelITSIS-2024 (м. Хмельницький, 2024); 2nd, 3rd International Workshop on Information Technologies: Theoretical and Applied Problems ITTAP (м. Тернопіль, 2022; м. Тернопіль (Україна) та м. Ополе (Польща), 2023).

Публікації. Основні результати дисертації опубліковані у 10 наукових працях ([60-69] та додаток А), серед яких 3 статті у періодичних виданнях, що індексуються в наукометричних базах Scopus, Web of Science [60-62] (в тому числі

1 стаття у періодичному виданні 2-го квартилю, що індексується в наукометричних базах Scopus та WoS [60] та 2 статті у періодичних виданнях 3-го квартилю, що індексуються в наукометричній базі Scopus [61, 62]); 2 статті у фахових наукових журналах України [63, 64], включених на дату опублікування до переліку наукових фахових видань України категорії Б; 3 публікації, які засвідчують апробацію матеріалів дисертації (статті в матеріалах конференцій, що індексуються в наукометричній базі Scopus) [65-67]; 2 свідоцтва про реєстрацію авторського права на твір [68, 69].

Структура та обсяг дисертації. Дисертація складається з анотації, змісту, переліку умовних скорочень, вступу, чотирьох розділів, висновків, списку використаних джерел із 165 найменувань на 17 сторінках та 4 додатків на 14 сторінках. Загальний обсяг дисертаційної роботи становить 169 сторінок друкованого тексту, з них 125 сторінок основного тексту. Дисертація містить 35 рисунків та 8 таблиць.

РОЗДІЛ 1
АНАЛІЗ ВІДОМИХ МОДЕЛЕЙ, МЕТОДІВ ТА ЗАСОБІВ ПРОГНОЗУВАННЯ РІВНЯ ЯКОСТІ ТА БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

1.1. Поняття та завдання забезпечення якості та безпеки програмного забезпечення комп'ютерних систем

Зростаючий інтерес до галузі інформаційних технологій приводить до створення великої кількості програмного забезпечення різного призначення, яке постійно збільшується за обсягом та за складністю [1].

Постійний розвиток комп'ютерних систем із застосуванням хмарних технологій, штучного інтелекту, доповненої та віртуальної реальності підвищує вимоги до безпеки та якості програмного забезпечення комп'ютерних систем, а також до процесу прогнозування та оцінювання їх якості та безпеки.

Успішне та якісне функціонування комп'ютерних систем залежить від якості їх різноманітних компонентів. Серед цих компонентів програмне забезпечення відіграє одну з ключових ролей, і якість цього програмного забезпечення суттєво впливає на загальну якість та успішність функціонування комп'ютерної системи [2].

Програмне забезпечення комп'ютерних систем включає в себе весь набір програм і процедур, пов'язаних з роботою комп'ютерної системи. ПЗКС організовує процес обробки інформації в комп'ютерній системі, воно тісно пов'язане з апаратними засобами. Основною метою ПЗКС є забезпечення максимальної продуктивності користувача за рахунок оптимального розв'язку. Критерієм оптимізації ПЗКС є мінімальний час розв'язку обчислювальних завдань, який досягається за рахунок розроблення та використання ефективних методів і засобів розпаралелювання алгоритмів, а також розподілу обчислювальних робіт за процесорами.

Ринок програмного забезпечення комп'ютерних систем швидко зростає. ПЗКС стає все складнішим, на нього покладається все більше відповідальності. Із збільшенням надійності апаратного забезпечення, якість і безпека програмного забезпечення комп’ютерних систем стають все більшими пріоритетами як для розробників, так і для користувачів.

На даний час розроблення програмного забезпечення комп’ютерних систем перетворилось на одну з найдорожчих індустрій. Компанія Statista показує, що наразі витрати на програмне забезпечення становлять 491 мільярд доларів США, розмір ринку програмного забезпечення в США становить 285 мільярдів доларів США [70]. Відтак будь-які проблемні моменти в технологічному процесі його проєктування та розроблення можуть призвести до небажаних наслідків (збільшення термінів розробки ПЗ, зниження продуктивності ПЗ, здорожчання ПЗ, зниження конкуретнтоздатності та репутації компанії-розробника, неможливість виконання необхідних функціональних задач розробленим ПЗ, неефективне прийняття рішень комп’ютерною системою, складовою частиною якої є розроблене ПЗ тощо) [71, 72].

Отже, оцінювання та прогнозування якості та безпеки програмного забезпечення є однією з ключових задач при розробці програмного забезпечення комп'ютерних систем, оскільки вплив програмного забезпечення на якість і безпеку апаратних та програмних компонентів критичних систем є надзвичайно значущим.

Якість ПЗКС – це ступінь, в якому ПЗ відповідає потребам користувача при його застосуванні у визначених умовах [3]. Безпека ПЗКС як характеристика якості ПЗ розглядається як забезпечення захисту від потенційних загроз (несанкціонований доступ, зловживання даними тощо) [3].

Наразі чимало програмних проєктів не відповідають бізнес-вимогам і потребам користувача та не виконуються у встановлені часові рамки в рамках виділеного бюджету, тобто не є якісними та успішними [73]. На кожен вкладений 1 мільярд доларів організації втрачають в середньому 97 мільйонів доларів через недостатню ефективність та якість програмних проєктів [74]. Ціна низької якості програмного забезпечення у 2020 році в США склала 260 млрд доларів США у

порівнянні з 177,5 млрд доларів США у 2018 році – рис. 1.1 [75]. Низька якість програмного забезпечення спричиняє численні катастрофи, аварії, фінансові втрати та витоки інформації [4, 76, 77].

У звітах «Software Fail Watch» компанії Tricentis [78, 79], наявна інформація, що через неякісне ПЗ у 2017 році постраждали 3,6 млрд осіб і було завдано збитків на 1,7 трильйонів доларів США [78]; у 2016 році через неякісне ПЗ постраждали 4,4 млрд осіб і завдано збитків на 1,1 трильйони доларів США [79].

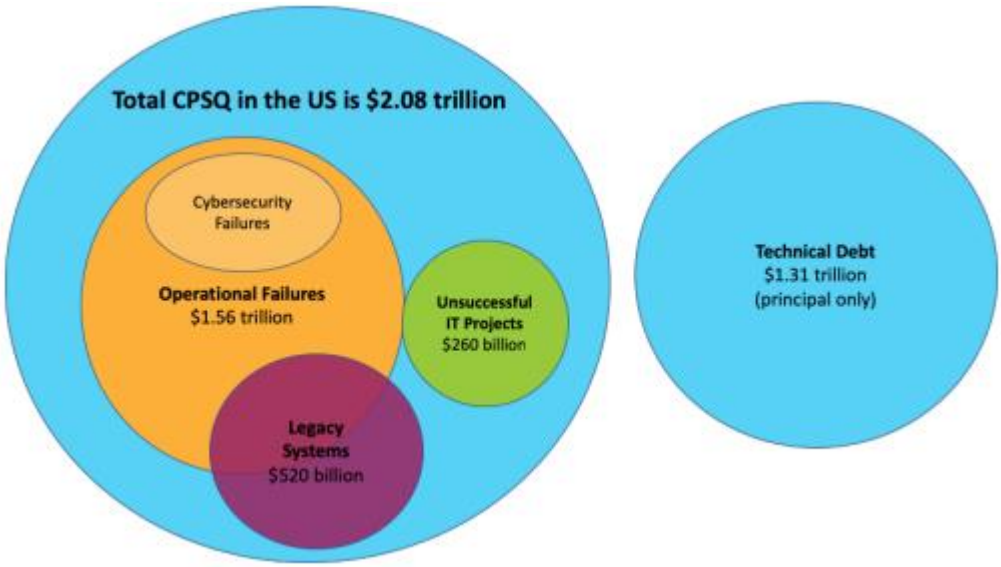


Рис.1.1 – Ціна низької якості програмного забезпечення у 2020 році в США [75]

Отже, на сьогодні розробка програмного забезпечення комп'ютерних систем має завищену ймовірність провалу проєкту [58, 75]. Загалом, близько 10-20% усіх програмних проєктів не доходять до фінішу, 40-60% проєктів завершуються із запізненням в середньому на 150-200% від запланованого терміну, 25-40% проєктів не виконують поставлені завдання в повному обсязі, 40-55% проєктів потребують додаткового фінансування, а в 20% проєктів не враховуються всі зміни, що вносилися замовником [80]. Статистика успішності програмних проєктів за версією CHAOS Report 2020 року представлена на рис. 1.2 [75].

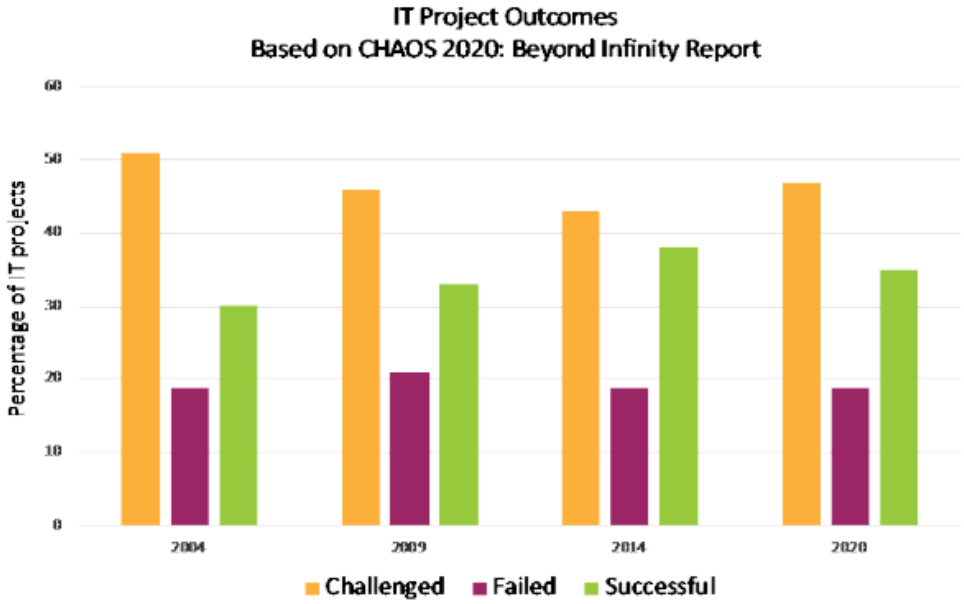


Рис.1.2 – Статистика успішності програмних проєктів за версією CHAOS Report 2020 року [75]

Низька якість розроблюваного ПЗКС вимагає резервування до 70% бюджету програмного проєкту на етап супроводу. До 60% усіх модифікацій ПЗКС здійснюються для усунення помилок, тоді як лише 40% спрямовані на корекцію ПЗ в рамках бізнес-процесу, покращення певних показників або запобігання потенційним проблемам. Вартість якості ПЗ сукупно складають вартість внутрішніх і зовнішніх збоїв під час роботи ПЗКС та вартість попередження дефектів [81].

Проєктний трикутник складається з трьох основних елементів – час, бюджет функціональність (рис. 1.3). При зміні принаймні одного з цих елементів змінюються всі інші [82, 83]. Характер впливу змін одного з елементів на інші залежить переважно від специфіки проєкту та конкретних обставин. Наприклад, скорочення терміну виконання проєкту може в одному випадку збільшити його вартість через необхідність залучення більшої кількості розробників, а в іншому випадку зменшити вартість завдяки відмові від реалізації певних функціональних можливостей. Четвертим елементом проєктного трикутника, який є в його центрі, є якість. Таким чином, навіть незначна зміна будь-якого елемента обов’язково впливає на якість.

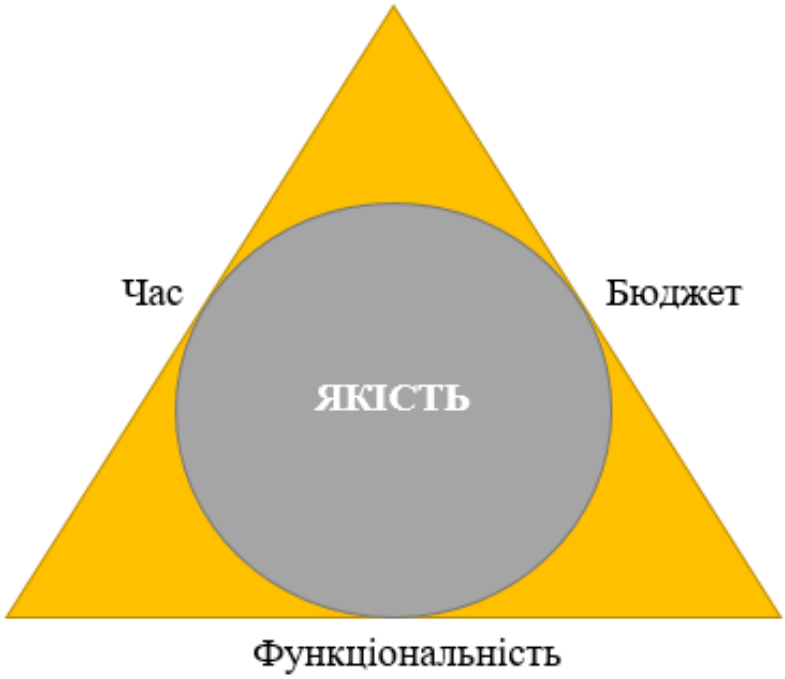


Рис.1.3 – Проєктний трикутник

Складність і якість вимог до програмного забезпечення різко змінилися за останні роки, а споживачі стали значно вибагливішими щодо вартості, термінів і якості програмних рішень [84, 85]. Швидке зростання складності і розмірів сучасного програмного забезпечення при одночасному зростанні відповідальності функцій, що ним виконуються, різко підвищили вимоги замовників і користувачів до якості та ефективності використання ПЗ. Успіх будь-якого проєкту залежить від його здатності відповідати потребам споживача. Відтак стратегічною задачею життєого циклу сучасних програмних проєктів стало забезпечення необхідної якості ПЗ. Саме тому трендом галузі інформаційних технологій останніх років є підвищена увага до теми контролю та управління якістю розробки програмного забезпечення. Основу менеджмента якості складає оцінювання якості програмного забезпечення, особливо на початкових етапах його розроблення (по факту, прогнозування безпеки та якості ПЗ, що розробляється), оскільки найвищою (порядку третини) є частота появи та вплив на якість ПЗ факторів ранніх етапів життєвого циклу [86, 87]. Програмне забезпечення комп’ютерних систем навіть критичного застосування все ще призводить до збоїв через недостатні та неадекватні вимоги до програмного забезпечення. Після десятиліть розвитку

критичного інжинірингу більшість нещасних випадків, пов'язаних зі збоями критичного програмного забезпечення, простежуються через проблеми у специфікації вимог. Інженерія вимог все ще залишається чимось середнім між інженерією та мистецтвом, тобто вимоги до програмного забезпечення все ще залежать від особистості, досвіду та навичок розробників. Загальновизнано, що якісна інженерія вимог приводить до підвищення якості програмного забезпечення і значно знижує ризик невдачі або перевитрати бюджету проєктів з розробки програмного забезпечення [88].

Оскільки неможливо виявити та виправити всі дефекти програмного забезпечення, важливо, щоб негативний вплив дефектів був виявлений, усунутий та мінімізований якомога раніше. Раннє виявлення та виправлення дефектів програмного забезпечення забезпечує мінімізацію збитків, спричинених дефектами програмного забезпечення, оскільки кількість часу та грошей збільшується, коли в програмному забезпеченні є дефекти. Хоча моніторинг та виправлення дефектів програмного забезпечення також є дорогими та трудомісткими процедурами.

Отже, забезпечення якості програмного забезпечення, як правило, призводить до більших витрат грошей і часу. Розробники програмного забезпечення зазвичай розглядають забезпечення якості ПЗ як додаткову тривалу та документомістку операцію, яка не має особливої цінності для клієнта, проте вони помиляються, оскільки клієнти зацікавлені у високій якості продукту, з яким їм доведеться працювати, від якого може залежати їхнє здоров'я і навіть життя.

Оцінка якості програмного забезпечення згідно зі стандартом ISO 25010 [3] виконується таким чином: на базі атрибутів якості, що визначені у стандарті ISO 25023 [89], проводиться оцінка підхарактеристик та характеристик якості, за якими, у свою чергу, формується комплексна оцінка якості програмного забезпечення. Згідно із стандартом ISO 25010, якість програмного забезпечення визначається вісьмома основними характеристиками: ефективність (Performance Efficiency), функційна придатність (Functional Suitability), зручність використання (Usability), сумісність (Compatibility), безпека (Security), надійність (Reliability), можливість переносу (Portability), супроводжуваність (Maintainability). Кожна з цих

характеристик є агрегацією кількох підхарактеристик якості, яких, згідно зі стандартом ISO 25010 [3], налічується 31. На найнижчому рівні ієрархії розташовані атрибути якості, описані у стандарті ISO 25023 [89]. Всього характеристики якості залежать від 203 атрибутівякості , що включають у себе 138 атрибутів, які не повторюються.

Згідно зі стандартом ISO 25010 [3], безпека ПЗ є однією з характеристик якості ПЗ, залежить від 5-и підхарактеристик (конфіденційність (Confidentiality), цілісність (Integrity), автентичність (Authenticity), відповідальність (Accountability), безвідмовність (Non-Repudiation)) і, як і інші характеристики, визначається на основі певних атрибутів якості (15 різних атрибутів) зі стандарту ISO 25023 [89].

На даний момент оцінка атрибутів, яка спрямована на визначення якості та безпеки програмного забезпечення, переважно здійснюється лише для програмного (сирцевого) коду. Однак всі необхідні атрибути для цієї оцінки мають бути враховані вже у специфікації вимог до програмного забезпечення [90, 91]. Саме у вимогах до ПЗ описано поведінку майбутнього ПЗ, функціональні можливості та обмеження ПЗ, його властивості та атрибути. Значення атрибутів якості враховують і мету програмного проєкту, і його тип. Якщо атрибут(и) вписано у вимоги специфікації, визначено значення такого атрибуту, відтак розробники будуть зобов’язані забезпечити наявність такого атрибуту в своєму ПЗ і зазначене у вимогах значення атрибуту, інакше ПЗ не пройде верифікації та валідації. Отже, вже на основі значень атрибутів зі специфікації вимог можна прогнозувати рівень якості та безпеки програмного забезпечення для програмних проєктів будь-якого типу.

Розвиток сучасних технологій та методологій у сфері проєктування та розробки програмного забезпечення потребує постійного удосконалення засобів оцінки, зокрема, у напрямку прогнозування безпеки та якості програмного забезпечення на початкових етапах його життєвого циклу, враховуючи ефективність витрат часу та ресурсів. Своєчасне прогнозування якості та безпеки ПЗ може бути використано для прийняття будь-яких превентивних заходів для зменшення кількості збоїв ПЗ під час його роботи. Бажано вже за наявності

специфікації вимог до ПЗ розуміти, чи за цією специфікацією може бути розроблене якісне ПЗ. Ряд проведених досліджень [4, 76, 77] показали, що версії ПЗ, створені різними розробниками за однаковими вимогами, зазвичай містили ряд спільних помилок, зумовлених помилками або неточностями в самих вимогах, і навпаки версії ПЗ, написані одним колективом розробників за різними вимогами, суттєво різнились своєю якістю та успішністю. Такий підхід до прогнозування безпеки та якості ПЗКС на основі аналізу вимог до нього дозволить також реалізувати можливість вибору набору вимог серед різних альтернативних специфікацій, які пропонуються різними розробниками для вирішення одних і тих самих завдань.

Зараз можна спостерігати не лише зростання масштабів програмного забезпечення, але й стрімкий розвиток штучного інтелекту. Штучний інтелект є найкращим інструментом для аналізу величезної кількості даних та економії людських зусиль, зокрема для значного скорочення часу та підвищення ефективності розробки складного програмного забезпечення. За оцінками світового звіту з якості, 64% компаній впроваджують штучний інтелект для процесів забезпечення якості програмного забезпечення [92].

1.2. Аналіз методів та засобів прогнозування рівня якості програмного забезпечення комп’ютерних систем

Аналіз відомих методів та засобів показав, що на даний час розроблено чимало методів і засобів для роботи із вимогами до ПЗ, наприклад, методи оцінювання достатності інформації про якість у специфікаціях вимог [90, 91], онтологічна модель знань щодо забезпечення якості програмного забезпечення [93] тощо, проте всі ці методи і засоби більшою мірою спрямовані на оцінювання якості, достатності та повноти вимог до ПЗ, ніж на оцінювання та прогнозування якості, власне, самого ПЗ.

Нейромережний метод оцінки та прогнозування якості ПЗ, представлений в [90], використовує штучну нейронну мережу (ШНМ) для формування очікуваних

значень якості та складності програмного забезпечення на етапі проєктування з використанням значень метрик етапу проєктування. Недоліком такого методу є те, що прогнозування та оцінювання якості відбувається на основі аналізу метрик складності та якості, яких на даний час розроблена величезна кількість, причому всі вони нестандартизовані, і їх вибір здійснювався на розсуд розробника методу.

В роботі [94] розроблено методи оцінки якості тестування програмного забезпечення на основі теорії нечіткої математики, які дозволяють оцінювати якість тестування програмного забезпечення. Стаття [46] пропонує модель процесу оцінки якості програмного забезпечення, що ґрунтується на стандарті ISO/IEC 14598 та враховує різновиди інформаційних систем. Стаття [52] пропонує модель оцінки якості ПЗ системного рівня для управління апаратними пристроями на основі вимог операційного середовища, областей застосування та робочих характеристик системного ПЗ. У статті [95] виконано огляд різних методів аналізу різних аспектів якості ПЗ (схильність до помилок, схильність до змін, прогнозування дефектів, прогнозування надійності, прогнозування ремонтопридатності ПЗ). Ця категорія досліджень фокусується на оцінці якості тестування програмного забезпечення і якості готових програмних продуктів, але методи, що описані, не придатні для прогнозування якості програмного забезпечення на основі наданих вимог.

Дослідження [48] представляє гібридний метод прогнозування якості з підвищеною точністю прогнозу на основі використання вдосконаленої нейронної мережі в поєднанні з гібридним алгоритмом оптимізації пошуку Cuckoo. У статті [96] представлено статистичні підходи до прогнозування якості ПЗКС на основі даних про процеси розробки ПЗ із застосуванням узагальненого лінійного моделювання. Стаття [41] пропонує модель прогнозування якості ПЗКС на основі машинного навчання, яка, використовуючи програмні метрики та помилкові дані з попередніх проєктів, виявляє проблемні модулі з високим ризиком аномалій розробки для майбутніх проєктів. Стаття [50] пропонує систему нечітких висновків для прогнозування якості ПЗ на основі параметрів специфікації вимог (індекс зручності читання, розмір, складність, оцінка зв'язку). Стаття [97] порівнює різні методи інтелектуального аналізу даних для прогнозування якості ПЗКС. У статті

[98] висувається концепція використання байєсівської мережі як основи для комплексного прогнозування якості програмного забезпечення. Це прогнозування здійснюється на основі атрибутів якості, їх взаємозв'язків, експертних знань про конкретну сферу та емпіричних даних. Така група робіт присвячена прогнозуванню якості з використанням компонентів штучного інтелекту, проте жоден з описаних підходів не дозволяє встановити залежностей між характеристиками та атрибутами якості, а також між якістю та її характеристиками, які б дозволили розрахунок кількісних оцінок спочатку характеристик якості, а потім кількісної оцінки, власне, якості ПЗ на основі наявних значень атрибутів якості, як рекомендують стандарти ISO (ISO 25010, ISO 25023).

Надмірне повторне використання робить програмне забезпечення вразливим до несправностей через підвищену складність і схильність до старіння. У статті [35] пропонується модель прогнозування багаторазового використання стійкого до старіння ПЗ на основі об’єктно-орієнтованого проєктування. Оцінка продуктивності підтверджує, що модель ШНМ може використовуватися для більш ранньої оптимізації повторного використання, стійкої до старіння, для розробки ПЗ на основі об’єктно-орієнтованого проєктування. Стаття [36] присвячена розробці надійної та ефективної моделі прогнозування можливості повторного використання та оцінки економічної ефективності повторного використання. У статті розглянуто різні моделі прогнозування багаторазового використання (дерево рішень, Naive Bayes, штучна нейронна мережа, машина екстремального навчання, регресійні алгоритми, багатовимірний адаптивний регресійний сплайн та адаптивний генетичний алгоритм) на предмет їх економічності та ефективності прогнозування порівняно з об’єктно-орієнтованим проєктуванням ПЗКС. Зазначені роботи присвячені прогнозуванню можливості повторного використання ПЗКС, проте слабко висвітлюють питання прогнозування якості ПЗ.

Штучна нейронна мережа є одним із широко використовуваних методів машинного навчання, на якому базуються більшість запропонованих методів і моделей прогнозування дефектів. Дослідження [51] надає критичний аналіз використання ШНМ для прогнозування дефектів програмного забезпечення.

Стаття [99] описує модель прогнозування дефектів на початкових етапах життєвого циклу шляхом виконання прогнозування дефектів ПЗ в кінці кожного етапу ЖЦ з метою зменшення часових та фінансових витрат шляхом передбачення дефектів перед тестуванням. Для прогнозування дефектів використовується ШНМ, яка базується на опрацюванні дев’яти обраних авторами показників. Стаття [39] пропонує гібридну модель прогнозування дефектів на основі адаптивної штучної нейронної мережі, яка є більш ефективною, ніж інші моделі прогнозування дефектів на основі ШНМ. У дослідженні [100] запропоновано модель прогнозування дефектів у програмному забезпеченні на основі штучної нейронної мережі. Ця мережа призначена для класифікації та прогнозування, використовуючи п'ять об'єктно-орієнтованих метрик із наборів метрик CK та Martin. У статті [101] автори проаналізували найбільш використовувані алгоритми машинного навчання (Artificial Neural Network, Particle Swarm Optimization, Decision Trees, Naive Bayes, Linear classifier) за допомогою інструменту KEEL і за допомогою k-кратного методу перехресної перевірки – як інструменти для прогнозування дефектів програмного забезпечення, що безпосе-редньо впливає на якість ПЗ. Робота [40] порівнює два методи класифікації (support vector machine та штучна нейронна мережа) для створення моделей класифікації дефектів ПЗ з використанням показників програмного забезпечення та схильності до дефектів як незалежних, так і залежних змінних. Така група робіт присвячена прогнозуванню якості ПЗКС через прогнозування наявності та кількості дефектів ПЗ з використанням компонентів штучного інтелекту. Але жоден із наведених підходів не спрямований безпосередньо на кількісну оцінку якості програмного забезпечення на основі наявних атрибутів якості та прогнозування рівня якості програмного забезпечення за допомогою отриманого числового показника якості.

Робота [102] представляє застосування гібридної ШНМ та оптимізації рою квантових частинок для класифікації програмних модулів на категорії схильності до збоїв або несхильності до збоїв з метою мінімізації витрат і підвищення ефективності процесу розробки програмного забезпечення. Раннє виявлення модулів, схильних до збоїв, збільшує шанси на випуск безпомилкового ПЗ із

скороченням зусиль і витрат на тестування. Робота [42] пропонує новий підхід 3PcGE, який показує високу точність при прогнозуванні дефектів ПЗ. Дослідження [103] використовує комбінацію традиційної ШНМ і нового алгоритму Artificial Bee Colony для більш надійного і точного прогнозування схильності програмного модуля до збоїв. Ця група досліджень спрямована на передбачення схильності програмного модуля до відмов та збоїв з метою поліпшення ефективності процесу розробки програмного забезпечення. Такі дослідження не можуть використовуватись для прогнозування та оцінювання якості програмного забезпечення на основі критеріїв, визначених у стандартах ISO 25010 та ISO 25023.

У статті [104] запропоновано модель надійності програмного забезпечення із збільшеною кількістю нейронних мереж, які навчаються з використанням алгоритму зворотного поширення, та із збільшеною кількістю навчальних вибірок. Отримані результати показують, що запропонована модель підвищує точність прогнозу надійності ПЗКС. Основною метою роботи [49] є покращення прогнозування якості ПЗ та розробка моделі штучної нейронної мережі для задачі оптимізації надійності ПЗ. За допомогою ШНМ прогнозуються якісні характеристики програмних компонентів. Стаття [105] пропонує нейронечіткий робочий процес для пом’якшення виявлених ризиків програмного забезпечення. Запровадження такого процесу допомагає при оцінці ймовірності виникнення дефектів для забезпечення якості та надійності програмного забезпечення. Він також є базою для розробки нових моделей, методів та інструментів для забезпечення якості та надійності програмного забезпечення. Роботи [49, 104, 105] пропонують можливість оцінювання та прогнозування надійності програмного забезпечення, що є однією з восьми характеристик якості програмного забезпечення згідно із стандартом ISO 25010. Вони встановлюють залежність надійності програмного забезпечення від деяких атрибутів якості, проте не аналізують такі зв'язки для інших семи характеристик якості та загальної якості ПЗ.

Дослідження [106] пропонує QiUPS – експертну систему для прогнозування якості використання на ранніх етапах розробки програмного забезпечення на основі ШНМ зі згортковими нейронними мережами. Робота [107] пропонує

методологію на основі машини екстремального навчання з підтримкою багатоцільової gray wolf оптимізації для прогнозування якості ПЗ. Основною метою запропонованого підходу є визначення якості ПЗ за показниками складності на основі чотирьох системних метрик на основі компонентів (пропускна здатність, час реакції на відмову, метрики складності обмеженого інтерфейсу, узгодженість поверхні інтерфейсу). Ці роботи присвячені прогнозуванню якості з використанням компонентів штучного інтелекту, проте оцінювання та прогнозування якості ПЗ відбувається не на основі аналізу атрибуів зі стандартів ISO, а на основі метрик, обраних на розсуд науковців, які розробили дані методи.

Хоча досліджені моделі, методи та інструменти і можуть бути корисними в різних контекстах, вони не встановлюють залежності між характеристиками якості та атрибутами, не дозволяють обчислювати кількісні значення характеристик якості на основі значень атрибутів, не встановлюють залежності та не дозволяють обчислювати кількісне значення якості на основі значень її характеристик і не забезпечують прогнозування рівня безпеки та якості ПЗКС на основі отриманих числових значень.

Тоді виникає питання, чи можуть відомі моделі, методи та інструменти аналізу вимог і оцінювання якості ПЗКС відповідати наступним критеріям, які є важливими для забезпечення можливості прогнозування рівня безпеки та якості ПЗКС згідно зі стандартом ISO 25010: можливість аналізу атрибутів якості у вимогах – *критерій 1*, встановлення залежностей значень характеристик якості від значень атрибутів – *критерій 2*, обчислення кількісних значень характеристик якості – *критерій 3*, встановлення залежності між значенням якості та значеннями її характеристик – *критерій 4*, обчислення кількісного значення якості – *критерій 5*, прогнозування рівня якості (як характеристики ступеня відповідності ПЗКС конкретній потребі користувача за фіксованих умов використання) – *критерій 6*, представлення всіх вищезазначених сервісів в комплексі – *критерій 7*, методологічна узгодженість та інтеграція методів та інструментів (тобто, по суті, наявність системи прогнозування безпеки та/або якості ПЗКС на основі вимог, яка забезпечує всі вищезазначені сервіси) – *критерій 8*.

Результати аналізу відомих моделей, методів та інструментів аналізу вимог та оцінювання якості ПЗКС з точки зору відповідності їх вищезазначеним критеріям можна узагальнити у таблицю 1.1.

Таблиця 1.1

Аналіз відомих моделей, методів та засобів аналізу вимог до ПЗ та оцінювання якості ПЗКС

Моделі, методи та засоби аналізу вимог до ПЗ та оцінювання якості ПЗКС	**Автор(и)**	**Критерії**							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Методологія оцінки достатності інформації про якість у специфікаціях вимог до ПЗ [90]	Говорущенко (2018)	+	+		+				
Інформаційна технологія оцінки достатності інформації про якість [90]	Говорущенко (2018)	+	+		+				
Метод та програмний засіб оцінки якості програмного забезпечення [90]	Говорущенко (2018)			+	+	+	+		
Модель прогнозування якості програмного забезпечення на основі машинного навчання [41]	Goyal та інші (2020)			+		+			
Система нечіткого виводу для прогнозування якості програмного забезпечення на основі параметрів вимог [50]	Masood та інші (2018)	+		+		+			
Адаптивна модель прогнозування гібридних дефектів на основі штучних нейронних мереж [39]	Arora та інші (2018)			+		+			

Нейромережна модель для оптимізації надійності та якості програмного забезпечення [49]	Tomar та інші (2018)	+							
Методологія на основі машинного екстремального навчання з підтримкою багатоцільової grey wolf оптимізації для прогнозування якості ПЗКС [107]	Tripathi та інші (2022)					+	+		+
Експертна система для прогнозування якості використання на ранніх стадіях розробки на основі ШНМ [106]	Alshareet та інші (2018)					+	+		
Методи на основі нечіткої математики для оцінки якості тестування програмного забезпечення [94]	Sun та інші (2021)					+	+		
Методи аналізу різних аспектів якості програмного забезпечення [95]	Lakra та інші (2021)			+		+			
Моделі для прогнозування повторного використання програмного забезпечення комп’ютерних систем (в т.ч. нейромережні) [36]	Padhy та інші (2019)						+		
Застосування штучних нейронних мереж для прогнозування дефектів програмного забезпечення [51]	Khan та інші (2022)	+		+		+			
Модель на основі ШНМ для прогнозування дефектів програмного забезпечення [100]	Kaur та інші (2019)	+		+		+			
Підхід на основі трьох батьків і генетичної еволюції (3PcGE) для прогнозування дефектів ПЗКС [42]	Goyal (2023)			+		+			

Модель надійності та якості програмного забезпечення зі збільшеною кількістю нейронних мереж [104]	Kumaresan та інші (2019)					+	+		

Так, методологія оцінки достатності інформації про якість у специфікаціях вимог [90], а також інформаційна технологія оцінки достатності інформації про якість [90] передбачають аналіз атрибутів якості у вимогах, залежність характеристик якості від атрибутів та залежність якості від її характеристик. Метод та програмний засіб оцінки якості ПЗ [90] забезпечують кількісну оцінку характеристик якості та якості, встановлюють залежності (рівняння) якості від її характеристик та прогнозують рівень якості. Модель прогнозування якості ПЗ, що базується на машинному навчанні [41], адаптивна модель прогнозування гібридних дефектів на основі ШНМ [39], методи аналізу різних аспектів якості ПЗ [95], підхід трьох батьків і генетичної еволюції (3PcGE) для прогнозування дефектів ПЗ [42] забезпечують кількісне визначення характеристик якості та кількісне визначення якості. Система нечіткого виводу для прогнозування якості ПЗ на основі параметрів вимог [50], застосування нейромережних моделей та технологій для прогнозування дефектів ПЗ [51], модель на основі ШНМ для прогнозування дефектів ПЗ [100] забезпечують аналіз атрибутів якості у вимогах, кількісну оцінку характеристик якості та кількісну оцінку якості. Модель на основі штучних нейронних мереж для оптимізації надійності та якості програмного забезпечення [49] передбачає лише аналіз атрибутів якості у вимогах. Методологія на основі машинного екстремального навчання з підтримкою багатоцільової grey wolf оптимізації для прогнозування якості ПЗКС [107] передбачає кількісну оцінку якості, прогнозування рівня якості та використання єдиних методологічних підходів з можливістю інтеграції всіх методів та інструментів. Експертна система для прогнозування якості використання на ранніх стадіях розробки на основі штучних нейронних мереж [106], методи на основі нечіткої математики для оцінки якості тестування ПЗ [94], модель надійності та якості ПЗ зі збільшеною кількістю нейронних мереж [104] забезпечують кількісну оцінку якості та прогнозування рівня якості. Моделі

прогнозування повторного використання ПЗКС (в т.ч. штучна нейронна мережа) [36] забезпечують лише прогнозування рівня якості.

Отже, наявні рішення, спрямовані на аналіз атрибутів якості у вимогах, спрямовані на встановлення залежностей (рівнянь) характеристик якості від атрибутів та якості від характеристик, спрямовані на кількісну оцінку характеристик якості та кількісну оцінку якості, спрямовані також на прогнозування рівня якості. Нарешті є навіть рішення, які представляють певні методологічні підходи. Але критерій 7 не задовольняє жоден з проаналізованих методів/інструментів. Система прогнозування рівня якості ПЗ на основі вимог, що забезпечує вищезазначені сервіси в комплексі, на сьогоднішній день відсутня. Таким чином, дослідження значної кількості відомих моделей, методів, інструментальних засобів з точки зору відповідності наведеним вище критеріям показало, що жодне з проаналізованих рішень не задовольняє усім вісьмом визначеним (згідно із стандартом ISO 25010) критеріям у комплексі, тобто жодне з проаналізованих рішень не підходить для оцінювання та прогнозування якості ПЗКС за стандартом ISO 25010).

Отже, на даний момент часу спостерігається суперечність між зростаючою відповідальністю, яка покладається на програмне забезпечення комп’ютерних систем (ПЗКС), та розширенням вимог до якості ПЗКС, з одного боку, і недосконалістю методів та засобів прогнозування якості та безпеки ПЗКС, яка особливо проявляється на ранніх етапах життєвого циклу, з іншого боку.

1.3. Аналіз методів та засобів оцінювання і забезпечення безпеки програмного забезпечення комп’ютерних систем та ідентифікації відмов та вразливостей

Сучасне ПЗКС є складним багатофункціональним виробом, при створенні якого неминуче мають місце помилки, ненавмисні програмні дефекти, незахищені функції. У сучасну цифрову епоху ПЗКС широко адаптоване та стало невід’ємною складовою людського суспільства. Таке широке використання ПЗ пов’язане з

використанням великих і критичних даних, які неминуче потребують захисту. Вкрай важливо переконатися, що це ПЗ не тільки задовольняє потреби користувачів або функціональні вимоги, але не менш важливо забезпечити безпеку цього ПЗ. Створення програмного забезпечення комп’ютерних систем з високим рівнем безпеки є складним процесом. Це процес, неформально керований загальними знаннями, передовою практикою та незадокументованими експертними знаннями. Загалом безпеку ПЗ можна розглядати як одну з найважливіших проблем у галузі розробки програмного забезпечення, оскільки вона може впливати на ефективність програмного продукту через різноманітні технологічні вразливості та загрози.

Безпека програмного забезпечення є властивістю певного програмного забезпечення функціонувати без різних негативних наслідків для конкретної комп'ютерної системи. Безпека ПЗКС як характеристика якості ПЗ розглядається як забезпечення захисту від потенційних загроз, таких як несанкціонований доступ, зловживання даними та системні атаки [3].

Розглянемо відомі методи та інструменти для прогнозування рівня безпеки ПЗКС як складової якості ПЗ.

В роботі [108] запропоновано підхід до прогнозування безпеки структури та функцій ПЗКС з використанням рекурентних нейронних мереж з хорошим захисним ефектом, який може бути застосований до неправомірного використання інформації, інформаційних аномалій та реакцій системи безпеки.

У роботі [109] запропоновано метод на основі глибокого навчання для виявлення вразливостей, який може навчатися і автоматично генерувати шаблон вразливості, а також метод на основі графових нейронних мереж для виявлення та інтерпретації вразливостей на рівні зрізу. Ці методи нормалізують сирцевий код, виділяють зрізи, щоб зменшити втручання надлишкової інформації, а зрізи вразливостей подаються в інтерпретатор вразливостей для отримання конкретних рядків коду вразливостей. Ці методи коректно виявили 59 реальних вразливостей у чотирьох програмних продуктах з відкритим сирцевим кодом.

Автори [110] пропонують методологію для аналізу безпеки програмного забезпечення та виявлення інцидентів безпеки, створюють модель на основі глибокого навчання та штучного імунітету для виявлення інцидентів безпеки, розробляють метод на базі штучної імунної системи та згорткової нейронної мережі для класифікації та оптимізації інцидентів безпеки, а також створюють програмний пакет для виявлення інцидентів безпеки ПЗКС.

У роботі [72] запропоновано різні моделі машинного навчання для прогнозування дефектів програмного забезпечення, ефективність яких залежить від якості набору даних, від проблем з даними (розмірність даних, перекриття класів, дисбаланс класів, пропущені дані) і може бути покращена шляхом підвищення якості набору даних, включаючи якість даних, попередню обробку даних, моделювання даних, продуктивність даних тощо.

Автори [111] досліджують прогнозування безпеки для оцінки параметрів ефективності кожного програмного компонента, для аналізу ефективності та якості програмного забезпечення та для аналізу основних аспектів до етапу розробки ПЗКС, використовуючи розширену класифікацію ШНМ прямого поширення CatBoost.

У статті [112] розглядаються деякі моделі, методи, інструменти та стандарти оцінювання безпеки ПЗКС та забезпечення якості і безпеки за допомогою підходів на основі машинного навчання для прогнозування, оптимізації, виявлення особливостей та підвищення ефективності прогнозування дефектів ПЗ.

Автори роботи [113] розробляють оптимізовану модель на основі машинного навчання для прогнозування дефектів програмного забезпечення з метою підвищення безпеки ПЗКС. Важливі ознаки програмного забезпечення відбираються за допомогою методу оптимізації мурашиних колоній, після чого відібрані ознаки подаються на вхід опорно-векторної машини як вхідні дані.

Метою дослідження [114] є прогнозування безпеки програмного забезпечення з вищою точністю, ніж попередні методи та інструменти. Автори дослідження доводять, що алгоритми машинного навчання з попередньою обробкою даних та виділенням ознак на наборах даних з програмними метриками дають більш точні результати при прогнозуванні безпеки ПЗКС.

У статті [115] досліджено вплив предметної області та атрибутів якості ПЗ на прогнозування безпеки ПЗ методами глибокого навчання з використанням різних наборів даних. Цінність цього дослідження полягає в тому, що воно підвищує рівень ідентифікації атрибутів якості при підготовці вимог і допомагає інженерам по розробці вимог зрозуміти, на яких питаннях вимог слід зосередитися.

Автори роботи [116] аналізують зв'язок між покращенням вимог до ПЗ та якістю і безпекою ПЗКС. Аналіз показує, що безпека та якість ПЗ залежить від атрибутів стандартів ISO/IEC 25010, IS\IEC 25023. Дослідження емпірично показує, що покращення вимог призводить до покращення безпеки та якості ПЗ.

Автори роботи [117] розробляють модель прогнозування дефектів програмного забезпечення та модель прогнозування супроводжуваності програмного забезпечення, які базуються на дереві рішень як більш ефективному класифікаторі. Крім того, автори розробляють фреймворк на основі набору керівних принципів для покращення безпеки програмного забезпечення.

Автори [118] використовують узагальнену регресійну нейронну мережу з покращеним алгоритмом пошуку для відображення нелінійного зв'язку між метриками ПЗ та характеристиками якості ПЗ, а також пропонують модель прогнозування безпеки і якості ПЗ на основі GRNN для підвищення точності прогнозування дефектів ПЗ.

У дослідженні [119] була розроблена структура одношарової радіально-базисної мережі з використанням тонкопластинчастого сплайну RBF для прогнозування безпеки та якості програмного забезпечення. Ця запропонована мережа була протестована на п'яти невідомих зразках програмного забезпечення, і виявлено, що прогнозовані рівні безпеки та якості дуже точно відповідають фактичним характеристикам безпеки та якості ПЗКС.

Автори роботи [120] розробляють модель машинного навчання з семи ансамблів для прогнозування дефектів програмного забезпечення на основі Cat boost. Отримані результати доводять, що запропонована модель Cat boost забезпечує високу продуктивність для всіх трьох наборів даних дефектів, зменшуючи перенавчання та скорочуючи час навчання.

У роботі [121] емпірично продемонстровано ефективність передбачення дефектів десятьма ансамблевими предикторами. В роботі використано 15 програмних проєктів з репозиторію PROMISE, а результати експериментів демонструють, що ансамблеві предиктори покращують ефективність виявлення дефектів.

Автори [122] використовують методологію оптимізації і пропонують цільову функцію для прогнозування якості та безпеки програмного забезпечення з кращими результатами на MATLAB з реальними даними.

Методи та інструменти, що базуються на штучному інтелекті, для прогнозування рівня безпеки та якості програмного забезпечення, мають великий потенціал. Проте, вони не встановлюють зв'язок між безпекою ПЗ та атрибутами якості, не формують числове значення безпеки програмного забезпечення на основі цих атрибутів та не забезпечують прогнозування рівня безпеки програмного забезпечення на основі отриманого числового значення.

Причини порушення безпеки можуть бути різними: відмови та збої ПЗ, вразливості ПЗ через помилки програмістів та дефекти в програмах. Відмова ПЗКС (failure) – це подія, що характеризується порушенням роботоздатності ПЗ, внаслідок якого ПЗ припиняє виконувати свої функції (цілком або частково) [123].

Вразливість ПЗКС (vulnerability) – це недолік ПЗ (недолік проєктування ПЗ, помилка програмування, застосування шкідливого ПЗ), при використанні якого можна навмисне порушити цілісність ПЗ та викликати його некоректну роботу; це нездатність ПЗ протистояти реалізації певної загрози або сукупності загроз [124].

Щороку з’являються тисячі нових уразливостей, які потребують від компаній виправлення системного програмного забезпечення, застосунків, переналаштування безпекових параметрів всього мережевого середовища. Для запобігання використанню цих уразливостей у кібератаках, організації, які ставляться до безпеки мережевого середовища серйозно, використовують управління вразливостями з метою забезпечення максимально можливого рівня захисту та безпеки. Виявлення вразливостей ПЗ є важливим методом забезпечення безпеки ПЗКС. Сьогодні, коли розмір і складність ПЗ швидко зростають, вразливості стають різноманітними, і їх стає все важче ідентифікувати.

Основними причинами появи вразливостей є:

1) полегшення обміну інформацією між мережевими взулами та забезпечення спільного використання ресурсів;

2) суттєве ускладнення ПЗКС;

3) брак повної інформації про об’єкт;

4) велика кількість зловмисників при ненадійних джерелах даних;

5) недостатня кваліфікація користувачів ПЗКС, особливо з питань безпеки та захисту інформації, що призводить до несвідомого виконання руйнівних дій користувачами під впливом зловмисників;

6) складність нових технологій;

7) вбудовування сирцевого коду (макросів, сценаріїв) у документи, об’єднання даних і сирцевого коду;

8) відставання у розвитку нормативно-правової бази, стандартів від змін методів та технологій обробки інформації;

9) відсутність безпечних процесів у життєвому циклі розробки ПЗКС.

Швидке збільшення обчислювальної потужності комп'ютерів та обсягів даних, а також розширення кола завдань, які вирішуються програмним забезпеченням, ускладнюють проведення повного та детального аналізу можливих вразливостей та виключення умов їх виникнення. Наразі чимало провідних вчених провели ряд досліджень щодо підвищення безпеки ПЗКС, проте вразливості та відмови ПЗ все ще створюють серйозні проблеми для користувачів, проявляючись витоками інформації, втратою інформації, призводячи до фінансових та репутаційних втрат. Так, наприклад, через вразливості ПЗ стався витік інформації у вигляді доступу до 500 млн. записів користувачів Yahoo [125]; компанія Equifax втратила інформацію про 140 млн осіб, що призвело до фінансових втрат у 575 млн дол. США [126]; зловмисники отримали доступ до 50 млн. профілів користувачів Facebook [127]; викрадена інформація про 600 тис. водіїв і 57 млн облікових записів користувачів сервісу Uber, що призвело до фінансових втрат у 148,1 млн дол. США [128]; відбулась хакерська атака на урядові сайти України 14 січня 2022 року, спричинена вразливістю системи керування вмістом веб-сайтів October CMS [129].

Згідно зі статистикою [130], лише у 2018 році було зафіксовано 312 випадків вразливостей додатків Android та 87 випадків вразливостей додатків iOS. Взагалі, відкритий код Android робить цю операційну систему та Android додатки основною мішенню для шкідливих програм [131-133]. Механізм міжкомпонентної взаємодії Android може викликати проблеми з безпекою, такі як порушення політики безпеки додатків [134]. За даними бенчмарк-тестування NowSecure [135], 85% досліджених додатків мали один або більше ризиків для безпеки. Понад 50% досліджених додатків мали бот-вузли, які призводили до проблем із захистом даних під час передачі. Близько третини протестованих додатків мали проблеми з сирцевим кодом. Зокрема, додатки для Android мали проблеми з кодом, які могли призвести до зворотного інжинірингу та інших загроз. Згідно із [136], коли йдеться про безпеку мобільних додатків, основними проблемами, які виникають найчастіше, є неналежне використання платформи, незахищене зберігання даних, незахищений зв'язок між клієнтом і сервером, незахищена автентифікація (наприклад, парольна автентифікація користувачів накладає ряд обмежень і більше не вважається безпечною та зручною для мобільних користувачів, натомість останнім часом все більшу увагу привертає біометрична автентифікація користувачів як перспективне рішення для підвищення мобільної безпеки [137, 138]), ненадійна авторизація, недостатнє шифрування даних, низька якість коду, підробка коду, ризик зворотного інжинірингу та зайва функціональність.

Проведений аналіз відомих методів і технологій, що стосуються забезпечення безпеки програмного забезпечення комп'ютерних систем (як здатності ПЗКС працювати без негативних наслідків для конкретної комп'ютерної системи) та виявлення збоїв і вразливостей програмного забезпечення [139-164], наведений у Додатку В.

Так, метод оцінки безпеки ПЗКС на основі загроз з акцентом на існуючі загрози для об'єктів ПЗ [139], метод підтвердження безпеки ПЗКС для забезпечення створення прийнятно безпечного для замовника ПЗ [140], модель безпеки ПЗ на основі даних та методи вивчення детальної статистики ПЗ з одночасним забезпеченням диференційованої конфіденційності для його користувачів [141],

метод безпеки програмного забезпечення (CM-Sec), що фокусується на кінцевому продукті, який забезпечує розширення дерев атак та процес ідентифікації та визначення пріоритетності контрзаходів [142], метод Q-навчання, вбудований як частина самого ПЗКС для забезпечення механізму безпеки [143], методи, прийоми та найкращі практики інженерії та управління вимогами хмарних сервісів, що розвиваються (SSREMaaES), та керівництво з безпеки ПЗ як сервісу [144], ієрархічний метод розробки кейсів безпеки ПЗ [146], фреймворк моделювання та верифікації безпеки вбудованого ПЗ на основі напівформальних та формальних методів ZMsec [147], SMASHUP: інструментарій для уніфікованої верифікації спільних проєктів ПЗКС [148], метод виявлення безпекових вразливостей у вимогах до ПЗ, розроблених структурованою об'єктно-орієнтованою формальною мовою [149], формальний метод для моделювання архітектур програмного забезпечення та оцінки їх атрибутів якості (включаючи безпеку, надійність та продуктивність) кількісно та уніфіковано [150], модель Trustworthy Scrum (TS), що дозволяє безпековій діяльності співпрацювати з гнучкими методами та працювати в рамках Scrum [151] призначені лише для забезпечення безпеки програмного забезпечення комп’ютерних систем.

Таксономія для ідентифікації режимів відмов ПЗКС, які забезпечують вхідні дані для аналізу ризиків програмно-інтенсивних систем [154], використання розподілу позицій паттернів як ознак для виявлення відмов ПЗКС [153], Pangr: система для автоматичного виявлення, експлуатації та виправлення вразливостей [160] призначені лише для виявлення збоїв та вразливостей програмного забезпечення комп’ютерних систем.

Методологія мінімізації вразливостей ПЗ для підвищення його безпеки [145], метод аналізу відмов ПЗКС на основі моделювання надійності системи за допомогою System-Theoretic Accident Modeling and Processes (STAMP) [152], метод каскадної локалізації несправностей та програмний інструмент CaFL для прискорення трудомісткого процесу ідентифікації першопричини проявленої несправності [155], метод Failure Identification for Complex Mission Analysis (FICMA), який забезпечує загальний аналіз відмов функціональності системи [156], алгоритм

прогнозування відмов на основі багатошарової двонаправленої пам'яті [157], метод виявлення вразливостей потоку даних ПЗ на основі вдосконаленої згорткової нейронної мережі для ефективного виявлення помилок передачі в потоці даних ПЗ [158], метод виявлення та ізоляції вразливостей ПЗ як у неконтрольованому, так і в напівконтрольованому контекстах [159], автоматизований метод визначення кодових ознак наявності вразливостей у застарілих версіях ПЗКС [161], підхід до виявлення вразливостей на основі патернів [162], метод виявлення вразливостей у сирцевому коді ПЗКС на основі моделі інтерпретованості Convolution Neural Networks (CNN) та Global Average Pooling (GAP) [163], VUDENC (Vulnerability Detection with Deep Learning on a Natural Codebase): інструмент для виявлення вразливостей на основі глибокого навчання [164] призначені як для забезпечення безпеки програмного забезпечення комп’ютерних систем, так і для виявлення збоїв та вразливостей програмного забезпечення комп’ютерних систем.

Аналіз відомих методів і технологій для забезпечення безпеки програмного забезпечення та виявлення збоїв і вразливостей комп'ютерних систем підтвердив, що хоча ці методи та технології мають значний потенціал, жодне з відомих рішень не призначене для ідентифікації, класифікації та прогнозування відмов та вразливостей програмного забезпечення комп'ютерних систем.

У [165] наведено визначення типів відмов та вразливостей. Таким чином, незначна відмова програмного забезпечення комп’ютерних систем – це зупинка роботи програми без втрати даних і необхідності перезавантаження комп'ютера на час, що перевищує порогове значення. Значна відмова – це зупинка роботи програми з втратою всіх або частини даних без необхідності перезавантаження комп'ютера на час, що перевищує порогове значення. Критична відмова – це завершення роботи програми, яке вимагає перезавантаження комп'ютера [165].

Вразливість коректної роботи – це функційна можливість, що спричиняє припинення функціонування програми на час, що перевищує порогове значення, тобто до відмови програмного забезпечення. Вразливість цілісності інформації – це функційна можливість ПЗКС, яка призводить до несанкціонованої (зловмисної або випадкової) зміни даних та/або до втрати повноти даних під час виконання даної

функційної можливості. Вразливість конфіденційності інформації – це функційна можливість ПЗКС, яка призводить до несанкціонованого розголошення, витоку, незаконного використання інформації. Вразливість доступності інформації – це функційна можливість ПЗКС, яка призводить до неможливості безперешкодної реалізації наявних прав доступу суб'єктами.

Вразливістю доступності інформації є функціональна особливість ПЗКС, яка призводить до того, що суб'єкти, які мають права доступу до інформації, не можуть їх безперешкодно реалізувати [165].

Враховуючи наведені визначення типів відмов, у [165] запропоновано наступні критерії класифікації відмов:

1) втрата працездатності (здатності функціонувати) ПЗКС;
2) втрата даних (повністю або частково) програмного забезпечення;
3) необхідність перезавантаження комп'ютера.

Враховуючи наведені визначення типів вразливостей, у [165] запропоновано наступні критерії для класифікації вразливостей:

1) виникнення відмови ПЗКС;
2) втрата повноти даних;
3) витік несанкціонованої інформації;
4) неможливість отримати дозволену інформацію.

Враховуючи запропоновані типи та критерії класифікації відмов і вразливостей, у [165] запропоновано правила класифікації відмов та вразливостей, які будуть використовуватись в подальшій роботі при розробленні методу і системи ідентифікації та класифікації відмов та вразливостей ПЗКС.

1.4. Висновки. Постановка задачі

Постійний розвиток наявних комп’ютерних систем та розробка нових комп’ютерних систем з використанням хмарних обчислень, штучного інтелекту, віртуальної та доповненої реальності підвищують вимоги до процесу оцінювання та прогнозування якості програмного забезпечення комп’ютерних систем. З

приростом надійності апаратного забезпечення та збільшенням складності програмного забезпечення, якість і безпека програмного забезпечення комп'ютерних систем стають предметом все більшого хвилювання, як серед розробників, так і серед користувачів, особливо в контексті досягнення бізнес-цілей. Забезпечення необхідної якості та безпеки ПЗ стало зараз стратегічною задачею життєвого циклу сучасних програмних проєктів.

Оцінювання та прогнозування якості та безпеки ПЗКС є одним з ключових завдань, що виникають при розробці програмного забезпечення комп’ютерних систем через необхідність всебічного врахування його впливу на якість апаратних та програмних компонентів критично важливих систем.

Розвиток сучасних технологій та методологій проєктування та розроблення ПЗКС вимагає динамічного вдосконалення засобів оцінки, а особливо засобів прогнозування якості та безпеки ПЗКС на ранніх етапах життєвого циклу, щоб запровадити превентивні заходи для зменшення кількості помилок та збоїв у програмному забезпеченні; щоб обгрунтовано обирати той чи інший набір вимог з множини різних альтернатив, запропонованих різними розробниками для розв’язку однієї й тієї ж задачі; щоб розуміти, чи може бути розроблене якісне ПЗ на основі зібраних вимог до ПЗ.

Досліджені методи та засоби прогнозування безпеки та якості ПЗКС мають великий потенціал для розв’язання різних задач, можуть бути використані в різних контекстах, проте вони не забезпечують обчислення і не задають залежності значень характеристик якості від значень атрибутів, не забезпечують обчислення і не задають залежності значення якості від значень характеристик якості та не забезпечують прогнозування рівня якості та/або безпеки ПЗКС на основі отриманих кількісних значень якості та/або безпеки.

Отже, наразі існує суперечність між зростаючою відповідальністю, яка покладається на програмне забезпечення комп’ютерних систем (ПЗКС), та розширенням вимог до якості ПЗКС, з одного боку, і недосконалістю методів та засобів прогнозування якості та безпеки ПЗКС, особливо на ранніх етапах життєвого циклу, з іншого боку. Відтак, прогнозування і оцінювання рівня якості

та безпеки програмного забезпечення комп’ютерних систем на ранніх етапах життєвого циклу на основі атрибутів якості є *актуальною науково-прикладною задачею,* одним із шляхів розв’язання якої є розроблення методів і засобів прогнозування рівня якості та безпеки ПЗКС.

Розв’язання окресленої науково-прикладної задачі потребує комплексних досліджень за наступними напрямками:

– розробити метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп’ютерних систем;

– розробити метод прогнозування рівня якості програмного забезпечення комп’ютерних систем на основі атрибутів якості;

– розробити метод прогнозування рівня безпеки програмного забезпечення комп’ютерних систем;

– розробити метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем;

– спроєктувати та реалізувати системи прогнозування рівня якості та безпеки програмного забезпечення комп’ютерних систем;

– спроєктувати та реалізувати систему ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем.

Метою дисертаційного дослідження є забезпечення оцінювання наявного набору вимог з позиції прогнозованого рівня якості та безпеки програмного забезпечення комп’ютерних систем, яке планується до реалізації за таким набором вимог, шляхом розроблення методів та засобів прогнозування рівня якості та безпеки ПЗКС.

Об’єкт дослідження – процеси прогнозування рівня якості та безпеки програмного забезпечення комп’ютерних систем.

Предмет дослідження – методи та засоби прогнозування рівня якості та безпеки програмного забезпечення комп’ютерних систем.

РОЗДІЛ 2.
ПРОГНОЗУВАННЯ РІВНЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ АТРИБУТІВ ЯКОСТІ

2.1. Метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп'ютерних систем [64]

Своєчасне прогнозування якості ПЗКС може бути використано для прийняття будь-яких превентивних заходів для зменшення кількості збоїв ПЗ під час його роботи. Бажано вже за наявності специфікації вимог до ПЗ розуміти, чи за цією специфікацією може бути розроблене якісне ПЗ. Ряд проведених досліджень [76] показали, що версії ПЗКС, написані різними розробниками за однаковими вимогами, як правило, містять ряд спільних помилок, пов'язаних із помилками або неточностями вимог, і навпаки версії ПЗКС, написані одним колективом розробників за різними вимогами, суттєво різнились своєю якістю та успішністю. Загальновизнано, що якісна інженерія вимог приводить до підвищення якості програмного забезпечення комп'ютерних систем і значно знижує ризик невдачі або перевитрати бюджету проєктів з розробки програмного забезпечення [88].

Оцінка якості програмного забезпечення згідно зі стандартом ISO 25010 [3] виконується таким чином: на базі 138 атрибутів якості, що визначені у стандарті ISO 25023 [89], проводиться оцінка підхарактеристик та характеристик якості, за якими, у свою чергу, формується комплексна оцінка якості програмного забезпечення. Зараз оцінка атрибутів якості відбувається, в основному, для готового сирцевого коду, проте всі атрибути якості описані у вимогах до ПЗ. Саме у вимогах до ПЗ описано поведінку майбутнього ПЗ, функціональні можливості та обмеження ПЗ, його властивості та атрибути. Значення атрибутів якості враховують і мету програмного проєкту, і його тип. Якщо атрибут(и) вписано у вимоги специфікації, визначено значення такого атрибуту, відтак розробники будуть зобов'язані забезпечити наявність такого атрибуту в своєму ПЗ і зазначене

у вимогах значення атрибуту, інакше ПЗ не пройде верифікації та валідації. Отже, на основі значень атрибутів, наявних у вимогах, може бути спрогнозовано рівень якості ПЗКС будь-якого типу.

Тоді задача синтезу полягає у розробленні моделі, методу та системи прогнозування рівня якості ПЗКС на основі значень атрибутів якості з вимог до ПЗ, а задача аналізу – до отримання прогнозованого рівня якості ПЗКС.

При вирішенні таких завдань ключовим стає автоматизація обробки вимог і усунення участі людини у процесах обробки інформації та набуття знань. Тому, при створенні майбутньої системи для прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості, важливо забезпечити автоматичний аналіз вимог до програмного забезпечення з метою пошуку значень атрибутів якості.

Враховуючи атрибути якості, зазначені в ISO 25023 [89], та використовуючи метод ідеалізації (накладення певних обмежень), розробимо відповідну *структуру специфікації вимог до ПЗКС* для виконання препроцесінгу специфікації з метою її подальшого автоматичного аналізу.

Отже, специфікація вимог до ПЗ, придатна для автоматичного опрацювання майбутньою системою для прогнозування рівня якості програмного забезпечення комп’ютерних систем на основі атрибутів якості, повинна містити такі 138 атрибутів якості (атрибути можуть бути наявні у специфікації вимог тільки, якщо після назви атрибуту в специфікації наявне значення атрибуту): 1) кількість функцій (number of functions); 2) повнота функціональної реалізації (functional implementation completeness); 3) функціональна адекватність (functional adequacy); 4) покриття функціональної реалізації (functional implementation coverage); 5) час роботи (operation time); 6) кількість неточних обчислень, з якими мають справу користувачі (number of inaccurate computations encountered by users); 7) кількість елементів даних (number of data items); 8) точність обчислень (computational accuracy); 9) точність (precision); 10) кількість завдань (number of tasks); 11) час відгуку (response time); 12) кількість оцінок (number of evaluations); 13) час виконання (turnaround time); 14) час виконання завдання (task time); 15) середня

пропускна здатність (mean amount of throughput); 16) кількість відмов (number of failures); 17) кількість помилок, пов'язаних з вводом/виводом (number of IO related errors); 18) час очікування користувачем використання пристрою вводу/виводу (user waiting time of IO device utilization); 19) кількість помилок, пов'язаних з пам'яттю (number of memory related errors); 20) кількість помилок, пов'язаних з передачею даних (number of transmission related error); 21) пропускна здатність каналу передачі даних (transmission capacity); 22) завантаженість вводу/виводу (кількість буферів) (IO utilization (number of buffers)); 23) кількість рядків безпосередньо коду (number of line of code directly); 24) ліміти завантаження вводу/виводу (IO loading limits); 25) максимальне завантаження пам'яті (maximum memory utilization); 26) максимальне завантаження передачі (maximum transmission utilization); 27) середня частота виникнення помилок передачі (mean occurrence of transmission error); 28) кількість одночасних користувачів (number of concurrent users); 29) пропускна здатність зв'язку (communication bandwidth); 30) розмір бази даних (size of database); 31) кількість інструкцій (number of tutorials); 32) кількість елементів даних вводу/виводу (number of IO data items); 33) повнота опису (completeness of description); 34) зрозумілість функцій (function understandability); 35) зрозумілість вводу/виводу (understandable input and output); 36) легкість вивчення функцій (ease of function learning); 37) частота надання допомоги (help frequency); 38) ефективність документації для користувача та/або системи допомоги (effectiveness of the user documentation and/or help system); 39) доступність допомоги (help accessibility); 40) повнота документації користувача та/або довідкової системи (completeness of user documentation and/or help facility); 41) кількість виправлених помилок (error correction); 42) кількість екранів або форм (number of screens or forms); 43) кількість помилок, зроблених користувачем (number of user errors or changes); 44) кількість спроб налаштування (number of attempts to customize); 45) кількість операцій (number of operations); 46) кількість елементів, дані яких можна перевірити на валідність (number of items which could check for valid data); 47) кількість реалізованих повідомлень (number of messages implemented); 48) кількість елементів інтерфейсу (number of interface elements); 49)

фізична доступність (physical accessibility); 50) кількість легко зрозумілих повідомлень (number of easily understood messages); 51) кількість невдало вирішених ситуацій (number of unsuccessfully recovered situation); 52) період часу роботи при спостереженні (operation time period during observation); 53) кількість випадків помилкової діяльності користувача (number of occurrences of user's human error operation); 54) кількість помилок введення, успішно виправлених користувачем (number of input errors which the user successfully corrects); 55) кількість спроб виправлення помилок введення (number of attempts to correct input errors); 56) кількість помилкових умов, успішно виправлених користувачем (number of error conditions which the user successfully corrects); 57) загальна кількість помилкових умов, що підлягали тестуванню (total number of error conditions tested); 58) кількість функцій, реалізованих з толерантністю до помилок користувача (number of functions implemented with user error tolerance); 59) загальна кількість функцій, що вимагають толерантності (total number of functions requiring the tolerance capability); 60) загальна кількість неправильних шаблонів роботи (total number of incorrect operation patterns); 61) кількість графічних елементів інтерфейсу (number of interface graphical elements); 62) ступінь збільшення зручності використання для користувача (degree of increase the pleasure of user); 63) ступінь збільшення задоволення користувача (degree of increase the satisfaction of user); 64) ступінь ергономічної привабливості (degree of ergonomic attractiveness); 65) ступінь використання метафор реального світу (degree of real world metaphors use); 66) ступінь можливості використання програмного забезпечення користувачами з обмеженими можливостями (extent to which software can be used by users with specified disabilities); 67) ефективність роботи користувачів з особливими потребами (effectiveness of work of users with specified disabilities); 68) відсутність ризику для користувачів з особливими потребами (freedom from risk for users with specified disabilities); 69) задоволеність користувачів з обмеженими можливостями (satisfaction of users with specified disabilities); 70) наявність властивостей, що підтримують доступність (presence of properties that support accessibility); 71) кількість збоїв (number of faults); 72) розмір продукту (product size); 73) кількість

тестових кейсів (number of test cases); 74) кількість усунутих відмов (number of resolved failures); 75) кількість виправлених збоїв (number of corrected faults); 76) щільність відмов по відношенню до тестових кейсів (failure density against test cases); 77) кількість усунутих несправностей (failure resolution); 78) кількість усунутих збоїв (fault removal); 79) середній час між відмовами (mean time between failures); 80) зрілість тестів (test maturity); 81) оцінена щільність прихованих помилок (estimated latent fault density); 82) щільність помилок (fault density); 83) загальний час, протягом якого програмне забезпечення перебуває у працездатному стані (total time during which the software is in an up state); 84) кількість спостережуваних несправностей (number of observed breakdowns); 85) загальний час простою (total down time); 86) кількість несправностей (number of breakdowns); 87) кількість несанкціонованих операцій (number of illegal operations); 88) час ремонту (time to repair); 89) час простою (down time); 90) кількість перезапусків (number of restarts); 91) кількість відновлень (number of restoration); 92) відновлюваність (restartability); 93) кількість форматів даних, що розглядаються інструментом (number of data formats regarded by tool); 94) кількість форматів даних, які використовуються при обміні (number of data formats to be exchanged); 95) кількість інтерфейсних протоколів (number of interface protocols); 96) обмінність даних (data exchangeability); 97) кількість випадків пошкодження даних (number of instances of data corruption); 98) кількість типів доступу (number of access types); 99) кількість контрольованих вимог (number of controllability requirements); 100) контрольованість доступу (access controllability); 101) кількість правильно зашифрованих/розшифрованих елементів даних (number of data items correctly encrypted/decrypted); 102) кількість елементів даних, що потребують шифрування/розшифрування (number of data items to be required encryption/decryption); 103) кількість подій, що обробляються з використанням цифрового підпису (number of events processed using digital signature); 104) кількість подій, що потребують властивості неспростування (number of events requiring non-repudiation property); 105) кількість доступів до даних та системи, які реєструються у системному журналі (number of accesses to system and data recorded in the system

log); 106) кількість фактичних доступів (number of accesses actually occurred); 107) кількість наданих методів автентифікації (number of provided authentication methods); 108) кількість зроблених модифікацій (number of modifications made); 109) кількість змінних (number of variables); 110) кількість модулів (number of modules); 111) функціональна спільність (functional commonality); 112) нефункціональна спільність (nonfunctional commonality); 113) розмаїття варіативності (variability richness); 114) застосовність (applicability); 115) пристосованість (tailorability); 116) замінність компонентів (component replaceability); 117) час помилки (error time); 118) кількість елементів, що підлягають реєстрації (number of items required to be logged); 119) кількість необхідних діагностичних функцій (number of diagnostic functions required); 120) можливість ведення журналу аудиту (audit trail capability); 121) кількість переглянутих версій (number of revised versions); 122) можливість контролю за змінами (change control capability); 123) кількість несправностей за певний період часу до модифікації (number of troubles within certain period before modification); 124) кількість несправностей за той самий період після модифікації (number of troubles in same period after modification); 125) кількість необхідних вбудованих тестових функцій (number of built in test functions required); 126) кількість тестових залежностей від інших систем (number of test dependencies on other systems); 127) кількість контрольних точок (number of checkpoints); 128) зручність перенесення (porting user friendliness); 129) кількість структур даних (number of data structures); 130) адаптивність структур даних (adaptability of data structures); 131) адаптивність апаратного середовища (hardware environmental adaptability); 132) адаптивність програмного середовища (software environmental adaptability); 133) кількість операційних функцій, завдання яких є неадекватними або не були виконані (number of operational functions of which tasks were not completed or adequated); 134) загальна кількість функцій, які пддавались тестунню в різних середовищах (total number of functions which were tested in different environment); 135) кількість операцій з налаштування (number of setup operations); 136) кількість кроків інсталяції (number

of installation steps); 137) простота інсталяції (ease of installation); 138) кількість об'єктів (number of entities).

Розроблена структура специфікації вимог, яка може бути автоматично оброблена майбутньою системою для прогнозування рівня якості ПЗКС на основі його атрибутів якості, має певні обмеження для формулювання вимог, що містять ці атрибути, та призначена для подальшого препроцесінгу специфікацій вимог.

Метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп'ютерних систем включає наступні кроки:

вхідна інформація: специфікація вимог до програмного забезпечення;

1) препроцесінг специфікації вимог – приведення вимог до вигляду, придатного для автоматичної обробки майбутньою системою прогнозування якості ПЗ згідно із вищевизначеною структурою даних; атрибут може бути наявний у вимогах лише в тому випадку, якщо після його назви в специфікації присутнє відповідне значення;

2) автоматичний аналіз специфікації вимог – пошук кожного атрибуту якості в реальних, готових до обробки, вимогах до програмного забезпечення;

3) вибір значень кожного атрибуту якості у реальних, готових до обробки, вимогах до ПЗКС;

вихідна інформація: значення атрибутів якості.

Вхідними даними розробленого методу є вимоги до програмного забезпечення, структуровані за вищенаведеними правилами в частині опису атрибутів якості, результуючими даними методу є значення атрибутів якості, присутні у вимогах до ПЗ.

Етап препроцесінгу у вигляді накладання певних обмежень на формування специфікації вимог до ПЗ шляхом структурування застосовується лише до тих вимог, які містять атрибути якості. Решта вимог до ПЗ можуть бути викладені довільним чином.

Метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп'ютерних систем забезпечує вибір значень атрибутів якості ПЗ з природомовної специфікації вимог до ПЗ, які далі можуть бути використані для

визначення значень характеристик якості ПЗ та для комплексного оцінювання якості ПЗ. Розроблений метод має велике значення для автоматизації обробки вимог та усунення впливу людини на процеси аналізу інформації. Метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп’ютернизх систем є теоретичною основою для розробки модуля автоматичного аналізу вимог до ПЗ у майбутній системі, яка буде прогнозувати рівень якості ПЗ на основі значень його атрибутів якості.

2.2. Моделювання процесу прогнозування характеристик якості програмного забезпечення комп’ютерних систем на основі атрибутів якості [68]

Для вирішення поставленої задачі прогнозування рівня якості програмного забезпечення комп'ютерних систем на ранніх етапах програмного проєкту, базуючись на атрибутах якості, визначених у вимогах до ПЗКС, необхідно спочатку встановити значення 8-ми характеристик якості ПЗ з врахуванням значень 138 атрибутів якості ПЗ (відповідно до стандартів ISO 25010, ISO 25023), що вказані у вимогах до ПЗ. При такому встановленні значень характеристик якості слід врахувати взаємовплив атрибутів при визначенні кожної характеристики якості ПЗ. Це завдання є складним і важкоформалізованим. Теоретико-множинні моделі якості ПЗ згідно зі стандартом ISO 25010 розглядаються у роботах [90, 91]. Штучні нейронні мережі є засобом, який дозволяє узагальнити інформацію, представити багатовимірну функцію від декількох змінних із встановленням залежностей між вхідними і вихідними даними. Використання штучних нейронних мереж забезпечить урахування кількісних значень кожного атрибута якості та взаємного впливу атрибутів якості при кількісному оцінюванні характеристик якості програмного забезпечення комп'ютерних систем.

Отже, для вирішення проблеми формального задоволення якості необхідно створити концептуальну модель процесу прогнозування характеристик якості програмного забезпечення комп'ютерних систем на основі атрибутів якості. Така концептуальна модель буде абстрактною моделлю, яка визначає структуру системи,

що моделюється, властивості її елементів, їх причинно-наслідкові зв'язки, які є суттєвими для досягнення мети моделювання. Вона буде базуватися на штучній нейронній мережі, яка опрацьовуватиме значення атрибутів якості, видобуті автоматично з вимог до ПЗКС, та враховуватиме не тільки значення, а й взаємовплив атрибутів під час кількісного оцінювання характеристик якості ПЗКС (рис. 2.1).

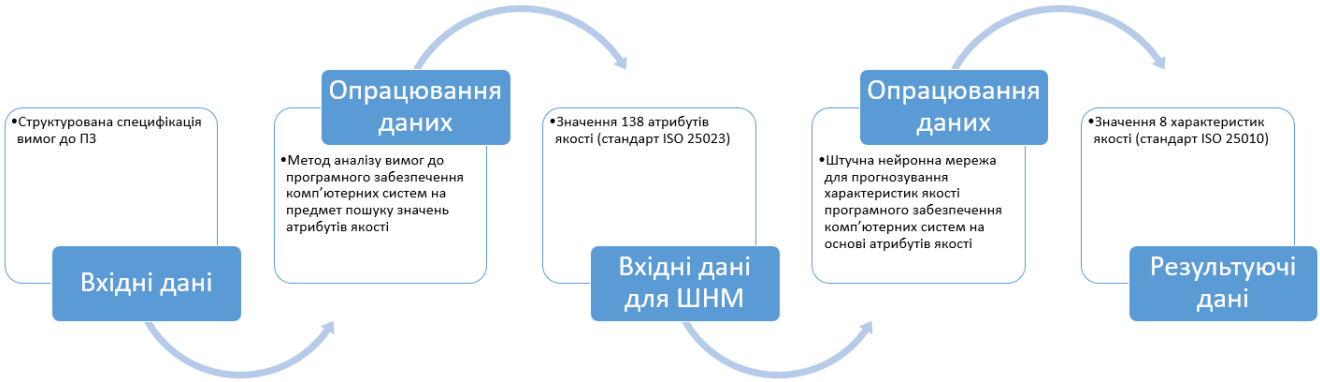


Рис. 2.1 – Концептуальна модель процесу прогнозування характеристик якості програмного забезпечення комп'ютерних систем на основі атрибутів якості

Розробимо тоді ШНМ, яка буде обробляти набір значень атрибутів якості, отриманих з вимог до ПЗКС, апроксимувати їх та надавати прогнозовані кількісні оцінки восьми характеристик якості ПЗ, як визначено у стандарті ISO 25010.

Враховуючи специфіку задачі, виберемо структуру та архітектуру нейронної мережі. Для деяких типів задач існують оптимальні конфігурації. Оскільки задача прогнозування характеристик якості ПЗКС на основі атрибутів не має зворотних зв'язків та властивостей числового ряду, а також не потребує класифікації чи кластеризації даних, то для цієї задачі ми виберемо найпростішу та найбільш досліджену архітектуру – багатошаровий персептрон.

Структура ШНМ для прогнозування характеристик якості ПЗКС на основі атрибутів подана на рис. 2.2.

Згідно із розробленими на основі стандарту ISO 25010 теоретико-множинними моделями якості ПЗ [90, 91], на рис. 2.2 нейрони вхідного шару – це множина значень атрибутів якості $QMS = \{ qms_i \}$, $i = (\overline{1,\ 138})$, визначені у вимогах до ПЗ,

де qms_i – значення i-го атрибуту якості, визначеного стандартом ISO 25023. Нейрони результуючого шару – це множина значень характеристик якості $QCH = \{ qch_l \},\ l = (\overline{1,\ 8})$, де qch_l – значення l-ої характеристики якості, визначеної стандартом ISO 25010. З рис. 2.2 видно, що ШНМ містить також два приховані шари – апроксимуючий та коригуючий.

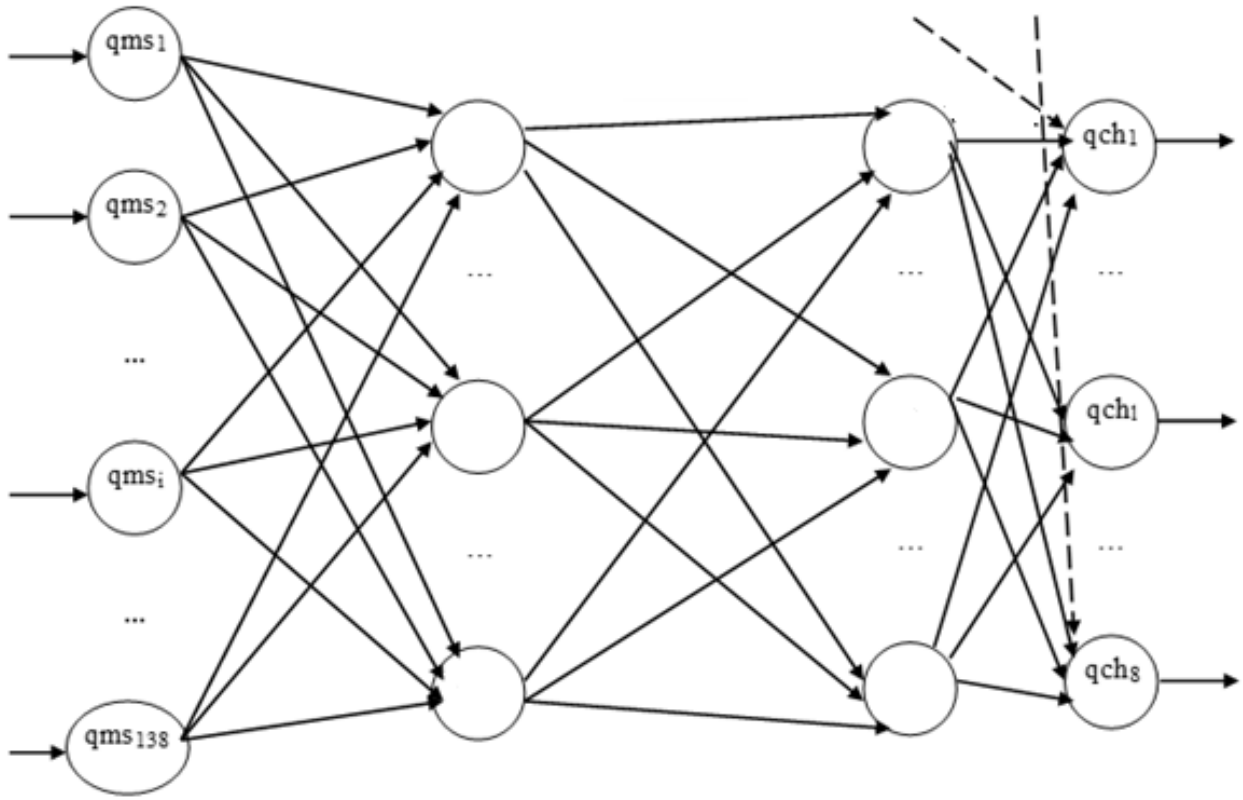


Рис. 2.2 – Структура ШНМ для прогнозування характеристик якості ПЗКС на основі атрибутів

Запропонована ШНМ для прогнозування характеристик якості ПЗКС на основі атрибутів опрацьовує значення 138 атрибутів якості, видобутих автоматично з вимог. Вона проводить їх апроксимацію та дає прогнозовані кількісні оцінки восьми характеристик якості ПЗ: ефективності, функційної придатності, надійності, зручності використання, безпеки, сумісності, можливості переносу та супроводжуваності. Цей метод дозволяє враховувати не лише кількісне значення кожного атрибуту якості, але й їх взаємний вплив при кількісному оцінюванні характеристик якості ПЗКС.

2.3. Метод прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості [60, 69]

Враховуючи запропоновану концепцію прогнозування характеристик якості ПЗКС за допомогою ШНМ, розробимо метод прогнозування рівня якості ПЗКС на основі атрибутів якості, який використовує ШНМ для апроксимації значень атрибутів якості ПЗ, доступних у вимогах до програмного забезпечення. Цей метод передбачає, що ШНМ надає прогнозовані оцінки восьми характеристик якості ПЗ у діапазоні [0;1], де значення «0» вказує на найнижчий рівень кожної з характеристик, а значення «1» вказує на найвищий (найкращий) рівень характеристик. Отже, ШНМ має 138 входів (значення 138 атрибутів якості ПЗ, визначених у вимогах до ПЗ) і 8 виходів (значення 8 характеристик якості ПЗ згідно зі стандартом ISO 25010 у діапазоні [0;1]).

Отримані з ШНМ значення характеристик якості в діапазоні [0;1] є малоінформативними, оскільки як розробнику, так і замовнику важко коректно інтерпретувати ці значення, враховуючи відсутність еталонних значень, а ще важче комплексно спрогнозувати якість програмного забезпечення комп'ютерних систем, тому введемо поняття комплексного показника прогнозованої якості ПЗ, який розраховується на основі значень характеристик якості ПЗ. Оскільки наразі відсутні формули та залежності якості ПЗ від характеристик якості ПЗ, то для розрахунку комплексного показника прогнозованої якості ПЗ використаємо його геометричну інтерпретацію. Для цього побудуємо систему координат з вісьмома напівосями – для вісьмох характеристик якості ПЗ – та кутами між ними, рівними 45° – рис. 2.3.

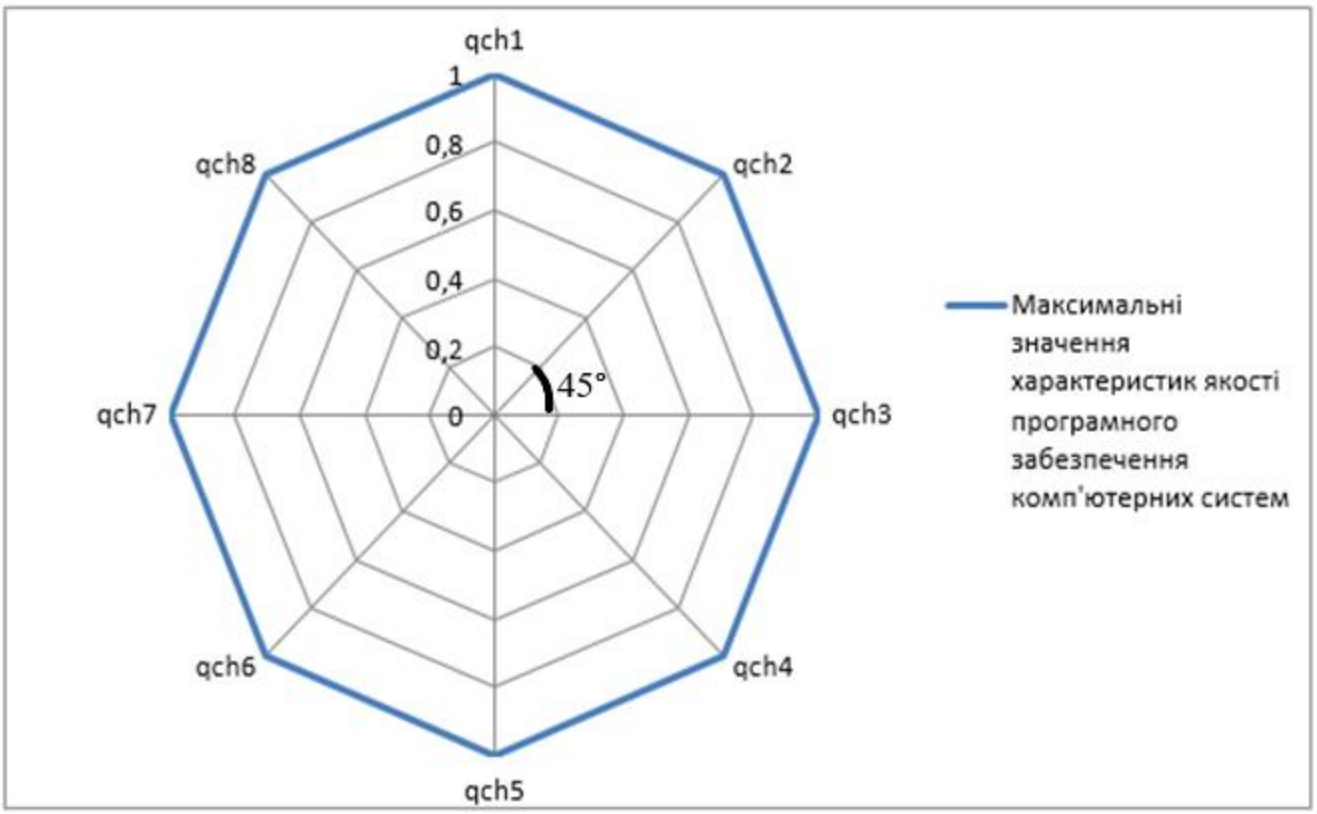


Рис. 2.3 – Система координат для побудови геометричної інтерпретації значень характеристик якості ПЗКС

Геометричною інтерпретацією комплексного показника прогнозованої якості ПЗ буде фігура (восьмикутник), сформована значеннями характеристик якості ПЗ в діапазоні [0;1], наданими ШНМ, а, власне, комплексним показником прогнозованої якості ПЗ буде площа такого восьмикутника (сума площ восьми трикутників, розрахованих за двома відомими сторонами (значення характеристик якості ПЗ) та кутом між ними (кут дорівнює 45° згідно побудованої системи координат)).

Мінімальним значенням комплексного показника прогнозованої якості ПЗКС, очевидно, буде 0 (якщо значення всіх характеристик дорівнюють 0). Розрахуємо максимальне значення комплексного показника прогнозованої якості ПЗ (якщо значення всіх характеристик дорівнюють 1), яке становить 2,8284 ($8 \cdot \frac{1}{2} \cdot 1 \cdot 1 \cdot sin45^{\circ}$).

ШНМ видає прогнозовані оцінки восьми характеристик якості ПЗКС в діапазоні [0;1], і не завжди всі вісім характеристик матимуть приблизно однакові

значення. Звісно, можлива компенсація низьких значень одних характеристик високими значеннями інших характеристик при обчисленні комплексного показника прогнозованої якості ПЗ. Проте всі вісім характеристик є рівноважливими для високої якості майбутнього ПЗ, тому компенсація надто низького значення одного показника значеннями інших показників для отримання однакового значення комплексного показника прогнозованої якості ПЗ не є коректною. Відтак, перш ніж визначати комплексний показник прогнозованої якості ПЗ, варто перевірити припустимість компенсації характеристик якості ПЗ. Оскільки геометричною інтерпретацією комплексного показника прогнозованої якості ПЗ є восьмикутник, сформований значеннями характеристик якості ПЗ, тоді опуклість такого восьмикутника (жоден з його внутрішніх кутів не перевищує 180°) вважатиметься ознакою припустимої компенсації характеристик якості ПЗ.

Для однозначного і простого тлумачення розрахованого значення комплексного показника прогнозованої якості ПЗ, визначимо порогові значення, згідно із якими й прийматиметься висновок про прогнозований рівень якості ПЗ. Для встановлення порогових значень комплексного показника прогнозованої якості ПЗ та для створення правил формування висновку щодо прогнозованого рівня якості ПЗ було проведено аналіз 230 наявних специфікацій вимог до ПЗ, розроблених декількома ІТ-фірмами м. Хмельницького (Україна), для яких згідно з вищеописаною процедурою було визначено прогнозовані оцінки якості ПЗ, та 230 відповідних готових програм, реалізованих за відповідними специфікаціями вимог, для яких відомим є рівень якості. Специфікації вимог до ПЗ та готові програми для аналізу надавались софтверними компаніями м. Хмельницького (Україна) в рамках співпраці із кафедрою комп’ютерної інженерії та інформаційних систем Хмельницького національного університету. Проведений аналіз дав можливість визначити порогові значення комплексного показника прогнозованої якості ПЗ для прогнозування одного з трьох рівнів якості ПЗ – низького (значення комплексного показника прогнозованої якості ПЗ лежать в діапазоні [0; 0,2545)), середнього (значення комплексного показника прогнозованої якості ПЗ лежать в діапазоні

[0,2545; 2,0435)), високого (значення комплексного показника прогнозованої якості ПЗ лежать в діапазоні [2,0435; 2,8284]).

Отже, *метод прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості* складається з наступних кроків:

вхідна інформація: значення атрибутів якості зі специфікації вимог до ПЗ;

1) формування штучною нейронною мережею прогнозованих оцінок восьми характеристик якості ПЗ (зручність використання - qch_1, функційна придатність - qch_2, надійність - qch_3, ефективність - qch_4, безпека - qch_5, сумісність - qch_6, можливість переносу - qch_7, супроводжуваність - qch_8) в діапазоні [0;1] («0» - найнижчий рівень кожної з характеристик, «1» - найвищий (найкращий) рівень кожної характеристики) після обробки та апроксимації значень 138 атрибутів якості з вимог до ПЗ; цей етап базується на концепції прогнозування характеристик якості ПЗ на основі атрибутів з використанням ШНМ;

2) геометрична інтерпретація значень характеристик якості ПЗ в системі координат, представленій на рис. 2.3 – у вигляді восьмикутника, сформованого значеннями характеристик якості ПЗ, наданими ШНМ і відкладеними на кожній з восьми напівосей;

3) перевірка припустимості компенсації характеристик якості ПЗ – шляхом визначення, чи опуклим є восьмикутник (жоден з його внутрішніх кутів не повинен перевищувати 180°); якщо восьмикутник не є опуклим, то компенсація характеристик якості ПЗ є неприпустимою, відтак комплексний показник прогнозованої якості ПЗ не розраховується;

4) розрахунок комплексного показника прогнозованої якості ПЗ як площі побудованого восьмикутника (сума площ восьми трикутників, розрахованих за двома відомими сторонами (значення характеристик якості ПЗ) та кутом між ними (кут дорівнює 45° згідно побудованої системи координат)) – згідно із формулою:

$$\begin{gathered} cifsq = 1/2 \cdot sin45° \cdot (qch_1 \cdot qch_2 + qch_2 \cdot qch_3 + qch_3 \cdot qch_4 + \\ + qch_4 \cdot qch_5 + qch_5 \cdot qch_6 + qch_6 \cdot qch_7 + qch_7 \cdot qch_8 + qch_8 \cdot qch_1) = \\ = 0{,}35355 \cdot (qch_1 \cdot qch_2 + qch_2 \cdot qch_3 + qch_3 \cdot qch_4 + qch_4 \cdot qch_5 + \\ + qch_5 \cdot qch_6 + qch_6 \cdot qch_7 + qch_7 \cdot qch_8 + qch_8 \cdot qch_1) \end{gathered} \quad (2.1)$$

5) якщо значення комплексного показника прогнозованої якості ПЗКС *cifsq* лежить в діапазоні [0; 0,2545), то майбутнє ПЗКС прогнозовано матиме низький рівень якості; якщо значення комплексного показника прогнозованої якості ПЗКС *cifsq* лежить в діапазоні [0,2545; 2,0435), то майбутнє ПЗКС прогнозовано матиме середній рівень якості; якщо значення комплексного показника прогнозованої якості ПЗКС *cifsq* лежить в діапазоні [2,0435; 2,8284], то майбутнє ПЗКС прогнозовано матиме високий рівень якості;

вихідна інформація: прогнозований рівень якості ПЗКС.

Отже, обмеженням розробленого методу є необхідність наявності специфікацій вимог до ПЗКС, а перевагою розробленого методу є його незалежність від мови програмування, якою розроблятиметься майбутнє ПЗКС.

Проілюструємо запропонований метод прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості наступною схемою – рис. 2.4.

Пропонований метод прогнозування рівня якості ПЗКС на основі атрибутів якості відрізняється від відомих тим, що забезпечує можливість прогнозувати якість розроблюваного ПЗКС на основі аналізу значень атрибутів якості, які містяться у вимогах до програмного забезпечення. Таким чином, запропонований метод дозволяє порівнювати специфікації вимог до ПЗ, одразу відмовлятись від реалізації ПЗКС на основі невдалих специфікацій (економія коштів та часу, зменшення ймовірності провальних і проблемних проєктів) та виконувати обґрунтований вибір специфікації для подальшої реалізації ПЗКС саме високої якості (звісно, за умови, що помилки не будуть внесені на наступних етапах життєвого циклу ПЗ).

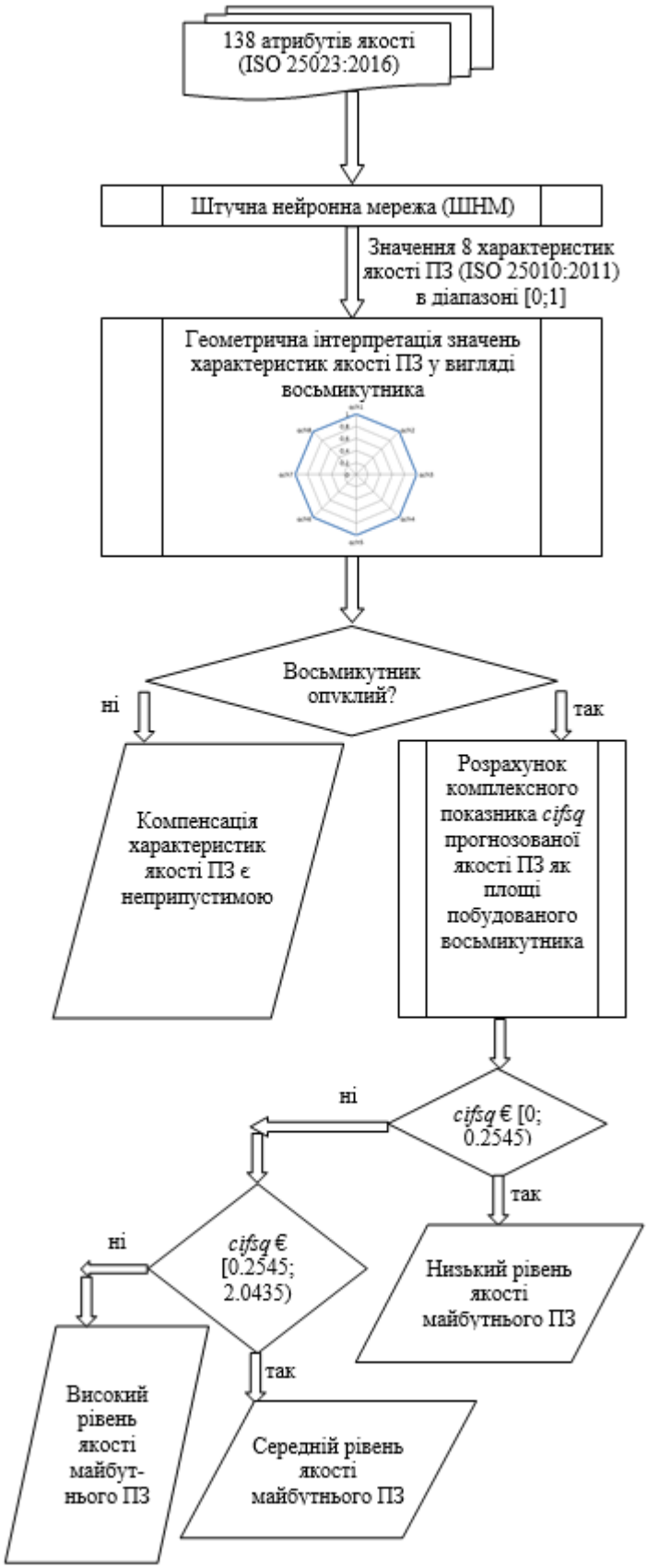


Рис. 2.4 – Схема методу прогнозування рівня якості програмного забезпечення комп’ютерних систем на основі атрибутів якості

2.3.1. Реалізація, навчання та тестування штучної нейронної мережі для прогнозування характеристик якості програмного забезпечення комп'ютерних систем на основі атрибутів якості

Для ефективної роботи розробленого методу прогнозування якості ПЗКС на основі атрибутів якості необхідна штучна нейронна мережа, яка буде використовуватися для апроксимації значень атрибутів якості ПЗ, визначених у вимогах до нього. На цій штучній нейронній мережі базується перший етап методу. Таку ШНМ було реалізовано у пакеті Matlab. При реалізації для зручності 138 вхідних нейронів були розбиті у 3 множини по 46 нейронів. Оператор `gensim(net)` надав візуалізацію розробленої ШНМ у Simulink (рис. 2.5, 2.6).

Для навчання одержаної ШНМ підготовлено навчальну вибірку з 35180 векторів на основі аналізу наявних 1100 специфікацій вимог до ПЗ, наданих для дослідження декількома IT-фірмами м. Хмельницького (Україна), та відповідних готових програм, реалізованих за цими специфікаціями вимог, для яких відомим є рівень якості (якщо програмний проєкт був завершений вчасно, в рамках виділеного бюджету, і розроблене ПЗ має всі визначені специфікацією функціональні можливості, то таке ПЗ має високий рівень якості; якщо програмний проєкт був завершений невчасно та/або з перевищенням виділеного бюджету, та/або розроблене ПЗ не має всіх визначених специфікацією функціональних можливостей, то таке ПЗ має середній рівень якості; якщо програмний проєкт мав суттєві перевитрати часу та бюджету, та/або розроблене ПЗ не має більше половини визначених специфікацією функціональних можливостей, то таке ПЗ має низький рівень якості). Для розрахунку необхідного об'єму навчальної вибірки для ШНМ, яку потрібно навчити з похибкою порядку 0.1 використаємо формулу: $N > \frac{h \cdot g}{e_0} = \frac{24 \cdot 138}{0.1} = 33120$, де g – кількість вхідних нейронів ШНМ ($g=138$); h – кількість нейронів прихованих шарів ШНМ ($h=24$), e_0 – допустима похибка навчання ($e_0=0,1$). Отже, 35180 векторів навчальної вибірки достатньо для того, щоб навчити ШНМ розпізнавати можливі ситуації з заданою точністю.

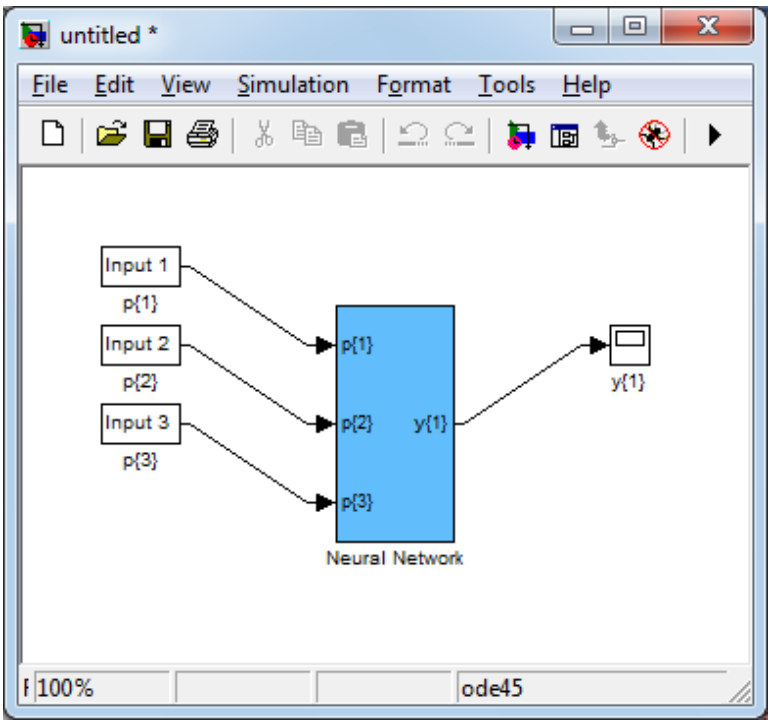


Рис. 2.5 – Архітектура ШНМ у Simulink

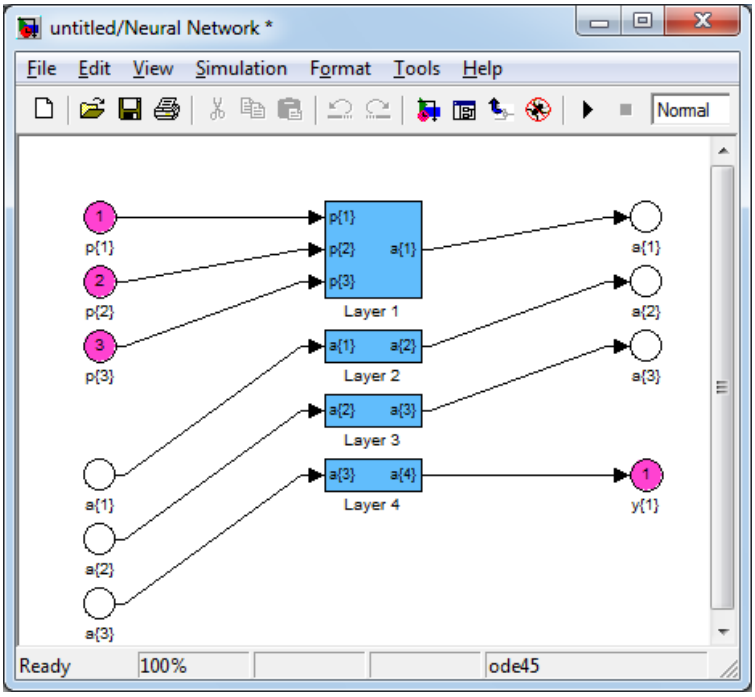


Рис. 2.6 – Структурна схема шарів ШНМ у Simulink

Тестування ШНМ здійснювалось з використанням тестової вибірки з 3518 векторів. Процес навчання і тестування ШНМ відображено на рис. 2.7. На цьому рисунку нижня крива (синього кольору) відображає графік навчання, а верхня крива (зеленого кольору) відображає графік тестування ШНМ.

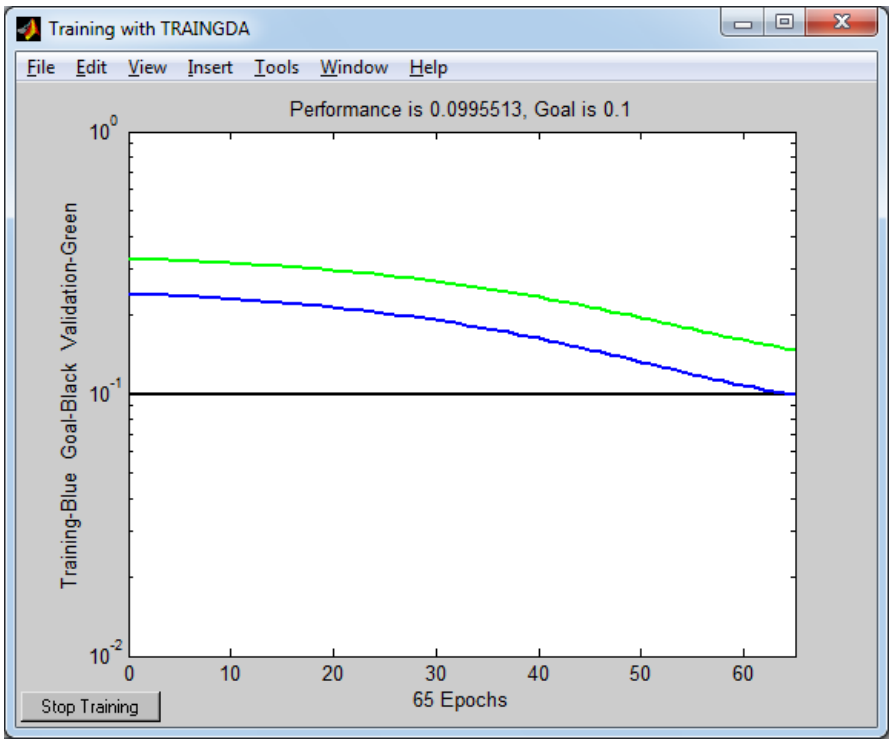


Рис. 2.7 – Процес навчання і тестування ШНМ

Як видно з рисунку 2.7, ШНМ коректно навчилась, оскільки було досягнуто мету навчання ШНМ (пряма чорного кольору), яка становила досягнення точності навчання на рівні 0.1 (про що свідчить пряма чорного кольору і напис на рисунку вище графіків), а при навчанні отримано результат 0.0995513 (про що свідчить крива синього кольору на рисунку і напис на рисунку вище графіків). Крива тестування ШНМ (крива зеленого кольору) повторює форму кривої навчання (крива синього кольору), що свідчить про коректно підібрані дані для тестування ШНМ.

Отже, аналіз графіків навчання і тестування ШНМ дозволив зробити висновок, що мережа навчилась з високою точністю.

2.3.2. Експериментальне дослідження методу прогнозування рівня якості програмного забезпечення комп’ютерних систем на основі атрибутів якості

Розглянемо 4 специфікації, які виконувались різними ІТ-фірмами м. Хмельницького для виконання одного й того ж замовлення (розв’язання однієї й тієї ж задачі). Наведемо у Додатку Д фрагмент однієї з розглянутих специфікацій з

накладанням певних обмежень шляхом структурування до тих вимог, які містять атрибути якості.

Спочатку з кожної специфікації було вибрано значення 138 атрибутів якості ПЗ (для фрагменту специфікації, наведеного у Додатку Д, було обрано такі значення атрибутів: …, 100, 0.1, 7, 10, 1, 12, 2, 1, 0.5, 1, 2, 15, 3, 10000, …), після чого були сформовані вхідні вектори ШНМ, які подавались на нейрони вхідного шару ШНМ.

Далі, згідно із першим кроком методу прогнозування рівня якості ПЗКС на основі атрибутів якості, штучна нейронна мережа надає прогнозовані оцінки восьми характеристик якості ПЗ для кожної із 4-х аналізованих специфікацій:

$QCH_I = \{0,9;0,87;0,78;0,95;0,89;0,93;0.91;0.88\}$, тобто для специфікації вимог №1: $qch_1 = 0,9; qch_2 = 0,87; qch_3 = 0,78; qch_4 = 0,95; qch_5 = 0,89; qch_6 = 0,93;$ $qch_7 = 0,91;\ qch_8 = 0,88$;

$QCH_{II} = \{0,3;0,27;0,28;0,25;0,23;0,33;0.28;0.32\}$;

$QCH_{III} = \{0,6;0,7;0,8;0,15;0,69;0,53;0.61;0.58\}$;

$QCH_{IV} = \{0,5;0,57;0,53;0,47;0,52;0,49;0.54;0.48\}$.

Згідно із другим кроком методу прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості побудуємо геометричну інтерпретацію значень характеристик якості ПЗ – рис. 2.8-2.11:

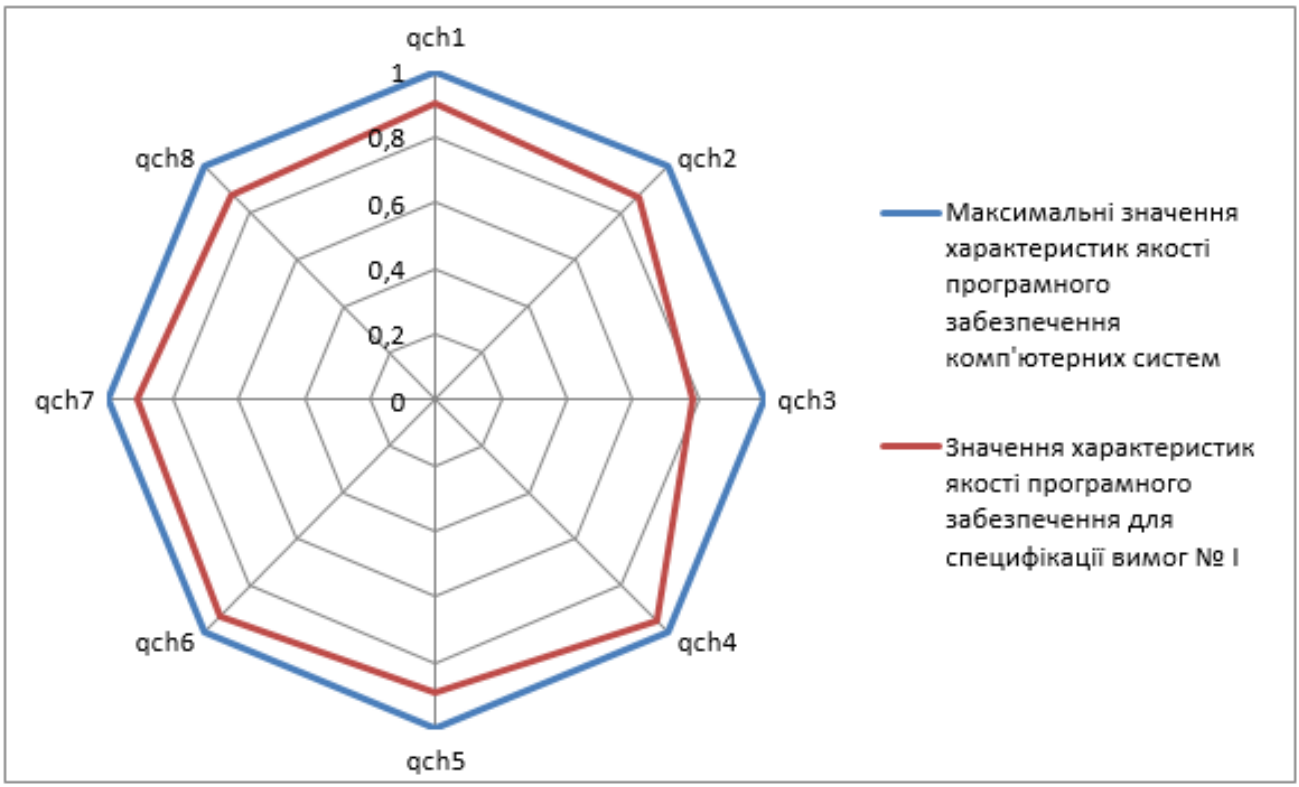


Рис. 2.8 – Геометрична інтерпретація значень характеристик якості ПЗКС для специфікації вимог № I

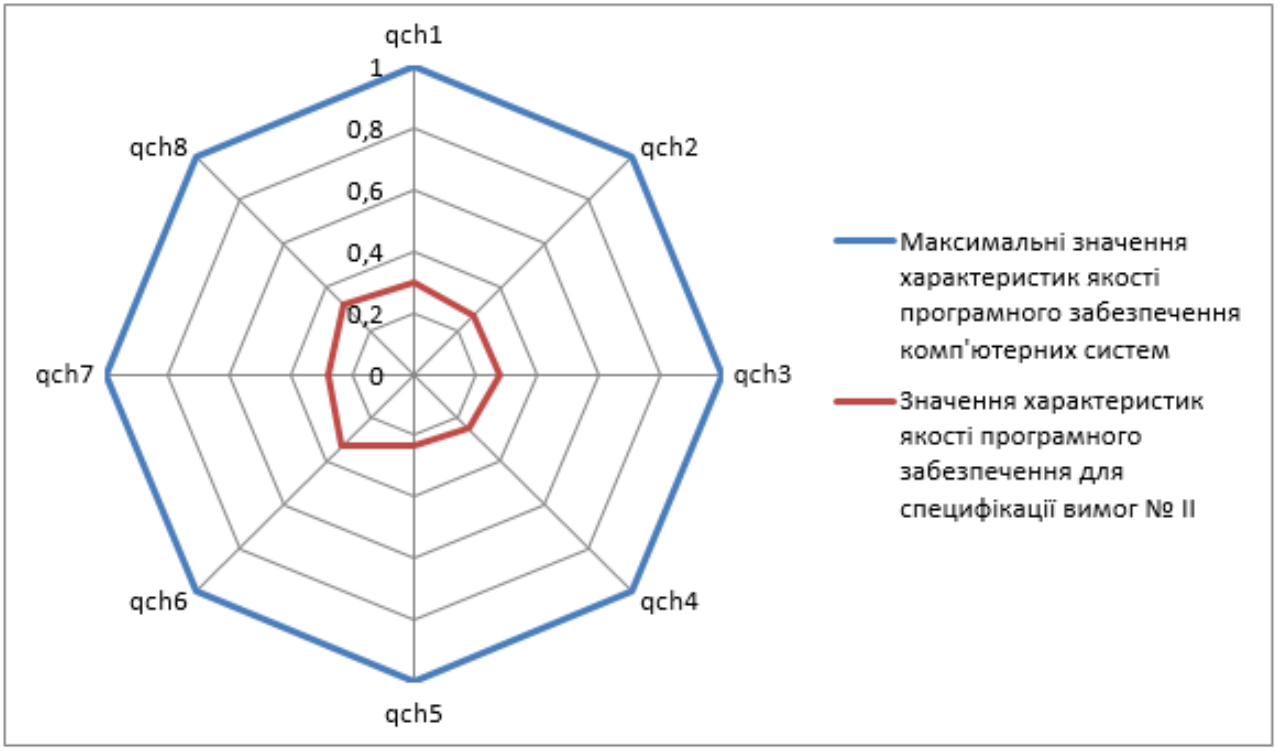


Рис. 2.9 – Геометрична інтерпретація значень характеристик якості ПЗКС для специфікації вимог № II

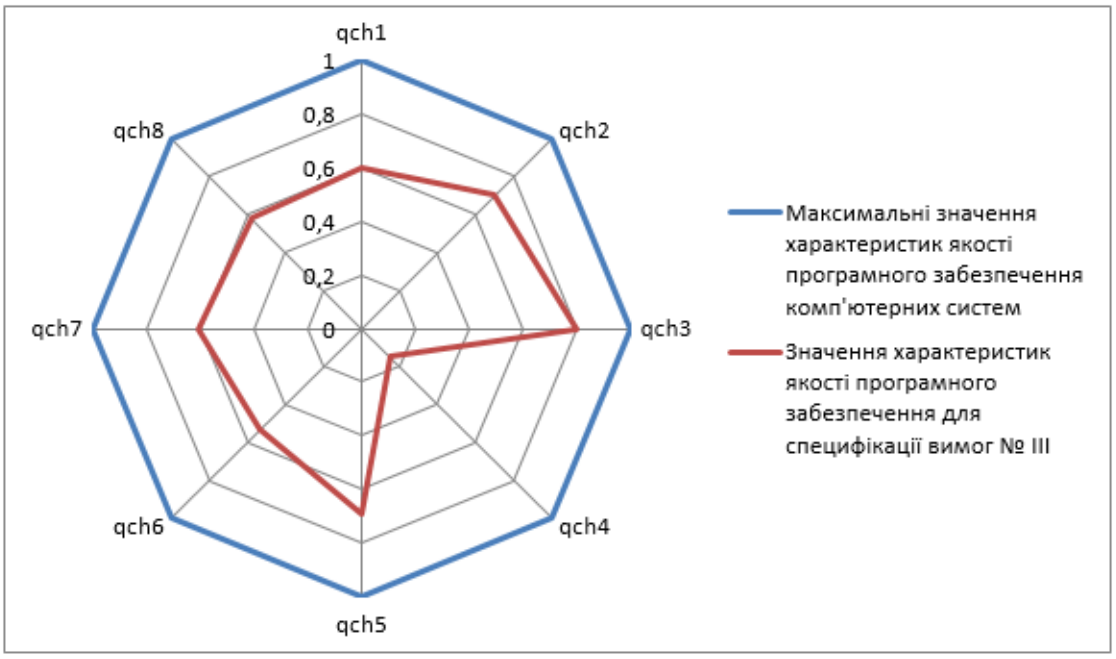


Рис. 2.10 – Геометрична інтерпретація значень характеристик якості ПЗКС для специфікації вимог № III

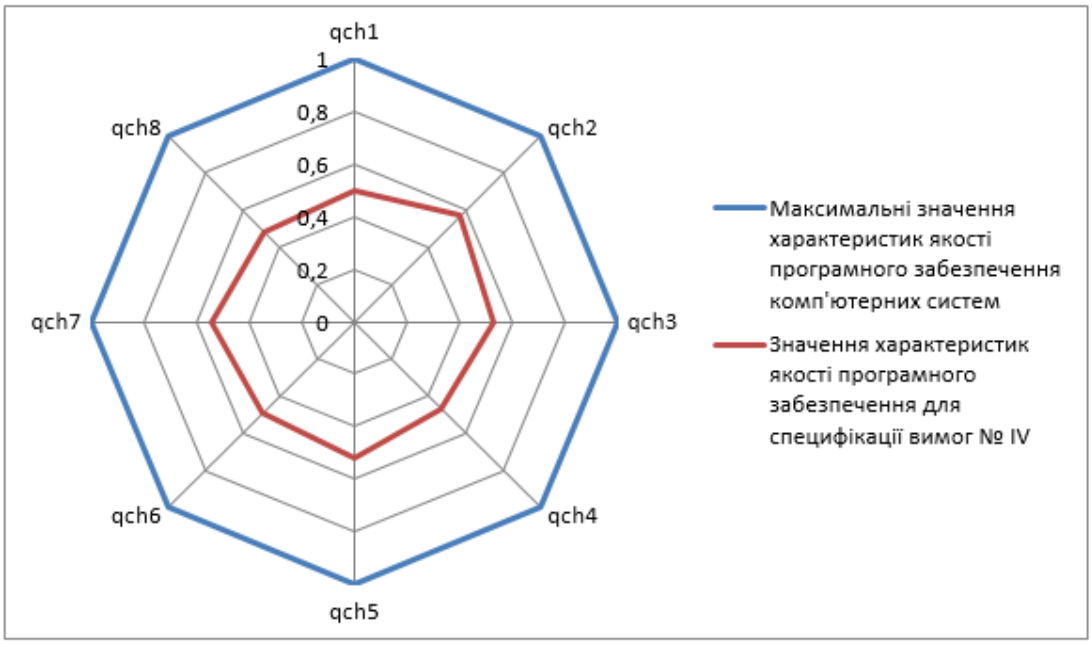


Рис. 2.11 – Геометрична інтерпретація значень характеристик якості ПЗКС для специфікації вимог № IV

Згідно із третім кроком методу прогнозування рівня якості ПЗКС на основі атрибутів якості перевріимо припустимість компенсації характеристик якості ПЗ. Розглянемо восьмикутники, окреслені червоними лініями на рис. 2.8-2.11. У восьмикутника, зображеного червоною лінією на рис. 2.8, всі внутрішні кути не

перевищують 180°, отже, цей восьмикутник є опуклим, тобто компенсація характеристик якості ПЗКС для специфікації вимог № I є припустимою. У восьмикутника, зображеного червоною лінією на рис. 2.9, також всі внутрішні кути не перевищують 180°, отже, цей восьмикутник також є опуклим, тобто компенсація характеристик якості ПЗ для специфікації вимог № II також є припустимою. У восьмикутника, зображеного червоною лінією на рис. 2.10, один з внутрішніх кутів (піввісь qch_4) перевищує 180°, отже, цей восьмикутник не є опуклим, тобто компенсація характеристик якості ПЗ для специфікації № III є неприпустимою, відтак комплексний показник прогнозованої якості ПЗ для специфікації вимог № III не розраховується. У восьмикутника, зображеного червоною лінією на рис. 2.11, всі внутрішні кути не перевищують 180°, отже, цей восьмикутник є опуклим, тобто компенсація характеристик якості ПЗ для специфікації вимог № IV є припустимою.

Далі, згідно із четвертим кроком методу прогнозування рівня якості ПЗКС на основі атрибутів якості розрахуємо комплексний показник прогнозованої якості ПЗ для специфікацій вимог № I, № II, № IV.

$$cifsq_I = 0{,}35355 \cdot (qch_1 \cdot qch_2 + qch_2 \cdot qch_3 + qch_3 \cdot qch_4 + qch_4 \cdot qch_5 + qch_5 \cdot qch_6 + \\ + qch_6 \cdot qch_7 + qch_7 \cdot qch_8 + qch_8 \cdot qch_1) = 0{,}35355 \cdot (0{,}9 \cdot 0{,}87 + 0{,}87 \cdot 0{,}78 + 0{,}78 \cdot 0{,}95 + \\ + 0{,}95 \cdot 0{,}89 + 0{,}89 \cdot 0{,}93 + 0{,}93 \cdot 0{,}91 + 0{,}91 \cdot 0{,}88 + 0{,}88 \cdot 0{,}9) = 2{,}2326;$$

$$cifsq_{II} = 0{,}35355 \cdot (qch_1 \cdot qch_2 + qch_2 \cdot qch_3 + qch_3 \cdot qch_4 + qch_4 \cdot qch_5 + qch_5 \cdot qch_6 + \\ + qch_6 \cdot qch_7 + qch_7 \cdot qch_8 + qch_8 \cdot qch_1) = 0{,}35355 \cdot (0{,}3 \cdot 0{,}27 + 0{,}27 \cdot 0{,}28 + 0{,}28 \cdot 0{,}25 + \\ + 0{,}25 \cdot 0{,}23 + 0{,}23 \cdot 0{,}33 + 0{,}33 \cdot 0{,}28 + 0{,}28 \cdot 0{,}32 + 0{,}32 \cdot 0{,}3) = 0{,}2256;$$

$$cifsq_{IV} = 0{,}35355 \cdot (qch_1 \cdot qch_2 + qch_2 \cdot qch_3 + qch_3 \cdot qch_4 + qch_4 \cdot qch_5 + qch_5 \cdot qch_6 + \\ + qch_6 \cdot qch_7 + qch_7 \cdot qch_8 + qch_8 \cdot qch_1) = 0{,}35355 \cdot (0{,}5 \cdot 0{,}57 + 0{,}57 \cdot 0{,}53 + 0{,}53 \cdot 0{,}47 + \\ + 0{,}47 \cdot 0{,}52 + 0{,}52 \cdot 0{,}49 + 0{,}49 \cdot 0{,}54 + 0{,}54 \cdot 0{,}48 + 0{,}48 \cdot 0{,}5) = 0{,}7422.$$

Згідно із п’ятим кроком методу прогнозування рівня якості ПЗКС на основі атрибутів якості, спрогнозуємо рівень якості майбутнього ПЗ, розроблюваного за кожною з аналізованих специфікацій. Отже, оскільки $cifsq_I = 2{,}2326$, тобто значення комплексного показника прогнозованої якості ПЗ для специфікації вимог

№ I лежить в діапазоні [2,0435; 2,8284], то майбутнє ПЗ, розроблюване за специфікацією вимог № I, прогнозовано матиме високий рівень якості.

Оскільки $cifsq_{II} = 0{,}2256$, тобто значення комплексного показника прогнозованої якості ПЗ для специфікації вимог № II лежить в діапазоні [0; 0,2545), то майбутнє ПЗ, що розробляється за специфікацією вимог № II, прогнозовано матиме низький рівень якості.

Оскільки $cifsq_{IV} = 0{,}7422$, тобто значення комплексного показника прогнозованої якості ПЗ для специфікації вимог № IV лежить в діапазоні [0,2545; 2,0435), то майбутнє ПЗ, що розробляється за специфікацією вимог № IV, прогнозовано матиме середній рівень якості.

Оскільки компенсація характеристик якості ПЗ для специфікації вимог № III є неприпустимою, то комплексний показник прогнозованої якості ПЗ для специфікації вимог № III не розраховувався, проте таку специфікацію вимог небажано реалізовувати через низьке значення однієї з характеристик якості в порівнянні зі значеннями інших характеристик якості, кожна з яких є рівноважливою для майбутнього ПЗ.

Враховуючи отриманий прогнозований рівень якості майбутнього ПЗ, розроблюваного за кожною з аналізованих специфікацій, порівняємо 4 аналізовані специфікації вимог до ПЗ, які виконувались різними ІТ-фірмами м. Хмельницького (Україна) для виконання одного й того ж замовлення, а також виконаємо обґрунтований вибір специфікації для подальшої реалізації ПЗ саме високої якості. Отже, оскільки майбутнє ПЗ, розроблюване за специфікацією вимог № I, прогнозовано матиме високий рівень якості, то для подальшої реалізації ПЗКС високої якості рекомендується обрати саме специфікацію вимог № I.

На рис. 2.12 представлено порівняльний аналіз пропонованого методу та поточного стану справ.

Рис. 2.12 ілюструє поточний стан галузі оцінювання якості ПЗКС, а саме: діючі поточні стандарти оцінки якості програмного забезпечення рекомендують ідентифікувати 138 атрибутів якості програмного забезпечення у специфікації вимог для прогнозування якості або підрахувати їх для оцінки якості готового

програмного забезпечення. На основі цих атрибутів можна оцінити 8 характеристик якості, але відсутні конкретні формули, що визначають залежність між значеннями характеристик якості та значеннями атрибутів. Іншими словами, у стандартах вказано лише, від яких атрибутів залежить кожна характеристика, проте сам характер цих залежностей залишається невідомим. Відтак, можливість знайти кількісні значення характеристик якості на основі атрибутів наразі відсутня. Аналогічно чинні стандарти із оцінювання якості ПЗ пропонують на основі оцінок 8 характеристик якості ПЗ оцінити, власне, якість ПЗ, при цьому знову-таки формули, які відображають залежність якості від характеристик, відсутні, а наявний лише факт, що якість ПЗ залежить від 8 характеристик якості в комплексі, проте характер такої залежності також невідомий. Відтак, можливість знайти кількісне значення якості на основі її 8 характеристик наразі знов-таки відсутня. Оскільки немає кількісного значення якості ПЗ, то відповідно оцінити або прогнозувати рівень якості ПЗ дуже важко, це відбувається, в основному, з врахуванням досвіду та інтуїції розробників, які дають лінгвістичну оцінку, не підкріплену жодними кількісними параметрами.

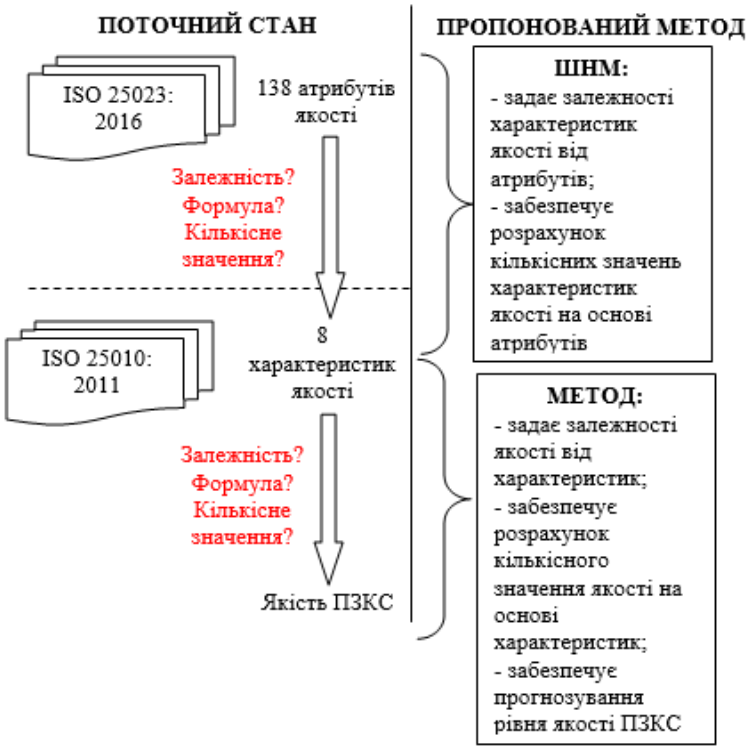


Рис. 2.12 – Порівняльний аналіз пропонованого методу та поточного стану справ

На відміну від поточного стану галузі оцінювання якості ПЗКС, пропонований метод дозволяє вирішити зазначені проблеми. Так, використовувана ШНМ задає залежності характеристик якості від атрибутів, а також забезпечує розрахунок кількісних значень характеристик якості на основі атрибутів, як видно з концепції запропонованого підходу (рис. 2.1), а розроблений метод, проілюстрований рис. 2.4, задає залежності якості від характеристик, забезпечує розрахунок кількісного значення якості на основі характеристик, а також забезпечує прогнозування рівня якості ПЗКС на основі отриманого кількісного значення.

2.4. Висновки

Метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп'ютерних систем забезпечує вибір значень атрибутів якості ПЗ з природомовної специфікації вимог до ПЗ, які далі можуть бути використані для визначення значень характеристик якості ПЗ та для комплексного оцінювання якості ПЗ. Розроблений метод є важливим для автоматизації опрацювання вимог та усунення людини з процесів опрацювання інформації. Метод пошуку значень атрибутів якості у вимогах до ПЗКС є теоретичним підгрунтям для розроблення модулю автоматичного аналізу вимог до ПЗ майбутньої системи для прогнозування рівня якості ПЗКС на основі атрибутів якості.

В основі концептуальної моделі процесу прогнозування характеристик якості ПЗКС лежить ШНМ, що опрацьовує значення 138 атрибутів якості, які визначаються у вимогах до ПЗ, проводить їх апроксимацію та надає прогнозовані кількісні оцінки восьми характеристик якості ПЗ (ефективність, функційна придатність, надійність, зручність використання, безпека, сумісність, можливість переносу, супроводжуваність). При цьому вона враховує не лише кількісне значення кожного атрибуту якості, але й взаємний вплив атрибутів під час кількісного оцінювання характеристик якості ПЗКС.

Запропоновано метод прогнозування рівня якості ПЗКС на основі атрибутів якості, який, на відміну від відомих, забезпечує прогнозування рівня якості розроблюваного ПЗКС на основі опрацювання атрибутів якості ПЗ, доступних у вимогах. Таким чином, запропонований метод дозволяє порівнювати специфікації вимог до ПЗ, одразу відмовлятись від реалізації ПЗКС на основі невдалих специфікацій (економія коштів та часу, зменшення ймовірності провальних і проблемних проєктів) та виконувати обґрунтований вибір специфікації для подальшої реалізації ПЗКС саме високої якості (звісно, за умови, що помилки не будуть внесені на наступних етапах життєвого циклу ПЗ).

Була розроблена, навчена і протестована штучна нейронна мережа для прогнозування характеристик якості ПЗКС на основі атрибутів якості. Аналіз графіків навчання і тестування мережі свідчить про високу точність її навчання.

Під час експериментального дослідження методу прогнозування рівня якості ПЗКС на основі атрибутів якості, на основі прогнозованого рівня якості майбутнього програмного забезпечення було проведено порівняння чотирьох аналізованих специфікацій вимог до програмного забезпечення, що були розроблені різними ІТ-фірмами міста Хмельницького, Україна, для виконання одного й того ж замовлення, а також виконано обґрунтований вибір специфікації для подальшої реалізації ПЗ саме високої якості. Оскільки майбутнє ПЗ, що розробляється за специфікацією вимог № І, прогнозовано матиме високий рівень якості, то для подальшої реалізації ПЗ високої якості рекомендується обрати саме специфікацію вимог № І.

Так, використовувана ШНМ задає залежності характеристик якості від атрибутів, а також забезпечує розрахунок кількісних значень характеристик якості на основі атрибутів, як видно з концепції запропонованого підходу, а розроблений метод задає залежності якості від характеристик, забезпечує розрахунок кількісного значення якості на основі характеристик, а також забезпечує прогнозування рівня якості ПЗКС на основі отриманого кількісного значення.

РОЗДІЛ 3.
ПРОГНОЗУВАННЯ ТА ОЦІНЮВАННЯ БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

3.1. Метод прогнозування рівня безпеки програмного забезпечення комп'ютерних систем [67]

Автор дослідження [90] представила онтологію безпеки ПЗКС як характеристики якості ПЗ, базуючись на стандартах ISO 25010, ISO 25023 – рис. 3.1.

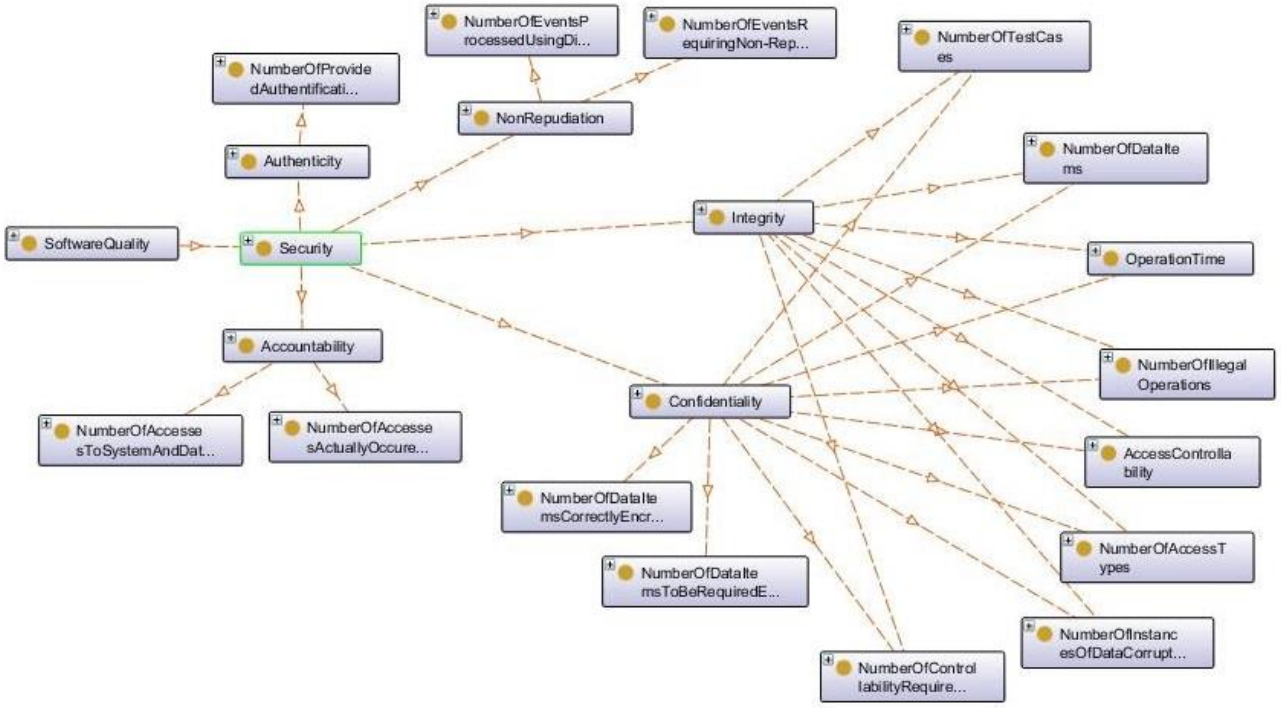


Рис. 3.1 – Онтологія безпеки ПЗКС як характеристики якості ПЗ на базі стандартів ISO 25010, ISO 25023 [90]

З рис. 3.1 видно, що безпека ПЗКС (як характеристика якості ПЗ) залежить від 5 підхарактеристик (конфіденційність (Confidentiality), цілісність (Integrity), автентичність (Authenticity), відповідальність (Accountability), безвідмовність (Non-Repudiation)), кожна з яких залежить від певних атрибутів якості. Так, згідно із

стандартами ISO 25010 [3] та ISO 25023 [89] безпека ПЗКС (як характеристика якості ПЗ) залежить від 15 різних атрибутів якості.

Для прогнозування рівня безпеки ПЗКС на основі атрибутів якості з вимог до ПЗ спочатку розрахуємо прогнозоване кількісне значення безпеки ПЗ на основі значень 15 визначених атрибутів, наведених на рис. 3.1, з врахуванням їх взаємозалежностей, що є важкоформалізованою задачею. Для встановлення та врахування взаємозалежностей між значеннями атрибутів якості з вимог до ПЗ і значенням безпеки ПЗКС (як характеристики якості ПЗ) використаємо штучну нейронну мережу типу «багатошаровий персептрон», яка на входи отримає значення 15 атрибутів якості, від яких залежить безпека ПЗ, та після їх апроксимації визначить прогнозоване кількісне значення безпеки ПЗ в інтервалі [0;1]. Концептуальна модель процесу прогнозування безпеки ПЗКС (як характеристики якості ПЗ) на основі атрибутів якості подана на рис. 3.2. Така концептуальна модель базується на розробленій у розділі 2 концептуальній моделі процесу прогнозування характеристик якості програмного забезпечення комп'ютерних систем на основі атрибутів якості.

Рис. 3.2 – Концептуальна модель процесу прогнозування безпеки програмного забезпечення комп'ютерних систем (як характеристики якості ПЗ) на основі атрибутів якості

ШНМ навчена так, що на основі опрацювання значень відповідних 15 атрибутів генерує прогнозоване кількісне значення безпеки ПЗКС *pnvss* в інтервалі

[0;1], де значення «0» означає найгірший рівень безпеки ПЗКС, а значення «1» означає найкращий рівень безпеки ПЗКС як характеристики якості ПЗ. Проте, враховуючи відсутність еталонних значень, і замовнику, і навіть розробнику складно правильно інтерпретувати та витлумачити отримане прогнозоване кількісне значення безпеки ПЗКС, відтак складно правильно оцінити рівень безпеки ПЗКС за отриманим з ШНМ значенням.

Тому, для спрощення інтерпретації та однозначності тлумачення прогнозованого кількісного значення безпеки ПЗКС, спочатку необхідно визначити порогові значення, за якими буде генеруватись висновок про рівень безпеки ПЗКС (як характеристики якості).

Для встановлення таких порогових значень було проведено аналіз 230 наявних специфікацій вимог до ПЗ на предмет пошуку значень атрибутів якості, від яких залежить безпека ПЗКС, для яких ШНМ визначила прогнозоване кількісне значення безпеки ПЗКС, а також 230 відповідних готових програм, написаних за цими вимогами, для яких при сертифікації визначено рівень безпеки (один з чотирьох – початковий, середній, достатній, високий). Специфікації вимог до ПЗ та готові програми для аналізу надавались софтверними компаніями м. Хмельницького (Україна) в рамках співпраці із кафедрою комп'ютерної інженерії та інформаційних систем Хмельницького національного університету.

В результаті проведеного аналізу сформуємо порогові значення прогнозованого кількісного значення безпеки ПЗКС *pnvss* для прогнозування рівня безпеки ПЗКС (як характеристики якості ПЗ):

1) початковий рівень безпеки – pnvss $\in$ [0; 0,22);

2) середній рівень безпеки – pnvss $\in$ [0,22; 0,49);

3) достатній рівень безпеки – pnvss $\in$ [0,49; 0,89);

4) високий рівень безпеки – pnvss $\in$ [0,89; 1].

Враховуючи розроблену концептуальну модель процесу прогнозування безпеки ПЗКС (як характеристики якості ПЗ) на основі атрибутів якості та визначені порогові значення прогнозованого кількісного значення безпеки ПЗКС *pnvss*, *метод прогнозування рівня безпеки ПЗКС* складається з таких кроків:

вхідна інформація: значення 15 атрибутів якості, від яких залежить безпека ПЗКС (як характеристика якості);

1) препроцесінг вимог до ПЗ – перетворення специфікації вимог у вигляд, придатний для аналізу на предмет виявлення значень атрибутів якості, відповідно до структури специфікації вимог до ПЗ, розробленої у розділі 2;

2) аналіз вимог до програмного забезпечення з метою пошуку значень 15 атрибутів якості, які впливають на безпеку ПЗКС, відповідно до методу пошуку значень атрибутів якості у вимогах до програмного забезпечення комп'ютерних систем, розробленого у розділі 2;

3) підготовка знайдених значень 15 атрибутів якості, від яких залежить безпека ПЗ, для подачі їх на вхід ШНМ – на цьому етапі відбувається підготовка вхідних векторів ШНМ з врахуванням факту, що підхарактеристики безпеки залежать від 23 атрибутів, в т.ч. від 15 різних атрибутів (рис. 3.1), а входи ШНМ сформовані як 5 множин (для 5 підхарактеристик безпеки ПЗ) з 1, 2, 8, 10 та 2 атрибутів згідно із онтологією, представленою на рис. 3.1;

4) опрацювання значень атрибутів штучною нейронною мережею;

5) аналіз результату ШНМ – прогнозованого кількісного значення безпеки ПЗКС *pnvss*;

6) формування висновку про прогнозований рівень безпеки програмного забезпечення комп'ютерних систем на основі наступних правил:

– якщо $pnvss \in [0; 0{,}22)$, то програмне забезпечення КС прогнозовано матиме початковий рівень безпеки;

– якщо $pnvss \in [0{,}22; 0{,}49)$, то програмне забезпечення КС прогнозовано матиме середній рівень безпеки;

– якщо $pnvss \in [0{,}49; 0{,}89)$, то програмне забезпечення КС прогнозовано матиме достатній рівень безпеки;

– якщо $pnvss \in [0{,}89; 1]$, то програмне забезпечення КС прогнозовано матиме високий рівень безпеки;

вихідна інформація: прогнозований рівень безпеки ПЗКС.

3.1.1. Реалізація, навчання та тестування штучної нейронної мережі для прогнозування безпеки програмного забезпечення комп'ютерних систем на основі атрибутів якості

Штучна нейронна мережа для прогнозування безпеки програмного забезпечення комп'ютерних систем на основі атрибутів якості реалізована у пакеті Matlab. Оператор `gensim(net)` згенерував візуалізацію розробленої ШНМ в пакеті Simulink (рис. 3.3 – 3.7).

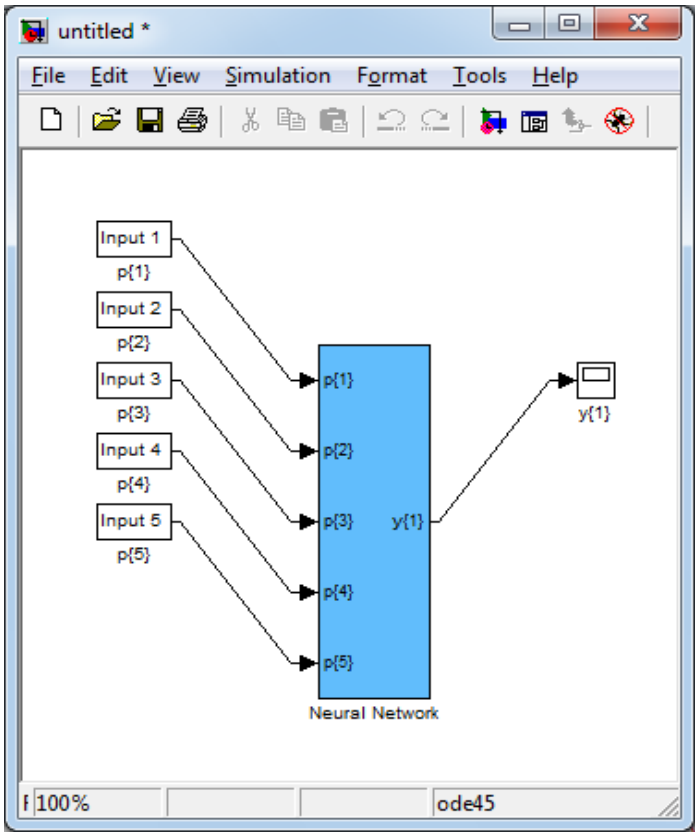


Рис. 3.3 – Архітектура ШНМ

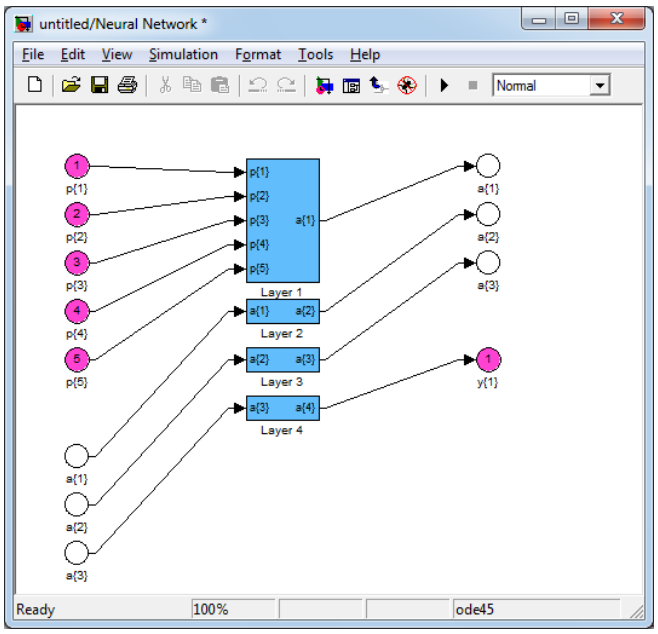


Рис. 3.4 – Структура шарів ШНМ

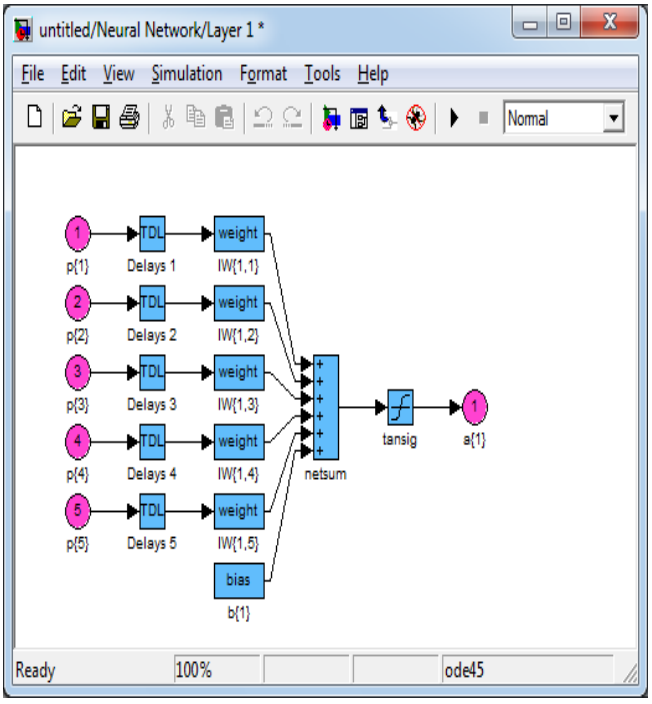


Рис. 3.5 – Структура першого (вхідного) шару ШНМ

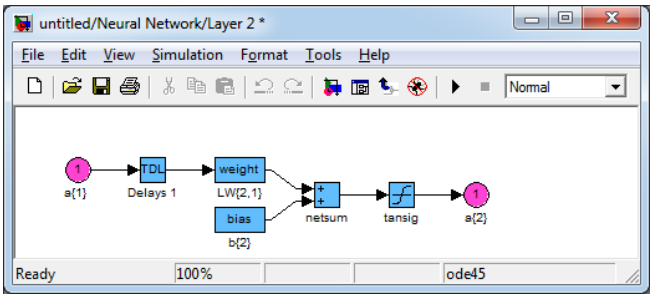


Рис. 3.6 – Структура другого шару ШНМ (третій шар ШНМ є аналогічним до другого шару)

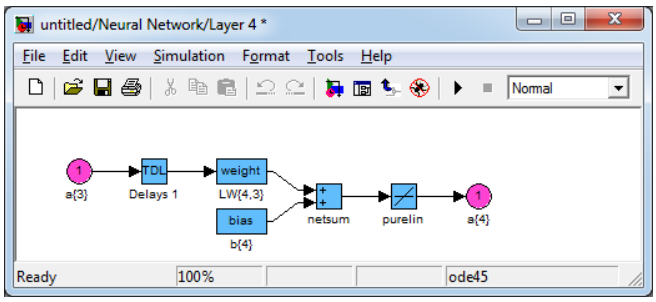


Рис. 3.7 – Структура четвертого (результуючого) шару ШНМ

Для навчання і тестування одержаної ШНМ сформовано навчальну вибірку з 4750 векторів та тестувальну вибірку з 867 векторів на основі аналізу наявних вимог до ПЗ та відповідних їм готових програм з відомим рівнем безпеки, наданих софтверними компаніями м. Хмельницького (Україна). Для розрахунку необхідного об'єму навчальної вибірки для ШНМ, яку потрібно навчити з похибкою порядку 0.1 використаємо формулу: $N > \frac{h \cdot g}{e_0} = \frac{20 \cdot 23}{0{,}1} = 4600$, де g – кількість вхідних нейронів ШНМ ($g=23$); h – кількість нейронів прихованих шарів ШНМ ($h=20$), e_0 – допустима похибка навчання ($e_0=0.1$). Отже, 4750 векторів навчальної вибірки достатньо для того, щоб навчити ШНМ розпізнавати можливі ситуації з заданою точністю. Процес навчання і тестування ШНМ відображено на рис. 3.8, 3.9, де синя крива – це графік навчання, зелена крива – графік тестування ШНМ, чорна пряма – мета навчання, яка, як видно з рисунків була досягнута.

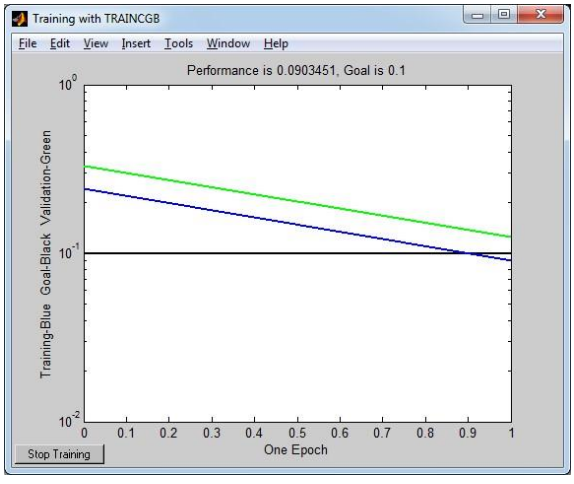


Рис. 3.8 – Навчання і тестування ШНМ за алгоритмом `traincgb` з критерієм якості `msereg`

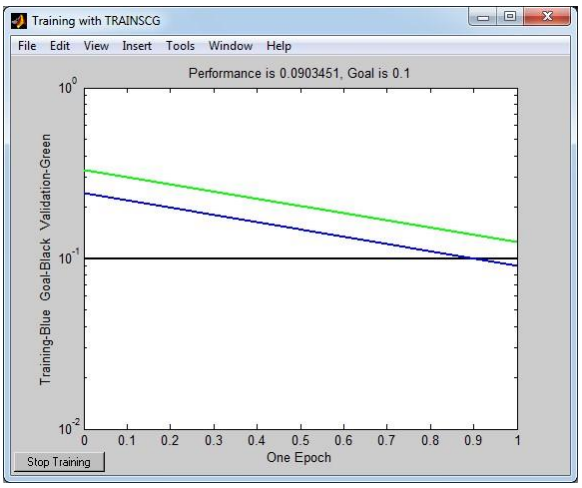


Рис. 3.9 – Навчання і тестування ШНМ за алгоритмом `trainscg` з критерієм якості `msereg`

Щоб вибрати найкращий алгоритм навчання ШНМ, проведемо аналіз процесу навчання ШНМ з використанням різних алгоритмів та різних критеріїв якості. Результати цього аналізу наведені у таблиці 3.1.

Таблиця 3.1

Аналіз процесу навчання ШНМ для прогнозування безпеки програмного забезпечення комп'ютерних систем на основі атрибутів якості

Алгоритм навчання ШНМ	**Критерій якості навчання ШНМ**	**Похибка навчання ШНМ**
`trainbfg`	`mse`	0.100291
`trainoss`		0.100291
`traincgb`		0.100291
`traingda`		0.100291
`trainlm`		0.100291
`trainrp`		0.100291
`trainscg`		0.100291
`trainbfg`	`msereg`	0.0964035
`trainoss`		0.0964035
`traincgb`		0.0903451

traingda		0.0995513
trainlm		0.0903451
trainrp		0.0952321
trainscg		0.0903451
trainbfg	mae	0.396053
trainoss		0.396053
traincgb		0.251652
traingda		0.262794
trainlm		0.264611
trainrp		0.251225
trainscg		0.395961

Проведений аналіз показав, що найгірший результат навчання ШНМ дає критерій якості навчання mae в комбінації з усіма алгоритмами навчання, а найкращий результат дає критерій якості навчання msereg. Аналіз результатів навчання з використанням критерію якості навчання msereg дає можливість визначити, що найбільш точний результат дають алгоритми навчання traincgb, trainlm, trainscg. Аналіз результатів тестування довів, що мережа навчилась із заданою точністю.

3.1.2. Експериментальне дослідження методу прогнозування рівня безпеки програмного забезпечення комп'ютерних систем

Розглянемо 10 специфікацій вимог до ПЗ, підготовлених десятьма різними софтверними компаніями м. Хмельницького (Україна) як результат етапу збору та аналізу вимог до одного й того ж ПЗ. Кожна із специфікацій пройшла етап препроцесінгу, під час якого була підготовлена для аналізу на предмет пошуку значень атрибутів якості. Далі кожна специфікація була проаналізована на предмет пошуку значень 15 атрибутів якості, від яких залежить безпека ПЗКС. Після цього було здійснено формування вхідних векторв ШНМ.

Наведемо у Додатку Д фрагмент однієї з розглянутих специфікацій з накладанням певних обмежень шляхом структурування до тих вимог, які містять атрибути якості, від яких залежить безпека ПЗКС. Для фрагменту специфікації, наведеного у Додатку Д, було обрано такі значення атрибутів: …, 1, 5, 234, 89, 128, 154, 3, … .

ШНМ на основі опрацювання знаень атрибутів надала такі значення прогнозованого кількісного значення безпеки ПЗКС *pnvss* для 10 аналізованих специфікацій (рис. 3.10): $pnvss_1 = 0.12$; $pnvss_2 = 0.93$; $pnvss_3 = 0.45$; $pnvss_4 = 0.67$; $pnvss_5 = 0.34$; $pnvss_6 = 0.09$; $pnvss_7 = 0.78$; $pnvss_8 = 0.98$; $pnvss_9 = 0.41$; $pnvss_{10} = 0.53$.

Далі було виконано аналіз отриманих прогнозованих кількісних значень безпеки ПЗ $pnvss_1$-$pnvss_{10}$, в результаті якого було сформовано висновки про прогнозований рівень безпеки програмного забезпечення комп'ютерних систем для 10 аналізованих специфікацій:

– оскільки $pnvss_1 \in [0; 0{,}22)$, то програмне забезпечення КС, яке буде розроблене за специфікацією вимог №1, прогнозовано матиме початковий рівень безпеки;

– оскільки $pnvss_2 \in [0{,}89; 1]$, то програмне забезпечення КС, яке буде розроблене за специфікацією вимог №2, прогнозовано матиме високий рівень безпеки;

– оскільки $pnvss_3 \in [0{,}22; 0{,}49)$, то програмне забезпечення КС, яке буде розроблене за специфікацією вимог №3, прогнозовано матиме середній рівень безпеки;

– оскільки $pnvss_4 \in [0{,}49; 0{,}89)$, то програмне забезпечення КС, яке буде розроблене за специфікацією вимог №4, прогнозовано матиме достатній рівень безпеки;

– оскільки $pnvss_5 \in [0{,}22; 0{,}49)$, то програмне забезпечення КС, яке буде розроблене за специфікацією вимог №5, прогнозовано матиме середній рівень безпеки;

– оскільки $pnvss_6 \in [0; 0{,}22)$, то ПЗКС, яке буде розроблене за специфікацією вимог №6, прогнозовано матиме початковий рівень безпеки;

– оскільки $pnvss_7 \in [0{,}49;\ 0{,}89)$, то програмне забезпечення КС, яке буде розроблене за специфікацією вимог №7, прогнозовано матиме достатній рівень безпеки;

– оскільки $pnvss_8 \in [0{,}89;\ 1]$, то програмне забезпечення КС, яке буде розроблене за специфікацією вимог №8, прогнозовано матиме високий рівень безпеки;

– оскільки $pnvss_9 \in [0{,}22;\ 0{,}49)$, то програмне забезпечення КС, яке буде розроблене за специфікацією вимог №9, прогнозовано матиме середній рівень безпеки;

– оскільки $pnvss_{10} \in [0{,}49;\ 0{,}89)$, то програмне забезпечення КС, яке буде розроблене за специфікацією вимог №10, прогнозовано матиме достатній рівень безпеки.

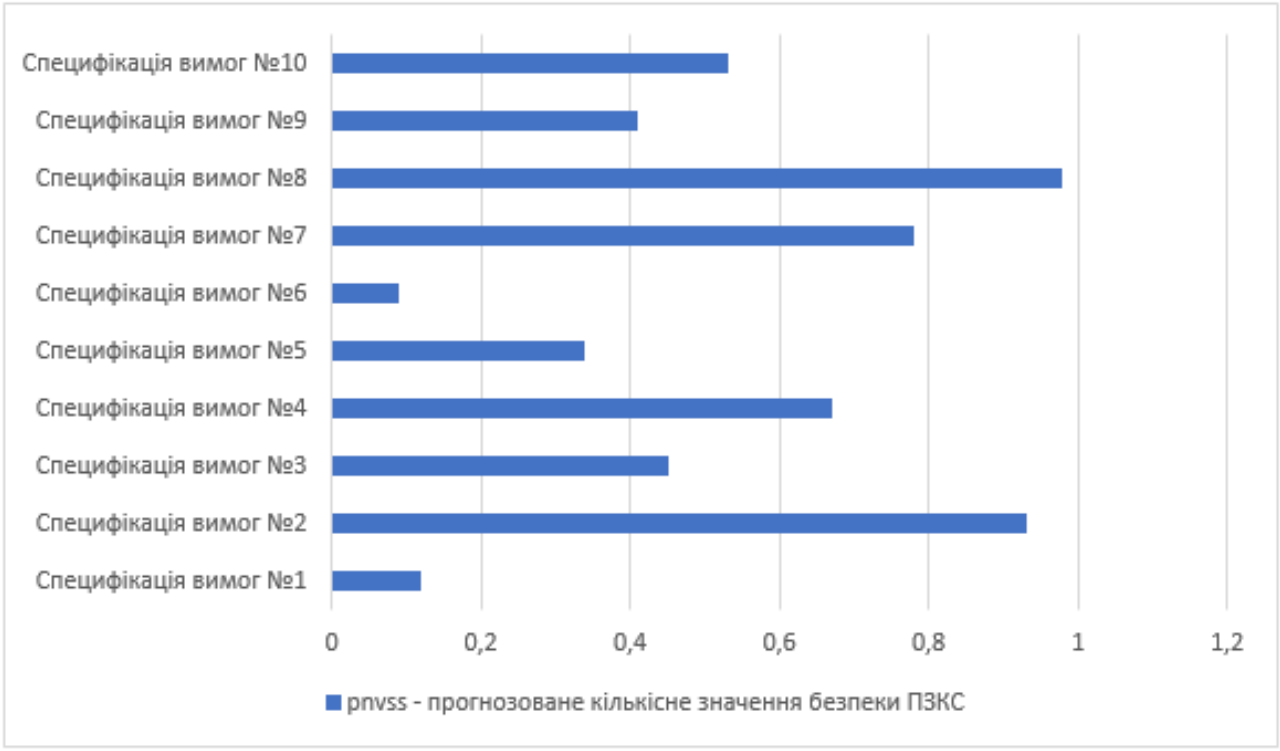


Рис. 3.10 – Значення прогнозованого кількісного значення безпеки ПЗКС *pnvss* для 10 аналізованих специфікацій

Отже, прогнозовано матиме високий рівень безпеки ПЗКС, яке буде розроблене за специфікаціями вимог №2 та №8 (звісно, якщо помилки не будуть допущені на наступних етапах життєвого циклу), тому замовнику рекомендується виконати замовлення розроблення програмного забезпечення саме у софтверних компаніях, які підготували специфікації вимог №2 та №8.

Враховуючи результати проведених експериментальних досліджень, було зроблено висновок, що розроблений метод прогнозування рівня безпеки програмного забезпечення комп'ютерних систем встановлює залежність безпеки ПЗКС від атрибутів якості, формує прогнозоване числове значення безпеки ПЗКС на основі атрибутів, забезпечує прогнозування рівня безпеки ПЗКС на основі отриманого числового значення, а також забезпечує порівняння специфікацій вимог до ПЗ за прогнозованим рівнем безпеки розробленого ПЗКС та можливість відбраковування невдалих специфікацій.

3.2. Метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем [65]

Враховуючи правила класифікації відмов та вразливостей ПЗ, розроблені у [165], розробимо опитувальники для збору інформації про відмову(и) та про вразливість(і), які мали місце в процесі функціювання ПЗ.

Опитувальник для збору інформації про відмову(и):

1) Чи є стан ПЗ після припинення його функціонування роботоздатним?

2) Чи за час припинення функціонування ПЗ відбулась втрата даних?

На кожне із запитань такого опитувальника користувач може дати відповідь «так» або «ні».

Враховуючи питання розробленого опитувальника, *правила класифікації відмов* мають вигляд:

1) якщо обрано відповідь «так» на перше запитання опитувальника та обрано відповідь «ні» на друге запитання опитувальника, то змінна $sf = 1$;

2) якщо обрано відповідь «так» на перше запитання опитувальника та обрано відповідь «так» на друге запитання опитувальника, то змінна *sf = 2*;

3) якщо обрано відповідь «ні» на перше запитання опитувальника, то змінна *sf = 3*.

Опитувальник для збору інформації про вразливість(і):

1) Чи під час виконання певної функційної можливості ПЗКС припинило функціонування на час, що перевищує заданий пороговий час?

2) Чи після виконання певної функційної можливості відбулась втрата повноти даних?

3) Чи після виконання певної функційної можливості відбувся витік даних?

4) Чи після виконання певної функційної можливості виникла неможливість одержання інформації, дозволеної користувачу?

На кожне із запитань такого опитувальника користувач може дати відповідь «так» або «ні».

Враховуючи питання розробленого опитувальника, *правила класифікації вразливостей* мають вигляд:

1) якщо обрано відповідь «так» на перше запитання опитувальника для збору інформації про вразливість(і), то елемент матриці *sv[1,1] = 1*;

2) якщо обрано відповідь «так» на друге запитання опитувальника для збору інформації про вразливість(і), то елемент матриці *sv[1,2] = 1*;

3) якщо обрано відповідь «так» на третє запитання опитувальника для збору інформації про вразливість(і), то елемент матриці *sv[1,3] = 1*;

4) якщо обрано відповідь «так» на четверте запитання опитувальника для збору інформації про вразливість(і), то елемент матриці *sv[1,4] = 1*.

Отже, були створені опитувальники для збору інформації про відмову(и) та про вразливість(і), а також розроблено правила для ідентифікації та класифікації відмов та вразливостей на основі аналізу відповідей на питання розроблених опитувальників. Розроблені правила дозволяють ідентифікувати та класифікувати випадки відмов та вразливостей, що виникли в процесі функціонування програмного забезпечення комп’ютерних систем.

Метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем складається з наступних кроків:

вхідна інформація: опитувальники для збору інформації про відмову(и) та вразливість(і);

1) змінна *sf = 0*; заповнення першого рядка матриці *sv* нулями; заповнення другого рядка матриці *sv* з метою подальшого формування висновку про тип вразливості(ей): *sv[2,1]* = «функційна можливість ПЗКС є вразливістю коректної роботи»; *sv[2,2]* = «функційна можливість ПЗКС є вразливістю цілісності інформації»; *sv[2,3]* = «функційна можливість ПЗКС є вразливістю конфіденційності інформації»; *sv[2,4]* = «функційна можливість ПЗКС є вразливістю доступності інформації»;

2) проведення опитування користувача ПЗКС (за допомогою спеціально розроблених опитувальників для збору інформації про відмову(и) та про вразливість(і));

3) аналіз отриманих відповідей користувачів на запитання опитувальника для збору інформації про відмову(и) із використанням правил класифікації відмов та формування значення змінної *sf*;

4) якщо *sf=1*, то користувачу видається висновок «відмова ПЗ є неістотною», інакше, якщо *sf=2*, то користувачу видається висновок «відмова ПЗ є істотною», інакше якщо *sf=3*, то користувачу видається висновок «відмова ПЗ є критичною», інакше, якщо *sf=0*, то користувачу видається висновок «відмови ПЗ не відбувались»;

5) аналіз наданих користувачем відповідей на запитання опитувальника для збору інформації про вразливість(і) із використанням правил класифікації вразливостей та заповнення першого рядка матриці *sv*;

6) якщо *sv[1,i]=1 (i=1..4)*, то користувачу видається висновок про тип(и) вразливості – елемент *sv[2,i] (i=1..4)* матриці *sv*, інакше, якщо всі елементи першого рядка матриці *sv* дорівнюють 0, то користувачу видається висновок «функційна можливість ПЗ не є вразливістю»;

вихідна інформація: висновок про наявність та тип відмови, висновок про наявність та тип вразливості.

Розроблений метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем забезпечує висновок щодо того, чи відбувалась відмова, і, якщо відмова відбулась, то користувачу видається її тип. Крім цього, розроблений метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем забезпечує висновок щодо того, чи є функційна можливість вразливістю, і, якщо функційна можливість є вразливістю, то користувачу видається її тип. Таким чином, розроблений метод забезпечує безпеку програмного забезпечення комп'ютерних систем

3.2.1. Експериментальне дослідження методу ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем

Розглянемо, як функціонує створений метод ідентифікації та класифікації відмов і вразливостей ПЗКС.

Згідно із першим кроком розробленого методу ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем, відбулось обнулення змінної *sf* та елементів першого рядка матриці *sv*, а також заповнення другого рядка матриці *sv*. Згідно із другим кроком розробленого методу, проведено опитування користувача реального ПЗКС з використанням складених опитувальників для збору інформації про відмову(и) та про вразливість(і).

Згідно із третім кроком розробленого методу, виконано аналіз наданих користувачем відповідей на запитання опитувальника для збору інформації про відмову(и) із використанням правил класифікації відмов та формування значення змінної *sf*. Оскільки користувач реального ПЗКС дав відповідь «так» на перше запитання опитувальника для збору інформації про відмову(и) та відповідь «ні» на друге запитання опитувальника збору інформації про відмову, то змінна *sf* = 1.

Згідно із четвертим кроком розробленого методу, оскільки *sf*=1, то користувачу видається висновок «відмова ПЗ є неістотною».

Згідно із п’ятим кроком розробленого методу ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем, виконується аналіз відповідей користувача на запитання опитувальника для збору інформації про вразливість(і) із використанням правил класифікації вразливостей, а також заповнення першого рядка матриці *sv*. Користувач реального ПЗКС дав відповіді «так» на перше, третє і четверте запитання, відтак матриця *sv* має вигляд – таблиця 3.2.

Таблиця 3.2

Матриця *sv*, в якій містяться ознаки наявності чи відсутності вразливості ПЗКС, а також тип вразливості

	I стовпець	**II стовпець**	**III стовпець**	**IV стовпець**
I рядок	1	0	1	1
II рядок	функційна можливість ПЗ є вразливістю коректної роботи	функційна можливість ПЗ є вразливістю цілісності інформації	функційна можливість ПЗ є вразливістю конфіденційності інформації	функційна можливість ПЗ є вразливістю доступності інформації

Згідно із шостим кроком розробленого методу, оскільки *sv[1,1]*=1, то користувачу видається висновок про тип вразливості – «Функційна можливість ПЗ є вразливістю коректної роботи» (елемент *sv[2,1]* матриці *sv*). Оскільки *sv[1,3]*=1, то користувачу видається висновок про тип вразливості – «Функційна можливість ПЗ є вразливістю конфіденційності інформації» (елемент *sv[2,3]* матриці *sv*). Оскільки *sv[1,4]=1*, то користувачу видається висновок про тип вразливості – «Функційна можливість ПЗ є вразливістю доступності інформації» (елемент *sv[2,4]* матриці *sv*). Отже, функційна можливість ПЗ є вразливістю коректної роботи, конфіденційності та доступності інформації.

Проведене експериментальне дослідження методу ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем виявило, що шляхом опитування користувача реального ПЗКС було надано встановлено факт неістотної відмови ПЗКС, а також було виявлено вразливості коректної роботи, конфіденційності та доступності інформації в реальному програмному забезпеченні комп'ютерної системи.

3.3. Висновки

Концептуальна модель процесу прогнозування безпеки ПЗКС (як характеристики якості ПЗ) на основі атрибутів якості грунтується на ШНМ, яка обробляє подані на її входи значення 15 атрибутів якості з вимог до ПЗ, від яких залежить безпека ПЗ, здійснює апроксимацію цих значень та надає прогнозовану кількісну оцінку безпеки ПЗКС в інтервалі [0;1], де значення «0» означає найгірший рівень безпеки ПЗКС, а значення «1» означає найкращий рівень безпеки ПЗКС як характеристики якості ПЗ, а також дає можливість враховувати не тільки кількісне значення та важливість кожного атрибуту якості, але й взаємний вплив атрибутів при кількісному оцінюванні безпеки ПЗКС.

Розроблено метод прогнозування рівня безпеки програмного забезпечення комп'ютерних систем, який, на відміну від відомих, встановлює залежність безпеки ПЗКС від атрибутів якості та формує прогнозоване числове значення безпеки ПЗКС на основі атрибутів, і забезпечує прогнозування рівня безпеки ПЗКС на основі отриманого числового значення, а також забезпечує порівняння специфікацій вимог до ПЗ за прогнозованим рівнем безпеки розроблюваного ПЗКС та можливість відбраковування невдалих специфікацій.

Було розроблено, навчено і протестовано штучну нейронну мережу для прогнозування безпеки програмного забезпечення комп'ютерних систем, що розглядається як одна з характеристик якості ПЗ. Аналіз графіків навчання і тестування ШНМ дозволив зробити висновок, що мережа навчилась з високою точністю.

Під час експериментального дослідження методу прогнозування безпеки програмного забезпечення комп’ютерних систем як однієї з характеристик якості ПЗ на основі атрибутів якості, враховуючи отриманий прогнозований рівень безпеки майбутнього ПЗ, розробленого за кожною з аналізованих специфікацій, було проведено порівняння десяти аналізованих специфікацій вимог до ПЗ, що розроблялись різними ІТ-фірмами м. Хмельницького (Україна) для виконання одного й того ж замовлення, а також виконано обґрунтований вибір специфікації для подальшої реалізації безпечного ПЗКС. Оскільки прогнозовано матиме високий рівень безпеки програмне забезпечення комп’ютерних систем, яке буде розроблене за специфікаціями вимог №2 та №8, тому замовнику рекомендується виконати замовлення розроблення програмного забезпечення саме у софтверних компаніях, які підготували специфікації вимог №2 та №8.

Розроблено метод ідентифікації та класифікації відмов і вразливостей ПЗКС, який забезпечує висновок щодо того, чи відбувалась відмова, і, якщо відмова відбулась, то користувачу видається її тип. Крім цього, розроблений метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем забезпечує висновок щодо того, чи є функційна можливість вразливістю, і, якщо функційна можливість є вразливістю, то користувачу видається її тип. Таким чином, розроблений метод забезпечує безпеку програмного забезпечення комп’ютерних систем.

Проведене експериментальне дослідження методу ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем показав, що на основі опитування користувача реального ПЗКС було надано висновок щодо факту неістотної відмови ПЗКС, а також висновок щодо наявності вразливості коректної роботи, конфіденційності та доступності інформації в реальному ПЗКС.

РОЗДІЛ 4.
ЗАСОБИ ПРОГНОЗУВАННЯ ТА ОЦІНЮВАННЯ РІВНЯ ЯКОСТІ ТА БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

4.1. Системи прогнозування рівня якості та безпеки (як характеристики якості) програмного забезпечення комп'ютерних систем [61, 67]

В основі систем прогнозування рівня якості та безпеки ПЗКС, є концепція прогнозування рівня якості та безпеки ПЗ (як характеристики якості) на основі атрибутів якості, наявних у вимогах до ПЗ.

Принципами проєктування та функціонування таких систем є:

1) принцип автоматизації опрацювання інформації – використання автоматизованих засобів на всіх етапах опрацювання інформації;

2) принцип розвитку – модифікація функцій і складу системи без порушення її роботи;

3) принцип ефективності – максимальний ефект при мінімальних витратах;

4) принцип сумісності – взаємодія системи з іншими системами за встановленими правилами;

5) принцип системності – базування на єдиному методологічному підході;

6) принцип адаптивності до нових задач – модифікація системи відповідно до нових задач;

7) принцип законності – дотримання вимог стандартів (зокрема, стандарту ISO 25010 [3] та стандарту ISO 25023 [89]);

8) принцип відкритості інформації – забезпечення оперативності, регулярності та достовірності оброблюваної інформації;

9) принцип етапності – послідовний розвиток системи.

Функціонування системи прогнозування рівня якості програмного забезпечення комп'ютерних систем базується на двох методах: розробленому у

підрозділі 2.1 методі пошуку значень атрибутів якості у вимогах до програмного забезпечення комп’ютерних систем та розробленому у підрозділі 2.3 методі прогнозування рівня якості програмного забезпечення комп’ютерних систем на основі атрибутів якості.

Узагальнена структура системи прогнозування рівня якості програмного забезпечення комп’ютерних систем представлена на рис. 4.1.

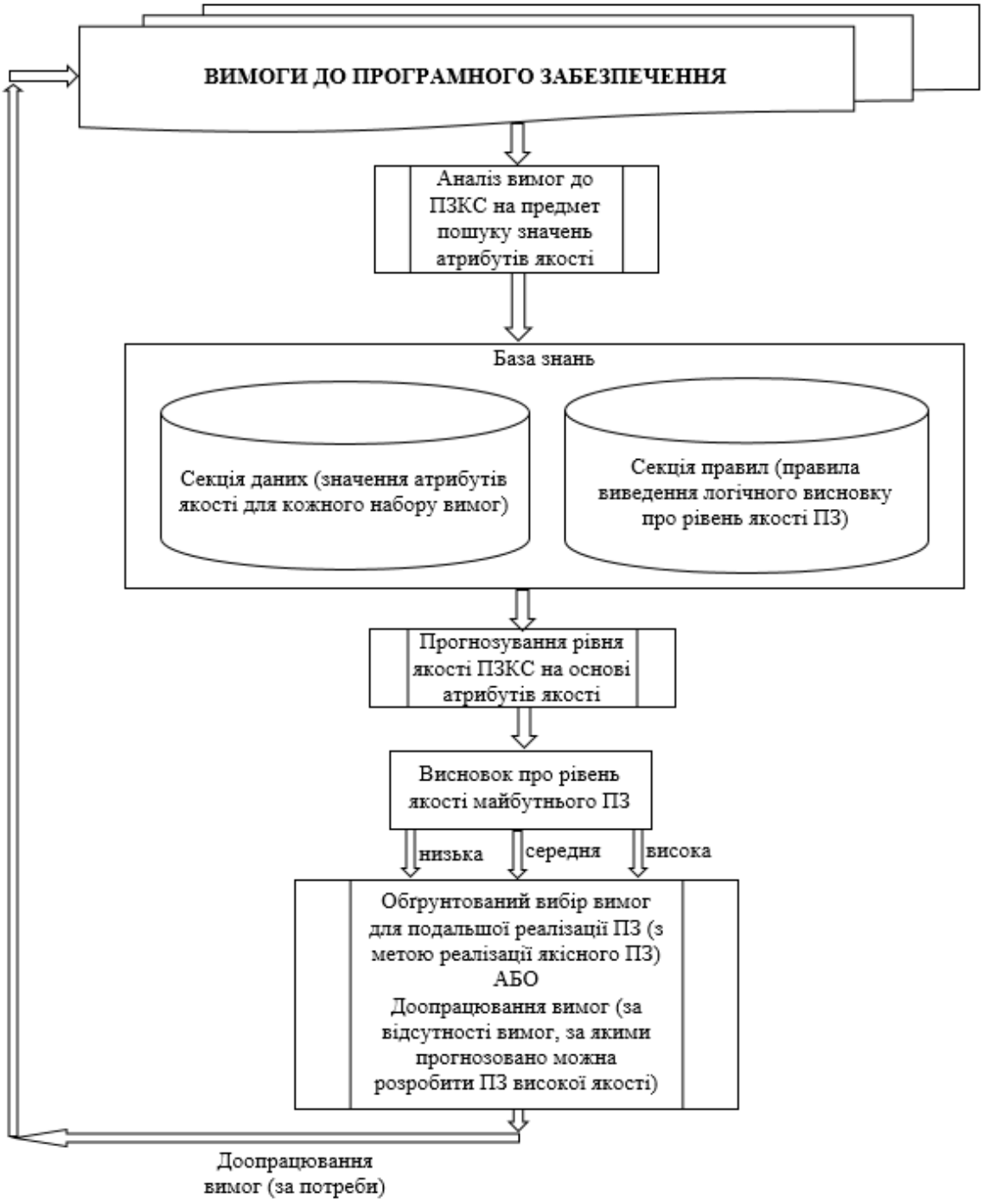


Рис. 4.1 – Узагальнена структура системи прогнозування рівня якості ПЗКС

Отже, пропонована система прогнозування рівня якості ПЗКС функціонує наступним чином:

1) на вхід системи подаються вимоги до ПЗ (наприклад, різні версії вимог, для кожної з яких потрібно прогнозувати рівень якості майбутнього ПЗКС);

2) проводиться аналіз вимог до ПЗ на предмет пошуку значень атрибутів якості (згідно із розробленим у підрозділі 2.1 методом пошуку значень атрибутів якості у вимогах до програмного забезпечення комп'ютерних систем);

3) в секції даних бази знань зберігаються значення атрибутів якості для кожного набору вимог, а також всі проміжні та результуючі дані системи;

4) згідно із розробленим у підрозділі 2.3 методом прогнозування рівня якості програмного забезпечення на основі атрибутів якості, штучна нейронна мережа надає прогнозовані оцінки восьми характеристик якості ПЗ, виконується геометрична інтерпретація значень характеристик якості ПЗ, виконується перевірка припустимості компенсації характеристик якості ПЗКС (за правилами перевірки припустимості компенсації характеристик якості ПЗ, які містяться у секції правил бази знань), відбувається розрахунок комплексного показника прогнозованої якості ПЗ та формування висновку про рівень якості майбутнього ПЗ, яке розробляється за аналізованим набором вимог (за правилами виведення логічного висновку про рівень якості ПЗ, які містяться у секції правил бази знань);

5) на основі отриманого висновку про рівень якості майбутнього ПЗКС, яке розробляється за певним набором вимог, виконується обґрунтований вибір вимог для подальшої реалізації ПЗКС (з метою реалізації якісного ПЗКС), або набори вимог відправляються на доопрацювання, якщо серед аналізованих наборів відсутній набір вимог, за яким прогнозовано можна розробити ПЗКС високої якості.

Запропонована система прогнозування рівня якості ПЗКС задовольняє всі вісім визначених критеріїв одночасно – забезпечує аналіз атрибутів якості у вимогах, відображає залежність характеристик якості від атрибутів, формує кількісну оцінку характеристик якості, відображає залежність якості від її характеристик, формує кількісну оцінку якості, виконує прогнозування рівня

якості, надає всі перераховані сервіси одночасно, базується на спільному методологічному підході.

Згідно із узагальненою структурою системи прогнозування рівня якості ПЗКС, представленою на рис. 4.1, розробимо деталізовану структуру з елементами архітектурних рішень такої системи – рис. 4.2.

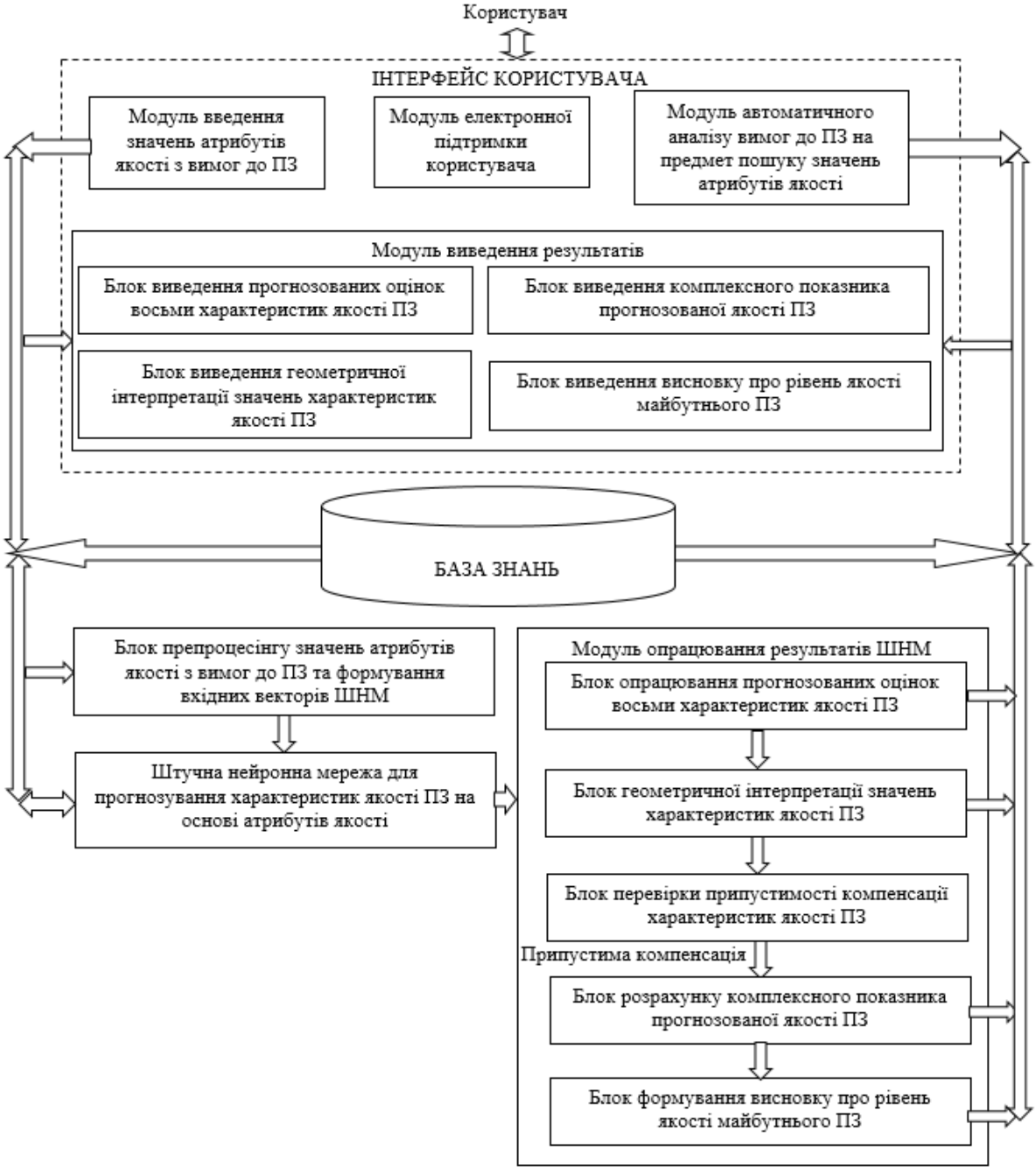


Рис. 4.2 – Структура з елементами архітектурних рішень системи прогнозування рівня якості ПЗКС

Інтерфейс користувача системи прогнозування рівня якості ПЗКС складається з декількох модулів:

– модуль введення значень атрибутів якості з вимог до ПЗ, який дозволяє користувачеві самостійно аналізувати вимоги до ПЗ та вручну вводити значення атрибутів якості, що містяться у вимогах;

– модуль автоматичного аналізу вимог на предмет пошуку значень атрибутів якості, який працює на основі розробленого у підрозділі 2.1 методу пошуку значень атрибутів якості у вимогах до програмного забезпечення комп'ютерних систем;

– модуль електронної підтримки користувача;

– модуль виведення результатів, який, в свою чергу, складається з таких блоків:

- блок виведення прогнозованих оцінок восьми характеристик якості ПЗ, що формуються ШНМ на основі атрибутів якості (вхідні значення для ШНМ формує блок препроцесінгу значень атрибутів якості з вимог до ПЗ та формування вхідних векторів ШНМ) та блоком опрацювання прогнозованих оцінок восьми характеристик якості ПЗ згідно із першим кроком розробленого у підрозділі 2.3 методу прогнозування рівня якості ПЗКС на основі атрибутів якості);
- блок виведення геометричної інтерпретації значень характеристик якості ПЗ – таку геометричну інтерпретацію формує блок геометричної інтерпретації значень характеристик якості ПЗ відповідно до другого кроку методу прогнозування рівня якості ПЗКС на основі атрибутів якості;
- блок виведення комплексного показника прогнозованої якості ПЗ – такий показник формується блоком розрахунку комплексного показника прогнозованої якості ПЗ відповідно до четвертого кроку методу прогнозування рівня якості ПЗКС на основі атрибутів якості, але після проходження перевірки припустимості компенсації характеристик якості ПЗ, яку проводить блок перевірки припустимості компенсації характеристик якості ПЗ відповідно до третього кроку методу прогнозування рівня якості ПЗКС на основі атрибутів якості;

- блок виведення висновку про рівень якості майбутнього ПЗКС – такий висновок формується блоком формування висновку про рівень якості майбутнього ПЗ відповідно до п’ятого кроку методу прогнозування рівня якості ПЗКС на основі атрибутів якості).

База знань системи включає в себе секцію даних, в якій зберігаються значення атрибутів якості для кожного набору вимог, а також всі проміжні дані та результати роботи системи (прогнозовані оцінки восьми характеристик якості ПЗ, геометрична інтерпретація значень характеристик якості ПЗ, комплексний показник прогнозованої якості ПЗ тощо). Також вона містить секцію правил, де зберігаються правила для перевірки припустимості компенсації характеристик якості ПЗ та правила виведення логічного висновку про рівень якості ПЗКС.

Запропонована система для прогнозування рівня якості ПЗКС виконує аналіз вимог, на основі якого видає користувачеві прогнозовані оцінки восьми характеристик якості ПЗКС, геометричну інтерпретацію цих оцінок, комплексний показник прогнозованої якості ПЗКС та висновок про рівень якості майбутнього ПЗКС, тобто дозволяє виконати порівняння специфікацій вимог до ПЗ та обґрунтований вибір специфікації вимог для подальшої реалізації.

Згідно із вище запропонованими концепцією та принципами проєктування та функціонування, розробимо також систему для прогнозування рівня безпеки програмного забезпечення комп’ютерних систем як характеристики якості – аналогічно до системи прогнозування рівня якості ПЗКС. Ця система буде базуватися на двох методах: розробленому у підрозділі 2.1 методі пошуку значень атрибутів якості у вимогах до програмного забезпечення комп’ютерних систем та розробленому у підрозділі 3.1 методі прогнозування рівня безпеки програмного забезпечення комп’ютерних систем.

Узагальнена структура системи прогнозування рівня безпеки програмного забезпечення комп’ютерних систем представлена на рис. 4.3.

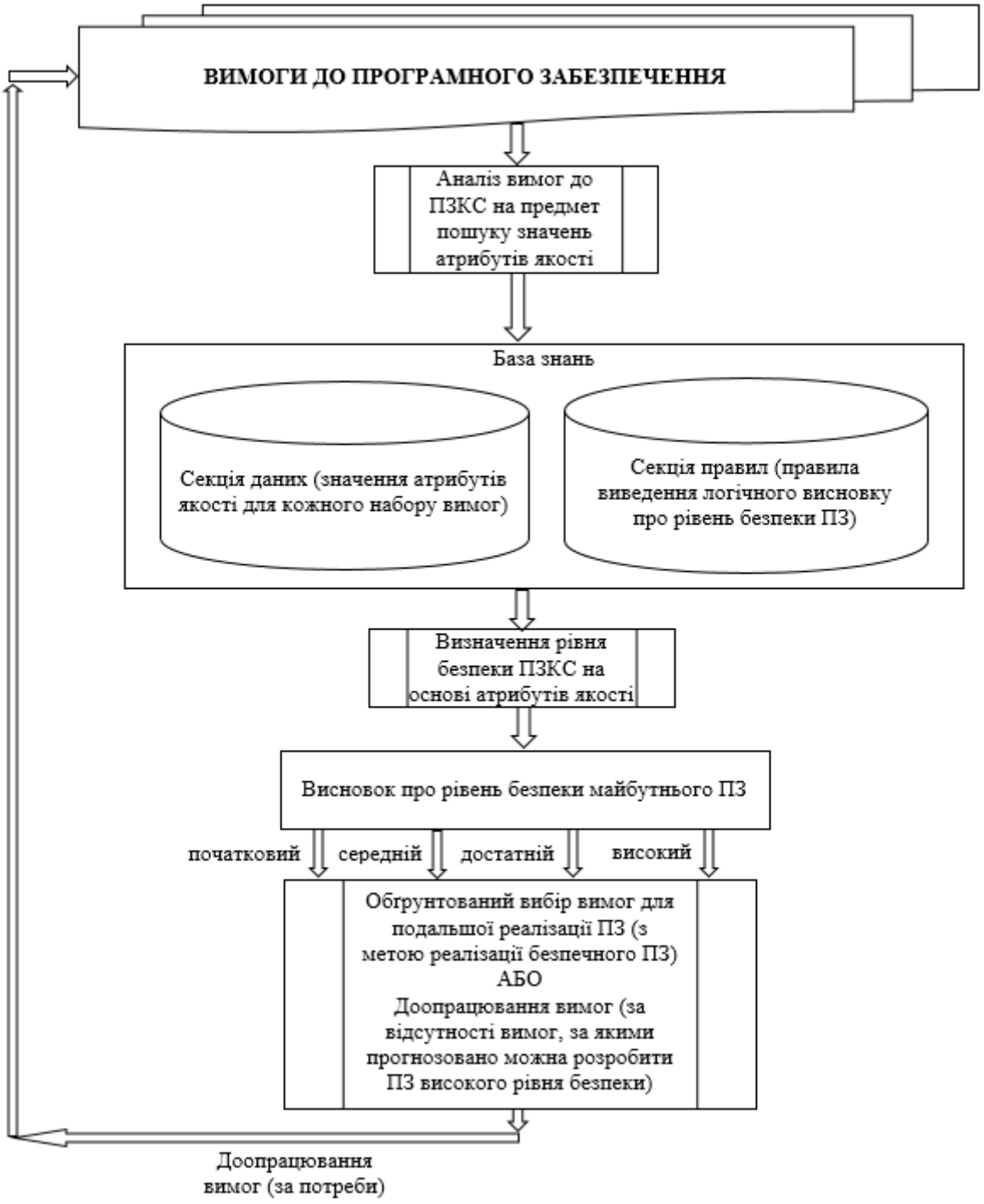


Рис. 4.3 – Узагальнена структура системи прогнозуваннч рівня безпеки ПЗКС (як характеристики якості)

Пропонована система прогнозування рівня безпеки ПЗКС (як характеристики якості) функціонує наступним чином:

1) на вхід системи подаються вимоги до ПЗ (наприклад, різні версії вимог, для кожної з яких потрібно визначити рівень безпеки майбутнього ПЗКС);

2) проводиться аналіз вимог до ПЗ на предмет пошуку значень атрибутів якості (згідно із розробленим у підрозділі 2.1 методом пошуку значень атрибутів якості у вимогах до програмного забезпечення комп'ютерних систем);

3) в секції даних бази знань зберігаються значення атрибутів якості для кожної специфікації вимог, а також всі проміжні та результуючі дані системи;

4) згідно із розробленим у підрозділі 3.1 методом прогнозування рівня безпеки ПЗКС, штучна нейронна мережа видає прогнозовану оцінку безпеки ПЗКС (як характеристики якості ПЗ) та формування висновку про рівень безпеки майбутнього ПЗКС, яке розробляється за аналізованою специфікацією вимог (за правилами виведення логічного висновку про рівень безпеки ПЗ, які містяться у секції правил бази знань);

5) на основі отриманого висновку про рівень безпеки майбутнього ПЗКС, яке розробляється за певною специфікацією вимог, виконується обґрунтований вибір вимог для подальшої реалізації ПЗКС (з метою реалізації ПЗКС високого рівня безпеки), або специфікації вимог відправляються на доопрацювання, якщо серед аналізованих специфікацій відсутня специфікація, за якою прогнозовано можна розробити ПЗКС високого рівня безпеки.

Запропонована система прогнозування рівня безпеки ПЗКС забезпечує аналіз атрибутів якості у вимогах, відображає залежність безпеки ПЗКС (як характеристики якості) від атрибутів, виконує прогнозування рівня безпеки та базується на спільному методологічному підході.

Згідно із узагальненою структурою системи прогнозування рівня безпеки ПЗКС (як характеристики якості), представленою на рис. 4.3, та за аналогією із пропонованою структурою з елементами архітектурних рішень системи прогнозування рівня якості програмного забезпечення комп'ютерних систем, розробимо деталізовану структуру з елементами архітектурних рішень системи прогнозування рівня безпеки ПЗКС – рис. 4.4.

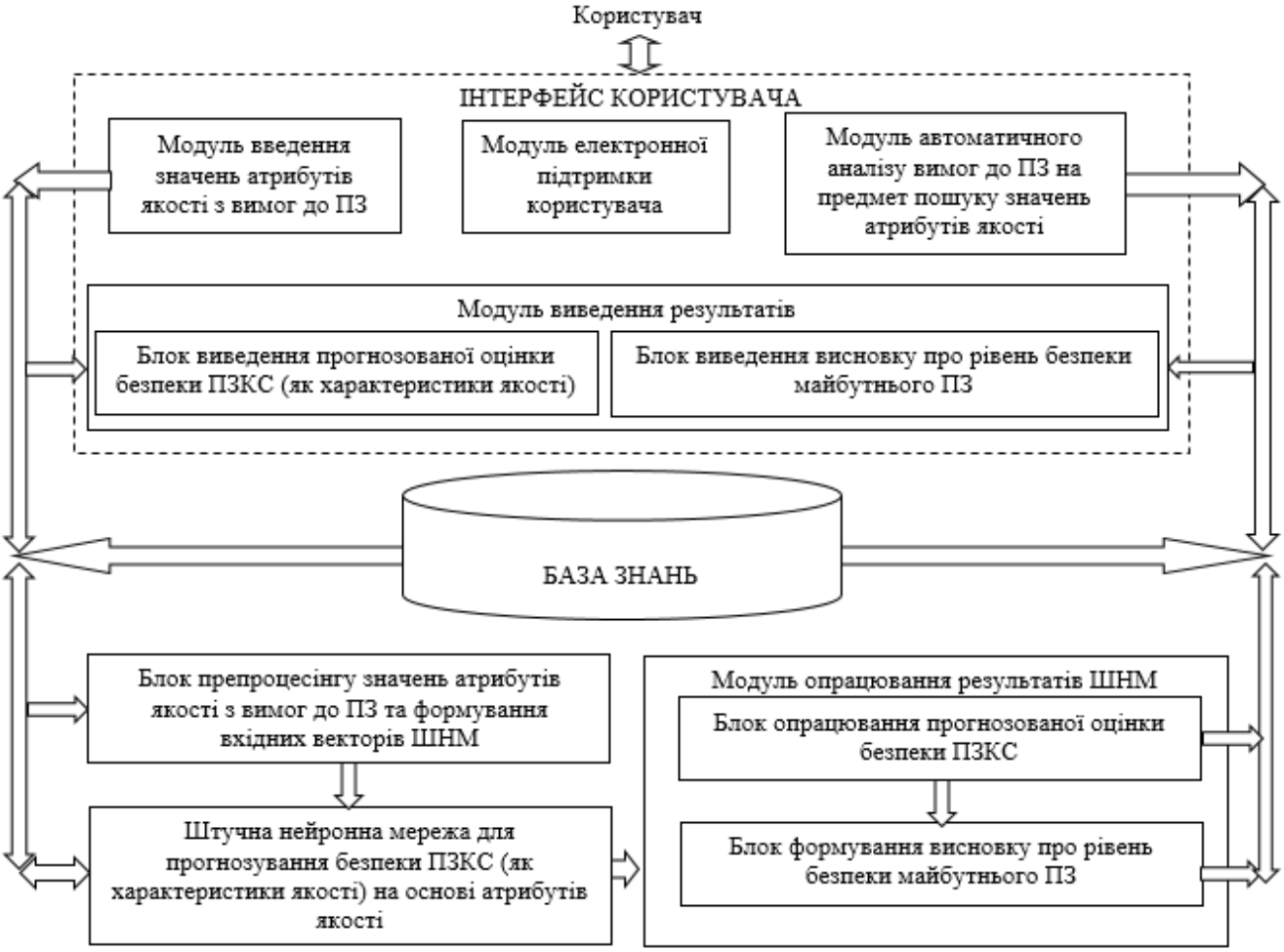


Рис. 4.4 – Деталізована структура з елементами архітектурних рішень системи прогнозування рівня безпеки ПЗКС (як характеристики якості)

Інтерфейс користувача системи прогнозування рівня безпеки ПЗКС (як характеристики якості) складається з: модуля введення значень атрибутів якості з вимог (орієнтований на ручний розбір вимог користувачем та ручне введення значень атрибутів якості, що містяться у вимогах), модуля автоматичного аналізу вимог до ПЗ з метою пошуку значень атрибутів якості (працює на основі розробленого у підрозділі 2.1 методу пошуку значень атрибутів якості у вимогах до програмного забезпечення комп’ютерних систем), модуля електронної підтримки користувача та модуля виведення результатів. Модуль виведення результатів складається з блоку виведення прогнозованої оцінки безпеки ПЗКС (як характеристики якості), що видається штучною нейронною мережею (її вхідні значення формуються блоком препроцесінгу значень атрибутів якості з вимог до

ПЗ та формування вхідних векторів ШНМ) та блоком опрацювання прогнозованої оцінки безпеки ПЗКС відповідно до кроків 4 і 5 розробленого у підрозділі 3.1 методу прогнозування рівня безпеки ПЗКС на основі атрибутів якості), а також з блоку виведення висновку про рівень безпеки майбутнього ПЗКС (який формується блоком формування висновку про рівень безпеки майбутнього ПЗКС відповідно до кроку 6 методу прогнозування рівня безпеки програмного забезпечення комп'ютерних систем на основі атрибутів якості). База знань системи складається з секції даних, в якій зберігаються значення атрибутів якості для кожного набору вимог, а також всі проміжні та результуючі дані системи (прогнозовані оцінки безпеки ПЗКС тощо), та секції правил, в якій зберігаються правила виведення логічного висновку про рівень безпеки ПЗКС.

Запропонована система прогнозування рівня безпеки програмного забезпечення комп'ютерних систем забезпечує аналіз вимог, на основі якого надає користувачу прогнозовану оцінку безпеки ПЗКС (як характеристики якості) та висновок про рівень безпеки майбутнього ПЗКС, на основі якого можна виконати порівняння специфікацій вимог до ПЗ та обґрунтований вибір специфікації вимог для подальшої реалізації.

4.2. Система ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем [63, 65]

Система ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем, представлена на рис. 4.5, надає висновок щодо наявності чи відсутності відмов(и) ПЗКС; висновок щодо наявності чи відсутності вразливості(ей) ПЗКС; висновок про тип відмови та тип вразливості за їх наявності.

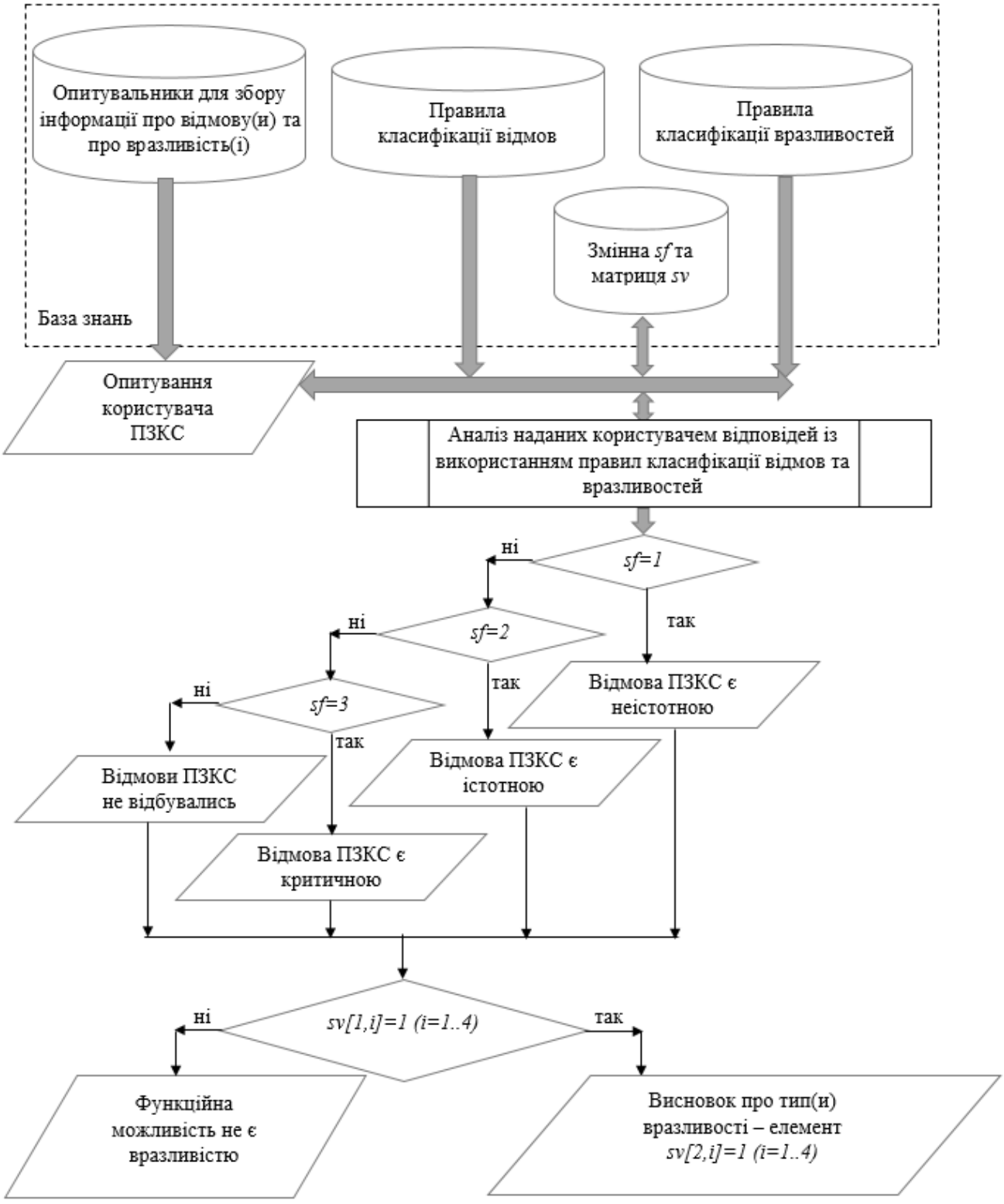


Рис. 4.5 – Структура системи ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем

Як видно з рис. 4.5, система ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем базується на

розробленому у підрозділі 3.2 методі ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем.

Робота системи починається з обнуління змінної *sf* та елементів першого рядка матриці *sv*, а також заповнення другого рядка матриці *sv*, згідно із першим етапом розробленого методу ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем.

Згідно із другим етапом розробленого методу, проводиться опитування користувача ПЗКС з використанням розроблених у підрозділі 3.2 опитувальників для збору інформації про відмову(и) та про вразливість(і), які містяться у базі знань системи.

Далі, згідно із третім етапом розробленого методу, відповіді користувача аналізуються із використанням розроблених у підрозділі 3.2 правил класифікації відмов та вразливостей, які також містяться у базі знань системи.

Після такого аналізу, згідно із четвертим етапом розробленого в підрозділі 3.2 методу ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем, формується значення змінної *sf* та заповнюється перший рядок матриці *sv*, а також відбувається аналіз значення змінної *sf*, завдяки чому відбувається класифікація відмов (за їх наявності), та аналіз значень елементів першого рядка матриці *sv*, завдяки чому визначається, чи є функційна можливість вразливістю, і якщо є, то який тип вона має.

Отже, розроблена система ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем забезпечує висновок щодо того, чи відбувалась відмова, і, якщо відмова відбулась, то користувачу видається її тип. Крім цього, розроблена система ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем забезпечує висновок щодо того, чи є функційна можливість вразливістю, і, якщо функційна можливість є вразливістю, то користувачу видається її тип.

4.3. Результати експериментальних досліджень

4.3.1. Результати функціонування системи прогнозування рівня якості програмного забезпечення комп'ютерних систем [61]

Для *експерименту 1* опрацюємо з використанням системи прогнозування рівня якості ПЗКС 6 специфікацій (наборів) вимог, які розроблялись різними ІТ-фірмами м. Хмельницького (Україна) для розв'язання однієї й тієї ж задачі.

Користувач скористався модулем автоматичного аналізу вимог до ПЗ на предмет пошуку значень атрибутів якості, який провів розбір 6 аналізованих наборів вимог та вибрав значення 138 атрибутів якості ПЗКС, що містяться у кожному наборі вимог.

Значення атрибутів якості з кожного набору вимог були збережені в секції даних бази знань та передані до блоку препроцесінгу значень атрибутів якості з вимог до ПЗ та формування вхідних векторів ШНМ, який сформував 6 наборів вхідних векторів для ШНМ. Такі набори вхідних даних ШНМ були подані один за одним на нейрони вхідного шару ШНМ для прогнозування характеристик якості ПЗ на основі атрибутів. Результати функціонування ШНМ були опрацьовані блоком опрацювання прогнозованих оцінок восьми характеристик якості ПЗ та виведені користувачу за допомогою блоку виведення прогнозованих оцінок восьми характеристик якості ПЗ в наступному вигляді:

– для специфікації (набору) вимог №1: зручність використання становить 0.52, функційна придатність – 0.48, надійність – 0.53, ефективність – 0.49, безпека – 0.45, сумісність – 0.44, можливість переносу – 0.55, супроводжуваність – 0.60;

– для специфікації (набору) вимог №2: зручність використання становить 0.12, функційна придатність – 0.40, надійність – 0.33, ефективність – 0.38, безпека – 0.36, сумісність – 0.43, можливість переносу – 0.45, супроводжуваність – 0.51;

– для специфікації (набору) вимог №3: зручність використання становить 0.93, функційна придатність – 0.97, надійність – 0.89, ефективність – 0.87, безпека – 0.96, сумісність – 0.91, можливість переносу – 0.90, супроводжуваність – 0.85;

– для специфікації (набору) вимог №4: зручність використання становить 0.80, функційна придатність – 0.77, надійність – 0.74, ефективність – 0.70, безпека – 0.65, сумісність – 0.10, можливість переносу – 0.82, супроводжуваність – 0.84;

– для специфікації (набору) вимог №5: зручність використання становить 0.13, функційна придатність – 0.16, надійність – 0.14, ефективність – 0.20, безпека – 0.15, сумісність – 0.13, можливість переносу – 0.12, супроводжуваність – 0.15;

– для специфікації (набору) вимог №6: зручність використання становить 0.92, функційна придатність – 0.84, надійність – 0.80, ефективність – 0.79, безпека – 0.95, сумісність – 0.96, можливість переносу – 0.99, супроводжуваність – 0.90.

Прогнозовані оцінки восьми характеристик якості ПЗ були передані на блок геометричної інтерпретації значень характеристик якості ПЗ, результати роботи якого були виведені користувачу за допомогою блоку виведення геометричної інтерпретації значень характеристик якості ПЗ у наступному вигляді – рис. 4.6.

Прогнозовані оцінки восьми характеристик якості ПЗКС та геометрична інтерпретація значень характеристик якості ПЗ також були передані на блок перевірки припустимості компенсації характеристик якості ПЗ, де всі 6 восьмикутників, утворених значеннями характеристик якості ПЗ для 6 різних наборів вимог, були перевірені системою на предмет опуклості. Так, восьмикутники, утворені для специфікацій (наборів) вимог № 1, № 3, №5, № 6, є опуклими, тому компенсація характеристик якості ПЗКС для таких наборів вимог є припустимою. Восьмикутники ж, утворені для специфікацій (наборів) вимог №2, №4 не є опуклими, відтак компенсація характеристик якості ПЗКС для таких наборів вимог не є припустимою, відповідно для специфікацій (наборів) вимог №2, №4 система не формує комплексний показник прогнозованої якості ПЗКС та висновок про рівень якості майбутнього ПЗКС, тому що такі набори вимог не варто розглядати та реалізовувати (без доопрацювання) через надто низьке значення однієї характеристики якості порівняно з іншими характеристиками, хоча всі вони мають однакову важливість для майбутнього програмного забезпечення комп’ютерних систем.

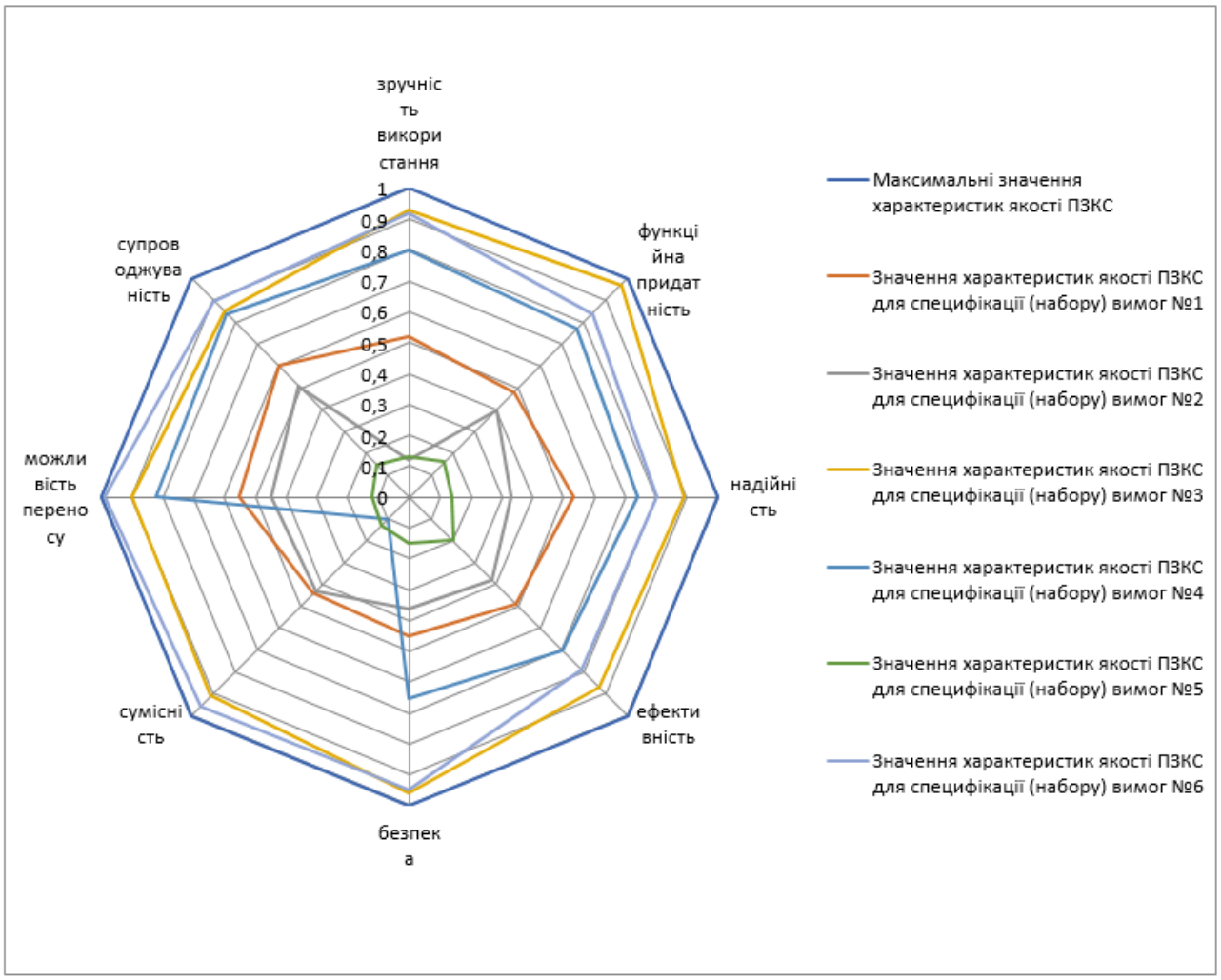


Рис. 4.6 – Геометрична інтерпретація значень характеристик якості ПЗКС для 6 аналізованих специфікацій (наборів) вимог до ПЗ (експеримент 1)

Прогнозовані оцінки восьми характеристик якості ПЗКС та геометрична інтерпретація значень характеристик якості ПЗ для наборів вимог №1, №3, № 5, №6 передавались на блок розрахунку комплексного показника прогнозованої якості ПЗ, результати роботи якого надавались користувачу за допомогою блоку виведення комплексного показника прогнозованої якості ПЗ у такому вигляді:

– для специфікації (набору) вимог №1 комплексний показник прогнозованої якості ПЗ становить 0.7305;

– для специфікації (набору) вимог №3 комплексний показник прогнозованої якості ПЗ становить 2.3416;

– для специфікації (набору) вимог №5 комплексний показник прогнозованої якості ПЗ становить 0.0615;

– для специфікації (набору) вимог №6 комплексний показник прогнозованої якості ПЗ становить 2.2658.

Значення комплексних показників прогнозованої якості ПЗКС для специфікацій (наборів) вимог №1, №3, №5, №6 були передані на блок формування висновку про рівень якості майбутнього ПЗ, результати роботи якого були виведені користувачу за допомогою блоку формування висновку про рівень якості майбутнього ПЗ у наступному вигляді:

– для специфікації (набору) вимог №1 майбутнє ПЗКС прогнозовано матиме середній рівень якості; за потреби реалізовувати саме цей набір вимог, вимоги повинні бути доопрацьовані і додатково проаналізовані на предмет прогнозування рівня якості ПЗКС;

– для специфікації (набору) вимог №2 компенсація характеристик якості ПЗКС для таких наборів вимог не є припустимою; за потреби реалізовувати саме цей набір вимог, вимоги повинні бути доопрацьовані і додатково проаналізовані на предмет прогнозування рівня якості ПЗКС;

– для специфікації (набору) вимог №3 майбутнє ПЗКС прогнозовано матиме високий рівень якості;

– для специфікації (набору) вимог №4 компенсація характеристик якості ПЗКС для таких наборів вимог не є припустимою; за потреби реалізовувати саме цей набір вимог, вимоги повинні бути доопрацьовані і додатково проаналізовані на предмет прогнозування рівня якості ПЗКС;

– для специфікації (набору) вимог №5 майбутнє ПЗКС прогнозовано матиме низький рівень якості; за потреби реалізовувати саме цей набір, вимоги повинні бути доопрацьовані і проаналізовані на предмет прогнозування рівня якості ПЗКС;

– для специфікації (набору) вимог №6 майбутнє ПЗКС прогнозовано матиме високий рівень якості.

Отже, майбутнє ПЗКС, реалізоване за специфікаціями (наборами) вимог №3 та №6, прогнозовано матиме високий рівень якості, тому система прогнозування рівня якості програмного забезпечення комп’ютерних систем рекомендує обрати для подальшої роботи саме такі специфікації (набори) вимог як такі, за якими

потенційно може бути реалізоване ПЗКС високої якості. Інші 4 аналізовані специфікації (набори) вимог (якщо є потреба у їх використанні) потребують доопрацювання та повторного аналізу з використанням розробленої системи прогнозування рівня якості програмного забезпечення комп'ютерних систем для встановлення рівня якості ПЗ, яке буде реалізовуватись за доопрацьованою специфікацією (набором) вимог.

Для *експерименту 2* опрацюємо з використанням системи прогнозування рівня якості програмного забезпечення комп'ютерних систем 4 специфікації (набори) вимог, які розроблялись різними ІТ-фірмами м. Хмельницького (Україна) для розв'язання однієї й тієї ж задачі на замовлення ТОВ «Деймос» (м. Хмельницький). Користувач також скористався модулем автоматичного аналізу вимог до ПЗ на предмет пошуку значень атрибутів якості, який провів розбір 4 аналізованих наборів вимог та вибрав значення 138 атрибутів якості ПЗКС, що містяться у кожному наборі вимог.

Значення атрибутів якості з кожного набору вимог були збережені в секції даних бази знань та передані до блоку препроцесінгу значень атрибутів якості з вимог до ПЗ та формування вхідних векторів ШНМ, який сформував 6 наборів вхідних векторів для ШНМ. Такі набори вхідних векторів для ШНМ були послідовно подані на нейрони вхідного шару штучної нейронної мережі для прогнозування характеристик якості ПЗ на основі атрибутів якості. Результати функціонування ШНМ були опрацьовані блоком опрацювання прогнозованих оцінок восьми характеристик якості ПЗ та виведені користувачу за допомогою блоку виведення прогнозованих оцінок восьми характеристик якості ПЗ в наступному вигляді:

– для специфікації (набору) вимог №1: зручність використання становить 0.15, функційна придатність – 0.18, надійність – 0.16, ефективність – 0.19, безпека – 0.15, сумісність – 0.14, можливість переносу – 0.15, супроводжуваність – 0.13;

– для специфікації (набору) вимог №2: зручність використання становить 0.45, функційна придатність – 0.50, надійність – 0.47, ефективність – 0.58, безпека – 0.54, сумісність – 0.51, можливість переносу – 0.49, супроводжуваність – 0.50;

– для специфікації (набору) вимог №3: зручність використання становить 0.90, функційна придатність – 0.91, надійність – 0.94, ефективність – 0.89, безпека – 0.95, сумісність – 0.88, можливість переносу – 0.92, супроводжуваність – 0.93;

– для специфікації (набору) вимог №4: зручність використання становить 0.23, функційна придатність – 0.27, надійність – 0.25, ефективність – 0.30, безпека – 0.25, сумісність – 0.21, можливість переносу – 0.22, супроводжуваність – 0.24.

Прогнозовані оцінки восьми характеристик якості ПЗ були передані на блок геометричної інтерпретації значень характеристик якості ПЗ, результати роботи якого були виведені користувачу за допомогою блоку виведення геометричної інтерпретації значень характеристик якості ПЗ у наступному вигляді – рис. 4.7.

Прогнозовані оцінки восьми характеристик якості ПЗКС та геометрична інтерпретація значень характеристик якості ПЗ також були передані на блок перевірки припустимості компенсації характеристик якості ПЗ, де всі 4 восьмикутники, утворені значеннями характеристик якості ПЗ для 4 різних специфікацій (наборів) вимог, були перевірені системою на предмет опуклості. Так, всі 4 восьмикутники, утворені для 4-х аналізованих специфікацій (наборів) вимог, є опуклими, тому компенсація характеристик якості ПЗКС для таких наборів вимог є припустимою.

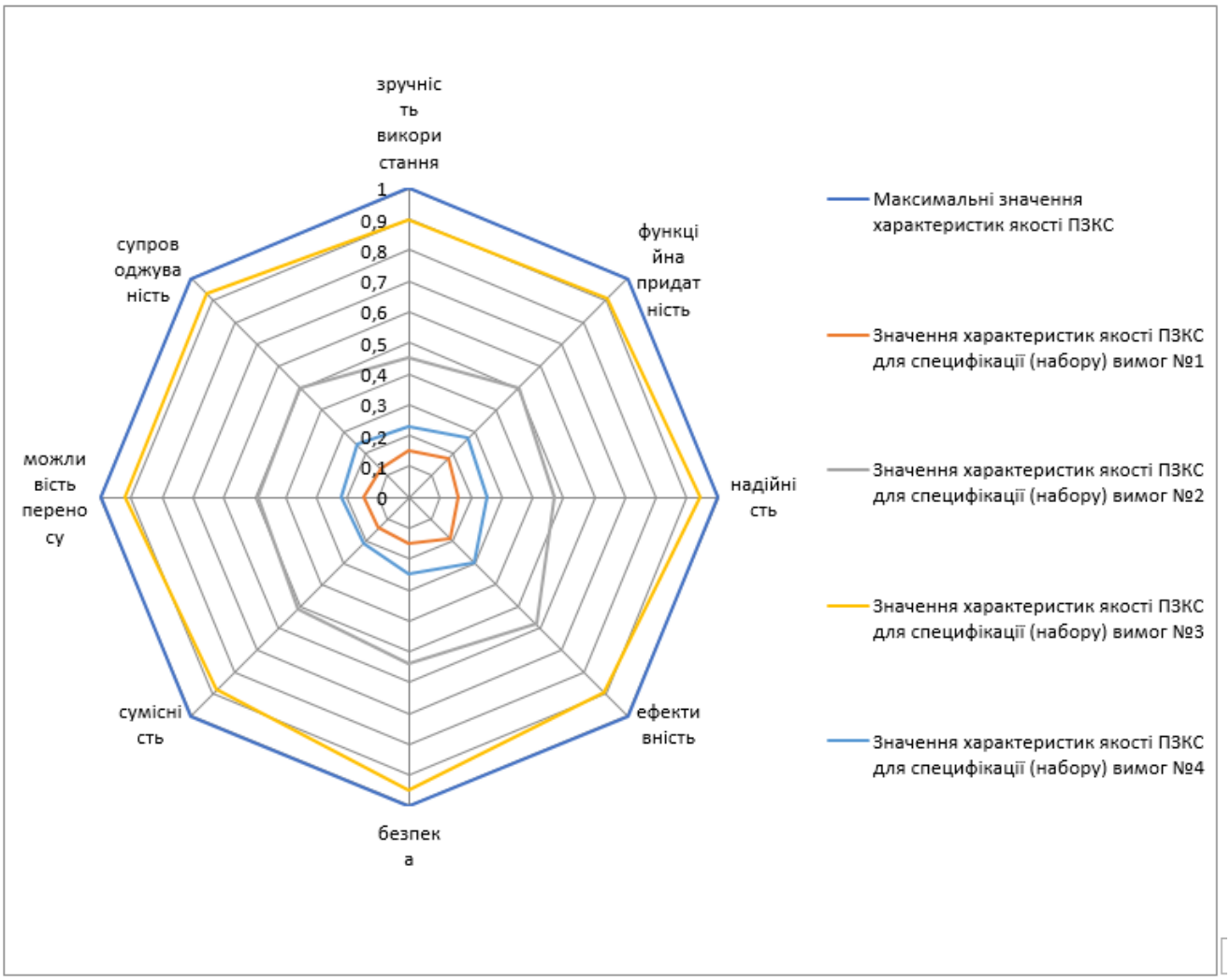


Рис. 4.7 – Геометрична інтерпретація значень характеристик якості ПЗКС для 4-х аналізованих специфікацій (наборів) вимог до ПЗ (експеримент 2)

Прогнозовані оцінки восьми характеристик якості ПЗКС та геометрична інтерпретація значень характеристик якості ПЗ для 4-х аналізованих специфікацій (наборів) вимог були передані на блок розрахунку комплексного показника прогнозованої якості ПЗ, результати роботи якого були виведені користувачу за допомогою блоку виведення комплексного показника прогнозованої якості ПЗ у наступному вигляді:

– для специфікації (набору) вимог №1 комплексний показник прогнозованої якості ПЗ становить 0.0692;

– для специфікації (набору) вимог №2 комплексний показник прогнозованої якості ПЗ становить 0.7216;

– для специфікації (набору) вимог №3 комплексний показник прогнозованої якості ПЗ становить 2.3669;

– для специфікації (набору) вимог №4 комплексний показник прогнозованої якості ПЗ становить 0.1719.

Значення комплексних показників прогнозованої якості ПЗКС для 4-х аналізованих специфікацій (наборів) вимог були передані на блок формування висновку про рівень якості майбутнього ПЗ, результати роботи якого були виведені користувачу за допомогою блоку формування висновку про рівень якості майбутнього ПЗ у наступному вигляді:

– для специфікації (набору) вимог №1 майбутнє ПЗКС прогнозовано матиме низький рівень якості; за потреби реалізовувати саме цей набір вимог, вимоги повинні бути доопрацьовані і додатково проаналізовані на предмет прогнозування рівня якості ПЗКС;

– для специфікації (набору) вимог №2 майбутнє ПЗКС прогнозовано матиме середній рівень якості; за потреби реалізовувати саме цей набір вимог, вимоги повинні бути доопрацьовані і додатково проаналізовані на предмет прогнозування рівня якості ПЗКС;

– для специфікації (набору) вимог №3 майбутнє ПЗКС прогнозовано матиме високий рівень якості;

– для специфікації (набору) вимог №4 майбутнє ПЗКС прогнозовано матиме низький рівень якості; за потреби реалізовувати саме цей набір вимог, вимоги повинні бути доопрацьовані і додатково проаналізовані на предмет прогнозування рівня якості ПЗКС.

Отже, майбутнє ПЗКС, реалізоване за специфікацією (набором) вимог №3, прогнозовано матиме високий рівень якості, тому система прогнозування рівня якості програмного забезпечення комп’ютерних систем рекомендує обрати для подальшої роботи саме таку специфікацію (набір) вимог як такий, за яким потенційно може бути реалізоване ПЗКС високої якості. Інші 3 аналізовані специфікації (набори) вимог (якщо є потреба у їх використанні) потребують доопрацювання та повторного аналізу з використанням розробленої системи

прогнозування рівня якості програмного забезпечення комп'ютерних систем для встановлення рівня якості ПЗ, яке буде реалізовуватись за доопрацьованою специфікацією (набором) вимог.

Для *експерименту 3* опрацюємо з використанням системи прогнозування рівня якості програмного забезпечення комп'ютерних систем 3 специфікації (набори) вимог, які розроблялись різними ІТ-фірмами м. Хмельницького (Україна) для розв'язання однієї й тієї ж задачі. Користувач також скористався модулем автоматичного аналізу вимог до ПЗ на предмет пошуку значень атрибутів якості, який провів розбір 3-х аналізованих наборів вимог та вибрав значення 138 атрибутів якості ПЗКС, що містяться у кожному наборі вимог.

Значення атрибутів якості з кожного набору вимог були збережені в секції даних бази знань та передані до блоку препроцесінгу значень атрибутів якості з вимог до ПЗ та формування вхідних векторів ШНМ, який сформував 6 наборів вхідних векторів для ШНМ. Такі набори вхідних векторів для ШНМ були послідовно подані на нейрони вхідного шару штучної нейронної мережі для прогнозування характеристик якості ПЗ на основі атрибутів якості. Результати функціонування ШНМ були опрацьовані блоком опрацювання прогнозованих оцінок восьми характеристик якості ПЗ та виведені користувачу за допомогою блоку виведення прогнозованих оцінок восьми характеристик якості ПЗ в наступному вигляді:

– для специфікації (набору) вимог №1: зручність використання становить 0.85, функційна придатність – 0.82, надійність – 0.80, ефективність – 0.81, безпека – 0.84, сумісність – 0.83, можливість переносу – 0.79, супроводжуваність – 0.80;

– для специфікації (набору) вимог №2: зручність використання становить 0.35, функційна придатність – 0.31, надійність – 0.37, ефективність – 0.38, безпека – 0.34, сумісність – 0.31, можливість переносу – 0.39, супроводжуваність – 0.30;

– для специфікації (набору) вимог №3: зручність використання становить 0.55, функційна придатність – 0.56, надійність – 0.57, ефективність – 0.54, безпека – 0.53, сумісність – 0.55, можливість переносу – 0.57, супроводжуваність – 0.52.

Прогнозовані оцінки восьми характеристик якості ПЗ були передані на блок геометричної інтерпретації значень характеристик якості ПЗ, результати роботи якого були виведені користувачу за допомогою блоку виведення геометричної інтерпретації значень характеристик якості ПЗ у наступному вигляді – рис. 4.8.

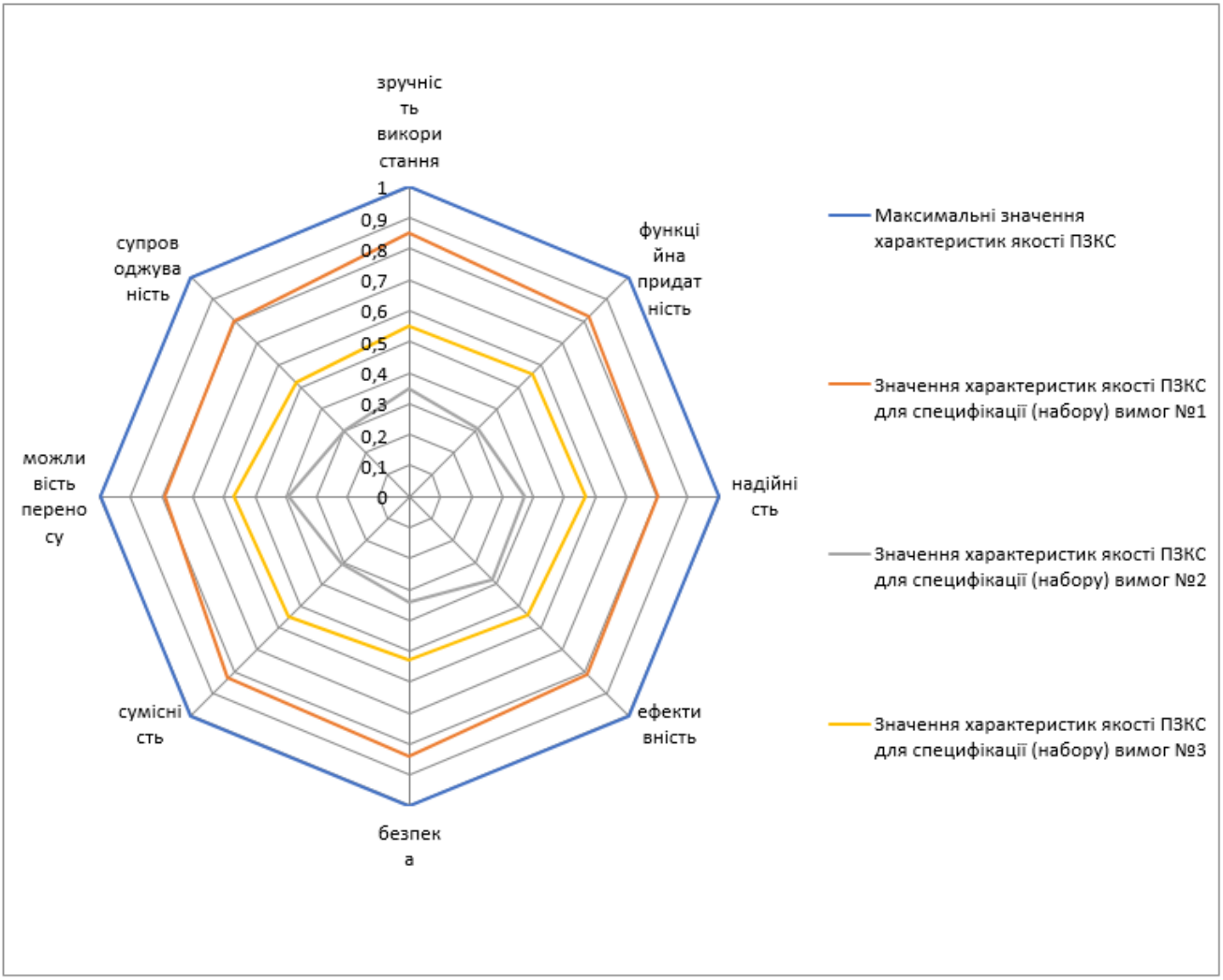


Рис. 4.8 – Геометрична інтерпретація значень характеристик якості ПЗКС для 3-х аналізованих специфікацій (наборів) вимог до ПЗ (експеримент 3)

Прогнозовані оцінки восьми характеристик якості ПЗКС та геометрична інтерпретація значень характеристик якості ПЗ також були передані на блок перевірки припустимості компенсації характеристик якості ПЗ, де всі 3 восьмикутники, утворені значеннями характеристик якості ПЗ для 3 різних специфікацій (наборів) вимог, були перевірені системою на предмет опуклості.

Так, всі 3 восьмикутники, утворені для 3-х аналізованих специфікацій (наборів) вимог, є опуклими, тому компенсація характеристик якості ПЗКС для таких наборів вимог є припустимою.

Прогнозовані оцінки восьми характеристик якості ПЗКС та геометрична інтерпретація значень характеристик якості ПЗ для 3-х аналізованих специфікацій (наборів) вимог були передані на блок розрахунку комплексного показника прогнозованої якості ПЗ, результати роботи якого були виведені користувачу за допомогою блоку виведення комплексного показника прогнозованої якості ПЗ у наступному вигляді:

– для специфікації (набору) вимог №1 комплексний показник прогнозованої якості ПЗ становить 1.8902;

– для специфікації (набору) вимог №2 комплексний показник прогнозованої якості ПЗ становить 0.3328;

– для специфікації (набору) вимог №3 комплексний показник прогнозованої якості ПЗ становить 0.8516.

Значення комплексних показників прогнозованої якості ПЗКС для 3-х аналізованих специфікацій (наборів) вимог були передані на блок формування висновку про рівень якості майбутнього ПЗ, результати роботи якого були виведені користувачу за допомогою блоку формування висновку про рівень якості майбутнього ПЗ у наступному вигляді:

– для специфікації (набору) вимог №1 майбутнє ПЗКС прогнозовано матиме середній рівень якості; за потреби реалізовувати саме цей набір вимог, вимоги повинні бути доопрацьовані і додатково проаналізовані на предмет прогнозування рівня якості ПЗКС;

– для специфікації (набору) вимог №2 майбутнє ПЗКС прогнозовано матиме середній рівень якості; за потреби реалізовувати саме цей набір вимог, вимоги повинні бути доопрацьовані і додатково проаналізовані на предмет прогнозування рівня якості ПЗКС;

– для специфікації (набору) вимог №3 майбутнє ПЗКС прогнозовано матиме середній рівень якості; за потреби реалізовувати саме цей набір вимог, вимоги

повинні бути доопрацьовані і додатково проаналізовані на предмет прогнозування рівня якості ПЗКС.

Отже, майбутнє ПЗКС, реалізоване за усіма трьома аналізованими специфікаціями (наборами) вимог, прогнозовано матиме середній рівень якості, тому система прогнозування рівня якості програмного забезпечення комп'ютерних систем рекомендує виконати доопрацювання таких специфікацій (наборів) вимог та повторний аналіз для встановлення рівня якості ПЗ, яке буде реалізовуватись за доопрацьованою специфікацією (набором) вимог.

Результати проведених трьох експериментів відображені в таблиці 4.1.

Таблиця 4.1

Результати функціонування системи прогнозування рівня якості програмного забезпечення комп'ютерних систем (3 експерименти)

		Комплексний показник прогнозованої якості ПЗКС	**Висновок про рівень якості майбутнього ПЗКС**
Експеримент 1	Специфікація (набір) вимог №1	0.7305	Середній
	Специфікація (набір) вимог №2	Не формується	Не визначається
	Специфікація (набір) вимог №3	2.3416	Високий
	Специфікація (набір) вимог №4	Не формується	Не визначається
	Специфікація (набір) вимог №5	0.0615	Низький
	Специфікація (набір) вимог №6	2.2658	Високий

Експеримент 2	Специфікація (набір) вимог №1	0.0692	Низький
	Специфікація (набір) вимог №2	0.7216	Середній
	Специфікація (набір) вимог №3	2.3669	Високий
	Специфікація (набір) вимог №4	0.1719	Низький
Експеримент 3	Специфікація (набір) вимог №1	1.8902	Середній
	Специфікація (набір) вимог №2	0.3328	Середній
	Специфікація (набір) вимог №3	0.8516	Середній

Отже, експериментально доведено, що запропонована система прогнозування рівня якості ПЗКС здатна проводити аналіз вимог на предмет пошуку значень атрибутів якості, визначати взаємозв'язок між значеннями характеристик якості та значеннями атрибутів, розраховувати кількісні оцінки характеристик якості, встановлювати зв'язок між значенням якості та значеннями характеристик якості, а також прогнозувати рівень якості. Ця система забезпечує всі вищезазначені функції одночасно і використовує єдиний методологічний підхід.

Під час подальшого використання розробленої системи прогнозування якості ПЗКС було розглянуто ще 290 різних наборів вимог, які були розроблені IT-фірмами м. Хмельницького (Україна) для вирішення різних задач.

Для 82 специфікацій (наборів) вимог з наявної множини з 290 прогнозів системою було сформовано висновок про високий рівень якості майбутнього ПЗКС; для 98 наборів було сформовано висновок про середній рівень якості майбутнього ПЗКС; для 75 наборів було сформовано висновок про низький рівень

якості майбутнього ПЗКС; для 35 наборів було сформовано висновок, що компенсація характеристик якості ПЗКС не є припустимою (тобто прогноз рівня якості для тих наборів вимог не надавався), отже, тільки за 28% наборами аналізованих вимог потенційно може бути реалізоване ПЗКС високої якості (рис. 4.9).

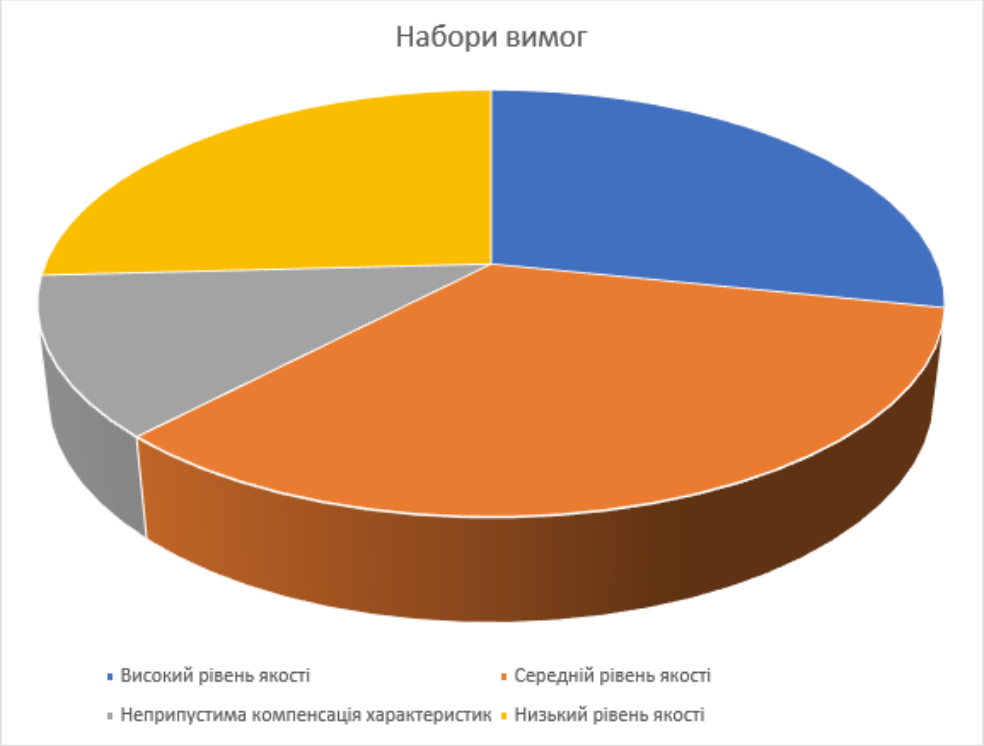


Рис. 4.9 – Розподіл множини прогнозів щодо якості майбутнього ПЗКС для проаналізованих 290 специфікацій (наборів) вимог

Як показав аналіз наявної множини прогнозів, ухвалених розробленою системою, лише за 28% наборами аналізованих вимог потенційно може бути реалізоване ПЗКС високої якості, в той час як за 34% наборами аналізованих вимог потенційно може бути реалізоване ПЗКС лише середньої якості, за 26% наборами аналізованих вимог може бути реалізоване ПЗКС низької якості, а за 12% наборами аналізованих вимог взагалі не варто виконувати розробку ПЗКС без їх ретельного доопрацювання через надто низьке значення однієї характеристики якості порівняно з іншими характеристиками, тобто фактично тільки чверть наборів вимог потенційно можуть привести до реалізації якісного ПЗКС.

Отже, проведені експерименти підтвердили, що пропонована система прогнозування якості ПЗКС надає можливість порівняння специфікацій вимог до ПЗ, створених, наприклад, різними розробниками для однієї і тієї ж задачі. Вона дозволяє обґрунтовано вибирати набори вимог для подальшої реалізації якісного програмного забезпечення комп'ютерних систем і відхиляти або відправляти на доопрацювання специфікації вимог, за якими розробка якісного ПЗКС є неможливою.

Крім цього, на відміну від існуючих засобів оцінювання якості ПЗКС, які часто вимагають участі людини на всіх етапах, пропонована система передбачає автоматизовану обробку інформації (зокрема, автоматизований пошук значень атрибутів у вимогах, автоматизовані розрахунки значення якості, автоматизований висновок про рівень якості ПЗКС тощо) та дозволяє мінімізувати суб'єктивний вплив людини та її участь у процесах обробки інформації, що є суттєвою перевагою пропонованої системи.

4.3.2. Результати функціонування системи прогнозування рівня безпеки програмного забезпечення комп'ютерних систем [67]

Для *експерименту 1* опрацюємо з використанням системи прогнозування рівня безпеки програмного забезпечення комп'ютерних систем 5 специфікацій (наборів) вимог, які розроблялись декількома ІТ-фірмами м. Хмельницького (Україна) для розв'язання однієї й тієї ж задачі на замовлення ТОВ «Деймос» (м. Хмельницький, Україна).

Користувач скористався модулем автоматичного аналізу вимог до ПЗ на предмет пошуку значень атрибутів якості, який провів розбір 5 аналізованих специфікацій (наборів) вимог та вибрав значення 15 атрибутів якості, які впливають на безпеку ПЗКС.

Значення атрибутів якості з кожного набору вимог були збережені в секції даних бази знань та передані до блоку препроцесінгу значень атрибутів якості з вимог до ПЗ та формування вхідних векторів ШНМ, який сформував 5 наборів

вхідних векторів для ШНМ. Такі набори вхідних векторів для ШНМ були послідовно подані на нейрони вхідного шару штучної нейронної мережі для прогнозування безпеки ПЗКС (як характеристики якості) на основі атрибутів якості. Результати функціонування ШНМ були опрацьовані блоком опрацювання прогнозованої оцінки безпеки ПЗКС та виведені користувачу за допомогою блоку виведення прогнозованої оцінки безпеки ПЗКС (як характеристики якості) у вигляді:

– для специфікації (набору) вимог №1 безпека ПЗКС становить 0.14;
– для специфікації (набору) вимог №2 безпека ПЗКС становить 0.27;
– для специфікації (набору) вимог №3 безпека ПЗКС становить 0.39;
– для специфікації (набору) вимог №4 безпека ПЗКС становить 0.91;
– для специфікації (набору) вимог №5 безпека ПЗКС становить 0.78.

Прогнозовані оцінки безпеки ПЗКС були передані на блок формування висновку про рівень безпеки майбутнього ПЗ, результати роботи якого були виведені користувачу за допомогою блоку виведення висновку про рівень безпеки майбутнього ПЗ у наступному вигляді:

– для специфікації (набору) вимог №1 майбутнє ПЗКС прогнозовано матиме початковий рівень безпеки;
– для специфікації (набору) вимог №2 майбутнє ПЗКС прогнозовано матиме середній рівень безпеки;
– для специфікації (набору) вимог №3 майбутнє ПЗКС прогнозовано матиме середній рівень безпеки;
– для специфікації (набору) вимог №4 майбутнє ПЗКС прогнозовано матиме високий рівень безпеки;
– для специфікації (набору) вимог №5 майбутнє ПЗКС прогнозовано матиме достатній рівень безпеки.

Отже, майбутнє ПЗКС, реалізоване за специфікацією (набором) вимог №4, прогнозовано матиме високий рівень безпеки, тому система прогнозування рівня безпеки програмного забезпечення комп’ютерних систем рекомендує обрати для подальшої роботи саме таку специфікацію (набір) вимог. Якщо з якихось причин (наприклад, висока вартість розробки) замовник не може чи не хоче виконувати

замовлення ПЗКС за специфікацією №4, то як альтернативний варіант системою прогнозування рівня безпеки ПЗКС може бути запропонована специфікація (набір) вимог №5, який прогнозовано матиме достатній рівень безпеки. Інші 3 аналізовані специфікації (набори) вимог (якщо є потреба у їх використанні) потребують доопрацювання та повторного аналізу з використанням розробленої системи прогнозування рівня безпеки програмного забезпечення комп'ютерних систем.

Для *експерименту 2* опрацюємо з використанням системи прогнозування рівня безпеки програмного забезпечення комп'ютерних систем 7 специфікацій (наборів) вимог, які розроблялись декількома ІТ-фірмами м. Хмельницького (Україна) для розв'язання однієї й тієї ж задачі. Користувач скористався модулем автоматичного аналізу вимог до ПЗ на предмет пошуку значень атрибутів якості, який провів розбір всіх 7 аналізованих специфікацій (наборів) вимог та вибрав з кожної з них значення 15 атрибутів якості, які впливають на безпеку ПЗКС.

Значення атрибутів якості з кожного набору вимог були збережені в секції даних бази знань та передані до блоку препроцесінгу значень атрибутів якості з вимог до ПЗ та формування вхідних векторів ШНМ, який сформував 7 наборів вхідних векторів для ШНМ. Такі набори вхідних векторів для ШНМ були послідовно подані на нейрони вхідного шару штучної нейронної мережі для прогнозування безпеки ПЗКС (як характеристики якості) на основі атрибутів якості. Результати функціонування ШНМ були опрацьовані блоком опрацювання прогнозованої оцінки безпеки ПЗКС та виведені користувачу за допомогою блоку виведення прогнозованої оцінки безпеки ПЗКС (як характеристики якості) у вигляді:

- для специфікації (набору) вимог №1 безпека ПЗКС становить 0.84;
- для специфікації (набору) вимог №2 безпека ПЗКС становить 0.34;
- для специфікації (набору) вимог №3 безпека ПЗКС становить 0.48;
- для специфікації (набору) вимог №4 безпека ПЗКС становить 0.65;
- для специфікації (набору) вимог №5 безпека ПЗКС становить 0.12;
- для специфікації (набору) вимог №6 безпека ПЗКС становить 0.81;
- для специфікації (набору) вимог №7 безпека ПЗКС становить 0.95.

Прогнозовані оцінки безпеки ПЗКС були передані на блок формування висновку про рівень безпеки майбутнього ПЗ, результати роботи якого були виведені користувачу за допомогою блоку виведення висновку про рівень безпеки майбутнього ПЗ у наступному вигляді:

– для специфікації (набору) вимог №1 майбутнє ПЗКС прогнозовано матиме достатній рівень безпеки;

– для специфікації (набору) вимог №2 майбутнє ПЗКС прогнозовано матиме середній рівень безпеки;

– для специфікації (набору) вимог №3 майбутнє ПЗКС прогнозовано матиме середній рівень безпеки;

– для специфікації (набору) вимог №4 майбутнє ПЗКС прогнозовано матиме достатній рівень безпеки;

– для специфікації (набору) вимог №5 майбутнє ПЗКС прогнозовано матиме низький рівень безпеки;

– для специфікації (набору) вимог №6 майбутнє ПЗКС прогнозовано матиме достатній рівень безпеки;

– для специфікації (набору) вимог №7 майбутнє ПЗКС прогнозовано матиме високий рівень безпеки.

Отже, майбутнє ПЗКС, реалізоване за специфікацією (набором) вимог №7, прогнозовано матиме високий рівень безпеки, тому система прогнозування рівня безпеки програмного забезпечення комп’ютерних систем рекомендує обрати для подальшої роботи саме таку специфікацію (набір) вимог. Якщо з якихось причин (наприклад, висока вартість розробки) замовник не може чи не хоче виконувати замовлення ПЗКС за специфікацією №7, то як альтернативний варіант системою прогнозування рівня безпеки ПЗКС можуть бути запропоновані специфікації (набори) вимог №1, №6, №4, які прогнозовано матимуть достатній рівень безпеки (причому саме в такому порядку – за спаданням значення безпеки ПЗКС). Інші 3 аналізовані специфікації (набори) вимог (якщо є потреба у їх використанні) потребують доопрацювання та повторного аналізу з використанням розробленої системи прогнозування рівня безпеки програмного забезпечення КС.

Для *експерименту 3* опрацюємо з використанням системи прогнозування рівня безпеки програмного забезпечення комп'ютерних систем 3 специфікації (набори) вимог, які розроблялись декількома ІТ-фірмами м. Хмельницького (Україна) для розв'язання однієї й тієї ж задачі.

Користувач скористався модулем автоматичного аналізу вимог до ПЗ на предмет пошуку значень атрибутів якості, який провів розбір всіх 3 аналізованих специфікацій (наборів) вимог та вибрав з кожної з них значення 15 атрибутів якості, які впливають на безпеку ПЗКС.

Значення атрибутів якості з кожного набору вимог були збережені в секції даних бази знань та передані до блоку препроцесінгу значень атрибутів якості з вимог до ПЗ та формування вхідних векторів ШНМ, який сформував 3 набори вхідних векторів для ШНМ. Такі набори вхідних векторів для ШНМ були послідовно подані на нейрони вхідного шару штучної нейронної мережі для прогнозування безпеки ПЗКС (як характеристики якості) на основі атрибутів якості. Результати функціонування ШНМ були опрацьовані блоком опрацювання прогнозованої оцінки безпеки ПЗКС та виведені користувачу за допомогою блоку виведення прогнозованої оцінки безпеки ПЗКС (як характеристики якості) у вигляді:

– для специфікації (набору) вимог №1 безпека ПЗКС становить 0.20;

– для специфікації (набору) вимог №2 безпека ПЗКС становить 0.14;

– для специфікації (набору) вимог №3 безпека ПЗКС становить 0.90.

Прогнозовані оцінки безпеки ПЗКС були передані на блок формування висновку про рівень безпеки майбутнього ПЗ, результати роботи якого були виведені користувачу за допомогою блоку виведення висновку про рівень безпеки майбутнього ПЗ у наступному вигляді:

– для специфікації (набору) вимог №1 майбутнє ПЗКС прогнозовано матиме низький рівень безпеки;

– для специфікації (набору) вимог №2 майбутнє ПЗКС прогнозовано матиме низький рівень безпеки;

– для специфікації (набору) вимог №3 майбутнє ПЗКС прогнозовано матиме високий рівень безпеки.

Отже, майбутнє ПЗКС, реалізоване за специфікацією (набором) вимог №3, прогнозовано матиме високий рівень безпеки, тому система прогнозування рівня безпеки програмного забезпечення комп'ютерних систем рекомендує обрати для подальшої роботи саме таку специфікацію (набір) вимог. Інші 2 аналізовані специфікації (набори) вимог потребують доопрацювання та повторного аналізу з використанням розробленої системи прогнозування рівня безпеки ПЗКС.

Результати проведених трьох експериментів відображені в таблиці 4.2.

Таблиця 4.2

Результати функціонування системи прогнозування рівня безпеки програмного забезпечення комп'ютерних систем (3 експерименти)

		Значення безпеки ПЗКС (як характеристики якості)	**Висновок про рівень безпеки майбутнього ПЗКС**
Експеримент 1	Специфікація (набір) вимог №1	0.14	Початковий
	Специфікація (набір) вимог №2	0.27	Середній
	Специфікація (набір) вимог №3	0.39	Середній
	Специфікація (набір) вимог №4	0.91	Високий
	Специфікація (набір) вимог №5	0.78	Достатній
Експеримент 2	Специфікація (набір) вимог №1	0.84	Достатній
	Специфікація (набір) вимог №2	0.34	Середній

	Специфікація (набір) вимог №3	0.48	Середній
	Специфікація (набір) вимог №4	0.65	Достатній
	Специфікація (набір) вимог №5	0.12	Низький
	Специфікація (набір) вимог №6	0.81	Достатній
	Специфікація (набір) вимог №7	0.95	Високий
Експеримент 3	Специфікація (набір) вимог №1	0.20	Низький
	Специфікація (набір) вимог №2	0.14	Низький
	Специфікація (набір) вимог №3	0.90	Високий

Отже, як було доведено експериментально, запропонована система прогнозування рівня безпеки програмного забезпечення комп'ютерних систем забезпечує аналіз вимог, на основі якого надає користувачу прогнозовану оцінку безпеки ПЗКС (як характеристики якості) та висновок про рівень безпеки майбутнього ПЗКС, на основі якого можна виконати порівняння специфікацій вимог до ПЗ та обґрунтований вибір специфікації вимог для подальшої реалізації.

При подальшому використанні розробленої системи прогнозування рівня безпеки програмного забезпечення комп'ютерних систем було розглянуто ще 150 різних специфікацій (наборів) вимог, розроблених ІТ-фірмами м. Хмельницького (Україна) для розв'язання різних задач.

Для 41 специфікації (набору) вимог з наявної множини зі 150 прогнозів системою було сформовано висновок про високий рівень безпеки майбутнього ПЗКС; для 29 наборів було сформовано висновок про достатній рівень безпеки

майбутнього ПЗКС; для 55 наборів було сформовано висновок про середній рівень безпеки майбутнього ПЗКС; для 25 наборів було сформовано висновок про низький рівень безпеки майбутнього ПЗКС, отже, тільки за 27% наборами аналізованих вимог потенційно може бути реалізоване ПЗКС з високим рівнем безпеки (рис. 4.10).

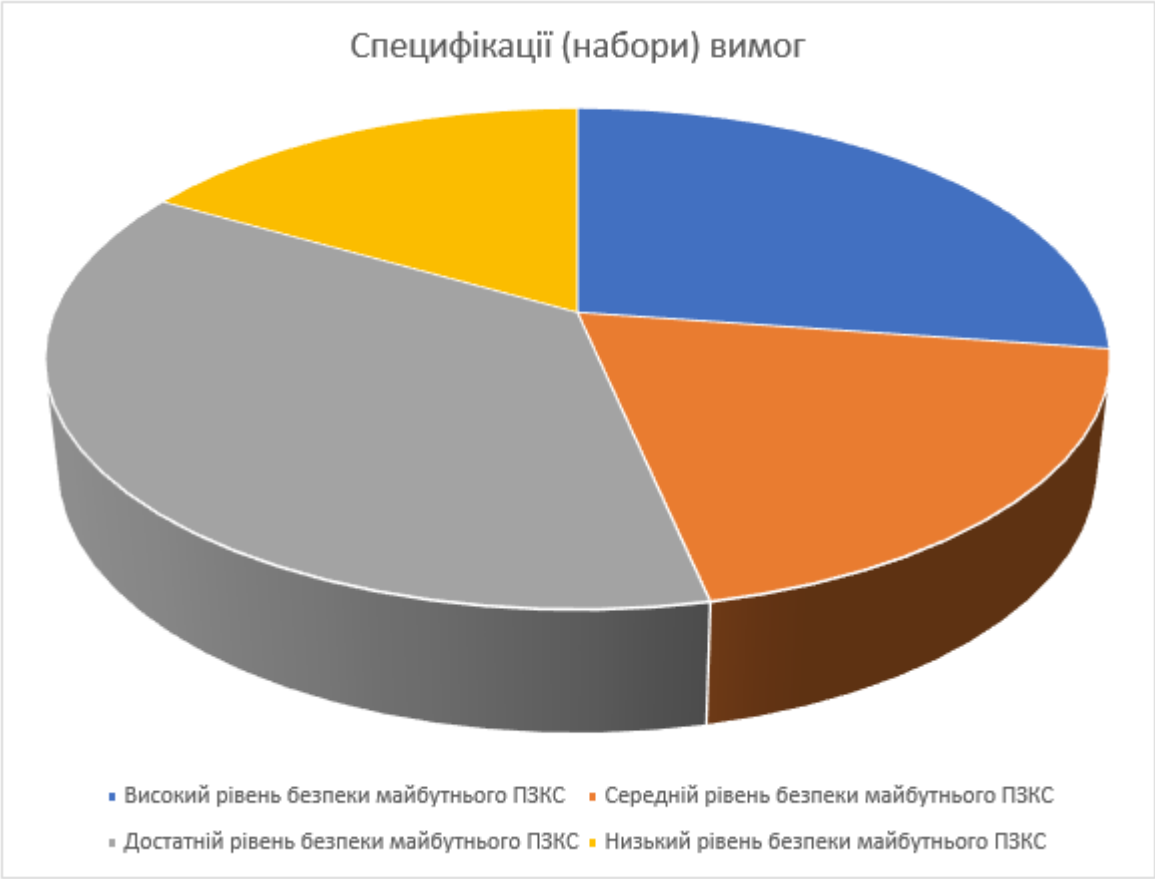


Рис. 4.10 – Розподіл множини прогнозів щодо безпеки майбутнього ПЗКС для проаналізованих 150 специфікацій (наборів) вимог

Як показав аналіз наявної множини прогнозів, ухвалених розробленою системою, лише за 27% наборами аналізованих вимог потенційно може бути реалізоване ПЗКС з високим рівнем безпеки та за 19% наборами аналізованих вимог потенційно може бути реалізоване ПЗКС з достатнім рівнем безпеки, в той час як за 37% наборами аналізованих вимог потенційно може бути реалізоване ПЗКС лише з середнім рівнем безпеки, за 17% наборами аналізованих вимог може бути реалізоване ПЗКС з низьким рівнем безпеки.

4.3.3. Результати функціонування системи ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем [63, 65]

Для *експерименту 1* опрацюємо з використанням системи ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем 15 різних функційних можливостей ПЗКС, розробленого на замовлення ТОВ «Деймос» (м. Хмельницький, Україна), які потенційно можуть бути вразливостями ПЗ.

За допомогою системи ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем проведено опитування користувача з використанням розробленого у підрозділі 3.2 опитувальника для збору інформації про вразливість(і), який міститься у базі знань системи. Відповіді користувача були проаналізовані із використанням розроблених у підрозділі 3.2 правил класифікації вразливостей, які також містяться у базі знань системи. Внаслідок проведеного аналізу було заповнено перший рядок матриці *sv*, а також виконано аналіз значень елементів першого рядка матриці *sv*, завдяки чому було визначено, чи є функційна можливість вразливістю, і якщо є, то який тип вона має (таблиця 4.3).

Таблиця 4.3

Результати функціонування системи ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем (експеримент 1)

№ функційної можливості	**Перший рядок матриці *sv***				**Висновок**
1	1	0	1	1	Функційна можливість є вразливістю коректної

					роботи, конфіденційності та доступності інформації
2	0	0	0	0	Функційна можливість не є вразливістю
3	1	0	0	0	Функційна можливість є вразливістю коректної роботи
4	0	0	1	1	Функційна можливість є вразливістю конфіденційності та доступності інформації
5	0	1	0	0	Функційна можливість є вразливістю цілісності інформації
6	0	0	0	0	Функційна можливість не є вразливістю
7	0	0	0	0	Функційна можливість не є вразливістю

8	0	0	0	0	Функційна можливість не є вразливістю
9	1	1	0	0	Функційна можливість є вразливістю коректної роботи та цілісності інформації
10	0	1	1	1	Функційна можливість є вразливістю цілісності, конфіденційності та доступності інформації
11	0	0	0	0	Функційна можливість не є вразливістю
12	0	0	0	1	Функційна можливість є вразливістю доступності інформації
13	0	0	0	0	Функційна можливість не є вразливістю
14	0	0	0	0	Функційна можливість не є вразливістю

15	0	0	0	0	Функційна можливість не є вразливістю

Для *експерименту 2* опрацюємо з використанням системи ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем 4 припинення функціонування ПЗКС, розробленого на замовлення ТОВ «Деймос» (м. Хмельницький, Україна).

За допомогою системи ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем проведено опитування користувача з використанням розробленого у підрозділі 3.2 опитувальника для збору інформації про відмову(и), який міститься у базі знань системи. Відповіді користувача були проаналізовані із використанням розроблених у підрозділі 3.2 правил класифікації відмов, які також містяться у базі знань системи. Внаслідок проведеного аналізу було сформовано значення змінної *sf*, а також виконано аналіз значень змінної *sf*, завдяки чому було визначено, чи є припинення функціонування ПЗКС відмовою, і якщо так, то який тип вона має (таблиця 4.4).

Таблиця 4.4

Результати функціонування системи ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем (експеримент 2)

№ припинення функціонування	**Змінна *sf***	**Висновок**
1	1	Відмова ПЗ є неістотною
2	1	Відмова ПЗ є неістотною
3	2	Відмова ПЗ є істотною
4	1	Відмова ПЗ є неістотною

Отже, як було доведено експериментально, розроблена система ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем забезпечує висновок щодо того, чи відбувалась відмова, і, якщо відмова відбулась, то користувачу видається її тип. Крім цього, розроблена система ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем забезпечує висновок щодо того, чи є функційна можливість вразливістю, і, якщо функційна можливість є вразливістю, то користувачу видається її тип.

4.4. Висновки

У даному розділі розроблено систему прогнозування рівня якості програмного забезпечення комп'ютерних систем, яка забезпечує аналіз вимог, на основі якого надає користувачу прогнозовані оцінки восьми характеристик якості ПЗКС, геометричну інтерпретацію значень характеристик якості ПЗ, комплексний показник прогнозованої якості ПЗКС та висновок про рівень якості майбутнього ПЗКС, що дозволяє виконати порівняння специфікацій вимог до ПЗ та обґрунтований вибір специфікації вимог для подальшої реалізації.

Крім цього, була розроблена система для прогнозування рівня безпеки програмного забезпечення комп'ютерних систем. Вона проводить аналіз вимог та надає користувачеві прогнозовану оцінку безпеки ПЗКС та висновок щодо рівня безпеки майбутнього ПЗКС, який дозволяє порівняти специфікації вимог до ПЗ та обґрунтовано вибрати вимоги для подальшої реалізації ПЗ з високим або принаймні достатнім рівнем безпеки.

Експерименти підтвердили, що розроблені системи прогнозування рівня якості ПЗКС та рівня безпеки ПЗКС забезпечують можливість порівняння специфікацій (наборів) вимог, що були створені, наприклад, різними розробниками для вирішення однієї й тієї ж задачі, а також обґрунтований вибір вимог для подальшої реалізації ПЗКС з високим рівнем якості та безпеки, а також виявлення

та відправлення на доопрацювання невдалих специфікацій, за якими неможливо розробити якісне та безпечне ПЗКС.

На відміну від інших методів оцінки якості та безпеки ПЗКС, які потребують участі людини на всіх етапах обробки інформації, розроблені системи передбачають автоматизацію процесу обробки даних та мінімізацію участі людини у процесах опрацювання інформації, що є їх *суттєвою перевагою*. Крім цього, перевагою розроблених систем є їх незалежність від мови програмування, якою розроблятиметься майбутнє ПЗКС.

Економічним ефектом від використання розроблених систем є збереження часу та бюджету програмних проєктів за рахунок вибору тих специфікацій вимог, які забезпечують створення якісного та безпечного програмного забезпечення комп'ютерних систем. Це досягається завдяки можливості прогнозування рівня якості та безпеки майбутнього ПЗКС та вибору відповідних специфікацій вимог.

Обмеженням розроблених систем є врахування характеристик якості ПЗКС, які регламентуються стандартом ISO 25010, та атрибутів якості ПЗ, які регламентуються стандартом ISO 25023. Крім цього, обмеженням розроблених систем є необхідність наявності специфікацій вимог до програмного забезпечення комп'ютерних систем.

У розділі розроблено також систему ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем. Ця система дозволяє визначити, чи відбулась відмова, надаючи інформацію про тип відмови, якщо вона відбулась, і дозволяє визначити, чи є певна функційна можливість вразливістю, надаючи інформацію про тип вразливості, якщо було ідентифіковано вразливість.

ВИСНОВКИ

У дисертації розв’яується актуальна науково-прикладна задача прогнозування і оцінювання рівня якості та безпеки програмного забезпечення комп’ютерних систем на основі атрибутів якості на початкових етапах життєвого циклу програмних проєктів шляхом розроблення методів і засобів прогнозування рівня якості та безпеки ПЗКС.

У роботі отримано такі наукові та практичні результати:

1. Проведений аналіз існуючих моделей, методів та інструментів прогнозування якості та безпеки ПЗКС виявив, що досліджені методи та засоби прогнозування безпеки та якості ПЗКС мають великий потенціал для розв’язання різних задач, можуть бути використані в різних контекстах, проте вони не забезпечують обчислення і не задають залежності значень характеристик якості від значень атрибутів, не забезпечують обчислення і не задають залежності значення якості від значень характеристик якості та не забезпечують прогнозування рівня якості та/або безпеки ПЗКС на основі отриманих кількісних значень якості та/або безпеки. Отже, наразі існує суперечність між зростаючою відповідальністю, яка покладається на програмне забезпечення комп’ютерних систем (ПЗКС), та розширенням вимог до якості ПЗКС, з одного боку, і недосконалістю методів та засобів прогнозування якості та безпеки ПЗКС, особливо на ранніх етапах життєвого циклу, з іншого боку.

2. У дисертаційній роботі був розроблений метод пошуку значень атрибутів якості у вимогах до програмного забезпечення комп’ютерних систем, який відрізняється від відомих структуруванням вимог за атрибутами якості, та забезпечує вибір значень атрибутів якості ПЗ з природомовної специфікації вимог до ПЗ, які використовуються для оцінювання значень характеристик якості ПЗ та для комплексного оцінювання якості ПЗ; розроблений метод дозволяє автоматизувати опрацювання вимог та мінімізувати участь людини у процесах оцінювання якості та безпеки ПЗКС.

3. Розроблено метод прогнозування рівня якості ПЗКС на основі атрибутів якості, який, на відміну від відомих, забезпечує можливість прогнозування рівня якості розроблюваного програмного забезпечення комп'ютерних систем на основі обробки атрибутів якості, наявних у вимогах до ПЗ. Розроблений метод задає залежності значення якості від значень атрибутів, забезпечує розрахунок кількісного значення якості на основі значень атрибутів, а також забезпечує прогнозування рівня якості ПЗКС на основі отриманого кількісного значення. Таким чином, запропонований метод дозволяє порівнювати специфікації вимог до ПЗ, одразу відмовлятись від реалізації ПЗКС на основі невдалих специфікацій (економія коштів та часу, зменшення ймовірності провальних і проблемних проєктів) та виконувати обґрунтований вибір специфікації для подальшої реалізації ПЗКС саме високої якості (звісно, за умови, що помилки не будуть внесені на наступних етапах життєвого циклу ПЗ).

4. Розроблено метод прогнозування рівня безпеки програмного забезпечення комп'ютерних систем, який, на відміну від відомих, встановлює залежність безпеки ПЗКС від атрибутів якості та формує прогнозоване числове значення безпеки ПЗКС на основі атрибутів, і забезпечує прогнозування рівня безпеки ПЗКС на основі отриманого числового значення, а також забезпечує порівняння специфікацій вимог до ПЗ за прогнозованим рівнем безпеки розроблюваного ПЗКС та можливість відбраковування невдалих специфікацій.

5. Розроблено метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем, який забезпечує висновок щодо того, чи відбувалась відмова, і, якщо відмова відбулась, то користувачу видається її тип. Крім цього, розроблений метод ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп'ютерних систем забезпечує висновок щодо того, чи є функційна можливість вразливістю, і, якщо функційна можливість є вразливістю, то користувачу видається її тип. Таким чином, розроблений метод забезпечує безпеку ПЗКС.

6. У дисертації розроблено систему прогнозування рівня якості програмного забезпечення комп'ютерних систем, яка забезпечує аналіз вимог, на основі якого

надає користувачу прогнозовані оцінки восьми характеристик якості ПЗКС, геометричну інтерпретацію значень характеристик якості ПЗ, комплексний показник прогнозованої якості ПЗКС та висновок про рівень якості майбутнього ПЗКС, що дозволяє виконати порівняння специфікацій вимог до ПЗ та обґрунтований вибір специфікації вимог для подальшої реалізації. Крім цього, розроблено систему прогнозування рівня безпеки ПЗКС, яка надає користувачу прогнозовану оцінку безпеки ПЗКС (як характеристики якості) та висновок про рівень безпеки майбутнього ПЗКС, забезпечує можливість порівняння специфікацій вимог та обґрунтований вибір специфікації вимог для наступної реалізації ПЗКС з високим рівнем безпеки.

На відміну від відомих засобів для галузі оцінювання якості та безпеки ПЗКС, які передбачають участь людини та інтерпретацію інформації людиною практично на всіх етапах, що може призводити до втрат інформації, пропоновані системи передбачають автоматизацію обробки інформації та мінімізацію або повне усунення участі людини з процесів опрацювання інформації, що є їх суттєвою перевагою. Економічним ефектом від використання розроблених систем є можливість економії часу та бюджету програмних проєктів за рахунок вибору (на основі прогнозованого рівня якості та безпеки майбутнього ПЗКС) для реалізації тих специфікацій (наборів) вимог, які забезпечать можливість побудови якісного та безпечного ПЗКС. Обмеженням розроблених систем є врахування характеристик якості ПЗКС, які регламентуються стандартом ISO 25010, та атрибутів якості ПЗ, які регламентуються стандартом ISO 25023.

7. У дисертації розроблено також систему ідентифікації та класифікації відмов і вразливостей програмного забезпечення комп’ютерних систем, яка забезпечує висновок щодо того, чи відбувалась відмова, і, якщо відмова відбулась, то користувачу видається її тип, та чи є аналізована функційна можливість вразливістю, і, якщо функційна можливість є вразливістю, то користувачу видається її тип.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Mahaju S., Carver J. C., Bradshaw G. L. Human error management in requirements engineering: Should we fix the people, the processes, or the environment? *Information and Software Technology*. 2023. P. 107223.

2. ISO/IEC 25012-based methodology for managing data quality requirements in the development of information systems: Data quality by design / C. Guerra-García et al. *Data & Knowledge Engineering*. 2023. P. 102152.

3. ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models. [Introduced 01.03.2011]. Geneva (Switzerland), 2011. 34 p. (International standard).

4. An Approach towards Missing Data Recovery within IoT Smart System / I. Izonin et al. *Procedia Computer Science*. 2019. Vol. 155. P. 11–18.

5. Effective Software Effort Estimation enabling Digital Transformation / A. Jadhav et al. *IEEE Access*. 2023. P. 1.

6. Huang F., Strigini L. HEDF: A Method for Early Forecasting Software Defects based on Human Error Mechanisms. *IEEE Access*. 2023. P. 1.

7. Requirements engineering framework for human-centered artificial intelligence software systems / K. Ahmad et al. *Applied Soft Computing*. 2023. P. 110455.

8. Element quality indicator: A quality assessment and defect detection method for software requirement specification / Q. Zhi et al. *Heliyon*. 2023. Vol. 9, no. 5. P. e16469.

9. Transient in the Software Systems. Untraditional Approach to Software Reliability / D. Maevsky et al. *2019 International Conference on Information Technologies (InfoTech)*, St. St. Constantine and Elena resort (near the city of Varna), Bulgaria, 19–20 September 2019. 2019.

10. Information Technology for Evaluating the Computer Energy Consumption at the Stage of Software Development / E. D. Stetsuyk et al. *Green IT Engineering: Social, Business and Industrial Applications*. Cham, 2018. P. 21–40.

11. Employers' requirements-oriented assessment of IoT curriculum: The projects CABRIOLET and ALIOT / V. Kharchenko et al. *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kiev, 24–27 May 2018. 2018.

12. Relationship between factors influencing the software development process and software defects / O. Gordieiev et al. *2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Athens, Greece, 13–15 October 2023. 2023.

13. Gordieiev O., Gordieieva D., Rainer A. Software Quality Assessment: Defect Life Cycle, Software Defect Profile, Its Types and Misalignments. *Software Quality: Higher Software Quality through Zero Waste Development*. Cham, 2023. P. 109–120.

14. Gordieiev O., Kharchenko V., Gordieieva D. Software Requirements Profile Quality Model. *International Journal of Computing*. 2022. Vol. 21. Issue 1. P. 111 – 119.

15. Requirements to Products and Processes for Software of Safety Important NPP I&C Systems / V. Sklyar et al. *Research Anthology on Agile Software, Software Development, and Testing*. 2022. P. 212–246.

16. Requirements to Products and Processes for Software of Safety Important NPP I&C Systems / V. Sklyar et al. *Cyber Security and Safety of Nuclear Power Plant Instrumentation and Control Systems*. 2020. P. 97–131.

17. Sholomii Y., Yakovyna V. Quality Assessment and Assurance of Machine Learning Systems: A Comprehensive Approach. *Information and Communication Technologies in Education, Research, and Industrial Applications*. Cham, 2023. P. 265–275.

18. Yakovyna V., Shakhovska N. Software failure time series prediction with RBF, GRNN, and LSTM neural networks. *Procedia Computer Science*. 2022. Vol. 207. P. 837–847.

19. Shakhovska N., Yakovyna V. Feature Selection and Software Defect Prediction by Different Ensemble Classifiers. *Lecture Notes in Computer Science*. Cham, 2021. P. 307–313.

20. Yakovyna V., Seniv M., Symets I. The Relation between Software Development Methodologies and Factors Affecting Software Reliability. *2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT)*, Zbarazh, Ukraine, 23–26 September 2020. 2020.

21. System-Information and Cognitive Technologies of Man-Made Infrastructure Cyber Security / L. S. Sikora et al. *Journal of Cyber Security and Mobility*. 2023.

22. Sikora L., Lysa N., Fedevych O., Fedyna B. Infrastructure Cybersecurity under Complex Man-Made Threats Conditions. *CEUR-WS*. 2022. Vol. 3422. Pp. 1-13.

23. Sabat V., Sikora L., Durnyak B., Fedevych O., Lysa N. Information Technologies of Active Control of Complex Hierarchical Systems under Threats and Information Attacks. *CEUR-WS*. 2022. Vol. 3156. Pp. 305-318.

24. Senkivskyy V., Pikh I., Babichev S., Kudriashova A., Senkivska N. Modeling of alternatives and defining the best options for websites design. *CEUR-WS*. 2021. Vol. 2853. Pp. 259-270.

25. Research of quality factors of software testing / A. V. Kudriashova et al. *Scientific Papers (Ukrainian Academy of Printing)*. 2020. Vol. 2, no. 61. P. 11–18.

26. Methodological principles of software quality formation (part 2: optimization of software quality factors model) / V. M. Senkivskyy et al. *Printing and Publishing*. 2023. Vol. 1, no. 85. P. 11–21.

27. Кудряшова А. Модель пріоритетного впливу факторів на якість післядрукарських процесів. *Measuring and computing devices in technological processes*. 2023. № 1. С. 187–192.

28. Application of Formal Verification Methods in a Safety-Oriented Software Development Life Cycle / O. Odarushchenko et al. *2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Athens, Greece, 13–15 October 2023. 2023.

29. Odarushchenko O., Odarushchenko E., Kopishynska O., Rudenko O., Gorbenko A. Improving the Accuracy of Software Reliability Modeling by Predicting the Number of Secondary Software Defects. *CEUR-WS*. 2022. Vol. 3156. Pp. 198-207.

30. Illiashenko O., Kharchenko V., Odarushchenko O. Towards Evidence-Based Cybersecurity Assessment of Programmable Systems to Ensure the Protection of Critical IT Infrastructure. *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Dortmund, Germany, 7–9 September 2023. 2023.

31. Software Fault Insertion Testing for SIL Certification of Safety PLC-Based System / O. Odarushchenko et al. *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kyiv, Ukraine, 14–18 May 2020. 2020.

32. Yatsyshyn V., Pastukh O., Lutskiv A., Tsymbalistyy V., Martsenko N. A Risks management method based on the quality requirements communication method in agile approaches. *CEUR-WS*. 2022. Vol. 3309. Pp. 1-10.

33. Pryimak M. Periodic functions with variable period – basic concepts and certain investigation results. *Scientific journal of the Ternopil national technical university*. 2024. Vol. 1, no. 113. P. 46–57.

34. Програмний засіб вимірювання метрик продуктивності реляційних систем керування базами даних / В. В. Яцишин та ін. *Математичне моделювання*. 2023. № 1(48). С. 7–17.

35. Padhy N., Singh R. P., Satapathy S. C. Enhanced evolutionary computing based artificial intelligence model for web-solutions software reusability estimation. *Cluster Computing*. 2017. Vol. 22, S4. P. 9787–9804.

36. Padhy N., Singh R. P., Satapathy S. C. Cost-effective and fault-resilient reusability prediction model by using adaptive genetic algorithm based neural network for web-of-service applications. *Cluster Computing*. 2018. Vol. 22, S6. P. 14559–14581.

37. Patnaik A., Padhy N. A Hybrid Approach to Enhance Software Quality by Sentiment Analysis of Developer. *Communications in Computer and Information Science*. Cham, 2024. P. 113–125.

38. Patnaik A., Padhy N. Sentiment Analysis of Software Project Code Commits. *Lecture Notes in Networks and Systems*. Singapore, 2022. P. 79–88.

39. Arora I., Saha A. Software fault prediction using firefly algorithm. International *Journal of Intelligent Engineering Informatics*. 2018. Vol. 6, no. 3/4. P. 356.

40. Arora I., Saha A. Software Defect Prediction: A Comparison Between Artificial Neural Network and Support Vector Machine. *Advanced Computing and Communication Technologies*. Singapore, 2017. P. 51–61.

41. Goyal S. Comparison of Machine Learning Techniques for Software Quality Prediction. *International Journal of Knowledge And Systems Science*. 2020. Vol. 11. Issue 2. P. 20-40.

42. Goyal S. 3PcGE: 3-parent child-based genetic evolution for software defect prediction. *Innovations in Systems and Software Engineering*. 2022.

43. Goyal S., Gupta A., Jha H. Current Trends in Methodology for Software Development Process. *Communication, Software and Networks*. Singapore, 2022. P. 621–629.

44. Goyal S. Software fault prediction using evolving populations with mathematical diversification. *Soft Computing*. 2022.

45. Goyal S., Bhatia P. K. Software Quality Prediction Using Machine Learning Techniques. *Advances in Intelligent Systems and Computing*. Singapore, 2020. P. 551–560.

46. Huang S., Chen W.-C., Chiu P.-Y. Evaluation Process Model of the Software Product Quality Levels. *2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, Wuhan, China, 3–4 December 2015. 2015.

47. The interval grey QFD method for new product development: Integrate with LDA topic model to analyze online reviews / S. Huang et al. *Engineering Applications of Artificial Intelligence*. 2022. Vol. 114. P. 105213.

48. Sheoran K., Tomar P., Mishra R. Software Quality Prediction Model with the Aid of Advanced Neural Network with HCS. *Procedia Computer Science*. 2016. Vol. 92. P. 418–424.

49. Tomar P., Mishra R., Sheoran K. Prediction of quality using ANN based on Teaching-Learning Optimization in component-based software systems. *Software: Practice and Experience*. 2018. Vol. 48, no. 4. P. 896–910.

50. Masood M. H., Khan M. J. Early Software Quality Prediction Based on Software Requirements Specification Using Fuzzy Inference System. *Intelligent Computing Methodologies*. Cham, 2018. P. 722–733.

51. Software Defect Prediction Using Artificial Neural Networks: A Systematic Literature Review / M. A. Khan et al. *Scientific Programming*. 2022. Vol. 2022. P. 1–10.

52. Cho S.-Y., Yoo S.-K. A Quality Evaluation Model for Hardware-Control Software. Advanced Science Letters. 2017. Vol. 23, no. 10. P. 9607–9611.

53. Cho S.-Y. A Program Optimization Method for Embedded Software Developed Using Open Sources. *International Journal on Advanced Science, Engineering and Information Technology*. 2018. Vol. 8, no. 4-2. P. 1692.

54. Inoue S., Yamashita N., Yamada S. On statistical models for predicting software quality/reliability: generalized linear and linear mixed modeling. *Life Cycle Reliability and Safety Engineering*. 2017. Vol. 6, no. 1. P. 15–21.

55. Efficiency Evaluation of Software Faults Correction Based on Queuing Simulation / Y. Minamino et al. *Mathematics*. 2022. Vol. 10, no. 9. P. 1438.

56. Deep Learning Approach Based on Fault Correction Time for Reliability Assessment of Cloud and Edge Open Source Software / H. Sone et al. *Springer Series in Reliability Engineering*. Cham, 2022. P. 1–17.

57. Yamada S. Debugging process modeling for quality/reliability assessment of software system. *Systems Performance Modeling*. 2020. P. 13–20.

58. Sadia H., Abbas S. Q., Faisal M. A Bayesian Network-Based Software Requirement Complexity Prediction Model. *Computational Methods and Data Engineering*. Singapore, 2022. P. 197–213.

59. Sadia H., Faisal M. A Systematic Literature Review Of Multi-Criteria Risk Factors (VUCA) In Requirement Engineering. *International Journal of Scientific & Technology Research*. 2019. Vol. 8. Issue 11. P. 13-20.

60. Hovorushchenko T., Medzatyi D., Voichur Yu., Lebiga M. Method for forecasting the level of software quality based on quality attributes. Journal of Intelligent & Fuzzy Systems. 2023. vol. 44, no. 3, pp. 3891-3905.

61. Hovorushchenko T., Voichur Yu., Medzatyi D., Boyarchuk A. Information Technology for Prediction Software Quality Level. Radioelectronic and Computer Systems. 2023. No. 3. Pp. 238-254.

62. E. Zaitseva, T. Hovorushchenko, O. Pavlova, Yu. Voichur. Identifying the Mutual Correlations and Evaluating the Weights of Factors and Consequences of Mobile Applications Insecurity. Systems. 2023. Vol. 11. Issue 5. Article No. 242.

63. Медзатий Д.М., Войчур Ю.О., Войчур О.Ю. Технологія ідентифікації та класифікації відмов і вразливостей програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах. 2023. №1. С. 53-57.

64. Ю. Войчур, Д. Медзатий. Метод аналізу вимог до програмного забезпечення на предмет пошуку значень атрибутів якості. Вимірювальна та обчислювальна техніка в технологічних процесах. 2024. №1. С.146-151.

65. Hovorushchenko T., Popov P., Medzatyi D., Voichur Yu. Method and Technology for Ensuring the Software Security by Identifying and Classifying the Failures and Vulnerabilities. CEUR-WS. 2022. Vol. 3309. Pp. 338-348. (індексована в наукометричній базі Scopus)

66. Hovorushchenko T., Voichur Yu. Method for Determining the Number of Lines of Manually Written Source Code. CEUR-WS. 2023. Vol. 3628. Pp. 520-525. (індексована в наукометричній базі Scopus)

67. T. Hovorushchenko, Yu. Voichur, D. Medzatyi, A. Boyarchuk, A. Hnatchuk. Method for Determining the Security Level of Software. CEUR-WS. 2024. Vol. 3675. Pp. 72-85.

68. А. с. 113734 Україна. Нейромережна модель прогнозування якості програмного забезпечення / Т. О. Говорущенко, М. М. Лебіга, Ю. О. Войчур. 2022.

69. А. с. 118851 Україна. Метод прогнозування рівня якості програмного забезпечення на основі атрибутів якості / Т. О. Говорущенко, М. М. Лебіга, Ю. О. Войчур. 2023.

70. Software. URL: https://www.statista.com/markets/418/topic/484/software/. (Last accessed: April 01, 2024).

71. Tamura Y., Yamada S. Deep Learning Based on Fine Tuning with Application to the Reliability Assessment of Similar Open Source Software. *International Journal of Mathematical, Engineering and Management Sciences*. 2023. Vol. 8, no. 4. P. 632–639.

72. Bhandari K., Kumar K., Sangal A. L. Data quality issues in software fault prediction: a systematic literature review. *Artificial Intelligence Review*. 2022.

73. Assessing the Success of R&D Projects and Innovation Projects through Project Management Life Cycle / M. R. Farokhad et al. *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Metz, France, 18–21 September 2019. 2019.

74. Success Rates Rise: Transforming the high cost of low performance. URL: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf. (Last accessed: April 01, 2024).

75. The Cost of Poor Software Quality in the US: A 2020 Report. URL: https://www.it-cisq.org/cisq-files/pdf/CPSQ-2020-report.pdf. (Last accessed: April 01, 2024).

76. Recent Catastrophic Accidents: Investigating How Software was Responsible / W. E. Wong et al. *2010 Fourth International Conference on Secure Software Integration and Reliability Improvement*, Singapore, Singapore, 9–11 June 2010. 2010.

77. The Additive Input-Doubling Method Based on the SVR with Nonlinear Kernels: Small Data Approach / I. Izonin et al. *Symmetry*. 2021. Vol. 13, no. 4. P. 612.

78. Tricentis Software Fail Watch Finds 3.6 Billion People Affected and $1.7 Trillion Revenue Lost by Software Failures Last Year. URL: https://www.tricentis.com/news/tricentis-software-fail-watch-finds-3-6-billion-people-affected-and-1-7-trillion-revenue-lost-by-software-failures-last-year/. (Last accessed: April 01, 2024).

79. Software Fail Watch Says $1.1 Trillion in Assets Affected by Software Bugs in 2016. URL: https://www.tricentis.com/news/software-fail-watch-says-1-1-trillion-in-assets-affected-by-software-bugs-in-2016. (Last accessed: April 01, 2024).

80. Pulse of the Profession 2023: Power Skills, Redefining Project Success. 14th edition. Available online: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pmi-pulse-of-the-profession-2023-report.pdf?v=7933da8f-304b-4fe3-a655-78dace54174a&rev=427949fcdb684485a020cc72ea219f32. (Last accessed: April 01, 2024).

81. Software Defect Prediction Using Dagging Meta-Learner-Based Classifiers / A. N. Babatunde et al. *Mathematics*. 2023. Vol. 11, no. 12. P. 2714.

82. Cloud-based bug tracking software defects analysis using deep learning / T. Hai et al. *Journal of Cloud Computing*. 2022. Vol. 11, no. 1.

83. Forecasting technical debt evolution in software systems: an empirical study / L. Aversano et al. *Frontiers of Computer Science*. 2022. Vol. 17, no. 3.

84. An Efficient Hybrid Mine Blast Algorithm for Tackling Software Fault Prediction Problem / M. Alweshah et al. *Neural Processing Letters*. 2023.

85. Alghamidi A., Niazi M. Toward Successful Secure Software Deployment: An Empirical Study. *27th International Conference on Evaluation and Assessment in Software Engineering*, Oulu, Finland, 14-16 June 2023. 2023.

86. Standish Group 2015 Chaos Report – Q&A with Jennifer Lynch. Available online: http://www.infoq.com/articles/standish-chaos-2015. (Last accessed: April 01, 2024).

87. Li J., Liu S. Requirements-related fault prevention during the transformation from formal specifications to programs. *IET Software*. 2023.

88. A systematic literature review of requirements engineering education / M. Daun et al. *Requirements Engineering*. 2022.

89. ISO 25023:2016. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). Measurement of system and software product quality. [Introduced 31.03.2016]. Geneva (Switzerland), 2016. 45 p. (International standard).

90. Говорущенко Т.О. Теоретичні та прикладні засади інформаційної технології оцінювання достатності інформації щодо якості у специфікаціях вимог до програмного забезпечення: дис. ... доктора техн. наук: 05.13.06. Львів, 2018. 441 с.

91. Павлова О.О. Агентно-орієнтована інформаційна технологія оцінювання початкових етапів життєвого циклу програмного забезпечення на основі онтологічного підходу: дис. ... доктора філософії: 122. Хмельницький, 2020. 175 с.

92. Ramchand S., Shaikh S., Alam I. Role of Artificial Intelligence in Software Quality Assurance. *Lecture Notes in Networks and Systems*. Cham, 2021. P. 125–136.

93. Bajnaid N., Benlamri R., Pakstas A., Salekzamankhani Sh. An ontological approach to model software quality assurance knowledge domain. *Lecture Notes on Software Engineering*. 2016. Vol. 4. No. 3. Pp. 193-198.

94. Software test quality evaluation based on fuzzy mathematics / T. Sun et al. *Journal of Intelligent & Fuzzy Systems*. 2020. P. 1–11.

95. Lakra K., Chug A. Application of metaheuristic techniques in software quality prediction: a systematic mapping study. *International Journal of Intelligent Engineering Informatics*. 2021. Vol. 9, no. 4. P. 355.

96. Inoue S., Yamada S. Statistical Prediction of Software Quality Based on Generalized Linear Models. *13th International Conference on Industrial Management*, Hirosima, Japan, 21-23 September 2016. 2016.

97. Canaparo M., Ronchieri E. Data Mining Techniques for Software Quality Prediction in Open Source Software. *EPJ Web of Conferences*. 2019. Vol. 214. P. 05007.

98. Radliński Ł. A Framework for Integrated Software Quality Prediction Using Bayesian Nets. *Computational Science and Its Applications - ICCSA 2011*. Berlin, Heidelberg, 2011. P. 310–325.

99. Sethi T., Gagandeep. Improved approach for software defect prediction using artificial neural networks. *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, 7–9 September 2016. 2016.

100. Kaur R., Sharma S. An ANN Based Approach for Software Fault Prediction Using Object Oriented Metrics. *Communications in Computer and Information Science*. Singapore, 2018. P. 341–354.

101. Deep Singh P., Chug A. Software defect prediction analysis using machine learning algorithms. *2017 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence (Confluence)*, Noida, India, 12–13 January 2017. 2017.

102. Jin C., Jin S.-W. Prediction approach of software fault-proneness based on hybrid artificial neural network and quantum particle swarm optimization. *Applied Soft Computing*. 2015. Vol. 35. P. 717–725.

103. Arar Ö. F., Ayan K. Software defect prediction using cost-sensitive neural network. *Applied Soft Computing*. 2015. Vol. 33. P. 263–277.

104. Kumaresan K., Ganeshkumar P. Software reliability modeling using increased failure interval with ANN. *Cluster Computing*. 2018. Vol. 22, S2. P. 3095–3102.

105. Malik V., Singh S. Artificial intelligent environments: Risk management and quality assurance implementation. Journal of Discrete Mathematical Sciences and Cryptography. 2020. Vol. 23, no. 1. P. 187–195.

106. Incorporation of ISO 25010 with machine learning to develop a novel quality in use prediction system (QiUPS) / O. Alshareet et al. *International Journal of System Assurance Engineering and Management*. 2017. Vol. 9, no. 2. P. 344–353.

107. Tripathi V. K., Singh M. An efficient metrics based self-adaptive design model by multiobjective gray wolf optimization with extreme learning machine for autonomic computing system application. *Concurrency and Computation: Practice and Experience*. 2021. Vol. 34, no. 4.

108. Mi Y., Gao E. Information Sharing Security Protection System Based on Artificial Intelligence. *2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, Tumkur, Karnataka, India, 2–3 December 2022. 2022.

109. Hu Y.-T., Wang S.-Y., Wu Y.-M., Zou D.-Q., Li W.-K., Jin H. A Slice-level vulnerability detection and interpretation method based on graph neural network. *Journal of Software*. 2023. Vol. 34. Issue 6. P. 2204 – 2221.

110. Proximal Instance Aggregator networks for explainable security vulnerability detection / Tanwar A. et al. *Future Generation Computer Systems*. 2022.

111. Software Quality Prediction by CatBoostFeed-Forward Neural Network in Software Engineering / D. Sudharson et al. *System Reliability and Security*. New York, 2023. P. 207–218.

112. Mona J., Al-Sagheer R., Alghazali S. Software Quality Assurance Models and Application to Defect Prediction Techniques. *International Journal of Intelligent Systems and Applications in Engineering*. 2023. Vol. 11. Issue 1. P. 169 – 178.

113. Scientific programming using optimized machine learning techniques for software fault prediction to improve software quality / M. Shafiq et al. *IET Software*. 2023.

114. Prediction of software quality with Machine Learning-Based ensemble methods / A. A. Ceran et al. *Materials Today: Proceedings*. 2022.

115. Airlangga G., Liu A. Investigating Software Domain Impact in Requirements Quality Attributes Prediction. *Journal of Information Science and Engineering*. 2022. Vol. 38. Issue 2. P. 295 – 316.

116. Canchari L., Angeleri P., Dávila A. Requirements Validation in the Information System Software Development Lifecycle: A Software Quality in Use Evaluation. *Programming and Computer Software*. 2023. Vol. 49, no. 8. P. 610–624.

117. Desai B., Sungkur R. K. Software Quality Prediction Using Machine Learning. *International Journal of Software Innovation*. 2022. Vol. 10, no. 1. P. 1–35.

118. Liu L., Han P. Application of improved cuckoo algorithm to optimize generalized regression neural network in software quality prediction. *International Conference on Neural Networks, Information, and Communication Engineering (NNICE 2022)*, Qingdao, China, 25–27 March 2022 / ed. by R. Tiwari. 2022.

119. Ritu, Sangwan O. Radial Basis Function Network Based Intelligent Scheme for Software Quality Prediction. *Communications in Computer and Information Science*. 2022. Vol. 1572. P. 327 – 340.

120. An Ensemble Learning Approach for Software Defect Prediction in Developing Quality Software Product / Y. K. Saheed et al. *Communications in Computer and Information Science*. Cham, 2021. P. 317–326.

121. Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability / F. Yucalar et al. *Engineering Science and Technology, an International Journal*. 2020. Vol. 23, no. 4. P. 938–950.

122. Firefly Optimization Technique for Software Quality Prediction / D. Pankwar et al. *Soft Computing: Theories and Applications*. Singapore, 2022. P. 263–273.

123. What is Software Failure. URL: https://www.igi-global.com/dictionary/investigation-of-software-reliability-prediction-using-statistical-and-machine-learning-methods/59093. (Last accessed: April 01, 2024).

124. Howard M., LeBlanc D., Viega J. 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them, McGraw-Hill Education, Redmond, 2010.

125. Yahoo says 500 million accounts stolen. URL: https://money.cnn.com/2016/09/22/technology/yahoo-data-breach. (Last accessed: April 01, 2024).

126. Equifax Made Major Errors That Led to Hack, Ex-CEO Concedes. URL: https://www.bloomberg.com/news/articles/2017-10-02/ex-equifax-ceo-says-human-tech-failures-allowed-breach-to-occur. (Last accessed: April 01, 2024).

127. Facebook Says Breach Affected About 50 Million Accounts. URL: https://www.bloomberg.com/news/articles/2018-09-28/facebook-says-security-breach-affected-about-50-million-accounts. (Last accessed: April 01, 2024).

128. A.G. Underwood Announces Record $148 Million Settlement With Uber Over 2016 Data Breach. URL: https://ag.ny.gov/press-release/2018/ag-underwood-announces-record-148-million-settlement-uber-over-2016-data-breach. (Last accessed: April 01, 2024).

129. It could have been prevented: it became known why government websites "went down". URL: https://www.epravda.com.ua/news/2022/01/14/681448/. (Last accessed: April 01, 2024).

130. CVEdetails.com: The Ultimate Security Vulnerability Data Source. URL: https://www.cvedetails.com/. (Last accessed: April 01, 2024).

131. An effective end-to-end android malware detection method / H. Zhu et al. *Expert Systems with Applications*. 2023. Vol. 218. P. 119593.

132. Keyvanpour M. R., Barani Shirzad M., Heydarian F. Android malware detection applying feature selection techniques and machine learning. *Multimedia Tools and Applications*. 2022.

133. Saraswat P. An inclusive analysis of Google's android operating system and its security. *INSTRUMENTATION ENGINEERING, ELECTRONICS AND TELECOMMUNICATIONS – 2021 (IEET-2021)*: Proceedings of the VII International Forum, Izhevsk, Russian Federation. 2023.

134. A model-based framework for inter-app Vulnerability analysis of Android applications / A. Nirumand et al. *Software: Practice and Experience*. 2022.

135. A Decade in, How Safe Are Your iOS and Android Apps? URL: https://www.nowsecure.com/blog/2018/07/11/a-decade-in-how-safe-are-your-ios-and-android-apps/. (Last accessed: April 01, 2024).

136. Understanding OWASP Mobile Top 10 Risks with Real-world Cases. URL: https://appinventiv.com/blog/owasp-mobile-top-10-real-world-cases/. (Last accessed: April 01, 2024).

137. A Survey on Quantitative Risk Estimation Approaches for Secure and Usable User Authentication on Smartphones / M. Papaioannou et al. *Sensors*. 2023. Vol. 23, no. 6. P. 2979.

138. Byun J. W. Towards serverless fast one round authentication with two mobile end devices. *The Journal of Supercomputing*. 2022.

139. Razian M. R., Sangchi H. M. A threatened-based software security evaluation method. *2014 11th International ISC Conference on Information Security and Cryptology (ISCISC)*, Tehran, Iran, 3–4 September 2014. 2014.

140. Extending the Agile Development Process to Develop Acceptably Secure Software / L. b. Othmane et al. *IEEE Transactions on Dependable and Secure Computing*. 2014. Vol. 11, no. 6. P. 497–509.

141. Erlingsson U. Data-Driven Software Security: Models and Methods. *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, Lisbon, 27 June – 1 July 2016. 2016.

142. Baca D., Petersen K. Prioritizing Countermeasures through the Countermeasure Method for Software Security (CM-Sec). *Product-Focused Software Process Improvement*. Berlin, Heidelberg, 2010. P. 176–190.

143. Randrianasolo A. S., Pyeatt L. D. Q-Learning: From Computer Network Security to Software Security. *2014 13th International Conference on Machine Learning and Applications (ICMLA)*, Detroit, MI, USA, 3–6 December 2014. 2014.

144. Ramachandran M. Software security requirements management as an emerging cloud computing service. *International Journal of Information Management*. 2016. Vol. 36, no. 4. P. 580–590.

145. Shehab Farhan A. R., Mostafa Mostafa G. M. A Methodology for Enhancing Software Security During Development Processes. *2018 21st Saudi Computer Society National Computer Conference (NCC)*, Riyadh, 25–26 April 2018. 2018.

146. Xu B., Lu M., Zhang D. A Layered Argument Strategy for Software Security Case Development. *2017 IEEE 28th International Symposium on Software Reliability Engineering: Workshops (ISSREW)*, Toulouse, 23–26 October 2017. 2017.

147. Hu X., Zhuang Y., Zhang F. A security modeling and verification method of embedded software based on Z and MARTE. *Computers & Security*. 2020. Vol. 88. P. 101615.

148. Lugou F., Apvrille L., Francillon A. SMASHUP: a toolchain for unified verification of hardware/software co-designs. *Journal of Cryptographic Engineering*. 2016. Vol. 7, no. 1. P. 63–74.

149. Emeka B. O., Liu S. Assessing and extracting software security vulnerabilities in SOFL formal specifications. *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, Honolulu, HI, 24–27 January 2018. 2018.

150. Sedaghatbaf A., Azgomi M. A. Software Architecture Modeling and Evaluation Based on Stochastic Activity Networks. *Fundamentals of Software Engineering*. Cham, 2015. P. 46–53.

151. Koc G., Aydos M., Tekerek M. Evaluation of Trustworthy Scrum Employment for Agile Software Development based on the Views of Software

Developers. *2019 4th International Conference on Computer Science and Engineering (UBMK)*, Samsun, Turkey, 11–15 September 2019. 2019.

152. A new software failure analysis method based on the system reliability modeling / J. Song et al. *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, 24–26 May 2019. 2019.

153. Using Pattern Position Distribution for Software Failure Detection / C. Li et al. *International Journal of Computational Intelligence Systems*. 2013. Vol. 6, no. 2. P. 234–243.

154. Incorporating software failure in risk analysis – Part 1: Software functional failure mode classification / C. A. Thieme et al. *Reliability Engineering & System Safety*. 2020. Vol. 197. P. 106803.

155. Explaining Software Failures by Cascade Fault Localization / Q. Yi et al. *ACM Transactions on Design Automation of Electronic Systems*. 2015. Vol. 20, no. 3. P. 1–28.

156. DeStefano C., Jensen D. Failure Identification for Mission Analysis for Complex Systems. *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Boston, Massachusetts, USA, 2–5 August 2015. 2015.

157. Gao J., Wang H., Shen H. Task Failure Prediction in Cloud Data Centers Using Deep Learning. *IEEE Transactions on Services Computing*. 2020. P. 1.

158. Study on Software Vulnerability Characteristics and Its Identification Method / C. Luo et al. *Mathematical Problems in Engineering*. 2020. Vol. 2020. P. 1–6.

159. Information-theoretic Source Code Vulnerability Highlighting / V. Nguyen et al. *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 18–22 July 2021. 2021.

160. Pangr: A Behavior-Based Automatic Vulnerability Detection and Exploitation Framework / D. Liu et al. *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, New York, NY, USA, 1–3 August 2018. 2018.

161. Nguyen V. H., Dashevskyi S., Massacci F. An automatic method for assessing the versions affected by a vulnerability. *Empirical Software Engineering*. 2015. Vol. 21, no. 6. P. 2268–2297.

162. Yamaguchi F. Pattern-based methods for vulnerability discovery. *Information Technology*. 2017. Vol. 59, no. 2.

163. Wang J., Kuang H., Li R., Su Y. Software Source Code Vulnerability Detection Based on CNN-GAP Interpretability Model. *Journal of Electronics & Information Technology*. 2022. Vol. 44. Issue 7. P. 2568-2575.

164. VUDENC: Vulnerability Detection with Deep Learning on a Natural Codebase for Python / L. Wartschinski et al. *Information and Software Technology*. 2022. Vol. 144. P. 106809.

165. T. Hovorushchenko. Criteria and Rules for Classification of Software Failures and Vulnerabilities. *CEUR-WS*. 2021. Vol. 3039. Pp. 217-224.

ДОДАТОК А.
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Статті у періодичних виданнях, включених до категорії «А» Переліку наукових фахових видань України, або у закордонних виданнях, проіндексованих у базах даних Web of Science Core Collection та/або Scopus:

1. Hovorushchenko T., Medzatyi D., Voichur Yu., Lebiga M. Method for forecasting the level of software quality based on quality attributes. *Journal of Intelligent & Fuzzy Systems*. 2023. vol. 44, no. 3, pp. 3891-3905. (https://doi.org/10.3233/JIFS-222394) *(індексована в наукометричних базах Scopus, Web of Science (Q2 by Scimago Journal & Country Rank))*

2. Hovorushchenko T., Voichur Yu., Medzatyi D., Boyarchuk A. Information Technology for Prediction Software Quality Level. *Radioelectronic and Computer Systems*. 2023. No. 3. Pp. 238-254. (https://doi.org/10.32620/reks.2023.3.19) *(індексована в наукометричній базі Scopus (Q3 by Scimago Journal & Country Rank))*

3. E. Zaitseva, T. Hovorushchenko, O. Pavlova, Yu. Voichur. Identifying the Mutual Correlations and Evaluating the Weights of Factors and Consequences of Mobile Applications Insecurity. *Systems*. 2023. Vol. 11. Issue 5. Article No. 242. (https://doi.org/10.3390/systems11050242) *(індексована в наукометричній базі Scopus (Q3 by Scimago Journal & Country Rank))*

Статті у наукових виданнях, включених до Переліку наукових фахових видань України:

4. Медзатий Д.М., Войчур Ю.О., Войчур О.Ю. Технологія ідентифікації та класифікації відмов і вразливостей програмного забезпечення. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2023. №1. С. 53-57. (https://doi.org/10.31891/2219-9365-2023-73-1-8)

5. Ю. Войчур, Д. Медзатий. Метод аналізу вимог до програмного забезпечення на предмет пошуку значень атрибутів якості. Вимірювальна та обчислювальна техніка в технологічних процесах. 2024. №1. С.146-151. (https://doi.org/10.31891/2219-9365-2024-77-18)

Публікації, які засвідчують апробацію матеріалів дисертації:

6. Hovorushchenko T., Popov P., Medzatyi D., Voichur Yu. Method and Technology for Ensuring the Software Security by Identifying and Classifying the Failures and Vulnerabilities. *CEUR-WS*. 2022. Vol. 3309. Pp. 338-348. *(індексована в наукометричній базі Scopus)*

7. Hovorushchenko T., Voichur Yu. Method for Determining the Number of Lines of Manually Written Source Code. *CEUR-WS*. 2023. Vol. 3628. Pp. 520-525. *(індексована в наукометричній базі Scopus)*

8. T. Hovorushchenko, Yu. Voichur, D. Medzatyi, A. Boyarchuk, A. Hnatchuk. Method for Determining the Security Level of Software. *CEUR-WS*. 2024. Vol. 3675. Pp. 72-85. *(індексована в наукометричній базі Scopus)*

Публікації, які додатково відображають наукові результати дисертації:

9. А. с. 113734 Україна. Нейромережна модель прогнозування якості програмного забезпечення / Т. О. Говорущенко, М. М. Лебіга, Ю. О. Войчур. 2022.

10. А. с. 118851 Україна. Метод прогнозування рівня якості програмного забезпечення на основі атрибутів якості / Т. О. Говорущенко, М. М. Лебіга, Ю. О. Войчур. 2023.

ДОДАТОК Б.
АКТИ ВПРОВАДЖЕННЯ

«Затверджую»
Директор ПП «Авіві»
_____ Аскеров В.В.
«10» 01 2024 р.

АКТ
впровадження результатів дисертаційної роботи
«Методи і засоби прогнозування рівня якості та безпеки
програмного забезпечення комп'ютерних систем»
Войчура Юрія Олексійовича

Даним актом засвідчується впровадження результатів дисертаційної роботи здобувача наукового ступеня доктора філософії Ю. О. Войчуру на підприємстві ПП «Авіві», який проводив роботи з впровадження методів і засобів прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем, розробленого розробниками ПП «Авіві» на замовлення різних компаній м. Хмельницького.

В процесі розв'язання здобувачем науково-прикладної задачі прогнозування рівня якості та безпеки програмного забезпечення на ранніх етапах життєвого циклу на основі атрибутів якості, Войчуром Ю. О. були одержані особисто та використані на підприємстві ПП «Авіві» такі наукові та практичні результати:

1) метод аналізу вимог до програмного забезпечення комп'ютерних систем на предмет пошуку значень атрибутів якості, який забезпечує вибір значень атрибутів якості ПЗ з природомовної специфікації вимог до ПЗ, які використовуються для оцінювання значень характеристик якості ПЗ та для комплексного оцінювання якості ПЗ; розроблений метод є важливим для автоматизації опрацювання вимог та мінімізації суб'єктивного впливу і участі людини у процесах опрацювання інформації та здобуття знань;
2) метод прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості, який дозволяє порівнювати специфікації вимог до ПЗ, одразу відмовлятись від реалізації ПЗКС на основі невдалих специфікацій (економія коштів та часу, зменшення ймовірності провальних і проблемних проєктів) та виконувати обґрунтований вибір специфікації для подальшої реалізації ПЗКС саме високої якості (за умови, що помилки не будуть внесені на наступних етапах життєвого циклу ПЗ);
3) система прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості, яка забезпечує аналіз атрибутів якості у вимогах, відображає залежність характеристик якості від атрибутів, формує кількісну оцінку характеристик якості, відображає залежність якості від її характеристик, формує кількісну оцінку якості, виконує прогнозування рівня якості, надає всі перераховані сервіси одночасно, в комплексі, базується на спільних методологічних підходах.

«Затверджую»

Директор ТОВ «Деймос»

_____ Пантелєєв В.І.

«15» 02 2024 р.

АКТ

впровадження результатів дисертаційної роботи

Войчура Юрія Олексійовича

Акт складено в тому, що здобувач наукового ступеня доктора філософії Войчур Ю.О. впроваджував результати своєї дисертаційної роботи «Методи і засоби прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем» на ТОВ «Деймос», проводив роботи з впровадження методів та системи прогнозування рівня якості програмного забезпечення комп'ютерних систем (ПЗКС), розробленого різними ІТ-компаніями м. Хмельницького на замовлення ТОВ «Деймос».

Так, Войчуром Ю.О. були одержані особисто і використані на ТОВ «Деймос» такі результати наукових досліджень та практичні результати:

1) метод аналізу вимог до програмного забезпечення комп'ютерних систем на предмет пошуку значень атрибутів якості, який відрізняється від відомих накладанням певних обмежень на формування специфікації вимог до ПЗ шляхом структурування вимог, що містять атрибути якості, та забезпечує вибір значень атрибутів якості ПЗ з природомовної специфікації вимог до ПЗ;

2) метод прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості, який відрізняється від відомих тим, що дозволяє прогнозувати рівень якості майбутнього програмного забезпечення на основі опрацювання атрибутів якості ПЗ, доступних у специфікації вимог до ПЗ;

3) система прогнозування рівня якості ПЗКС на основі атрибутів якості, яка забезпечує аналіз атрибутів якості у вимогах, формує кількісну оцінку якості, виконує прогнозування рівня якості.

«Затверджую»

Голова
ГО «ІТ-КЛАСТЕР міста Хмельницького»

_____ С. А. Танасійчук

«_22_» ____03_____ 2024 р.

АКТ
впровадження результатів дисертаційної роботи
Войчура Юрія Олексійовича

Голова ГО «ІТ-КЛАСТЕР міста Хмельницького» Танасійчук С.А. склав цього акта про впровадження результатів дисертаційної роботи «Методи і засоби прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем» здобувача наукового ступеня доктора філософії Ю. О. Войчура у ГО «ІТ-КЛАСТЕР м. Хмельницького» у тому, що він проводив роботи з впровадження методів і засобів прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем, розробленого розробниками компаній, які входять до ІТ-Кластеру, на замовлення різних компаній м. Хмельницького.

В процесі розв'язання здобувачем науково-прикладної задачі прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем (ПЗКС) на ранніх етапах життєвого циклу на основі атрибутів якості, Войчуром Ю.О. були одержані особисто та використані у ГО «ІТ-КЛАСТЕР міста Хмельницького» такі наукові та практичні результати:

- метод забезпечення безпеки програмного забезпечення комп'ютерних систем шляхом ідентифікації та класифікації відмов і вразливостей, який забезпечує висновок щодо того, чи відбувалась відмова, і, якщо відмова відбулась, то користувачу видається її тип, а також забезпечує висновок щодо того, чи є функційна можливість вразливістю, і, якщо функційна можливість є вразливістю, то користувачу видається її тип;
- метод визначення рівня безпеки програмного забезпечення комп'ютерних систем, який забезпечує прогнозування рівня безпеки ПЗКС на основі отриманого числового значення, а також забезпечує порівняння специфікацій вимог до ПЗ за прогнозованим рівнем безпеки розроблюваного ПЗКС та можливість відбраковування невдалих специфікацій;
- розроблена система ідентифікації та класифікації відмов і вразливостей, яка надає висновок щодо наявності чи відсутності відмов(и) ПЗ; висновок щодо наявності чи відсутності вразливості(ей) ПЗ; висновок про тип відмови та тип вразливості в разі їх наявності, завдяки чому пропонована технологія є корисною для користувачів ПЗ за рахунок ідентифікації та класифікації відмов і вразливостей.

«Затверджую»
Проректор з наукової роботи
Хмельницького національного університету
д.т.н., професор
_____ Олег СИНЮК
«13» 05 2024 р.

АКТ

впровадження результатів дисертаційної роботи
«Методи і засоби прогнозування рівня якості та безпеки програмного забезпечення комп'ютерних систем»
Войчура Юрія Олексійовича

Комісія Хмельницького національного університету в складі: декана факультету інформаційних технологій, професора кафедри комп'ютерної інженерії та інформаційних систем, д.т.н., професора Олега Савенка, професора кафедри комп'ютерної інженерії та інформаційних систем, д.т.н., професора Сергія Лисенка, доцента кафедри комп'ютерної інженерії та інформаційних систем, к.т.н., доцента Єлизавети Гнатчук склали цього акта в тому, що результати дисертаційної роботи здобувача наукового ступеня доктора філософії Войчура Ю.О. впроваджені у навчальний процес на кафедрі комп'ютерної інженерії та інформаційних систем, зокрема, в навчальних дисциплінах: «Технології проєктування програмних систем», «Системна інженерія програмного забезпечення комп'ютерних систем», «Безпека та якість програмного забезпечення комп'ютерних систем».

При викладанні цих навчальних дисциплін використовувались і використовуються такі матеріали досліджень, проведених автором:

1) метод аналізу вимог до програмного забезпечення комп'ютерних систем (ПЗКС) на предмет пошуку значень атрибутів якості, який забезпечує вибір значень атрибутів якості ПЗ з природомовної специфікації вимог до ПЗ, які використовуються для оцінювання значень характеристик

якості ПЗ та для комплексного оцінювання якості ПЗ; розроблений метод є важливим для автоматизації опрацювання вимог;

2) метод прогнозування рівня якості програмного забезпечення комп'ютерних систем на основі атрибутів якості, дозволяє порівнювати специфікації вимог до ПЗ, одразу відмовлятись від реалізації ПЗКС на основі невдалих специфікацій та виконувати обґрунтований вибір специфікації для подальшої реалізації ПЗКС саме високої якості (за умови, що помилки не будуть внесені на наступних етапах життєвого циклу ПЗ);
3) метод забезпечення безпеки програмного забезпечення комп'ютерних систем шляхом ідентифікації та класифікації відмов і вразливостей, який забезпечує висновок щодо того, чи відбувалась відмова, і, якщо відмова відбулась, то користувачу видається її тип, а також висновок щодо того, чи є функційна можливість вразливістю, і, якщо функційна можливість є вразливістю, то користувачу видається її тип;
4) метод визначення рівня безпеки програмного забезпечення комп'ютерних систем, який забезпечує прогнозування рівня безпеки ПЗКС на основі отриманого числового значення, а також забезпечує порівняння специфікацій вимог до ПЗ за прогнозованим рівнем безпеки розроблюваного ПЗКС та можливість відбраковування невдалих специфікацій.

Використання зазначених результатів дозволили підвищити якість викладання зазначених навчальних дисциплін.

Олег САВЕНКО

Сергій ЛИСЕНКО

Єлизавета ГНАТЧУК

ДОДАТОК В.
АНАЛІЗ ВІДОМИХ МЕТОДІВ І ТЕХНОЛОГІЙ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

Таблиця В.1

Аналіз відомих методів і технологій забезпечення безпеки програмного забезпечення КС та виявлення збоїв і вразливостей ПЗ

Метод та/або технологія	**Забезпечення безпеки ПЗКС**	**Виявлення збоїв і вразливостей ПЗКС**
Метод оцінки безпеки ПЗКС на основі загроз з акцентом на існуючі загрози для об'єктів ПЗ [139]	+	
Метод підтвердження безпеки ПЗКС для забезпечення створення прийнятно безпечного для замовника ПЗ [140]	+	
Модель безпеки ПЗ на основі даних та методи вивчення детальної статистики ПЗ з одночасним забезпеченням диференційованої конфіденційності для його користувачів [141]	+	
Метод безпеки програмного забезпечення (CM-Sec), що фокусується на кінцевому продукті, який забезпечує розширення дерев атак та процес ідентифікації та визначення пріоритетності контрзаходів [142]	+	

Метод Q-навчання, вбудований як частина самого ПЗКС для забезпечення механізму безпеки [143]	+	
Методи, прийоми та найкращі практики інженерії та управління вимогами хмарних сервісів, що розвиваються (SSREMaaES), та керівництво з безпеки ПЗ як сервісу [144]	+	
Методологія мінімізації вразливостей ПЗ для підвищення його безпеки [145]	+	+
Ієрархічний метод розробки кейсів безпеки ПЗ [146]	+	
Фреймворк моделювання та верифікації безпеки вбудованого ПЗ на основі напівформальних та формальних методів ZMsec [147]	+	
SMASHUP: інструментарій для уніфікованої верифікації спільних проєктів ПЗКС [148]	+	
Метод виявлення вразливостей безпеки ПЗ у специфікаціях вимог до ПЗ, написаних структурованою об'єктно-орієнтованою формальною мовою [149]	+	
Формальний метод для моделювання архітектур програмного забезпечення та оцінки їх атрибутів якості (включаючи безпеку, надійність та продуктивність) кількісно та уніфіковано [150]	+	

Модель Trustworthy Scrum (TS), що дозволяє безпековій діяльності співпрацювати з гнучкими методами та працювати в рамках Scrum [151]	+	
Метод аналізу відмов ПЗКС на основі моделювання надійності системи за допомогою System-Theoretic Accident Modeling and Processes (STAMP) [152]	+	+
Використання розподілу позицій паттернів як ознак для виявлення відмов ПЗКС [153]		+
Таксономія для ідентифікації режимів відмов ПЗКС, які забезпечують вхідні дані для аналізу ризиків програмно-інтенсивних систем [154]		+
Метод каскадної локалізації несправностей та програмний інструмент CaFL для прискорення трудомісткого процесу ідентифікації першопричини проявленої несправності [155]	+	+
Метод Failure Identification for Complex Mission Analysis (FICMA), який забезпечує загальний аналіз відмов функціональності системи [156]	+	+
Алгоритм прогнозування відмов на основі багатошарової двонаправленої пам'яті [157]	+	+

Метод виявлення вразливостей потоку даних ПЗ на основі вдосконаленої згорткової нейронної мережі для ефективного виявлення помилок передачі в потоці даних ПЗ [158]	+	+
Метод виявлення та ізоляції вразливостей ПЗ як у неконтрольованому, так і в напівконтрольованому контекстах [159]	+	+
Pangr: система для автоматичного виявлення, експлуатації та виправлення вразливостей [160]		+
Автоматизований метод визначення кодових ознак наявності вразливостей у застарілих версіях ПЗКС [161]	+	+
Підхід до виявлення вразливостей на основі патернів [162]	+	+
Метод виявлення вразливостей у сирцевому коді ПЗКС на основі моделі інтерпретованості Convolution Neural Networks (CNN) та Global Average Pooling (GAP) [163]	+	+
VUDENC (Vulnerability Detection with Deep Learning on a Natural Codebase): інструмент виявлення вразливостей на основі глибокого навчання [164]	+	+

ДОДАТОК Д.
ПРИКЛАДИ СПЕЦИФІКАЦІЇ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Фрагмент однієї з розглянутих специфікацій з накладанням певних обмежень шляхом структурування вимог, які містять атрибути якості:

…

кількість завдань 100;

час відгуку 0.1;

кількість оцінок 7;

час виконання 10;

час виконання завдання 1;

середня пропускна здатність 12;

кількість відмов 2;

кількість помилок, пов'язаних з вводом/виводом 1;

час очікування користувачем використання пристрою вводу/виводу 0.5;

кількість помилок, пов'язаних з пам'яттю 1;

кількість помилок, пов'язаних з передачею даних 2;

пропускна здатність каналу передачі даних 15;

завантаженість вводу/виводу (кількість буферів) 3;

кількість рядків безпосередньо коду 10000;

…

Фрагмент однієї з розглянутих специфікацій з накладанням певних обмежень шляхом структурування вимог, які містять атрибути якості, від яких залежить безпека ПЗКС:

…

кількість випадків пошкодження даних 1;

кількість типів доступу 5;

кількість контрольованих вимог 234;

контрольованість доступу 89;

кількість правильно зашифрованих/розшифрованих елементів даних 128;

кількість елементів даних, що потребують шифрування/розшифрування 154;

кількість подій, що обробляються з використанням цифрового підпису 3;

…